

## Giving In To The Urge To Purge

You've probably heard the age-old DBA's complaint – "if it weren't for all these darned users putting data into the database, it would run just fine." The powerless DBA watches as her database grows by megabytes and gigabytes, always wondering if all that data really needs to be saved.

Oracle offers many purge and delete programs for different parts of the applications that you may be able to run to keep the size of your database under control. Deciding which programs to run and which parameters to use, however, is a daunting task. Your best bet is to track what the largest objects are in your database, determine which module owns those objects, and then work with your functional users to determine if any of those tables have data that can be either completely removed or moved to a history table. Your functional users will need to refer to a combination of the Oracle manuals for the module in question, and most likely MetaLink, to see if other users have had any issues with Oracle's purge programs. You'll likely find that business rules concerning data retention make it impossible to purge data from many of the large tables, but it's certainly worth investigating to determine if this is the case. You and your functional users will need to be extremely careful in testing Oracle's purge and delete programs – once the data is gone, it's gone for good! You should carefully research how programs work, what tables they delete from, what criteria they use for deleting, what other companies say they're doing, and whether other companies have had problems – either with functionality or performance, with the programs.

This article describes the few tables that belong to the DBA, Workflow Administrator or Applications System Administrator. Rather than take you down the track of debating whether you really need all that pesky Order Entry or General Ledger data, this article focuses on the administrative tables that can be purged. Your administrator gets to decide, with input from management and users in some cases, when to purge or delete from these tables. Best of all, this article will tell you what will happen if you *never* delete from these tables, and what will happen if you delete too much, and provides guidelines for deciding which parameters to choose as you schedule the concurrent programs that maintain these tables.

Purging topics covered include:

- How to find potential purge/delete candidates
- The Purge Concurrent Requests and/or Manager Data concurrent program
- The Purge Obsolete Workflow Runtime Data concurrent program
- The Purge Signon Login Data concurrent program
- The Delete Data from Temporary Tables concurrent program
- The Delete temporary data of PO revisions in Self Service Applications concurrent program

### Look for the Big Tables!

To get started on determining if there are candidates for purging, your Database Administrators can help your Applications System Administrators proactively spot storage problems by monitoring the size and

growth of the largest tables and indexes in the database. A simple query like the following, which is looking for tables and indexes that are larger than 100 megabytes and are not owned by the SYS user, run perhaps once a month, can give insight. In fact, we've discovered the usefulness of the different purge and delete programs described in the next few paragraphs for the most part from looking for big tables and searching on MetaLink to see if there are maintenance programs to control their size:

```
select substr(OWNER,1,15) "OWNER",
       substr(segment_NAME,1,30) "NAME",
       substr(segment_type,1,12) "TYPE",
       bytes "BYTES",
       extents "EXTENTS"
  from sys.dba_segments where
       segment_type in ('INDEX','TABLE') and
       bytes > 100000000 and owner not in ('SYS') order by bytes desc;
```

OWNER	NAME	TYPE	BYTES	EXTENTS
<b>APPLSYS</b>	<b>FND_ENV_CONTEXT</b>	<b>TABLE</b>	<b>2245386240</b>	<b>183965</b>
<b>APPLSYS</b>	<b>FND_ENV_CONTEXT_U1</b>	<b>INDEX</b>	<b>1292255232</b>	<b>102467</b>
APPLSYS	DR\$FND_LOBS_CTX\$I	TABLE	540106752	13
CS	CS_CP_AUDIT	TABLE	479158272	1448
HRI	HR_EDW_WRK_CMPSTN_FSTG	TABLE	310525952	13
APPLSYS	DR\$FND_LOBS_CTX\$X	INDEX	237740032	11
GL	GL_BALANCES	TABLE	189874176	189
FII	FII_AR_TRX_DIST_FSTG	TABLE	134348800	1
OPI	OPI_EDW_PERD_MARGIN_F	TABLE	134316032	1
OPI	OPI_EDW_RES_UTIL_F	TABLE	134316032	1
ISC	ISC_EDW_BOOK_SUM1_FSTG	TABLE	134283264	1
OKC	OKC_TIMEVALUES_B	TABLE	125730816	1
APPLSYS	FND_CONCURRENT_REQUESTS	TABLE	115261440	1
APPLSYS	WF_ITEM_ATTRIBUTE_VALUES_PK	INDEX	110804992	1
HR	PAY_RUN_RESULT_VALUES_PK	INDEX	100466688	1

15 rows selected.

