# *PAMtutorials* 3: Database for Apps Administrators

## Entropy antidotes
## (Maintenance must haves for the more technically inclined)

## PIPER-Rx *A*pplication *M*onitor – *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 *PAMtutorials* 3 ..... 3
2 Invalid object check ..... 4
2.1 What is an invalid object? ..... 4
2.2 How *PAM* monitors and alerts on invalid objects ..... 4
2.2.1 DB-001 – Invalid Objects ..... 5
2.2.2 What alerts do I receive if many objects become invalid? ..... 6
2.2.3 What to do next when you get a *PAM* DB-001 alert ..... 6
2.2.4 Changing the *PAM* invalid object check run frequency ..... 7
2.2.5 Excluding objects from the *PAM* invalid object check ..... 8
2.2.6 How do I add another object to the *PAM* invalid object check exclusion list? ..... 9
2.2.7 Temporarily suspending an object's exclusion from the *PAM* invalid object check ..... 9
2.2.8 How do I delete a monitored object from the *PAM* invalid object exclusion list? ..... 10
2.2.9 Cleaning up old entries ..... 10
2.2.10 Recompiling *PAM* objects ..... 10
3 Index rebuild reminder ..... 12
3.1 Why do you need to rebuild indexes? ..... 12
3.2 How do you know when an index needs rebuilding? ..... 13
3.3 How the *PAM* index rebuild reminder works ..... 14
3.3.1 DB-002 - Index rebuild reminder ..... 14
3.3.2 Index rebuild list ..... 15
3.3.3 What indexes should be rebuilt? ..... 16
3.3.4 Adding an index to the *PAM* index check list ..... 17
3.3.5 How do I temporarily suspending an index from the *PAM* index rebuild check? ..... 17
3.3.6 Deleting an index from the *PAM* index rebuild check list ..... 17
4 Free space alert ..... 18
4.1 What makes the *PAM* free space alert different from all the other free space checks? ..... 18
4.2 How *PAM* monitors and alerts on database free space ..... 18
4.2.1 DB-003 - Tablespace free space alert ..... 18
4.2.2 Excluding a tablespace from the tablespace free space check ..... 21
4.2.3 Changing the free space alert threshold limit ..... 22
4.2.4 Changing the free space check run frequency ..... 22
5 DB-005 Alert when the database has been re-stared ..... 24
5.1 *PAM* Database restated e-mail alert ..... 24
5.2 What to do with this information ..... 25
5.3 Turning DB-005 alert off and on ..... 25
6 Disclaimer ..... 26

# 1 What you'll get out of *PAMtutorials* 3

***PAM*** is not database monitor (there are too many good Oracle database monitoring tools in the market already) but there are a couple of database related issues that are of interest to the pure Applications Administrator.

In this tutorial ***PAM*** alerts are installed to notify you by e-mail about:

- matters that may affect the normal running of your application
- some normal database maintenance tasks that are often forgotten:

  - ❖ Invalid objects – do I have any new ones?
  - ❖ Index rebuilds –DBAs often miss including this as part of a their normal maintenance plan
  - ❖ Database free space – am I running out?

# 2 Invalid object check

Not another invalid object check I hear you say…. well yes.

## 2.1 What is an invalid object?

The thing to understand is that with OEBS is that there are over 370,000 objects within the application (or more, depending on the number of modules you are running). Any of these can be come invalid at any time (because the application is having a bad hair day, the sun came up this morning or the DBAs applied a patch… you get the drift!)

Invalid objects can be very frustrating. You can waste half your life trying to find out why a particular object became invalid. When you find them, compile them; only investigate those that keep becoming invalid (the DBA should be doing this….) To compound the issue all sites have one or more invalid objects that just won't go away; in fact it is very rare for a site to have no invalid objects.

On site findings on invalid objects never cease to surprise....
I encountered one site that had no invalid objects. When asked about this, they simply said they deleted any object that remained invalid….
I have encountered many sites that have had invalid objects for several months, no one checked for them….

The question one needs to ask is why am I, an Applications Administrator, interested in invalid objects when the DBA should be monitoring and fixing where possible?

The answer is that to be forewarned is to be forearmed. If I know I have a new invalid object for AR and I receive a support request that is unusual related to AR it may be related to the new invalid object. You can then appear confident when you answer the support call.....***"we are aware of this and have the DBA's looking at it"***

In fact, in some cases by looking at the object name you may be able to identify if the support request is related to the invalid object

## 2.2 How *PAM* monitors and alerts on invalid objects

How ***PAM*** monitors and alerts on invalid objects is different to the "usual" invalid object checks. The common approach to monitoring invalid objects is to count the number of invalid objects and generate an alert when the number increases or in some cases decreases.

