

Workflow Performance Tuning

Karen Brownfield
Solution Beacon, LLC

Workflow is similar to other tools in the application technology toolset in that it needs to be monitored and occasionally tuned. It's not just about deleting history and ensuring all workflows are functioning (although this is certainly part of it). This paper discusses several additional methods to increase the performance of the workflow components.

The discussion will focus on Release 11.5.10 with ATG_PF.H.delta.5 (RUP 5) or ATG_PF.H.delta.6 (RUP 6) and Release 12 (all releases)

Which Are You?

Workflow is an integral part of E-Business Suite Releases 11i and Release 12. Even if your company declares it is not using workflow, the E-Business Suite is. Every time you make changes to employees, to users, to customer contacts, to positions, and/or to responsibilities, Oracle initiates Business Events and these Business Events are part of workflow.



Workflow is both a technology and an application. It is only supported within E-Business Suite and is part of the Application Technology Group (ATG). Like any E-Business Suite application it should be properly setup and like any technology it should be monitored. Companies that ignore workflow and expect it to perform perfectly without any intervention or monitoring are behaving like ostriches.



“The opposite of the ostrich is the rooster who is alert and awake

early to see what is on the horizon. Rather than fear, he crows loudly a warning to be heeded by all.” (<http://users.cybertime.net/~ajgood/ostrich.html>) Roosters read not only the manuals for the workflow technology, but also the workflow sections in each application they are running. These companies monitor MetaLink for new documents and new practices that can positively impact the performance of workflow in their environments.

Patch Current

Because workflow is part of the ATG group, and this group underlies all applications, it is important to patch current. Oracle states that current for Oracle Applications Technology supports the current and previous rollups (RUPn and RUPn-1). In addition to these RUPs, there are many one-offs that should be applied.

The paper “What’s New in Workflow 11i RUP5, RUP6 and R12” contains a discussion of products that are intricately linked with workflow and a list of MetaLink Doc. IDs that will help users keep track of current and critical patches. In addition, the OAUG Workflow SIG, <http://workflowsig.oaug.org/index.html>, contains a list of one-off patches and the MetaLink Doc. IDs that reference these patches.

Remember, if there are issues with a specific workflow, information about that workflow will be found in that product's documentation and MetaLink section.

Another source for workflow information is the book *The ABCs of Workflow for E-Business Suite Release 11i and Release 12*. This book is available from www.solutionbeacon.com and www.amazon.com.

Clean up Errored Workflows

Workflows can error for a variety of reasons. Regardless of the reason, these errors must be corrected and the workflow either continued or aborted. The administration screens in Oracle Administration Manager (OAM) provide an excellent method to identify all errored workflows and the activity that errored. See the paper by Susan Behn "Workflow Troubleshooting" for an explanation of how to use these screens.

The following query will show if there are any errored workflows

```
SELECT COUNT (*)
      ,item_type
      ,activity_name
      ,MIN (item_begin_date)
      ,MAX (item_begin_date)
FROM wf_item_activity_statuses_v
WHERE activity_status_code = 'ERROR'
      AND item_end_date IS NULL
GROUP BY item_type
      ,activity_name
ORDER BY 3 DESC, 1 DESC, 2;
```

Following is a sample result of this query. Look first at the most recent errors and then the workflows with the highest count. This will stop unnecessary increase of workflow data as well as ensure your applications are running smoother. Working on older workflow errors is more complex. Older dates usually indicate that a more recent patch has fixed the error. Now you must decide whether to restart these workflows (and continue the transactions on possibly old, abandoned records) or to abort them.

COUNT(*)	ITEM_TYPE	ACTIVITY_NAME	MIN(ITEM_BEGIN_DATE)	MAX(ITEM_BEGIN_DATE)
13	WFERROR	INITIALIZE_EVENT_ERROR	1/12/2009 5:08:04 PM	2/28/2009 9:49:52 AM
39	PAXWFHRU	NTF_ASSIGNMENT_UPD_FAIL	6/2/2006 7:22:21 PM	2/6/2009 6:28:35 PM
1	PONAUCT	AUCTION_START_EVENT	2/6/2009 3:13:04 PM	2/6/2009 3:13:04 PM
5	REQAPPRV	PO_REQ_APPROVED	7/5/2004 6:06:53 AM	10/1/2008 10:42:02 AM
1	XXHRVEMP	ESCALATE_TO_HR_MANAGER	3/29/2008 5:27:11 PM	3/29/2008 5:27:11 PM
9861	FTEPSUPD	INVOICE_STATUS_UPDATE	3/23/2005 12:46:18 AM	11/14/2007 1:51:54 PM
2	APINVAPR	ESCALATE_DOC_APPROVAL	10/31/2007 3:54:00 PM	11/1/2007 2:36:07 PM

It isn't enough to just clean up the errored workflows. You must also clean up the associated error item types as whenever a workflow errors, it calls another workflow to report the error. Most people are familiar with the "System: Error" (WFERROR) workflow, but this isn't the only workflow that is used to report errors. The following SQL will report on all the workflows used as error workflows. Note that some of them, such as HRSSA, are both the workflow and the error reporting workflow.

```
SELECT item_type
      ,parent_item_type
      ,DECODE (end_date, NULL, 'OPEN', 'CLOSED')
      error_type_status
      ,COUNT (*)
FROM wf_items
WHERE parent_item_type is not null
      AND item_type in ('CUNNLWF','DOSFLOW','DOSFLOWE',
```

```

'ECXERROR', 'HRSSA', 'HRSTAND', 'HXCEMP', 'IBUHPSUB', 'OKLAMERR',
'OMERROR', 'PARMAAP', 'PARMATRX', 'POERROR', 'WFSTD', 'XDPWFSTD',
'ZPBWFERR', 'WFERROR')
GROUP BY item_type
      ,parent_item_type
      ,DECODE (end_date, NULL, 'OPEN', 'CLOSED')
ORDER BY item_type,parent_item_type;

```

If these workflows are not ended, then you won't be able to purge the workflow that originally errored even if the original workflow is now complete. Complicated chains such as OEOH -> OMERROR -> WFERROR or OEOH -> OEOL -> WFERROR can exist.

ITEM_TYPE	PARENT_ITEM_TYPE	ERROR_TYPE_STATUS	COUNT(*)
DOSFLOWE	DOSFLOWE	CLOSED	1
ECXERROR	OKLLPLBO	CLOSED	10
ECXERROR	OZFRESO	CLOSED	1
ECXERROR	SYNCITEM	CLOSED	13
HRSSA	HRSSA	CLOSED	12
OMERROR	OEOH	OPEN	36
OMERROR	OEOL	OPEN	1
PARMATRX	PARMATRX	OPEN	1
POERROR	REQAPPRV	CLOSED	19
POERROR	REQAPPRV	OPEN	3
WFERROR	ADSCUSTM	CLOSED	2
WFERROR	ADSSALES	CLOSED	749
WFERROR	AMSLISTG	CLOSED	10
WFERROR	APEMPVIFY	CLOSED	2
WFERROR	APEXP	CLOSED	7
WFERROR	APEXP	OPEN	5

In addition to errored workflows, there is data left by events that error. Events move through the Advanced Queuing system and are purged regardless of whether they executed correctly or not (purging of these queues will be discussed later). If an event errors, it does call WFERROR to report the error. These WFERRORs are not attached to any other workflow and they do send messages to SYSADMIN. And these messages are usually ignored, leaving thousands of records cluttering the workflow tables. Additionally, the work designed to be done by these events is not done which can cause other issues. Since the events are purged, the following query identifies the erroring events by the WFERROR the events leave behind.

```

SELECT COUNT (*)
      ,v.text_value
      ,min(i.begin_date)
      ,max(i.begin_date)
FROM wf_item_attribute_values v
      ,wf_items i
WHERE v.item_key=i.item_key
      AND v.item_type = i.item_type
      AND v.item_type = 'WFERROR'
      AND v.NAME = 'EVENT_NAME'
      AND v.text_value IS NOT NULL
GROUP BY text_value
ORDER BY text_value;

