

Oracle Trace File Analyzer

Overview

ORACLE WHITE PAPER | NOVEMBER 2017





Table of Contents

Introduction	1
Autonomous Diagnostic Collections	2
Command Interfaces	3
Configure Email Notification	3
Mask Sensitive Data	4
On-demand Analysis and Collection	5
Real-time System & Cluster Status Summary	5
Investigate Logs & Look for Errors	6
Perform Analysis Using the Included Tools	6
Collect Diagnostic Data & Use One Command SRDCs	8
Change Default User Access	10
Change Clusterware Trace Levels	11
Maintain Oracle Trace File Analyzer to the Latest Version	11
Conclusion	12



Introduction

As a DBA, you're expected to do more work, with fewer resources all the time. You're under pressure to keep mission-critical apps up and running. When something goes wrong, everyone looks to you to understand what happened and how to fix it.

It's not always easy. You've got to run the right tools, at the right time. If you're using Oracle Clusterware, then you have to also collect from all the database nodes. You might need to use lots of different tools that you only ever use now and again, and they each have their own syntax.

Once you've managed to get all the data, it can be huge. Only a fraction of what you've collected is useful, but who knows which bit is relevant? That's if you were able to get everything. If you were quick enough before it was overwritten.

In the meantime, you still have a problem. This is costing your company money and you still need to get it fixed.

Oracle Trace File Analyzer helps you perform real-time health monitoring, fault detection & diagnosis via a single interface. It will securely consolidate all distributed diagnostic data.

Its continuously available and watching your logs for significant problems that may affect your service. If desired it can also automatically collect the relevant diagnostics, when it sees these problems.

Oracle Trace File Analyzer knows what is relevant in log files. This allows it to trim them to the smallest size, yet still gather everything necessary. It also collects data across cluster nodes and consolidates everything in one place. Once done collecting it can automatically upload the collection to Oracle Support.

Using important database diagnostic tools is easy with Oracle Trace File Analyzer. It hides the complexity by providing a single interface and syntax to them all.

This all combines to ensure you can get exactly what you need, when you need it and save your business money.

