|  |  |  |
| --- | --- | --- |
| **Best Practices: Proactive Data Collection for Performance Issues (Doc ID 1477599.1)** | [IMG_256](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326%26parent=DOCUMENT%26sourceId=1523048.1%26id=1477599.1%26_afrWindowMode=0%26_adf.ctrl-state=131s153wz5_175%20/o%20To%20Bottom)  [To Bottom](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\o To Bottom) | IMG_257 |

IMG_258

IMG_259

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **In this Document**   |  |  | | --- | --- | |  | [Purpose](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l PURPOSE) |  |  |  | | --- | --- | |  | [Ask Questions, Get Help, And Share Your Experiences With This Article](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section11) |  |  |  | | --- | --- | |  | [Scope](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l SCOPE) |  |  |  | | --- | --- | |  | [Details](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l BODYTEXT) |  |  |  | | --- | --- | |  | [Executive Summary](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section31) |  |  |  | | --- | --- | |  | [Methodology](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section32) |  |  |  | | --- | --- | |  | [Top-Down approach:](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section33) |  |  |  | | --- | --- | |  | [Data collection at Operating System (O/S) level.](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section34) |  |  |  | | --- | --- | |  | [OSWatcher](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section35) |  |  |  | | --- | --- | |  | [Data collection at Database level](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section36) |  |  |  | | --- | --- | |  | [AWR](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section37) |  |  |  | | --- | --- | |  | [ASH](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section38) |  |  |  | | --- | --- | |  | [Establishing multiple baselines:](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section39) |  |  |  | | --- | --- | |  | [Be Prepared!: Having the right tools already installed and running before a problem occurs](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section310) |  |  |  | | --- | --- | |  | [HangFG](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section311) |  |  |  | | --- | --- | |  | [SQLHC](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section312) |  |  |  | | --- | --- | |  | [SQLTXPLAIN (SQLT)](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section313) |  |  |  | | --- | --- | |  | [Deploying specialized tools for unstable environments:](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section314) |  |  |  | | --- | --- | |  | [LTOM](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section315) |  |  |  | | --- | --- | |  | [Procwatcher](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section316) |  |  |  | | --- | --- | |  | [What to collect prior to an upgrade](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section317) |  |  |  | | --- | --- | |  | AWR baselines |  |  |  | | --- | --- | |  | [SQL Plan management Baselines](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section319) |  |  |  | | --- | --- | |  | Checklist of Proactive Best Practices |  |  |  | | --- | --- | |  | [Logging A Service Request](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section321) |  |  |  | | --- | --- | |  | [What to do if no proactive collection has occurred?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section322) |  |  |  | | --- | --- | |  | [Performance Service Request Diagnostic Collection (SRDC) documents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section323) |  |  |  | | --- | --- | |  | [Discuss Proactive Avoidance!](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section324) |  |  |  | | --- | --- | |  | [References](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l REF) |   IMG_260  **APPLIES TO:**  Oracle Database - Standard Edition - Version 10.1.0.2 and later  Oracle Database - Enterprise Edition - Version 10.1.0.2 and later  Oracle Database - Personal Edition - Version 10.1.0.2 and later  Information in this document applies to any platform.  **PURPOSE**  This document describes a best practice methodology to ensure that sufficient performance data is pro-actively gathered enough at first occurrence of the problem to enable effective root cause determination. It can be used alongside the following document to both attempt to avoid problems and then if that is unavoidable, collect the information required for a quick diagnosis:  [Document 1482811.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1482811.1) Best Practices: Proactively Avoiding Database and Query Performance issues  NOTE: These recommendations are best practices that apply to the MAJORITY of scenarios encountered by support. Each issue is different and specific additional diagnostic may be required in some circumstances to completely achieve a root cause diagnosis. It is not necessarily feasible to collect this targeted information up front for every issue since specific diagnostics required to resolve one particular issue may not be relevant in all cases. The goal is to provide a solid starting point that will collect enough for most issues to be resolved or to enable prompt recommendations for further tracing.  **Ask Questions, Get Help, And Share Your Experiences With This Article**  **Would you like to explore this topic further with other Oracle Customers, Oracle Employees, and Industry Experts?   Click** [here](https://community.oracle.com/message/11981941 /o Proactive Performance Problem Avoidance and Proactive Diagnostic Collection /t _blank) to join the discussion where you can ask questions, get help from others, and share your experiences with this specific article.  