```

COUNT(*)	TEXT_VALUE	MIN(I.BEGIN_DATE)	MAX(I.BEGIN_DATE)
48	oracle.apps.ap.format.create	2/17/2006 4:41:25 AM	8/4/2006 2:08:25 AM
5	oracle.apps.ap.payment	7/30/2007 10:52:02 PM	11/14/2007 1:54:15 PM
22	oracle.apps.ar.hz.Location.create	4/12/2006 3:29:15 PM	7/25/2006 11:35:57 AM
12	oracle.apps.ar.hz.Location.update	2/7/2006 11:23:28 AM	6/26/2006 10:01:26 AM
6	oracle.apps.ar.hz.OrgContact.create	11/3/2008 1:48:14 PM	2/10/2009 5:17:39 PM
408	oracle.apps.fnd.system.exception	2/19/2007 3:42:42 AM	12/7/2008 9:27:43 PM
58	oracle.apps.fnd.wf.ds.user.updated	2/17/2008 6:07:14 PM	12/8/2008 3:49:08 PM
372	oracle.apps.ita.setup.record	2/20/2006 12:32:25 AM	2/28/2009 9:49:52 AM
1	oracle.apps.wf.notification.cancel	3/29/2008 10:29:42 PM	3/29/2008 10:29:42 PM
6	oracle.apps.wf.notification.send	4/12/2006 11:55:28 AM	3/29/2008 10:29:43 PM
5	sb.apps.xxhr.employee.create	9/4/2007 7:43:50 PM	9/4/2007 8:48:31 PM

As with errored workflows, look for the most recent errors and the events with the highest count. The associated WFERROR will list why the event is erroring. Correct the error. If required, the notification to SYSADMIN also contains a link where the event can be re-raised. After re-raising any events that should still execute, abort the remaining WFERRORs (that are called by these events, not all WFERRORs).

Purge

Outside of eliminating all errors and thus stopping the data growth and performance hits caused by all the WFERROR and other error type workflows, nothing increases performance like eliminating old workflow history.

OAM displays a list of completed workflows and whether they are eligible for purging.

Select Work Item Type ^	Persistence Type	Retention Period	Count	Purgeable
<input checked="" type="radio"/> ADG HR Custom Workflows	PERM	0	2	2
<input type="radio"/> ADS Sales Notification	TEMP	0	1	1
<input type="radio"/> AME update FND user name	TEMP	0	1	1
<input type="radio"/> ASO Approval	TEMP	0	6	0
<input type="radio"/> Address Change	TEMP	0	5	5
<input type="radio"/> CLN Update Delivery Outbound	TEMP	0	1	1
<input type="radio"/> Change Action	PERM	0	9	9
<input type="radio"/> Change Workflow Routing Step	PERM	0	1	0
<input type="radio"/> Dossier Approval Flow Enhanced	TEMP	0	1	0
<input type="radio"/> EPB Business Process	TEMP	0	2	2
<input type="radio"/> Expenses	TEMP	0	7	7
<input type="radio"/> HR	TEMP	0	16	4
<input type="radio"/> IBC - Content Manager Workflow	TEMP	0	7	0
<input type="radio"/> Knowledge Management Workflow Process	SYNC	0	34	34
<input type="radio"/> Knowledge Management Workflow Process (New)	SYNC	0	165	165

Notice that if the Persistence Type is PERM, Purgeable will be always be zero.

For the Persistence Type TEMP, if Purgeable = 0, then either this workflow is a child of a non-completed workflow or this workflow has children that are not in completed status.

The concurrent program "Purge Obsolete Workflow Runtime Data" should be scheduled at least weekly and if still running the 9i database, nightly. Two copies of this program will be required, one with "Persistence Type" = Temporary, and one with "Persistence Type" = Permanent. The other parameters are:

- Item Type – leave this blank
- Item Key – leave this blank
- Age – recommend at least 7, no more than 60
- Core Workflow Only – set to Y
 - At least once/month run set to N
- Commit Frequency – leave at default value of 500 (that's 500 workflows, not records)
- Signed Notifications – this option governs whether electronically signed notifications should be purged. If using this feature, will have to create special schedules to delete these notifications based on legal requirements to keep signatures

The following MetaLink Doc. IDs provide additional information about the purge process. The patches recommended in these notes are already included in Release 11.5.10.2.

- 337923.1 “A closer examination of the Concurrent Program Purge Obsolete Workflow Runtime Data”
- 132254.1 “Speeding Up And Purging Workflows”
- 298550.1 “Troubleshooting Workflow Data Growth Issues”
- 780446.1 “Is It Possible To Run Multiple “Purge Obsolete Workflow Runtime Data” Programs Simultaneously With Different Item Type value.?” (sic)

This program performs the following functions:

- Aborts WFERROR where PARENT_ITEM_TYPE matches Item Type parameter and where linked activity (PARENT_CONTEXT) is no longer in error status
 - Note: Other error item types, like POERRER or OMERROR, are not touched
- Purges Item Types matching Item Type parameter if END_DATE is not NULL and not linked to an open parent or child workflow
- If “Core Workflow Only” = ‘N’
 - Purges WF_ACTIVITIES table where END_DATE is NOT NULL and ACTIVITY_ID is not referenced in any active workflows
 - End-dates, then deletes, notifications not referenced in WF_ITEM_ACTIVITY_STATUSES, WF_ITEM_ACTIVITY_STATUSES_H
 - Example: Notifications from completed concurrent programs
 - Purges ad-hoc roles where ORIG_SYSTEM = WF_LOCAL_ROLES or WF_LOCAL_USERS and not referenced in WF_ROLE_HIERARCHIES or WF_NOTIFICATIONS or WF_ITEMS.OWNER_ROLE

While the additional activities performed when “Core Workflow Only” = N are necessary to ensure that the workflow table data growth is controlled, there is overhead associated with running these extra procedures, especially if running the purge program nightly. And generally the data involved is not significant in volume. This is why Solution Beacon recommends running a special purge with “Core Workflow Only” set to N only once per month.

If your company has not been regularly purging, then you need to do special catch-up purge runs. “Look at the Completed Work Items” page in OAM. For each item type with over 500 workflows eligible for purging (or for those item types where the persistence type = PERM, each item type with over 500 completed workflows), run a purge request specifying that item type (set “Core Workflow Only” = ‘Y’). Note that if the volumes are in the thousands, each run may take a long time, even hours.

And if your company is still running on a 9i database, these purges will actually decrease performance due to the table high water marks. Thus after all the purges are finished, export/import the following tables: WF_ITEMS, WF_ITEM_ATTRIBUTE_VALUES, WF_ITEM_ACTIVITY_STATUSES,

WF_ITEM_ACTIVITY_STATUSES_H, and WF_NOTIFICATIONS. This is not necessary if your database is 10g or higher as these databases will automatically adjust the high water marks.

MetaLink Doc. ID 266614.1 “Troubleshooting Workflow Issues in Applications 11i”, section “Purging Unreferenced Notifications”, contains an alternate script, wfntfprg.sq, for purging unreferenced notifications. Note the instructions required to purge the notifications of finished concurrent requests (FNDCMMSG).

To see if unreferenced notifications are an issue in your environment, use the following SQL:

```
SELECT    COUNT (*)
          ,MESSAGE TYPE
FROM      applsys.wf notifications n
WHERE     NOT EXISTS (
          (SELECT 'X'
           FROM applsys.wf item activity statuses_h s
           WHERE s.notification_id = n.GROUP_ID
           AND EXISTS (
               SELECT 'X'
               FROM applsys.wf items i
               WHERE i.item type = s.item type
                   AND i.item_key = s.item_key))
          UNION
          (SELECT 'X'
           FROM applsys.wf item activity statuses s
           WHERE s.notification_id = n.GROUP_ID
           AND EXISTS (
               SELECT 'X'
               FROM applsys.wf items i
               WHERE i.item type = s.item type
                   AND i.item_key = s.item_key)))
GROUP BY  MESSAGE TYPE
ORDER BY  MESSAGE_TYPE;
```

Configure (Setup) Seeded Workflows

The workflows delivered with each application may require setup before they can be successfully used. These setups may require opening the workflow in the Builder tool and setting performers, values for item attributes, and even customizing procedures. In addition, these workflows may require Profile Option values, hierarchy setups, rules for the Approvals Management Engine (AME), and other setups.