Right off the bat we'll tell you what this instance's problem is – the “Purge Concurrent Requests and/or Manager Data” concurrent program is not being run. Once that request starts running, the FND\_ENV\_CONTEXT table will shrink down to something reasonable. We figured this out, by they way, by looking up FND\_ENV\_CONTEXT on MetaLink – we found a user complaining that their table was huge and growing quickly, and found a hit on MetaLink that said that the “Purge Concurrent Requests and/or Manager Data” program needed to be run to remove those unneeded records. A good DBA might also watch for opportunities to regain lost space, so once the Purge is scheduled to run on a regular basis, it is worthwhile to schedule maintenance downtime and export/import the FND\_ENV\_CONTEXT table to try to lower the high water mark on this 2+ gigabyte table. The export/import will also reduce the size of the FND\_ENV\_CONTEXT\_U1 index.

Another way to search for purge-able data is to search for concurrent programs seeded by Oracle that do just that – if a program has the word “purge” or “delete” in it, then you can search for that program on Metalink to see if it's something you should be running. To do so, simply log in as a user with the System Administrator responsibility, select Concurrent | Program | Define, and query on the program name field. Try %urges% and %elete% and see what you find. When you research your purge and delete programs on Metalink, search both on the concurrent program name as well as the program's short name.

Based on our experience, following are concurrent programs that your Applications System Administrator should consider running:

## Purge Concurrent Requests and/or Manager Data

We had a customer who, when we first started working with them, complained that whenever they submitted a request, they had to wait for about half a minute for the request to commit, and when they tried to query up the status of records in the concurrent manager, they often waited for more than a minute. Our investigation showed that they hadn't run the "Purge Concurrent Requests and/or Manager Data" concurrent program in the three years that they had been running the applications. Performance improved dramatically once they started running this program.

The "Purge Concurrent Requests and/or Manager Data" concurrent program will delete information about *completed* concurrent requests from a set of tables. It can also delete the log and output files for those concurrent requests from your UNIX file system. Most companies run this report and delete all records older than 7 -31 days.

If you choose Entity ALL, the purge program deletes rows for completed requests from the FND\_CONCURRENT\_REQUESTS, FND\_RUN\_REQUESTS, FND\_CONC\_REQUEST\_ARGUMENTS, FND\_CONC\_STAT\_LIST, FND\_CONCURRENT\_PROCESSES, FND\_CONC\_STAT\_SUMMARY, FND\_CONC\_PP\_ACTIONS, FND\_RUN\_REQ\_PP\_ACTIONS and FND\_DUAL tables. It also deletes your older log and out files. Using this program rather than deleting log and out files using UNIX commands works better, because the Purge will check against the FND tables and only delete files that have a status of completed.

In the example, we've chosen ALL for Entity. Valid choices are ALL, MANAGER and REQUEST. If you choose MANAGER, then the log information stored in APPLLOG will be deleted, but the output files stored in APPLOUT will not. If you choose REQUEST, then the output files will be deleted but the log files will not be deleted, and the associated rows in the FND\_ENV\_CONTEXT table will not be deleted. Rows are written to FND\_ENV\_CONTEXT when a new concurrent manager process or service is started so this table can get very large if you don't select ALL for Entity.

For Mode, we chose Age so we could delete files older than the number of days specified in Mode Value. You could also choose Count for Mode – you might use a Mode of Count after cloning an instance to delete all the old completed concurrent request data that carried over from the source system. In that case, you would choose Mode of Count and Mode Value of 0 to delete all completed records.

There are two schools of thought on the best way to clean out the concurrent manager tables after cloning an instance. Some folks recommend that you simply truncate the FND\_CONCURRENT\_REQUESTS table on the target instance after cloning and before starting the concurrent manager. We prefer to run the "Purge Concurrent Requests" concurrent program rather than truncate the tables because truncating the table will remove all of the scheduled programs along with all the completed programs – we put all scheduled requests on hold before cloning to save the scheduled requests and then allow users to either cancel the scheduled requests or take them off hold after cloning on the theory that this is easier for users than resubmitting all of their requests.

