

## ***PAMtutorials* 12: More on Concurrent Managers**

**Proficiently psychic...(Is bad stuff going on?)**

**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> 12.....	4
2	CP-001 Scheduled Maintenance Programs .....	6
2.1	Monitored Programs .....	6
2.2	<i>PAM</i> referential cleanup.....	7
2.3	Viewing the <i>PAM</i> monitored programs list .....	8
2.4	<i>PAM</i> missing program e-mail alert.....	9
2.5	What to do with this information .....	10
2.5.1	Where the program being monitored is a purge program.....	10
2.5.2	Plan your purge .....	13
2.6	How do I turn the <i>PAM</i> program monitor alert off and on again? .....	15
2.7	Changing alert check frequency and / or severity .....	15
3	CP-002 - Monitored Programs that Complete with Error or Warning.....	16
3.1	Viewing the <i>PAM</i> program monitored programs list.....	17
3.2	<i>PAM</i> Complete Error / Warning e-mail alert.....	17
3.3	What to do with this information .....	19
3.3.1	Complete error .....	19
3.3.2	Completed Warning.....	20
3.3.3	Request set stages.....	21
3.4	How do I turn the <i>PAM</i> completed error / warning alert off and on again? .....	22
3.5	Changing alert check frequency and / or severity .....	22
4	CP-003 Submitted Program Alert .....	23
4.1	Viewing the <i>PAM</i> program monitored programs list.....	23
4.2	<i>PAM</i> submitted concurrent program e-mail alert.....	24
4.3	What to do with this information .....	25
4.4	How do I turn the <i>PAM</i> monitored program - submitted requests alert off and on again?.....	25
4.5	Changing alert check frequency and / or severity .....	26
5	CR-005 Purge Sign-on audit date increment may not be incrementing...	27
5.1	<i>PAM</i> Sign-on audit date increment e-mail alert.....	27
5.2	What to do with this information .....	28
5.3	Turning CR-005 alert off and on .....	28
6	CR-001 Aged On-hold .....	29
6.1	Changing the aged on-hold threshold months .....	29
6.2	<i>PAM</i> Aged on-hold requests e-mail alert .....	29
6.3	What to do with this information .....	31
6.4	How do I turn the <i>PAM</i> aged on-hold requests alert off and on again? .....	32

6.5	Changing alert check frequency and / or severity .....	32
7	Updating <i>PAM</i> Program Exceptions .....	33
7.1	Adding a program to <i>PAM</i> monitored programs exceptions.....	33
7.1.1	Deleting a program from the <i>PAM</i> exceptions.....	34
7.1.2	Changing <i>PAM</i> monitored programs items.....	35
7.1.3	Referential Integrity .....	35
8	How to find a concurrent program ID .....	36
9	<i>PAM</i> Internal Checks .....	38
9.1	IN-011 Invalid <i>PAM</i> Package Check.....	38
9.1.1	Turning the <i>PAM</i> internal invalid package alert feature on or off	40
9.1.2	Resetting an Invalid Object Disabled Alert .....	40
9.2	Internal <i>PAM</i> Error Check (IN-012).....	41
9.2.1	Changing the <i>PAM</i> IN-012 error threshold .....	42
9.2.2	Turning the <i>PAM</i> internal package error alert feature on or off ..	42
9.2.3	Resetting a Suspended Alert.....	43
10	<i>PAM</i> Alert e-mail links .....	44
10.1	Turning the <i>PAM</i> alert e-mail links feature on or off.....	44
10.2	Listing current e-mail links .....	45
10.3	Link Format.....	45
10.4	Turning individual links on and off .....	46
10.5	Deleting individual links .....	46
10.6	Creating a <i>PAM</i> e-mail link .....	47
10.7	Updating <i>PAM</i> alert e-mail links.....	47
11	Disclaimer.....	48

## 1 What you'll get out of *PAMtutorials* 12

In *PAMtutorials* 12 we will show how *PAM*:

- ❖ Monitors once per day for scheduled maintenance concurrent programs that have been placed on-hold or have been cancelled (CP-001)
- ❖ Monitors every 5 minutes for selected concurrent programs that complete with a status of error or warning (CP-002)
- ❖ Monitors every 5 minutes for selected concurrent programs that have been submitted (CP-003)
- ❖ Monitors for programs that have been on-hold for over 1 month (CR-001)

In addition we cover two (2) *PAM* internal checks:

### *PAM* package check (IN-010)

Once per hour (default) *PAM* will check each of its packages to determine if any of its packages have become invalid. Where a *PAM* package has been found to be invalid, *PAM* will “auto disable” any *PAM* checks associated with the invalid package until, on a subsequent check, *PAM* detects that the package has been re-compiled and is valid. When this occurs the *PAM* checks will be re-enabled. This internal check is designed to limit the number of *PAM* errors being generated as a result of invalid *PAM* objects.

### Invalid *PAM* error check (IN-012)

Once per hour (default) *PAM* will check the number of *PAM* errors being generated. When the number of errors exceeds the *PAM* threshold, the *PAM* check generating the errors will be “auto suspended” for the remainder of the day.

## Customising *PAM*

In this tutorial we will show you how to customise the links section at the bottom of each *PAM* alert e-mail. This is the first time the customisation feature has been described in the *PAMtutorials*; subsequent *PAMtutorials* will cover more *PAM* customisation features.

## 2 CP-001 Scheduled Maintenance Programs

There are a number of concurrent programs that should be run on a scheduled basis as part of your routine maintenance program and for the normal operation of the business; the most obvious is the Purge Concurrent requests program.

I have encountered numerous instances where an Applications Administrator believed that the purge concurrent requests program was running when it had in fact been either cancelled or placed on-hold during some maintenance activity and had not been released from hold and consequently had not been running for some months. In one case the program had not been running for over a year and in this instance the size and performance of the [fnd\\_concurrent\\_requests](#) table was severely affected, which in turn affected the overall performance of the application. Remember over 90% of all application activity is via the concurrent managers.

### 2.1 Monitored Programs

Concurrent programs to be monitored are stored in the **PAM** [piper\\_rx\\_pam\\_cp\\_monitor\\_tl](#) object. We have already introduced the **PAM** program exception object [piper\\_rx\\_pam\\_cp\\_monitor\\_tl](#) in **PAMtutorials 7** (Managing user behavior) as part of the **PAM** duplicates (CP-004) and long running requests (CM-003) checks.

Out of the box **PAM** will monitor for the following concurrent programs:

- ❖ Purge Concurrent Requests
- ❖ Purge Sign-on Audit Data
- ❖ GL Program Optimiser
- ❖ Purge Workflows
- ❖ Purge OAM log
- ❖ Purge obsolete generic file manager data
- ❖ FNDPGHST Purge FND\_STATS History Records (added in later versions of 11i and in 12i )
- ❖ Purge Self Service ( 11i ICXDLTMP )
- ❖ Purge Self Service (12i FNDDL TMP )

**Note:** Some of the above mentioned programs are OEBS version dependent. To deal with this, as part of the **software pack install process only**

*PAM* will remove any program from the *PAM* monitored program list that does not exist in your version of the OEBS application. In the case of the General Ledger Program Optimiser program, during the install process *PAM* will check for the existence of any GL FSG or GL posting programs that have been run in the available online concurrent request activity. If none are found the GL Program Optimiser program will be removed from the *PAM* monitored program list.