Users should use the product documentation to research the required setups for each workflow. MetaLink white papers are another excellent resource. And the book *The ABCs of Workflow for E-Business Suite Release 11i and Release 12* contains a chapter detailing some common workflow setups and contains scripts to identify timeouts and performers.

Background Engines

Scheduling

The Background Engine has been a required part of workflow since its inception. And most people just run a “generic” copy of the program “Workflow Background Process) (no specified Item Type, no value for Minimum or Maximum Threshold) with “Process Deferred”=Yes, Process Timeout” = Yes and “Process Stuck” = Yes. Resubmission times may vary from 5 minutes to once/day.

This practice can result in unnecessary processing cycles. Oracle and Solution Beacon recommend splitting the background process into at least 3 processes. One process will be dedicated to timed out activities (Process Timeout = Yes). For this process, “Process Deferred” should be No and “Process Stuck” should be No. The resubmission interval for this background engine should be set based on the

minimum timeout defined for the workflows being used. Usually timeouts are set to 1 day or higher, so scheduling this background engine to run once/day is usually sufficient.

The second process will be for "Process Stuck" = Yes (set "Process Deferred" and "Process Timeout" to No). The code for detecting Stuck activities is very processor intensive. And when such activities are found, the background engine just sets the status of the workflow to Error. Thus background engine for stuck activities should be run no more than once/day. And once all workflows that end in the status #STUCK are corrected, the resubmission interval should be increased to once/week or even once/month.

Scheduling the third background engine, "Process Deferred" = Yes ("Process Timeout" and "Process Stuck" = No), can require monitoring the activity in WF_DEFERRED_TABLE_M. This is the queue that stores activities waiting for the background engine to run. Usually it is sufficient to run this engine once every 15-60 minutes. One exception to this rule is companies that run Order Management and have a high volume of order lines or need to ensure that all booked lines are quickly picked. In this case, scheduling a targeted engine for the Order Line workflow to repeat every 5 minutes may be appropriate. Note: per MetaLink Doc. ID 560144.1 "11.5.10.4: Workflow Background Process Seems To Take Longer After Rup4", never set the resubmission time for this process to be less than 5 minutes.

Background Engine Runs for a Long Time

Once the background engine has been separated into three processes, monitor the execution time of the engine that processes deferred activities. If the engine runs more than a few seconds, further research is warranted. As stated, the background engine works against the queue table WF_DEFERRED_TABLE_M. The engine looks at all records where STATE=0. It processes these records in PRIORITY, ENQ_TIME order. Once a record is selected, the engine starts the next activity in the workflow and waits until either that workflow ends or that workflow reaches another deferred activity. Then the engine processes the next record. How long it takes to process each record can be determined by DEQ_TIME – ENQ_TIME. Remember, the records are purged from this queue 24 hours after DEQ_TIME. The problem is usually not the background engine; it's the code in the workflow activity. See MetaLink Doc. ID: 186361.1 "WF 2.x: Workflow Background Process Performance Troubleshooting Guide".

This same note also points out that bad performance can also be traced to workflows that are seemingly in a loop, i.e. the same workflow continuously appears in WF_DEFERRED_TABLE_M at the same node. You can identify these workflows using the diagnostic "Workflow Status and Purgeable Items", section "Large Activity History". The Order Header (OEOH) workflow usually shows up on this list. This workflow goes to deferred status at a Wait node. When it starts again, it checks to see if all the lines are closed. If they are, the header is closed and the workflow ends. If not, the workflow returns to the Wait node. For this particular workflow, companies need to configure the wait time to the average time it takes to process all lines on an order. Then look at the specific workflows that show on the diagnostic report and investigate whether the lines for this order are really still open. Chances are the workflow for one of the lines errored and never completed successfully and that line has been abandoned. See the section later in this paper on OEOH/OEOL workflow for some scripts that will help with this issue.

Performance Patches

There are two patches that can impact performance. One is the 10g Release 2 Interoperability patch for 11.5.10 (4653225). See MetaLink Doc. ID 469702.1 "Workflow Background Process Takes Long Time to Run After Conversion To ATG Rup5 DB 10g". The other patch applies to companies running database version 10.2.0.4 and RAC. This patch is 6600051 and is described in MetaLink Doc. ID "How to Resolve the Most Common Workflow Background Engine Problems".

INIT.ORA Parameters

There has been some confusion over the parameter AQ_TM_PROCESSES. In order for the background processor to run, this parameter must have a value of at least 1. However, in 10g+ databases, Queue

Monitoring can “auto-tune” this parameter if not set in the database parameter file. But since the OAM screens and diagnostics use this file to print the value of the parameter, these queries will show a value of zero. Auto-tuning will yield better performance, just make sure that the DBAs and the workflow administrators are all aware of the false reading of zero. Note: if you decide to continue to use the init.ora file to set a value for this parameter, never set it >9 as this disables some Queue Monitoring features. See MetaLink Doc. IDs: 560144.1 “11.5.10.5: Workflow Background Process Seems To Take Longer After Rup4” and 746313.1 “What should be the Correct Setting for Parameter AQ_TM_PROCESSES in E-Business Suite Instances”.

The first note also states that JOB_QUEUE_PROCESSES should be at least 5. This is the 11.5.9 setting. For Release 11.5.10 and Release 12, this parameter should be set to 10. As Oracle seeds the value to be 2, change the init.ora file as soon as possible.

For databases prior to 10g, JOB_QUEUE_INTERVAL should be 60

Advanced Queuing Performance

Workflow is heavily dependent on Advanced Queuing. MetaLink Doc. ID 560144.1 “11.5.10.5: Workflow Background Process Seems To Take Longer After Rup4” states that the performance of the queues is increased when regular rebuilds/coalesces are performed on all the Indexes/IOTS. How to do this is described in MetaLink Doc. ID 271855.1 “Procedure to manually Coalesce all the IOTs/indexes Associated with Advanced Queuing tables to maintain Enqueue/Dequeue performance, reduce QMON CPU usage and Redo generation”. Other MetaLink Doc. IDs that impact pre-10g performance are 284692.1 “How to Reduce the Highwater Mark for Advanced Queuing objects using truncate in pre-10g databases”, and 285692.1 “How To Rebuild Queue Tables via Export/Import for pre-10g Databases”.

Another good note for ensuring the queues are working properly is MetaLink Doc. ID 469009.1 “Troubleshooting Workflow Agent Listener’s failure to start”.

Performance of the queues can suffer if they are set for statement logging or debug logging. To ensure that debug logging is turned off, set the following profile options:

- FND: Debug Log Enabled – No
- FND: Debug Log Level – Unexpected
- FDN: Debug Log Module – NULL

Check the setup for each Listener and ensure that the log level is set to Error.

If changes are made to the profile options and/or the log level, stop and restart the Workflow Agent Listener Container. See MetaLink Doc. ID 741087.1 “High Logging Messages on WF_EVENT_OJMSTEXT_QH procedure”.

If performance is suffering either because the containers do not have enough memory or because they have been allocated too much memory, see MetaLink Doc. IDs 444938.1 and 444939.1 “How do you Change the Maximum Memory Size taken by Workflow Service Container”.

Events are retained in a queue once processed only until the retention time for the queue. When a listener wakes up, one of the tasks is to delete expired events from the queue. It is true that setting retention to 0 will increase the queue performance. However, it destroys the ability to tune the queues or perform any troubleshooting. This is one case where the hit to performance should be taken. The recommended value is 86400 seconds (1 day). Customers should increase the retention from 0 to 86400 for the WF_ERROR and WF_JAVA_ERROR queues and decrease the retention for the queues WF_IN, WF_OUT, WF_REPLAY_IN, and WF_REPLAY_OUT. This is one parameter that cannot be set from OAM. Instead use the following:

```
Dbms_aqadm.alter_queue (queue_name=>'<queue>', retention_time=>86400);
```

Ensure the Workflow Agent Listener Container is down first.

