Workflow Performance Tuning in Release 12

Karen Brownfield
Infosemantics, Inc

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 E-Business Suite Release 121.3, however most of the content also applies to 11.5.10 with ATG_PF.H.delta.6 (RUP 6) or ATG_PF.H.delta.7 (RUP 7) and earlier R12 releases. Note that ATG_PF.H.delta.6 is the minimum requirement to be eligible for 11.5.10 Extended Support.

Which Are You?

Workflow has been an integral part of E-Business Suite Releases since Release 11. Even if your company declares it is not using workflow, the E-Business Suite is. Beginning with 11.5.8, 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 My Oracle Support for new documents and new practices that can positively impact the performance of workflow in their environments.

Many companies fall in neither category. They either weren't aware of how to effectively monitor the environment or they inherited an environment that is out of control and they are overwhelmed.

Workflow Analyzer

This is a script delivered by Oracle as part of the Support Health-Check (Proactive Support) series. The script is delivered via MOS note 1369938.1 “Workflow Analyzer script for E-Business Suite Workflow Monitoring and Maintenance”. It is based on workflow Best Practices and feedback from customers who have used this script. It is updated often, so administrators should visit the note often and verify they have the
latest copy. MOS note 1425053.1 “How to run EBS Workflow Analyzer Tool as a Concurrent Request” provides setups that allow this script to be run as a concurrent request. (Note that display issues are encountered when running this script from TOAD or SQL Developer although it works from SQL*Plus). MOS note 1452224.1 “Workflow Analyzer Tool FAQ” provides FAQs for this script and MOS note 1386194.1 “EBS – Technology area – Webcast recording ‘Workflow Analyzer Healthcheck – Introduction’ provides a recorded webcast that demonstrates many features of this script.

Following is the table of contents from script run on a sample environment.

Table of Contents

Workflow Analyzer Overview
- E-Business Suite Version
- Workflow Database Parameter Settings

Workflow Administration
- Verify, the Workflow Administrator Role
- ESGADM User Setup for Error Notifications
- ESGADM Worklist Access
- ESGADM Notification Routing Rules
- Verify AutoClose, PI Setting
- E-Business Suite Profile Settings
- Workflow Profile Settings
- Verify Error Messages
- Verify RSTIXY Activities
- Verify Execution Preference
- Verify the Status of Workflow Services

Workflow Concurrent Programs
- Verify Concurrent Programs Scheduled to Run
- Verify Workflow Background Processes that can
- Verify Status of the Workflow Background Engine Deferred Queue Table
- Verify Workflow Queues Concurrent Programs
- Verify Workflow Central Queue Cleanup Programs

Workflow Notification Mailing
- Check the status of the Workflow Services
- Check the status of the Workflow Notification Mailing Table
- Check Status of WF_NOTIFICATIONS Table
- Check Status of WF_NOTIFICATIONS Q Table
- Check for Closed Notifications

Workflow Patch Levels
- Applied ATG Patches
- Known 5-OfF Patches on top of ATG RUP6
- Verify Status of Workflow Log Levels
- Verify Workflow Services Log Locations

References

This paper will use select sections from this script where they pertain to performance.

Patch Current

Because workflow is part of the ATG group, and this group underlies all applications, it is important to patch current. If still running Release 11i, to be eligible for Extended, you must be at 11i.ATG.RUP6 or higher. If running RUP6, there are also many one-offs that should be applied. Release 12 adds this functionality to the overall rollups. So you should be at 12.0.6 or any release of 12.1.

Remember, if there are issues with a specific workflow, information about that workflow will be found in that product’s documentation and My Oracle Support section.


As part of preparing this paper, all workflow patches for 12.1.n were put into a spreadsheet and published at the Workflow SIG site: [http://workflowsig.oaug.org/reference_info.html](http://workflowsig.oaug.org/reference_info.html). I have no personal knowledge of these patches. Many of them are for bugs, but a few are for performance. If you are using this list, please be aware that many patches provide lower versions of objects that other patches also listed. Oracle has not marked any of these patches superceded.