## 2.2 *PAM* referential cleanup

After a period of time you may have a number of programs in the *PAM* monitored programs exceptions list that no longer exist. The following *PAM* API performs a *PAM* referential cleanup which will remove any programs from the *PAM* exceptions list that no longer exist in the OEBS application.

```
exec PIPER_RX_PAM_API.PAM_REFERENTIAL_CLEANUP;
```

As the name of this API implies, it removes items that no longer exist in the OEBS application from the following *PAM* tables:

- ❖ [piper\\_rx\\_pam\\_cp\\_monitor\\_tl](#)
- ❖ [piper\\_rx\\_pam\\_db\\_object\\_ex](#)
- ❖ [piper\\_rx\\_pam\\_lr\\_workflows](#)
- ❖ [piper\\_rx\\_pam\\_account\\_ex](#)
- ❖ [piper\\_rx\\_pam\\_conc\\_mgr\\_ex](#)

## 2.3 Viewing the PAM monitored programs list

You can use **PAMreports** - Config PAMC009 PAM Program Monitor List report to list all **PAM** concurrent program exceptions which includes monitored programs:

Example PAMC009 PAM Program Monitor List report

PAMC009-20		PIPER-RX - APPLICATION MONITOR PAM Concurrent Program Monitor List As at 28-Feb-11 08:13:59 For APPS 12i						PIPER - Rx	
Application	Prog Id	Program Name	Check Status	Exists	Completed		Submitted	Exclude Duplicates	Exclude Long Running
					Error	Warning			
0 (FND)	0	Activate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	1	Deactivate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	3	Restart Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	4	Abort Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	5	Shutdown Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	6	Startup Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	31659	Report Set	Enabled	No	No	No	No	No	Yes
0 (FND)	32263	Purge Concurrent Request and/or Manager Data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	32592	Purge Signon Audit data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	36034	Request Set Stage	Enabled	No	No	No	No	Yes	Yes
0 (FND)	36888	Workflow Background Process	Enabled	No	No	No	No	Yes	Yes
178 (ICX)	36662	Delete data from temporary table	Enabled	Yes	No	No	No	No	Yes

Adding and removing programs from the **PAM** program exception table will be covered at the end of this tutorial.



## 2.4 PAM missing program e-mail alert

When a **PAM** monitored concurrent program cannot be found in either a pending, pending scheduled or running state or, that program has been placed on-hold, a **PAM** alert e-mail is raised:

### Example **PAM** CP-001 – **PAM** missing program 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 = 23-Feb-11 (Wed) 08:10:52  
Alert Frequency = 1 Day

---

**The concurrent program - Purge Obsolete Generic File Manager Data - could not be found in either a running or pending state**

---

#### **Alert Information:**

#### **CP-001 - Monitored Program not found or is On-hold**

**A MONITORED CONCURRENT PROGRAM COULD NOT BE FOUND IN EITHER A RUNNING OR PENDING STATE AND MAY HAVE BEEN DELETED OR PLACED ON-HOLD.**

If you want to view the current list of monitored concurrent programs run **PAMreports** - Config **PAMC009 PAM Program Monitor List**

If a monitored program is On-hold and you want to obtain more information on the program run **PAMreports** - Actions **PAMACP001 Monitored programs On-hold**

If you expected a monitored program to exist and it does not, it may have been deleted. Look in the **fnd\_concurrent\_requests** table for the last occurrence of the program. If the program exists in your on-line history this will let you know when the program was last run

**Note 1:** This alert will continue to repeat until the selected program can be found in a running or pending normal state

**Note 2:** If you want to add or change the programs that are monitored refer to the FAQs for more information

## 2.5 What to do with this information

First check to see if the program has been placed on-hold and not released. This is commonly found after application maintenance activity where scheduled requests are placed on-hold for the duration of the maintenance activity.

You can use **PAMreports** - Actions **PAMACP001 Monitored Programs On-hold** report to list monitored programs that exist but are on-hold:

### Example PAMACP001 Monitored Programs On-hold report

PAMACP001-20				
PAM - PIPER-RX - APPLICATION MONITOR Program Monitor - Exists / On-Hold Check As at 28-Feb-11 08:19:32 For APPS 12i				
Request ID	Requestor	Program Name	Arguments	Requested Start
306319	GPIPER	Purge Signon Audit data	26-JAN-11	27-Feb-11 (Sun) 22:00

### Example PAMACP001 Monitored Programs On-hold report where no monitored programs have been found to be on-hold

PAMACP001-20				
PAM - PIPER-RX - APPLICATION MONITOR Program Monitor - Exists / On-Hold Check As at 28-Feb-11 08:19:54 For APPS 12i				
Request ID	Requestor	Program Name	Arguments	Requested Start
No Programs Found		No Programs Found		

**Note:** If you have several purge programs in existence, as may be the case with workflow purging, **PAM** will alert if one or more of these programs has been placed on-hold.

It may just be a simple case of releasing the program from hold. However, if the program you are releasing has been on-hold for some time or that program has been placed on-hold for business reasons, releasing that program may cause more issues than you expect. You should always carefully assess the impact of releasing a program from hold regardless of whether you placed it on-hold in the first place or not.

### 2.5.1 Where the program being monitored is a purge program

If the program is a purge program and has been on-hold for a long period of time or the program has never been run, you could do a lot of damage by just running the program given the volume of data you may be purging.

### 2.5.1.1 Assess the potential for damage before you act

If you have just installed **PAM** and get one or more alerts stating one or more **PAM** monitored programs cannot be found, please **DON'T** just run the program. In a recent case with `fnl_lobs` not being purged, the `fnl_lobs` table was over 30 Gb; just running the purge would cause all sorts of problems with such things as rollback segments.

First check the size of the table, the number of rows and if possible the age of records in the table; this will give you some idea of the extent of the potential damage.

To illustrate, the following examples use the `icx_sessions` objects which is purged via the OEBS 'Delete Data from Temporary Table' program.

Firstly, a simple count of records will let you know if there is a major problem; the more rows the bigger the cleanup:

```
SELECT count(*) FROM table_name;
```

Identifying the spread of data in the table to be purged may help in running purge programs that selectively purge old data. The following SQL will show the spread of data in the `icx_sessions` table:

```
SELECT to_char(first_connect, 'YYYY-MM') sort_column,  
       to_char(first_connect, 'YYYY-Mon') year_month,  
       count(*) records  
FROM icx.icx_sessions  
GROUP by to_char(first_connect, 'YYYY-MM'),  
         to_char(first_connect, 'YYYY-Mon')  
ORDER by to_char(first_connect, 'YYYY-MM');
```

**Note:** You will need to pick a suitable date column in the table of interest

In the case of purging self-service data, one of the program runtime parameters is the purge date. In this case you can run the program once for records older than 12 months, then another for records older than 11 months and so on... until you have purged all the appropriate data. Remember to keep 32 days on-line if possible.

The next step is to assess the size of the object being purged; you may need the DBA to rebuild the table object once it has been purged for the “normal” number of records (size) you expect in the table.

The current size can be found using the following SQL:

```
SELECT (bytes /1024/1024) allocated_mb
FROM dba_segments
WHERE owner = 'ICX'
and segment_name = 'ICX_SESSIONS';
```

It is a good idea not to rely on the databases analysed stats for this information as these may have not been collected or could be out of date.

The next step is to identify the number of indexes that will need to be rebuilt after a major purge.

```
SELECT owner,
index_name
FROM dba_indexes
WHERE table_owner = 'ICX'
and table_name = 'ICX_SESSIONS';
```

Once any major purge has been completed the indexes on the purged object should be rebuilt.

**Warning:** Some indexes may be partitioned indexes as is the case of index [wf\\_item\\_activity\\_statuses\\_pk](#) on the workflow OEBS table [wf\\_item\\_activity\\_statuses](#)

You may have already received a **PAM** index rebuild reminder for a number of indexes on purgable tables as part of the **PAM** Index rebuild reminder alert (DB-002) which was covered in **PAMtutorials** 2. A full list of indexes being monitored by **PAM** can be found using **PAMreports** - Config [PAMC016 PAM DB Object Exceptions](#):

Example **PAMC016 PAM DB Object Exceptions** report

PAMC016-20		PAM - PIPER-RX - APPLICATION MONITOR		PIPER - Rx		
Database Object Check - Exclusions / Inclusions						
As at 28-Feb-11 08:23:46						
For APPS 12i						
Object Owner	Object Name	Overall Status	Exclude From Invalid	Include Index Rebuild	Index Rebuild Months	
APPLSYS	FND_CONCURRENT_REQUESTS_U1	Enabled	No	Yes	6	
APPLSYS	FND_LOGINS_U1	Enabled	No	Yes	6	
APPLSYS	FND_LOGIN_RESPONSIBILITIES_U1	Enabled	No	Yes	6	
APPLSYS	FND_LOGIN_RESP_FORMS_N1	Enabled	No	Yes	6	
APPLSYS	WF_ITEM_ACTIVITY_STATUSES_H_N1	Enabled	No	Yes	6	
APPLSYS	WF_ITEM_ACTIVITY_STATUSES_PK	Enabled	No	Yes	6	
ICX	ICX_SESSIONS_U1	Enabled	No	Yes	6	

Any number of additional indexes can be added to this alert. Refer to [PAMtutorials 2](#) for full details.

