PAMtutorials 18: Specialisation Rules

Getting to grips with Specialisation Rules

"When requests are not running in the managers you expect them to run in"

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* 18

PAMtutorials 18 is only suitable for sites that have either implemented custom concurrent managers or are planning to implement custom concurrent managers.

In essence a specialisation rule is the method you use to assign a concurrent program to a specific concurrent manager.

Why a module for specialisation rules?

Too many sites that create custom managers, do not assign the concurrent programs correctly and very rarely, if ever revisit the custom manager configuration. The most common issues I have encountered over the past 25 years and continue to encounter are at sites that:

- Create too many custom managers.
- Create custom managers without lowering the number of standard manager processes.
- Incorrectly define specialisation rules
- Create overly complex specialisation rules
- Have created mismatched specialisation rules that have been running for years

On the upside, I have seen that very few sites create custom work shifts, which in my opinion is a good thing.

2 Background – the basics

2.1 What determines where my requests will run?

A concurrent manager will run requests based on both its assigned:

Work shift/s - when the manager can run and the number of manager processes the manager should run and
Specialisation Rules - The programs the custom manager can run.

Note: In general, all requests run through the Standard manager unless you create a custom manager with a work shift and assign a program via a specialisation rule and then exclude the program from the standard manager.

2.2 Custom Concurrent Managers

A word of warning: - Creating a custom manager is relatively easy to process, getting rid of one is not. In fact you are not meant to delete a custom manager, only disable them, so plan carefully before you go creating custom managers.

Creating and configuring custom managers is not covered in this tutorial.

2.3 Work shifts

A work shift defines when one or more manager/s will run. Out of the box – all Concurrent managers are assigned the "standard" work shift. This work shift runs between 00:00 am and 23:59 seven (7) days a week.

Before you go creating additional work shifts you really want to be very sure of why you require a custom work shift, what will run in that work shift and what will happen to any program assigned to the manager / work shift when the work shift defines the manager as inactive.

One of the most complex custom work shifts is one to only run selected concurrent programs overnight. There are three (3) basic steps to achieving this setup:

In order for a manager to only run requests overnight (spans midnight) you must create two (2) work shifts and assign both those work shifts to the "Overnight" Manager:

- Overnight_am to run between 18:00 and 23:59
- Overnight_pm to run between 00:00 and 07:00

You then create a custom concurrent manager usually with a descriptive name such as "Overnight_only" and assign it both the overnight_am and overnight_pm work shifts.

The final step is to assign the programs you want to be run by the "overnight_only" manager using specialisation rules.

Note: When a request is submitted to run in the "Overnight_only" manager before or after the "Overnight_only" manager is running, the request will be marked as **Pending Error**, as there are no managers to run the request until the managers start.

2.3.1 Assigning a work shift to a manager

Once your work shift has been created you then assign that work shift to a concurrent manager.

Assigning one or more work shifts to a manager defines when that manager can run requests. In addition, for each work shift assigned to a manager you also define both the managers sleep time and the number of processes the manager can run.

2.3.2 Work Shift Order of Precedence

Where there are several work shifts assigned to a manager, which, by the way, should very rarely happen, there is an order of priority. In general the work shift with the highest selectivity takes precedence.

Work shift Order of Precedence

Precedence	Work shift Range	Example Range
Highest	Date and time range	30-July-11 14:00 to 15:00
↓	Date only	30-July-11
	Range of days and time range	Mon – Fri 09:00 to 18:00
	Range of days	Mon – Fri 09:00 to 18:00
	Time range	09:00 to 18:00
Lowest	Standard work shift	24 * 7 * 365

I am lead to believe that if you you have one or more work shifts with the same level of precedence, the work shift with the higher number of concurrent manager processes will take precedence.

Unfortunately using the OEBS application screens there is no easy way to identify which managers are assigned which work shifts without visiting each managers definition and selecting the Work Shifts tab.

2.3.3 *PAM* workshifts reports

To aid in the quick identification of workshifts and the work shifts applied to the concurrent managers, ***PAM*** provides the following two (2) reports:

PAMreports – Specialisation Rules PAMSPR001 Work shifts lists all the currently defined application work shifts:

Example **PAMSPR001 Work shifts** report

PAM SPR001.20 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER - Rx
Concurrent Manager Work Shifts
As at 27-JUL-11 08:53
For OEBS 12 DEMO

Work Shift Name	Descriptio	Start Time	End Time	Start Day	End Day	Specific Day	Assigned
One Hour Per Day	Gary Piper Special	07:00	08:00	Monday	Friday		1
Standard	Active 24 hours every day						42
Xmas Day	Xmas Day Only	00:00	23:59			25-Dec-2011	1

This report lists all the currently defined work shifts including the number of managers that have been **assigned** each work shift. I.e. The “One Hour Per Day” work shift has been assigned to one manager only.

PAMreports – Specialisation Rules PAMSPR002 Work Shift Assignments lists all the concurrent managers and the work shifts assigned to those managers:

Example **PAMSPR002 Work Shift Assignments** report

PAMSPR002-30

PAM - PIPER-RX - APPLICATION MONITOR
Concurrent Manager Work Shifts
As at 04-MAR-12 09:22
For APPS 12i

PIPER-Rx

Work Shift Name	Description	Start Time	End Time	Start Day	End Day	Specific Day	Process	Sleep
STANDARD - Standard queue for handling requests								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		2	60
ARTAXMGR - Used in 10SC to process tax calculations								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		1	60
CRPINQMGR - Manager to run Capacity Inquiries								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		2	60
FNDCRM - Manager that resolves request conflicts.								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		1	30
FNDSCH - Scheduler/Prereleaser Manager							Disabled	
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		1	30
INVMGR - Manager to run Immediate Concurrent Programs defined in INVLIBR								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		1	60
INVTMRPM - Inventory Remote Procedure Manager								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		3	30
MRPMGR - Manager to run Immediate conc. program of MRP								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		1	60
OESHTM - Shipping Transaction Manager for R11								
Standard	Active 24 hours every day	00:00	23:59	Sunday	Saturday		2	60

Note: Disabled managers will be identified by a red “disabled” value against the manager name as shown above.

2.4 Specialisation rules

Specialisation rules are used to assign specific concurrent manager programs to specific queues.

Specialisation rules can be one of the more complex Oracle Applications features to set up and once set up, one of the more complex to find using the current E-Business Suite screens. This makes it hard for an administrator taking over a site to understand and troubleshoot. Remember, with specialisation rules simplicity is the key.

A common problem I have witnessed is where a concurrent program has been "included" in the custom queue but has not been "excluded" from its default manager (usually standard). This will cause the concurrent program to be run in either queue depending on availability of a queue at the time the request is run. Having a concurrent program run in a number of queues I believe defeats the purpose of assigning the job to a specific queue.

To add to the complexity, there are a number of specialisation rule combinations:

- By Program
- By Application user
- By Oracle User
- By Request Type
- By Complex Rule, which is a combination of Program, application user and / or Oracle user

In the case of a program rule, it's a simple process of "including" a program to run in the custom manager and "excluding" the program from the standard manager. How simple can that be? Well most sites get it wrong.

Example:

I have a custom manager named "Slow" designed to process slow running requests so as not to bottleneck the standard manager. I have a concurrent program that has an average run time of 1 hour that I wish to assign to the "Slow" manager.

Firstly I need to create a specialisation rule for the slow manager to "include" the program to be run in that manager. I then need to create a second specialisation rule for the standard manager to "exclude" the program.

Note: At each of the two steps the managers will be bounced and a new ***fnd_concurrent_worker_requests*** view will be created. So don't do this during the business day….