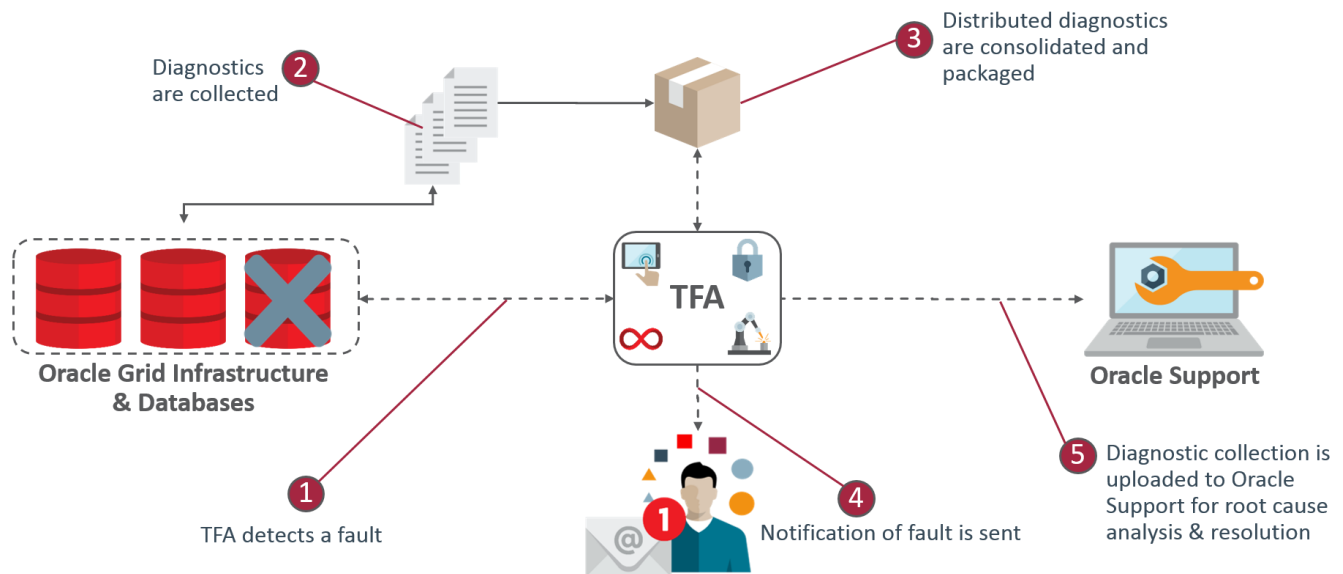
Autonomous Diagnostic Collections

The resource footprint is small. You will not usually be aware it is running. The only times Oracle Trace File Analyzer will consume noticeable CPU are:

- » When performing an inventory of diagnostic files
- » During diagnostic collection

Oracle Trace File Analyzer will watch your logs for significant problems, such as internal errors like ORA-00600 or node evictions. If detected it will:

- Invoke any necessary diagnostics and collect all relevant log data at the time of a problem
- Trim log files around the time of the problem, so it only collects what is necessary for diagnosis
- Collect and package all trimmed diagnostics. From all nodes in the cluster, consolidating everything on a single node
- Store the collection in the Oracle Trace File Analyzer repository
- Send you email notification of the problem and details of diagnostic collection, ready for upload to Oracle Support
- You can then either use TFA to upload the collection to Oracle Support, if you can make a connection from that environment, or transfer the collection somewhere else for upload



Oracle Trace File Analyzer uses a flood control mechanism. Repeated errors do not flood the system with automatic collections.

The identification of an event triggers the start point for a collection. 5 minutes later diagnostic gathering starts. This is to capture any other relevant events together. If after 5 minutes events are still occurring, diagnostic collection continues to wait. It will wait for a period of 30 seconds with no events occurring, up to a further 5 minutes.

If events are still occurring after 10 minutes, a diagnostic collection happens. A new collection point starts.

Once collection completes TFA will send email notification to relevant people, including details of where the collection results are.

Command Interfaces

The tfactl tool functions as:

- » Command line interface
- » Shell interface
- » Menu interface

INTERFACE TYPES & USAGE

Interface Type	Command	How to use
Command line	<code>\$ tfactl <command></code>	Specify all command options at the command line
Shell interface	<code>\$ tfactl</code>	Set and change context. Then run commands from within the shell
Menu Interface	<code>\$ tfactl menu</code>	Select menu navigation options then choose the command you want to run

Automatic collections are ON by default. To change automatic collections use:

```
$ tfactl set autodiagcollect=<ON|OFF>
```

Configure Email Notification

You can provide Oracle Trace File Analyzer with one or more comma separated email addresses to send notification of problems.

To set the notification email address to use for a specific ORACLE_HOME, include the OS owner in the command:

```
$ tfactl set notificationAddress=oracle:some.person@acompany.com
```

To set the notification email to use for any ORACLE_HOME use:

```
$ tfactl set notificationAddress=another.person@acompany.com
```

After receiving notification of a problem, you should:

- Inspect the referenced collection details to determine if you know the root cause
- Resolve the underlying cause of the problem if you know how
- If you do not know the root cause of the problem then log an SR with Oracle Support and upload the relevant collection