**Note:** In the case of purging [icx\\_sessions](#), refer to the [piper-rx](#) paper [How to purge self service sessions](#) before running the purge program.

There are of course other ways of finding out this information so you should consult your DBA for alternate ideas.

### 2.5.2 Plan your purge

The following steps are not by any stretch of the imagination meant as a guide, but merely an indication as to the complexity of performing a simple purge on tables that have not been purged in some time, like the 32 Gb [fnd\\_lob](#)s table example mentioned above.

As expected, with anything OEBS, it is not a simple case of just running a purge. In many cases more than one table is purged as part of the single purge program. In the case of the self-service purge, the following tables are purged: [icx\\_sessions](#), [icx\\_session\\_attributes](#), [icx\\_transactions](#), [icx\\_text](#), [icx\\_context\\_results\\_temp](#), [icx\\_failures](#), [icx\\_requisitioner\\_info](#), [fnd\\_session\\_values](#) (tables are OEBS version dependent).

Normally you would expect the base table to have the largest number of rows. Again, in the case of self-service purging you would expect the table [icx\\_sessions](#) to have the most rows to purge. However I have encountered very large numbers of rows in the [icx\\_text](#) table; in one case a ratio of 5:1 was found.

## Find out what objects are purged

What you will need to do here is find and review the code that performs the purge. This can be a simple process if the program being called is a piece of SQL or unwrapped PLSQL, however in some cases we may not be that lucky.... It is a good idea to assign this task to an experienced Apps DBA or an Apps developer.

## Calculate the size of the purgeable table/s for normal activity

You can find out how many rows per month you have typically been adding to the table from the following SQL:

```
SELECT to_char(first_connect, 'YYYY-MM') sort_column,
       to_char(first_connect, 'YYYY-Mon') year_month,
       count(*) records
FROM   icx.icx_sessions
GROUP by to_char(first_connect, 'YYYY-MM'),
         to_char(first_connect, 'YYYY-Mon')
ORDER by to_char(first_connect, 'YYYY-MM');
```

Your DBA can work out the required table size to hold 1 to 2 months data, thus once you have run the purge your DBA will know what size to resize the table to, thus reclaiming any excess space.

You will find several instances where the target table does not have a date value to work with, as is the case with [icx\\_session\\_attributes](#). In this case you will need to estimate the table size based on the base table rows in the main purgeable table, in this case [icx\\_sessions](#).

## Purge the table

If the table/s have never been purged or has not been purged for some time it may not be possible to run the normal maintenance purge program, as there may be too many rows to delete in a single pass. What you may need to do is run several purge programs deleting much older records first. In the case of purging self-service data, one of the program runtime parameters is the purge date. In this case you can run the program once for records older than 12 months, then another for records older than 11 months and so on... until you have purged all the data necessary. Remember to keep 32 days on-line if possible.

## Rebuild the table to reclaim the space

This is a task for your DBA, (they love this kind of stuff). Extreme care should be taken when rebuilding OEBS tables given the large number of interdependencies in the application. Get it wrong and you will be surprised at the number of invalid object you will generate...

## Rebuild the indexes

The next step is to rebuild the indexes on the target tables; this is a relatively simple process, again one that should be completed by your DBA. TOAD provides a simple, effective method for rebuilding indexes, including partitioned indexes.

## Final steps

Analyse the purged tables to bring the application stats up to date and implement a normal maintenance purge run.

## 2.6 How do I turn the *PAM* program monitor alert off and on again?

The *PAM* program monitor alert can be turned off using the following *PAM* API:

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

and can be re-enabled using the following *PAM* API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CP-001', 'Y' );
```

## 2.7 Changing alert check frequency and / or severity

Both the alert check frequency and alert severity can be changed. Please refer to the *PAM* FAQs for more information on how to change an alert frequency and alert severity.

### 3 CP-002 - Monitored Programs that Complete with Error or Warning

**PAM** will check every 5 minutes (default) for any monitored concurrent program that completes with a status of either Error or Warning.

<p><b>Note:</b> One alert will be generated per concurrent request that completes with a status of either error or warning.</p>
---

There are any number of programs that, should they fail, require immediate action and cleanup.

A simple example of the power of this check can be demonstrated by considering its usefulness with the application's statements run. The statement run can be quite resource intensive and can take some time to complete. If it was to fail, wouldn't it be good to be notified when it fails, rather than wait until the user gets round to noticing that the statements have not started printing yet.

You can also quickly be alerted to the failure of a printer with reports failing with a status of warning.



### 3.1 Viewing the PAM program monitored programs list

You can use **PAMreports** - Config **PAMC009 PAM Program Monitor List** to list all **PAM** concurrent program exceptions which includes monitored programs that complete with a status of error or warning:

Example **PAMC009 Program Monitor List** report

PAMC009-20 PIPER-RX - APPLICATION MONITOR PAM Concurrent Program Monitor List As at 28-Feb-11 09:37:32 For APPS 12i PIPER - Rx									
Application	Prog Id	Program Name	Check Status	Completed			Submitted	Exclude Duplicates	Exclude Long Running
				Exists	Error	Warning			
0 (FND)	0	Activate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	1	Deactivate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	3	Restart Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	4	Abort Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	5	Shutdown Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	6	Startup Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	31659	Report Set	Enabled	No	No	No	No	No	Yes
0 (FND)	32263	Purge Concurrent Request and/or Manager Data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	32592	Purge Signon Audit data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	36034	Request Set Stage	Enabled	No	No	No	No	Yes	Yes
0 (FND)	36888	Workflow Background Process	Enabled	No	No	No	No	Yes	Yes
0 (FND)	37815	PIPER-RX Pam Long running request	Enabled	No	Yes	Yes	No	No	No
178 (ICX)	36662	Delete data from temporary table	Enabled	Yes	No	No	No	No	Yes

Adding and removing programs from the **PAM** program exception table will be covered at the end of this tutorial.

### 3.2 PAM Complete Error / Warning e-mail alert

When a **PAM** monitored concurrent program completes with a status or error or warning a **PAM** alert e-mail is raised:

Example **PAM CP-002 – PAM Completed error 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 = 28-Feb-11 (Mon) 09:40:02  
 Alert Frequency = 5 Minutes

**Program PIPER-RX Pam Long running request - Request ID (306342) complete with a status of error**

---

**Alert Information:**

**CP-002 - Monitored program completed with status of Error or Warning**

**A CONCURRENT PROGRAM THAT YOU HAVE SELECTED TO MONITOR FOR COMPLETION WITH EITHER AN ERROR OR WARNING STATUS HAS COMPLETED WITH A STATUS OF ERROR OR WARNING.**

If you want to obtain a list of the program/s completed with a status of error run *PAMreports* - Actions **PAMACR001 Completed Error (day)**

If you want to obtain a list of the program/s completed with a status of warning run *PAMreports* - Actions **PAMACR002 Completed Warning (day)**

If you want to obtain a list of the current list of monitored concurrent programs you can run *PAMreports* - Config **PAMC009 PAM Program Monitor List**

**Note 1:** If you want to add or change the programs that are monitored refer to the FAQs for more information

**Example PAM CP-002 – PAM Completed warning 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 = 28-Feb-10 (Mon) 09:40:04  
Alert Frequency = 5 Minutes

---

**Program PIPER-RX Pam Long running request - Request ID (2172205) complete with a status of warning**

---

**Alert Information:**

**CP-002 - Monitored program completed with status of Error or Warning**

**A CONCURRENT PROGRAM THAT YOU HAVE SELECTED TO MONITOR FOR COMPLETION WITH EITHER AN ERROR OR WARNING STATUS HAS COMPLETED WITH A STATUS OF ERROR OR WARNING.**

If you want to obtain a list of the program/s completed with a status of error run *PAMreports* - Actions **PAMACR001 Completed Error (day)**

If you want to obtain a list of the program/s completed with a status of warning run *PAMreports* - Actions **PAMACR002 Completed Warning (day)**

If you want to obtain a list of the current list of monitored concurrent programs you can run *PAMreports* - Config **PAMC009 PAM Program Monitor List**

**Note 1:** If you want to add or change the programs that are monitored refer to the FAQs for more information

### 3.3 What to do with this information

#### 3.3.1 Complete error

A concurrent request that completes with a status of error usually indicates the concurrent program encountered an error during program execution.

For the technically minded, the `fnd_concurrent_request.status_code = 'E'`

For a list of concurrent requests that completed with a status of error for any given day you can use *PAMreports* - Actions **PAMACR001 Completed Error (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 **PAMACR001 Completed Error (day)** report

PAMACR001-20				PAM - PIPER-RX - APPLICATION MONITOR		PIPER - Rx
Completed Error Requests - For 28-FEB-11 (Mon)						
As at 28-Feb-11 09:40:00						
For APPS 12I						
Request ID	Requestor Argument	Program Name	Start Date	Completion Text		
306342	GPIPER	PIPER-RX Pam Long running request	28-Feb-11 09:35	None		
				Concurrent Manager encountered an error while running SQL*Plus for your concurrent request 306342.		
306343	GPIPER	PIPER-RX Pam Long running request	28-Feb-11 09:35	None		
				Concurrent Manager encountered an error while running SQL*Plus for your concurrent request 306343.		
306344	GPIPER	PIPER-RX Pam Long running request	28-Feb-11 09:35	None		
				Concurrent Manager encountered an error while running SQL*Plus for your concurrent request 306344.		

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

**3.3.2 Completed Warning**

A concurrent request that completes with a status of warning usually indicates the concurrent program completed normally but encountered an error when being sent to the designated printer.

For the technically minded, the `fnd_concurrent_request.status_code = 'G'`

For a list of concurrent requests that completed with a status of warning for any given day you can use **PAMreports** - Actions **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 report **PAM** displays the target printer to aid in the identification of the issue. 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...

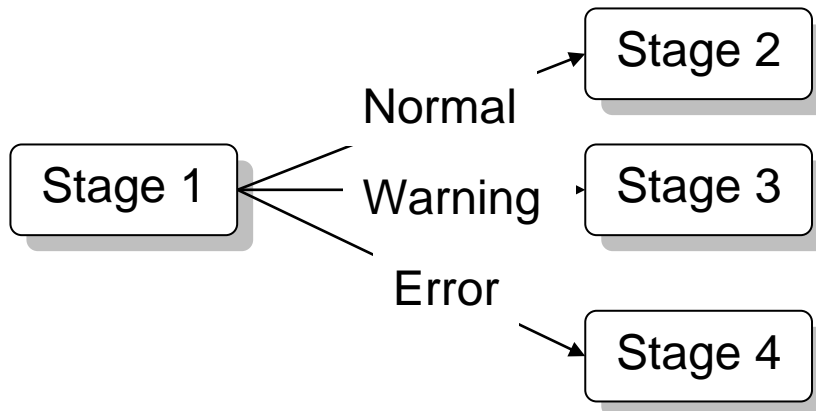
Once the printer issue has been rectified, it is better for the user to reprint the request rather than run it again which uses up more resources to re generate the report.

### 3.3.3 Request set stages

“Request set stages” are a special case and should not be included in the **PAM** completed error / warning check.

Out of the box **PAM** excludes “Reports Set Stages” from this check as these programs are basically control programs that are designed, as part of normal behavior, to complete with a status of error, warning and normal.

A request set stage can finish with a status of normal, warning or error. The completion status of a stage will determine which of the next stages to run. In the example below if stage 1 completes normal then stage 2 is processed but if stage 1 fails (error) stage 4 is processed.



As it is normal behavior to see stages complete with a status of error or warning **PAM** excludes these from the alert check.

Example **PAMC009 Program Monitor List** report

PAMC009-20 <span style="float: right;">PIPER - Rx</span> PIPER-RX - APPLICATION MONITOR PAM Concurrent Program Monitor List As at 28-Feb-11 09:44:30 For APPS 12i									
Application	Prog Id	Program Name	Check Status	Exists	Completed		Submitted	Exclude Duplicates	Exclude Long Running
					Error	Warning			
0 (FND)	0	Activate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	1	Deactivate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	3	Restart Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	4	Abort Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	5	Shutdown Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	6	Startup Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	31659	Report Set	Enabled	No	No	No	No	No	Yes
0 (FND)	32263	Purge Concurrent Request and/or Manager Data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	32592	Purge Signon Audit data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	36034	Request Set Stage	Enabled	No	No	No	No	Yes	Yes
0 (FND)	36888	Workflow Background Process	Enabled	No	No	No	No	Yes	Yes
0 (FND)	37815	PIPER-RX Pam Long running request	Enabled	No	Yes	Yes	No	No	No
178 (ICX)	36662	Delete data from temporary table	Enabled	Yes	No	No	No	No	Yes

Out of the box **PAM** has excluded report set stages from both the completed error and warning checks. In addition report sets have also been excluded from the **PAM** duplicate requests and long running request checks. This is because there will be duplicate request set stages and request set stages run times are variable as they are dependent on the concurrent programs run within the stage.

### 3.4 How do I turn the **PAM** completed error / warning alert off and on again?

The **PAM** completed error / warning alert can be turned off using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CP-002', 'N' );
```

and can be re-enabled using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CP-002', 'Y' );
```

### 3.5 Changing alert check frequency and / or severity

Both the alert check frequency and alert severity can be changed. Please refer to the **PAM** FAQs for more information on how to change an alert frequency and alert severity.

## 4 CP-003 Submitted Program Alert

Every 5 minutes (default) **PAM** will check for selected concurrent programs that have been submitted. Out of the box **PAM** will check for and alert if any of the concurrent manager control programs have been run and you will be alerted if the concurrent managers have been shut down.

**Note:** Whilst all concurrent manager control programs have been added to the **PAM** program watch list not all manager control programs are run via or recorded as concurrent requests. As such, alerts will not be raised for these programs e.g. starting the internal manager.

You can add any number of your own programs to be monitored by **PAM**. You may wish to include known “resource hog” programs in this check so you are aware when they are running (**to be forewarned is to be for forearmed**).

Adding and removing programs from the **PAM** program exception table will be covered at the end of this tutorial.

### 4.1 Viewing the **PAM** program monitored programs list

You can use **PAMreports** - Config **PAMC009 PAM Program Monitor List** to list all **PAM** concurrent program exceptions which includes monitored programs that are to alert when submitted:

Example **PAMC009 Program Monitor List** report

PAMC009-20		PIPER-RX - APPLICATION MONITOR PAM Concurrent Program Monitor List As at 28-Feb-11 09:44:30 For APPS 12i						PIPER - Rx	
Application	Prog Id	Program Name	Check Status	Exists	Completed			Exclude Duplicates	Exclude Long Running
					Error	Warning	Submitted		
0 (FND)	0	Activate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	1	Deactivate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	3	Restart Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	4	Abort Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	5	Shutdown Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	6	Startup Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes
0 (FND)	31659	Report Set	Enabled	No	No	No	No	No	Yes
0 (FND)	32263	Purge Concurrent Request and/or Manager Data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	32592	Purge Signon Audit data	Enabled	Yes	No	No	No	No	Yes
0 (FND)	36034	Request Set Stage	Enabled	No	No	No	No	Yes	Yes
0 (FND)	36888	Workflow Background Process	Enabled	No	No	No	No	Yes	Yes
0 (FND)	37815	PIPER-RX Pam Long running request	Enabled	No	Yes	Yes	No	No	No
178 (ICX)	36662	Delete data from temporary table	Enabled	Yes	No	No	No	No	Yes

Adding and removing programs from the **PAM** program exception table will be covered at the end of this tutorial.

## 4.2 *PAM* submitted concurrent program e-mail alert

When a *PAM* monitored concurrent program is submitted a *PAM* alert e-mail is raised:

### Example *PAM* CP-003 – *PAM* Submitted Program alert message

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

Company = Company name  
Site = Site name  
Alert Level = **Informational**  
Detected = 28-Feb-11 (Mon) 09:00:17  
Alert Frequency = 5 Minutes

---

**Program Deactivate Concurrent Manager Request ID (2172259) By SYSADMIN has been submitted at 07-Jun-10 19:49**

---

#### **Alert Information:**

**CP-003 - Monitored Program submitted**

**A CONCURRENT PROGRAM THAT YOU HAVE SELECTED TO BE MONITORED FOR SUBMISSION WAS DETECTED.**

If you want to obtain the list of *PAM* monitored programs you can use *PAMreports* - Config **PAMC009 PAM Program Monitor List**

**Note 1:** If you want to add or change the programs that are monitored refer to the FAQs for more information



### 4.3 What to do with this information

As mentioned in the opening paragraph for this section **“to be forewarned is to be for forearmed”**. In the above example we are notified that the concurrent managers have been shut down so we know when and by whom...

If you are monitoring “resource hog” concurrent programs that you have ascertained that the business does not have to run during peak user load times, you will be alerted when that program has been submitted and can take the appropriate action. “Seek out the guilty” ☺

You can use **PAMreports** - General **PAMRCP004 Monitored Programs Submitted** to list all **PAM** monitored programs that have been submitted:

Example **PAMRCP004 Monitored Programs Submitted** report

PAMRCP004.20		PAM - PIPER-RX - APPLICATION MONITOR Monitored Programs - Submitted As at 28-Feb-11 09:50 For APPS 12i						PIPER - Rx	
Request Id	Submitted By	Phase	Status	Scheduled	Resubmit	Hold	Requested Start	Actual Start	Completion
<b>Deactivate Concurrent Manager</b>									
306336	SYSADMIN	Completed	Normal				27-Feb-11 09:24		27-Feb-11 09:26
306318	SYSADMIN	Completed	Normal				26-Feb-11 12:32		26-Feb-11 12:34
306273	SYSADMIN	Completed	Normal				25-Feb-11 11:31		25-Feb-11 11:32
306243	SYSADMIN	Completed	Normal				24-Feb-11 15:34		24-Feb-11 15:36
306178	SYSADMIN	Completed	Normal				23-Feb-11 15:36		23-Feb-11 15:38
306119	SYSADMIN	Completed	Normal				22-Feb-11 20:01		22-Feb-11 20:02
306032	SYSADMIN	Completed	Normal				14-Feb-11 10:25		14-Feb-11 10:26
306009	SYSADMIN	Completed	Normal				11-Feb-11 16:10		11-Feb-11 16:12
305923	SYSADMIN	Completed	Normal				08-Feb-11 12:09		08-Feb-11 12:11
305897	SYSADMIN	Completed	Normal				31-Jan-11 13:34		31-Jan-11 13:37
305852	SYSADMIN	Completed	Normal				28-Jan-11 11:39		28-Jan-11 11:40

This report could be used to assess if there is a pattern associated with when monitored programs are run, it could also be used to identify the guilty parties...

### 4.4 How do I turn the PAM monitored program - submitted requests alert off and on again?

The **PAM** monitored program - submitted requests alert can be turned off using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CP-003', 'N' );
```