If you “include” the concurrent program in the custom “slow” queue but forget to “exclude” the program from the standard queue, the program will run in either the standard or slow queue depending on which manager has processes available at the time the request is submitted.

To limit the site’s complexity, I believe that with the exception of request types, any rules created other than program rules should be avoided unless there is a specific business need for such rules.

2.5 Request types

Request types are one of the most powerful features of concurrent processing; they allow you to assign new requests of a program from one concurrent manager to another without the need to bounce the concurrent managers.

Note: Once a program has been submitted the queue cannot be changed.

A request type is a logical folder if you will, which is assigned to a specific queue by specialisation rules the same way as a program is assigned e.g. The request type “Slow” is “included” to the concurrent manager “Slow” queue and “excluded” from the standard manager.

Note: The process of assigning the request type to a concurrent manager via the specialisation rules requires the managers to be bounced and the fnd_concurrent_worker_requests view to be rebuilt.

Now that the specialisation rule for the request type has been created, when a program is assigned the request type “Slow” via the applications define program screens that program will run in the “Slow” queue, and when the request type is removed from the program, the program will run in its default queue.

So in effect a program is assigned to a request type and that request type will determine the manager that will run the program.

Note: Changing a program’s request type does not require the managers to be bounced or the fnd_concurrent_worker_requests view to be rebuilt.

An example where this could be useful might be where the standard manager is bottlenecked and the slow manager is relatively free; you could assign particular programs to the slow manager for a period until the backlog is cleared then assign them back to the standard manager.

As mentioned above, the limitation is that you cannot change a request’s queue once the request has been submitted. For scheduled jobs that have a resubmit, the

current scheduled job will run in the originally assigned manager whereas the new submission will run in the manager assigned by the request type you set.

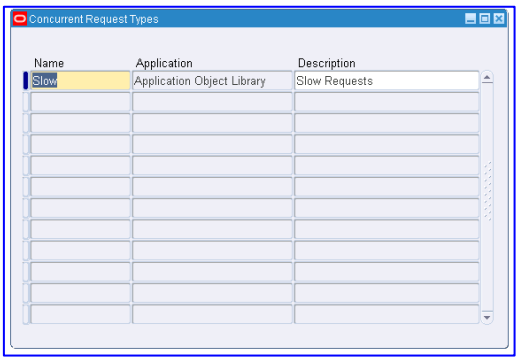
2.6 Creating and assigning a request type

Creating and assigning request type is a relatively simple process that can be removed at any time.

The following steps are a simplified example only and should not be used as a reference guide.

2.6.1 Creating a request type

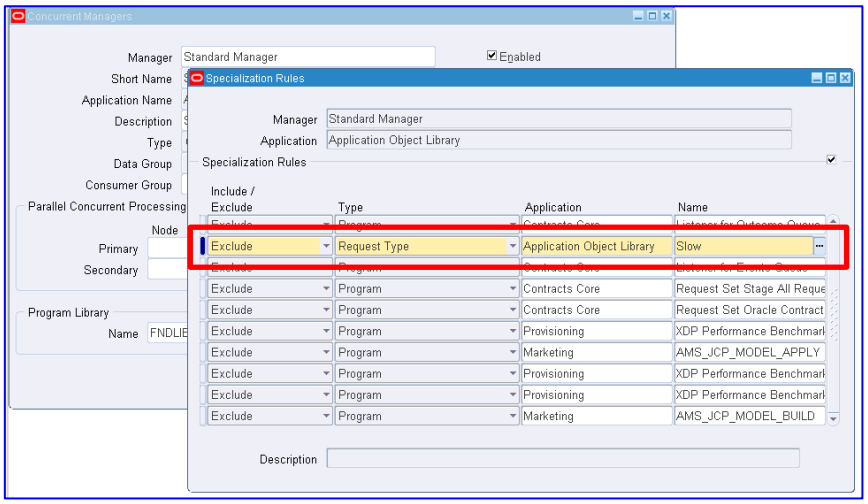
A request type is created via the application screens Concurrent > Program > Types



2.6.2 Assigning a request type to a custom manager

First exclude the request type from the standard manager
Concurrent > Manager > Define

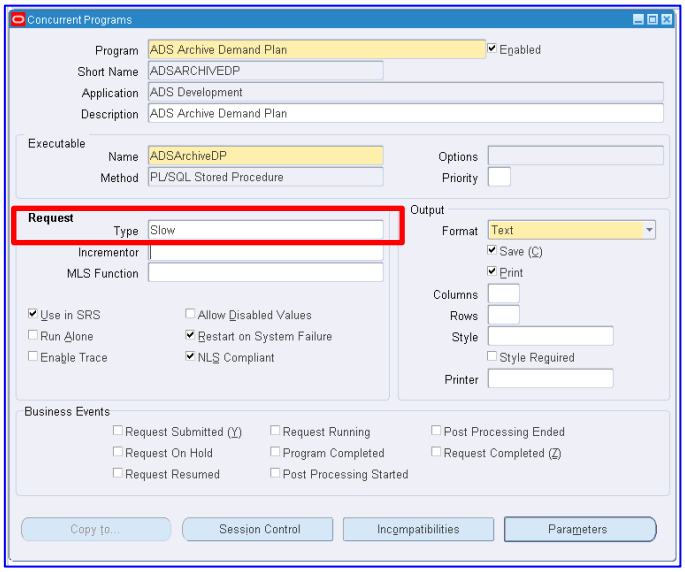
In this example we are a request type of “Slow” and will exclude from running in the standard manager:



The repeat the process for the target custom manager, but this time include the request type.

2.6.3 Assigning a request type to a concurrent program

The request type for a concurrent program is set via the applications Concurrent > Program > Define screen:



Note: A list of values is provided on this field

From this point on the concurrent program will run in the custom manager defined by the request type.

2.7 Incompatibility Rules

Programs may also not run when you expect based on one or more incompatibility rules.

An incompatibility rule defines that a selected program cannot run whilst a program that has been defined as incompatible is running.

An example of this is the GL Posting program “GLPPOS”, this program has been defined as incompatible with itself, as such a second or subsequent posting program being submitted must wait until the initial posting program has completed.

Much like specialisation rules, incompatibility rules are easy to create but not so easy to keep track of.

2.7.1 *PAM* Incompatibility Rules reports

To make identifying incompatibility rules simpler ***PAMreports*** provides two (2) reports:

PAMreports – Specialisation Rules PAMSPR030 Programs Incompatibilities List lists all programs that have a defined incompatibility rule including the number of programs that the source program is incompatible with. E.g. In the following example the program the concurrent program “Response Processor” is incompatible with three (3) other programs:

Example **PAMSPR030 Programs Incompatibilities List** report

PAMSPR030-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Program Incompatibilities List
As at 27-FEB-12 13:36
For APPS 12i

Appn Id	Appn	Program Id	Program Name	Program Full Name	Incompatibility Count
601	AK	33717	AKRMOVAL	Multi-Org Setup Validation Report	1
160	ALR	20393	ALECTC	Check Event Alert	1
160	ALR	20392	ALECDC	Check Periodic Alert	1
160	ALR	20394	ALEPPE	Periodic Alert Scheduler	1
160	ALR	20995	ALPPIM	Response Processor	3
200	AP	36071	APXTRF99	1099 Forms	3
200	AP	36067	APXTRINE	1099 Invoice Exceptions Report	1
200	AP	36069	APXTRRVT	1099 Payments Report	3

Using the application id and program id as inputs from ***PAMreports*** – Specialisation Rules PAMSPR030 Programs Incompatibilities List, ***PAMreports*** – Specialisation Rules PAMSPR031 Programs Incompatibilities By Program lists all programs that are incompatible with the selected program:

Example **PAMSPR031 Programs Incompatibilities By Program** report

PAM SPR031-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx

Program Incompatibilities List For Program
(160:20995) Response Processor
As at 27-FEB-12 13:42
For APPS 12i

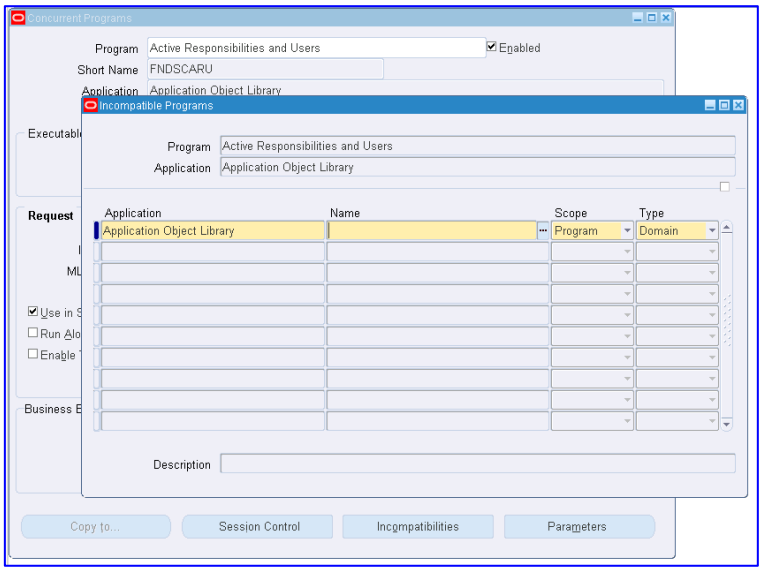
Appn Id	Appn	Program Id	Program Name	Program Full Name	Rule Status	Prog Type
160	ALR	20393	ALECTC	Check Event Alert	Enabled	Program
160	ALR	20392	ALECDC	Check Periodic Alert	Enabled	Program
160	ALR	20995	ALPPIM	Response Processor	Enabled	Program

In this example the concurrent program “Response Processor” is incompatible with itself and two other concurrent programs, “Check Event Alert” and “Check Periodic Alert”.

Incompatibility rules are enforced by the conflict resolution manager.

2.7.2 Managing Incompatibility Rules

Incompatibility rules are managed vie the application Concurrent > Programs > Define Screen. Select the incompatibilities button.



Note: You should only manage custom programs, as changing existing application rules could cause any number of application issues.

2.8 Role of the Conflict Resolution Manager (FNDCRM)

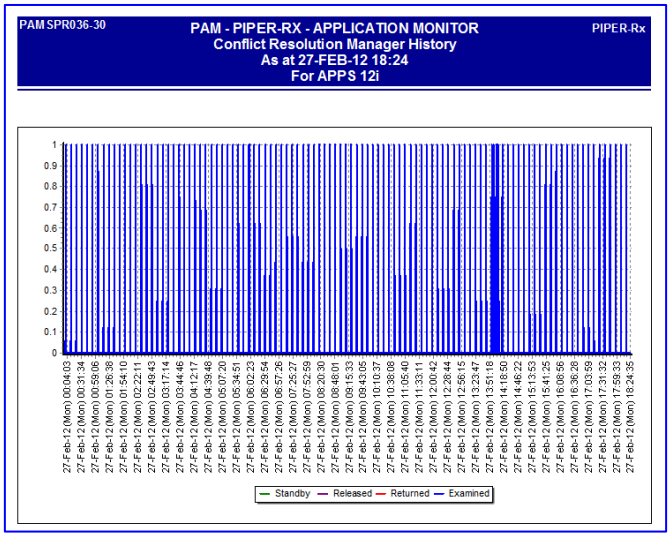
The conflict resolution manager assesses each requested concurrent program against programs that are running, referencing the applications incompatibility rules.

When a program lists other programs as being incompatible with it, the Conflict Resolution Manager prevents the program from starting until any incompatible programs in the same domain have completed running.

2.8.1 *PAM* Conflict Resolution Manager Report

You can use ***PAMreports*** – Specialisation Rules PAMSPR036 CRM Activity History (Today) to show the overall activity of the conflict resolution manager for the current day:

Example **PAMSPR036 CRM Activity History (Today)** report



2.9 Run Alone Programs

You can make a program incompatible with all other concurrent programs by defining the program to be run-alone via the run alone option in the concurrent program define screen. Care should be exercised here as once this program is running no other programs can run at the same time.

2.9.1 *PAM* Run Alone Programs report

You can use ***PAMreports*** – Specialisation Rules PAMSPR032 Run Alone Programs to list those programs defined as “Run Alone”.

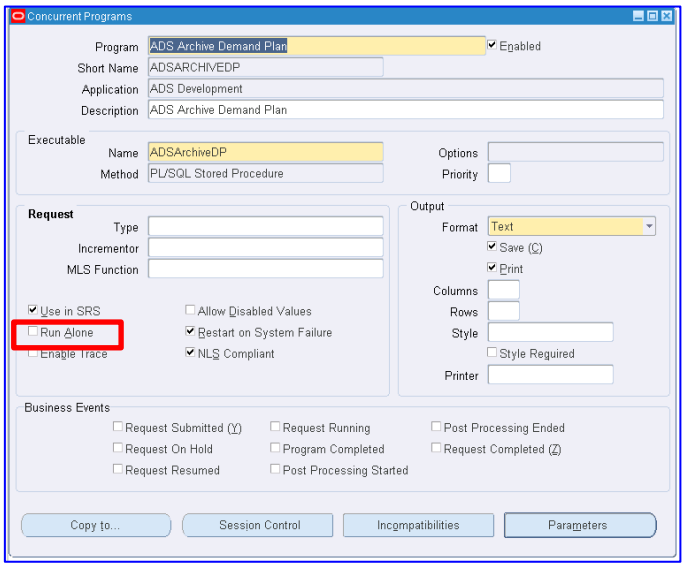
Example **PAMSPR032 Run Alone Programs** report

PAM SPR032:30 PAM - PIPER-RX - APPLICATION MONITOR PIPER-Rx
Run Alone Programs
As at 27-FEB-12 14:09
For APPS 12i

App'n	Program	Program Name	Program Status	Last Run Details: Request Id	Last Run Details: Start Date
		No Run Alone Programs Found			

Estimated on-line history days in fnd_concurrent_requests: 19

The concurrent program run alone setting is set via the applications Concurrent > Program > Define screen:



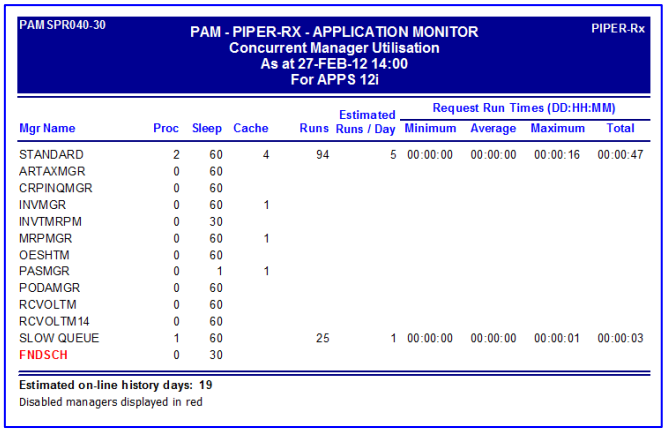
2.10 Manager Utilisation

As mentioned at the start of this tutorial, once established, custom manager configuration is very rarely re visited. You should be reviewing the concurrent manager configuration and overall utilisation at least annually.

2.10.1 *PAM* Manager Utilisation rules report

You can use ***PAMreports*** – Specialisation Rules PAMSPR040 Manager Utilisation rules to list the current concurrent manager configuration and the number of requests that have been running through the managers.