***PAM*** will only report on objects that are owned by registered application schemas or a ***PAM*** object. The DBA should be monitoring for all objects including non OEBS application related objects

***PAM*** does **not** use the total invalid object count as the prompt for an alert. ***PAM*** only reports on objects that have become invalid after the last ***PAM*** invalid object check – in this way ***PAM*** does not continually report on the same object.

***PAM*** allows you to exclude selected object from the invalid object check.

### 2.2.1 DB-001 – Invalid Objects

The ***PAM*** invalid object check will run every 4 hours (default) and if one or more application objects have become invalid since the last ***PAM*** check the following e-mail alert message is sent:

**Example *PAM* DB-001 - *PAM* Invalid object 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-Feb-11 (Fri) 17:00:51
Alert Frequency = 4 Hours

---

**3 new PIPER_RX objects have been identified as invalid since the last invalid object check ( 04-Feb-11 13:00 )**

---

**Alert Information:**

**DB-001 Invalid Objects**

**ONE OR MORE APPLICATION OBJECTS HAVE BECOME INVALID SINCE THE LAST CHECK.**

**Note 1:** Objects will not be reported on again unless they are fixed and become invalid again.

**Note 2:** Whilst all sites have invalid objects, it is good practice to investigate any new

invalid object as soon as possible. If you want to obtain a list of the objects that have become invalid since the last ***PAM*** invalid object check you can use ***PAMreports*** - Actions **PAMADB001 Invalid Objects** using the last check date provided in the alert message.

If you are aware of new invalid objects within a functional module then you are more attuned for any "left field" errors associated with that module that may be related to the invalid objects.

**Note 3:** Objects that continue to become invalid can be excluded from this check by adding the object to the ***piper_rx_pam_db_object_ex*** table setting the exclude_from_invalid_check to [Y].

**Note 4:** If you want to obtain a list of the current excluded objects you can use ***PAMreports*** - Config **PAMC016 PAM DB Object Exclusions**

### 2.2.2 What alerts do I receive if many objects become invalid?

The last thing anyone wants is a mail storm… Where there are more than **50** invalid objects detected ***PAM*** will send one e-mail alert message. Example:

> 124 objects have been identified as invalid since the last ***PAM*** invalid object check

Where there are 50 or less invalid objects detected ***PAM*** will generate one e-mail alert for each schema owner with new invalid objects. Example:

> 5 APPLSYS objects has been identified as invalid since the last invalid object check

**Note:** The group value of 50 is hard coded in the ***PAM*** source file piper_rx_pam_db_monitor.pkb (piper_rx_pam_db_monitor. invalid_objects)

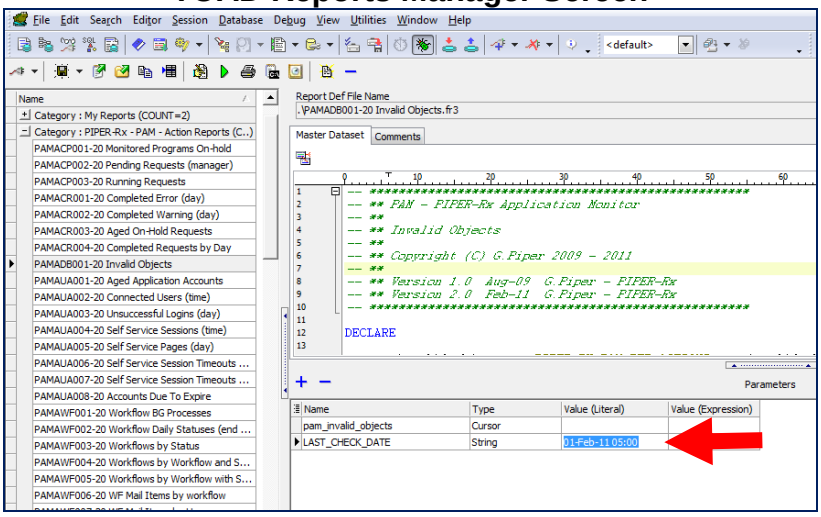
```
v_group_count number(3) := 50;
```

### 2.2.3 What to do next when you get a *PAM* DB-001 alert

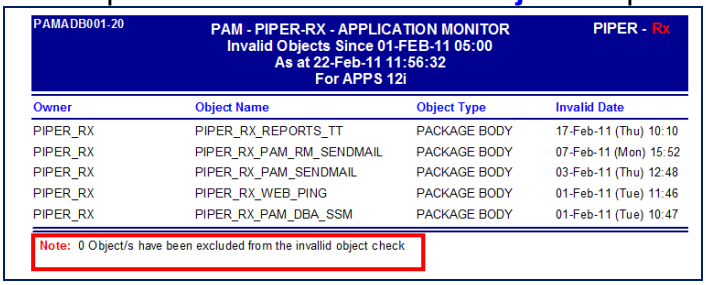
If you receive a **DB-001** alert from ***PAM*** then you can obtain a list of the newly invalid objects using ***PAMreports*** - Action PAMADB001- Invalid Objects by entering the last check date which can be found on the alert e-mail you received as indicated in the following example outlined in red:

**3 new PIPER_RX objects have been identified as invalid since the last invalid object check ( 01-Oct-09 12:00 )**

**TOAD Reports Manager Screen**




Example **PAMADB001 Invalid Objects** report




**Note:** As a reminder, the report will indicate the number of excluded objects from the invalid object check.

You can then use TOAD, a similar tool or script to recompile these objects.

