Automatic Fault Diagnostics

An Oracle White Paper
November 2007

ORACLE

INTRODUCTION

The goal of the new diagnostic infrastructure is to reduce the time to resolve customer problems

Oracle Database 11g introduces a new diagnosability infrastructure. The main goal of this new framework is to reduce the time it takes to resolve customer problems.

Background

Historically, diagnosing errors in Oracle software has been problematic for both customers and Oracle. Several factors contribute to this.

The main method of finding information about critical errors is to review the alert log. Since critical errors are mixed with normal messages, it's hard to get an overview. It's necessary to either navigate through the alert log, or use some operating system utility to search through it.

After deciding to engage Oracle Support, the customer needs to determine which diagnostic data to include. The alert log entry identifies the process trace file containing the error dump. This file can be in different directories, depending on which process encountered the error.

After establishing which trace file to send to Oracle, the DBA needs to make a judgment call on whether to include additional information. Typically, the alert log is also included, or an excerpt from it.

When Oracle receives the information from the customer, it's possible that some piece of information required for diagnosis was not included. If the information is missing because the wrong files were sent to Oracle, the customer has to go back and try to locate the correct files. If the information was not dumped at all, Oracle Support might request that the customer reproduce the problem with some debug switches set. In either case, the result is multiple round trips between the customer and Oracle. This adds to the time it takes before an error can be diagnosed.

It can also be difficult to establish the extent of an issue. Is the entire database corrupt, or just a single block? The customer often depends on assistance from Oracle Support to determine this.

There is also a maintenance burden on DBAs. The diagnostic data on the customer system tends to grow over time. In particular, debug switches can cause the data to

grow dramatically. When removing diagnostic data to free up space, the DBA has to determine what can be removed.

Making sure that the diagnostic data doesn't fill up the disk is particularly problematic when the data grows quickly and unexpectedly. This can happen if a large number of processes hit the same error at the same time, due to a shared root cause.

Benefits

The new diagnostic infrastructure in Oracle Database 11g addresses these shortcomings.

Issues can be detected proactively using new health checks. These health checks also provide a way to determine the extent of an issue. The ability to diagnose issues based on the first failure is improved by enhanced dumping and using tools to automatically select and package diagnostic data. Maintenance is made easier by automatically purging old diagnostic data.

Other Oracle products

While this paper focuses on Oracle Database, the diagnostic framework presented here is used in other Oracle products as well, which itself allows for even more workflow automation across the technology stack. The key to this is that the framework allows diagnostic data generated by Oracle products to be handled in a consistent way.

At the time of writing, the diagnostic framework is used by RDBMS, ASM, OCI, Net and Application Server 11.1.

TERMINOLOGY

The diagnostic framework groups all critical errors into "problems", and calls each occurrence of a problem an "incident"

To be able to manage diagnostic data more efficiently, a few new concepts have been introduced.

Problems and incidents

A **problem** is a critical error in the database. Critical errors manifest as internal errors, such as ORA-600, and other severe errors, such as ORA-7445 (operating system exception) or ORA-4031 (out of memory in the shared pool).

Each problem has a **problem key**, which is a text string that describes the problem. The problem key includes the error code (such as `ORA 600`), and in most cases, one or more error parameter values.

An **incident** is a single occurrence of a problem. When a problem occurs multiple times, an incident is created for each occurrence.

Traces and dumps

There are some differences between traces and dumps, and to understand the new framework, it is helpful to know the characteristics of both types.

Each server process can write to an associated trace file. Trace files are updated periodically over the life of the process and can contain information on the process environment, status, activities, and errors. In addition, when a process detects a critical error, it writes information about the error to its trace file.

A dump is a specific type of trace file. A **dump** is typically a one-time output of diagnostic data in response to an event (such as an incident), whereas a **trace** is a continuous output of diagnostic data. When an incident occurs, the database writes one or more dumps to the incident directory created for the incident.

ARCHITECTURE OVERVIEW

The Oracle-generated diagnostic data has been combined into a single directory structure, called the Automatic Diagnostic Repository (ADR). This repository stores diagnostic data, including logs, traces, dumps, cores, metadata and reports.