Example **PAMSPR040 Manager Utilisation** report



PAMSPR040-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx

Concurrent Manager Utilisation
As at 27-FEB-12 14:00
For APPS 12i

Mgr Name	Proc	Sleep	Cache	Runs	Estimated Runs / Day	Request Run Times (DD:HH:MM) Minimum	Average	Maximum	Total
STANDARD	2	60	4	94	5	00:00:00	00:00:00	00:00:16	00:00:47
ARTAXMGR	0	60							
CRPINQMGR	0	60							
INVMGR	0	60	1						
INVTMRPM	0	30							
MRPMGR	0	60	1						
OESHTM	0	60							
PASMGR	0	1	1						
PODAMGR	0	60							
RCVOLTM	0	60							
RCVOLTM14	0	60							
SLOW QUEUE	1	60		25	1	00:00:00	00:00:00	00:00:01	00:00:03
FNDSCH	0	30							

Estimated on-line history days: 19
Disabled managers displayed in red

> Note: The estimated available online history is shown at the bottom of the report

2.10.2 *PAM* concurrent program run time profiles reports

When assigning concurrent managers programs to custom managers or maintaining the programs that are running in standard and custom managers it is important to understand what programs have been running in what manager and the programs runtime profile.

PAM provides two (2) ***PAMreports*** – Specialisation Rules that list concurrent program run time profiles:

- PAMSPR041 Program Run Times By Program
- PAMSPR042 Program Run Times By Manager

These are in effect the same report with different sort profiles.

Example **PAMSPR041 Program Run Times By Program** report

PAM SPR041-30

PAM - PIPER-RX - APPLICATION MONITOR
Concurrent Program Run Time Profile By Program
As at 27-FEB-12 14:33
For APPS 12i

PIPER-Rx

Program Name	Mgr Name	Runs	Estimated Runs / Day	Request Run Times (DD:HH:MM) Minimum	Average	Maximum	Total
Active Responsibilities and Users	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
CN_R1212_CNCMAUPD	STANDARD	10	0	00:00:00	00:00:00	00:00:00	00:00:00
CN_R1212_CNCMHUPD	STANDARD	10	0	00:00:00	00:00:00	00:00:00	00:00:00
Compile Non-Compiled Flexfields	STANDARD	2	0	00:00:02	00:00:02	00:00:02	00:00:04
Compile Security	STANDARD	2	0	00:00:00	00:00:00	00:00:00	00:00:00
Contract Expert: Activate Rules	STANDARD	14	0	00:00:00	00:00:00	00:00:01	00:00:03
Diagnostics patching CP	STANDARD	1	0	00:00:01	00:00:01	00:00:01	00:00:01
Drop obsolete products schema	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
FNDIRLOAD	STANDARD	5	0	00:00:02	00:00:04	00:00:06	00:00:20
Flexfield View Generator	STANDARD	361	0	00:00:00	00:00:00	00:00:00	00:00:04
Generic Loader	STANDARD	10	0	00:00:00	00:00:01	00:00:03	00:00:19
Import Configuration Rules	STANDARD	7	0	00:00:00	00:00:00	00:00:00	00:00:01
Items Data Scripts Execution	STANDARD	2	0	00:00:00	00:00:00	00:00:00	00:00:00
Migrate Accruals for Multi Currency Changes	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
OAM Applications Dashboard Collection	OAMCOLMGR	4,136	3	00:00:00	00:00:00	00:00:00	00:04:34
OAM Applications Dashboard Collection	STANDARD	18,221	15	00:00:00	00:00:00	00:00:03	00:20:36
Open Period Balances	STANDARD	401	0	00:00:00	00:00:00	00:00:00	00:00:02
iPayment FP.G Upgrade Program	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
iRep Loading Post Processor	STANDARD	5	0	00:00:00	00:00:00	00:00:00	00:00:01

Estimated on-line history days: 1,243

The hidden gem in this report is that you will be able to clearly identify a program that has been running in more than one manager. I.e. See the OAM Application Dashboard Collector program in the above example report.

Example **PAMSPR042-30 Program Run Times By Manager** report

PAM SPR042-30

PAM - PIPER-RX - APPLICATION MONITOR
Concurrent Program Run Time Profile By Manager
As at 27-FEB-12 14:33
For APPS 12i

PIPER-Rx

Program Name	Mgr Name	Runs	Estimated Runs / Day	Request Run Times (DD:HH:MM) Minimum	Average	Maximum	Total
OAM Applications Dashboard Collection	OAMCOLMGR	4,136	3	00:00:00	00:00:00	00:00:00	00:04:34
Active Responsibilities and Users	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
CN_R1212_CNCMAUPD	STANDARD	10	0	00:00:00	00:00:00	00:00:00	00:00:00
CN_R1212_CNCMHUPD	STANDARD	10	0	00:00:00	00:00:00	00:00:00	00:00:00
Compile Non-Compiled Flexfields	STANDARD	2	0	00:00:02	00:00:02	00:00:02	00:00:04
Compile Security	STANDARD	2	0	00:00:00	00:00:00	00:00:00	00:00:00
Contract Expert: Activate Rules	STANDARD	14	0	00:00:00	00:00:00	00:00:01	00:00:03
Diagnostics patching CP	STANDARD	1	0	00:00:01	00:00:01	00:00:01	00:00:01
Drop obsolete products schema	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
FNDIRLOAD	STANDARD	5	0	00:00:02	00:00:04	00:00:06	00:00:20
Flexfield View Generator	STANDARD	361	0	00:00:00	00:00:00	00:00:00	00:00:04
Generic Loader	STANDARD	10	0	00:00:00	00:00:01	00:00:03	00:00:19
Import Configuration Rules	STANDARD	7	0	00:00:00	00:00:00	00:00:00	00:00:01
Items Data Scripts Execution	STANDARD	2	0	00:00:00	00:00:00	00:00:00	00:00:00
Migrate Accruals for Multi Currency Changes	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
OAM Applications Dashboard Collection	STANDARD	18,221	15	00:00:00	00:00:00	00:00:03	00:20:36
Open Period Balances	STANDARD	401	0	00:00:00	00:00:00	00:00:00	00:00:02
iPayment FP.G Upgrade Program	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
iRep Loading Post Processor	STANDARD	5	0	00:00:00	00:00:00	00:00:00	00:00:01

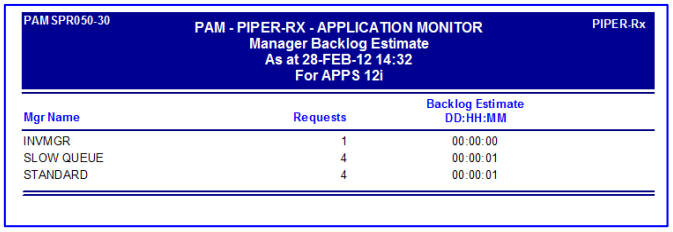
Estimated on-line history days: 1,243

2.10.3 *PAM* Manager Backlog Report

There are occasions where one or more of your managers are backlogged with requests.

You can use ***PAMreports*** – Specialisation Rules **PAMSPR054 Manager Backlog Estimate** to list the number of pending requests per manager and the estimated time for the back log to clear.

Example **PAMSPR054 Manager Backlog Estimate** report



PAM SPR050-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx

Manager Backlog Estimate
As at 28-FEB-12 14:32
For APPS 12i

Mgr Name	Requests	Backlog Estimate DD:HH:MM
INVMGR	1	00:00:00
SLOW QUEUE	4	00:00:01
STANDARD	4	00:00:01

Note: This is an estimate only based on the average historical run times of pending concurrent request. It should be used as a guide only.

