23

ADRCI: ADR Command Interpreter

The Automatic Diagnostic Repository Command Interpreter (ADRCI) utility is a command-line tool that you use to manage Oracle Database diagnostic data.

> **Note:**
>
> Do not use `UIDRVCI.exe` file as it is used to access diagnostic data.

- About the ADR Command Interpreter (ADRCI) Utility
  The Automatic Diagnostic Repository Command Interpreter (ADRCI) is a command-line tool that is part of the Oracle Database fault diagnosability infrastructure.
- Definitions for Oracle Database ADRC
  To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.
- Starting ADRCI and Getting Help
  You can use ADRCI in interactive mode or batch mode.
- Setting the ADRCI Homepath Before Using ADRCI Commands
  When diagnosing a problem, you may want to work with diagnostic data from multiple database instances or components, or you may want to focus on diagnostic data from one instance or component.
- Viewing the Alert Log
  To view the ADR Command Interpreter alert log (ADRCI), use this procedure to view the alert log in your default editor.
- Finding Trace Files
  ADRCI enables you to view the names of trace files that are currently in the automatic diagnostic repository (ADR).
- Viewing Incidents
  The ADRCI `SHOW INCIDENT` command displays information about open Oracle Database incidents.
- Packaging Incidents
  You can use ADRCI commands to *package* one or more incidents for transmission to Oracle Support for analysis.
- ADRCI Command Reference
  Learn about the commands you can use with the Automatic Diagnostic Repository Command Interpreter (ADRCI).
- Troubleshooting ADRCI
  To assist troubleshooting, review some of the common ADRCI error messages, and their possible causes and remedies.

23.1 About the ADR Command Interpreter (ADRCI) Utility

The Automatic Diagnostic Repository Command Interpreter (ADRCI) is a command-line tool that is part of the Oracle Database fault diagnosability infrastructure.

The ADRCI utility assists you with diagnosing the cause of problems in your database (incidents). It can assist you with collecting data in an incident package that Oracle Support may need to help you to address the root cause of issues.

ADRCI assists you to do the following:

- View diagnostic data within the Automatic Diagnostic Repository (ADR).
- View Health Monitor reports.
- Package incident and problem information into a zip file for transmission to Oracle Support.

Diagnostic data includes incident and problem descriptions, trace files, dumps, health monitor reports, alert log entries, and more.

ADR data is secured by operating system permissions on the ADR directories, so there is no need to log in to `ADRCI`.

`ADRCI` has a rich command set. You can use these commands either in interactive mode, or within scripts.

> **Note:**
>
> The easier and recommended way to manage diagnostic data is with the Oracle Enterprise Manager Support Workbench (Support Workbench). ADRCI provides a command-line alternative to most of the functionality of the Support Workbench, and adds capabilities, such as listing and querying trace files.
>
> See *Oracle Database Administrator's Guide* for more information about the Oracle Database fault diagnosability infrastructure.

Related Topics

- *Oracle Database Administrator's Guide* Diagnosing and Resolving Problems

23.2 Definitions for Oracle Database ADRC

To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.

The following terms are associated with the Oracle Database automatic diagnostic repository incident fault diagnosability infrastructure (ADRCI), and the Oracle Database fault diagnosability infrastructure:

Automatic Diagnostic Repository (ADR)

The **Automatic Diagnostic Repository (ADR**) is a file-based repository for database diagnostic data such as traces, dumps, the alert log, health monitor reports, and more. It has a unified directory structure across multiple instances and multiple products. Beginning with Oracle Database 11g and later releases, Oracle Automatic Storage Management (Oracle

ASM), and other Oracle Database products or components store all diagnostic data in the ADR. Each instance of each product stores diagnostic data underneath its own ADR home directory. For example, in an Oracle Real Application Clusters (Oracle RAC) environment with shared storage and Oracle ASM, each database instance and each Oracle ASM instance has a home directory within the ADR. The ADR's unified directory structure enables customers and Oracle Support to correlate and analyze diagnostic data across multiple instances and multiple products.

Problem

A **problem** is a critical error in the database. Critical errors include internal errors, such as `ORA-00600` and other severe errors, such as `ORA-07445` (operating system exception) or `ORA-04031` (out of memory in the shared pool). Problems are tracked in the ADR. Each problem has a **problem key** and a unique **problem ID**.

Incident

An **incident** is a single occurrence of a problem. When a problem occurs multiple times, an incident is created for each occurrence. Incidents are tracked in the ADR. Each incident is identified by a numeric incident ID, which is unique within the ADR. When an incident occurs, the database makes an entry in the alert log, sends an **incident alert** to Oracle Enterprise Manager, gathers diagnostic data about the incident in the form of dump files (incident dumps), tags the incident dumps with the **incident ID**, and stores the incident dumps in an ADR subdirectory created for that incident.