The repository uses metadata to keep track of problems, incidents and other items.

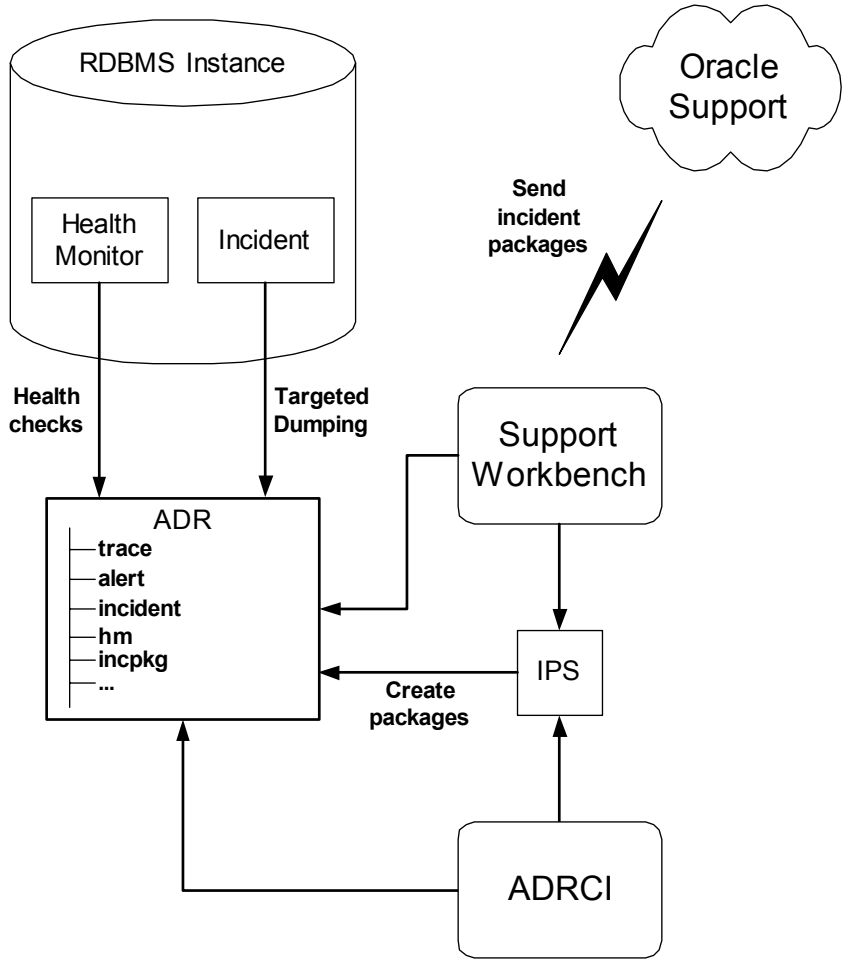
A diagnostic component called Health Monitor runs health checks proactively, or as part of an incident, and stores the results in ADR.

When incidents occur, detailed targeted dumps are written to ADR.

Incident data and other ADR contents can be automatically packaged and uploaded to Oracle using the new Incident Packaging Service (IPS).

There are two user interfaces for managing diagnostic data. The graphical interface is Enterprise Manager's Support Workbench. The command-line interface is called adrci.

The following diagram shows the diagnostic components, and how they interact with each other.



AUTOMATIC DIAGNOSTIC REPOSITORY (ADR)

The Automatic Diagnostic Repository (ADR) provides a single location for diagnostic data

The Oracle-generated diagnostic data has been combined into a single directory structure, called the Automatic Diagnostic Repository (ADR). This repository stores diagnostic data (including logs, traces, dumps, cores, metadata and reports).

For files that already exist in earlier versions, this is mostly a move, and the files can be accessed as before in their new location. There are some exceptions – please see section Customer-Side Changes below.

This single directory hierarchy can be shared by multiple Oracle products to store their diagnostic data.

The location of the directory structure is referred to as the ADR base. In RDBMS, the ADR base is determined by the initialization parameter DIAGNOSTIC_DEST. Inside ADR, diagnostic data for each instance is kept in a subdirectory structure, which is referred to as an ADR home. The directory structure within ADR base is diag/*product_type*/*product_id*/*instance_id*.