Discover discussions about other articles and helpful subjects by clicking [here](https://community.oracle.com/community/support/oracle_database/database_tuning /o My Oracle Support Community - Database Tuning /t _blank) to access the main *My Oracle Support Community* page for Database Tuning.  **SCOPE**  This document is intended to advise database users on proactive best practices for the collection of diagnostic performance data  **DETAILS**  **Executive Summary**  It is well known that collecting sufficient data to resolve complex performance issues is difficult to achieve. Historically, users would encounter a performance problem, contact Oracle support, only to be told insufficient data had been collected at the time or no data exists that would enable support to resolve the issue after the first occurrence of the problem. users were then advised to turn on additional data collection after the fact (followed by an iterative process of collecting more data), sending it in to support, only to be told again that insufficient data was collected requiring further outages and more data collection.  This document describes a methodology to eliminate or reduce unnecessary data collections to reduce the amount of time and effort spent, to resolve the issue in a timely manner. All of the methods outlined have a minimal impact on the database itself in terms of performance and some (such as those associated with the Automatic Workload repository (AWR)) are already integrated.  **Methodology**  Our best practice methodology consists of the following:   * Top-Down approach to data collection * Establishing multiple baselines * Having the right tools already installed and running before the problem occurs * Deploying specialized tools for unstable environments   **Top-Down approach:**  ***Data collection at Operating System (O/S) level.***  Oracle can only perform optimally when the server it runs on is also performing optimally. It is sensible therefore to begin data collection at the server level by using OSWatcher to capture operating system metrics so that the performance of the server can be monitored and adjusted.   * ***OSWatcher***   OSWatcher (OSW) contains a built in analyzer that allows the data that has been collected to be automatically analyzed, pro-actively looking for cpu, memory, io and network issues. It is recommended that all users install and run OSW since it is invaluable for looking at issues on the OS and has very little overhead.   Once installed and running, OSWatcher will provide 48 hours worth of "look back" data on the OS by default. So if, for example, a node eviction occurred at 2am, Oracle support would be able to see what was happening on the OS during that time from the OSWatcher logs. Prior to the existence of OSWatcher, there was no way to look back at what might have happened on the OS during the time of an outage or serious performance problem, nor was Oracle aware of what was happening on the OS.    **Refer to the following for download, user guide and usage videos on OSWatcher**  [Document 301137.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=301137.1) OSWatcher User Guide (Includes: [Video])  ***Data collection at Database level***  Several utilities are available that can collect comprehensive data for performance analysis:   * ***AWR***   AWR is the most comprehensive utility for data collection around performance problems on the database. It is used primarily to collect metrics around the database (although it also includes some OS metrics).   Provided you do not anticipate having performance issues and are on a stable environment, our best practice recommendation is to enable AWR snapshots at the default rate of 60 minutes. If you are concerned about having performance issues, then more frequent snapshots are recommended. In this case, we recommend snapshots at a maximum interval of 20 minutes; if you can afford to do so, more frequent snapshots than this are always better.  More frequent snapshots allows us to see what is happening on the database with higher levels of granularity and can be used to compare the times to when the database performance was good. Whatever snapshot interval you choose, try to stick to it so as to facilitate comparison between reports.   It is important to have captured some snapshots that can be considered good baselines of normal performance that can be used later to compare to when problems occur. Many times just having AWR data alone will allow for enough information for bug identification and in some instances will provide enough information to diagnose database hangs and other issues without the need to take special additional diagnostics like systemstate dumps and hanganalyze traces.   AWR can also be used to drill down on specific sql statements. If the problem is at the session level, an AWR report can be obtained and analyzed before trying to take additional 10046 or sql trace diagnostics. This information can also be used in conjunction with ASH reports (see below).   See the following article for more information about AWR:  [Document 1363422.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1363422.1) Automatic Workload Repository (AWR) Reports - Start Point   * ***ASH***   Active Session History (ASH) reports provide very granular metric collection being as they drills down to the session level. In contrast to the aggregated view of performance data provided by AWR, ASH provides information at 1 second level accuracy for each individual database session. This is very important for intermittent performance problems or hangs.  Leveraging ASH data can sometimes be enough to diagnose a problem at the session level thus preventing the need to take additional 10046 or sql trace diagnostics. ASH reports can be obtained as needed through the Advanced Workload Repository(AWR).  **For more details about Active Session History(ASH), See:**  [Document 243132.