The following listeners should always be running: Workflow Deferred Agent listener, Workflow Deferred Notification Agent Listener, Workflow Error Agent listener, Workflow Java Deferred Agent Listener,

Workflow Java Error Agent Listener. If allowing email responses to notifications, then the listener Workflow Inbound Notifications Agent Listener. If using XML Gateway, the listeners ECX Inbound Agent Listener and ECX Transaction Agent Listener must be running. The listeners Workflow Inbound JMS Agent Listener and Web Services IN Agent are required when using web services and/or receiving Java messages from external systems. Use the Agent Activity page in the OAM Workflow Manager to monitor whether events are flowing through these queues. If the Ready column >0, ensure the associated listener is running. Note: any Agent with _OUT in the name is controlled through scheduling propagation, not starting a listener.

WF_CONTROL

WF_CONTROL is the queue that manages all the other queues. Oracle recommends running the "Control Queue Cleanup" concurrent program every 12 hours. This is such an important recommendation that whether this program is running or not is part of the OAM Workflow Manager dashboard. MetaLink Doc. ID: 469045.1 "Troubleshooting WF_CONTROL Agent Issues" discusses what this queue does and provides additional scripts to ensure the queue is performing optimally.

WF_DEFERRED

WF_DEFERRED is the queue where all events that have subscriptions that start workflows and that have subscriptions with Phase>100 start. Performance of this queue is managed very similarly to that of WF_DEFERRED_TABLE_M. MetaLink Doc. IDs 334348.1 "Low Performance Processing Messages in WF_DEFERRED Queue" and 468650.1 "Troubleshooting WF_DEFERRED Agent Listeners Performance" provide SQL to determine the Events in the queue. First identify if the events are not dequeued in a timely manner. Events where STATE=0 that are not picked up within 2X of the queue's sleep time indicate that there may more volume running through the queue than a single listener can handle. Suggestions are to:

- Create additional generic agent listeners
- Create specific agent listeners for events with high volume
- Increase 'Inbound Thread Count' (PROCESSOR_IN_THREAD_COUNT) by 1 until the performance is acceptable

The other issue is that it takes a long time to process the event. This is usually the fault of the code the event is executing. Trace the code and identify the issues.

If you discovered that the Agent Listeners were down and not running for awhile, use the Dbms_aqadm.alter_queue statement above to temporarily set the retention_time to zero. After the volume in the queue subsides, reset it back to 86400.

As a last resort, consider that the queue may be corrupt, especially if you are receiving the error "ORA-24033: No Recipients for Message". See MetaLink Doc. ID 286394.1 "How to rebuild the WF_DEFERRED queue".

Diagnostics

Oracle provides Diagnostic scripts that can identify issues before they become big problems. These tests are run from the Diagnostic responsibilities. See the paper "Maximize Efficiency of Oracle E-Business Suite Release 11i/12 through Diagnostics" by Susan Behn for details how to setup and run these tests.

The following two diagnostics provide quite a bit of information that will aide in increasing workflow performance. Unfortunately they are currently (as of 12.0.6) only available for Release 11i.

Workflow Status and Purgeable Items

This test reports on the following sections:

Oracle Diagnostics Information	Product Installation Status and Patchset Level	Parameters
Environment Variables	Patchset Levels	Execution of Seeded SQL Scripts
Tablespace Size	Workflow Table Sizes	Open and Closed Items Annually
Large Activity History	Workflow GSM Component Queue Status	Concurrent Program FNDWFPRG Schedule
Workflow Background Engine Status	Concurrent Program FNDWFBKG Schedule	Workflow Deferred Events Status
Workflow Errors Status	Recommendations	References

The sections that deal with performance are:

- Large Activity History
- Open and Closed Items Annually
- Workflow Background Engine Status
- Concurrent Program FNDWFBKG Schedule
- Recommendations, References

This diagnostic provides a [See SQL](#) link for each section so that you can see (and use/modify) the code that produced that section of the report.

The “Large Activity History” section lists workflows where a single activity has been executed more than 300 times in a single workflow instance. The reasons that would cause OEOH to appear have already been discussed. Each workflow listed should be further investigated.

Large Activity History [\[Top\]](#)

Item Type	Item Key	Count	Root Activity	Version	Begin Date	End Date	Description
CNCOMPPR	11590--03-aug-2001 11:34:31	4394	CN_COMP_PLAN_PROCESS	4	03-AUG-2001	19-DEC-2007	11590
OEOH	45185	1720	R_STANDARD_HEADER	6	02-APR-2001		Sales Order 777788
OEOH	45187	1720	R_STANDARD_HEADER	6	02-APR-2001		Sales Order 777789
OEOH	45188	1720	R_STANDARD_HEADER	6	02-APR-2001		Sales Order 777790
OEOH	45189	1720	R_STANDARD_HEADER	6	02-APR-2001		Sales Order 777791
OEOH	45190	1720	R_STANDARD_HEADER	6	02-APR-2001		Sales Order 777792
OEOH	45352	1720	R_STANDARD_HEADER	8	21-MAY-2001		Sales Order 777795
OEOH	45354	1720	R_STANDARD_HEADER	8	21-MAY-2001		Sales Order 777797
OEOH	45355	1720	R_STANDARD_HEADER	8	21-MAY-2001		Sales Order 777798
SERVEREQ	18860-1371	960	CUG_GENERIC_WORKFLOW	2	22-AUG-2003		

10 rows retrieved

- [See SQL](#)

The “Open and Closed Items Annually” section indicates purge issues and abandoned workflows.

Open and Closed Items Annually [\[Top\]](#)

- This output indicates old data open or closed still in the tables
This data is runtime, and once completed (closed) is obsolete and should be purged for system performance

Open and Closed Items Annually					
Opened Year	Item Type	Display Name	Persistence Type	End Date Year	Count
2000	OEOH	OM Order Header	TEMP	2003	1
2000	OEOL	OM Order Line	TEMP	2003	3
2000	WFERROR	System: Error	TEMP	2003	5
2000	APEXP	Expenses	TEMP	OPEN	1
2000	GLALLOC	GL AutoAllocation	TEMP	OPEN	1
2000	HRSSA	HR	TEMP	OPEN	113

In the picture above, the closed OEOH and OEOL workflows indicate there are open children somewhere. The open APEXP from 2000 indicates that the expense report has been abandoned. Cleaning up this old history will improve performance.

The "Workflow Background Engine Status" section shows what workflows are in the WF_DEFERRED_TABLE_M table and the status of the records.

Workflow Background Engine Status [\[Top\]](#)

Workflow Background Engine Status		
Corrid	State	Count
APPSOEOH	0 = Ready	215
APPSOEOH	1 = Delayed	8
APPSPACRMUPD	0 = Ready	5

3 rows retrieved

- [See SQL](#)

If the State is 1, then the time to process this activity is still > SYSDATE. For the OEOH workflow, this is the Wait node where the wait time has not expired yet.

The "Workflow Status and Purgeable Items" section shows the execution times and parameters for the most current executions of "Workflow Background Process". Even with a small number of workflows, you can see that running an engine with "Process Stuck" set to Yes increases runtime significantly.