As an example, consider a database named prod, with an instance name prod1. The ADR home would be diag/rdbms/prod/prod1.

The ADR data is highly structured. There is metadata stored in database-like relations. Incident dumps are split into separate files, so that an incident can be reviewed in isolation if required. The alert log is in a new XML-based format. A traditional text-based alert log is also maintained for backwards compatibility.

ADR can be accessed through Enterprise Manager's Support Workbench, or using the command-line utility adrci

The main interface to ADR is Support Workbench, which is part of Enterprise Manager. There is also a command-line utility called "adrci", which is useful when EM is not available. It is possible to run command scripts with adrci.

The contents of ADR are available even when the database is down.

ADR Directory Structure

Each ADR home contains a number of subdirectories. Some of them hold diagnostic data that can be useful to an advanced user, but some only hold internal information.

/trace

Holds normal trace files (continuously generated tracing info). These include both background traces and foreground traces (which in earlier versions went into BACKGROUND_DUMP_DEST and USER_DUMP_DEST, respectively). It also holds a text version of the alert log, for backwards compatibility.

For the database, this directory contains both traces and trace metadata (.trm) files.

/incident

Contains a subdirectory per incident, named incdir_*incidentid*. Each incident directory has one of more dump files with diagnostic data, as well as trace metadata.

/alert

Contains the XML-based alert log (log.xml).

/metadata

Contains files for each relation ("table"). Together, the files in this directory serve as the ADR data dictionary.

/cdump

Contains core dumps.

/hm

Contains Health Monitor reports.

/incpkg

This directory structure is used as a staging area for the Incident Packaging Service (IPS). For instance, exported metadata is in a subdirectory of this directory.

/ir

Internal directory with RMAN repair scripts used by the Data Repair Advisor (DRA).

/lck

Internal directory with lock files, which control access to ADR metadata files.

/sweep

Internal directory with temporary files written by processes that encounter incidents (writing directly to ADR metadata files is inappropriate due to concurrency issues)

ADR Files

The Automatic Diagnostic Repository (ADR) provides a single location for diagnostic data

The diagnostic files from earlier versions still exist, such as traces, dumps, log and core dumps. They have been moved into ADR, and in some cases, there are other changes too.

Trace Files

The trace files that used to be in BACKGROUND_DUMP_DEST and USER_DUMP_DEST are now in *adr_home*/trace. The old parameters have been deprecated and will be ignored.

The format of the trace files is the same as before, but there is also an additional metadata file (.trm), which

Dump Files

The dumps that happen when a critical error is encountered have been broken out from the process trace file, and are now in *adr_home*/incident/incdir_*incidentid*. The format is the same as the one used for trace files, including metadata.

The file name convention is somewhat different - incident dumps contain the incident number in the file name.

Core Files

The core dumps that used to be in CORE_DUMP_DEST are now in *adr_home*/cdump. The old parameter is still available if there is a need to override the default.

Alert log

There is a new alert log in XML format. It is in *adr_home*/alert, and is called log.xml.

For backwards compatibility, a text version of the alert log is also maintained in *adr_home*/trace.

Purging

Diagnostic data is automatically purged as data grows old, which reduces the need for DBA intervention

To prevent diagnostic data from growing out of control, ADR has a purging mechanism.

In most cases, diagnostic data of a certain age can be safely removed. For instance, let's assume that the first incident of a particular problem occurred 30 days ago. If there have been no more incidents of that problem, it can be considered resolved (or at least non-recurring). If there have been more incidents of the problem, there is newer diagnostic data for that problem. In either case, it should be safe to purge that first incident.

There are various other checks before data is purged. For instance, it's possible to set a "Disable Purge" flag for any incident using Support Workbench. As another example, IPS package information isn't purged until the package status has been "uploaded" for at least 30 days.

By default, trace files and dump files are purged after 30 days, and metadata is purged after 365 days

The purging logic uses two different time ranges. Some types of data are kept for a relatively short period before purging, and other types are kept for a longer period.

