Trace File Analyzer Collector

User Guide

Version 12.1.2.7.0

Contents

Introduction

1. Overview of TFA Collector

One of the biggest challenges for customers and support engineers particularly when working Grid Infrastructure (GI) and Real Applications Clusters (RAC) issues is collecting pertinent data in a timely fashion possibly across multiple nodes in a cluster. A big part of this challenge is frequently the data that is required is lost or overwritten, due to the diagnostic collection not happening until sometime after the problem occurred. For single Instance databases Automatic Diagnostic Repository (ADR) does a good job of generating trace packages when a database incident occurs, but ADR does not include clusterware trace files and is currently not RAC aware. For the clusterware the diagcollection tool (diagcollection.pl) gathers all the logs when manually requested to do so, and though this tool has proved to be useful, it is run sometime after the problem has occurred and is generally run to gather all possible logs without regard to their relevance which can make the data set very large. Further it has to be run as root across all nodes of a cluster which makes it more difficult to manage from an operational perspective.

TFA Collector (from this point referred to as TFA) overcomes some of these problems by running a diagnostic JVM on each node in the cluster synchronized to determine when diagnostic collection is required (when enabled) as well as collecting and extracting only the data that is pertinent to resolving the issue. TFA is written outside of the GI /RAC/RDBMS product lines and as such could be used for any trace data, and is version and platform agnostic.

2. Basic Architecture of TFA Collector

TFA runs on each node of your cluster or on a single node where no Grid Infrastructure is installed. TFA consists of a daemon, and a Command Line Interface (CLI). The TFA daemon is a Java VM (JVM) that by default runs at all times on any configured node and can be identified as TFAMain.

```
# ps -ef | grep TFAMain
root     5190     1  5 01:06 ?        00:01:36 /u01/app/tfa/tfa_home/jre1.6.0_18/bin/java -Xms128m -
Xmx512m -classpath /u01/app/tfa/tfa_home/jlib/RATFA.jar:/u01/app/tfa/tfa_home/jlib/je-
5.0.84.jar:/u01/app/tfa/tfa_home/jlib/ojdbc6.jar:/u01/app/tfa/tfa_home jlib/commons-io-2.2.jar
oracle.rat.tfa.TFAMain /u01/app/tfa/tfa_home
```

It is also possible to determine the status of TFA from the CLI as described later. TFAMain runs as root, is multi-threaded and completes automatic, peer TFAMain and CLI driven tasks. For peer TFAMain and CLI driven tasks TFAMain listens on a secure socket for instructions.

The CLI is made up of a Java command line processor that sends commands to the TFAMain secure socket and a perl script 'tfactl'. This two level approach is used to ensure that unwanted commands cannot be sent to the TFAMain JVM for processing by unprivileged users.

For new installations from 2.5.1.5 forward to support shared file system installation all the binaries for TFA are held in a TFA_HOME that sits under a TFA_BASE/<nodename> directory which can be set at install time unless TFA is installed as part of Oracle Grid Infrastructure when TFA_BASE will be the Grid Infrastructure home directory. All the diagnostic collections are written to a repository directory which by default sits under the TFA_BASE unless TFA is installed as part of Oracle Grid Infrastructure

when the repository will be the Grid Infrastructure owners ORACLE_BASE/tfa directory. If TFA 2.5.1.5 or above patches an existing installation the binaries remain in the originally installed TFA_HOME directory unless the patching is done as part of a Grid Infrastructure install/patch when the TFA_HOME will be moved under the Grid Infrastructure Home directory.

It is possible to limit the amount of space TFA will take up for diagnostics by setting a maximum size. TFA stores any metadata that it needs to persist in a Berkeley Database (BDB) that is placed under the TFA_HOME. The metadata for files and their events will be stored until the file is physically deleted or the directory it resides in is removed from TFA. TFA monitors free space and will halt data collections free space falls below 1GB in the TFA_HOME file system or the repository hits its maximum repository size.

3. Supported Platforms

On all platforms for version 12.1.2.x.x of TFA Collector the bash shell (version 3.2 or higher) must be available to install and run. For all platforms except Linux Exadata or Oracle Database Appliance (ODA) Dom0 systems a version 1.5 or later JRE must also be installed and available at the same location on all nodes. So long as those requirements are met the following Platforms are supported.

- Intel Linux (Enterprise Linux, RedHat Linux, SUSE Linux)
- Linux Itanium
- Oracle Solaris SPARC
- Oracle Solaris x86-64
- AIX
- HPUX Itanium
- HPUX PA-RISC
-zLinux

Note: When TFA is installed as part of Grid Infrastructure the JRE supplied with the Grid Infrastructure home is used.

TFA Collector Install/Patch/De-install

4. Managing TFA Collector Installation

On the Oracle Database Appliance bare metal or VM guest systems TFA comes preinstalled and configured under the /opt/oracle/tfa directory. This is true for ODA versions that do not have Oracle Grid Infrastructure 11.2.0.4 installed. If the ODA has Oracle Grid Infrastructure 11.2.0.4 installed then TFA will run from that Grid Infrastructure home directory.

When TFA is shipped with Oracle Grid Infrastructure it is installed and configured through the Oracle Universal Installer, then patched through regular Patch Set Updates, or downloading the latest TFA from My Oracle Support.

For all other systems installation is from a self extracting archive file named installTFALite. The installation requires a JRE version 1.5 to already be installed at the same location on all nodes that are to be added to the TFA configuration (e.g. Oracle RAC cluster nodes). A JRE is provided by Oracle as part of an RDBMS or Grid Infrastructure HOME. To install TFA in this configuration simply download and execute the installation script.

TFA must be installed as root. For automatic cluster wide installation, passwordless ssh as root user is required to distribute the software. Installation will set up and break down ssh user equivalency for the root user as part of this install process so long as the user answers Yes to do this at the prompt on installation. If it is not possible due to security policy to allow passwordless ssh for root even just for the time of the installation then the software must be installed locally on each node individually. The installer may be called through sudo for local only installations if required.

The goal of TFA is zero configuration and as such the installation scans the system to discover the data it needs from the environment. Predominantly it will use Oracle Grid Infrastructure when available to determine the nodes in any cluster and what database homes and databases are installed/running on the system. The user can change the nodes and trace file directories to analyze once the installation is completed.

Upon installation completion a TFAMain JVM will be running on each node in the configuration and will run an inventory process to discover files in the trace directories found. The inventory process determines first and last timestamp as well as the file type for all files in these directories. Any alert type files discovered will then be monitored continuously for significant events and if any of those events occur then TFA can automatically gather relevant diagnostics if configured to do so, though manual collections may be initiated at any time. Note that in this release Alert type files are just CRS, ASM and RDBMS alert logs and automatic diagnostic collections are disabled by default.

If the installer detects that it is on a machine that contains Exadata storage servers then it will discover them and attempt to set up access for them such that TFA can make diagnostic collections from them as required.

4.1. TFA installer options

If TFA needs to be installed manually then there are a number of options that can be used to allow for silent or modified installation.

#./installTFALite -help

Usage for installTFALite

./installTFALite [-local][-deferdiscovery][-tfabase <install dir>][-javahome < JRE Path>][-silent]

-local - Only install on the local node
-deferdiscovery - Discover Oracle trace directories after installation completes
-tfabase - Install into the directory supplied
-javahome - Use this directory for the JRE
-silent - Do not ask any install questions
-debug - Print debug tracing and do not remove TFA_HOME on install failure

Note: Without parameters TFA will take you through an interview process for installation
/tfa will be appended to -tfabase if it is not already there.

- local : This option means that TFA will only be installed on the local node and there will be no requirement for ssh.
- deferdiscovery : TFA will not try to discover all databases and trace directories as part of the installation process but instead will begin the discovery process immediately after installation. This can significantly speed up the installation time when there are many databases. Using this flag will also result in a local only installation.
- tfabase: This is the directory TFA will be installed under. If the directory path given does not end with '/tfa/ then '/tfa' will be appended to the path. If the path given is your Grid Infrastructure home then TFA will place it's repository, log, and database directories in the Grid Home owners ORACLE_BASE directory. If no -tfabase is supplied then TFA will be installed under your current working directory.
- javahome: This is the path to a JRE of version 1.5 or above.
- silent: If the silent flag is provide then no questions will be asked as part of the installation. This requires that a javahome is supplied unless a valid JRE can be found in the shell PATH.
- Debug: By default the TFA install generates minimal logging and cleans itself up on install failure which leaves no trace of the failure. If TFA fails to install then please simply run again with this option to diagnose the problem.

4.2. Automatic cluster wide installation

To run an automatic cluster wide installation of TFA the CRS stack must be up and responding to requests on the node from which TFA is being installed.

The installation could be done silently with a command such as

As the root user run

*#./installTFALite -tfabase /u01/app/tfa -javahome /*usr/java6_64/ *-silent*

In this case passwordless ssh would have to have been set up for root to all nodes prior to the installation otherwise the installation would fail.

The same can be achieved through the interview process only using this method it is possible to set up ssh just for the duration of the installation

As the root user run

#./installTFALite
Starting TFA installation

- By default the install will create a directory tfa_home under your current directory but the user may optionally specify any location.

  Enter a location for installing TFA (/tfa will be appended if not supplied)
  [/u01/app/tfa]:

- The Installer will then request the path to JAVA_HOME containing Java 1.5 or later. This location must be common across all nodes..

  Enter a Java Home that contains Java 1.5 or later :

  Alternatively the JRE location can be supplied prior to starting the installation.

  export JAVA_HOME=/usr/java6_64/
  #./installTFALite

  Note: Oracle RAC Grid Infrastructure home normally includes a 1.5 JRE

- Next choose a cluster wide installation.

  Would you like to do a [L]ocal only or [C]lusterwide installation ? [L|l|C|c] [C] :C

  Running Auto Setup for TFA as user root...
  The following installation requires temporary use of SSH.
  if SSH is not configured already then we will remove SSH when complete.
  Do you wish to Continue ? [Y|y|N|n] [Y]

- Note that the default here is 'Y' so TFA will be setting up and breaking down passwordless ssh user equivalency for root if it was not previously configured. If you specify 'N' here then the install will stop. You may be required to enter the root password multiple times for each node to complete this process, if you do not want to setup passwordless ssh for root then you should complete a local only install on each node..

- TFA will then attempt to discover all the nodes and all the resources required to monitor. If CRS is not running at the time of TFA install then the cluster nodes will have to be supplied manually.

  TFA Will be Installed on the Following Nodes
  ++

  Install Nodes
  ==========
  Node1
  Node2
  Do you wish to make changes to the Node List ? [Y/y/N/n] [N]

- If you wish to change the discovered node list then the installer will exit after printing the following message.

  Please restart the installation to change the node list
  You can either:-
  1) Enter all required nodes into a file tfa_nodemap in the directory this is executed from
  2) Ensure Oracle GI/CRS is running so we can discover the cluster nodes

- Next the trace file directories are determined and listed
- Once you accept the discovered directory list the installation will continue for all nodes. Upon completion an installation summary is printed and the tfactl syntax is shown.

- You should note that by default in TFA non root users can perform many operations using TFA (eg., perform diagnostic collections) as described in Section 6: Managing TFA with tfactl. So on installation you will see :-

  Enabling Access to Non-root Users on node1...

  Adding default users and groups to TFA Access list...

  If you wish to prevent non root users from performing those TFA operations then this can be done using '*tfactl access*' once the installation completes.