and can be re-enabled using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CP-003', 'Y' );
```

## 4.5 Changing alert check frequency and / or severity

Both the alert check frequency and alert severity can be changed. Please refer to the *PAM* FAQs for more information on how to change an alert frequency and alert severity.

## 5 CR-005 Purge Sign-on audit date increment may not be incrementing

It is not uncommon to find that the Purge Sign-on Audit program date parameter is not incrementing. As a result the purge program may be purging Sign-on Audit data from a fixed date and not incrementing each purge run, thus purging no records each run.

Once per week (default) **PAM** will check your scheduled Purge Sign-on Audit program and check if the date is being incremented.

### 5.1 PAM Sign-on audit date increment e-mail alert

When **PAM** detects that the Purge Sign-on Audit programs date parameter is not being incremented a **PAM** alert e-mail is raised:

#### Example **PAM** CR-015 – **PAM** Sign-on audit date increment 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 = 04-Jan-11 (Wed) 05:00:03  
Alert Frequency = 1 Week

---

**The (Purge Signon Audit data) date parameter may not be incrementing.**

---

**Alert Information:**

**CR-005 Purge Sign-on Audit date increment may not be set**

**PAM** has detected that the (Purge Sign-on Audit data) date parameter may not be incrementing.

The concurrent program **Purge Sign-on Audit data** is generally run daily as part of a sites normal maintenance program. When the scheduled request is created the first argument is generally set to 32 days prior to the current date. As such the purge program will purge all records older than 32 days.

Failure to check the **increment date parameter each run** check box will cause the argument date not to be incremented each run, thus on each subsequent execution of the concurrent program the same purge date will be used and any records added after that date will not be purged.

The fix is to remove the current scheduled request and create a new one ensuring the **increment date parameter each run** check box is checked.

**Note:** If the purge has not been running for some time, check with your DBA to ensure you have sufficient roll back segments to perform the initial purge.

## 5.2 What to do with this information

The fix is to remove the current scheduled request and create a new one ensuring the **increment date parameter each run** check box is checked.

**Note:** If the purge has not been running for some time, check with your DBA to ensure you have sufficient roll back segments to perform the initial purge.

## 5.3 Turning CR-005 alert off and on

The **PAM** CR-005 alert can be turned off using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CR-005', 'N' );
```

The alert can be re-enabled using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CR-005', 'Y' );
```

## 6 CR-001 Aged On-hold

Once per month **PAM** will check for any concurrent requests that have been on-hold for more than 6 months (default) and alert when any are found.

Whilst aged on-hold requests are not strictly a bad thing, I do often find concurrent requests that have been on-hold for long periods of time. I have actually found some that have been on hold for several years... In reality this is just a regular cleanup process and a method of letting your users know you are on top of things and nothing escapes you...

**Note:** On-hold requests will not be purged from the `fnl_concurrent_requests` table.

Once a request has been on hold for longer than the **PAM** threshold period (6 months default) **PAM** will continue to raise an alert for that request until the request is removed or run.

### 6.1 Changing the aged on-hold threshold months

The number of months can be changed using the following **PAM** API:

```
exec PIPER_RX_PAM_API_2.PAM_THRESHOLD_CR001_SET ( 6 );
```

Parameter 1: The number of month's on-hold

### 6.2 PAM Aged on-hold requests e-mail alert

When **PAM** detects requests that have been on-hold for longer than the threshold months a **PAM** alert e-mail is raised:

#### Example **PAM** CR-001 – **PAM** Aged on-hold requests alert message

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

Company = Company name  
Site = Site name  
Alert Level = **Informational**  
Detected = 28-Feb-11 (Mon) 10:00:25  
Alert Frequency = 1 Month

---

**Request 2162772 has been on hold for over 72.1 months**

---

**Alert Information:**

**CR-001 Aged On-hold Requests**

**ONE OR MORE REQUESTS HAVE BEEN ON HOLD FOR LONGER THAN THE THRESHOLD PERIOD AND ARE MOST LIKELY NO LONGER REQUIRED.**

If you want to obtain a list of aged on-hold requests you can use *PAMreports* - Actions **PAMACR003 Aged On-Hold Requests**. Check with the user who submitted the request and cancel any requests that are no longer required

**Note 1:** Cleaning up aged on-hold requests is basic housekeeping task of best practice OEBS management

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

Often there are number of requests that are found on-hold. Where this is the case the *PAM* grouped e-mail function will send one alert covering all the alerts raised.

**Example *PAM* CR-001 – *PAM* Aged on-hold requests alert Group message**

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

Company = Company name  
Site = Site name  
Alert Level = **Informational**  
Detected = 28-Feb-11 (Wed) 11:00:30

---

**Alert when Aged On-hold Request are found**

Occurrences = 34  
First Alert = 23-Jun-10 (Wed) 11:57:25

First Alert = 23-Jun-10 (Wed) 12:07:30

Alert Frequency = 1 Month

**Alert Information:**

**CR-001 Aged On-hold Requests**

**ONE OR MORE REQUESTS HAVE BEEN ON HOLD FOR LONGER THAN THE THRESHOLD PERIOD AND ARE MOST LIKELY NO LONGER REQUIRED.**

If you want to obtain a list of aged on-hold requests you can use [PAMreports](#) - Actions [PAMACR003-10 Aged On-Hold Requests](#). Check with the user who submitted the request and cancel any requests that are no longer required

**Note 1:** Cleaning up aged on-hold requests is basic housekeeping task of best practice OEBS management

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

**6.3 What to do with this information**

A list of the aged on-hold requests can be found using [PAMreports](#) - Actions [PAMACR003 Aged On-Hold Requests](#):

Example [PAMACR003 Aged On-Hold Requests](#) report

Request ID	Requestor	Program Name Arguments	Requested Start Date	Age Months
2160221	OPERATIONS	Program - Update Limit Utilizations (None)	14-Jun-03 (Sat) 02:00	84.3
2160224	OPERATIONS	Program - Update Today's Average Rate (None)	14-Jun-03 (Sat) 02:00	84.3
2160533	SYSADMIN	Workflow Control Queue Cleanup (None)	05-Dec-03 (Fri) 18:31	78.6
2160534	EMCMGR	email Center Standalone Workflow Worker 1, F, T, MAILPREPROC, IEM_MAIL, NO_WAIT, 10	05-Dec-03 (Fri) 11:08	78.6
2161164	SYSADMIN	CP PLSQL Regression Test BASIC	24-Dec-03 (Wed) 11:49	78.0
2161165	SYSADMIN	CP PLSQL Regression Test BASIC	24-Dec-03 (Wed) 11:55	78.0
2161166	SYSADMIN	CP PLSQL Regression Test BASIC	24-Dec-03 (Wed) 11:56	78.0
2161167	SYSADMIN	CP PLSQL Regression Test BASIC	24-Dec-03 (Wed) 12:01	78.0
2161480	GPIPER	Completed Concurrent Requests Report FND, . . .	03-Mar-05 (Thu) 09:05	63.6
2162260	GPIPER	Active Responsibilities	21-Mar-04 (Sun) 21:07	75.1

This report uses the [PAM](#) aged on-hold threshold value to filter “old” on-hold requests.

You can then use this information to determine if the user still requires the request. If not (which should not be the case after 6 months...) it should be cancelled. Make sure you get the user to cancel their own request; they caused the issue...

## 6.4 How do I turn the *PAM* aged on-hold requests alert off and on again?

The *PAM* aged on-hold requests alert can be turned off using the following *PAM* API:

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

and can be re-enabled using the following *PAM* API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'CR-001', 'Y');
```