The default "short' retention time period is 30 days. This is the retention used for regular trace files (which tend to be the bulk of the diagnostic data).

The default "long" retention time period is 365 days. This is the retention used for most of the metadata, such as the incident relation.

Other Maintenance

There are a few other aspects of maintaining ADR.

ADR is automatically created (or recreated) when the database starts up

Removing ADR

If ADR is removed for some reason, a new ADR is created automatically.

Backups

Due to the way that ADR metadata is written, it is not safe to copy ADR while the database is up and running. Instead, use the adrci command BEGIN BACKUP before copying the files, and END BACKUP when done.

This only concerns the metadata relations - normal files can be copied at any time.

INTELLIGENT DIAGNOSTIC DATA DUMPS

Context-sensitive dumping provides significant improvements in first-failure analysis capabilities

The intelligent dumping mechanism is a key feature in providing first-failure analysis capabilities. There are several improvements to make sure that the most relevant diagnostic information is dumped for each failure.

Targeted Dumps

Instead of dumping the same types of items for each incident, the context decides what and how much to dump

Instead of just dumping the same set of information for every error, sophisticated rules are used to determine what to dump in each incident. The rules can consider such conditions as specific error numbers or arguments, or whether a specific component or function is active (on the stack). Default rules per product determine the default dumping behavior.

The targeted dumps save space by skipping unnecessary dumps, allowing individual dumps to be more detailed.

Some dumps are done by a background slave process, which can result in more than one dump file in the incident directory. Such background dumps are controlled by Resource Manager, to ensure that the instance is not negatively impacted.

Automatic Flood Control

If the number of incidents exceeds certain limits, the incidents are flood-controlled

In many cases, the system can encounter a large number of errors in a short period. One example would be if a data block in a heavily used table is corrupted. After the first few incidents, the dumps do not add a lot of value. In fact, filling up the disk with redundant dumps makes it more difficult to diagnose other issues.

The solution to this is called incident flood control. This mechanism counts the number of incidents and suppresses dumps after a threshold value has been met.

The default behavior for flood control is to allow up to five dumps an hour per problem. Incidents over the threshold are still written to the ADR metadata, but there are no dump files created in the incident directory.

There are also additional sanity checks in place to prevent "flooding" the diagnostic framework. When a very large number of incidents of the same problem occur in a short period of time, the framework will stop recording incidents for the problem in ADR all together for some period of time.

Certain dumps are also flood-controlled

A similar mechanism controls dumps instead of incidents. The dump flood control is only done for a small set of specified dumps. One example is the system state dump, which can grow very large. The dump flood control allows dumps to be very detailed without the risk of using too much space.

HEALTH MONITOR

The Health Monitor facility allows components to run health checks to verify their state and assess the extent of any damage

Among the challenges when diagnosing an issue is to determine the scope of the problem, and the current health of the system. For instance, assume that the database issues an error indicating that a block in a table is corrupt. What is the state of other blocks in the same table? What about other blocks in the same data file? For another example, assume that Oracle Support suspects an inconsistency in the data dictionary. How do you determine the extent of the damage?

Health checks can be run on demand, or automatically in response to an incident

The Health Monitor functionality provides answers to questions like these. In the event of a block corruption incident, one of the actions run in the background is a block checker, which inspects the affected block and a range of neighboring blocks. This information can later be used by the Data Repair Advisor to determine what blocks and segments to operate on. In the case where a dictionary inconsistency is suspected, the DBA can invoke a dictionary health check from Support Workbench.

Any Health Monitor findings related to an incident are included as a report in the IPS package.

There are rules configured that automatically trigger health checks for incidents that match the conditions. It is also possible for the DBA to run health checks on demand. This can be done from Support Workbench, or using the PL/SQL package DBMS_HM.

Examples of available health checks:

- Database Structure Integrity Check
- Data Block Integrity Check
- Redo Integrity Check
- Undo Segment Integrity Check
- Transaction Integrity Check
- Dictionary Integrity Check

All health checks can be run when the database is open. In addition, a few can be run when the database is closed.