Mask Sensitive Data

Oracle Trace File Analyzer can mask sensitive data such as hostnames or IP addresses. To configure masking, edit or create the file **tfa_home/resources/mask_strings.xml**, then copy to each node. The mask_strings.xml should use the following format to define data replacements:

```
<mask_strings>

  <mask_string>

    <original>WidgetNode1</original>

    <replacement>MyReplacementName</replacement>

  </mask_string>

  <mask_string>

    <original>192.168.5.1</original>

    <replacement>Node1-IP</replacement>

  </mask_string>

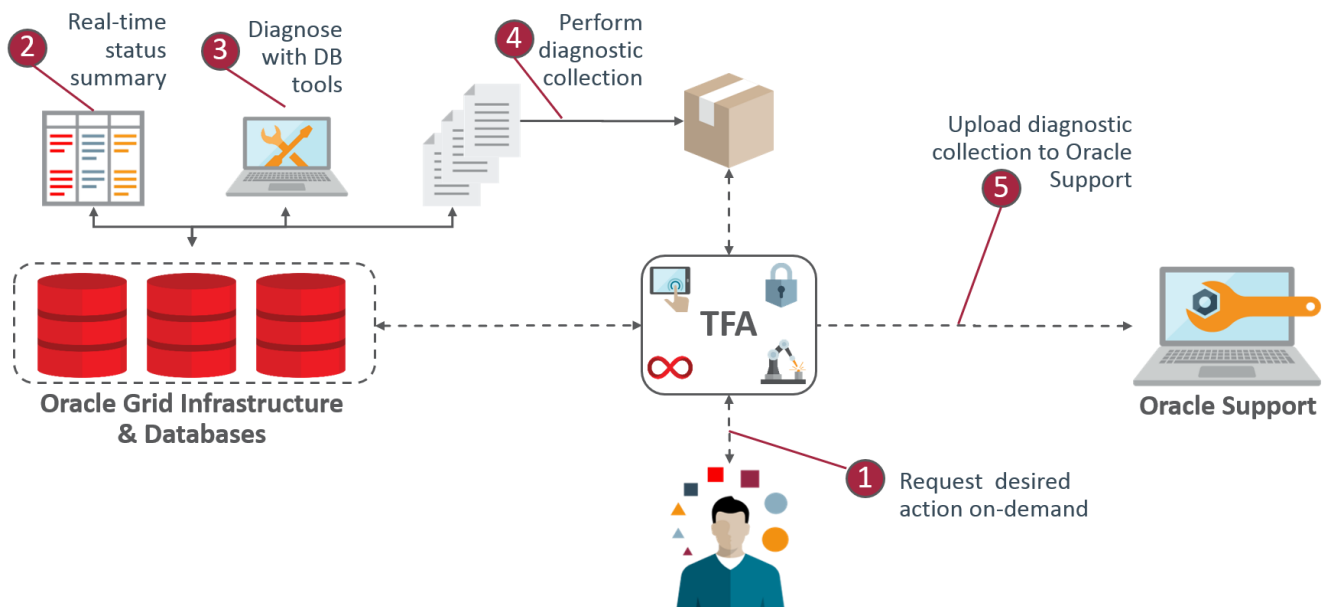
</mask_strings>
```

On-demand Analysis and Collection

You can run Oracle Trace File Analyzer on-demand via the command line tool **tfactl**.

The tfactl command can:

- » Provide you a real-time status summary
- » Perform analysis using a combination of different database tools, using a common syntax.
- » Collect all relevant diagnostic log data, with logs trimmed files around the time, collecting only what is necessary for diagnosis
- » Securely consolidate all distributed collections on the node where tfactl was run from
- » Upload the collection to Oracle Support



Real-time System & Cluster Status Summary

Use the summary command for a real-time report of system and cluster status. It shows a fast, easy to read summary of the status including any potential problems with important elements.

Usage:

```
$ tfactl summary [options]
```

For more help use:

```
$ tfactl summary -help
```

Investigate Logs & Look for Errors

You can use Oracle Trace File Analyzer to analyze all your logs across your cluster and tell you about any recent errors.

For example:

```
$ tfactl analyze -last 1d
```

or

```
$ tfactl analyze -last 18h
```

This will report all errors it finds over the specified duration.