**Clean up Errored Workflows**

![Workflow Analyzer Overview](image)

The above workflow analyzer picture come from an environment where workflows are routinely purged. Errored workflows are one of the reasons for the red status.

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 various papers by Susan Behn or myself on “Workflow Troubleshooting” for an explanation of how to use these screens.

The following query will show if there are any errored workflows

```sql
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 1 DESC, 5 DESC, 2; -- orders by highest count
```

You can also try running the script with an Order By of 5 DESC, 1 DESC, 2 -- orders by most recent
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.

<table>
<thead>
<tr>
<th>COUNT(*)</th>
<th>ITEM_TYPE</th>
<th>ACTIVITY_NAME</th>
<th>MIN(ITEM_BEGIN_DATE)</th>
<th>MAX(ITEM_BEGIN_DATE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>806</td>
<td>GMDQMSM...</td>
<td>GMDQMSMC_VERIFY</td>
<td>3/15/2004 3:04:40 PM</td>
<td>7/24/2012 7:34:01 PM</td>
</tr>
</tbody>
</table>

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.

```sql
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', 'PARMARTX', 'POERROR', 'WFSTD', 'XDPWFSTD', 'ZPBWFERR', 'WFERROR')
GROUP BY item_type,
       parent_item_type,
       DECODE (end_date, NULL, 'OPEN', 'CLOSED')
ORDER BY 4 DESC, 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. Note that the “Purge Obsolete Workflow Runtime Data” concurrent program will close WFERRORs if the associated activity is now closed. Complicated chains such as OEOH -> OMERROR -> WFERROR or OEOH -> OEOE -> WFERROR can exist.

<table>
<thead>
<tr>
<th>ITEM_TYPE</th>
<th>PARENT_ITEM_TYPE</th>
<th>ERROR_TYPE_STATUS</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFERROR</td>
<td>FTEPSUPD</td>
<td>CLOSED</td>
<td>16528</td>
</tr>
<tr>
<td>WFERROR</td>
<td>FTEPSUPD</td>
<td>OPEN</td>
<td>1571</td>
</tr>
<tr>
<td>WFERROR</td>
<td>REQAPPRV</td>
<td>CLOSED</td>
<td>1550</td>
</tr>
<tr>
<td>OMERROR</td>
<td>OEOH</td>
<td>OPEN</td>
<td>38</td>
</tr>
<tr>
<td>POERROR</td>
<td>REQAPPRV</td>
<td>CLOSED</td>
<td>35</td>
</tr>
<tr>
<td>ECXERROR</td>
<td>OKLFLBO</td>
<td>CLOSED</td>
<td>10</td>
</tr>
<tr>
<td>PARMARTX</td>
<td>PARMARTX</td>
<td>OPEN</td>
<td>1</td>
</tr>
</tbody>
</table>
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. The Purge concurrent program will not close these workflows. 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.

```sql
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 4 DESC, 1 DESC, text_value;
```

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).

In order to aid with cleaning up these WFERRORs, I modified the wfrumtype.sql script provided by Oracle ($FND_TOP/sql). The parameter TYPE was modified to be the event name that invoked the WFERROR.

```sql
PROMPT
PROMPT Enter NUMBER OF days OF OPEN history TO KEEP
ACCEPT nodays prompt "Enter number of days of open history to keep:"

ACCEPT TYPE prompt "Enter WFERROR event to purge:"

UPDATE wf_items wi
SET wi.end_date = SYSDATE
WHERE wi.item_type = 'WFERROR'
AND wi.end_date IS NULL
AND TRUNC(wi.begin_date) < TRUNC(SYSDATE - &nodays)
AND EXISTS (SELECT wiav.item_key
FROM wf_item_attribute_values wiav
WHERE wiav.item_key = wi.item_key
AND wiav.item_type = 'WFERROR'
AND wiav.name = 'EVENT_NAME'
AND wiav.text_value = &TYPE);
```

After ensuring the event would not error again, this script will end-date all records in WF_ITEMS for the WFERROR that references the event. Subsequent updates in the script (see below for example) were modified to key off the records end-dated in WF_ITEMS. The script then purges the end-dated workflows.

Portions of this script were provided as a guide. This script is not sanctioned by Oracle. It should NEVER be run in a
production instance without extensive testing in a non-production instance.