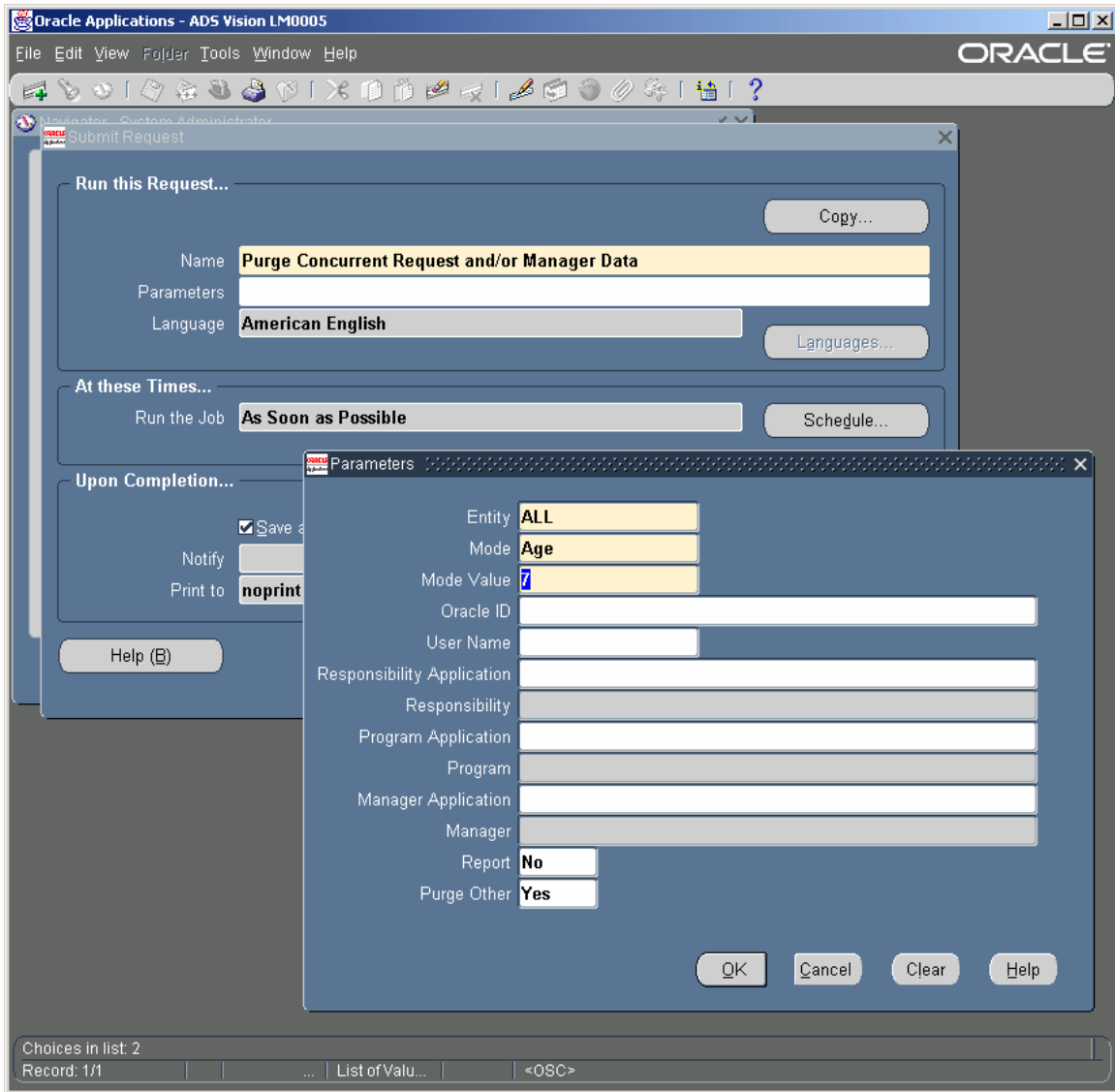


Figure 1 Purge Concurrent Request and/or Manager Data

In Figure 1, the Applications System Administrator would schedule the “Purge Concurrent Request and/or Manager Data” program to run nightly, deleting all completed requests older than 7 days. Since you run nightly backups (you do, don’t you?), deleting data and files shouldn’t be a problem – you can always retrieve them from tape if necessary.

What if you delete too much data? It certainly can happen – you need to keep completed concurrent request information on your system long enough so users can see the output! Don’t delete everything older than one day, then, or the user who schedules a request to run on Saturday won’t get to see the results on Monday when they come in to work. Generally folks keep 7-31 days worth of concurrent request output.