## 6.5 Changing alert check frequency and / or severity

Both the alert check frequency and alert severity can be changed. Please refer to the *PAM* FAQs for more information on how to change an alert frequency and alert severity.



## 7 Updating PAM Program Exceptions

PAM uses a single PAM program exceptions table `piper_rx_pam_cp_monitor_tl` to hold data on PAM monitored concurrent programs. This table is used by the following PAM alerts:

- ❖ CP-001 Selected programs must exist
- ❖ CP-002 Selected programs compete with a status of error or warning
- ❖ CP-003 Selected programs are submitted
- ❖ CP-004 Duplicate requests
- ❖ CM-003 Long running requests
- ❖ SPR-001 Possible inconsistent specialisation rules

### 7.1 Adding a program to PAM monitored programs exceptions

Concurrent programs can be added to the PAM monitored programs exception list using the following PAM API:

```
BEGIN
PIPER_RX_PAM_API.PAM_MONITORED_PROGRAM_ADD
( 101, -- Program Application ID
  101, -- Program ID
  'Y', -- Alert status - alert on or off
  'N', -- Alert if program does not exist
  'Y', -- Alert if completed error
  'Y', -- Alert if completed earning
  'N', -- Alert if submitted
  'N', -- Exclude from duplicates check
  'Y', -- Exclude from long running check
  'N' ); -- Exclude from specialisation rule check
END;
```

In this example we have added the General Ledger posting program (Application id 101 Program id = 101) alerting if the program completes with a status of either error or warning and to exclude the program from the PAM long running requests check.

**Note:** In this example it is pointless excluding the General Ledger posting program from duplicate requests check as the program has a defined conflict with itself so only one can run at a time.

When a program is first added, the program name will be displayed as “**unknown**” in **PAMreports** - Config **PAMC009 PAM Program Monitor List** report. This value will be updated with the program name when **PAM** check program exists (CP-001) is run.

Example **PAMC009 PAM Program Monitor List** report after the General Ledger posting program was added using the above example **PAM** API:

PAMC009-10		PIPER-RX - APPLICATION MONITOR PROGRAM MONITOR LIST As at 23-Mar-10 09:16:19 For PROD 121							PIPER - Rx	
Application	Prog Id	Program Name	Check Status	Exists	Completed			Exclude Duplicates	Exclude Long Running	
					Error	Warning	Submitted			
0 (FND)	0	Activate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	
0 (FND)	1	Deactivate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	
0 (FND)	3	Restart Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	
0 (FND)	4	Abort Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	
0 (FND)	5	Shutdown Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	
0 (FND)	6	Startup Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	
0 (FND)	31659	Report Set	Enabled	No	No	No	No	No	Yes	
0 (FND)	32263	Purge Concurrent Request and/or Manager Data	Enabled	Yes	No	No	No	No	Yes	
0 (FND)	32592	Purge Signon Audit data	Enabled	Yes	No	No	No	No	Yes	
0 (FND)	36034	Request Set Stage	Enabled	No	No	No	No	Yes	Yes	
0 (FND)	36888	Workflow Background Process	Enabled	No	No	No	No	Yes	Yes	
101 (GL)	101	Unknown	Enabled	No	Yes	Yes	No	No	Yes	
101 (GL)	32045	Program - Optimizer	Enabled	Yes	No	No	No	No	Yes	
178 (CX)	36662	Delete data from temporary table	Enabled	Yes	No	No	No	No	Yes	

To validate your entry prior to the **PAM** check program exists check (CP-001) being run, you can use **PAMreports** - Config **PAMC010 PAM Program Monitor Validate**:

Example **PAMC010 PAM Program Monitor Validate** report

PAMC010-10		PIPER-RX - APPLICATION MONITOR PROGRAM MONITOR LIST - VALIDATION As at 23-Mar-10 09:20:24 For PROD 121			PIPER - Rx
Prog Appn	Prog Id	Program Name	Validation		
0	0	Activate Concurrent Manager	Program Exists		
0	1	Deactivate Concurrent Manager	Program Exists		
0	3	Restart Concurrent Manager	Program Exists		
0	4	Abort Concurrent Manager	Program Exists		
0	5	Shutdown Concurrent Manager	Program Exists		
0	6	Startup Concurrent Manager	Program Exists		
0	31659	Report Set	Program Exists		
0	32263	Purge Concurrent Request and/or Manager	Program Exists		
0	32592	Purge Signon Audit data	Program Exists		
0	36034	Request Set Stage	Program Exists		
0	36888	Workflow Background Process	Program Exists		
101	101	* Posting	Program Exists		
101	32045	Program - Optimizer	Program Exists		
178	36662	Delete data from temporary table	Program Exists		

\* Indicates the program name has not been updated by the program monitor  
This will occur the next time the program monitor runs

### 7.1.1 Deleting a program from the **PAM** exceptions

A concurrent program can be removed from the **PAM** program exception list by using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_MONITORED_PROGRAM_DEL ( 101, 101);
```

Parameter 1: The program application id

Parameter 2: The concurrent program id

### 7.1.2 Changing **PAM** monitored programs items

The **PAM** monitored programs settings can be changed by first deleting the existing entry and then adding the program with the revised parameters.

### 7.1.3 Referential Integrity

After a period of time you may have a number of programs in the **PAM** monitored program list that no longer exist. The following **PAM** API performs a **PAM** referential cleanup which will remove any program from the **PAM** exceptions list that no longer exists in the OEBS application.