3 *PAM* Specialisation Rule Alerts

PAM provides a series of specialisation rule alerts to aid in the management of your specialisation rules and subsequent concurrent processing.

3.1 SPR-001 - Check for possible inconsistent SPR rules

Once per week *PAM* will check for possible specialisation rule inconsistencies. When *PAM* detects an inconsistency *PAM* will generate an alert e-mail.

Example *PAM* SPR-001 – *PAM* Possible Inconsistent SPR rule alert message

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

Customer = Company name
Site = Site name
Alert Level = **Informational**
Detected = 28-Feb-12 (Tue) 20:01:05
Alert Frequency = 1 Week

Program Sign-on Audit Data has been run in multiple managers.

Alert Information:

SPR-001 – Possible specialisation rule issues are found

A CONCURRENT PROGRAM HAS BEEN IDENTIFIED AS RUNNING IN MORE THAN ONE MANAGER.

This *PAM* alert is raised when a concurrent program has been found to have run in more than one manager.

This alert is generally as a result on mismatched specialisation (exclude / include) rules.

Where a program has been assigned (Included) to a custom concurrent manager and not been excluded from the standard manager, that program will run in either manager, whichever manager has available processes available at the time the program is requested.

As specialisation rules are easy to create but difficult to find once created, ***PAMreports*** provides a number of reports to list the various specialisation rule types:

PAMSPR010 Program Rules
PAMSPR011 User Rules
PAMSPR012 Oracle Schema Rules
PAMSPR013 Complex Rules
PAMSPR014 Request Type Rules

From these report you should easily be able to identify mismatched rule/s including when and who created the rule.

Note 1: You may receive this alert if you have recently changes a specialisation rule transferring a program from one concurrent manager to another within the last ***PAM*** specialisation rule sample period (Default Weekly).

Note: Once a program has been reported with a possible specialisation rule issue that program will not be reported again for a period of 31 days.

3.1.1 What to do with this information

You can use ***PAMreports*** – Specialisation Rules PAMSPR041 Program Run Times By Program to list all the concurrent programs alerted and the manager/s that have been running them. In this report you should be able to identify the managers that have been running the alerted program.

Example **PAMSPR041 Program Run Times By Program** report

PAMSPR041-30 — PIPER-Rx

PAM - PIPER-RX - APPLICATION MONITOR
Concurrent Program Run Time Profile By Program
As at 27-FEB-12 14:33
For APPS 12i

Program Name	Mgr Name	Runs	Estimated Runs / Day	Request Run Times (DD:HH:MM) Minimum	Average	Maximum	Total
Active Responsibilities and Users	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
CN_R1212_CNCMAUPD	STANDARD	10	0	00:00:00	00:00:00	00:00:00	00:00:00
CN_R1212_CNCMHUPD	STANDARD	10	0	00:00:00	00:00:00	00:00:00	00:00:00
Compile Non-Compiled Flexfields	STANDARD	2	0	00:00:02	00:00:02	00:00:02	00:00:04
Compile Security	STANDARD	2	0	00:00:00	00:00:00	00:00:00	00:00:00
Contract Expert: Activate Rules	STANDARD	14	0	00:00:00	00:00:00	00:00:01	00:00:03
Diagnostics patching CP	STANDARD	1	0	00:00:01	00:00:01	00:00:01	00:00:01
Drop obsolete products schema	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
FNDIRLOAD	STANDARD	5	0	00:00:02	00:00:04	00:00:06	00:00:20
Flexfield View Generator	STANDARD	361	0	00:00:00	00:00:00	00:00:00	00:00:04
Generic Loader	STANDARD	10	0	00:00:00	00:00:01	00:00:03	00:00:19
Import Configuration Rules	STANDARD	7	0	00:00:00	00:00:00	00:00:00	00:00:01
Items Data Scripts Execution	STANDARD	2	0	00:00:00	00:00:00	00:00:00	00:00:00
Migrate Accruals for Multi Currency Changes	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
OAM Applications Dashboard Collection	OAMCOLMGR	4,136	3	00:00:00	00:00:00	00:00:00	00:04:34
OAM Applications Dashboard Collection	STANDARD	18,221	15	00:00:00	00:00:00	00:00:03	00:20:36
Open Period Balances	STANDARD	401	0	00:00:00	00:00:00	00:00:00	00:00:02
iPayment FP.G Upgrade Program	STANDARD	1	0	00:00:00	00:00:00	00:00:00	00:00:00
iRep Loading Post Processor	STANDARD	5	0	00:00:00	00:00:00	00:00:00	00:00:01

Estimated on-line history days: 1,243

PAM provides four (4) ***PAMreports*** – Specialisation Rules which may help in identifying the cause of the issue:

3.1.2 Program Rules

You can use ***PAMreports*** – Specialisation Rules PAMSPR010 Program Rules to list the current program based specialisation rules:

Example **PAMSPR010 Program Rules** report

PAM SPR010-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Specialisation Rules - Program
As at 27-Feb-12 14:28
For APPS 12i

Prog Appn	Program Name	Manager Name	Include	Exclude	Rule Status	Last Updated By	Last Updated Age Y:M:D
AMS	AMS_JCP_MODEL_APPLY	AMSDMIN	Include			INITIAL SETUP	08:04:10
AMS	AMS_JCP_MODEL_APPLY	STANDARD		Exclude		INITIAL SETUP	08:04:10
AMS	AMS_JCP_MODEL_BUILD	STANDARD		Exclude		INITIAL SETUP	08:04:10
AMS	AMS_JCP_MODEL_BUILD	AMSDMIN	Include			INITIAL SETUP	08:04:10
BOM	Overhead Cost Worker	INVMGR	Include			INITIAL SETUP	18:04:06
BOM	Resource Cost Manager	INVMGR	Include			INITIAL SETUP	18:06:01
BOM	Resource Cost Worker	INVMGR	Include			INITIAL SETUP	18:06:01
FND	OAM Applications Dashboard Collection	OAMCOLMGR	Include		Check	INITIAL SETUP	09:09:11
FND	Run Diagnostic Tests	OAMCOLMGR	Include			INITIAL SETUP	07:05:00
FTE	LTL Rate Chart Loader	FTE_TXN_MANAGER	Include			INITIAL SETUP	09:03:05
IEU	Session History - UWQ Cleanup Process	IEU_SH_CS	Include			Unknown	05:06:14
IEU	UWQ Worklist Items Release for Crashed session	IEU_WL_CS	Include			INITIAL SETUP	07:03:22
IEX	IEX: Age Delinquencies	IEXCONMGR	Include			Unknown	09:11:28
IEX	IEX: Create Delinquencies	IEXCONMGR	Include			Unknown	09:11:28
IEX	IEX: Create Dunning And Broken Promise Call Backs	IEXCONMGR	Include			Unknown	09:11:28
IEX	IEX: Process Payments for Collections	IEXCONMGR	Include			Unknown	09:11:28
IEX	IEX: Promise Reconciliation	IEXCONMGR	Include			Unknown	09:11:28
IEX	IEX: Score Collections Customers	IEXCONMGR	Include			Unknown	09:11:28
IEX	IEX: Send Dunnings for Delinquent Customers	IEXCONMGR	Include			Unknown	09:11:28

In this example the concurrent program OAM Application Dashboard Collector has been included in the OAMCOLMGR but not excluded from the standard manager. This is a default setting provided by Oracle, as such ***PAM*** has excluded this program from the specialisation rules alert check. Programs excluded from the ***PAM*** SPR-001 – Possible specialisation rules issues alert are displayed in gray.