What if you want to hold onto historical information to assess performance? Let’s say your manager wants to know what performance is like during the month-end close, or say you know you need to tune the different concurrent managers and want to know how the concurrent managers are performing. You

have two options – hold onto the data for a longer period of time, which risks performance issues for everyone who needs to submit or query on a submitted request, or save the historical information elsewhere. You could, for example, create a delete trigger on the `fnd_concurrent_requests` table that puts data into a historical table whenever a delete occurs. Whenever the purge runs, then, the data would be deleted from `fnd_concurrent_requests` after landing in another table, say, `fnd_concurrent_requests_history`. You could then run programs that gather performance information against a view that joins `fnd_concurrent_requests` with `fnd_concurrent_requests_history`.

### **Purge Obsolete Workflow Runtime Data**

Similar to the FND tables that are purged with the “Purge Concurrent Request and/or Manager Data” concurrent program, the workflow tables also need to be purged depending on the complexity of the workflows and how often they are used. Oracle provides the `WF_PURGE` package to assist with this effort, and a concurrent program called “Purge Obsolete Workflow Runtime Data”. MetaLink Note: 132254.1: “Speeding up and Purging Workflow v2.5 & 2.6”, gives more details on how to improve workflow performance.

Workflow records data about each step of a running workflow in the `WF_ITEMS`, `WF_ITEM_ACTIVITY_STATUSES`, `WF_ITEM_ACTIVITY_STATUSES_H`, `WF_ACTIVITY_ATTR_VALUES`, `WF_NOTIFICATIONS`, and `WF_NOTIFICATION_ATTRIBUTES` tables. The size of these tables grows very rapidly and if not managed can cause performance degradation. The “Purge Obsolete Workflow Runtime Data” concurrent program is very similar to purging concurrent manager history with the “Purge Concurrent Requests and/or Manager Data” concurrent program. When you first set up workflow, you can choose scheduling timings and history retention period to be the same as your concurrent programs. Even if you are running the “Purge Obsolete Workflow Runtime Data” concurrent program, you should continue to monitor the size of your tables and whether you are experiencing performance issues and adjust your parameters accordingly.

The parameters for the “Purge Obsolete Workflow Runtime Data” concurrent program are:

- Item Type - You can specify Item Type when you are trying to get rid of a specific workflow (for example, the FA accounts generator is seeded to be permanent and if you are de-bugging it, then you would run the purge specifying the item-type, age=0, and persistence type = Permanent to get rid of all of the debug history).
- Item Key - We haven't really found a reason to use the Item Key - that gets rid of a specific instance of a specific workflow. Since this program only works on workflows that are completed, there isn't much call for removing data for a specific instance. For the "generic" purge, Item Type and Item Key should be left blank.
- Age - The Age field is for how long AFTER the workflow finishes you want to keep the data before purging it. That depends on what a company does with the workflow data - are they gathering stats on number of workflows finished? Are they moving the approval history to other tables where Oracle didn't keep an approval history? We believe the history ought to be purged often, at least for data older than 30 days. Many companies purge data older than a week. If your company is doing statistics gathering and

moving approval history, it will do so by using custom programs, so these programs could be run prior to the purge by including the custom programs and the purge in a request set.

- Persistence Type - Persistence type is assigned to a workflow when you define it (look at the properties page of the Item Type in the builder tool). The choices are temporary or permanent. Most of the Oracle seeded workflows have a persistence type of temporary.