- As the last part of installation TFA will inventory the state of all trace files in the discovered directories. This inventory includes name, type, first and last timestamp etc. You can see when that has completed as follows:

  ./tfa_home/bin/tfactl print actions

HOST	*START TIME*	*END TIME*	*ACTION*	*STATUS*	*COMMENTS*
Node1	*Jan 01 05:30:01 PST*	*Jan 01 05:40:25 PST*	*Run inventory*	*COMPLETE*	*Requested in all nodes*
Node2	*Jan 01 05:30:01*	*Jan 01 05:40:25*	*Run inventory*	*COMPLETE*	

	PST	*PST*			

4.3. **Single Node (Local) Installation.**

Single node installation is required when the Oracle clusterware is not available or if you do not have or wish to set up passwordless ssh user equivalence for the root user to allow for deployment across nodes.

The installation could be done silently with a command such as

As the root user run
*#./installTFALite -tfabase /tmp -javahome /*usr/java6_64/ *-silent -local*

The same can be achieved through the interview process

As the root user run
#./installTFALite
Starting TFA installation
CRS is not up
Enter a location for installing TFA [/tmp]:
Enter a Java Home that contains Java 1.5 or later :

Running Auto Setup for TFA as user root...

Would you like to do a [L]ocal only or [C]lusterwide installation ? [L|l|C|c] [C] : L
Installing TFA at /tmp in all hosts
Discovering Nodes and Oracle resources
Checking whether CRS is up and running

Checking Status of Oracle Software Stack - Clusterware, ASM, RDBMS

. . . . . .

TFA Will be Installed on Node1

Note that it is not possible to change the node list for a local only install.

After TFA is installed and running on all of the required nodes if Grid Infrastructure is available on all nodes then TFA will automatically build the TFA configuration for your cluster. When Grid Infrastructure is not available you must complete 'tfactl add node' for each node in the TFA cluster to complete configuration.

When adding a new node manually the node to be added must have the correct ssl certificates copied to it from an existing node in the configuration.

The following files must be copied from the exiting configuration node to the node to be added, be owned by root on the machine to be added and have '700' permissions.

tfa_home/server.jks
tfa_home/client.jks
tfa_home/internal/ssl.properties

Once this is completed and TFA restarted on the node to be added an authorization key must be set in each new node to authenticate that the node can be added to the TFA cluster. When a new node is added to an existing cluster the add node command must be run on one of the existing nodes. In the case of 2 nodes currently not members of any cluster the add node can be run from either node.

To check if TFA is running use tfactl on each node.

#./tfactl print status

Host	*Status of TFA*	*PID*	*Port*	*Version*	*BuildID*	*Inventory Status*
node1	*RUNNING*	*23396*	*5000*	*12.1.2.7.0*	*12127020160303*	*COMPLETE*

To add the remote hosts you must get the authorization key from the existing node and set it in the new node to be added as explained when you run the command.

#./tfactl host add node2
Failed to add host: node2 as the TFA cookies do not match.

To add the host successfully, try the following steps:
1. Get the cookie in node1 using:
./tfa_home/bin/tfactl print cookie
2. Set the cookie copied from Step 1 in node 2 using:
./tfa_home/bin/tfactl set cookie=<COOKIE>
3. After Step 2, add host again:
./tfa_home/bin/tfactl host add node2

The add node command should now succeed normally.

#./tfactl host add node2
Successfully added host: node2

To Check the configured hosts after adding the node

#./tfactl print hosts

Host Name : node1
Host Name : node2

Note: The addition of a new host from any existing host will synchronize the host list across all hosts in the TFA cluster. After installation an auto discovery process will run periodically out of TFAMain so if the Oracle clusterware stack is started at some point later any previously missed and known directories will be added to the directory list.

4.4. Patching TFA Collector

For non ODA or Grid Infrastructure installations the TFA Collector install scripts detect if TFA is already configured and patches the installation, so all that is required for a new version of TFA is to run the install scripts as previously described.

When patching a pre 2.5.1.5 TFA release to TFA Collector 2.5.1.6 or above on a cluster ssh root user equivalence is used to deploy the changes and restart TFA. If ssh root user equivalence is not configured then the root password for remote nodes will be requested a number of times. Another option for patching is to install the patch on each node separately by passing the –local flag to the installer. Once TFA Collector 2.5.1.6 or above has been installed patching is accomplished by running the install script for the new version on one node in the TFA configuration and the other nodes will be patched automatically with no requirement for passwordless ssh root user equivalence.

4.5. De-installing TFA Collector

TFA should not be removed from ODA systems, and for Grid Infrastructure installations will be removed when a node is deinstalled or deconfigured. For other configurations the following applies:-

4.5.1. Clusterwide uninstall

To remove the TFA Collector from all nodes the uninstalltfa script should be executed from any node in the cluster from that nodes TFA_HOME.

#tfa_home/bin/uninstalltfa

With no flags provided this script will stop TFA on all nodes TFA is installed on and remove the software. Where passwordless ssh is not available for the root user the password will be requested each time a remote task needs to run or the uninstall must be run on each node separately.

If you specify the silent flag as below and passwordless ssh for the root user is not available then a local only uninstall will be run.

#./tfa_home/bin/uninstalltfa.sh –silent

4.5.2. Local uninstall

To remove TFA Collector from just one node of the cluster you can use the ‘tfactl uninstall ‘ command.

#./tfactl uninstall

Controlling TFA

5. Starting and Stopping TFA

TFA runs out of init on Unix systems or init/upstart/systemd on Linux systems so that it will be started automatically whenever a node is started. The init control file is /etc/init.d/init.tfa (platform dependent)

The preferred method to start and stop TFA is to use the 'tfactl' command as described later.

TFA can also be started and stopped using the following arguments to /etc/init.d/init.tfa

```
# /etc/init.d/init.tfa -h

Usage: /etc/init.d/init.tfa {stop|start|shutdown|restart}
```

- start – Ensures all init or upstart files are in place and Starts TFAMain process
- stop – Stops TFAMain process
- restart – Stops and then starts TFAMain process
- shutdown – stops TFAMain process and removes entries from inittab and/or upstart configuration

If the TFAMain process ever fails then it will be automatically restarted by init/upstart/systemd.

6. Managing TFA with tfactl

To Manage TFA tfactl can be run from the tfa_base/bin directory and can be run as the root user, or any other user authorized to run TFA. Non root users will only able to run commands that complete diagnostic collections or make directories they own available for collection. If user is given sudo access to run 'tfactl' then they will have access to all commands.

The syntax for tfactl is continually revised and can be seen using
tfa_home/bin/tfactl –help (or use –h instead of –help in all cases).

Root user:-

```
# /sharedfs/tfa/bin/tfactl -h
Usage : / sharedfs /tfa/node1/tfa_home/bin/tfactl <command> [options]
<command> =
        start           Starts TFA
        stop            Stops TFA
        disable         Disable TFA Autostart
        enable          Enable TFA Autostart
        print           Print requested details
        access          Add or Remove or List TFA Users and Groups
```

purge	*Delete collections from TFA repository*
directory	*Add or Remove or Modify directory in TFA*
host	*Add or Remove host in TFA*
diagcollect	*Collect logs from across nodes in cluster*
collection	*Manage TFA Collections*
analyze	*List events summary and search strings in alert logs.*
set	*Turn ON/OFF or Modify various TFA features*
toolstatus	*Prints the status of TFA Support Tools*
run <tool>	*Run the desired support tool*
start <tool>	*Starts the desired support tool*
stop <tool>	*Stops the desired support tool*
restart <tool>	*Restarts the desired support tool*
uninstall	*Uninstall TFA from this node*
diagnosetfa	*Collect TFA Diagnostics*

For help with a command: /sharedfs/tfa/bin/tfactl <command> -help

Non Root user:-

 # /sharedfs/tfa/bin/tfactl -h
Usage : / sharedfs /tfa/node1/tfa_home/bin/tfactl <command> [options]
<command> =

print	*Print requested details*
analyze	*List events summary and search strings in alert logs.*
diagcollect	*Collect logs from across nodes in cluster*
directory	*Add or Remove or Modify directory in TFA.(Only directories that the non root user has access to)*
toolstatus	*Prints the status of TFA Support Tools*
run <tool>	*Run the desired support tool*
start <tool>	*Starts the desired support tool*
stop <tool>	*Stops the desired support tool*
restart <tool>	*Restarts the desired support tool*

For help with a command: /sharedfs/tfa/bin/tfactl <command> -help

Note: ‘tfactl ‘ also has a set of commands for managing Exadata storage server collections. These are documented in Section 0 Using TFA to collect data from Exadata Storage Servers.

6.1. tfactl start

This command will start the TFA daemon.

6.2. tfactl stop

This command will stop the TFA daemon.

6.3. tfactl enable

This command will enable the autorestart of TFA on failure/reboot

6.4. tfactl disable

This command will stop any running TFA daemon and disable autorestart of TFA.

6.5. tfactl print

The print command allows you to print information from the BDB.

tfactl print -h
Usage: /sharedfs /tfa//bin/tfactl print
[status|config|directories|hosts|actions|repository|cookie]

Prints requested details.

Options:
status Print status of TFA across all nodes in cluster
config Print current tfa config settings
directories Print all the directories to Inventory
hosts Print all the Hosts in the Configuration
actions Print all the Actions requested and their status
repository Print the zip file repository information

6.6. tfactl print status

To check if TFA is running use tfactl on each node.

./tfactl print status

Host	*Status*	*PID*	*Port*	*Version*	*Build ID*	*Inventory Status*
node1	*RUNNING*	*23396*	*5000*	*12.1.2.7.0*	*12127020160303214632*	*RUNNING*
node2	*RUNNING*	*23456*	*5000*	*12.1.2.7.0*	*12127020160303214632*	*RUNNING*

6.7. tfactl print config

Printing the configuration shows the current local state of TFA as follows.

Configuration parameter	Value
TFA Version	12.1.2.7.0
Automatic diagnostic collection	OFF
Alert Log Scan	ON
Trimming of files during diagcollection	ON
Repository current size (MB) in node1	526
Repository maximum size (MB) in node1	10240
Inventory Trace Level	1
Collection Trace level	1
Scan Trace level	1

Other Trace level	1
Max Size of TFA Log (MB)	50
Max Number of TFA Logs	10
Max Size of Core File (MB)	20
Max Collection Size of Core Files (MB)	200
Automatic Purging	ON
Minimum Age of Collections to Purge (Hours)	12
Minimum Space Free to enable Alert Log Scan (MB)	500

- Automatic diagnostic collection : When ON (default OFF) if scanning an Alert log then finding specific strings in logs can trigger diagnostic collections.
- Alert Log Scan can be turned off if you do no wish to scan the alert log for errors. Turning this off may cause larger alert log collections and is not compatible with having Automatic Collection turned on.
- Trimming of files during diagcollection: Determines if we will trim large files to only contain data that is within specified time ranges. When this is OFF no trimming of trace files will occur for any diagnostic collection.
- Repository current size in MB : How much space in the repository is currently used.
- Repository maximum size in MB : The max size of stored data in the repository. Initially the max size is set to the smaller of 10GB or 50% of free space in the file system.
- Trace Level : Current JVM trace level . 1 is default, 2 is verbose , 3 is very verbose. Trace level can be dynamically set for the running JVM. Increasing this level significantly impacts the performance of TFA so should only be done at the request of Oracle Support.
- Core file collections are not enabled in 12.1.2.6.0. These values are for future use.
- Automatic purging of TFA collections is enabled by default in 12.1.2.3.0. and above. TFA collections can be purged if they are more than 'Minimum Age of Collections to Purge' hours old, and the repository space is exhausted. This value is configurable using '*tfactl set minagetopurge=X'* where X is a value in hours.
- Minimum Space free to enable alert Log Scan ensures TFA is a free space limit that will temporarily suspend Alert log scanning until space becomes free. TFA will not store alert log events if space on the filesystem used for the metadata database is low.