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=243132.1) 10g and above Active Session History (Ash) And Analysis Of Ash Online And Offline  Automatic Workload Repository (AWR) and Active Session History (ASH) Reports are separate components of Oracle Diagnostic Pack and must be licensed as a separate option. As a best practice it is advised to obtain and use this license so you can access this data. See:  [Document 1490798.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1490798.1) AWR Reporting - Licensing Requirements Clarification  Otherwise, you must use the statspack utility.  **Establishing multiple baselines:**  Baseline captures should be obtained and stored for various time periods depending upon your business profile. Suggested baseline collections would be:   * normal activity * non-busy times * the busiest time of the day * month end or business cycle processing * batch processing.   Having these multiple baselines will give you a good idea on how the system was performing normally. When a problems occurs, comparison to these baselines will help in the resolution of the problem. Failure to establish baselines makes it more difficult to understand the nature of the performance problem. If users only supply an AWR when the system is performing poorly it is much more difficult  to analyze the performance of the database; with nothing to compare with database performance can become a "subjective observation".  As a best practice support recommends creating baselines for both the O/S (OSW) and the database (AWR).  **Be Prepared!: Having the right tools already installed and running before a problem occurs**  In addition to installing and running OSW and collecting AWR at specified intervals, Oracle support has some specialized tools that should be installed on your server and ready to use if a problem happens.  NOTE: These tools do not have to be running, but pre-installation allows you to quickly collect information in the event of an issue rather than missing the opportunity and having to wait for a re-occurrence   * ***HangFG***   HangFG allows for the collection of hang diagnostics without a user having to know what kind and level of trace to take. If HangFG is installed, and a hang occurs, the user has a simple unix shell command line interface that allows them to choose how "heavy" a data collection they can afford.   **See the following for download and user guide for Hangfg.**  [Document 362094.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=362094.1) HANGFG User Guide  There are currently 3 levels the user can choose from to initiate automatic generation of hang diagnostics. This provides the user the flexibility to take hang diagnostics while being as unobtrusive as possible (if the database still remains in a functional state).     1. Light impact on system. This option collects 2 hanganalyze level 3 traces and then determines whether it can also collect 1 hanganalyze level 4 trace with minimal impact to the system. If so, it collects the hanganalyze level 4 trace. If not, it does not collect an additional trace file. 2. Medium impact on system (default value). This option collects 1 hanganalyze level 3 trace and then determines whether it can also collect 2 hanganalyze level 4 traces with minimal impact to the system. If so, it collects the 2 additional hanganalyze level 4 traces. If not, it collects an additional hanganalyze level 3 trace. This option also collects 1 systemstate level 266 trace. 3. Heavy impact on system. This option collects 2 hanganalyze level 4 traces and 2 systemstate level 266 traces.      * ***SQLHC***   The SQL Tuning Health-Check Script is a tool developed by the Oracle Server Technologies Center of Expertise. The tool, also known as SQLHC, is used to check the environment in which a single SQL Statement runs, checking Cost-based Optimizer (CBO) statistics, schema object metadata, configuration parameters and other elements that may influence the performance of the one SQL being analyzed.  More details about the tool can be found here:  [Document 1366133.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1366133.1) SQL Tuning Health-Check Script (SQLHC)  The intention of SQLHC is to allow users to ensuring that the environment that an individual SQL runs in is sound and hopefully avoid SQL Performance Issues from avoidable problems. It does this while leaving "no database footprint" ensuring it can be run on all systems. When executed for one SQL\_ID, this script generates an HTML report with the results of a set of health-checks around the one SQL statement provided.   * ***SQLTXPLAIN (SQLT)***   An even more sophisticated tool exists for resolving SQL performance problems (but this requires a footprint on the database). SQLTXPLAIN, also known as SQLT, is a tool provided by support, inputs one SQL statement and outputs a set of diagnostics files. These files are commonly used to diagnose SQL statements performing poorly. SQLT connects to the database and collects execution plans, Cost-based Optimizer CBO statistics, schema objects metadata, performance statistics, configuration parameters, and similar elements that influence the performance of the SQL being analyzed.   See the following for more details on SQLT.  [Document 215187.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=215187.1) SQLT (SQLTXPLAIN) - Tool that helps to diagnose a SQL statement performing poorly    **Deploying specialized tools for unstable environments:**  Most users are running without any performance issues in stable environments. For those users that do not have a stable environment and are experiencing hangs or transient performance problems that cannot be resolved through the traditional data collections described above, Oracle Support has some specialized tools to aid in debugging of these kinds of issues.   * ***LTOM***   LTOM is a very specialized tool that can be installed on customer unix/linux platforms. This tool can be configured to automate data collection around hangs and transient issues that happen infrequently or happen to quickly for users to manually recognize they have an issue and try and take manual diagnostics. For example, LTOM can look for transient issues or database hangs and automatically collect diagnostics and notify the user when these issues happen via email and produce the required diagnostic traces needed to resolve the issue. If you database hangs at 2am, when no one is around, LTOM can automatically detect and take diagnostics so support can have the required diagnostic traces when you log an SR.   See the following article for download and user guide.  [Document 352363.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=352363.1) LTOM - The On-Board Monitor User Guide   * ***Procwatcher***   Procwatcher is a tool to examine and monitor Oracle database and/or clusterware processes at an interval. The tool will collect stack traces of these processes using Oracle tools like oradebug short\_stack and/or OS debuggers like pstack, gdb, dbx, or ladebug and collect SQL data if specified.   See the following article for details:  [Document 459694.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=459694.1) Procwatcher: Script to Monitor and Examine Oracle DB and Clusterware Processes  **What to collect prior to an upgrade**  An upgrade can be viewed as a special case where you know something is going to change; specifically the version of the database. Since a version change is likely to contain new features and fixes for defects that may alter the performance of some of your queries, it makes sense to collect baseline information from before the upgrade so that you have something to compare against after the upgrade. To do this we would suggest:  ***AWR baselines***  In a similar fashion to the standard baselines suggested earlier, take AWR snapshots of the key baseline performance operations so that these can be compared to the post upgrade situation in case of issues. Suggested baseline collections would be:   * normal activity * the busiest time of the day * month end or business cycle processing * batch processing   ***SQL Plan management Baselines***  SQL Plan Management can be used to preserve SQL performance across releases. If you wish to preserve the performance of SQL before and after an upgrade, create baselines of the SQL statements that you wish to preserve. We would recommend doing this at least for the key SQL statements in the application. Transport them to the new system and enable them.  **Checklist of Proactive Best Practices**  \_\_\_\_\_\_ Install and run OSWatcher on every node where Oracle Database is installed.               Run the OSWatcher analyzer daily to look for performance issues on the server.  \_\_\_\_\_\_ Obtain Diagnostic Pack License  \_\_\_\_\_\_ Configure AWR snapshot interval and verify AWR snapshots are happening at expected intervals  \_\_\_\_\_\_ Establish Multiple baselines for both the O/S (using OSW) and the database (using AWR).  \_\_\_\_\_\_ Download Hangfg and have it ready to run if you experience a database hang  \_\_\_\_\_\_ Install SQLHC and run at expected intervals  \_\_\_\_\_\_ Download SQLT and have it ready if you need to install it on the database  \_\_\_\_\_\_ Consider downloading and installing LTOM if you are running in an unstable environment and issue can not be resolved by the use of above tools  **Logging A Service Request**  In the event that a SR is required, please refer to the following for details of what to include:  [Document 210014.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=210014.1) How to Log a Good Performance Service Request  **What to do if no proactive collection has occurred?**  There will be times when pro-active steps have not been taken to collect information in advance of a problem having occurred. In those cases we have numerous articles that outline how to handle each particular occurrence and suggest best practices for collecting the relevant data. See:  [Document 1377446.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1377446.1) Troubleshooting Performance Issues  Note that in these cases it may be necessary for the problem to be repeated in order to be able to collect information since it may be that it is not possible to retrospectively gather diagnostics in all cases.  **Performance Service Request Diagnostic Collection (SRDC) documents**  Service Request Data Collection (SRDC) documents have been specially designed to provide the reader with the necessary instructions to provide a step by step guide to collecting information for a various common Database Performance Issues.  [Document 1938786.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1938786.1) List of Available Database Performance Related SRDC Documents  **Discuss Proactive Avoidance!**  **The window below is a live discussion of this article (not a screenshot). We encourage you to join the discussion by clicking the "Reply" link below for the entry you would like to provide feedback on. If you have questions or implementation issues with the information in the article above, please share that below.** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IMG_261 | 9876  IMG_262   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | |  |  |  |  | | --- | --- | --- | --- | |  | **Doc ID** | **…** |  | |  | |  |  |  |  | | [IMG_263](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326%26parent=DOCUMENT%26sourceId=1523048.1%26id=1477599.1%26_afrWindowMode=0%26_adf.ctrl-state=131s153wz5_175%20/o%20/Click%20to%20add%20to%20Favorites/) | IMG_264 | **Best Practices: Proactive Data Collection for Performance Issues (Doc ID 1477599.1)** | [IMG_265](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326%26parent=DOCUMENT%26sourceId=1523048.1%26id=1477599.1%26_afrWindowMode=0%26_adf.ctrl-state=131s153wz5_175%20/o%20To%20Bottom)  [To Bottom](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\o To Bottom) | IMG_266 |   IMG_267  IMG_268   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **In this Document**   |  |  | | --- | --- | |  | [Purpose](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l PURPOSE) |  |  |  | | --- | --- | |  | [Ask Questions, Get Help, And Share Your Experiences With This Article](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section11) |  |  |  | | --- | --- | |  | Scope |  |  |  | | --- | --- | |  | [Details](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l BODYTEXT) |  |  |  | | --- | --- | |  | [Executive Summary](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section31) |  |  |  | | --- | --- | |  | [Methodology](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section32) |  |  |  | | --- | --- | |  | [Top-Down approach:](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section33) |  |  |  | | --- | --- | |  | [Data collection at Operating System (O/S) level.](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section34) |  |  |  | | --- | --- | |  | [OSWatcher](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section35) |  |  |  | | --- | --- | |  | [Data collection at Database level](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section36) |  |  |  | | --- | --- | |  | [AWR](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section37) |  |  |  | | --- | --- | |  | [ASH](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section38) |  |  |  | | --- | --- | |  | [Establishing multiple baselines:](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section39) |  |  |  | | --- | --- | |  | [Be Prepared!: Having the right tools already installed and running before a problem occurs](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section310) |  |  |  | | --- | --- | |  | [HangFG](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section311) |  |  |  | | --- | --- | |  | [SQLHC](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section312) |  |  |  | | --- | --- | |  | [SQLTXPLAIN (SQLT)](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section313) |  |  |  | | --- | --- | |  | [Deploying specialized tools for unstable environments:](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section314) |  |  |  | | --- | --- | |  | [LTOM](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section315) |  |  |  | | --- | --- | |  | [Procwatcher](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section316) |  |  |  | | --- | --- | |  | [What to collect prior to an upgrade](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section317) |  |  |  | | --- | --- | |  | AWR baselines |  |  |  | | --- | --- | |  | [SQL Plan management Baselines](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section319) |  |  |  | | --- | --- | |  | Checklist of Proactive Best Practices |  |  |  | | --- | --- | |  | Logging A Service Request |  |  |  | | --- | --- | |  | [What to do if no proactive collection has occurred?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section322) |  |  |  | | --- | --- | |  | [Performance Service Request Diagnostic Collection (SRDC) documents](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section323) |  |  |  | | --- | --- | |  | [Discuss Proactive Avoidance!](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l aref_section324) |  |  |  | | --- | --- | |  | [References](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=437035475904326&parent=DOCUMENT&sourceId=1523048.1&id=1477599.1&_afrWindowMode=0&_adf.ctrl-state=131s153wz5_175 \\l REF) |   IMG_269 **APPLIES TO:** Oracle Database - Standard Edition - Version 10.1.0.2 and later  Oracle Database - Enterprise Edition - Version 10.1.0.2 and later  Oracle Database - Personal Edition - Version 10.1.0.2 and later  Information in this document applies to any platform. **PURPOSE** This document describes a best practice methodology to ensure that sufficient performance data is pro-actively gathered enough at first occurrence of the problem to enable effective root cause determination. It can be used alongside the following document to both attempt to avoid problems and then if that is unavoidable, collect the information required for a quick diagnosis:  [Document 1482811.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1482811.1) Best Practices: Proactively Avoiding Database and Query Performance issues  NOTE: These recommendations are best practices that apply to the MAJORITY of scenarios encountered by support. Each issue is different and specific additional diagnostic may be required in some circumstances to completely achieve a root cause diagnosis. It is not necessarily feasible to collect this targeted information up front for every issue since specific diagnostics required to resolve one particular issue may not be relevant in all cases. The goal is to provide a solid starting point that will collect enough for most issues to be resolved or to enable prompt recommendations for further tracing. **Ask Questions, Get Help, And Share Your Experiences With This Article** **Would you like to explore this topic further with other Oracle Customers, Oracle Employees, and Industry Experts?   Click** [here](https://community.oracle.com/message/11981941 /o Proactive Performance Problem Avoidance and Proactive Diagnostic Collection /t _blank) to join the discussion where you can ask questions, get help from others, and share your experiences with this specific article.  Discover discussions about other articles and helpful subjects by clicking [here](https://community.oracle.com/community/support/oracle_database/database_tuning /o My Oracle Support Community - Database Tuning /t _blank) to access the main *My Oracle Support Community* page for Database Tuning. **SCOPE** This document is intended to advise database users on proactive best practices for the collection of diagnostic performance data **DETAILS****Executive Summary** It is well known that collecting sufficient data to resolve complex performance issues is difficult to achieve. Historically, users would encounter a performance problem, contact Oracle support, only to be told insufficient data had been collected at the time or no data exists that would enable support to resolve the issue after the first occurrence of the problem. users were then advised to turn on additional data collection after the fact (followed by an iterative process of collecting more data), sending it in to support, only to be told again that insufficient data was collected requiring further outages and more data collection.  