## 2.2.4 Changing the *PAM* invalid object check run frequency

The ***PAM*** invalid object check has been set to run once every 4 hours (default). The run frequency can be changed using the following ***PAM*** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_FREQUENCY_CHANGE ( 'DB-001', 3 , 'HR');
```

Parameter 1: The ***PAM*** alert_id for the alert frequency you wish to change.
Parameter 2: The alert frequency
Parameter 3: The frequency unit

In the above example the ***PAM*** Invalid Object check (DB-001) will be changed to every 3 hours.

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

**Note:** If the frequency unit entered is invalid the change will not be applied.

### 2.2.5 Excluding objects from the *PAM* invalid object check

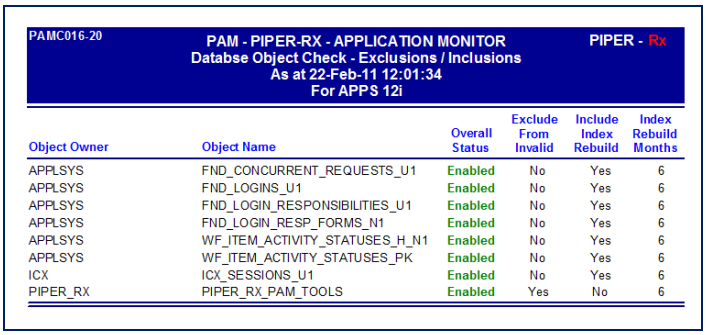
There may be occasions where an object regularly becomes invalid and is recompiled as part of a business process. For such instances ***PAM*** provides the option to exclude one or more objects from the ***PAM*** invalid object check.

***PAM*** has a single table (piper_rx_pam_db_object_ex) for holding both objects to be excluded from the ***PAM*** invalid object check and for holding a list of indexes to be included in the ***PAM*** index rebuild check. The structure of the table is as follows:

- object_owner
  - The schema owner of the object
- object_name
  - The object name
- object_check_status
  - "Y" indicates the object should be checked. This value can be used to temporarily suspend an object from the invalid object check and/or the index rebuild check
- exclude_from_invalid_check
  - "Y" indicates exclude from the invalid object check. This is generally set to "Y" if the object is to be checked for index rebuild only
- index_rebuild_check
  - "Y" indicates the object is to be checked for index rebuild
- index_rebuild_freq_months
  - The number of months between rebuilds

The ***PAMreports*** - Config PAMC016 PAM DB Object Exceptions will list the contents of this table:

Example **PAMC016 PAM DB Object Exceptions** report



PAMC016-20 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER - Rx
Databse Object Check - Exclusions / Inclusions
As at 22-Feb-11 12:01:34
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 |
| PIPER_RX | PIPER_RX_PAM_TOOLS | Enabled | Yes | No | 6 |

In this example the object piper_rx.piper_rx_pam_tools has been excluded from the invalid object check (Exclude from Invalid = “Yes”) and as it is not an index it is excluded from the index check (Include Index Rebuild = “No”)

### 2.2.6 How do I add another object to the *PAM* invalid object check exclusion list?

An object can be added to the invalid object check exclusion list using the following ***PAM*** API:

```
BEGIN

   PIPER_RX_PAM_API.PAM_MONITORED_OBJECT_ADD
      ( 'PIPER_RX',            -- Object owner
        'PAM_PIPER_RX_TOOLS', -- object name
        'Y',                   -- Check status
        'N',                   -- Exclude from invalid object check
        'N',                   -- Index rebuild check
        null );                -- index rebuild check frequency months

END;
```

### 2.2.7 Temporarily suspending an object’s exclusion from the *PAM* invalid object check

An excluded object can be temporarily suspended from the exclusion where by temporarily including the object in the invalid object check using the following ***PAM*** API:

```
BEGIN

   PIPER_RX_PAM_API.PAM_MONITORED_OBJECT_STATUS
           ( 'PIPER_RX',            -- Object owner
             'PAM_PIPER_RX_TOOLS', -- object name
             'N');                  -- Check status


END;
```

Setting the check status to “N” will temporarily suspend the object from the check.

Using the same API setting the check status to “Y” will re enable the check.

### 2.2.8 How do I delete a monitored object from the *PAM* invalid object exclusion list?

An object can be deleted from the ***PAM*** invalid object exclusion list using the following ***PAM*** API:

```
BEGIN

   PIPER_RX_PAM_API.PAM_MONITORED_OBJECT_DEL
           ( 'PIPER_RX',            -- Object owner
             'PIPER_RX_PAM_TOOLS' ); -- object name

END;
```

### 2.2.9 Cleaning up old entries

It is understood that over time database objects will be added to the ***PAM*** piper_rx_pam_db_object_ex table that were either incorrect (e.g. misspelt) or no longer exist in the application.

The piper_rx_pam_db_object_ex can be cleaned up using the following ***PAM*** API:

```
exec PIPER_RX_PAM_API.PAM_REFERENTIAL_CLEANUP;
```

This will remove any entries from the piper_rx_pam_db_object_ex table that either never existed or no longer exist in the application (i.e. a referential clean up).

### 2.2.10 Recompiling *PAM* objects

If one or more ***PAM*** objects become invalid, the following ***PAM*** API can be used to recompile ***PAM*** objects only:

```
exec PIPER_RX_PAM_API.PAM_INVALID_PAM_PACKAGES_FIX;
```

The list of objects to recompile comes from the list of ***PAM*** registered objects held in the piper_rx_pam_objects table.

**Note:** Of course if the piper_rx_pam_api package is invalid this will not work.

# 3 Index rebuild reminder

## 3.1 Why do you need to rebuild indexes?

When a record is added to a table an entry is added to the index. When a record is deleted from a table it is physically deleted, however it is logically deleted from the index. Thus if I add and remove 100 records to a table there will be no rows in the table, but the index will contain 100 logically deleted entries.

Let's work through a practical example: Your site is generating 5,000 concurrent requests per day and you are purging daily whilst maintaining the past 31 days history on-line.

At the end of month 1

> Table 5,000 rows * 30 days = 150,000 rows
> Index 5,000 rows * 30 days = 150,000 rows

At the end of month 2

> Table 5,000 rows * 30 days = 150,000 rows added
> Purge 5,000 rows * 30 days = 15,000 rows deleted
> Net 150,000 rows

Index 5,000 rows * 30 days = 150,000 rows new rows (150,000 logically deleted rows) = 300,000 rows (50% deleted rows)

At the end of month 3

> Table 5,000 rows * 30 days = 150,000 rows added
> Purge 5,000 rows * 30 days = 15,000 rows deleted
> Net 150,000 rows

Index 5,000 rows * 30 days = 150,000 rows new rows (300,000 logically deleted rows) = 450,000 rows (66% Deleted rows)

Before long you have a table with a relatively static level of 150,000 rows and an index that is full of deleted rows and very inefficient.

## 3.2 How do you know when an index needs rebuilding?

The general rule of thumb is if the index contains more than 30% deleted rows it should be rebuilt. I have personally seen some fantastic performance gains achieved just through cleaning up indexes.

The "analyse statistics" doesn't really tell you too much and does not give you the number of deleted rows. The only real way to know how many rows are deleted is to VALIDATE the index. If you have the time available, you may wish to try the code below on one of your high transaction indexes to see exactly how many deleted index entries there are:

```
VALIDATE index applsys.fnd_concurrent_requests_u1;