Concurrent Program FNDWFBKG Schedule [\[Top\]](#)

Concurrent Program FNDWFBKG Schedule								
Request Id	Program	Status	Phase	Actual Start Date	Actual Completion Date	Total Minutes	Time to Run	Arguments
6980690	Workflow Background Process	C=Normal	Completed	2006-05-08 04:37:09.0	2006-05-08 04:37:10.0	.02	0:hrs 0:Mins 1:Secs	... Y, N, N
6980689	Workflow Background Process	C=Normal	Completed	2006-05-08 04:32:09.0	2006-05-08 04:32:09.0	0	0:hrs 0:Mins 0:Secs	... Y, N, N
6980687	Workflow Background Process	C=Normal	Completed	2006-05-08 04:27:08.0	2006-05-08 04:27:09.0	.02	0:hrs 0:Mins 1:Secs	... Y, N, N
6980686	Workflow Background Process	C=Normal	Completed	2006-05-08 04:22:08.0	2006-05-08 04:22:08.0	0	0:hrs 0:Mins 0:Secs	... Y, N, N
6980684	Workflow Background Process	C=Normal	Completed	2006-05-08 04:17:07.0	2006-05-08 04:17:08.0	.02	0:hrs 0:Mins 1:Secs	... Y, N, N
6980683	Workflow Background Process	C=Normal	Completed	2006-05-08 04:12:03.0	2006-05-08 04:12:04.0	.02	0:hrs 0:Mins 1:Secs	... Y, N, N
6980681	Workflow Background Process	C=Normal	Completed	2006-05-08 04:07:02.0	2006-05-08 04:07:02.0	0	0:hrs 0:Mins 0:Secs	... Y, N, N
6980680	Workflow Background Process	C=Normal	Completed	2006-05-08 04:02:01.0	2006-05-08 04:02:01.0	0	0:hrs 0:Mins 0:Secs	... Y, N, N
6980678	Workflow Background Process	C=Normal	Completed	2006-05-08 03:57:00.0	2006-05-08 03:57:01.0	.02	0:hrs 0:Mins 1:Secs	... Y, N, N
6980677	Workflow Background Process	C=Normal	Completed	2006-05-08 03:51:57.0	2006-05-08 03:51:57.0	0	0:hrs 0:Mins 0:Secs	... Y, N, N
6765866	Workflow Background Process	C=Normal	Completed	2006-04-20 21:00:20.0	2006-04-20 21:00:39.0	.32	0:hrs 0:Mins 19:Secs	... Y, Y, Y

Note the runtimes where the arguments are Y,N,N. If these runtimes are stated in minutes rather than seconds then consider creating targeted engines for the item types that occurred the most frequently. Also evaluate the frequency of the submission of the background engine. In our example the engine repeats every 5 minutes. The execution times are 0 seconds and 1 second. This indicates the engine is running too frequently. Cut the resubmission time in half and monitor again.

The last sections of this diagnostic are "Recommendations" and "References". For this execution, the recommendations are to ensure the purge is running. Because this is the recommendation, the references list MetaLink Doc. IDs that will provide additional information about purging.

Recommendations [\[Top\]](#)

ATTENTION - To improve performance on the instances where Workflow is used intensively please purge runtime data either by running 'Purge APIs' or by running 'Purge Obsolete Workflow Runtime Data' concurrent request. For best performance, run WF_PURGE.TOTAL frequently, and WF_PURGE.TOTALPERM periodically, but be aware of site's policy on status history retention.

References [\[Top\]](#)

- Note [144806.1](#) Purging Workflow Runtime Tables
- Note [132254.1](#) Speeding up and Purging Workflow
- Note [139745.1](#) A System Administrators' Guide: Oracle Applications 11i: Workflow
- Note [277124.1](#) FAQ on Purging Oracle Workflow Data
- Note [141853.1](#) Workflow Performance is Very Slow And Workflow Background Tables Are Large

Workflow Performance

This test reports on the following sections:

Oracle Diagnostics Information	Parameters	Product Installation Status and Patchset Level
Key Profile Options	Workflow Item Types	Activity Statuses
Historical Activity Statuses	Notifications	Notification Mails ✖
Item Attribute Values	Miscellaneous	Recommendations
References		

All sections for this diagnostic deal with performance. This paper will focus on:

- Key Profile Options
- Activity Statuses in conjunction with Historical Activity Statuses, Item Attribute Values, Notifications
- Miscellaneous
- Recommendations, References

This diagnostic do not provide a [See SQL](#) link for each section.

The “Key Profile Options” show the values at all levels of profile options that can significantly impact performance. The picture below lists only one of them.

Key Profile Options [\[Top\]](#)

Profile Options

Account Generator:Run in Debug Mode							
Long Name	Short Name	Level	Level Value	Profile Value	Profile Id	Updated Date	Updated By
Account Generator:Run in Debug Mode	ACCOUNT_GENERATOR:DEBUG_MODE	Site	Site	N	3101	10-OCT-2000 16:45	JPALMER
Account Generator:Run in Debug Mode	ACCOUNT_GENERATOR:DEBUG_MODE	Responsibility	iProcurement, Vision Services (USA)	N	3101	08-JUL-2003 20:51	SYSADMIN

2 rows retrieved

“Account Generator:Run in Debug Mode” is used to troubleshoot the account generators. If set to yes, all the account generators will log activity to the runtime tables. While this is a tremendous aide to troubleshooting, it is a tremendous drag on performance. This profile option should normally be set to No at all levels.

“PO:Workflow Processing Mode” governs whether approving the PO should be deferred to a background process or done immediately. If the buyer doesn’t have enough authority to self-approve, the workflow will go to deferred status anyway when the notification requesting approval is created. This will just cause delays in the screen for the user with no benefit. Therefore for each buyer that has sufficient authority to approve all the POs they create, set this value to Yes. Otherwise set this to No.

“HR:Defer Update After Approval” governs whether approvals are applied immediately or deferred to the next background engine. The behavior of this profile option is described in MetaLink Doc. ID 317002.1 “Approval Is Delayed In Defer Thread Activity”. If this profile option is set to Yes, the HR records will not reflect approval until the next background engine is run. If the background engine is already processing a lot of records, the addition to the load has to be considered and a targeted engine may be required.

If your business process requires that the approval be recorded immediately, then set the value to No. However, this can cause the FYI notifications to be automatically closed and to contain no information.

MetaLink Doc. ID 472387.1 "FYI Notifications Closed Automatically and Not Showing Up on Worklist" details the adjustment that must be made to avoid this.

The "Closed Item Types" section lists the number of closed items by item type, the minimum and maximum time each workflow was open, and the average life of that item type.

Closed Item Types								
Item Type	Persistence Type	Persistence Days	Count	Average Life	Deviation Life	Min Life	Max Life	Statement Suggestion
XDPWFSTD	PERM		5374	482.04	22.48	63.95	1102.79	exec WF_PURGE.ITEMS ('XDPWFSTD',NULL,SYSDATE,FALSE);
XDPPROV	PERM		7	149.75	20.86	119.22	161.96	exec WF_PURGE.ITEMS ('XDPPROV',NULL,SYSDATE,FALSE);
ENGCACT	PERM		2	656.05	0	656.05	656.05	exec WF_PURGE.ITEMS ('ENGCACT',NULL,SYSDATE,FALSE);
GMSWF	PERM		2	218.88	12	210.39	227.36	exec WF_PURGE.ITEMS ('GMSWF',NULL,SYSDATE,FALSE);
ADSCUSTM	PERM		1	0	0	0	0	exec WF_PURGE.ITEMS ('ADSCUSTM',NULL,SYSDATE,FALSE);
ENGCSSTEP	PERM		1	11.95	0	11.95	11.95	exec WF_PURGE.ITEMS ('ENGCSSTEP',NULL,SYSDATE,FALSE);
OKLAMPPT	PERM		1	.01	0	.01	.01	exec WF_PURGE.ITEMS ('OKLAMPPT',NULL,SYSDATE,FALSE);
CS_KB_W1	SYNC	0	40	0	0	0	0	exec WF_PURGE.ITEMS ('CS_KB_W1',NULL,SYSDATE,FALSE);
CS_KB_WF	SYNC	0	34	490.78	8.38	447.81	496.07	exec WF_PURGE.ITEMS ('CS_KB_WF',NULL,SYSDATE,FALSE);

Workflows with a persistence type of PERM won't show Persistence Days. Other types will show the required number of days the workflow must remain in history before becoming eligible to be purged.

Look at the average life for each item type and the deviation. Large Deviations show that a workflow was probably abandoned and then later finally aborted. Small deviations and large lives indicate business processes that may need tuning. Large counts indicate that either the purge isn't running or that the item type has unclosed parents/children.