This document describes a methodology to eliminate or reduce unnecessary data collections to reduce the amount of time and effort spent, to resolve the issue in a timely manner. All of the methods outlined have a minimal impact on the database itself in terms of performance and some (such as those associated with the Automatic Workload repository (AWR)) are already integrated. **Methodology** Our best practice methodology consists of the following:   * Top-Down approach to data collection * Establishing multiple baselines * Having the right tools already installed and running before the problem occurs * Deploying specialized tools for unstable environments   **Top-Down approach:** ***Data collection at Operating System (O/S) level.*** Oracle can only perform optimally when the server it runs on is also performing optimally. It is sensible therefore to begin data collection at the server level by using OSWatcher to capture operating system metrics so that the performance of the server can be monitored and adjusted.   * ***OSWatcher***   OSWatcher (OSW) contains a built in analyzer that allows the data that has been collected to be automatically analyzed, pro-actively looking for cpu, memory, io and network issues. It is recommended that all users install and run OSW since it is invaluable for looking at issues on the OS and has very little overhead.   Once installed and running, OSWatcher will provide 48 hours worth of "look back" data on the OS by default. So if, for example, a node eviction occurred at 2am, Oracle support would be able to see what was happening on the OS during that time from the OSWatcher logs. Prior to the existence of OSWatcher, there was no way to look back at what might have happened on the OS during the time of an outage or serious performance problem, nor was Oracle aware of what was happening on the OS.    **Refer to the following for download, user guide and usage videos on OSWatcher**  [Document 301137.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=301137.1) OSWatcher User Guide (Includes: [Video]) ***Data collection at Database level*** Several utilities are available that can collect comprehensive data for performance analysis:   * ***AWR***   AWR is the most comprehensive utility for data collection around performance problems on the database. It is used primarily to collect metrics around the database (although it also includes some OS metrics).   Provided you do not anticipate having performance issues and are on a stable environment, our best practice recommendation is to enable AWR snapshots at the default rate of 60 minutes. If you are concerned about having performance issues, then more frequent snapshots are recommended. In this case, we recommend snapshots at a maximum interval of 20 minutes; if you can afford to do so, more frequent snapshots than this are always better.  More frequent snapshots allows us to see what is happening on the database with higher levels of granularity and can be used to compare the times to when the database performance was good. Whatever snapshot interval you choose, try to stick to it so as to facilitate comparison between reports.   It is important to have captured some snapshots that can be considered good baselines of normal performance that can be used later to compare to when problems occur. Many times just having AWR data alone will allow for enough information for bug identification and in some instances will provide enough information to diagnose database hangs and other issues without the need to take special additional diagnostics like systemstate dumps and hanganalyze traces.   AWR can also be used to drill down on specific sql statements. If the problem is at the session level, an AWR report can be obtained and analyzed before trying to take additional 10046 or sql trace diagnostics. This information can also be used in conjunction with ASH reports (see below).   See the following article for more information about AWR:  [Document 1363422.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1363422.1) Automatic Workload Repository (AWR) Reports - Start Point   * ***ASH***   Active Session History (ASH) reports provide very granular metric collection being as they drills down to the session level. In contrast to the aggregated view of performance data provided by AWR, ASH provides information at 1 second level accuracy for each individual database session. This is very important for intermittent performance problems or hangs.  Leveraging ASH data can sometimes be enough to diagnose a problem at the session level thus preventing the need to take additional 10046 or sql trace diagnostics. ASH reports can be obtained as needed through the Advanced Workload Repository(AWR).  **For more details about Active Session History(ASH), See:**  [Document 243132.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=243132.1) 10g and above Active Session History (Ash) And Analysis Of Ash Online And Offline  Automatic Workload Repository (AWR) and Active Session History (ASH) Reports are separate components of Oracle Diagnostic Pack and must be licensed as a separate option. As a best practice it is advised to obtain and use this license so you can access this data. See:  [Document 1490798.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1490798.1) AWR Reporting - Licensing Requirements Clarification  Otherwise, you must use the statspack utility. **Establishing multiple baselines:** Baseline captures should be obtained and stored for various time periods depending upon your business profile. Suggested baseline collections would be:   * normal activity * non-busy times * the busiest time of the day * month end or business cycle processing * batch processing.   