You can also use Oracle Trace File Analyzer to find all occurrences of a specific error on any node. For example, this command will search for ORA-00600 errors:

```
$ tfactl analyze -search "ora-00600" -last 8h
```

Perform Analysis Using the Included Tools

Oracle Trace File Analyzer with database support tools bundle includes the following tools. These tools are only available when Oracle Trace File Analyzer is downloaded from [Document 1513912.1](#).

TOOLS INCLUDED ON LINUX / UNIX

Tool	Description
<code>orachk</code> or <code>exachk</code>	Provides health checks for the Oracle stack. Oracle Trace File Analyzer will install either » <i>Oracle EXAchk for Engineered Systems, see document 1070954.1 for more details</i> or » <i>Oracle ORAchk for all non-Engineered Systems, see document 1268927.2 for more details</i>
<code>oswatcher</code>	Collects and archives OS metrics. These are useful for instance or node evictions & performance issues. See document 301137.1 for more details
<code>procwatcher</code>	Automates & captures database performance diagnostics and session level hang information. See document 459694.1 for more details
<code>oratop</code>	Provides near real-time database monitoring. See document 1500864.1 for more details.
<code>sqlt</code>	Captures SQL trace data useful for tuning. See document 215187.1 for more details.
<code>alertsummary</code>	Provides summary of events for one or more database or ASM alert files from all nodes
<code>ls</code>	Lists all files Oracle Trace File Analyzer knows about for a given file name pattern, across all nodes
<code>pstack</code>	Generates the process stack for the specified processes, across all nodes
<code>grep</code>	Searches for a given string in the alert or trace files with a specified database
<code>summary</code>	Provides high level summary of the configuration
<code>vi</code>	Opens alert or trace files for viewing a given database and file name pattern in the vi editor
<code>tail</code>	Runs a tail on an alert or trace files for a given database and file name pattern
<code>param</code>	Shows all database and OS parameters that match a specified pattern
<code>dbglevel</code>	Sets and unsets multiple CRS trace levels with one command



history	Shows the shell history for the tfactl shell
changes	Reports changes in the system setup over a given time period. This includes database parameters, OS parameters and patches applied
calog	Reports major events from the Cluster Event log
events	Reports warnings and errors seen in the logs
managelogs	Shows disk space usage and purges ADR log and trace files
ps	Finds processes
trriage	Summarize oswatcher/exawatcher data

TOOLS INCLUDED ON WINDOWS

Tool	Description
calog	Reports major events from the Cluster Event log
changes	Reports changes in the system setup over a given time period. This includes database parameters, OS parameters and patches applied
dir	Lists all files Oracle Trace File Analyzer knows about for a given file name pattern, across all nodes
events	Reports warnings and errors seen in the logs
findstr	Searches for a given string in the alert or trace files with a specified database
history	Shows the shell history for the tfactl shell
managelogs	Shows disk space usage and purges ADR log and trace files
notepad	Opens alert or trace files for viewing a given database and file name pattern in the notepad editor
param	Shows all database and OS parameters that match a specified pattern
summary	Provides high level summary of the configuration
tasklist	Finds processes


To verify which tools you have installed run:

```
$ tfactl toolstatus
```

Each tool can be run using tfactl either in command line or shell mode.

To run a tool from the command line use:

```
$ tfactl run <tool>
```



The following example shows how to use tfactl in shell mode. It will start tfactl, connect to the database MyDB and run oratop:

```
$ tfactl
```

```
tfactl > database MyDB
```

```
MyDB tfactl > oratop
```

Collect Diagnostic Data & Use One Command SRDCs

To perform an on-demand diagnostic collection use:

```
$ tfactl diagcollect
```

This will trim & collect all important log files updated in the past 12 hours across the whole cluster. Oracle Trace File Analyzer will store collections in the repository directory. You can change the diagcollect timeframe with the option **-last <n>h|d**.

Oracle Support will often ask you to run a Service Request Data Collection (SRDC). The SRDC is dependent on the type of problem you experienced. It is a series of many data gathering instructions aimed at diagnosing your problem. Collecting the SRDC manually can be difficult, with many different steps required.