The "Opened Item Types" section shows all open workflows and the average number of days each item type remains open.

Opened Item Types								
Item Type	Persistence Type	Persistence Days	Count	Average Life	Deviation Life	Min Life	Max Life	
XDPPROV	PERM		39	2386.53	143.92	2225.02	2555.07	
OKLLPLBO	PERM		14	1788.89	116.76	1701.55	1966.93	
HRBISWF	PERM		9	1697.8	28.16	1622.74	1707.67	
STR_IBD2	PERM		6	2703.73	71.83	2557.17	2735.2	
ADSSALAP	PERM		4	2290.39	95.98	2241.08	2434.33	
GMSWF	PERM		3	2610.18	.13	2610.1	2610.33	
ENGCACT	PERM		2	1700.64	.78	1700.09	1701.19	
XDPWFSTD	PERM		2	155.26	58.88	113.63	196.9	
ENGCR	PERM		1	1700.08	0	1700.08	1700.08	
ENGCSSTEP	PERM		1	1688.13	0	1688.13	1688.13	
GHR_SF52	PERM		1	2441.38	0	2441.38	2441.38	
IBEALERT	PERM		1	1688.24	0	1688.24	1688.24	
OKLAMERR	PERM		1	1702.2	0	1702.2	1702.2	
OKLCSCRM	PERM		1	1701.24	0	1701.24	1701.24	
SERVEREQ	PERM		1	1685.31	0	1685.31	1685.31	
CS_KB_WF	SYNC	0	114	2013.53	97	1881.1	2234.31	
HRSSA	TEMP	0	1138	2117.71	345.64	239.37	2995.35	
OEOL	TEMP	0	339	1784.45	369.51	237.92	2843.17	
OEOL	TEMP	0	336	1683.4	356.83	56.87	2842.44	

Workflows with a persistence type of PERM won't show Persistence Days. Other types will show the required number of days the workflow must remain in history before becoming eligible to be purged.

If the Deviation Life is large, there are probably open workflows that are either in error status or that have been abandoned. If the Deviation Life is small and the Average Life is large (assuming there is more than 1 workflow), then the underlying business process should be examined. A well tuned environment should have average lives that reflect your business process cycles and deviations close to zero.

The "Activity Statuses - Closed" section shows for closed workflows, the number of records in WF_ITEM_ACTIVITY_STATUSES and WF_ITEM_ACTIVITY_STATUSES_H.

Activity Statuses [Top]		
Closed		
Item Type	Count	Statement Suggestion
OEOL	21415	exec WF_PURGE.ITEMS('OEOL',NULL,SYSDATE,FALSE);
XDPWFSTD	16166	exec WF_PURGE.ITEMS('XDPWFSTD',NULL,SYSDATE,FALSE);
WFERROR	1888	exec WF_PURGE.ITEMS('WFERROR',NULL,SYSDATE,FALSE);
OAM_BE	1055	exec WF_PURGE.ITEMS('OAM_BE',NULL,SYSDATE,FALSE);
REQAPPRV	660	exec WF_PURGE.ITEMS('REQAPPRV',NULL,SYSDATE,FALSE);
CNCOMPPR	270	exec WF_PURGE.ITEMS('CNCOMPPR',NULL,SYSDATE,FALSE);
CS_KB_WF	244	exec WF_PURGE.ITEMS('CS_KB_WF',NULL,SYSDATE,FALSE);

This section does not count the records that are in WF_NOTIFICATIONS or WF_ITEM_ATTRIBUTE_VALUES. So each purge removes many more records than the counts shown here.

The "Activity Statuses – Open" section shows for open workflows, the number of records in WF_ITEM_ACTIVITY_STATUSES and WF_ITEM_ACTIVITY_STATUSES_H.

Opened	
Item Type	Count
HRSSA	5638
OEOL	4436
OEOH	3019
REQAPPRV	1448
POWFRQAG	1430

This table should be correlated with the Open Items section. Divide this number by the number of open workflows and you can see the average number of records added to the activity tables whenever a workflow of this type is running.

There are other sections that list record counts for WF_NOTIFICATIONS and WF_ITEM_ATTRIBUTES, but doesn't break them out by whether the workflow is open or closed.

The "Miscellaneous – Statistic Statements" list recommended method of gathering stats for the workflow tables.

Miscellaneous [\[Top\]](#)

Statistic Statements

Statement Suggestion

```
EXEC DBMS_STATS.DELETE_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEM_ACTIVITY_STATUSES');
EXEC FND_STATS.GATHER_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEM_ACTIVITY_STATUSES' ,
percent=>10 , granularity=>'PARTITION' );
EXEC DBMS_STATS.DELETE_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEM_ACTIVITY_STATUSES_H');
EXEC FND_STATS.GATHER_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEM_ACTIVITY_STATUSES_H' ,
percent=>10 , granularity=>'PARTITION' );
EXEC DBMS_STATS.DELETE_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEM_ATTRIBUTE_VALUES');
EXEC FND_STATS.GATHER_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEM_ATTRIBUTE_VALUES' ,
percent=>10 , granularity=>'DEFAULT' );
EXEC DBMS_STATS.DELETE_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEMS');
EXEC FND_STATS.GATHER_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_ITEMS' , percent=>10 ,
granularity=>'DEFAULT' );
EXEC DBMS_STATS.DELETE_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_NOTIFICATIONS');
EXEC FND_STATS.GATHER_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_NOTIFICATIONS' , percent=>10 ,
granularity=>'DEFAULT' );
EXEC DBMS_STATS.DELETE_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_NOTIFICATION_ATTRIBUTES');
EXEC FND_STATS.GATHER_TABLE_STATS(ownname=>'APPLSYS' , tabname=>'WF_NOTIFICATION_ATTRIBUTES' ,
percent=>10 , granularity=>'DEFAULT' );
```

The “Miscellaneous – Histogram Statements” section shows how to create appropriate histograms for the workflow tables.

Histogram Statements

- Check/create a histogram for ACTIVITY_STATUS in WF_ITEM_ACTIVITY_STATUSES table

Statement Suggestion

```
begin
FND_STATS.LOAD_HISTOGRAM_COLS(
action=>'INSERT'
,appl_id=>0
,tabname=>'WF_ITEM_ACTIVITY_STATUSES'
,colname=>'ACTIVITY_STATUS'
);
FND_STATS.GATHER_TABLE_STATS(
ownname=>'APPLSYS'
,tabname=>'WF_ITEM_ACTIVITY_STATUSES'
);
end;
```

- Check/create a histogram for MAIL_STATUS and STATUS in WF_NOTIFICATIONS table

Statement Suggestion

```
begin
FND_STATS.LOAD_HISTOGRAM_COLS(
action=>'INSERT'
,appl_id=>0
,tabname=>'WF_NOTIFICATIONS'
,colname=>'MAIL_STATUS'
);
FND_STATS.LOAD_HISTOGRAM_COLS(
action=>'INSERT'
,appl_id=>0
,tabname=>'WF_NOTIFICATIONS'
,colname=>'STATUS'
);
FND_STATS.GATHER_TABLE_STATS(
ownname=>'APPLSYS'
,tabname=>'WF_NOTIFICATIONS'
);
end;
```

The "Recommendations" section lists recommendations to improve workflow performance and lists MetaLink Doc. IDs that can assist in improving workflow performance.

Recommendations [\[Top\]](#)

ATTENTION - To improve performance on the instances where Workflow is used intensively please purge runtime data either by running 'Purge APIs' or by running 'Purge Obsolete Workflow Runtime Data' concurrent request. The "Statement Suggestion" columns, available in some of the sections of this test, can be used to purge Workflow data. For best performance, run WF_PURGE.TOTAL frequently, and WF_PURGE.TOTALPERM periodically, but be aware of site's policy on status history retention.