INCIDENT PACKAGING SERVICE (IPS)

The packaging of diagnostic data is automated, and intelligently correlates incidents and traces to determine what to include

In earlier versions, the diagnostic data sent to Oracle were typically either selected by customers, based on their understanding of the problem, or based on a "shopping list" from Oracle Support.

This is inconvenient for the customer, and increases the time until analysis can begin. It also introduces uncertainty. When some piece of information is missing, it's unclear why it was not included. Maybe it wasn't dumped at all, or maybe the user ended up not including it for some reason. By automating this, the user factor is removed.

The mechanism for selecting and packaging diagnostic information for transmission to Oracle is called the Incident Packaging Service (IPS).

IPS will intelligently gather and package diagnostic data for one or more incidents, specified by the user.

To improve the likelihood that the package includes sufficient information to diagnose the problem, it automatically includes additional files and incidents that appear to be possibly related to the main incidents.

There are multiple criteria for including files and incidents. Time is applicable to both files and incidents. The default setting is to include incidents that occurred within five minutes of main incidents, and trace files that were modified in the same interval.

Incidents will also be included if they match the process id or session id of the main incidents. Some criteria only apply to specific products. As an example, RDBMS incidents will be included if the process was executing a Parallel Query along with one of the main incident processes.

The physical package is a zip file containing a subset of the original ADR contents. It also contains a manifest, detailing the contents.

When an IPS package is unpacked, a valid ADR is recreated, containing a subset of the original ADR

When the physical package is unpacked on the Oracle side, a valid ADR directory structure is recreated. The ADR metadata is exported on the customer side, and imported when unpacking, which means that packages from different platforms can be unpacked on the same system.

Adding and Scrubbing Files

Any file can be added to an IPS package. For instance, if a customer wants to include a screenshot of some error message, a diagram of network topology or a complete test case, such files can be included and shipped with the package.

It's also possible to replace one file in a package with a modified copy. This is useful when a file contains sensitive data. Support Workbench has buttons for copying the file out of the package. When it has been modified as required, another button allows it to be copied back in. The modified copy is stored separately, so

that the original isn't lost. The modified copy is included in the package instead of the original.

User-Confirmed Actions

Heavy dumps must be approved by the DBA before they can be run

There are certain dumps that are considered too heavy to run without user confirmation. Examples would be dumping all blocks in the archive logs, getting configuration information or running the SQL Test Case Builder.

The output of these actions is written to the incident directory, so they will be included in the IPS package.

User-Reported Problems

Some parts of the workflow automation is available even for issues without explicit errors

Certain types of problems cannot be automatically detected. One example would be query performance issues. For such cases, the DBA can generate an incident based on that type.

For each type of incident, certain user-confirmed actions may be automatically recommended, which will help gather information about the problem. As an example, in the case of query performance issues, the SQL Test Case Builder will be automatically recommended.

ENTERPRISE MANAGER SUPPORT WORKBENCH

The Support Workbench provides a graphical interface for managing incidents, problems and packages

The graphical interface for the diagnosability features is Enterprise Manager's Support Workbench.

It provides a complete workflow for handling diagnostic issues – managing incidents and problems, packaging incidents, creating Service Requests (SR) and uploading packages. It's also possible to review the alert log and trace files within Support Workbench.

Enterprise Manager alerts can be configured to inform the DBA when an incident has occurred.

It's possible to add comments for problems, incidents and packages. The comments form a log where DBAs can note down what diagnostic action was taken at what point, and who requested or suggested it.

Quick Packaging and Advanced Packaging

Quick Packaging is a wizard that creates and uploads a package

Support Workbench provides a Quick Package wizard, which allows the DBA to create, generate and upload a package with a few simple clicks. For more complicated cases, an advanced packaging mode is also available. The advanced workflow allows the DBA to add and remove individual files or incidents, copy out files to scrub them, and so on.

As part of the packaging flow, Support Workbench automatically creates an SR, and uploads the package to Oracle.

Repair Advisors

Support Workbench also provides a graphical interface to different Repair Advisors that can be used to resolve some of the issues found by the diagnostic framework. Examples of such repair advisors are the Data Repair Advisor (DRA) and the SQL Repair Advisor.