Diagnosis and resolution of a critical error usually starts with an incident alert. You can obtain a list of all incidents in the ADR with an ADRCI command. Each incident is mapped to a single problem only.

Incidents are **flood-controlled**, so that a single problem does not generate too many incidents and incident dumps.

Problem Key

Every problem has a **problem key**, which is a text string that includes an error code (such as `ORA-600`) and in some cases, one or more error parameters. Two incidents are considered to have the same root cause if their problem keys match.

Incident Package

An **incident package (Package)** is a collection of data about incidents for one or more problems. Before sending incident data to Oracle Support, you must collect the date into a package, using the **Incident Packaging Service (IPS)**. After a package is created, you can add external files to the package, remove selected files from the package, or **scrub** (edit) selected files in the package to remove sensitive data.

A package is a logical construct only, until you create a physical file from the package contents. That is, an incident package starts out as a collection of metadata in the ADR. As you add and remove package contents, only the metadata is modified. When you are ready to upload the data to Oracle Support, you create a physical package by using ADRCI, which saves the data into a zip file. You can then upload the zip file to Oracle Support.

Finalizing

Before ADRCI can generate a physical package from a logical package, the package must be finalized. This means that other components are called to add any correlated diagnostic data files to the incidents already in this package. Finalizing also adds recent trace files, alert log entries, Health Monitor reports, SQL test cases, and configuration information. This step is run automatically when a physical package is generated, and can also be run manually using the

ADRCI utility. After manually finalizing a package, you can review the files that were added and then remove or edit any that contain sensitive information.

ADR Home

An **ADR home** is the root directory for all diagnostic data—traces, dumps, alert log, and so on—for a particular instance of a particular Oracle product or component. For example, in an Oracle RAC environment with Oracle ASM, each database instance and each Oracle ASM instance has an ADR home. All ADR homes share the same hierarchical directory structure. Some of the standard subdirectories in each ADR home include alert (for the alert log), trace (for trace files), and incident (for incident information). All ADR homes are located within the ADR base directory.

Some ADRCI commands can work with multiple ADR homes simultaneously. The current ADRCI **homepath** determines the ADR homes that are searched for diagnostic data when an ADRCI command is issued.

ADR Base

To permit correlation of diagnostic data across multiple ADR homes, ADR homes are grouped together under the same root directory called the **ADR base**. For example, in an Oracle RAC environment, the ADR base could be on a shared disk, and the ADR home for each Oracle RAC instance could be located under this ADR base.

The location of the ADR base for a database instance is set by the `DIAGNOSTIC_DEST` initialization parameter. If this parameter is omitted or is null, the database sets it to a default value.

When multiple database instances share an Oracle home, whether they are multiple single instances or the instances of an Oracle RAC database, and when one or more of these instances set ADR base in different locations, the last instance to start up determines the default ADR base for ADRCI.

Homepath

All ADRCI commands operate on diagnostic data in the **current** ADR homes. More than one ADR home can be current at any one time. Some ADRCI commands (such as `SHOW INCIDENT`) search for and display diagnostic data from all current ADR homes, while other commands require that only one ADR home be current, and display an error message if more than one are current.

The ADRCI **homepath** determines the ADR homes that are current. It does so by pointing to a directory within the ADR base hierarchy. If it points to a single ADR home directory, that ADR home is the only current ADR home. If the homepath points to a directory that is above the ADR home directory level in the hierarchy, all ADR homes that are below the directory that is pointed to become current.

The homepath is null by default when ADRCI starts. This means that all ADR homes under ADR base are current.

The `SHOW HOME` and `SHOW HOMEPATH` commands display a list of the ADR homes that are current, and the `SET HOMEPATH` command sets the homepath.

Related Topics

- *Oracle Database Administrator's Guide* About Incidents and Problems
- *Oracle Database Administrator's Guide*About Correlated Diagnostic Data in Incident Packages

23.3 Starting ADRCI and Getting Help

You can use ADRCI in interactive mode or batch mode.

Details are provided in the following sections:

- Using ADRCI in Interactive Mode
  When you use ADRCI in interactive mode to diagnose Oracle Database incidents, it prompts you to enter individual commands one at a time.
- Getting Help
  Learn how to obtain help when using the ADR Command Interpreter (ADRCI) Utility..
- Using ADRCI in Batch Mode
  Batch mode enables you to run a series of ADRCI commands using script or batch files, without being prompted for input.

23.3.1 Using ADRCI in Interactive Mode

When you use ADRCI in interactive mode to diagnose Oracle Database incidents, it prompts you to enter individual commands one at a time.

1. Ensure that the `ORACLE_HOME` and `PATH` environment variables are set properly.

   On Microsoft Windows platforms, these environment variables are set in the Windows registry automatically during installation. On other platforms, you must set and check environment variables with operating system commands.