6.8. **tfactl print directories**

Lists all the directories that will be inventoried for trace or log file data. Also shows the database/ASM and instance the trace directory is allocated to where the directory is for an RDBMS or ASM. Shows SYS/SYS, or CRS/CRS as appropriate for non RDBMS/ASM files. Eg.

Node1			
Trace Directory	Component	Permission	Added by
/cloudfs/ /diag/rdbms/mydb/MYDB1/trace	RDBMS: mydb,mydb1	public	root

6.9. tfactl print hosts

Lists the hosts that are part of the TFA cluster and will receive clusterwide commands. For Example:

Host Name : Node1
Host Name : Node2

6.10. tfactl print actions

Lists all the actions submitted to TFA such as run inventory. Any action that gets created as part of a request from tfactl or a remote TFA should be visible with that actions outcome. By default print actions only shows actions that are running or completed in the last hour.

#tfactl print actions -help
Usage: /sharedfs/tfa/bin/tfactl print actions [-status <status> | -since <n><h|d>]
Print TFA Actions.

Options:

-status <status>	*Action status can be one or more of COMPLETE,RUNNING,FAILED,REQUESTED Specify comma separated list of statuses*
-since <n><h|d>	*Actions from past 'n' [d]ays or 'n' [h]our*

6.11. tfactl print collections

A simple list of all the collections with statistics . EG.

CollectionId: 20160223082908mynode1
TFAVersion: 12.1.2.7.0
TFABuildId: 12127020160303214632
Manual Collection: true
RequestUser: root
MasterHost: mynode1
NodeList: []
Collection Name: /u01/app/oragrid/tfa/repository/t5
ZipFileName: tfa_Tue_Sep_23_08_29_07_UTC_2014.zip
DiagcollectLog: diagcollect_20140923082908_rws1270037.log
ComponentList: [crs, os, chmos]
StartTime: Sun Sep 21 08:29:17 UTC 2014
EndTime: Tue Sep 23 08:29:17 UTC 2014
FilesInRange: 444
SizeOfFilesInRange: 1050977
SizeOfExtraFiles: 538614
SizeOfFilesBeforeZip: 1546369
ZipFileSize: 59200
NumTrimmedFiles: 8
TrimSavings: 43222

CollectionTime: 297
Collection Score: 0

6.12. tfactl print repository

Reports the repository directory and the current usage for all nodes. For Example:

Node1	
Repository Parameter	Value
Location	/opt/oracle/tfa/tfa_home/repository
Maximum Size (MB)	10240
Current Size (MB)	526
Free Size (MB)	10240
Status	OPEN

6.13. tfactl access

Use 'tfactl access' to manage the use of TFA by non root users. See 9.

6.14. tfactl purge

tfactl purge will delete collections from the TFA repository that are older than the supplied time.

#tfactl purge –h
Usage: /sharedfs/tfa/bin/tfactl purge -older x[h|d] [-force]

Remove file(s) from repository that are older than the time specified.

Examples:
/sharedfs/tfa/bin/tfactl -older 30d - To remove file(s)older than 30 days.
/sharedfs/tfa/bin/tfactl -older 10h - To remove file(s)older than 10 hours

Use –force option to disable file deletion confirmation.

6.15. tfactl directory

The *tfactl directory* command set are used to add a directory to or remove a directory from the list of directories that will have their trace or log files analyzed. The command can also be used to change the directory permissions and collection policy. When a directory is added by auto discovery it will get added as public and that means any file in that directory can be collected by any user that can run tfactl diagcollect. This is important when non root users or sudo users run TFA collections. If a directory is marked as private then TFA determines the real user that is calling tfactl and verifies that the user has permissions to see the files in the directory before allowing files to be collected. **Note:** that a user can only ever add a directory to TFA or modify a directory that they own and also that TFA auto collections when

configured run as root so will always collect all available files. Manual addition by default makes directories private within TFA.

#tfactl directory -h
*Usage: /sharedfs/tfa/bin/tfactl directory **add** <dir>*
[- public] [-exclusions | -noexclusions | -collectall] [-node <all | n1,n2..>]

*Usage: /sharedfs/tfa/bin/tfactl directory **remove** <dir> [-node <all | n1,n2..>]*

*Usage: /sharedfs/tfa/bin/tfactl directory **modify** <dir>*
[-private | -public] [-exclusions | -noexclusions | -collectall]

Add or Remove or Modify directory in TFA

-private This flag implies that if the person executing this command does not have privileges to see files in those underlying directories then they will not be able to collect those files.
-public The Oracle directories are public as this is the purpose of the tool to allow for collection of these files.
-exclusions This flag implies that files in this directory are eligible for collection if they satisfy type, name and time range restrictions.
-noexclusions This flag implies that any files in this directory are eligible for collection if they satisfy time range restictions.
-collectall This flag implies that any files in this directory are eligible for collection irrespective of type and time range when "-collectalldirs" flag is specified in 'tfactl diagcollect'.

Examples:
/sharedfs/tfa/bin/tfactl directory add /u01/app/grid/diag/asm/+ASM1/trace
/sharedfs/tfa/bin/tfactl directory remove /u01/app/grid/diag/asm/+ASM1/trace
/sharedfs/tfa/bin/tfactl directory remove /u01/app/grid/diag/asm/+ASM1/trace -node all
/sharedfs/tfa/bin/tfactl directory modify /u01/app/grid/diag/asm/+ASM1/trace -private
/sharedfs/tfa/bin/tfactl directory modify /u01/app/grid/diag/asm/+ASM1/trace -noexclusions
/sharedfs/tfa/bin/tfactl directory modify /u01/app/grid/diag/asm/+ASM1/trace -private -noexclusions
/sharedfs/tfa/bin/tfactl directory modify /tmp/for_support -private -collectall

All trace directory names need to be added to the BDB so TFA will know to do a file inventory on that directory. The install process finds some directories but if new or undiscovered directories are required then these can be added manually through *tfactl*. When adding through *tfactl* the code will try to determine whether the directory is for database or CRS or O/S logs etc and which database / instance etc. If TFA cannot perform that operation successfully then it will give an error and request that information be entered.

#tfactl directory add /tmp
Failed to add directory to TFA. Unable to determine parameters for directory: /tmp
Please enter component for this Directory
[RDBMS|CRS|ASM|INSTALL|OS|CFGTOOLS|TNS|DBWLM|ACFS|ALL] : RDBMS
Please enter database name for this Directory :MYDB
Please enter instance name for this Directory :MYDB1

Note: For OS, CRS,CFGTOOLS,ACFS,ALL or INSTALL files only the component is requested, and for ASM only the instance is created. No verification is done for these entries so care should be taken when entering this data.

6.16. tfactl host

Use the tfactl host command to add hosts to or remove hosts from the TFA configuration.

#tfactl host -h

Usage: /sharedfs/tfa/bin/tfactl host [add <hostname> | remove <hostname>]

Add or Remove a host in TFA

Examples:
/sharedfs/tfa/bin/tfactl host add myhost.domain.com
/sharedfs/tfa/bin/tfactl host remove myhost.domain.com

Using the host add command simply tells the local TFA about other nodes on your network. When a host is added TFA contacts that host and if TFA is running on that host then both hosts synchronize their host list. TFA authenticates that a host can be added using cookies. If the host to be added does not have the correct cookie then that must be retrieved from an existing host in the cluster and set at the host to be added.

./tfactl host add node2
Failed to add host: node2 as the TFA cookies do not match.
To add the host successfully, try the following steps:
1. Get the cookie in node1 using:
./tfa_home/bin/tfactl print cookie
2. Set the cookie from Step 1 in node using:
./tfa_home/bin/tfactl set cookie=<COOKIE>
3. After Step 2, add host again:
./tfa_home/bin/tfactl host add node2

The add node command should now succeed normally.

./tfactl host add node2
Successfully added host: node2

Once the host is added all cluster wide commands will activate on all nodes registered in the BDB.

6.17. tfactl collection

Currently this command only has one option. Use ‘stop’ to stop a TFA collection when it is running. The stop options requires a collection id that can be found using ‘tfactl print collections’.

./tfactl collection -h
Usage./tfactl collection [stop <collectionid>]

Manages TFA Collections.

Options:
stop *Stop the requested collection*

6.18. **tfactl set**

The set command allows us to adjust the way TFAMain is running. This command makes the changes that you can see when doing *a tfactl print config*. By default changes are made locally so if the change is required on all nodes you must supply the '-c' flag.

#/tfactl set -h
Usage: /sharedfs/tfa//bin/tfactl set [
autodiagcollect=<ON | OFF> | trimfiles=<ON | OFF> |
tracelevel=<COLLECT|SCAN|INVENTORY|OTHER>:<1|2|3|4> | reposizeMB=<n> |
repositorydir=<dir> | logsize=<n> [-local] | logcount=<n> [-local]] [-c]

Turn ON/OFF or Modify various TFA features

autodiagcollect	*allow automatic diagnostic collection*
autopurge	*allow automatic purging of collections when less space is observed in repository (default ON)*
minagetopurge	*the minimum age for a collection in hours before it is considered for purging.*
repositorydir	*Set the zip file repository*
reposizeMB	*Set the maximum zip file repository size*
trimfiles	*allow trimming of files during diagcollection (default ON)*
tracelevel	*control the trace level of log files.(default 1 for all facilities)*
logsize=<n>	*set the maximum size of each TFA log to <n>MB (default 50 MB)*
logcount=<n>	*set the maximum number of TFA logs to <n> (default 10)*
-c	*Make the change apply to all TFA nodes.*

Examples:
/sharedfs/tfa/bin/tfactl set autodiagcollect=ON
/sharedfs/tfa/bin/tfactl set tracelevel=INVENORY:4
/sharedfs/tfa/bin/tfactl set repositorydir=/sharedfs/tfa/repository
/sharedfs/tfa/bin/tfactl set reposizeMB =10240

- autodiagcollect : means that the log scanner can do diagnostic collection when it finds a search string event.
- trimfiles: means that files will be trimmed to only have what we believe to be relevant data when diagnostic collection is done as part of a scan. Note that trimming for diagnostic collection when done through tfactl diagcollect will be determined by the parameters given.
- tracelevel: changing the trace level should only be done at the request of Oracle support as extra tracing can affect the performance and resource consumption of TFA . Trace levels

can be set for certain operations INVENTORY:n, SCAN:n, COLLECT:n, OTHER:n. Where n is 1 to 4 and OTHER handles all messages not covered by the first three.

- logsize: TFA will rotate it's trace logs and when they reach this size in MB.
- logcount: TFA will keep this many rotated trace logs of the above size.

6.19. tfactl toolstatus

From TFA 12.1.2.3.0 when downloaded from MOS TFA will also deliver the contents of the support tools bundle, previously available from MOS via RAC and DB Support Tools Bundle (Doc ID 1594347.1). The tfactl toolstatus command shows the current state of the support tools and other tools delivered with TFA.