References [\[Top\]](#)

- Note [144806.1](#) A Detailed Approach To Purging Oracle Workflow Runtime Data
- Note [132254.1](#) Speeding up and Purging Workflow 2.5 and 2.6
- Note [139745.1](#) A System Administrators' Guide: Oracle Applications 11i Workflow
- Note [186361.1](#) WF 2.x: Workflow Background Process Performance Troubleshooting Guide
- Note [277124.1](#) FAQ on Purging Oracle Workflow Data
- Note [187735.1](#) Workflow FAQ - All Versions
- Note [141853.1](#) Workflow Performance is Very Slow And Workflow Background Tables Are Large

Notification Mailer

This part of the workflow technology seems to generate the most MetaLink Doc. IDs. The mailer isn't required to use workflow; notifications can be viewed/answered through the Worklist screens. Mailers can be set up to send only (not allow responses). Mailers can be set up to process all item types or specific item types or even specific messages for a specific item type. One thing that must be remembered is that since ATG_PF.H.delta.4 (RUP 4), Alert uses the Mailer, so if your company uses Alert, you must setup and start the mailer. See MetaLink Doc. ID 739491.1 "How To Configure Workflow Mailer ONLY For Oracle Alerts in R 11i And/Or R 12" for instructions on setting up an Alert only mailer. One note, set the Correlation id to "ALR:%" once ATG_PF.H.delta.5 (RUP 5) or higher is applied or using Release 12.

Global Preference is "Do not send me mail" (QUERY)

If your company does not want to use the mailer, make sure that users cannot override the global preference. This can be done by click on the Preferences link, then using Framework Personalizations to prohibit users from changing the value in this field. Of course, access to the Preferences link can be blocked by setting the profile option "General Preferences Show Flag" to No.

Since users may have already set preferences to another value, monitor the table FND_USER_PREFERENCES WHERE MODULE_NAME = 'WF' and PREFERENCE_NAME = 'MAILTYPE'. PREFERENCE_VALUE should be 'QUERY' for all records. Unfortunately administrators cannot override this setting through any screens. So either contact the user and ask them to change the value or use SQL.

Because workflow adds records to the WF_NOTIFICATION_OUT queue even if all preferences are set to QUERY, Oracle recommends disabling the Local subscription to the event oracle.apps.wf.notification.send.group. See MetaLink Doc. ID 453137.1 "Oracle Workflow Best Practices Release 12 and Release 11i", section "Choosing Not to Use E-mail Notifications".

Continue to monitor WF_NOTIFICATION_IN and WF_NOTIFICATION_OUT for records and WF_DEFERRED for oracle.apps.wf.notification.% events. If any records are found, query FND_USER_PREFERENCES and find out whose preference was reset and how.

Responses Through Email Are Not Allowed

If the mailer is setup to process outbound but not inbound notifications, set the startup mode for the Workflow Inbound Notifications Agent Listener to Manual. Monitor the WF_NOTIFICATION_IN queue and if records are found, investigate why.

If using Alert and Alert emails require a response, setup a mailer just for Alert that allows inbound responses.

Increase Performance by Reducing Polling Interval

MetaLink Doc. ID: 315748.1 "How To Change The Java Workflow Mailer Inbound Polling Interval" explains how to reduce the number of times the mailer polls for incoming mail (this of course assumes that notification responses are allowed). The recommendation is to increase the "Processor Min Loop Sleep" (PROCESSOR_LOOP_SLEEP) parameter (measured in seconds). If you change this value, ensure that the "Processor Max Loop Sleep" (PROCESSOR_MAX_LOOP_SLEEP) parameter is set to 5X this value.

Despite the recommendation in this note, do NOT unclick "Process Close on Read Timeout". This can cause the mailer to stop processing. This is documented in MetaLink Doc. IDs: 422870.1 "Java Mailer not Removing Processed Emails from Inbox after Folder ATG.H Rup4", 332152.1 "OWF.H Diagnostics, Solutions and Information", 437086.1 "E-Mail Notifications Are Not Getting Processed, Remain Sitting In The Inbox", 741352.1 "Unable to Connect to Mail Store in Workflow Notification Mailer Logs", 601423.1 "Emails Not Received or Delayed Log file contains: javax.mail.MessagingException: 451 Timeout waiting for client input", and 418077.1 "Workflow Notification Mailer Stops Processing".

Avoid java.lang.OutOfMemoryError

MetaLink Doc. ID: 467516.1 "Users suddently (sic) Stop Receiving Email Notifications" states that if the mailer is processing a lot of notifications, such as yearend or open enrollment, it may run out of memory. The situation will be detected by seeing the following in the mailer log file:

```
Maximum number of errors (1000) have been reached for this Service Component ->
oracle.apps.fnd.cp.gsc.ProcessorException: An unexpected RuntimeException or Throwable
occurred -> java.lang.OutOfMemoryError
```

```
This Automatic Service Component has been restarted the maximum of 10 times after stopping
with error. Thus, it has been system deactivated -> oracle.apps.fnd.cp.gsc.ProcessorException:
An unexpected RuntimeException or Throwable occurred -> java.lang.OutOfMemoryError
```

The solution is to edit the file \$APPL_TOP/admin/advvars.env and add/change the following variable.

```
APPSJREOPT="-Xms128m -Xmx3072m"
export APPSJREOPT
```

After making the change, bounce the concurrent managers. While this is an 11.5.9 note, 11.5.9 also used the same Java Mailer, so the note still applies.

Ensure the Mailer Session Does Not Expire

If users are receiving emails with a login prompt and/or "You have insufficient privileges for the current operation" and the problem is intermittent or disappears after you restart the mailer, the ICX Limit Connect profile option is set too low. This is the same profile option that timeouts self-service pages and forces you to re-open the page or re-login. For users and the forms, a low setting is good practice. For the mailer, it causes lots of issues. The answer is to set high values for the user assigned as the FRAMEWORK_USER. By default, this is the user SYSADMIN. Since no one should be regularly logged in as this user, setting high values at the user level for this user will not allow forms to stay open too long, but will eliminate the mailer issues. Set the following profile options to the following values:

- ICX:Session Timeout 12000

- ICX: Limit time 192
- ICX: Limit connect 1000000

In addition to these settings, ensure that the mailer is re-started weekly. This can be done manually or by scheduling events to do this. See the paper by Susan Behn “Workflow Troubleshooting” for an explanation of how to setup these events and the two patches required to make it work. See also MetaLink Doc. ID: 414376.1 “You Have Insufficient Privileges For The Current Operation” On Reqapprv Notif”.

In addition to the profile options, this note recommends setting the “Framework URL timeout” parameter to 120.

Tag Files are important also. These are used to recognize beginning words in the subject of an email and allow the mailer to act appropriately. For example, if someone turns on their “Out of Office” Assistant and the mailer parameter “Send warning for unsolicited e-mail” is checked, then when an email is sent to this user, the Out-of-Office response is sent, which triggers the mailer to send an Unsolicited Mail notification which can trigger another Out-of-Office response, etc. So include a tag that tells the mailer to ignore Out Of Office responses. This issue is detailed in MetaLink Doc. ID 388709.1 “Email Notification Failures Are Causing The Email Servers To Crash”, only the issue listed in this note is the email sent by outside mailer servers when they receive an email to an invalid address. So while adding tags, add one for “Undeliverable:” or whatever words identify this type of response.

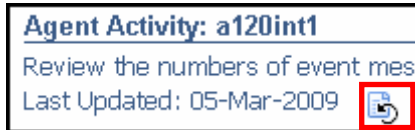
An alternative method is to unclick the parameter “Send warning for unsolicited e-mail”. See MetaLink Doc. ID 431359.1 “Setting up a Tag in the Mailer configuration files to handle unsolicited mail”.

Another parameter that affects performance is “Send e-mails for canceled notifications”. If this is checked, when a notification times out or a group email is responded to, the mailer sends a notification informing you that the previous notification has been canceled. Since usually this mailer behavior just annoys the users, unclicking this parameter and restarting the mailer is a quick performance win.