Having these multiple baselines will give you a good idea on how the system was performing normally. When a problems occurs, comparison to these baselines will help in the resolution of the problem. Failure to establish baselines makes it more difficult to understand the nature of the performance problem. If users only supply an AWR when the system is performing poorly it is much more difficult  to analyze the performance of the database; with nothing to compare with database performance can become a "subjective observation".  As a best practice support recommends creating baselines for both the O/S (OSW) and the database (AWR). **Be Prepared!: Having the right tools already installed and running before a problem occurs** In addition to installing and running OSW and collecting AWR at specified intervals, Oracle support has some specialized tools that should be installed on your server and ready to use if a problem happens.  NOTE: These tools do not have to be running, but pre-installation allows you to quickly collect information in the event of an issue rather than missing the opportunity and having to wait for a re-occurrence   * ***HangFG***   HangFG allows for the collection of hang diagnostics without a user having to know what kind and level of trace to take. If HangFG is installed, and a hang occurs, the user has a simple unix shell command line interface that allows them to choose how "heavy" a data collection they can afford.   **See the following for download and user guide for Hangfg.**  [Document 362094.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=362094.1) HANGFG User Guide  There are currently 3 levels the user can choose from to initiate automatic generation of hang diagnostics. This provides the user the flexibility to take hang diagnostics while being as unobtrusive as possible (if the database still remains in a functional state).     1. Light impact on system. This option collects 2 hanganalyze level 3 traces and then determines whether it can also collect 1 hanganalyze level 4 trace with minimal impact to the system. If so, it collects the hanganalyze level 4 trace. If not, it does not collect an additional trace file. 2. Medium impact on system (default value). This option collects 1 hanganalyze level 3 trace and then determines whether it can also collect 2 hanganalyze level 4 traces with minimal impact to the system. If so, it collects the 2 additional hanganalyze level 4 traces. If not, it collects an additional hanganalyze level 3 trace. This option also collects 1 systemstate level 266 trace. 3. Heavy impact on system. This option collects 2 hanganalyze level 4 traces and 2 systemstate level 266 traces.      * ***SQLHC***   The SQL Tuning Health-Check Script is a tool developed by the Oracle Server Technologies Center of Expertise. The tool, also known as SQLHC, is used to check the environment in which a single SQL Statement runs, checking Cost-based Optimizer (CBO) statistics, schema object metadata, configuration parameters and other elements that may influence the performance of the one SQL being analyzed.  More details about the tool can be found here:  Document 1366133.1 SQL Tuning Health-Check Script (SQLHC)  The intention of SQLHC is to allow users to ensuring that the environment that an individual SQL runs in is sound and hopefully avoid SQL Performance Issues from avoidable problems. It does this while leaving "no database footprint" ensuring it can be run on all systems. When executed for one SQL\_ID, this script generates an HTML report with the results of a set of health-checks around the one SQL statement provided.   * ***SQLTXPLAIN (SQLT)***   An even more sophisticated tool exists for resolving SQL performance problems (but this requires a footprint on the database). SQLTXPLAIN, also known as SQLT, is a tool provided by support, inputs one SQL statement and outputs a set of diagnostics files. These files are commonly used to diagnose SQL statements performing poorly. SQLT connects to the database and collects execution plans, Cost-based Optimizer CBO statistics, schema objects metadata, performance statistics, configuration parameters, and similar elements that influence the performance of the SQL being analyzed.   See the following for more details on SQLT.  [Document 215187.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=215187.1) SQLT (SQLTXPLAIN) - Tool that helps to diagnose a SQL statement performing poorly   **Deploying specialized tools for unstable environments:** Most users are running without any performance issues in stable environments. For those users that do not have a stable environment and are experiencing hangs or transient performance problems that cannot be resolved through the traditional data collections described above, Oracle Support has some specialized tools to aid in debugging of these kinds of issues.   * ***LTOM***   LTOM is a very specialized tool that can be installed on customer unix/linux platforms. This tool can be configured to automate data collection around hangs and transient issues that happen infrequently or happen to quickly for users to manually recognize they have an issue and try and take manual diagnostics. For example, LTOM can look for transient issues or database hangs and automatically collect diagnostics and notify the user when these issues happen via email and produce the required diagnostic traces needed to resolve the issue. If you database hangs at 2am, when no one is around, LTOM can automatically detect and take diagnostics so support can have the required diagnostic traces when you log an SR.   See the following article for download and user guide.  [Document 352363.