Oracle Trace File Analyzer can run SRDC collections for you with a single command:

```
$ tfactl diagcollect -srdc <srdc_type>
```

If your environment can make a connection to oracle.com you can add **-sr <sr_number>** and Oracle Trace File Analyzer will automatically upload the collection to your Service Request for analysis by Oracle Support. Prior to uploading, you will need to store your My Oracle Support credentials in the Oracle Trace File Analyzer wallet. This is a onetime task and must be done by root:

```
$ tfactl setupmos
```

Upload a collection directly to Oracle Support

```
$ tfactl diagcollect -srdc <srdc_type> -sr <sr_number>
```

Upload any other file to Oracle Support:

You can upload using the wallet, which was setup previously by root using tfactl setupmos:

```
tfactl upload -wallet -sr <sr_number> <space-separated list of files to upload>
```

You can also upload without the wallet. When uploading without the wallet tfactl prompts for the password.

```
tfactl upload -user user_id -sr <sr_number> <space-separated list of files to upload>
```

To run SRDCs use one of the Oracle privileged user accounts:

- » ORACLE_HOME owner
- » GRID_HOME owner

These are the different SRDC types available:

Type of Problem	Available SRDCs
ORA Errors	<ul style="list-style-type: none"> » ORA-00020 » ORA-00060 » ORA-00600 » ORA-00700 » ORA-01555 » ORA-01628 » ORA-04030 » ORA-04031 » ORA-07445 » ORA-27300 » ORA-27301 » ORA-27302 » ORA-30036
Other internal database errors	» <i>internalerror</i>
Database resource problems	» <i>dbunixresources</i>
Database performance problems	» <i>dbperf</i>
Database patching problems	<ul style="list-style-type: none"> » <i>dbpatchinstall</i> » <i>dbpatchconflict</i>
Database install / upgrade problems	<ul style="list-style-type: none"> » <i>dbinstall</i> » <i>dbupgrade</i> » <i>dbpreupgrade</i>
Database storage problems	» <i>asm</i>
Excessive SYSAUX space used by the Automatic Workload Repository (AWR)	» <i>dbawrspace</i>
Database startup / shutdown problems	<ul style="list-style-type: none"> » <i>dbshutdown</i> » <i>dbstartup</i>
XDB installation or invalid object problems	» <i>dbxdb</i>
Data Guard problems	» <i>dbdataguard</i>
Enterprise Manager tablespace usage metric problems	» <i>emtbsmetrics</i>
Enterprise Manager general metrics page or threshold problems	» <i>emmetricalert</i>
Enterprise Manager debug log collection Run emdebugon , reproduce the problem then run emdebugoff , which disables debug again and collects debug logs	<ul style="list-style-type: none"> » <i>emdebugon</i> » <i>emdebugoff</i>
Enterprise Manager target discover / add problems	<ul style="list-style-type: none"> » <i>emcliadd</i> » <i>emclusdisc</i> » <i>emdbsys</i> » <i>emgendisc</i> » <i>emprocdisc</i>
Enterprise Manager OMS restart problems	» <i>emrestartoms</i>

What the SRDCs collect varies for each type, here are two examples:

ONE COMMAND SRDC AND COLLECTIONS

Command	What gets collected
<code>\$ tfactl diagcollect -srcd ORA-04031</code>	<ul style="list-style-type: none">» IPS Package» Patch Listing» AWR report» Memory information» RDA HCVE output
<code>\$ tfactl diagcollect -srcd dbperf</code>	<ul style="list-style-type: none">» ADDM report» AWR for good period and problem period» AWR Compare Period report» ASH report for good and problem period» OS Watcher» IPS Package (if errors during problem period)» ORAchk (performance related checks)

Oracle Trace File Analyzer will prompt you to enter the information it needs based on the SRDC type.