SELECT name,
       lf_rows,
       del_lf_rows,
       round((del_lf_rows * 100 / lf_rows),1) del_pct
  FROM sys.index_stats;
```

lf_rows - Is the total number of rows in the index
del_lf_rows - Is the number of deleted rows

Thus:

- (lf_rows - del_lf_rows) should equal the number of rows in the table
- ((del_lf_rows * 100) / lf_rows) is the percentage of delete rows in the index

Determining when an index contains more than 30% deleted rows using this method is very resource intensive. There are over 60,000 indexes… and you can only analyse one index at a time as the sys.index_stats object only holds one row at a time. So, to make life easier, a good rule of thumb is to rebuild your high transaction rate indexes every 3 to 6 months.

**PIPER-Rx has developed a method of validating every index in approximately 2-4 hours.** The overhead is still very high and thus it is only suitable to run this process on a fresh copy of production. In fact, **PIPER-Rx** can calculate the growth rate of every table and index in a **single pass** of the application database in less than half a day so no more having to record row counts over a period of 2 – 3 months for capacity planning! Remember, objects such as gl_balances only grows once per month by the number of rows in code combinations, so most tools that are DBA focussed and collect row counts once per week to calculate growth and capacity would miss this.

## 3.3 How the *PAM* index rebuild reminder works

To get around the issues associated with determining when an index rebuild is required and the overhead of this type of check, ***PAM*** provides the ability to add one or more indexes to an index rebuild check list and to specify the desired period of time between index rebuilds. Each index in the list is then checked to determine the time between the last rebuild time (DDL time) and the time of the alert check; if that time has exceeded the month's value for that for the index a ***PAM*** alert e-mail will be raised.

Where there are multiple indexes on a table ***PAM*** chooses just one index, usually the primary key index. If this index needs rebuilding so do the other indexes on that table.

By way of example if you add the primary key index for fnd_concurrent_requests fnd_concurrent_requests_u1 to the ***PAM*** index check list setting the index rebuild months to 6, if the index has not been rebuilt within the past 6 months a ***PAM*** alert e-mail will be sent.

### 3.3.1 DB-002 - Index rebuild reminder

The following is an example ***PAM*** index rebuild reminder e-mail alert:

**Example *PAM* DB-002 - *PAM* index rebuild reminder 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 = 23-Feb-11(Wed) 12:10:36
Alert Frequency = 1 Month

---

**Index APPLSYS.WF_ITEM_ACTIVITY_STATUSES_PK has not been rebuilt in over 76 months**

---

**Alert Information:**

**DB-002 Index Rebuild Reminder**

**THIS ALERT IS A REMINDER THAT ONE OR MORE MONITORED**

**INDEXES IS DUE FOR A REBUILD.**

There are a number of application tables that have high transaction turnover (inserts and deletes), the most recognisable table being ***fnd_concurrent_requests***

In order to maintain peak performance of these tables and the overall application it is advisable to periodically rebuild the indexes on high transaction tables

***PAM*** provides a reminder alert to alert when monitored index on selected tables have not been rebuilt in ***N*** months. Whilst ***PAM*** generally monitors one index per table, this alert should be interpreted to mean that all indexes for the base table should be rebuilt

Use ***PAMreports*** - Config **PAMC016 PAM DB Object Exclusions** for a list of indexes in the reminder list

**Note 4:** Beware the partitioned index!

**Note 4:** If you want to add or change the indexes that are monitored refer to the FAQs for more information

## 3.3.2 Index rebuild list

***PAM*** has a single table (piper_rx_pam_db_object_ex) for holding both objects to be excluded from the ***PAM*** invalid object check and a list of indexes to be **included** in the ***PAM*** index rebuild check. The structure of the table is as follows:

- object_owner
  - The schema owner of the object
- object_name
  - The object name
- object_check_status
  - "Y" indicates the object should be checked. This value can be used to temporarily suspend an object from the invalid object check and/or the index rebuild check
- exclude_from_invalid_check
  - "Y" indicates exclude from the invalid object check. This is generally set to "Y" if the object is to be checked for index rebuild only
- index_rebuild_check
  - "Y" indicates the object is to be checked for index rebuild
- index_rebuild_freq_months
  - The number of months between rebuilds

***PAMreports*** -Config PAMC016 PAM DB Object Exceptions will list the contents of this table. An example report is as follows:

Example **PAMC016 PAM DB Object Exceptions** report

PAMC016-20 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER - Rx
Databse Object Check - Exclusions / Inclusions
As at 22-Feb-11 12:01:34
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 |
| PIPER_RX | PIPER_RX_PAM_TOOLS | Enabled | Yes | No | 6 |

In this example the index applsys.fnd_concurrent_requests_u1 is not excluded from the invalid object check, is included in the index rebuild check and an e-mail alert will be sent if the index has not been rebuilt within the last 6 months.

### 3.3.3 What indexes should be rebuilt?

The ***PAM*** index monitor is preconfigured with the following known high transaction rate indexes:

- FND_CONCURRENT_REQUESTS_U1
- WF_ITEM_ACTIVITY_STATUSES_PK
- WF_ITEM_ACTIVITY_STATUSES_H_N1
- ICX_SESSIONS_U1
- FND_LOGINS_U1
- FND_LOGIN_RESPONSIBILITIES_U1
- FND_LOGIN_RESP_FORMS_N1

These indexes cover the basic application concurrent request, workflow, self-service sessions and sign-on audit tables.

Other indexes you should consider including in this check are any high transaction rate tables (inserts and deletes) and any interface tables that your site uses gl_interface, ar_interface etc…. and any custom interface tables.

Where a table has more than one index ( fnd_concurrent_requests has 9 or 10 indexes depending on the version of OEBS) you should select only one of those indexes (preferably the primary key index) to be checked; this is based on the understanding that if one of the indexes requires rebuilding then they all do.

### 3.3.4 Adding an index to the *PAM* index check list

An index can be added to the index check list using the following ***PAM*** API:

```
BEGIN

   PIPER_RX_PAM_API.PAM_MONITORED_OBJECT_ADD
      ( 'PIPER_RX',           -- Object owner
        'PAM_PIPER_RX_TOOLS', -- object name
        'Y',                  -- Check status
        'N',                  -- Exclude from invalid object check
        'Y',                  -- Index rebuild check
        6 );                  -- index rebuild check frequency months

END;
```

### 3.3.5 How do I temporarily suspending an index from the *PAM* index rebuild check?

An index can be temporarily suspended from the rebuild using the following ***PAM*** API:

```
BEGIN

   PIPER_RX_PAM_API.PAM_MONITORED_OBJECT_STATUS
           ( 'PIPER_RX',           -- Object owner
             'PAM_PIPER_RX_TOOLS', -- object name
             'N');                 -- Check status