```
UPDATE wf_item_activity_statuses
SET end_date = SYSDATE
WHERE item_type = 'WFERROR'
AND end_date IS NULL
AND item_key IN (  
    SELECT item_key  
    FROM wf_items  
    WHERE item_type = 'WFERROR'  
    AND TRUNC (end_date) = TRUNC (SYSDATE));
```

**Clean up Old Workflows**

Another reason for the red dashboard is old workflows that were never completed. This chart from the Workflow Analyzer shows a count of workflow by year. Since the SQL for this query is provided, it would be very easy to modify this script to also group by item_type. Then you have to question whether you would want a workflow that is 13 years old to continue. The answer may vary by workflow type, but most organizations would be very upset if a 13-year-old PO suddenly surfaced and was approved and issued.

<table>
<thead>
<tr>
<th>Year</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>272</td>
</tr>
<tr>
<td>2001</td>
<td>469</td>
</tr>
<tr>
<td>2002</td>
<td>920</td>
</tr>
<tr>
<td>2003</td>
<td>905</td>
</tr>
<tr>
<td>2004</td>
<td>1,727</td>
</tr>
<tr>
<td>2005</td>
<td>8,556</td>
</tr>
<tr>
<td>2006</td>
<td>7,711</td>
</tr>
<tr>
<td>2007</td>
<td>7,487</td>
</tr>
<tr>
<td>2008</td>
<td>120,012</td>
</tr>
<tr>
<td>2012</td>
<td>490</td>
</tr>
</tbody>
</table>

**Total OPENED Items Started over 90 Days Ago**

**SQL Script**

**Warning**
There are 148,479 OPEN item_types in WF_ITEMS table that were started over 90 days ago. Remember that once a Workflow is closed, its runtime data which is stored in Workflow Runtime Tables (WF_*) becomes obsolete. All pertinent information is stored in the functional tables (PMO_*, POL_*, AP_*, HR_*, OR_*, etc.), like who approved what, for how much, for whom, etc... and the each single row in WF_ITEMS can represent 100s or 1000s of rows in the subsequent Workflow Runtime tables, so it is important to close these open workflows once completed so they can be purged.

**Action**
Ask the Question: How long should these workflows take to complete? A 30 Days... 60 Days... 6 months... 1 Year? There may be valid business reasons why these OPEN items still exist after 90 days so that should be taken into consideration. However, if this is not the case, then once a workflow item is closed then all the runtime data associated to completing this workflow is now obsolete and should be purged to make room for new workflows.

Please review [Note 144806.1 - A Detailed Approach To Purging Oracle Workflow Runtime Data](#) for details on how to drill down to discover the reason why these OLD items are still open, and ways to close them so they can be purged.

**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.
Notice that if the Persistence Type is PERM, Purgeable may be zero even if the workflow are purgeable (in 11i, the value is always 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 (Oracle recommends daily). 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/week run set to N to purge orphaned notifications
- 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 My Oracle Support documents 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)
- 144806.1 "A Detailed Approach to Purging Oracle Workflow Runtime Data"
- 458886.1 “How To Diagnose Issues Related To Purge Of Purchasing Workflow Data That Remain Even After Running The ‘Purge Obsolete Workflow Runtime Data’ Concurrent Program?”
  - Contains scripts that will aide in closing attached children
- 751026.1 “FNDWFPR – Purge Obsolete Workflow Runtime Data – OEOH / OEOL Performance Issues”
  - Contains scripts that will aide in closing attached children
- 398822.1 “Order Management Suite – Data Fix Script Patch”
- 405275.1 “How to Detect Data Corruption and Purge More Eligible OEOH/OEOL Workflow Items for Order Management Workflow”
- 752383.1 “Purge Obsolete Workflow Runtime Data Concurrent Request (FNDWFPR) Is Not Purging Data”
- 1378954.1 “bde_wf_process_tree.sql – For analyzing the Root, Children, Grandchildren Associations of a Single Workflow”
- 878032.1 “How To Use Concurrent Program “Purge Order Management Workflow” (this is provided for 11i via patch 9845873, it is included in 12.1.2+ and is not available for 12.0.x releases)