3.1.3 User Rules

You can use ***PAMreports*** – Specialisation Rules PAMSPR011 User Rules to list the current application user based specialisation rules:

Example **PAMSPR011 User Rules** report

PAM SPR011-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Specialisation Rules - User
As at 27-Feb-12 14:28
For APPS 12i

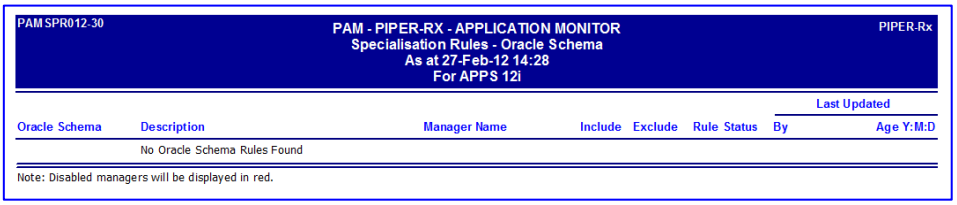
User Name	Description	Manager Name	Include	Exclude	Rule Status	Last Updated By	Last Updated Age Y:M:D
	No User Rules Found						

Note: Disabled managers will be displayed in red.

3.1.4 Oracle Schema Rules

You can use ***PAMreports*** – Specialisation Rules PAMSPR012 Oracle Schema Rules to list the current application user based specialisation rules:

Example **PAMSPR012 Oracle Schema Rules** report



PAM SPR012-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Specialisation Rules - Oracle Schema
As at 27-Feb-12 14:28
For APPS 12i

Oracle Schema	Description	Manager Name	Include	Exclude	Rule Status	Last Updated By	Last Updated Age Y:M:D
	No Oracle Schema Rules Found						

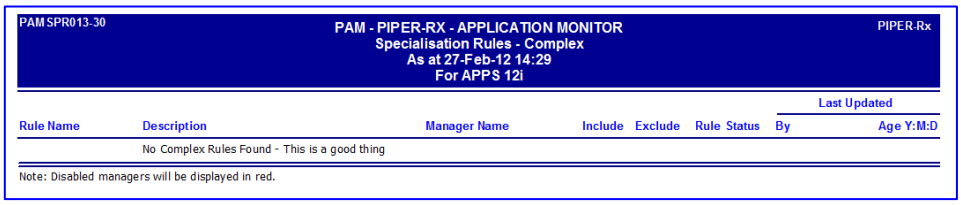
Note: Disabled managers will be displayed in red.

3.1.5 Complex Rules

Complex rules are a combination of Program, User and Oracle schema rules. They are easy to establish and complex to manage and maintain and for these reasons in my opinion should be avoided at all costs.

If you have any complex rules, you can use ***PAMreports*** – Specialisation Rules PAMSPR013 Complex Rules to list the current application complex specialisation rules:

Example **PAMSPR013 Complex Rules** report



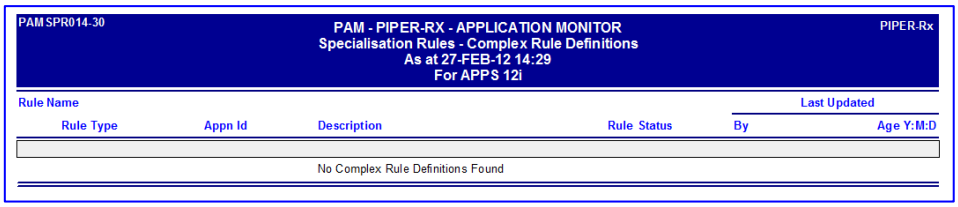
PAM SPR013-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Specialisation Rules - Complex
As at 27-Feb-12 14:29
For APPS 12i

Rule Name	Description	Manager Name	Include	Exclude	Rule Status	Last Updated By	Last Updated Age Y:M:D
	No Complex Rules Found - This is a good thing						

Note: Disabled managers will be displayed in red.

If you were forced to create or inherit complex specialisation rules, ***PAMreports*** – Specialisation Rules provides report **PAMSPR014-30 Complex Rule Definitions** that lists the current complex rule definitions:

Example **PAMSPR014-30 Complex Rule Definitions** report



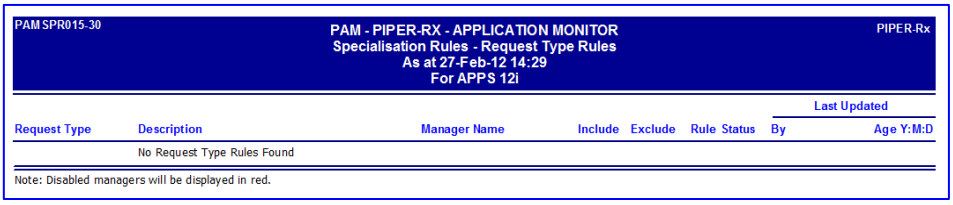
PAM SPR014-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Specialisation Rules - Complex Rule Definitions
As at 27-FEB-12 14:29
For APPS 12i

Rule Name	Rule Type	Appn Id	Description	Rule Status	Last Updated By	Last Updated Age Y:M:D
			No Complex Rule Definitions Found			

3.1.6 Request Types

You can use ***PAMreports*** – Specialisation Rules PAMSPR015 Request Type Rules to list the current application request type based specialisation rules:

Example **PAMSPR015-30 Request Type Rules** report



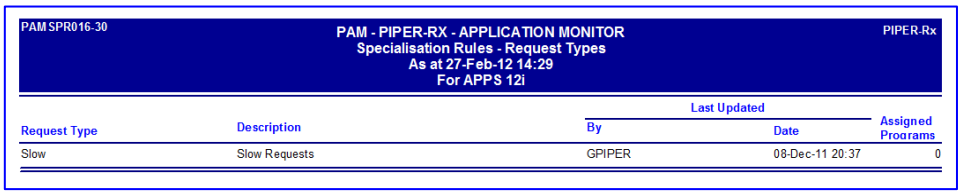
PAM SPR015-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Specialisation Rules - Request Type Rules
As at 27-Feb-12 14:29
For APPS 12i

Request Type	Description	Manager Name	Include	Exclude	Rule Status	Last Updated By	Last Updated Age Y:M:D
	No Request Type Rules Found						

Note: Disabled managers will be displayed in red.

The next question with request types is what request types have been defined? ***PAMreports*** – Specialisation Rules PAMSPR016-30 Request Types will list the currently defined request types.

Example **PAMSPR016-30 Request Types** report



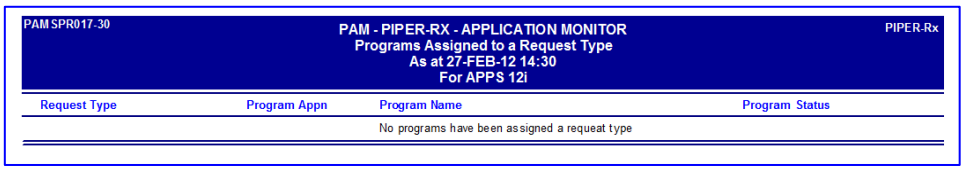
PAM SPR016-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Specialisation Rules - Request Types
As at 27-Feb-12 14:29
For APPS 12i

Request Type	Description	Last Updated By	Last Updated Date	Assigned Programs
Slow	Slow Requests	GPIPER	08-Dec-11 20:37	0

The final question with request types is what programs have been assigned to the request type?

You can use ***PAMreports*** – Specialisation Rules PAMSPR017 Request Type Programs to list the current application request type based specialisation rules:

Example **PAMSPR017 Request Type Programs** report



PAM SPR017-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx
Programs Assigned to a Request Type
As at 27-FEB-12 14:30
For APPS 12i

Request Type	Program Appn	Program Name	Program Status
		No programs have been assigned a requeat type	

This group of ***PAMreports*** should help you navigate the complexities of specialisation rules.