#/tfactl toolstatus

```
.-----------------------------------------------------------.
|        External Support Tools                             |
+---------------+------------------+------------------------+
| Host          | Tool             | Status                 |
+---------------+------------------+------------------------+
| node1         | alertsummary     | DEPLOYED               |
| node1         | ls               | DEPLOYED               |
| node1         | pstack           | DEPLOYED               |
| node1         | orachk           | DEPLOYED               |
| node1         | sqlt             | DEPLOYED               |
| node1         | grep             | DEPLOYED               |
| node1         | summary          | DEPLOYED               |
| node1         | prw              | NOT RUNNING            |
| node1         | vi               | DEPLOYED               |
| node1         | tail             | DEPLOYED               |
| node1         | param            | DEPLOYED               |
| node1         | dbglevel         | DEPLOYED               |
| node1         | darda            | DEPLOYED               |
| node1         | history          | DEPLOYED               |
| node1         | oratop           | DEPLOYED               |
| node1         | oswbb            | DEPLOYED               |
| node1         | changes          | DEPLOYED               |
| node1         | events           | DEPLOYED               |
| node1         | ps               | DEPLOYED               |
'---------------+------------------+------------------------'
```

TFA tools are alertsummary, ls, pstack, grep, summary,vi, tail, param, dbglevel, history, changes, events, ps. The delivered Support Tools are Diagnostic Assistant/RDA, oratop, orachk, OSWatcher black box, SQLT, Procwatcher.
The TFA bundled tools are described fully in Chapter 11.

6.20. tfactl uninstall

'tfactl uninstall' will remove TFA from the local node only.

6.21. tfactl diagnosetfa

'tfactl diagnosetfa' collects various TFA diagnostic data from the local node to help diagnose issues with TFA itself.

Diagnostic collection with TFA

7. On Demand Diagnostic Collection

The diagnostic collection module takes a number of different parameters to determine how large or detailed the required evidence set is. It is possible to give a specific time of an incident or a time range for data to be collected, as well as determine if whole files that have relevant data should be collected or just a time slice of data from those files (only collect the section of the file that applies to a time range).

Note: With no flags specified the default is for *tfactl diagcollect* to collect files from all nodes for all components where the file has been updated in the last 4 hours and will trim files it considers excessive. This collection could be large where TFA is monitoring trace files for many databases, so care should be taken when using defaults, for example, in a large database consolidation environment. In cases like that care should be taken to specify a specific database or databases in order to avoid collecting large amounts of diagnostics that might not be relevant or which would make the collection unnecessarily large and time consuming. The goal should be to make a precisely targeted diagnostic collection. The defaults are only for convenience on relatively small systems If an incident occurred prior to the default 4 hour period then the parameters documented below should be used to target the correct data collection.

The following parameters do not show options for collections on Exadata storage servers. See Section 0. to review Exadata storage server collections.

./tfactl diagcollect -h
Usage: /sharedfs/tfa/bin/tfactl diagcollect [-all | -database <all|d1,d2..> |
-asm | -crs | -dbwlm | -acfs | -os | -install | -cfgtools
-ashtext | –ashhtml | -awrtext | -awrhtml | -chmos | -srdc]
[-node <all | local | n1,n2,..>] [-tag <description>] [-z <filename>]
[-since <n><h|d>| -from <time> -to <time> | -for <time>] [-nocopy]
[-notrim][-nomonitor] [-collectalldirs]

Options:

-all	Collect all logs (If no time is given for collection then files for the last 4 hours will be collected) This is the default option.
-oda	Collect ODA/OS logs
-odastorage	Collect ODA Storage logs and Data
-crs	Collect CRS logs
-dbwlm	Collect DBWLM logs
-acfs	Collect ACFS logs
-asm	Collect ASM logs
-database	Collect database logs from databases specified
-install	Collect Oracle Installation related files
-cfgtools	Collect CFGTOOLS logs
-os	Collect OS files such as /var/log/messages
-ashhtml	Collect Generate ASH HTML Report (requires –database)
-ashtext	Collect Generate ASH TEXT Report (requires –database)
-zdlra	Collect Zero Data Loss Recovery Appliance specific logs and data.

-awrhtml	Collect html awr reports for all snapshots in the given period – requires database.
-awrtext	Collect text awr reports for all snapshots in the given period – requires database.
-node	Specify comma separated list of host names for collection
-nocopy	Does not copy back the zip files to initiating node from all nodes
-notrim	Does not trim the files collected
-nomonitor	This option is used to submit the diagcollection as a background Process.
-collectalldirs	Collect all files from a directory marked "Collect All"
-since <n><h|d>	Files from past 'n' [d]ays or 'n' [h]ours
-from "MMM/dd/yyyy hh:mm:ss"	From <time>
-to "MMM/dd/yyyy hh:mm:ss"	To <time>
-for "MMM/dd/yyyy"	For <date>.
-tag <tagname>	The files will be collected into tagname directory inside repository
-z <zipname>	The files will be collected into tagname directory with the specified zipname

- -all: This flag means that all files in the inventory will be collected (default) . If no time is given for collection then files for the last 4 hours will be collected.
- -crs: tells diagcollect to collect files that are crs log or trace files.
- -asm: tells diagcollect to collect asm logs.
- -dbwlm: tells diagcollect to collect dbwlm logs.
- -acfs: tells diagcollect to collect acfs, asm and crs logs.
- -database: tells diagcollect to collect database logs from the databases specified (all, or a comma separated list)
- -os: Collect O/S logs and CHMOS/OSW data.
- -install: Collect install logs and trace files.
- -node: provides a list of nodes that are to complete diagnostic collection. Default is all.
- -nocopy: This stops the resultant trace file collection being copied back to the initiating node. The file will simply remain in the executing nodes TFA repository.
- -nomonitor: Do not display progress to the calling terminal.
- -collectalldirs: With this flag any directory marked as “Collect All” using ‘tfactl directory’ command will be collected in its entirety.
- -since: Collect all files that have relevant data for the past X hours or days. By Default also trim files that are large and have many days data to show only the requested timespan.
- -from: Collect all files that have relevant data after a given time, and trim all data before this time where files are large. Must be used with –to.
- -to : Collect all files that have relevant data after a given date and trim all data after this time where files are large. Must be used with –from.
- -for: Collect all files that have relevant data for the time give. ie. have a first and last timestamp that contains the given time. No data trimming is done for this option.
- -z: Supply an output file name. File will always be placed in the TFA repository directory.
- -tag: Supply a tag that is also used to create a subdirectory for the resulting collection in the TFA repository.

Examples:

tfactl diagcollect

Trim and Zip all files updated in the last 4 hours as well as chmos/osw data from across the cluster and collect at the initiating node.

Note: This collection could be larger than required but is the simplest way to capture diagnostics if an issue has recently occurred.

#tfactl diagcollect –all –since 8h

Trim and Zip all files updated in the last 8 hours as well as chmos/osw data from across the cluster and collect at the initiating node.

#tfactl diagcollect -database hrdb,fdb -since 1d -z foo

Trim and Zip all files from databases hrdb & fdb updated in the last 1 day and collect at the initiating node

#tfactl diagcollect –crs -os -node node1,node2 -since 6h

Trim and Zip all crs files, o/s logs and chmos/osw data from node1 & node2 updated in the last 6 hours and collect at the initiating node

#tfactl diagcollect -asm -node node1 -from Mar/4/2013 -to "Mar/5/2013 21:00:00"

Trim and Zip all ASM logs from node1 updated between from and to time and collect at the initiating node.

tfactl diagcollect -for "Mar/2/2013"

Trim and Zip all log files updated on "Mar/2/2013" and collect at the initiating node.

#tfactl diagcollect -for "Mar/2/2013 21:00:00"

Trim and Zip all log files updated from 09:00 on March 2 to 09:00 on March 3 (i.e. 12 hours before and after the time given) and collect at the initiating node.

8. Automatic Diagnostic Collection.

After the initial inventory is completed by TFA all files that are determined to be Alert Logs are tailed so that TFA can take action when certain messages are seen (see Appendix B. Scan Events). By default these logs are RDBMS alert logs, ASM alert logs and CRS alert logs. When specific strings are seen in the logs on information on the strings is saved to the BDB and then some automatic diagnostic collection

may take place. Exactly what is collected is dependent on the string and in which alert it is found but potentially trimmed versions of all O/S, CRS, ASM and RDBMS logs could be collected for each string. Clearly this type of operation could be called many times a second if not controlled so there can never be a collection generated at less than 5 minute intervals due to the TFA implementation of flood control. Also note that if the TFA repository reaches its repository max size then no auto collection will take place.

8.1. Managing Automatic diagnostic collection

To manage the automatic diagnostic collection you can use:-

#tfactl set autodiagcollect=<ON|OFF> [-c]

When set to OFF (default) automatic diagnostic collection will be disabled. If ON diagnostics will be collected when the required search strings (see Appendix) are detected whilst scanning the alert logs. To set automatic collection for all nodes of the TFA cluster the '-c' flag must be used.

The trimming of files collected by Automatic Diagnostic Collection can be controlled by:-

#tfactl set trimfiles=<ON|OFF>[-c]

When ON (default) files will be trimmed to only include data around the time of the event, when OFF any file that was written to at the time of the event will be collected in its entirety.
To set trimfiles for all nodes of the TFA cluster the '-c' flag must be used.

9. TFA Support for Non Root Users

In TFA some non root users can run a subset of tfactl commands. This allows non root users to have controlled access to TFA and to run diagnostic collections, but root access is still required to install and administer TFA. Non root Users and Groups can be controlled using '*tfactl access'*

The range of commands is restricted for non root users.

tfactl -help: (For Non-Root Users)

Usage : /u01/tfa/bin/tfactl <command> [options]

<command> =

print	*Print requested details*
analyze	*List events summary and search strings in alert logs.*
diagcollect	*Collect logs from across nodes in cluster*
directory	*Add or Remove or Modify directory in TFA*

For help with a command: /u01/tfa/bin/tfactl <command> -help

9.1. Managing Non-Root Users using tfactl

TFA Non-root users can be managed and controlled by the root user using '*tfactl access'*. Users and groups can be added or removed depending upon business requirements.
Note: When tfactl access is first enabled with installation by default all Oracle Home owners and OS DBA groups will be added to the list of authorized users/groups.

/u01/tfa/bin/tfactl access –help

Usage: /u01/tfa/bin/tfactl access access [lsusers |
add -user <user_name> [-group <group_name>] [-local] |
remove -user <user_name> [-group <group_name>] [-all] [-local] |
block -user <user_name>[-local] |
unblock -user <user_name> [-local] |
enable [-local] |
disable [-local] |
reset [-local] |
removeall [-local]

Add or Remove or List TFA Users and Groups

Options:

lsusers	*List all the TFA Users and Groups*
enable	*Enable TFA access for Non-root users*
disable	*Disable TFA access for Non root users*
add	*Add user or a group to TFA access list*
remove	*Remove user or a group from TFA access List*
block	*Block TFA Access for Non-root User*
unblock	*Allow TFA Access for Non-root who was blocked before*
reset	*Reset to default TFA Users and Groups*
removeall	*Remove All TFA Users and Groups*

Examples:
/u01/app/tfa/bin/tfactl access add -user abc

User 'abc' is able to access TFA accross cluster

/u01/app/tfa/bin/tfactl access add -group xyz -local
All members of group 'xyz' will be able to access TFA on localhost

/u01/app/tfa/bin/tfactl access remove -user abc
User 'abc' will not be to access TFA

/u01/app/tfa/bin/tfactl access block -user xyz
Access to user 'xyz' will be blocked

/u01/app/tfa/bin/tfactl access removeall
All TFA Users and groups will be removed

9.2. tfactl access lsusers

The access lsusers command will show the users and groups that have access to TFA, when Non-Root user access is enabled. By default TFA we will add Oracle Owner, DBA Group and ASM Group to TFA Access Manager List while installing or upgrading TFA.