Note that most referenced patches are already in 11i and R12

**How Does Purge Work?**

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 POERROR 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 I recommend running a special purge with “Core Workflow Only” set to N only once per week or even once per month.

The Workflow Analyzer provides a section on Orphaned Notifications. Monitoring this count will help you determine how often to run with Core Workflow set to N.
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.

Note that with database 10g and higher, Oracle states that it is no longer necessary to import/export tables to adjust the high water marks. However, as the following shows, it may still be recommended.

Purging usually deletes records from the beginning of the table. Thus the high water mark is not changed. New workflows will be written to the empty space, but full table scans will read through the empty space until purges remove all the records at the end of the table. Any code that executes a full table scan (like count(*)) may be time-consuming. If an Export/Import is done after a “catch-up” purge, then the database high water mark is reset and the native functionality to manage it will work effectively.

The workflow analyzer contains a section that will aide in deciding whether to perform the export/import and which tables to consider.

---

**Workflow Footprint**

<table>
<thead>
<tr>
<th>Workflow Table</th>
<th>Logical Table Size</th>
<th>Physical Table Size</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>WF_COMMENTS</td>
<td>57</td>
<td>44</td>
<td>13</td>
</tr>
<tr>
<td>WF_DLG_SIDS</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>WF_ITEMS</td>
<td>15</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>WF_ITEM_ACTIVITY_STATUSES</td>
<td>146</td>
<td>60</td>
<td>85</td>
</tr>
<tr>
<td>WF_ITEM_ACTIVITY_STATUSES_H</td>
<td>889</td>
<td>830</td>
<td>59</td>
</tr>
<tr>
<td>WF_ITEM_ATTRIBUTE_VALUES</td>
<td>780</td>
<td>500</td>
<td>280</td>
</tr>
<tr>
<td>WF_NOTIFICATIONS</td>
<td>122</td>
<td>107</td>
<td>15</td>
</tr>
<tr>
<td><strong>TOTALS</strong></td>
<td>2009</td>
<td>1550</td>
<td>458</td>
</tr>
</tbody>
</table>

---

Attention
The Workflow Runtime Tables logical space which is used for all full-table scans is only at 33% greater than the physical or actual table space being used. It is recommended at levels above 50% to resize these tables to maintain or reset the table High Water Mark for optimum performance. Please have a DBA monitor these tables going forward to ensure they are being maintained at optimal levels.

Please review Note 558072.1 – How to Reorganize Workflow Tables, on how to manage workflow runtime tablespaces for optimum performance.
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. My Oracle Support 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 I 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/week. And once all workflows that end in the status #STUCK are corrected, the resubmission interval should be increased to 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.

The Workflow Analyzer contains a section that will help you determine the appropriate interval for the Deferred Background Engine. Monitor these counts. If they grow, either run the Background Engine more frequently or setup a background engine for the specific workflow with the high count (note that APPS is appended to the front of every workflow name). If the counts decrease, then increase the interval between runs.
Note per MOS Note 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 MOS Note 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” or the new WF Analysis script from MOS note 1369938.1. 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.

The Workflow Analyzer will provide a list of the top 30 Large Item Activity Status History Items
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. And make sure you follow the instructions in MOS note 428441.1 “‘Warning: Aq_tm_processes is Set To 0’ Message in Alert Log After Upgrade to 10.2.0.3 or Higher” to ensure that the setting is truly set to auto-tune, not 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 MOS Notes: 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”.