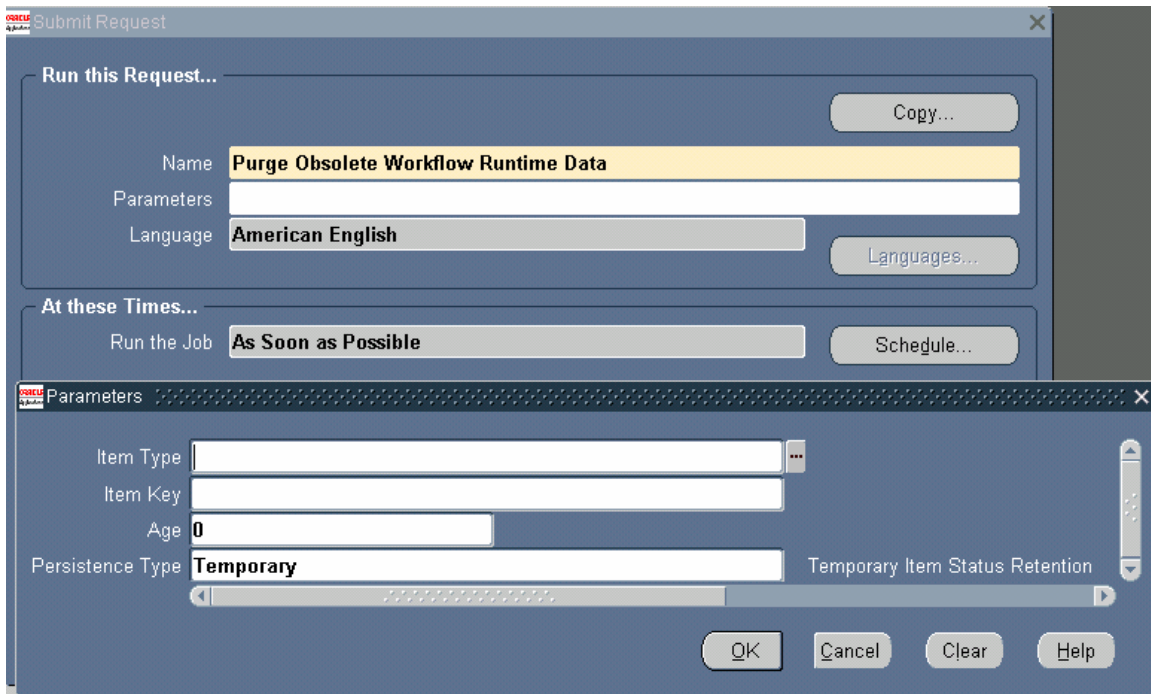


Figure 2 The Purge Obsolete Workflow Runtime Data concurrent program

Just as you don't keep concurrent manager data forever, you don't want to keep workflow data forever either. It makes administration burdensome to have to sort through all that history. The System Administrator should set up the “Purge Obsolete Workflow Runtime Data” concurrent program for items with a Persistence Type of Temporary, and should schedule this program to run daily, choosing an appropriate Age for data retention, and leaving Item Type and Item Key blank.

If you've been running the Applications for some time and did not know you were supposed to run the “Purge Obsolete Workflow Runtime Data” concurrent program, your Database Administrator may need to resize the affected tables once you've gotten a significant amount of obsolete data removed. Your Database Administrator may conclude that exporting and importing the Workflow tables is necessary to release the empty space that the Purge frees up. This is not a difficult task, but does require that the database be made unavailable to users temporarily while your DBA runs the export/import processes.

Each company must decide how long they wish to save the history of completed workflows. While for some workflows (such as Journal Batch Approval) the workflow runtime tables provide the only history of the approvals, these workflow tables are not the appropriate place for approval history storage. If you wish to save the approval history, the approval records can be extracted to custom tables prior to purging the workflow history. If you only need to keep approval history for a short time, then just adjust the age

parameter. In order to extract this history, you will need to know the internal name of the activity that asks for approval.

### Purge Signon Login data

The “Purge Signon Login data” concurrent program deletes audit information from the FND\_LOGINS, FND\_UNSUCCESSFUL\_LOGINS, FND\_LOGIN\_RESPONSIBILITIES, and FND\_LOGIN\_RESP\_FORMS tables. You should schedule this program to run daily. You should pick a date for the Audit Date based on your company’s security data retention requirements. Most companies pick a date a week or two prior to the current date.