```
exec PIPER_RX_PAM_API.PAM_REFERENTIAL_CLEANUP;
```

## 8 How to find a concurrent program ID

In many of the APIs mentioned above you are required to use the concurrent programs application ID and program ID. Whilst we have covered this process in prior [PAMtutorials](#); it is worth refreshing your memory.

### Step 1 Find the Application ID

Using [PAMreports](#) - General [PAMRCP001 Concurrent Program Overview](#) report you can obtain a list of application IDs:

Example [PAMRCP001 Concurrent Program Overview](#) report

PAMRCP001-20		PAM - PIPER-RX - APPLICATION MONITOR				PIPER - Rx
Concurrent Program Overview						
As at 28-Feb-11 09:58						
For APPS 12i						
Total Active Application Program Count = 2074						
App ID	Appn	Application Name	Programs	Pct of Total	Disabled	
50	AD	Applications DBA	5	0.2	0	
601	AK	Oracle Common Modules-AK	2	0.1	0	
160	ALR	Oracle Alert	8	0.4	1	
200	AP	Oracle Payables	181	8.7	10	
222	AR	Oracle Receivables	201	9.7	25	
279	AS	Oracle Sales and Marketing	9	0.4	0	
600	AX	Global Accounting Engine	41	2.0	8	
702	BOM	Oracle Bills of Material	122	5.9	4	
260	CE	Oracle Cash Management	16	0.8	0	
202	CHV	Oracle Supplier Scheduling	3	0.1	0	
283	CN	Oracle Sales Compensation	44	2.1	7	
705	CRP	Oracle Capacity	15	0.7	0	
170	CS	Oracle Service	20	1.0	9	
20001	DEM	Demo Order Entry (AOL Class)	1	0.0	0	
175	EC	Oracle EDI Gateway	14	0.7	0	
703	ENG	Oracle Engineering	9	0.4	0	
802	FF	Oracle FastFormula	1	0.0	0	
0	FND	Application Object Library	63	3.0	1	
101	GL	Oracle General Ledger	168	8.1	29	

This report is also useful as it shows the total number of active reports (the count exclude disabled reports) in your application in the header, and for each application shows the number of reports, the percentage of total reports and the number of disabled reports in each application.

You may wish to run this report each month and determine if the number of reports has increased or decreased. If the change in value is different from your understanding of what the change should be then you can go on a hunt to seek out the guilty party 😊

### Step 2 Find the program ID

Using the application ID as the input variable for [PAMreports](#) - General [PAMRCP002 Concurrent Programs By Application](#) report, you can obtain a list of program IDs:

Example **PAMRCP002 Concurrent Programs By Application** report

PAMRCP002-20		PAM - PIPER-RX - APPLICATION MONITOR		PIPER - Rx
Concurrent Programs by Application ( 0 ) - ( FND ) - Application Object Library as at 28-Feb-11 10:00 for APPS 12i				
Prog ID	Prog Name	Program Name	Status	
4	ABORT	Abort Concurrent Manager	Enabled	
0	ACTIVATE	Activate Concurrent Manager	Enabled	
1	DEACTIVATE	Deactivate Concurrent Manager	Enabled	
20427	FDADCP	Retrieve Audit Data	Enabled	
31757	FDCHY	Compile value set hierarchies	Enabled	
11013	FDLEXPP	Flex PreProcessor	Enabled	
32880	FDVGN	Flexfield View Generator	Enabled	
31872	FNDADINS	Index Summary Report	Enabled	
31873	FNDADSQS	Sequence Summary Report	Enabled	
31874	FNDADTBD	Table Detail Report	Enabled	
31875	FNDADTBS	Table Summary Report	Enabled	

**Note:** Any program that has been disabled will be highlighted in red for easy identification

Example **PAMRCP002 Concurrent Programs by Application** report

36201	FNDPCRS	Summarize and Purge Concurrent Request Statistics	Enabled
31855	FNDPCGLC	Build program library catalog	Enabled
38091	FNDPLIST	Available Concurrent Programs (JSP Test)	Disabled
32344	FNDPPGD	Concurrent Program Details Report	Enabled
32329	FNDPPGS	Concurrent Programs Report	Enabled

From this report you should be able to identify the concurrent program ID for use in the **PAM** APIs.

When you add a new program, the program name will be shown as Unknown until the next **PAM** check. The program name is populated by the **PAM** check.

## 9 PAM Internal Checks

In an ongoing effort to limit any unbounded activity within the **PAM** application we are covering two additional internal checks in this tutorial.

### 9.1 IN-011 Invalid PAM Package Check

Once per hour (default) **PAM** will check each of its packages to determine if any of its packages have become invalid. Where a **PAM** package has been found to be invalid, **PAM** will “auto disable” any **PAM** checks associated with the invalid package until, on a subsequent check, **PAM** detects the package has been re-compiled and is valid. At this time the **PAM** checks will then be re-enabled. This internal check is designed to limit the number of **PAM** errors being generated as a result of invalid **PAM** objects.

When a **PAM** check is “auto disabled” by the **PAM** IN-011 check, the `alert_activty_flag` for that check held in the **PAM** config table `piper_rx_pam_config` will be set to ‘AD’ (**A**uto **D**isabled) and will not be executed by the **PAM** collector process.

When the **PAM** procedure is found to be valid the `alert_activty_flag` for checks associated with the **PAM** package will be re-enabled.

**Note:** **PAM** checks that have been manually disabled will not be affected by this **PAM** check.

#### Example PAM IN-011 – PAM Package Check alert message (Disabled)

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

Company = Company name  
Site = Site name  
Alert Level = **Warning**  
Detected = 28-Feb-11 (Mon) 12:00:56  
Alert Frequency = 1 Hour

---

**PAM Package PIPER\_RX\_PAM\_DBA\_SSM was found to be invalid. PAM has disabled the 2 PAM alert/s associated with this package**

**Alert Information:**

**IN-011 PAM invalid PAM package alert**

PAM has detected a PAM package has been identified as invalid.

This alert indicates that a PAM package has been identified as invalid. Should a PAM package become invalid, it will generate a PAM error each time that package is called by PAM and can generate PAM error at a rate of every 5 minutes.

When PAM identifies a PAM package as invalid, PAM will suspend that package until that package has been fixed and recompiled. When PAM identifies the package is valid, PAM will re-enable the package and it will continue to run as normal.

You can use PAMreports - Config PAMC004 PAM Errors to view the current PAM error log

**Note 1:** If you want to re-enable a PAM alert prior to the alert being auto re-enabled refer to the FAQs for more information

When PAM determines that a PAM package that has been auto disabled due to the PAM invalid PAM package check has been “fixed” and is now valid PAM will re-enable the package checks associated with the PAM package and send a notification alert:

**Example PAM IN-011 – PAM Package Check alert message (Enabled)**

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

Company = Company name  
Site = Site name  
Alert Level = **Informational**  
Detected = 28-Feb-11 (Wed) 13:00:02  
Alert Frequency = 1 Hour

**PAM Package PIPER\_RX\_PAM\_CP\_MONITOR was found to have been re-compiled and is now valid. PAM has enabled the 4 alert/s associated with this package**

**Alert Information:**

**IN-011 PAM invalid PAM package alert**

PAM has detected a PAM package has been identified as invalid.

This alert indicates that a PAM package has been identified as invalid. Should a PAM package become invalid, it will generate a PAM error each time that package is called by PAM and can generate PAM error at a rate of every 5 minutes.

When PAM identifies a PAM package as invalid, PAM will suspend that package until that package has been fixed and recompiled. When PAM identifies the package is valid, PAM will re-enable the package and it will continue to run as normal.

You can use PAMreports - Config PAMC004 PAM Errors to view the current PAM error log

**Note 1:** If you want to re-enable a PAM alert prior to the alert being auto re-enabled refer to the FAQs for more information

### 9.1.1 Turning the PAM internal invalid package alert feature on or off

The PAM internal invalid package alert can be turned off using the following PAM API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'IN-011', 'N');
```

and can be re-enabled using the following PAM API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'IN-011', 'Y');
```

### 9.1.2 Resetting an Invalid Object Disabled Alert

Should you “fix” the PAM error and recompile the PAM object and you wish the PAM check to be enabled prior to the next PAM invalid object check, you can re-enabled the check using the following PAM API:

```
exec PIPER_RX_PAM_API_2.PAM_RESET_DISABLED_INVALID ( 'IN-010');
```



Parameter 1: The **PAM** alert you wish to re-enable

## 9.2 Internal PAM Error Check (IN-012)

When **PAM** cannot process an individual package procedure, **PAM** will generate an error in the **PAM** `piper_rx_pam_errors` table. As the package failed, **PAM** will continue to run the package each time the collector is run (default 5 minutes) generating an error each time until the package runs unsuccessfully. So as to prevent **PAM** from generating too many error messages, once per hour (default) **PAM** will check the number of **PAM** errors being generated. When the number of errors exceeds the **PAM** threshold, the **PAM** check generating the errors will be “auto suspended” for the remainder of the day.

When a **PAM** check is “auto suspended” by the **PAM** IN-012 check, the `alert_activty_flag` for that check held in the **PAM** config table `piper_rx_pam_config` will be set to ‘AE’ (**A**uto Disabled on **E**rror) and will not be executed by the **PAM** collector process for the remainder of the day.

Once the application time passes midnight **PAM** will re-enable the **PAM** auto suspended check. If the **PAM** check has not been “fixed”, **PAM** will auto disable the check again once the threshold number of errors have been generated.

**Note:** **PAM** checks that have been manually disabled will not be affected by this **PAM** check.

### Example PAM IN-011 – PAM Package Check alert message

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

Company = Company name  
Site = Site name  
Alert Level = **Warning**  
Detected = 28-Feb-11 (Mon) 10:00:45  
Alert Frequency = 1 Hour

---

**PAM Package  
PIPER\_RX\_PAM\_CP\_MONITOR.PROGRAM\_CHECK\_SUBMITTED with  
11 errors been suspended for today**

**Alert Information:**

IN-012 **PAM** package errors

This alert is generated when a **PAM** package has been generating a high number of **PAM** errors.

When a high number of errors are generated for any **PAM** package that package will be suspended for the remainder of the day. This is designed to prevent **PAM** generating a large number of errors and error messages.

The **PAM** alert will be re-enabled at midnight, and will be suspended again if the **PAM** package has not been fixed and continues to error.

You can use **PAMreports** - Config **PAMC004 PAM Errors** to view the current **PAM** error log.

**Note 1:** If you want to re-enable a **PAM** alert prior to the alert being auto re-enabled at midnight refer to the FAQs for more information.

### 9.2.1 Changing the **PAM** IN-012 error threshold

The **PAM** IN-012 error threshold can be changed using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ERROR_THRESHOLD_SET ( 50 );
```

Parameter 1: The number of **PAM** errors for a given **PAM** check that once exceeded will result in the **PAM** check being suspended

### 9.2.2 Turning the **PAM** internal package error alert feature on or off

The **PAM** internal package error alert can be turned off using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'IN-012', 'N' );
```

and can be re-enabled using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'IN-012', 'Y');
```

### 9.2.3 Resetting a Suspended Alert

A suspended alert can be re-enabled prior to the end of the day using the following *PAM* API:

```
exec PIPER_RX_PAM_API_2.PAM_RESET_SUSPENDED_ERROR ( 'IN-012' );
```

Parameter 1: The *PAM* alert you wish to remove from suspension

This API will add an asterisk ( \* ) to the *PAM* package body name in the *PAM* errors table so as to prevent the alert from being re-suspended on the next *PAM* IN-012 check.

## 10 PAM Alert e-mail links

We now turn our attention to the **PAM** alert e-mail links section which is displayed on the bottom of each **PAM** alert e-mail. This is the first of the **PAM** customisation features.

At the bottom of each **PAM** alert e-mail there is a section providing “useful” links. You can add, remove or modify any or all of these links.

### Example

Twitter: [piper-rx](#) - Web: [www.piper-rx.com](#) - Mail: [pam@piper-rx.com](mailto:pam@piper-rx.com)

There are three (3) link types supported by **PAM**:

- ❖ Twitter
- ❖ Web
- ❖ E=mail (mailto)

Each link requires its own HTML format link text and this will be covered later in this tutorial.

Whilst we have provided three (3) examples linking to **PIPER-Rx** locations, you may wish to replace these with your company support center information (e-mail etc...)

### 10.1 Turning the **PAM** alert e-mail links feature on or off

The **PAM** alert e-mail links section can be turned off using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_EMAIL_LINK_DISPALY_SET ( 'N' );
```

and can be re-enabled using the following **PAM** API:

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

## 10.2 Listing current e-mail links

A list of the current **PAM** e-mail links can be found using **PAMreports** - Config **PAMC020 PAM Alert E-mail Links**:

Example **PAMC020 PAM Alert E-mail Links** report

Display Order	Status	Link Description
1	Enabled	Link to piper-rx twitter page Link ID: T1 Link Type: T - Twitter HREF Value: http://twitter.com/piper_rx Mouse Over Title: Twitter PIPER_RX Link Display Text: piper-rx Generated HTML: < >Twitter:</ > <A href="http://twitter.com/piper_rx" TARGET="_blank" title="Twitter PIPER_RX">piper-rx</A>
2	Enabled	Link to www.piper-rx.com web site Link ID: W1 Link Type: W - Web HREF Value: http://www.piper-rx.com Mouse Over Title: Web link to piper-rx.com Link Display Text: www.piper-rx.com Generated HTML: < >Web:</ > <A href="http://www.piper-rx.com" TARGET="_blank" title="Web link to piper-rx.com">www.piper-rx.com</A>
3	Enabled	Mailto link to pam@piper-rx.com Link ID: E1 Link Type: E - Mailto HREF Value: mailto:pam@piper-rx.com Mouse Over Title: mail to piper-rx.com Link Display Text: pam@piper-rx.com Generated HTML: < >Mail:</ > <A href="mailto:pam@piper-rx.com"> pam@piper-rx.com </a>

The global setting for the **PAM** e-mail alerts can be found in the report.

**Note:** Whilst individual items are enabled, none will be displayed if the e-mail link function is disabled.

## 10.3 Link Format

The link format items of note are as follows:

- ❖ Display order is the order in which the link will be displayed across the bottom of the **PAM** alert e-mail
- ❖ The link ID is the unique link identifier; the ID is made up of two components, the first letter indicates the link type (E – **E**-mail, W – **W**eb address, T – **T**witter account). The second component is a sequence number within each category. Uniqueness is enforced by a unique index on the link ID
- ❖ The HREF value is the html value to be used in the link

- ❖ Link display text is the text that is displayed in the e-mail e.g. [www.piper-rx.com](http://www.piper-rx.com)
- ❖ Generated HTML is the html that will be generated by **PAM** based on the above information and passed to the **PAM** e-mail generator

## 10.4 Turning individual links on and off

Individual links can be disabled or enabled using the follow **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_EMAIL_LINK_ENABLE_LINK ( 'T1' , 'N' );
```

Parameter 1: Is the unique link ID

Parameter 2: Is the link status 'Y' Enabled, 'N' Disabled

To re-enable the link:

```
exec PIPER_RX_PAM_API.PAM_EMAIL_LINK_ENABLE_LINK ( 'T1' , 'Y' );
```

## 10.5 Deleting individual links

Individual links can be deleted using the following **PAM** API:

**PAM** alert e-mail links can be permanently deleted using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_EMAIL_LINK_DEL ( 'T1' );
```

Parameter 1: Link ID is the unique ID of the link

The Link ID's and the current status of the e-mail links can be found using **PAMreports** - Config [PAMC020 PAM Alert Email Links](#)

## 10.6 Creating a PAM e-mail link

PAM alert e-mail links can be added using the following PAM API:

```
BEGIN

  PIPER_RX_PAM_API.PAM_EMAIL_LINK_ADD ( 'T1',
                                         'T',
                                         'Link to piper-rx twitter page',
                                         'Y',
                                         1,
                                         'http://twitter.com/piper_rx',
                                         'Twitter PIPER_RX',
                                         'piper-rx' ) ;

  COMMIT;

END;
```

Parameter 1: The unique link ID (primary key)

Parameter 2: The link type valid values are T = Twitter, W = Web, E = E-mail

Parameter 3: Free format link description

Parameter 4: Link display status 'Y' Enabled, 'N' Disabled

Parameter 5: The link display order - The order with which to display in item  
in the PAM e-mail alert

Parameter 6\*: The link "href" value

Parameter 7\*: The message to be displayed on mouse over

Parameter 8\*: The link text

\* indicates the value will be used by the PAM e-mail process to generate the resulting HTML that will be used in the PAM alert e-mail

## 10.7 Updating PAM alert e-mail links

PAM alert e-mail links can only be updated by deleting the existing link and recreating the link.

## 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*