3.1.7 Turning SPR-001 alert off and on

The ***PAM*** SPR-001 alert can be turned off using the following ***PAM*** API:

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

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

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

3.2 SPR-002 - Alert when a manager is disabled and there are specialisation rules assigned

Once per day ***PAM*** will check for specialisation rules that are assigned to a disabled concurrent manager. When ***PAM*** detects such an occurrence ***PAM*** will generate the following alert e-mail:

Example *PAM* SPR-002 – SPR rules assigned to a disabled manager alert message

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

Customer = Company name
Site = Site name
Alert Level = **Informational**
Detected = 28-Feb-12 (Tue) 20:01:05
Alert Frequency = 1 Week

A specialisation rule was found assigned to a disabled manager ABC_MANAGER;

Alert Information:

SPR-002 – ONE OR MORE CONCURRENT PROGRAMS ARE ASSIGNED TO A CONCURRENT MANAGER THAT HAS BEEN DISABLED

This ***PAM*** alert is raised when a concurrent manager has been disabled and one or more programs are assigned to that manager via specialisation rules.

Where this occurs one or more concurrent programs may no longer have a concurrent manager to run them. These requests will most likely have a status of pending error.

You can use ***PAMreports*** – Specialisation Rules **PAMSPR020 Rules for Disabled Managers** to list all specialisation rules assigned to a disabled concurrent manager.

3.2.1 What to do with this information

You can use ***PAMreports*** – Specialisation Rules PAMSPR020 Rules for Disabled Managers to list all the specialisation rules (excluding Complex rules) assigned to a disabled manager:

Example **PAMSPR020 Rules for Disabled Managers** report

PAM SPR020-30

PAM - PIPER-RX - APPLICATION MONITOR
Specialisation Rules Assigned to Disabled Managers
As at 27-Feb-12 14:30
For APPS 12i

PIPER-Rx

Manager Name	Rule Type	Type Id	Description	Include	Exclude
			No Rules Found		

3.2.2 Turning SPR-002 alert off and on

The ***PAM*** SPR-002 alert can be turned off using the following ***PAM*** API:

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

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

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

3.3 SPR-003 - Alert when a program is assigned a request type that does not exist

It is possible in some releases to delete a request type that is still assigned to a concurrent program. Once per day ***PAM*** will check the application's request type assignments for programs assigned to a request type that does not exist. When ***PAM*** detects such an occurrence ***PAM*** will generate the following alert e-mail:

Example *PAM* SPR-003 – SPR Programs assigned to a request type that does not exist

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

Customer = Company name
Site = Site name
Alert Level = **Informational**
Detected = 28-Feb-12 (Tue) 20:01:05
Alert Frequency = 1 Week

A program was found assigned to a Request Type that does not exist

Alert Information:

SPR-003 – ONE OR MORE CONCURRENT PROGRAMS HAS BEEN ASSIGNED TO A REQUEST TYPE THAT HAS BEEN DELETED

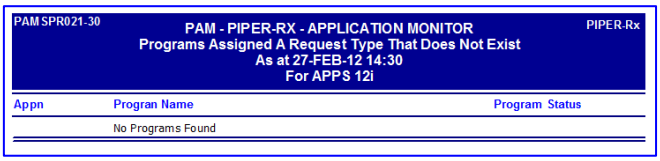
This ***PAM*** alert is raised when a request type is deleted and one or more concurrent programs have been assigned to that request type.

You can use ***PAMreports*** – Specialisation Rules **PAMSPR021 Programs With Missing Request Type** to list concurrent programs assigned to a deleted request type.

3.3.1 What to do with this information

You can use ***PAMreports*** – Specialisation Rules PAMSPR021-30 Programs With Missing Request Type to list all the programs that have been assigned a request type that no longer exists:

Example **PAMSPR021-30 Programs With Missing Request Type** report


PAM SPR021-30
PAM - PIPER-RX - APPLICATION MONITOR
Programs Assigned A Request Type That Does Not Exist
As at 27-FEB-12 14:30
For APPS 12i
PIPER-Rx

Appn	Progran Name	Program Status
	No Programs Found	

You can then use this list to remove the request type from the program definition via the application concurrent > Program > define screen.

3.3.2 Turning SPR-003 alert off and on

The ***PAM*** SPR-003 alert can be turned off using the following ***PAM*** API:

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

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

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

3.4 SPR-004 - Alert when a run alone program has been submitted

As mentioned above, when a program defined as “Run Alone” is submitted, no other programs can run whilst that program is running. ***PAM*** will check and alert as follows when a run alone program has been submitted:

Example *PAM* SPR-004 – Run alone program has been submitted alert

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

Customer = Company name
Site = Site name
Alert Level = **Warning**
Detected = 28-Feb-12 (Tue) 20:01:05
Alert Frequency = 1 Week

A run alone program (ABC Program)-(1234567) has been submitted by GPIPER at 28-**Feb-12 (Tue) 20:00**

Alert Information:

SPR-004 – A concurrent program defined as run alone has been submitted

This ***PAM*** alert is raised when ***PAM*** detects a concurrent program that has been defined as run alone has been submitted or is running.

When a run alone program has been requested, no other program can run at the same time.

The effect of running a program configured to run alone is that all pending requests will be set to standby until the run alone request completes.

You can use ***PAMreports*** – Specialisation Rules **PAMSPR032 Run Alone Programs** to obtain a list of all programs set to run alone and the last time those programs were submitted.

You can use ***PAMreports*** – Specialisation Rules **PAMSPR033 Pending Standby List** to obtain a list of all pending requests with a status of standby.

3.4.1 What to do with this information

You can use ***PAMreports*** – Specialisation Rules PAMSPR032-30 Run Alone Programs to list all the programs defined as run alone:

Example **PAMSPR032-30 Run Alone Programs** report

PAM SPR032-30 — PAM - PIPER-RX - APPLICATION MONITOR — PIPER-Rx

Run Alone Programs
As at 27-FEB-12 14:31
For APPS 12i

App'n	Program	Program Name	Program Status	Last Run Details: Request Id	Last Run Details: Start Date
AP	JLZZPRAD	Argentine/Colombian Payables Remittance Advice	Enabled		
AR	ARHAPREL	Account to Party Relationships Migration Program	Disabled		
CN	CN_ODI_CALL	Incentive Compensation Analytics - ODI	Enabled		
FV	FVRDIEFR	Delinquent Invoices Eligible for Referral Report	Enabled		
FV	FVFBWTIM	Fund Balance with Treasury Reconciliation Import Process	Disabled		
JL	JLZZPRAD	Argentine/Colombian Payables Remittance Advice	Disabled		
MRP	MRCBLC	MRP BOM/Wip Loop Checking Program	Enabled		
POS	POSSJAVA	POS Supplier Hub Java Concurrent Program	Enabled		
RRS	RRSCREATEASSETS	RRS : Create Asset Instances For Sites	Enabled		
RRS	RRSCREATEPL	RRS : Create Property Locations For Sites	Enabled		
RRS	RRSSJAVA	RRS Spreadsheet Java Concurrent Program	Enabled		

Estimated on-line history days in fnd_concurrent_requests: 1,243

This report lists the last run details of the run alone request which can be useful for identifying if the program was run during peak processing periods.