/u01/tfa/bin/tfactl access lsusers

```
.---------------------------------.
|            TFA Users            |
+-----------+-----------+---------+
| User Name | User Type | Status  |
+-----------+-----------+---------+
| oinstall  | GROUP     | Allowed |
| asmadmin  | GROUP     | Allowed |
| oracle    | USER      | Allowed |
'-----------+-----------+---------'
```

9.3. tfactl access enable

Enable's TFA non root user access for any users that have been assigned access to TFA. Use the –local flag to only change setting on the local node.

9.4. tfactl access disable

Disables TFA non root user access for all users but the list of users that were assigned access to TFA is stored in case the non-root access is enabled later. Use the –local flag to only change setting on the local node.

9.5. tfactl access add

Add a user or group to the TFA access list.

9.6. tfactl access remove

Remove a user or group from the TFA access list.

9.7. tfactl acces block

Use this command to specifically block a user even though they may be a member of an allowed group.

9.8. tfactl access unblock

Use this command to unblock a user previously blocked with ‘tfactl access block’

9.9. tfactl access reset

Reset to the default access list which will include all Oracle Home owners and DBA groups.

9.10. tfactl access removeall

Remove all users from the TFA access list (useful if default user access in not wanted)

10. Using TFA to collect data in Exadata storage servers

When TFA detects that the system it is being installed on has Exadata Storage cells it will attempt to configure itself to make cell collections. TFA has to be able to access the cells to make these collections and it does this using ssh as the root user. During installation TFA will test ssh to see if cell access can be made without a password to all cells. If this connection cannot be made to all cells from all compute nodes then you will have to provide the cell password for storage cells access in an Oracle Wallet.

Note: Every compute node must be able to access every cell using the same method (all passwordless ssh or all using saved password) otherwise the cell collection may fail from some nodes.

10.1. Installation with Storage Servers

When TFA is being installed from the command line the installer will detect the storage cells and test connection. If it can connect to all cells then it simply reports the configuration and carries on. If any cells fail connection or require a password to be provided for connection then the installer will ask if cells should be configured.

TFA will configure Storage Cells using SSH Setup:

Unable to determine the Exadata Cells. Do you want to Enter Cells manually. [Y|y|N|n] [Y]:

CELLS: cel01, cel02,cel03

Do you want us to store the Password for Cells in Oracle Wallet: [Y|y|N|n] [Y]: Y

Please Enter Password for Oracle Wallet:
Please Confirm Password for Oracle Wallet:
Is password the same for all Exadata Cells: [Y|y|N|n] [Y]: N
Please Enter Password for CELL [IP: scai02cel01]:
Verifying Password for Cell [IP: scai02cel01]...
Please Enter Password for CELL [IP: scai02cel02]:
Verifying Password for Cell [IP: scai02cel02]...
Password provided for Cell [IP: scai02cel03] is incorrect. Please try again.
Please Enter Password for CELL [IP: scai02cel03]:
Verifying Password for Cell [IP: scai02cel03]...Connect to REMOTE HOST was timed out.

As you can see fro the above if all cells have the same password then it will be asked for only once but if not then you must supply a password for all cells. If we fail to access a cell as above then that cell will remain unconfigured and the configuration will need to be run again at some later stage when the cell is online.

10.2. Managing TFA with Storage Servers

TFA management of storage cells is achieved through the 'tfactl cell' command.

Usage: /u01/app/tfa/bin/tfactl cell [status | config | add walletpassword>| remove walletpassword | invstat | diagstat | configure | deconfig]

Print or Modify various Storage Cell features

Options:
status Print the current status of Storage Cells
config Print the current configuration of Storage Cells
add Add Wallet or Wallet Password for Storage Cells
remove Remove Wallet Password
invstat Print the Inventory Statistics of Storage Cells
diagstat Print the Diagcollection Statistics of Storage cells
configure To Configure Storage Cells
deconfigure To remove all the Storage Cell Configuration

10.2.1. tfactl cell status

This command will show all cells that TFA was able to configure and their current status.

/u01/app/tfa/bin/tfactl cell status

```
.-------------------------------------------------------.
|   | EXADATA CELL | CURRENT STATUS |
+---+--------------+----------------+
| 1 | cel01        | ONLINE         |
| 2 | cel02        | ONLINE         |
'---+--------------+----------------'
```

10.2.2. tfactl cell config

The configuration command shows information on the current configuration of cell collection support.

/u01/app/tfa/bin/tfactl cell config

```
------------------------------------------------------------------------------------ ----------.
|                          Storage Cell Configuration                                          |
+-----------------------------------------+------------------------------------------------------+
| Configuration Parameter                 | Value                                                |
+-----------------------------------------+------------------------------------------------------+
| Exadata Support                         | YES                                                  |
| Configured Storage Cells                | YES                                                  |
| Oracle Wallet Used                      | YES                                                  |
| Oracle Wallet Location                  | /u01/app/tfa/db01/tfa_home/internal/tfawallet        |
| Oracle Wallet Password is with TFA      | YES                                                  |
| Oracle Wallet Password Storage Status | Stored                                               |
'-----------------------------------------+------------------------------------------------------'
```

10.2.3. tfactl cell add walletpassword

TFA uses an Oracle wallet to store cell user passwords. The wallet password is required to access the wallet to get these passwords when TFA runs. Running this

command will store the password securely for use when collections are made. If the password is not stored then it will have to be entered at collection time.

 # /u01/app/tfa/bin/tfactl cell add walletpassword
Please Enter Password for Oracle Wallet:
Oracle Wallet Password is successfully added.

10.2.4. tfactl cell remove walletpassword

Use this command to stop TFA storing the wallet password.

 # /u01/app/tfa/bin/tfactl cell remove walletpassword
Please Enter Password for Oracle Wallet:
Oracle Wallet Password is successfully removed.

10.2.5. tfactl cell invstat

The invstat command shows the status of inventory runs on all the cells

#/u01/app/tfa/bin/tfactl cell invstat

```
.-------------------------------------------------------------------------------------------.
|           Storage Cell Inventory Run Statistics                                           |
+---------------------+-----------------------------+---------------------------+----------+
| STORAGE CELL | LAST RUN STARTED | LAST RUN ENDED  | STATUS       |
+---------------------+-----------------------------+---------------------------+----------+
| cel01                | Feb  3 06:44:20           | Feb  3 06:45:24          | COMPLETE |
| cel02                | Feb  3 06:44:21           | Feb  3 06:45:29          | COMPLETE |
'---------------------+-----------------------------+---------------------------+----------'
```

10.2.6. tfactl cell diagstat

Shows if collections are running on cells or not.

 # /u01/app/tfa/bin/tfactl cell diagstat

```
.-------------------------------------------------.
| Cell Diagcollect Run Statistics                 |
+--------------------+------------------------+
| STORAGE CELL  | CURRENT STATUS  |
+--------------------+------------------------+
| cel01                | RUNNING             |
| cel02                | RUNNING             |
'---------------------+------------------------'
```

10.2.7. tfactl cell configure

'tfactl cell configure' can be used to configure cell collections where this was not completed at installation time, was not completed due to upgrade or following a previous deconfigure. The configuration follows the same path as described in the 10.1.

10.2.8. tfactl cell deconfigure

Use this command to remove support for storage cell collections from all compute nodes.

/u01/app/tfa/bin/tfactl cell deconfigure
Removing Storage Cell Configuration...
Successfully removed Storage Cell Configuration.

10.3. Making Storage Server collections

Storage cell collections must be requested from compute nodes and are in addition to any other collection so all previously documented parameters are valid for use at the same time as a cell collection. To add cell collections use the –cell parameter to '*tfactl diagcollect'*

#/u01/app/tfa/bin/tfactl diagcollect –cell <all|cel01,cel02>

11. Using 'collectall' directories

It is possible to add directories to collections manually as well as mark some directories to allow for extra file types to be collected that may previously have been ignored.

11.1. Default Bucket Directory

At install time a directory is created by TFA called diagnostics_to _collect. This directory will either be under tfa_home for a non GI install or under the Grid Owners ORACLE_BASE/tfa/<node> directory. Any files placed in this directory will be made part of the next manual diagnostic collection that runs on this host (even those initiated from another host). **Once collected any files in this directory are deleted**.

11.2. Collect All Directory

Either at 'tfactl directory add' time or using a modify command it is possible to mark a directory as 'Collect All'. These directories will have all of their contents collected every time a collection is made with the –collectalldirs flag.

11.3. No Exclusions Directories

Either at 'tfactl directory add' time or using a modify command it is possible to mark a directory as 'noexclusions'. When a directory is noexclusions only files that are considered relevant for the specified time period are collected, however no other exclusions are applied to these files, where normally we will not collect certain file types.

12. Using TFA to collect AWR and ASH Reports

When AWR and or ASH reports are needed they can be collected using TFA in the following way.

$ */u01/app/tfa/bin/tfactl diagcollect –awrhtml –database <db name> -from <begin time> -to <end time>*
or
$ */u01/app/tfa/bin/tfactl diagcollect –awrtext –database <db name> -from <begin time> -to <end time>*
or
$ */u01/app/tfa/bin/tfactl diagcollect –ashhtml –database <db name> -from <begin time> -to <end time>*
or
$ */u01/app/tfa/bin/tfactl diagcollect –ashhtml –database <db name> -from <begin time> -to <end time>*

They can also be requested as part of the same collection.

For example:

$ */u01/app/tfa/bin/tfactl diagcollect –awrtext –ashtext –database orcl -from "Dec/25/2015 04:00:00" -to "Dec/25/2014 06:00:00"*

For AWR TFA will login into the specified database and extract the list of AWR snapshots falling between the times specified and render a report in html or text as specified for every snapshot period. The AWR reports created will be collected as part of the diagnostic collection.

For ASH TFA will login into the specified database and extract an ash report for the whole period between the times specified and render a report in html or text as specified. The ASH report created will be collected as part of the diagnostic collection.

If the database is a RAC database AWR and/or ASH reports will be run for each instance of the database on the remote nodes.

13. Using TFA SRDC Driven collections.

SRDC collections allow us to collect for a given incident or error as well as time period. For TFA 12.1.2.7.0 there are initially just 3 SRDC collections, these are ORA-04030, ORA-04031 and basic DB performance. To generate these collections you use the '*tfactl srdc'* command.