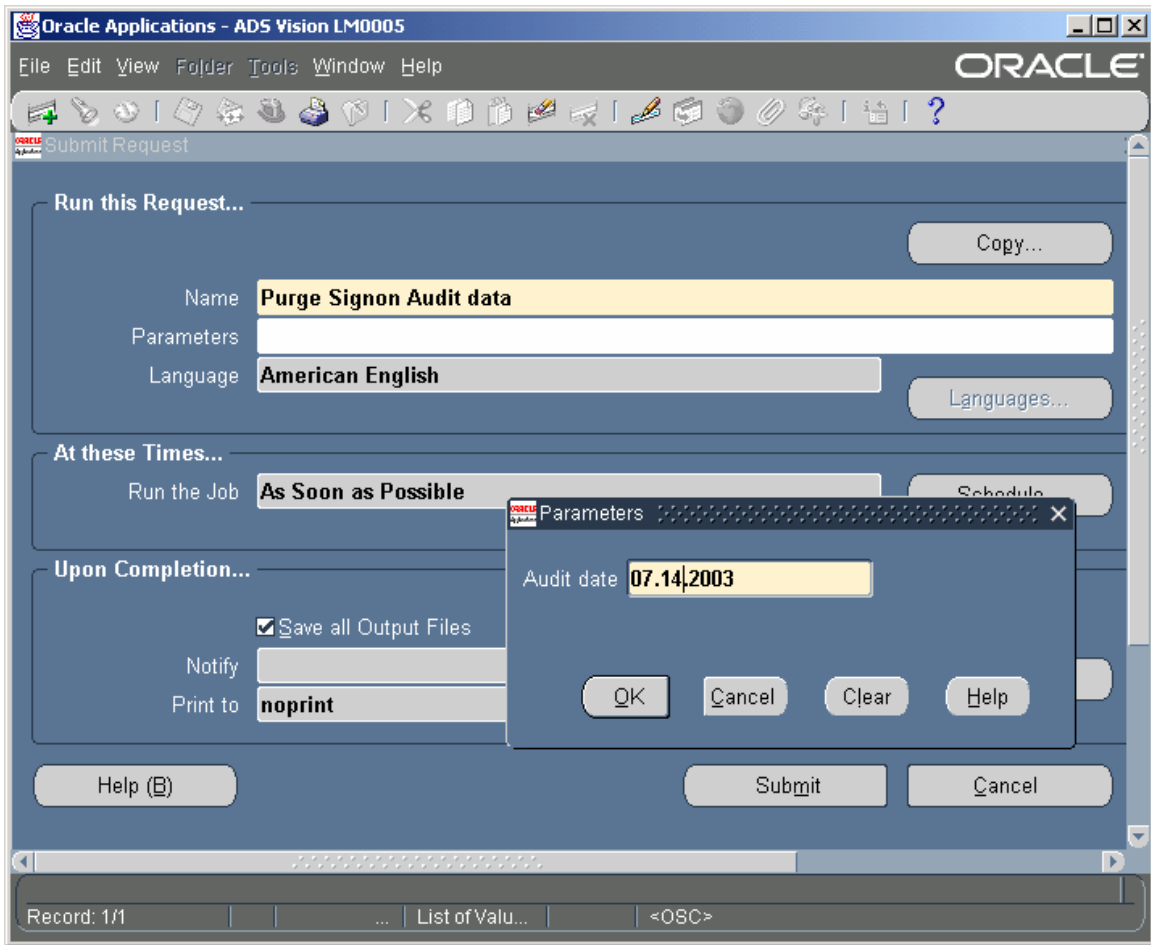


Figure 3 Schedule the Purge Signon Audit data concurrent program to run every day

Check the Increment Date Parameters box so that the Audit date will advance a day each time the program runs:

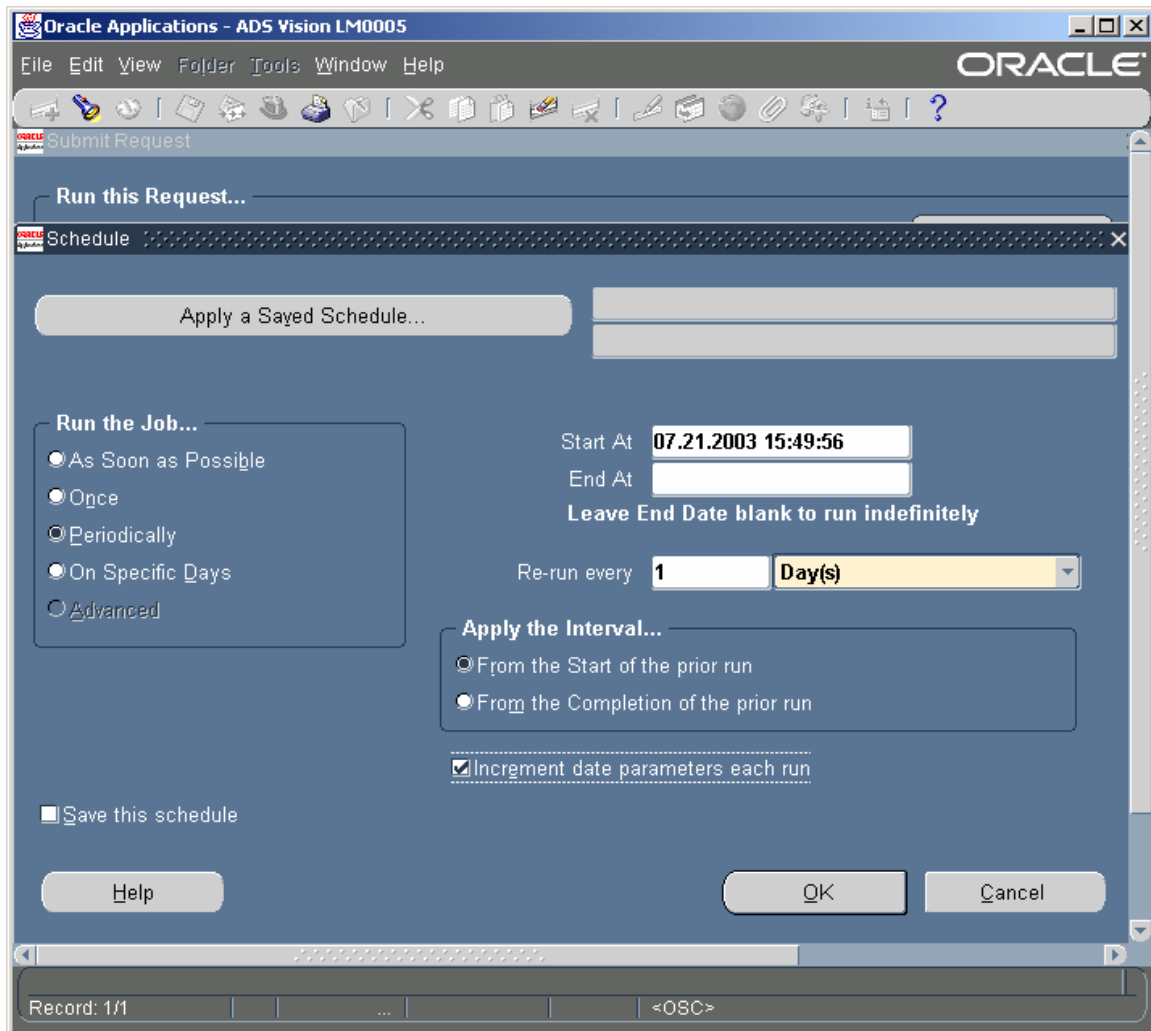


Figure 4 Check the "Increment date parameters each run" box to the Audit Date will advance each time

What happens if you don't run the purge? We had one customer who ran the applications for several years without running the "Purge Signon Audit data" program. After migrating from RDBMS Version 8.0.6 to Version 8.1.7.4, they found that the System Administrator's Security | User | Monitor screen hung trying to pull up records. Testing showed the problem was caused because the FND\_LOGINS table had more than 1.5 million rows in it. Once we ran the "Purge Signon Audit data" program, performance returned to normal.

### Delete data from temporary tables

This program deletes data from the icx\_sessions, icx\_transactions, icx\_text, icx\_context\_results\_temp, icx\_failures and icx\_requisitioner\_info tables. The program hard codes the deletes to remove all data older than the current time minus 4 hours, but the program code says you can change how much time you wait to delete by modifying the code. Of course, that would be a customization, subject to being overwritten in the future, so unless you're very uncomfortable with the 4 hour number, we recommend leaving the code, which is located in \$ICX\_TOP/sql/ICXDLTMP.sql, alone. These tables will get very large over time –

the icx\_sessions table has an entry for every time someone logs into the self services web applications. You should schedule this program to run daily.

### Delete temporary data of PO revisions in Self Service Applications

The “Delete temporary data of PO revisions in Self Service Applications” concurrent program deletes records from the temporary table ICX\_PO\_REVISIONS\_TEMP, where all the records for differences are stored. When you run this concurrent request, it asks for a date and purges all records that are older than this date. We recommend scheduling this report to run nightly, with the date set back one or two weeks. When you schedule this report, set it up to run daily and click on the “Increment date parameters each run” so the date will advance each time the concurrent program runs. You should work with your Purchasing Functional Users to ensure that they concur on how much data to save.

While both the “Delete data from temporary tables” and “Delete temporary data of PO revisions in Self Service Applications” concurrent programs can be run from the Self Service Web Applications Manager responsibility, it might be easier to run all administrative reports such as these from the System Administrator responsibility. To add these reports to the System Administrator’s Report Group, choose Security | Responsibility | Request and query the Group “System Administrator Reports”. Add the two programs:

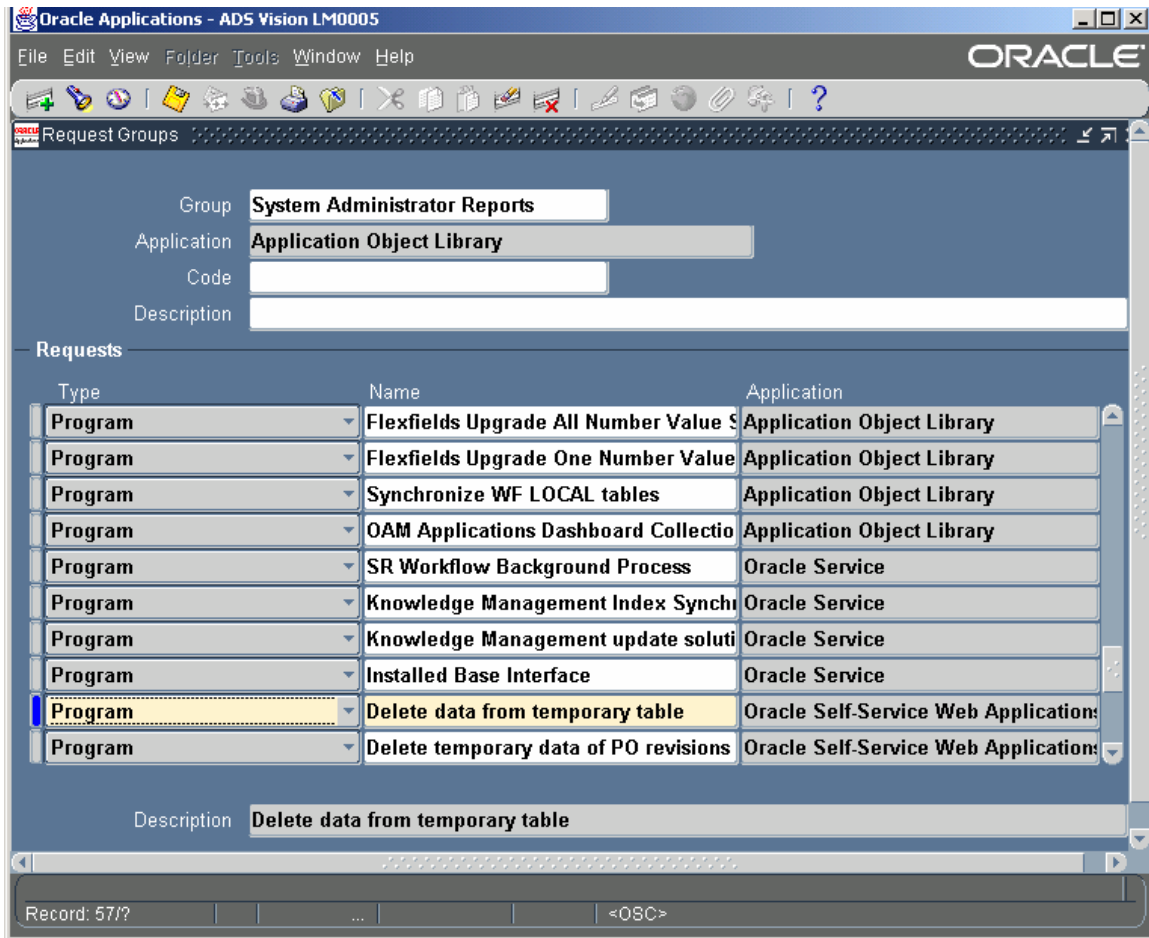


Figure 5 Add the two programs to the "System Administrator Reports" Request Group

## Conclusion

If you haven't been running these five concurrent programs in your environment, there's a good chance you'll look like a hero once you discover them. You should schedule these concurrent programs, understand what happens if they don't run, and document their significance so that they don't slip through the cracks if you move on to another position. Also, if the affected tables have grown very large before you began purging from them, consider exporting/importing those tables to bring their high water mark back down.

## For More Information

This article is an excerpt from the book "Installing, Upgrading and Maintaining Oracle Applications 11*i* (or When Old Dogs Herd Cats – Release 11*i* Care and Feeding)", available at [www.solutionbeacon.com](http://www.solutionbeacon.com). Written by Barb Matthews, John Stouffer, Karen Brownfield and Randy Giefer, this book covers topics that include 11*i* Concepts and Architecture, 11*i* New Administration Features, 11*i* Installation, Upgrading or Migrating to 11*i*, Maintaining 11*i*, Administering 11*i*, Setting Up the Concurrent Manager, Using the Concurrent Manager, Workflow Setup, Using Workflow Builder, Workflow Care and Feeding, and Tuning & Troubleshooting. Two websites, [www.oncalldb.com](http://www.oncalldb.com) and [www.solutionbeacon.com](http://www.solutionbeacon.com) offer additional papers and scripts for managing your Oracle Applications 11*i* environments.