Support Workbench automatically suggests corrective actions where applicable, and provides the right context so that the DBA can just go through a wizard to fix the issue.

RAC clusters

Enterprise Manager is aware of the cluster topology, and can automatically create correlated packages from other nodes

Since Enterprise Manager is aware of the cluster topology, it can automatically invoke IPS for each instance in a cluster, and create so-called correlated packages. A "main package" is created on the first RAC node. On the remaining nodes, a "correlated package" is created, which contains incidents related to the incidents in the main package.

Incident Meter

Support Workbench keeps track of the number of active incidents for a target

Support Workbench also keeps track of the number of active incidents in the system. The impacts of the incidents are used to determine if the incident is considered active or not.

Some impacts, like "out of shared memory" are considered transient. Transient incidents are considered active for 24 hours.

Incidents with other impacts, like "disk corruption", are considered active until the DBA explicitly closes the incident after resolving the corruption.

The Incident Meter is shown on the Enterprise Manager database home page. The number of incidents is displayed, along with an icon that turns red if there is at least one active incident with an impact that is considered critical.

Oracle Configuration Manager (OCM)

Support Workbench uses Oracle Configuration Manager (OCM) to handle the upload of packages to Oracle. It's important to ensure that OCM is configured. If not, the DBA will have to upload the packages to Metalink manually, which means that they will miss some of the end-to-end automation.

ADR COMMAND INTERPRETER

The command-line alternative to Support Workbench is called "adrci"

The command-line interface to ADR is called "adrci". It provides an alternative to Support Workbench for environments where EM isn't available. For products that aren't managed through EM, adrci is the main interface. For instance, the Instant Client includes adrci, since the client is not an EM target.

Some features are only available in adrci: accessing multiple homes simultaneously, and the ability to run scripts.

Multiple Homes

It's possible to access multiple homes simultaneously using adrci

The tool supports multiple homes simultaneously. This can be an efficient way of getting an overview of recent incidents. Assume that there are four instances in a RAC cluster, all sharing the same ADR_BASE. A single SHOW INCIDENT command can show incidents from all four instances, instead of the DBA having to issue the same command four times in four separate locations.

Scripting

The command-line interface to ADR can easily be scripted using Perl or shell scripts

The adrci can run scripts through the -script option. The output can be redirected into a file using the "spool" command. There are other commands familiar from sqlplus, such as "echo", "host" and "run".

It's also possible to write scripts that generate adrci scripts and parse the output. Some appropriate languages are Perl, Expect or shell scripts. This method is used for much of Oracle's internal automation of diagnostic processes.

Commands

The adrci command HELP shows available commands and examples of syntax

These are the most important commands when starting out with adrci:

Help – online help for all commands

Set home/base – point adrci to the correct base and home(s)

Show home/base – verify the current base and home(s)

Show incident – show incidents in the current home(s)

Show problem – show problems in the current home(s)

Show alert – show alert log entries in the current home(s) in an editor ($EDITOR on Unix)

Show tracefile – show the traces and dumps for a particular incident

Examples

These are examples of some convenient commands.

Show alert -tail – show alert log entries in the current home(s), and then display each new line added to the alert log (similar to the Unix command "tail -f")

Show alert -term – show alert log entries in the current terminal instead of in an editor window

Show incident -last 10 – show the last ten incidents

Show incident -p "problem_id=5" – show incidents for problem 5 only

Show problem -p "problem_key='ORA-04031'" – show the problem with problem key "ORA-4031"

CUSTOMER-SIDE PROCESS CHANGES

Most of the diagnostic framework is transparent to customers, and shouldn't require any significant changes to the daily operations. However, there are a few important things to be aware of.

Location of traces has changed

Traces, core dumps and alert log have been moved into a new repository

The traces files, dump files and logs have moved. They used to be in the directories pointed to by the initialization parameters USER_DUMP_DEST and BACKGROUND_DUMP_DEST. In the new framework, they're all inside ADR.

Traces (and the text form of the alert log) are in *adr_home*/trace.

The alert log (the XML form) is in *adr_home*/alert.