Workflow Statistics Concurrent Programs

In early versions of 11.5.10, the pages in OAM would time out if there were large volumes of workflows. Oracle introduced three programs to denormalize these queries “Workflow Mailer Statistics Concurrent Program”, “Workflow Work Items Statistics Concurrent Program”, and “Workflow Agent Activity Statistics Concurrent Program”. Although these programs are helpful, they should not be scheduled more than

once/day. Since Administrators can click the refresh icon to get the latest counts, the schedule can even be set to once/week.



MetaLink Doc. ID 787228.1 “Cannot Abort Old Open Items in Workflow Manager Because Errored Items are not Returned” reports that in 12.0.4 an item type would not list in the OAM Errored Work Items page unless the “Workflow Work Items Statistics Concurrent Program” had run. In 12.0.6, clicking the refresh icon will generate the correct counts regardless of whether this program has run.

Workflow Concurrent Managers

There are two concurrent managers that were used by the old c-based mailer. These managers should be disabled. If enabled, follow the instructions in MetaLink Doc. ID 434161.1 “When Attempting To Stop the Concurrent Managers – All Services are Terminated Except FNDSM and WFMLRGSM”. These old concurrent managers are:

- WFMGSMS – Workflow Summary Mailer
- WFMGSMD – Workflow Mailer

The following concurrent managers are used by the workflow technology

- WFALSNRSVC – Workflow Agent Listener Service – this manager must ALWAYS be running
- WFMLRSVC – Workflow Mailer Service – this manager must be running is emailing notifications or using Alert
- WFWSSVC – Workflow Document Web Services – this manager must be running if using Web Services

Pinning

One of the greatest gains in performance occurs when objects are “pinned” into memory so that they do not need to be constantly reloaded from disk, flushed out of memory and reloaded. MetaLink Doc. ID 301171.1 “Toolkit for dynamic marking of Library Cache objects as Kept (PIND)” explains how to set this up. Once setup, Oracle will select the objects that cause the biggest gain in performance.

Pinning requires lots of memory and a large SGA.

64-bit Operating System, 10gR2 or 11g Database

Since Pinning is so important to performance and effective pinning requires lots of memory, customers should be running 64-bit operating systems and 64-bit versions of the database.

Each new version of the database increases Oracle’s ability to self-tune. If still running 9i or 10gR1 databases, upgrade to 10.2.0.4 or 10.2.0.5 or 11g

Partition Tables

When tables are partitioned, queries on these tables can be limited to specific areas, thus increasing performance. Oracle provides a script to partition the following runtime tables:

- WF_ITEM_ACTIVITY_STATUSES
- WF_ITEM_ACTIVITY_STATUSES_H

- WF_ITEM_ATTRIBUTE_VALUES
- WF_ITEMS

This is best done after cleaning up errors and purging unnecessary data as the volume of data will be lower. If still running a 9i database, this also replaces the need to export/import these tables after the cleanup/purge.

This step must be done with the database down. Since the script uses DDL operations running in nologging mode, rollback is not possible, so back up these tables. Failure will require a restore. The step only has to be done once.

Partitioning workflow tables was introduced early in Release 11i. The 11.5.10+ documentation still references that wfupartb.sql is the script to use. However, OWF.G (11.5.9) replaced this script with wfpart.sql. The wfupartb.sql script starting with version 115.11 is just an empty script. The correct method of partitioning tables is described in MetaLink Doc. ID: 260884.1 "How to Partition tables in OWF.G". The script can be found in the \$FND_TOP/PATCH/115/SQL directory. Log into SQLPlus as <apps_user>. Then execute the following script:

```
@wfpart <fnd_user> <fnd_passwd> <apps_user> <apps_passwd> <utl_dir_location>
```

The fnd user is by default applsys and the apps user is by default apps.

MetaLink Doc. ID: 329738.1 "OWF.G Workflow Partitioning Fails At index Wf_item_activity_statuses_n4" explains an error in the script and provides the fix.

The new script also now allows for unpartitioning. See MetaLink Doc. ID: 567008.1 "How to Unpartition Workflow Tables After Partitioning (sic) Them Using \$FND_TOP/PATCH/115/SQL/WFPART.SQL.

Synchronize WF LOCAL tables

This program helps ensure that all changes to users, responsibilities and employees are accurately reflected in the Directory Services tables. There is an option that allows you to run in NOLOGGING mode. While faster, running in this mode can cause corruption if for some reason the database crashes and the DBA runs recovery.

This is one instance where it is better to sacrifice the small gain with NOLOGGING and ensure that these critical tables will not become corrupt.

There is a bug that causes the database to mark some of the partitions NOLOGGING. The following MetaLink Doc IDs explain how to deal with this bug and ensure all partitions in these tables remain in LOGGING mode:

- MetaLink Doc. ID: 433280.1 "Partitions in Workflow Local Tables are Automatically Switched to NOLOGGING"
- MetaLink Doc. ID 369535.1 "During a Backup, WF_LOCAL_ROLES Is Showing Corrupt Block"

Wffngen.sql

Oracle has provided a script that translates activity function calls into static calls. Oracle states that this will provide a 25% increase in performance. The script is \$FND_TOP/SQL/wffngen.sql. Open the script and look for the variable itemtypeList_t. The seeded value is "= itemtypeList_t ('WFSTD','FNDFFWF')". You are allowed to add other item types to this list. Add WFEROR. If using purchasing, add POERROR. If using Order Management, add OMERROR. If there are other workflows that have very high counts in WF_ITEMS, these can be added as well. However, do not add any workflows to this list if you are planning on customizing them.

Item Attributes "As Needed"

By default when a workflow is initiated a record is inserted into WF_ITEM_ATTRIBUTE_VALUES for each defined attribute. When looking at completed workflows, 66% of these records never received a value

(that excludes attributes of type EVENT). You can use the following SQL to see how many records in your environment never receive a value

```
SELECT    COUNT (*)
          ,v.item_type
FROM      wf_item_attribute_values v
          ,wf_item_attributes a
WHERE     a.item_type = v.item_type
          AND a.NAME = v.NAME
          AND a.TYPE <> 'EVENT'
          AND v.text_value IS NULL
          AND v.number_value IS NULL
          AND v.date_value IS NULL
GROUP BY v.item_type
ORDER BY 1 DESC;
```

Compare this count to a total count of records from WF_ITEM_ATTRIBUTE_VALUES.

Oracle recognizes the savings of not creating unused records, both in the execution of the workflow as well as in tasks like purging. Therefore a special attribute #ONDEMANDATTR can be added to the top-level runnable process activity. This attribute can be almost any type (do not pick Item Attribute) and have any Display Name. It doesn't need a value. If this attribute is detected, records are inserted into WF_ITEM_ATTRIBUTE_VALUES only when they are first referenced in WF_ENGINE.SetItemAttr<>. If for some reason a WF_ENGINE.GetItemAttr<> call is made before a value is established, the default value is used (usually NULL).

Using this attribute will require customizing the workflow. So if you are interested in using this attribute, experiment with a particular workflow. Try HRSSA, XDPWFSTD, OEOL, WFERROR, APEXP, POWFRQAG, or REQAPPRV.

Help for OEOH/OEOL

Due to the parent/child relationships with the Order Header (OEOH) and Order Line (OEOL) workflows, these workflows are hard to manage, hard to purge, and tend to remain in the tables. This issue is further complicated if any of the workflows error and call either OMERROR or WFERROR. If a company has been running Order Management for several years, the effort to research all the old workflows may seem undoable. Oracle has provided two MetaLink notes that provide scripts to help with this issue and that will automatically detect workflows that are safe to close. These notes are:

- MetaLink Doc. ID 398822.1 "Order Management Suite – Data Fix Script Patch"
- MetaLink Doc. ID 405275.1 "How to Detect Data Corruption and Purge More Eligible OEOH/OEOL Workflow Items for Order Management Workflow"

Like any new code, it should be thoroughly tested in a test environment before applying to production.

Conclusion

There are myriad ways to monitor workflow and increase performance. The key is to treat workflow as both an application and a technology. Applications require setup, technology requires monitoring. Do both and you will be rewarded with a system that aides users in performing their jobs.