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=352363.1) LTOM - The On-Board Monitor User Guide   * ***Procwatcher***   Procwatcher is a tool to examine and monitor Oracle database and/or clusterware processes at an interval. The tool will collect stack traces of these processes using Oracle tools like oradebug short\_stack and/or OS debuggers like pstack, gdb, dbx, or ladebug and collect SQL data if specified.   See the following article for details:  [Document 459694.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=459694.1) Procwatcher: Script to Monitor and Examine Oracle DB and Clusterware Processes **What to collect prior to an upgrade** An upgrade can be viewed as a special case where you know something is going to change; specifically the version of the database. Since a version change is likely to contain new features and fixes for defects that may alter the performance of some of your queries, it makes sense to collect baseline information from before the upgrade so that you have something to compare against after the upgrade. To do this we would suggest: ***AWR baselines*** In a similar fashion to the standard baselines suggested earlier, take AWR snapshots of the key baseline performance operations so that these can be compared to the post upgrade situation in case of issues. Suggested baseline collections would be:   * normal activity * the busiest time of the day * month end or business cycle processing * batch processing   ***SQL Plan management Baselines***  SQL Plan Management can be used to preserve SQL performance across releases. If you wish to preserve the performance of SQL before and after an upgrade, create baselines of the SQL statements that you wish to preserve. We would recommend doing this at least for the key SQL statements in the application. Transport them to the new system and enable them. **Checklist of Proactive Best Practices** \_\_\_\_\_\_ Install and run OSWatcher on every node where Oracle Database is installed.               Run the OSWatcher analyzer daily to look for performance issues on the server.  \_\_\_\_\_\_ Obtain Diagnostic Pack License  \_\_\_\_\_\_ Configure AWR snapshot interval and verify AWR snapshots are happening at expected intervals  \_\_\_\_\_\_ Establish Multiple baselines for both the O/S (using OSW) and the database (using AWR).  \_\_\_\_\_\_ Download Hangfg and have it ready to run if you experience a database hang  \_\_\_\_\_\_ Install SQLHC and run at expected intervals  \_\_\_\_\_\_ Download SQLT and have it ready if you need to install it on the database  \_\_\_\_\_\_ Consider downloading and installing LTOM if you are running in an unstable environment and issue can not be resolved by the use of above tools **Logging A Service Request** In the event that a SR is required, please refer to the following for details of what to include:  [Document 210014.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=210014.1) How to Log a Good Performance Service Request **What to do if no proactive collection has occurred?** There will be times when pro-active steps have not been taken to collect information in advance of a problem having occurred. In those cases we have numerous articles that outline how to handle each particular occurrence and suggest best practices for collecting the relevant data. See:  [Document 1377446.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1377446.1) Troubleshooting Performance Issues  Note that in these cases it may be necessary for the problem to be repeated in order to be able to collect information since it may be that it is not possible to retrospectively gather diagnostics in all cases. **Performance Service Request Diagnostic Collection (SRDC) documents** Service Request Data Collection (SRDC) documents have been specially designed to provide the reader with the necessary instructions to provide a step by step guide to collecting information for a various common Database Performance Issues.  [Document 1938786.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1938786.1) List of Available Database Performance Related SRDC Documents **Discuss Proactive Avoidance!** **The window below is a live discussion of this article (not a screenshot). We encourage you to join the discussion by clicking the "Reply" link below for the entry you would like to provide feedback on. If you have questions or implementation issues with the information in the article above, please share that below.** **REFERENCES** [NOTE:215187.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=215187.1) - All About the SQLT Diagnostic Tool [NOTE:1377446.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1377446.1) - \* Troubleshooting Performance Issues [NOTE:250655.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=250655.1) - How to use the Automatic Database Diagnostic Monitor [NOTE:301137.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=301137.1) - OSWatcher (Includes: [Video]) [NOTE:1482811.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1482811.1) - Best Practices: Proactively Avoiding Database and Query Performance Issues [NOTE:1363422.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1363422.1) - Automatic Workload Repository (AWR) Reports - Main Information Sources [NOTE:210014.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=210014.1) - How to Log a Good Performance Service Request [NOTE:352363.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=352363.1) - LTOM - The On-Board Monitor User Guide [NOTE:459694.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=459694.1) - Procwatcher: Script to Monitor and Examine Oracle DB and Clusterware Processes [NOTE:243132.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=243132.1) - Analysis of Active Session History (Ash) Online and Offline [NOTE:362094.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=362094.1) - HANGFG User Guide [NOTE:1366133.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1477599.1&id=1366133.1) - SQL Tuning Health-Check Script (SQLHC) | |