The incident dumps are in *adr_home*/incident/incdir_*incidentid*.

If customer scripts require some particular location for traces, symbolic links pointing to the ADR directories could be useful.

The database view V$DIAG_INFO shows the location of the currently used directories.

Traces and dumps have been separated

Normal traces are written to the trace directory

Traces are typically diagnostic information that's written at regular intervals to monitor the progress of a process. By contrast, dumps are generated when a failure occurs.

When a critical error occurs, the diagnostic dumps are written to a separate file in the incident directory

In previous versions, traces and dumps were written to the same files ("trace files"). In the new framework, traces are written to the trace directory, *adr_home*/trace. Incident dumps are written to the incident directory, *adr_home*/incident/incdir_*incidentid*.

Note that if a process encounters an incident, the process trace file includes a line that identifies the incident dump file, and shows the error. Conversely, the incident dump file includes a line that identifies the process trace file.

Alert log format has changed

There is now an XML-based alert log

There is a new, XML-based alert log format. It is located in the alert directory, *adr_home*/alert, and it's normally named log.xml.

The old text-style alert log is still available

However, the previous text-based alert log is still available in *adr_home*/trace, and it has the same name as in previous versions (which is port-dependent).

Diagnostic information is automatically purged

Diagnostic data is automatically purged

There is less need for DBA to keep track of the space used for diagnostic data, due to the automatic purging. Flood control and targeted dumping also contribute to keeping the space requirements down.

Diagnostic information can be automatically packaged and uploaded

Automatic packaging is the easiest and fastest way to get diagnostic data to Oracle

Diagnostic data should be sent to Oracle by using Support Workbench to automatically package and upload. This is the easiest way to do it, and it's the fastest way to get data to Oracle.

ORACLE-SIDE PROCESS CHANGES

IPS packages uploaded to Oracle are automatically unpacked and stored in a central location

The new diagnosability framework has allowed several processes inside Oracle to be more automated.

When customers upload IPS packages to Metalink, the packages are automatically unpacked. The contents are made available to support analysts through a centralized internal application. This means that support analysts can get to the diagnostic information faster.

The IPS package contains the Oracle Configuration Management (OCM) identifier, which allows Oracle analysts to quickly correlate the package contents with the stored configuration information.

There are several other areas that are automated already, or are in the process of being automated. One example is SQL test cases created by the SQL Test Case Builder, which can be automatically set up when a package is unpacked.

END-TO-END WORKFLOW

The new diagnostic framework provides an automated, complete workflow from the point of an error occurring to the point where Oracle Support analyzes it

All of the new features and enhancements above work together to improve the entire diagnostic workflow, from the point where an incident happens, until it is analyzed by Oracle.

The typical diagnostic workflow for a critical error in Oracle Database 11g is as follows:

- The critical error is considered an incident
- As part of the error processing, incident metadata is created in the ADR relations
- The error dump is written to a file in the incident directory inside ADR
- An alert is generated in EM concerning the new incident
- The DBA receives the alert using the configured notification method
- When logged in to EM, the alert includes a link to the incident page
- From the incident page, the DBA can take a number of actions, such as viewing traces, run checkers and user-confirmed actions, and creating packages
- When the DBA has concluded that the incident needs to be reviewed by Oracle Support, it is time to create an SR and generate a package
- For creating the package, the "Quick Package" wizard can be used, but there is also an advanced packaging mode that provides more features
- The SR is created automatically by Support Workbench
- When a package has been generated, it is automatically uploaded to Metalink
- On the Oracle side, the package is automatically unpacked into a central repository, and is immediately available to the Oracle analyst

CONCLUSION

The new diagnosability features make it easier and faster to detect, diagnose and repair problems

The new diagnosability features in Oracle Database 11g make it easier and faster to detect, diagnose and repair problems.

Database downtime is reduced, since customer problems can be resolved faster, and without waiting for the problem to occur again. DBAs are made more efficient by not having to spend as much time on the diagnostic process. Oracle Support can spend more time actually analyzing customer problems instead of gathering information to be analyzed.

Automatic Fault Diagnostics
November 2007
Author: Marcus Fallen, Yair Sarig

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com