Usage: tfactl srdc <MenuName> [-sid <oracle_sid>] [-sr <SR#>] [-db <dbName>]
[-inc_date <YYYY-MM-DD>,-inc_time <HH:MI:SS>,
-perf_base_sd <YYYY-MM-DD> etc]

Options:

sid	ORACLE SID for which you collect data
db	Database for which you collect data
sr	SR number to which collected data will get uploaded
inc_date	Incident date
inc_time	Incidnet time
perf_base_sd	Start date for a good performance period
perf_base_st	Start time for a good performance period
perf_base_ed	Enter End date for a good performance period
perf_base_et	End time for a good performance period
perf_comp_sd	Start date for a bad performance period
perf_comp_st	Start time for a bad performance period
perf_comp_ed	Enter End date for a bad performance period
perf_comp_et	End time for a bad performance period

Example:

```
tfactl srdc ora4030 -sid orcl
tfactl srdc ora4031 -db RDBMS121
tfactl srdc ora4030 -sid orcl -inc_date 2015-02-09 -inc_time 02:48:23
```

Support Tools and Analyzer

14. Support Tools Bundle in TFA

The Support Tools Bundle up until this time has been maintained by Support as part of RAC and DB Support Tools Bundle (Doc ID 1594347.1). From TFA 12.1.2.3.0 these tools are provided as part of the TFA MOS download.

For the full documentation of each tool please refer to the tool MOS documentation.

The tools included from the Support Tools Bundle are:

ORAchk (formerly RACcheck) - Proactive, self service tool to prevent rediscovery of known issues. See Doc ID 1268927.1 for additional details
OSWatcher (formerly OSWatcher Black Box) - Script to collect and archive OS metrics. OSWatcher is required for many reactive types of issues including Instance/Node Evictions and Performance Issues. See Doc ID 301137.1 for additional details.
Procwatcher - A script used to automate and capture diagnostic output for Severe Database Performance issues and Session Level Hangs. See Doc ID 459694.1 for additional details.
ORATOP - A utility allowing for near real-time monitoring of databases (RAC and Single Instance), this utility is built for the Linux platform but can remotely monitor databases on ANY platform from a Linux Client. See Doc ID 1500864.1 for additional details.
SQLT - A tool designed to assist in the tuning of a given SQL Statement. See Doc ID 215187.1 for additional details.
RDA - Powerful diagnostic tool which provides a unified package of support diagnostics tools which provides comprehensive picture of the customer's environment to aid in problem diagnosis. See Doc ID 314422.1 for additional details.
DA - (Installed with RDA) The Diagnostic Assistant (DA) tool provides a common, light-weight interface to multiple diagnostic collection tools (ADR, RDA, OCM, Explorer, and others). See Doc ID 210804.1 for additional details.

The new tools as part of TFA are.

alertsummary – Generates a summary of events for a database or ASM alert file from all nodes.
ls – lists all files TFA knows about for a given file name pattern across all nodes.
pstack– takes a process stack for specified processes across all nodes.
grep – searches alert or trace files with a given database and file name pattern, for a search string.
summary – Gives a high level summary of the configuration.
vi – opens up an editor to view alert or trace files for a given database and file name pattern.
tail – runs a tail on an alert or trace files for a given database and file name pattern.
param – shows all database and OS parameters that match a specified pattern.
dbglevel – A tool to set and unset multiple CRS trace levels with one command
history– Shows the shell history for the tfactl shell
changes – reports any noted changes in the system setup over a given time period. This includes database a parameters, OS parameters, patches applied etc.

The tools will be delivered and deployed where required but not tools that require daemon will not be started by default.

The tools will not replace existing versions that are running from different locations, so it may be worth decommissioning your current version of the tool if you wish to manage that tool through TFA.

FromThe tools only run on the local node for this version.

BGLEVEL

The dbglevel tool can be used to set up a tracing profile that sets ytrace levels for multiple modules in one command. This tool should be used at the request of Oracle Support.ls, pstack, grep, summary, , param, changes, ps above tools, and data or process is running , ls .

For the pstack tool NOT for all processes matching the specified pattern. see pstack of all lmd0 processesfor ALLfrom all nodes

pstack

Again o

The changes, param and ps tools currently do not support context setting.

14.1. **Running TFA Tools**

The tools can be run from the command line or through the 'tfactl' shell. When in the shell it is possible to set a 'database' context that can either be a 'database name', 'ASM', or 'CRS'. Once a context is set then any commands that understand context such as oratop, tail, rgrep, ncdvi and alertsummary will use that context to determine the correct resource target. It is also possible to either type 'run <tool> or simply supply the tool name.

e.g.

tfactl> database RDB11204
Set db to RDB11204

We have set the database name context to RDB11204

*RDB11204 tfactl> run tail alert_*
Mon Dec 22 01:51:09 2014
Closing Resource Manager plan via scheduler window
Clearing Resource Manager plan via parameter

This runs a tail of the local alert_ log for the database RDB11204

*RDB11204 tfactl> tail alert_*
Mon Dec 22 01:51:09 2014
Closing Resource Manager plan via scheduler window
Clearing Resource Manager plan via parameter

This is the same command but showing the 'run' is not required.

RDB11204 tfactl> database
Removed db from analysis context.
tfactl>

Here we can see that the database context has now been removed.

14.2. Passing Arguments to tools

Any part of the command line after the tool name is specified is passed directly to the tool so as to ensure the tool is not dependent on TFA and to allow the tool to be developed in isolation. The tool itself maintains the correct syntax for such arguments.

e.g.

tfactl> database RDB11204
Set db to RDB11204

tfactl> run oratop –bn 5

This would be equivalent to calling the tool outside of TFA with the commands
#oratop –database RDB11204 –bn 5

All tools syntax can be checked through the standard help mechanism.

E.g.

tfactl> tail -h
Usage : /u01/app/11.2.0/grid_11204/bin/tfactl tail [-f] <file name pattern>

Tail all files matching input pattern

e.g:
*/u01/app/11.2.0/grid_11204/ bin/tfactl tail alert_*
/u01/app/11.2.0/grid_11204/bin/tfactl tail -f alert_testdb1.log

Note that the help always shows the full command line (non shell) option in the examples

15. TFA Log Analyzer Tool

TFA Analyze provides three kinds of log analysis.

- Show the most common messages within the logs

This is a quick indication of where your biggest problems maybe. Analyzer gets just the important messages out of your Alert logs and strips all extra information from log messages effectively rolling up the most common messages displaying them in the order from most common to least common. It lets you get straight to the worst problem more quickly. By default, error messages are analyzed but you can specify the type of message if you'd like to analyze warning or other type of messages.

- Search for text within log messages

This is similar to using grep to search but much faster because analyzer checks the time of each message and only shows those matching the last X minutes or any interval of time.

- Analyze the Oracle OSWatcher log statistics

Analyzer reads the various statistics available in the OSWatcher log files and provides detailed analysis showing first, highest, lowest, average and last three readings of each statistic. You can choose any interval even down to a specific minute or second. Analyzer optionally provides the original data from the OSWatcher logs for each value reported on (data point).

15.1. Using 'tfactl analyze'

TFA analyze command provides analysis of system by parsing Database, ASM & CRS Alert Logs, System Message Log, OSW Top and OSW Slabinfo files. The output of the command can be filtered by component, error type and time.

The detailed usage of command is below :-

Usage : /opt/oracle/tfa/tfa_home/bin/tfactl analyze [-search "pattern"] [-comp <db|asm|crs|acfs|os|osw|oswslabinfo|all> [-type <error|warning|generic>] [-since <n>[h|d]] [-from "MMM/DD/YYYY HH24:MI:SS"] [-to "MMM/DD/YYYY HH24:MI:SS"] [-for "MMM/DD/YYYY HH24:MI:SS"] [-node <all | local | n1,n2,..>] [-verbose] [-o <file>]

Options:

-search "pattern"	Search for pattern in system and alert logs in specified time range.
-comp	Components to analyze. Default is all.
-type	Analyze messages of specified type. Default is ERROR.
-node	Specify comma separated list of host names. Pass "local" for only analyzing files in local node. Default is all.

Time Options: Specify one of 3 options, -since or -for or -from,-to

-since	Analyze for specified duration before current time
-for	Analyze for specified time
-from,-to	Analyze for specified time period

Other Options:

-verbose	Show verbose output.
-o	Write the output to <file> instead of printing on screen.

Description of Components:

Specify the component to analyze or search in using -comp flag.
Default is ALL components.
Valid options are:

DB	: Database Alert Logs
ASM	: ASM Alert Logs
CRS	: CRS Alert Log
ACFS	: ACFS Log
OS	: System Message File
OSW	: OSW Top Output
OSWSLABINFO	: OSW Slabinfo Output

Description of Message Types:

Specified using -type flag. Default is ERROR.
tfactl analyze classifies all the messages into different categories. The analysis component provides count of messages by message types configured and lists all unique messages grouped by count within specified filters.

The message type patterns for different components are specified in a property file.

Existing patterns for DB/ASM Alert Logs :

ERROR :	.*ORA-00600:.*
	.*ORA-07445:.*
	.*IPC Send timeout detected. Sender: ospid.*
	.*Direct NFS: channel id .* path .* to filer .* PING timeout.*
	.*Direct NFS: channel id .* path .* to filer .* is DOWN.*
	.*ospid: .* has not called a wait for .* secs.*
	.*IPC Send timeout to .* inc .* for msg type .* from opid.*
	.*IPC Send timeout: Terminating pid.*
	.*Receiver: inst .* binc .* ospid.*
	.* terminating instance due to error.*
	.*: terminating the instance due to error.*
	.*Global Enqueue Services Deadlock detected
WARNING:	NOTE: process .* initiating offline of disk .*
	.*WARNING: cache read a corrupted block group.*
	.*NOTE: a corrupted block from group FRA was dumped to
generic :	All messages that do not match above patterns

Existing patterns for CRS Alert Logs :

ERROR:	.*CRS-8011:.*,.*CRS-8013:.*,.*CRS-1607:.*,.*CRS-1615:.*,
	.*CRS-1714:.*,.*CRS-1656:.*,.*PRVF-5305:.*,.*CRS-1601:.*,
	.*CRS-1610:.*,.*PANIC. CRSD exiting:.*,.*Fatal Error from AGFW Proxy:.*
WARNING :	.*CRS-1603:.*,.*CRS-10051:.*,.*CRS-2409:.*,.*CRS-1625:.*
generic :	All messages that do not match above patterns

15.2. Searching with 'tfactl Analyze'

The '–search' flag supports case sensitive and case in-sensitive search in Alert/System message files across cluster with in specified filters.

Default is case insensitive.

Example of case in-sensitive search:

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -search "error" -since 2d
Search string "error" in alert and system logs in past 2 days

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -comp os -for "Feb/27/2014 11" -search "."
Show all system log messages at time Feb/27/2014 11

Example of case sensitive search:

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -search "/ORA-/c" -comp db -since 2d
Search case sensitive string "ORA-" in past 2 days in DB Alert Logs

15.3. Analysis with 'tfactl analyze'

When '-search' flag is not specified, the analyze command provides the summary of messages within specified filters from alert/system log messages across cluster.

The output will show message counts grouped by type, ERROR, WARNING and generic and shows unique messages in a table from message type selected for analysis, default analysis type is ERROR. Note that generic type is assigned to all messages which are not in ERROR/WARNING type.

When OSWatcher data is available OSW and OSWSLABINFO components provide summary view of oswatcher data.

Examples of Alert/System log analysis:

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -since 5h
Show summary of events from alert logs, system messages in last 5 hours.

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -comp os -since 1d
Show summary of events from system messages in last 1 day.

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -since 1h -type generic
Analyze all generic messages in last one hour.