END;
```

Setting the check status to “N” will temporarily suspend the object from the check.

Using the same API setting the check status to “Y” will re enable the check.

### 3.3.6 Deleting an index from the *PAM* index rebuild check list

An index can be deleted from the ***PAM*** index check list using the following ***PAM*** API:

```
BEGIN

   PIPER_RX_PAM_API.PAM_MONITORED_OBJECT_DEL
           ( 'PIPER_RX',           -- Object owner
             'PIPER_RX_PAM_TOOLS' ); -- object name

END;
```

# 4 Free space alert

## 4.1 What makes the *PAM* free space alert different from all the other free space checks?

***PAM*** provides a very simple check and does not profess to provide any more than that:

- It does not attempt estimate the remaining free space based on the application's ability to add additional extents
- It does not calculate growth rates
- It does not monitor temp or rollback segments

It is provided as an indicator only.

## 4.2 How *PAM* monitors and alerts on database free space

The ***PAM*** database free space check is run once per day (default) and is designed to alert when:

- One or more tablespaces have dropped below the percentage free space threshold and has shrunk since the last ***PAM*** free space check
- Data files have been added or removed

### 4.2.1 DB-003 - Tablespace free space alert

When ***PAM*** detects that one or more tablespaces have dropped below the percentage free space threshold and has shrunk since the last ***PAM*** free space check, or data files have been added or removed, ***PAM*** will send an e-mail alert.

The following are two examples of these ***PAM*** e-mail alerts and the associated ***PAMreports*** – General PAMRDB001- DB Space report:

## Example *PAM* DB-003 - *PAM* Tablespace free space 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 = 23-Feb-11 (Wed) 05:04:15
Alert Frequency = 1 Day

---

**Tablespace SYSAUX ( Total size 863 Mb - 1 File/s ) has decreased by .3 pct to 5.2 pct free space - approx 45 Mb Free Space**

---

**Alert Information:**

**DB-003 Tablespace Free Space**

**THE PERCENTAGE FREE SPACE IN THE TABLESPACE IDENTIFIED IN THIS ALERT HAS DROPPED BELOW THE ALERT THRESHOLD AND HAS SHRUNK SINCE THE PRIOR *PAM* FREE SPACE CHECK.**

**OR**

**A DATAFILE HAS BEEN ADDED OR REMOVED**

**If you wish to obtain a list of tablespaces and the available free space you can use *PAMreports* – General PAMRDB001 DB Space**

**Note 1: This alert is only raised when the free space in the tablespace has decreased since the last *PAM* free space check**

**Note 2: One or more tablespaces can be excluded from this check. Refer to the FAQs for more information**

## Example *PAM* DB-003 - *PAM* Tablespace free space 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 = 23-Feb-11 (Wed) 15:10:56
Alert Frequency = 1 Day

---

**Tablespace APPS_TS_TX_DATA has been Increased by 10000 Mb**

---

**Alert Information:**

**DB-003 Tablespace Free Space**

**THE PERCENTAGE FREE SPACE IN THE TABLESPACE IDENTIFIED IN THIS ALERT HAS DROPPED BELOW THE ALERT THRESHOLD AND HAS SHRUNK SINCE THE PRIOR *PAM* FREE SPACE CHECK.**

**OR**

**A DATAFILE HAS BEEN ADDED OR REMOVED**

**If you wish to obtain a list of tablespaces and the available free space you can use *PAMreports* – General PAMRDB001 DB Space**

**Note 1: This alert is only raised when the free space in the tablespace has decreased since the last *PAM* free space check**

**Note 2: One or more tablespaces can be excluded from this check. Refer to the FAQs for more information**

Example **PAMRDB001 DB Space** report (Page 1)

PAMRDB001-10

PAM - PIPER-RX - APPLICATION MONITOR
DB SPACE - TABLESPACE LIST
As at 05-Oct-09 14:48
For APPS 12i
Alert Threshold < 10 % Free Space and Decreasing

PIPER - Rx

| Tablespace | Alert Status | Sample Date 04-Oct-09 17:53 File Count | Total Size Mb | PCT Free | Sample Date 05-Oct-09 05:04 File Count | Total Size Mb | PCT Free |
|---|---|---|---|---|---|---|---|
| APPS_CALCLIP | Include | 4 | 5,501 | 15.0 | 4 | 5,501 | 15.0 |
| APPS_OMO | Include | 1 | 709 | 10.0 | 1 | 709 | 10.0 |
| APPS_TS_ARCHIVE | Include | 1 | 1,151 | 21.3 | 1 | 1,151 | 21.3 |
| APPS_TS_DISCO | Include | 1 | 1,000 | 23.4 | 1 | 1,000 | 23.4 |
| APPS_TS_DISCO_OLAP | Include | 1 | 1,350 | 99.3 | 1 | 1,350 | 99.3 |
| APPS_TS_INTERFACE | Include | 27 | 5,081 | 9.5 | 27 | 5,081 | 9.5 |
| APPS_TS_MEDIA | Include | 7 | 2,816 | 10.0 | 7 | 2,816 | 10.0 |
| APPS_TS_NOLOGGING | Include | 1 | 200 | 54.1 | 1 | 200 | 54.1 |
| APPS_TS_QUEUES | Include | 2 | 2,244 | 10.0 | 2 | 2,244 | 10.0 |
| APPS_TS_SEED | Include | 3 | 4,509 | 9.0 | 3 | 4,509 | 9.0 |
| APPS_TS_SUMMARY | Include | 13 | 14,056 | 64.1 | 13 | 14,056 | 64.1 |
| APPS_TS_TOOLS | Include | 1 | 250 | 100.0 | 1 | 250 | 100.0 |
| APPS_TS_TX_DATA | Include | 52 | 39,720 | 15.1 | 52 | 39,720 | 15.1 |
| APPS_TS_TX_IDX | Include | 24 | 25,124 | 9.3 | 24 | 25,124 | 9.3 |
| APPS_UNDOTS1 | Include | 1 | 1,993 | 75.3 | 1 | 1,993 | 74.1 |
| B2B_DT | Include | 1 | 68 | 10.5 | 1 | 68 | 10.5 |
| B2B_IDX | Include | 1 | 16 | 10.5 | 1 | 16 | 10.5 |
| B2B_LOB | Include | 1 | 12 | 10.9 | 1 | 12 | 10.9 |
| B2B_RT | Include | 1 | 42 | 9.5 | 1 | 42 | 9.5 |
| BAM | Include | 1 | 8 | 21.1 | 1 | 8 | 21.1 |
| CTXSYS | Include | 1 | 79 | 44.8 | 1 | 79 | 44.8 |
| CWMLITE | Include | 2 | 20 | 99.4 | 2 | 20 | 99.4 |
| DCM | Include | 2 | 198 | 10.0 | 2 | 198 | 10.0 |

06-Oct-09 8:54:53 AM © Gary Piper 2009 – All rights reserved Page 1 of 3

Example **PAMRDB001 DB Space** report (Page 2)

| Tablespace | Alert Status | Sample Date 04-Oct-09 17:53 File Count | Total Size Mb | PCT Free | Sample Date 05-Oct-09 05:04 File Count | Total Size Mb | PCT Free |
|---|---|---|---|---|---|---|---|
| RE | Include | 1 | 8 | 18.0 | 1 | 8 | 18.0 |
| SYNCSERVER | Include | 1 | 28 | 12.1 | 1 | 28 | 12.1 |
| SYSAUX | Include | 1 | 863 | 5.5 | 1 | 863 | 5.2 |
| SYSTEM | Include | 10 | 16,316 | 8.0 | 10 | 16,316 | 8.0 |
| UDDISYS_TS | Include | 1 | 21 | 8.9 | 1 | 21 | 8.9 |
| USERS | Include | 1 | 16 | 71.1 | 1 | 16 | 71.1 |
| WCRSYS_TS | Include | 1 | 3 | 43.8 | 1 | 3 | 43.8 |
| XDB | Include | 1 | 53 | 9.3 | 1 | 53 | 9.3 |

A red colour indicates change since prior sample

**Note:** The value shown in red for Pct free indicates that the pct free has decreased since the prior sample

## 4.2.2 Excluding a tablespace from the tablespace free space check

There may be one or more tablespaces that you do not want to be included in the ***PAM*** tablespace free space check.

A tablespace can be excluded from the ***PAM*** tablespace free space check using the following ***PAM*** API:

```
exec piper_rx_pam_api.pam_tablespace_exclude ( 'APPS_TS_SEED', 'Y');
```

Parameter 1: Tablespace name
Parameter 2: Alert status;Y = exclude from check, N = include in check

Any tablespace that has been excluded the from the ***PAM*** free space check will display a red Exclude value in the alert status column of ***PAMreports*** – General PAMRDB001 DB Space**:**

Example **PAMRDB001 DB Space** report

PAMRDB001-10

PAM - PIPER-RX - APPLICATION MONITOR
DB SPACE - TABLESPACE LIST
As at 05-Oct-09 15:01
For APPS 12i
Alert Threshold < 10 % Free Space and Decreasing

PIPER - Rx

| | | Sample Date 04-Oct-09 17:53 | | | Sample Date 05-Oct-09 05:04 | | |
|---|---|---|---|---|---|---|---|
| Tablespace | Alert Status | File Count | Total Size Mb | PCT Free | File Count | Total Size Mb | PCT Free |
| APPS_CALCLIP | Include | 4 | 5,501 | 15.0 | 4 | 5,501 | 15.0 |
| APPS_OMO | Include | 1 | 709 | 10.0 | 1 | 709 | 10.0 |
| APPS_TS_ARCHIVE | Include | 1 | 1,151 | 21.3 | 1 | 1,151 | 21.3 |
| APPS_TS_DISCO | Include | 1 | 1,000 | 23.4 | 1 | 1,000 | 23.4 |
| APPS_TS_DISCO_OLAP | Include | 1 | 1,350 | 99.3 | 1 | 1,350 | 99.3 |
| APPS_TS_INTERFACE | Include | 27 | 5,081 | 9.5 | 27 | 5,081 | 9.5 |
| APPS_TS_MEDIA | Include | 7 | 2,816 | 10.0 | 7 | 2,816 | 10.0 |
| APPS_TS_NOLOGGING | Include | 1 | 200 | 54.1 | 1 | 200 | 54.1 |
| APPS_TS_QUEUES | Include | 2 | 2,244 | 10.0 | 2 | 2,244 | 10.0 |
| APPS_TS_SEED | Exclude | 3 | 4,509 | 9.0 | 3 | 4,509 | 9.0 |
| APPS_TS_SUMMARY | Include | 13 | 14,056 | 64.1 | 13 | 14,056 | 64.1 |
| APPS_TS_TOOLS | Include | 1 | 250 | 100.0 | 1 | 250 | 100.0 |
| APPS_TS_TX_DATA | Include | 52 | 39,720 | 15.1 | 52 | 39,720 | 15.1 |
| APPS_TS_TX_IDX | Include | 24 | 25,124 | 9.3 | 24 | 25,124 | 9.3 |
| APPS_UNDOTS1 | Include | 1 | 1,993 | 75.3 | 1 | 1,993 | 74.1 |
| B2B_DT | Include | 1 | 68 | 10.5 | 1 | 68 | 10.5 |
| B2B_IDX | Include | 1 | 16 | 10.5 | 1 | 16 | 10.5 |
| B2B_LOB | Include | 1 | 12 | 10.9 | 1 | 12 | 10.9 |
| B2B_RT | Include | 1 | 42 | 9.5 | 1 | 42 | 9.5 |
| BAM | Include | 1 | 8 | 21.1 | 1 | 8 | 21.1 |
| CTXSYS | Include | 1 | 79 | 44.8 | 1 | 79 | 44.8 |
| CWMLITE | Include | 2 | 20 | 99.4 | 2 | 20 | 99.4 |
| DCM | Include | 2 | 198 | 10.0 | 2 | 198 | 10.0 |

## 4.2.3 Changing the free space alert threshold limit

The default alert level for free space is 10% for all tablespaces. This value can be changed using the following ***PAM*** API:

```
exec piper_rx_pam_api_2.pam_threshold_db003_set ( 15 );
```

Parameter: The free space alert level percentage (this must be assigned a value greater than 0 and less than 100).

## 4.2.4 Changing the free space check run frequency

The ***PAM*** free space check has been set to run once a day (default). The run frequency can be changed using the following ***PAM*** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_FREQUENCY_CHANGE ( 'DB-003', 2 , 'DAY');
```

Parameter 1:
The ***PAM*** alert_id for the alert frequency you wish to change.
Parameter 2:
The alert frequency
Parameter 3:
The frequency unit In the above example the tablespace free space check (DB-003) will be changed to every 2 days.

The list of valid values for the alert frequency units can be found using ***PAMreports*** – Config PAMC003 PAM Lookups - Lookup type “FREQUENCY UNITS”.

**Note:** If the frequency unit entered is invalid the change will not be made.

# 5 DB-005 Alert when the database has been re-stared

When the database is down for whatever reason, we all want to know when it is back up. The ***PAM*** DB-005 alert:- Alert when the database has been re-started, will send an e-mail alert letting you know the database has been restarted.

**Note:** For this alert to be raised the database must have been restarted and the ***PAM*** collector and email DBMS jobs must be running.

## 5.1 *PAM* Database restated e-mail alert

When ***PAM*** detects the database has been restated a ***PAM*** alert e-mail is raised:

**Example *PAM* DB-005 – *PAM* Database restated 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 = 10-Jan-12 (Tue) 13:25:04
Alert Frequency = 5 Minutes

**The database or an instance of the database may have been restarted 10-Jan-12 13:18**

**Alert Information:**

**DB-005 Database / Database Node Restart Alert**

***PAM*** has detected that one or more database instances have been restarted.

This alert is designed to inform you that the database may have been restarted and may now be available for use. Please check with your DBA prior to accessing the database.

In a multi node implementation this alert indicates that one or more of the database nodes may have been restated.

## 5.2 What to do with this information

This alert is informational only, when the database has been down for an extended period, this alert will inform you when it is back up. It is always best to check with your DBA's to make sure the database is available for use.

## 5.3 Turning DB-005 alert off and on

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

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

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

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

# 6 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.