For example, this is what happens when you run the ora4031 SRDC:

```
$ tfactl diagcollect -srcd ORA-04031
```

You are prompted to enter event date/time and database name

Oracle Trace File Analyzer scans the system to identify recent events in the system (up to 10)

Once the relevant event is chosen, it proceeds with diagnostic collection

All required files are identified

All files are trimmed where applicable

All data is package in a zip ready to provide to support

You can also run an SRDC collection in non-interactive silent mode. Just provide all the required parameters up front like this:

```
$ tfactl diagcollect -srcd <srcd_type> -database <db> -from "<date> <time>" -to "<date> <time>"
```

Change Default User Access

By default the GRID_HOME owner and all ORACLE_HOME owners will have access to their respective information via Oracle Trace File Analyzer. No one else will be able to perform collections. User access is only applicable if installed as root on Linux / Unix. User access is not applicable if installed as non root or on Microsoft Windows. Enable or disable user access via:

```
$ tfactl access enable
```

or

```
$ tfactl access disable
```

All commands will apply cluster wide unless you specify **-local** to restrict to only the local node.

USEFUL USER ACCESS COMMANDS

Command	Description
<code>\$ tfactl access lsusers</code>	List user access
<code>\$ tfactl access add -user <user> [-local]</code>	Add users
<code>\$ tfactl access remove -user <user> [-local]</code>	Remove users
<code>\$ tfactl access removeall [-local]</code>	Remove everyone
<code>\$ tfactl access reset</code>	Rest to default access

Change Clusterware Trace Levels

Oracle Support may ask you to enable certain trace levels when reproducing a problem. This is to allow you to collect enough diagnostics to diagnose the cause of the problem.

Oracle Trace File Analyzer makes it easy to enable and then disable the correct trace levels. Use `dbglevel`.

You can find the required trace level settings grouped by problem trace profiles. To set a trace profile use:

```
$ tfactl dbglevel -set <profile >
```

For a list of the available profiles run:

```
$ tfactl dbglevel -help
```

Maintain Oracle Trace File Analyzer to the Latest Version

Oracle releases a new version of Oracle Trace File Analyzer approximately every 3 months.

Applying standard PSUs will automatically update Oracle Trace File Analyzer. However the PSU version does not contain the rest of the Database Support Tools Bundle updates. So if you want the latest and greatest version of Oracle Trace File Analyzer with the database support tools bundle download it from [Document 1513912.1](#)

Upgrading is similar to first time install. As root use the `installTFA<platform>` script. If Oracle Trace File Analyzer is already installed, it will update the existing installation. When already installed, a cluster upgrade does not need ssh. The cluster upgrade uses the existing daemon secure socket communication between hosts.

```
$.installTFA<platform>
```

If you're not able to install as root then install as the ORACLE_HOME owner. Use the `-extractto` and `-javahome` options:

```
$.installTFA<platform> -extractto <dir> -javahome <jre_home>
```

Conclusion

Oracle Trace File Analyzer helps you keep mission-critical apps up and running.

The key features are:

- » A single interface to all database diagnostic tools
- » Monitoring of logs to detect significant problems
- » Real-time monitoring of system and cluster
- » Logs analysis to determine which diagnostic information is relevant to a problem
- » Simple collection of data across cluster nodes, consolidated in one place
- » Upload of collections to Oracle Support
- » Reduction in time required to obtain diagnostic data, which saves your business money

If something does go wrong, Oracle Trace File Analyzer helps you analyze and collect what you need, to fix it.

For more information and to download Oracle Trace File Analyzer see [Document 1513912.1](#).



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US



Integrated Cloud Applications & Platform Services

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. This document is provided *for* information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1117

Oracle Trace File Analyzer – Overview Guide
November 2017
Author: Gareth Chapman & Bob Caldwell
Contributing Authors: Bill Burton & Sandesh Rao



Oracle is committed to developing practices and products that help protect the environment