Examples of OSW Top and Slabinfo analysis:

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -comp osw -since 6h
Show OSWatcher Top summary in last 6 hours

/opt/oracle/tfa/tfa_home/bin/tfactl analyze -comp oswslabinfo -from "Feb/26/2014 05:00:01" -to "Feb/26/2014 06:00:01"
Show OSWatcher slabinfo summary for specified time period

16. Data Redaction with TFA

Some customers have significant problems providing diagnostic data to support as that data contains what they consider private or sensitive information like hostnames, IP Addresses, etc. In this release TFA offers a simple high level way of hiding sensitive data by replacing that data in the files or their names. To utilize this feature the file mask_strings.xml simply needs to be placed in the tfa_home/resources directory. TFA will see this file exists and will then use the data inside to replace strings in file names and contents. The format of the file mask_strings.xml should be as follows.

```
<mask_strings>
    <mask_string>
            <original>dummy1</original>
            <replacement>Node1</replacement>
    </mask_string>
    <mask_string>
            <original>dummy2</original>
            <replacement>Node2</replacement>
    </mask_string>
</mask_strings>
```

Appendix A. TFA_HOME Directory

For new installations the directory structure has been changed to more easily support installation on a file system that is shared across the nodes of the cluster. TFA is installed under a TFA_BASE that is the same cluster wide. The TFA_HOME location is now a node specific directory that resides under the TFA_BASE. TFA_BASE is set at install time and cannot be changed without a reinstallation. There will be a common bin directory created TFA_BASE/bin that has the tfactl binary. By default the repository directory also sits directly under TFA_BASE except when installed in an Oracle Grid Infrastructure home which means that on a shared file system no zip files are actually copied across nodes as the files are visible to all nodes. For installation in an Oracle Grid Infrastructure home the repository will be placed in the Grid Home owners ORACLE_BASE directory as ORACLE_BASE/tfa/repository.
Note that TFA can support a cluster of nodes where not all nodes can see the same shared file systems. For Example when a TFA cluster is made up of a primary and standby RAC cluster.

The TFA_BASE directory for a full install (with JRE) by default takes approximately 125 MB after installation. This consists of the binary files , configuration files and the initial Berkeley Database. The Berkeley database will grow as files and scan events are saved however file data is removed when the files are deleted. For installation in an Oracle Grid Infrastructure home the database will be placed in the Grid Home owners ORACLE_BASE directory as ORACLE_BASE/tfa/<nodename>/database.

Initially the diagnostic data repository is sized to be the smaller of 10GB or 50% of the free space in the file system. The *tfactl set* command may be used to change the repository size or to assign it to a different directory. If the repository fills or the file system housing the repository goes to below 1GB free space then TFA will stop collecting diagnostics until more space becomes available. To remove old collections form the repository use ‘tfactl purge’.

Appendix B. Scan Events

The following events will trigger Automatic Diagnostic Collection when they are detected whilst scanning the alert logs and the automatic collection is set to true as described in Section 8.1 .

String:
- Search for:
  - .*ORA-297(01|02|03|08|09|10|40).*
- In Files :
  - Alert Logs - ASM and RDBMS
- Error Description:
  - 29701, unable to connect to Cluster Synchronization Service
  - 29702,error occurred in Cluster Group Service operation
  - 29703, error occurred in global enqueue service operation
  - 29708, error occurred in Cluster Synchronization Services
  - 29709, Communication failure with Cluster Synchronization Services
  - 29710, Current operation aborted by Cluster Synchronization Services
  - 29740 ,evicted by instance number %s, group incarnation %s
  - 29770,global enqueue process %s (OSID %s) is hung for more than %s seconds
- When trimming buffer:
  - 600 seconds before the event
  - 600 seconds after the event
- Collect files from:
  - Operating System
  - Clusterware
  - ASM

String:
- Search for:
  - .*ORA-00600.*
- In Files :
  - Alert Logs - ASM and RDBMS
- Error Description:
  - ORA-00600 internal errors
- When trimming buffer the data :
  - 600 seconds before the event
  - 600 seconds after the event
- Collect files from:
  - Reporting Database instance only.

String:
Search for:
.*ORA-07445.*
In Files :
Alert Logs - ASM and RDBMS
Error Description:
ORA-07445 internal errors
When trimming buffer the data :
600 seconds before the event
600 seconds after the event
Collect files from:
Reporting Database instance only.

String:
Search for:
.*ora-4(69|([7-8][0-9]|9([0-3]|[5-8]))).*
In Files :
Alert Logs - ASM and RDBMS
Error Description:
ORA-469 to 498 process failure errors
When trimming buffer the data :
600 seconds before the event
600 seconds after the event
Collect files from:
Reporting Database instance only.

String:
Search for:
.*ORA-32701.*
In Files :
Alert Logs - ASM and RDBMS
Error Description:
Error 32701 is reported when Hang manager suspects a hang.
When trimming buffer:
1200 seconds before the event
600 seconds after the event
Collect files from:
Operating System
Clusterware
ASM

String:

Search for:
.*ORA-494.*
In Files :
Alert Logs - ASM and RDBMS
Error Description :
ORA-494 Enqueue timeout.
When trimming buffer:
1200 seconds before the event
600 seconds after the event
Collect files from:
RDBMS
ASM

String:
Search for:
.*Sytem\sState\sdumped.*
In Files :
Alert Logs - ASM and RDBMS
Error Description:
.
When trimming buffer:
600 seconds before the event
300 seconds after the event
Collect files from:
Operating System
Clusterware
ASM

String:
Search for:
.*CRS-16(07|10|11|12).*
In Files :
Alert Log CRS
Error Description:
.
When trimming buffer:
3600 seconds before the event
300 seconds after the event
Collect files from:
Operating System
Clusterware
ASM

Appendix C. Using Custom SSL Certificates

From TFA version 12.1.2.6.0 you can replace the Self-Signed certificates that are generated at install time with either personal self signed certificates or CA signed certificates.

Using Self-Signed Certificates.

The example process below uses the java tool keytool to generate . Java version 1.5 or above is required. Note that the location on the myserver.jks and myclient .jks files is not fixed however we would suggest placing under the tfa_home directory.

1. Generate a private key and keystore file containing the private key and self signed certificate for the server.

keytool -genkey -alias server_full -keyalg RSA –keysize 2048 -validity 18263 -keystore myserver.jks
-- The command will prompt for input of keystore and key passwords as well for distinguished name information.

2. Generate a private key and keystore file containg the private key and self signed certificate for the for client.

keytool -genkey -alias client_full -keyalg RSA –keysize 2048 -validity 18263 -keystore myclient.jks
-- The command will prompt for input of keystore and key passwords as well for distinguished name information.

3. Export the Server Public Key certificate from the server keystore.

keytool -export -alias server_full -file myserver_pub.crt -keystore myserver.jks -storepass <password>

4. Export the Client Public Key certificate from the client keystore

keytool -export -alias client_full -file myclient_pub.crt -keystore myclient.jks –storepass <password>

5. Import the Client Public certificate into the client keystore

keytool -import -alias server_pub -file myserver_pub.crt -keystore myclient.jks -storepass <password>

6. Import the keytool -import -alias client_pub -file myclient_pub.crt -keystore myserver.jks -storepass <password>

7. Lock down the permissions on the JKS files for root user read only

chmod 700 myclient.jks
chmod 700 myserver.jks

8. Manually copy the certificates to each node.

9. Follow the instruction below to use ‘tfactl set sslconfig’

Using CA signed certificates

The process below uses the java keytool and openssl commands to generate a certificate signing request and then utilize the signed certificate for TFA. Note that the location on the myserver.jks and myclient .jks files is not fixed however we would suggest placing under the tfa_home directory.

1. Generate a private key for the client and server request.

openssl genrsa -aes256 -out myserver.key 2048
openssl genrsa -aes256 -out myclient.key 2048

2. Create Certificate Signing Request(CSR) for both server and client

openssl req -key myserver.key -new -sha256 -out myserver.csr
openssl req -key myclient.key -new -sha256 -out myclient.csr

3. Send the CSR for client and server to the relevant signing authority and receive back the signed certificates (myserver.cert and myclient.cert) and the CA root certificate.

4. Convert the certificates to JKS format for server and client.

openssl pkcs12 -export -out serverCert.pkcs12 -in myserver.cert -inkey myserver.key

keytool -v -importkeystore -srckeystore serverCert.pkcs12 -srcstoretype PKCS12 -destkeystore myserver.jks -deststoretype JKS

openssl pkcs12 -export -out clientCert.pkcs12 -in myclient.cert -inkey myclient.key

keytool -v -importkeystore -srckeystore clientCert.pkcs12 -srcstoretype PKCS12 -destkeystore myclient.jks -deststoretype JKS

5. As TFA used 2 way Authentication we need to import server's public key to client jks file and client's public key to server jks file.

keytool -import -v -alias server-ca -file myserver.cert -keystore myclient.jks
keytool -import -v -alias client-ca -file myclient.cert -keystore myserver.jks

6. Import the Root CA cert from the signing Authority into the TFA server certificate.

keytool -importcert -trustcacerts -alias inter -file caroot.cert -keystore myserver.jks

7. Lock down the permissions on the JKS files for root user read only

chmod 700 myclient.jks
chmod 700 myserver.jks

8. Manually copy the certificates to each node.

9. Follow the instruction below to use 'tfactl set sslconfig'

Using 'tfactl set sslconfig' to use the new self signed or CA signed certificates.

Note this must be completed on each node.

1. Run the command to make tfa utilize the new certificates

./tfactl set sslconfig

Please Enter server certificate path : server.jks
Please Enter Password for server keystore keypass :
Please Confirm Password for server keystore keypass :
Please Enter Password for server keystore storepass :
Please Confirm Password for server keystore storepass :
Please Enter client certificate path? :client.jks
Please Enter Password for client keystore keypass :
Please Confirm Password for client keystore keypass :
Please Enter Password for client keystore storepass :
Please Confirm Password for client keystore storepass :
SSL certificate details successfully set

2. Restart the TFA process to utilize the new keys.

./tfactl stop
./tfactl start

Appendix D. Troubleshooting TFA

All problems with TFA should be reported to Oracle Support, and they will be able to advise on any required action to resolve your issues. Please however note some basic details that may help when engaging support.

By default TFAMain writes it's trace data to 2 files under the tfa_home/log directory except when installed in an Oracle Grid Infrastructure home. For installation in an Oracle Grid Infrastructure home the logs will be placed in the Grid Home owners ORACLE_BASE directory as ORACLE_BASE/tfa/<nodename>/log.

1) Syserrorout.<timestamp>
2) tfa.<timestamp>

Both of these files should be examined when a problem occurs as they may help to determine the cause of errors within the TFAMain process, or the java side of the Command line handler. Any java exceptions will be written to the syserrorout file.

If an installation error occurs with TFA then TFA cleans itself up which will remove all the logs so as part of the cleanup process all the files in the TFA log directory are copied to /tmp/tfa.

To increase the trace verbosity for the java code you should use the command 'tfactl set' as described earlier to dynamically increase the trace level from 1 which is the default to 2 or 3 as requested by support. Once the required trace data has been generated the trace level should immediately be set back to 1 to ensure trace data does not cause performance and resource usage issues. The trace level setting persists across restarts.

tfactl is a command line tool that writes all of it's trace data to

tfa_home/ diag/tfa/tfactl/user_<user>/trace

except when part of a GI installation where it will be in

$ORACLE_BASE/<node>/diag/tfa/tfactl/user_<user>/trace

There are 6 levels of tracing.