3.4.2 Turning SPR-004 alert off and on

The ***PAM*** SPR-004 alert can be turned off using the following ***PAM*** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'SPR-004', 'N');
```

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

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'SPR-004', 'Y');
```

3.5 SPR-005 - Alert when the number of standby requests exceeds threshold

When a program is prevented from running as a result of an incompatibility rule, that program is place in a status of “pending standby” by the conflict resolution manager. ***PAM*** will generate an alert e-mail when the number of pending standby requests exceeds the ***PAM*** threshold value:

Example *PAM* SPR-005 – Pending Standby Alert e-mail

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

Customer = Company name
Site = Site name
Alert Level = **Warning**
Detected = 28-Feb-12 (Tue) 20:01:05
Alert Frequency = 1 Week

The number of Pending Standby requests 28 has exceeded the threshold value of 10

Alert Information:

SPR-005 –THE NUMBER OF REQUESTS WITH A STATUS OF PENDING STANDBY HAS EXCEEDED THE *PAM* THRESHOLD VALUE

This ***PAM*** alert is raised when the number of pending requests with a status of standby has exceeded the ***PAM*** threshold value.

A request that has a status of standby indicates that the Conflict Resolution Manager (FNDCRM) has determined that the request is incompatible with one or more requests that are running and will place the request in standby until the incompatible request has completed.

You can use ***PAMreports*** – Specialisation Rules **PAMSPR034 Requests Waiting On** to obtain a list of running requests and the pending requests that are waiting for the running request to complete.

PAMreports – Specialisation Rules also provides the following reports to list the application's program incompatibility rules:

PAMSPR030 Programs Incompatibilities List
PAMSPR031 Programs Incompatibilities By Program

3.5.1 What to do with this information

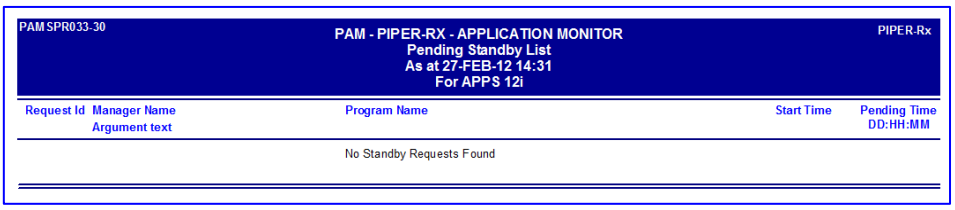
As mentioned in the alert e-mail text you can use the two (2) incompatibility reports both of which have been described earlier in this tutorial.

- PAMSPR030 Programs Incompatibilities List
- PAMSPR031 Programs Incompatibilities By Program

When a request is prevented from running by the conflict resolution manager albeit because of an incompatibility rule or a run alone request, that request will be suspended with a status of “Standby” until the conflict resolution manager releases the program to run.

PAMreports – Specialisation Rules PAMSPR033 Pending Standby List lists all requests with a current status of standby:

Example **PAMSPR033 Pending Standby List** report


PAM SPR033-30

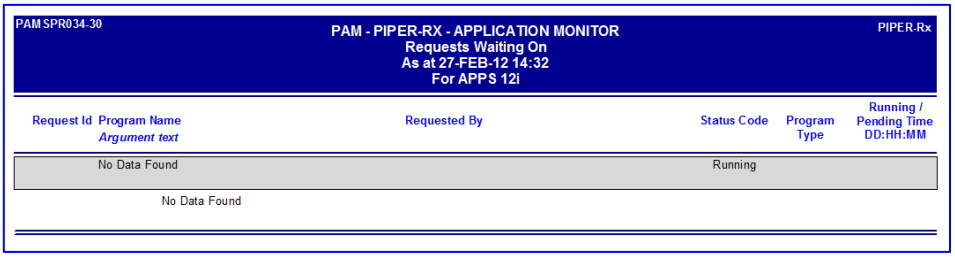
PAM - PIPER-RX - APPLICATION MONITOR
Pending Standby List
As at 27-FEB-12 14:31
For APPS 12i

PIPER-Rx

Request Id	Manager Name Argument text	Program Name	Start Time	Pending Time DD:HH:MM
		No Standby Requests Found		

When there are a number of pending standby requests it is often beneficial to identify the concurrent program that is causing the backlog of standby requests. ***PAMreports*** – Specialisation Rules PAMSPR034-30 Requests Waiting on lists those programs that are running and the programs that are being held on standby:

Example **PAMSPR034-30 Requests Waiting on** report


PAM SPR034-30

PAM - PIPER-RX - APPLICATION MONITOR
Requests Waiting On
As at 27-FEB-12 14:32
For APPS 12i

PIPER-Rx

Request Id	Program Name *Argument text*	Requested By	Status Code	Program Type	Running / Pending Time DD:HH:MM
	No Data Found		Running		
	No Data Found				

The information in this report is based on the application specialisation rules.

3.5.2 Setting the Pending standby requests threshold

The pending standby requests threshold can be set using the following ***PAM*** API:

```
exec piper_rx_pam_api_2.pam_threshold_spr005_set (10);
```

Parameter 1:

The number of pending standby requests that must be in existence for the alert to be raised

3.5.3 Turning SPR-005 alert off and on

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

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

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

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

3.6 Excluding programs from the *PAM* specialisation rule check

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

You can use ***PAMreports*** -Config **PAMC009 PAM Program Monitor List t**o list the current monitored programs exceptions.

Example **PAMC009 PAM Program Monitor List** report

PAMC009-30 — PAM - PIPER-RX - APPLICATION MONITOR — PAM Concurrent Program Monitor List — As at 04-Mar-12 11:00 — For APPS 12i — PIPER-Rx

Application	Prog Id	Program Name	Check Status	Exists	Completed: Error	Completed: Warning	Submitted	Exclude From: Duplicates	Exclude From: Long Running	Exclude From: SP Rules
0 (FND)	0	Activate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	1	Deactivate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	3	Restart Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	4	Abort Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	5	Shutdown Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	6	Startup Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	31659	Report Set	Enabled	No	No	No	No	No	Yes	Yes
0 (FND)	32263	Purge Concurrent Request and/or Manager Data	Enabled	Yes	No	No	No	No	Yes	No
0 (FND)	32592	Purge Signon Audit data	Enabled	Yes	No	No	No	No	Yes	No
0 (FND)	36034	Request Set Stage	Enabled	No	No	No	No	Yes	Yes	Yes
0 (FND)	36888	Workflow Background Process	Enabled	No	No	No	No	Yes	Yes	No
178 (ICX)	36662	Delete data from temporary table	Enabled	Yes	No	No	No	No	Yes	No

4 Changing PAM monitored programs exceptions

4.1 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;
```

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 the ***PAM*** check program exists check (CP-001) is run.

Example **PAMC009 PAM Program Monitor List** report

PAMC009-30 — PAM - PIPER-RX - APPLICATION MONITOR — PAM Concurrent Program Monitor List — As at 04-Mar-12 11:00 — For APPS 12i — PIPER-Rx

Application	Prog Id	Program Name	Check Status	Completed: Exists	Completed: Error	Completed: Warning	Submitted	Exclude From: Duplicates	Exclude From: Long Running	Exclude From: SP Rules
0 (FND)	0	Activate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	1	Deactivate Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	3	Restart Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	4	Abort Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	5	Shutdown Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	6	Startup Concurrent Manager	Enabled	No	Yes	Yes	Yes	No	Yes	No
0 (FND)	31659	Report Set	Enabled	No	No	No	No	No	Yes	Yes
0 (FND)	32263	Purge Concurrent Request and/or Manager Data	Enabled	Yes	No	No	No	No	Yes	No
0 (FND)	32592	Purge Signon Audit data	Enabled	Yes	No	No	No	No	Yes	No
0 (FND)	36034	Request Set Stage	Enabled	No	No	No	No	Yes	Yes	Yes
0 (FND)	36888	Workflow Background Process	Enabled	No	No	No	No	Yes	Yes	No
178 (ICX)	36662	Delete data from temporary table	Enabled	Yes	No	No	No	No	Yes	No

4.2 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

4.3 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.

5 Disclaimer

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