Note that MOS note 396009.1 “Database Initialization Parameters for Oracle E-Business Suite Release 12” states that auto-tuning has not been tested with EBS and not to use it. This isn’t true according to the author of the WF Analyzer script and is an uncorrected doc bug. MOS note 578831.1 “How to determine the correct setting for JOB_QUEUE_PROCESSES” gives a good discussion of how to determine the optimal setting for this parameter and recommends periodic monitoring to ensure the setting remains correct.

Note: Significance of the AQ_TM_PROCESSES for 10gR1~ (10.1) databases:

Starting from 10gR1, Queue Monitoring can utilize a feature called “auto-tune”. That means Queue Monitoring does not need AQ_TM_PROCESSES to be defined, it is instead able to adapt to the number of AQ background processes to the system load. However, if you do specify a value, then that value is taken into account but the number of processes can still be auto-tuned and so the number of running qXXX processes can be different from what was specified by AQ_TM_PROCESSES.

Oracle Workflow recommends to UNSET the AQ_TM_PROCESSES parameter as per DB recommendations to enable auto-tune feature.

Note: For more information refer to Note 746313.1 - What should be the Correct Setting for Parameter AQ_TM_PROCESSES in E-Business Suite Instance?

It should be noted that if AQ_TM_PROCESSES is explicitly specified then the process(es) started will only maintain persistent messages. For example if AQ_tm_processes=1 then at least one queue monitor slave process will be dedicated to maintaining persistent messages. Other process can still be automatically started to maintain buffered messages. Up to and including version 11.1.1 if you explicitly set AQ_tm_processes = 10 then there will be no processes available to maintain buffered messages. This should be borne in mind in environments which use Streams replication and from 10.2 onwards user enqueued buffered messages.

It is also recommended to NOT DISABLE the Queue Monitor processes by setting qaq_tm_processes on a permanent basis. As can be seen above, disabling will stop all related processing in relation to tasks outlined. This will likely have a significant affect on operation of queues - PROCESSED messages will not be removed and any time related, TM actions will not succeed, AQ objects will grow in size.

Note: There is a known issue viewing the true value of AQ_TM_PROCESSES for 10gR2~ (10.2) from the v$parameters table. Review the details in Note 928441.1 - Warning: Aq_tm_processes is Set To 0’ Message in Alert Log After Upgrade to 10.2.0.3 or Higher.

To check whether AQ_TM_PROCESSES Auto-Tuning is enabled, follow the steps outlined in Note 505661.1 - Master Note for AQ Queue Monitor Process (QMON) under Section: Significance of the AQ_TM_PROCESSES Parameter in 10.1 onwards.
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. Note 396009.1 recommends a value of 2. This is wrong.

Note: Significance of the JOB_QUEUE_PROCESSES for 11gR1 (11.1) databases:

Starting from 11gR1, the init.ora parameter job_queue_processes does NOT need to be set for AQ propagations. AQ propagation is now likewise handled by DBMS_SCHEDULER jobs rather than DBMS_JOBS. Reason: propagation takes advantage of the event based scheduling features of DBMS_SCHEDULER for better scalability. If the value of the JOB_QUEUE_PROCESSES database initialization parameter is zero, then that parameter does not influence the number of Oracle Scheduler jobs that can run concurrently. However, if the value is non-zero, it effectively becomes the maximum number of Scheduler jobs and Job queue jobs than can run concurrently. If a non-zero value is set, it should be large enough to accommodate a Scheduler job for each Messaging Gateway agent to be started.

Oracle Workflow recommends to UNSET the JOB_QUEUE_PROCESSES parameter as per DB recommendations to enable the scheduling features of DBMS_SCHEDULER for better scalability.

To update the JOB_QUEUE_PROCESSES database parameter file (init.ora) file:

```
job_queue_processes=10
```

or set dynamically via

```
alter system set job_queue_processes=10;
```

Remember that after bouncing the DB, dynamic changes are lost, and the DB parameter file settings are used.

To determine the proper setting of JOB_QUEUE_PROCESSES for Oracle Workflow, follow the queries outlined in Note 578831.1 - How to determine the correct setting for JOB_QUEUE_PROCESSES.

Advanced Queuing Performance

Workflow is heavily dependent on Advanced Queuing. MOS Note 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 MOS Note 271855.1 "Procedure to manually Coalesce all the IOTs/indexes Associated with Advanced Queuing table to maintain Enqueue/Dequeue performance, reduce QMON CPU usage and Redo generation".

Another good Note for ensuring the queues are working properly is MOS Note 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 – Yes
- FND: Debug Log Level – Unexpected
- FND: Debug Log Module – %

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.

If performance is suffering either because the containers do not have enough memory or because they have been allocated too much memory, see MOS Note 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.

For more information on Queue performance, download the white paper “Application Development with Oracle Advanced Queuing” by Jeff Jacobs from http://jefferyjacobs.com/news.htm. Jeff recommends adding an index on the CORRID column. The trick is to get the CBO to recognize this index, which can be done by generating an appropriate set of statistics or by the various forms of SQL plan management.

**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. MOS Note 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. MOS Notes 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 MOS Note 286394.1 “How to rebuild the WF_DEFERRED queue”.

Profile Options

Check the following profile options:

- “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 MOS Note 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. MOS Note 472387.1 “FYI Notifications Closed Automatically and Not Showing Up on Worklist” details the adjustment that must be made to avoid this.

Notification Mailer

This part of the workflow technology seems to generate the most MOS Notes. 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 MOS Notes 743344.1 “How To Create a dedicated mailer for Alerts” and 463777.1 “How to Disable all Workflow related Email Notifications Except for the Ones Sent from Oracle Alerts?” 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 MOS Note 453137.1 “Oracle Workflow Best Practices Release 12 and Release 11i”, section “Choosing Not to Use E-mail Notifications”.

```
+-----------------+-----------------+---+-----------------+-----------------+-----------------+
| System          | Source Type     | Out Agent             | To Agent Function          | Workflow         | Status          |
| A1200NT1.SOLUTIONBEACON.NET | Local         | wf.xml.SendNotification | WF_RULE.DEFAULT_RULE          | Enabled          |
| A1200NT1.SOLUTIONBEACON.NET | Local         | WF_NOTIFICATION_OUT@A1200NT1.SOLUTIONBEACON.NET | WF_RULE.DEFAULT_RULE          | Enabled          |
| A1200NT1.SOLUTIONBEACON.NET | Error         | WF_NOTIFICATION_OUT@A1200NT1.SOLUTIONBEACON.NET | WF_XML_ERROR_RULE          | Enabled          |
+-----------------+-----------------+---+-----------------+-----------------+-----------------+
```

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**

MOS Note 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 MOS Notes: 422870.1 “Java Mailer not
Removing Processed Emails from Inbox after Folder ATG.H Rup4", 332152.1 “OWF.H Diagnostics, Solutions and Information", 741352.1 “Unable to Connect to Mail Store in Workflow Notification Mailer Logs", and 601423.1 “Emails Not Received or Delayed Log file contains: javax.mail.MessagingException: 451 Timeout waiting for client input”.

**Avoid java.lang.OutOfMemoryError**

MOS Note 467516.1 “Users suddenly (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/adovars.env and add/change the following variable.

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

After making the change, bounce the concurrent managers. While this is an 11.5.9 Note, 11.5.10 also used the same Java Mailer, so the Note still applies.

**Mailer Should Have a Dedicated User – “You Have Insufficient Privileges”**

Oracle has fixed the issue with the ICX Limit Connect profile option that used to cause the mailer to timeout and throw the error “You Have Insufficient Privileges". But studies show that the mailer still does better if the user managing it can provide undivided attention. Plus it’s easier to read the log files when the only activity is the mailer. Log files for SYSADMIN include quite a few other functions.

This decision will force the workflow administrator to be set to a responsibility as SYSADMIN must still have the ability to administer workflows. So this new user must have the workflow administrator responsibility and the responsibility “System Administrator”. This user should not have any other duties other than to manage the mailer.
In addition to setting a dedicated user, set the “Framework URL timeout” parameter to 12. See MOS Note 414376.1 “You Have Insufficient Privileges For The Current Operation” On Reqapprv Notif.

Tag Files

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 MOS Note 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 MOS Note 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.
MOS Note 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 and higher, 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 MOS Note 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. MOS Note 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, 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. Release 10.2.0.5 is in Extended Support phase, versions below that are desupported or in Sustaining Support. Upgrade to latest certified version of 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 MOS Note 260884.1 “How to Partition tables in OWF.G”. The script can be found in the $FND_TOP/PATCH/115/SQ directory. Log into SQLPlus as <apps_user>. Then execute the following script:

```sql
@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.

MOS Note 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 MOS Note 567008.1 “How to Unpartition Workflow Tables After Partitioning (sic) Them Using $FND_TOP/PATCH/115/SQ/WFPART.SQL.

**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:

```sql
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

The section on purging had quite a few notes specifically aimed at the OM workflows. And the Workflow Analyzer has a section specifically for OM workflows

**Product Specific Workflows**

**ONT - Order Management Workflow Specific Summary**

There are 3,630 Order Header (ODON) workflow items found in WF_ITEMS. Currently 76% (2,821) of OEOHs are OPEN, while 24% (879) are CLOSED, but still found in the runtime tables. The following collection of information is a sample of the more complete Order Management Review that you can get from running OMSuiteDataCheck.sql, found in [Note 555991.1](#). The purpose of this script is to collect information related to data integrity in OM and Shipping products.

**Show the status of the Order Management Workflows for this instance**

There are 7,167 Order Management Workflows found on this instance.

**SUMMARY of Workflow Processes by Item Type and Status**

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Active</th>
<th>Completed</th>
<th>Purgeable</th>
<th>Item Name</th>
<th>Display Name</th>
<th>Persistence Type</th>
<th>Persistence Days</th>
<th>Errored</th>
<th>Deferred</th>
<th>Suspended</th>
</tr>
</thead>
<tbody>
<tr>
<td>OEOH</td>
<td>352</td>
<td>9,359</td>
<td>4,070</td>
<td>OEOH</td>
<td>OM Order Line</td>
<td>TEMP</td>
<td>0</td>
<td>77</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>3,202</td>
<td>1,237</td>
<td>POWFRQAG</td>
<td>OM Order Line</td>
<td>TEMP</td>
<td>0</td>
<td>0</td>
<td>222</td>
<td>0</td>
</tr>
</tbody>
</table>

Elapsed time: 0.12 seconds

**Back to Top**

It is normal for Order Management Workflows to use WAITS and other looping activities to process delayed responses and other criteria. Each revisit of a node replaces the previous data with the current activities status and stores the old activity information into a activities history table. Looking at this history table (WF_ITEM_ACTIVITY_STATUSES.gif) can help to identify possible long running workflows that appear to be stuck in a loop over a long time, or a poorly designed workflow that is looping excessively and can cause performance issues.
MOS note 130511.1 “PERFORMANCE Issues in OM, SE, QP” recommends the following:
- Remove unnecessary activities from the line processes in OEOL
- Avoid calling sub-processes
- Make Scheduling a deferred activity (increase the cost)
- Consider using the seeded Line process "Line Flow Generic: Performance" which does avoid sub-processes and did remove unnecessary activities

**Release 12.2**

Release 12.2 introduces two new features that will help with performance. The first one is “Deferred Worklist Processing”. Currently when using the Worklist to respond to notifications, control is not returned to the screen until the workflow is ended or reaches another deferred activity. With this new feature, control is returned immediately. The next activity in the workflow will be executed when the Background Engine runs again for deferred activities. This feature will be configurable by item type.

The second feature is RAC Affinity for item types whose processes are either launched and completed in one online session or deferred once and completed later by the Background Engine. This feature is also configurable by item type using the lookup type WF_RAC_ENABLED_TYPES.

**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.