Level 0 - SCREEN - Trace levels 1-5, the output is sent to console.
Level 1 - ERRORS - Tracing errors in execution path.
Level 2 - WARNINGS - Tracing warnings in execution path.
Level 3 - NORMAL - Tracing normal messages such as execution paths.
Level 4 - INFO - Tracing decision statements and loops.
Level 5 - DEBUG - Tracing for debugging purposes.Comments and Questions.

To enable the tracing during the execution of 'tfactl' set the verbose mode to the desired level,

#tfactl [-v {screen|errors|warnings|normal|info|debug|none}] [command]

#tfactl -v screen print config

The TFAMain JVM runs continuously and will be restarted by init if it is killed, to stop TFAMain being restarted by init the 'init.tfa stop' or 'init.tfa shutdown' must be run. To stop the 'init.tfa run' process from being respawned by init the 'init.tfa shutdown' command must be run.

If at any time there is concern that the TFAMain process is causing a problem on the system a call to 'init.tfa stop' should stop it even if is not responding to normal commands.

It should be noted that TFAMain will be seen to consume most CPU just after install when the first inventory is run as it will go through all directories and files within those directories to determine the file data. After this time even if TFA is tailing a number of alert logs the CPU usage should be negligible. Inventories after the initial inventory take considerably less resources as TFA only looks at new or changes files.

Appendix E. What's new in 12.1.2.7.0

New in 12.1.2.7.0

1) Collect ash report as part of diagcollection for a given database.
2) Collect from ODA Dom0 when collecting from the Base VM
3) Fixed init integration Issues including providing rc script links to ensure clean shutdown and startup of TFA on reboot.
4) TFA Installation now writes a verbose install log. When patching it provides location of the patch log.
5) TFA No longer tries to collect for components that are not valid for the specific system when doing default 'ALL' collections.
6) Fixed issue where some Exadata cells are not discovered by TFA.
7) TFA now ensures collections can be read by the requesting user on all platforms.
8) TFA will now wait for the TFA_HOME to be available after boot before trying to start the Daemon process.
9) Initial support for SRDC driven collections. Initially ORA-04030, ORA-04031, and dbperf to collect data comparing baseline and poor performance times. EG '*tfactl srdc ora4030*'

New in 12.1.2.6.3

1) Support for TLS Protocol restriction (see Appendix I).

New in 12.1.2.6.0

2) Support for User supplied SSL certificates
3) Critical Bug Fixes

New in 12.1.2.5.2

1) Critical Bug Fixes.

New in 12.1.2.5.0

1) Critical Bug Fixes.

New in 12.1.2.4.0

1) New TFA tools summary, events, changes, ps, pstack. Param all with cluster support
2) New dbglevel tool to help when setting CRS trace levels for multiple modules across nodes.
3) Cluster support for existing TFA tools alertsummary, grep, tail.
4) Updates to Support Tools DARDA, SQLT, ORACHK
5) New Collections from ODA Dom0
6) Support for Dom0 collections on Exadata.
7) Critical Bug Fixes.

New in 12.1.2.3.0

1) TFA will Auto Purge collections older than a specified age when repository space is exhausted.
2) TFA supports collection od ZDLRA Specific data.
3) Collection of AWR reports from specified databases is supported in either text or html format for every snapshot interval in the collection period. Only supports the non RAC report initially.
4) Inclusion of the Support tools (Orachk, oratop, sqlt, OSWatcher, Procwatcher) with a single interface to run these tools on the local node.

5) Tools to easily view local alert and trace logs from a single interface without knowing their exact location.
6) 'tfactl diagnosetfa' initial version of this option for TFA to collect it's own diagnostics on the local node.
7) Critical Bug Fixes.

New in 12.1.2.1.0

1) TFA will now collect trace files even when they have not been allocated a type if their directory is of the right component and they have been updated within the time period requested.
2) There is now a tfactl shell that can be used to call commands. This is a beta feature that will be enhanced in upcoming releases to provide a flexible user interface.
3) TFA collections can now be stopped using 'tfactl collection'
4) Diagcollect will not return to the prompt until all nodes have finished their collections.
5) Critical Bug fixes.

New in 12.1.2.0.0

1) Support for Oracle GI and RAC 12.1.0.2 trace files
2) TFA Inventory performance improvements.
3) TFA Collection performance improvements.
4) New –oda and –odastorage flags for enhanced ODA data collections
5) Collection of available oracheck and oratop reports
6) More granular trace level adjustment.
7) Use of a 2048 bit certificate.
8) Critical Bug fixes

New in 3.2.0.0

1) First version of the Log Analyzer.
2) Addition of –notrim for diagnostic collections.
3) Support for TFA on zLinux.
4) Collection of ExaWatcher data.
5) Sundiag collection
6) Aging of TFA log files restricting maximum size.
7) Multiple performance enhancements
8) Critical Bug fixes

Appendix F. Changing TFA Default Ports

By default TFA will try to use ports 5000 to 5005 for secure root communications however if that port range is not available on you system then this can be changed.

The file TFA_HOME/internal/usableports.txt when installed looks like this.

```
# cat /opt/oracle/tfa/tfa_home/internal/usableports.txt
5000
5001
5002
5003
5004
5005
```

To change the ports TFA will try to use you need to :-

1) Stop TFA on all nodes
2) Edit usableports.txt to replace the ports you wish tfa to use.
3) Remove the file tfa_home/internal/port.txt on all nodes.
4) Start TFA on all nodes.

Note: The usableports.txt file must be identical on all nodes

TFA will also use one of ports 5006 to 5011 on the loopback interface for non root communications however if that port range is not available on you system then this can also be changed.

The file TFA_HOME/internal/NonRootusableports.txt when installed looks like this.

```
# cat /opt/oracle/tfa/tfa_home/internal/usableNonRootports.txt
5006
5007
5008
5009
5010
5011
```

To change the ports TFA will try to use you need to :-

1) Stop TFA on all nodes
2) Edit usableports.txt to replace the ports you wish tfa to use.
3) Remove the file tfa_home/internal/NonRootport.txt on all nodes.
4) Start TFA on all nodes.

Appendix G. Known issues

When upgrading TFA through TFA sockets the remote nodes may fail to upgrade due to a socket issue. Once the upgrade is completed you should see a report showing all nodes with the same version, buildid, and stating UPGRADED.

```
.-----------------------------------------------------------------------------------------------.
| Host    | TFA Version   | TFA Build ID                        | Upgrade Status           |
+---------+---------------+-------------------------------------+--------------------------+
| node1   |  12.1.2.6.0   | 12126020151019114604                | UPGRADED                 |
| node2   |  12.1.2.6.0   | 12126020151019114604                | UPGRADED                 |
'---------+---------------+-------------------------------------+--------------------------'
```

If you do not see this but instead see something like

```
.-----------------------------------------------------------------------------------------------.
| Host    | TFA Version   | TFA Build ID                        | Upgrade Status           |
+---------+---------------+-------------------------------------+--------------------------+
| node1   |  12.1.2.6.0   | 12126020151019114604                | UPGRADED                 |
| node2   |  12.1.2.3.0   | 12120020140619094932                | NOT UPGRADED             |
'---------+---------------+-------------------------------------+--------------------------'
```

Then you will have to copy the TFA installer to all nodes that failed to upgrade and run it locally on those nodes.

#installTFALite –local

Once the binaries are upgraded you will have to replace the root ssl certificates from the upgrade initiating node. The following files must be copied from the exiting configuration node to the node to be added, be owned by root on the machine to be added and have ‘700’ permissions.

tfa_home/server.jks
tfa_home/client.jks
tfa_home/internal/ssl.properties

When OSWatcher is already installed on a system TFA will not try to move it to the TFA default location and will generate errors regarding OSWatcher if that tool is used.

tfactl> oswbb start
Error: Cannot find OSWatcher files under
/u01/app/grid/tfa/repository/suptools/slcac460/oswbb/root/archive
OSWatcher analyzer commands are supported only when it is running from TFA_HOME

You will also see errors if TFA is uninstalled.

Notifying Other Nodes about TFA Uninstall...

Sleeping for 10 seconds...
Stopping Tools...
Can't read /u01/app/grid/tfa/repository/suptools/slcac460/oswbb/.osw.prop. Exiting..
SQLT can not be configured to run in daemon mode.

These messages can be ignored or you can redeploy OSW under TFA control except on Engineered systems (Oracle Database Appliance, Exadata, ZDLRA).

The help for the TFA tools shows the command line be be tfactl.pl in the tfa_home/bin directory however this the correct command to run would be tfactl from the tfa_base/bin directory or the GI_HOME/bin directory for a Grid Infrastructure installation.

Non Root access for all DBA users should be activated by default when Non Root access is enabled however this does not happen in all cases. To enable TFA access for other users the root user must run '*tfactl access add -user xyx'* where xyz represents the group that needs to be added to TFA Non Root access.

In TFA 12.1.2.5.0 and above non root access by group has been removed. Each user that needs to access tfa has to be specifically added.

When an existing free standing TFA is installed and TFA is then patched with Grid Infrastructure as part of Oracle 12.1.0.2 the tfa_home is moved into the Grid Infrastructure home and the repository directory will be moved to the Grid Infrastructure owners Oracle Base directory. If the repository directory has been changed to a non default location then that change will be lost. Use the '*tfactl repository'* command to reset the required value.

If an attempt to install TFA 3.2.0.0 is run on a machine where TFA 12.1.2.0.0 or above is installed then the TFA installation may be corrupted. Due to the fact that the old TFA version cannot handle the new Build ID's that TFA uses to determine if a patch is required or not.

Appendix H. Disabling SSL/TLS Protocols

With TFA 12.1.2.6.3 and above it is possible to restrict the TFA daemon from communicating with specific TLS protocols. The 'tfactl print protocols' command shows us which protocols are available and which ones are restricted.

Note: The available protocols depend on the java version TFA is using:-

Java 1.5/6 – Supports only TLSv1
Java 1.7 and above Support TLSv1.1 and TLSv1.2

TFA always restricts use of SSLv3 and SSLv2Hello.

#tfactl print protocols

Node1
Protocols
Available : [TLSv1, TLSv1.2, TLSv1.1] Restricted : [SSLv3, SSLv2Hello]

Node2
Protocols
Available : [TLSv1, TLSv1.2, TLSv1.1] Restricted : [SSLv3, SSLv2Hello]

To restrict TFA to only use the TLSv1.2 protocol you would run
tfactl restrictprotocol TLSv1
Protocol TLSv1 restricted from TFA Cluster
tfactl restrictprotocol TLSv1.1
Protocol TLSv1.1 restricted from TFA Cluster

If you are using a Java version that does not support all the TLS versions and you try to restrict a protocol that would break TFA then the following message is reported.

tfactl restrictprotocol TLSv1
FAILED Restricting TLSv1 will leave no supported enabled protocol to run TFA in Node1. You will need a newer java version to support other protocols. Please set new java home using tfactl set java_home

As stated you can point TFA to a more recent Java version using
tfactl set java_home= /etc/alternatives/jre_1.8.0/

Once TFA is restarted it will be possible to restrict the required protocols.

Appendix I. Licenses for Third Party Components

G.1. Apache 2.0 License

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

   Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Revised: January 4, 2016 Bill Burton, Oracle

Contributors: Amrita Chaurasia, Girish Adiga, Bryan Vongray, Chandrabhushan Nagur, Bob Caldwell, Sandesh Rao

DOCUMENTATION ACCESSIBILITY
Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market- leading technology.