   The `PATH` environment variable must include *`Oracle_home`*`/bin`

   .
2. Enter the following command at the operating system command prompt:

   ```
   ADRCI
   ```

   The utility starts and displays the following prompt:

   ```
   adrci>
   ```
3. Enter ADRCI commands, following each with the Enter key.
4. To exit ADRCI, Enter one of the following commands:

   ```
   EXIT
   QUIT
   ```

23.3.2 Getting Help

Learn how to obtain help when using the ADR Command Interpreter (ADRCI) Utility..

With the ADRCI help system, you can:

- View a list of ADR commands.
- View help for an individual command.
- View a list of ADRCI command line options.

To view a list of ADRCI commands

1. Start ADRCI in interactive mode.

2. At the ADRCI prompt, enter the following command:

```
HELP
```

To get help for a specific ADRCI command

1. Start ADRCI in interactive mode.
2. At the ADRCI prompt, enter the following command, where *command* is the ADRCI command about which you want more information:

```
HELP command
```

For example, to obtain help on the `SHOW TRACEFILE` command, enter the following:

```
HELP SHOW TRACEFILE
```

To view a list of command line options

- Enter the following command at the operating system command prompt:

```
ADRCI -HELP
```

The utility displays output similar to the following:

```
Syntax:
   adrci [-help] [script=script_filename] [exec="command [;command;...]"]

Options      Description                    (Default)
-----------------------------------------------------------------
script       script file name               (None)
help         help on the command options    (None)
exec         exec a set of commands         (None)
-----------------------------------------------------------------
```

Related Topics

- Using ADRCI in Interactive Mode
  When you use ADRCI in interactive mode to diagnose Oracle Database incidents, it prompts you to enter individual commands one at a time.

23.3.3 Using ADRCI in Batch Mode

Batch mode enables you to run a series of ADRCI commands using script or batch files, without being prompted for input.

To use batch mode, you add a command line parameter to the ADRCI command when you start `ADRCI`. Batch mode enables you to include ADRCI commands in shell scripts or Microsoft Windows batch files. As with interactive mode, the `ORACLE_HOME` and `PATH` environment variables must be set before starting ADRCI.

ADRCI Command Line Parameters for Batch Operation

The following command line parameters are available for batch operation:

Table 23-1 ADRCI Batch Operation Parameters

Parameter	Description
EXEC	Enables you to submit one or more ADRCI commands on the operating system command line that starts ADRCI. Commands are separated by semicolons (;).
SCRIPT	Enables you to run a script containing ADRCI commands.

How to Submit ADRCI Commands on the Command Line

- Enter the following command at the operating system command prompt:

  ```
  ADRCI EXEC="COMMAND[; COMMAND]..."
  ```

  For example, to run the SHOW HOMES command in batch mode, enter the following command at the operating system command prompt:

  ```
  ADRCI EXEC="SHOW HOMES"
  ```

  To run the SHOW HOMES command followed by the SHOW INCIDENT command, enter the following:

  ```
  ADRCI EXEC="SHOW HOMES; SHOW INCIDENT"
  ```

How to Run ADRCI Scripts:

Enter the following command at the operating system command prompt:

```
ADRCI SCRIPT=SCRIPT_FILE_NAME
```

For example, to run a script file named adrci_script.txt, enter the following command at the operating system command prompt:

```
ADRCI SCRIPT=adrci_script.txt
```

A script file contains a series of commands separated by semicolons (;) or line breaks. For example:

```
SET HOMEPATH diag/rdbms/orcl/orcl; SHOW ALERT -term
```

23.4 Setting the ADRCI Homepath Before Using ADRCI Commands

When diagnosing a problem, you may want to work with diagnostic data from multiple database instances or components, or you may want to focus on diagnostic data from one instance or component.

To work with diagnostic data from multiple instances or components, you must ensure that the ADR homes for all of these instances or components are *current*. To work with diagnostic data from only one instance or component, you must ensure that only the ADR home for that instance or component is current. You control the ADR homes that are current by setting the ADRCI homepath.

If multiple homes are current, this means that the homepath points to a directory in the ADR directory structure that contains multiple ADR home directories underneath it. To focus on a single ADR home, you must set the homepath to point lower in the directory hierarchy, to a single ADR home directory.

For example, if the Oracle RAC database with database name `orclbi` has two instances, where the instances have SIDs `orclbi1` and `orclbi2`, and Oracle RAC is using a shared Oracle home, the following two ADR homes exist:

```
/diag/rdbms/orclbi/orclbi1/
/diag/rdbms/orclbi/orclbi2/
```

In all ADRCI commands and output, ADR home directory paths (ADR homes) are always expressed relative to ADR base. So if ADR base is currently /u01/app/oracle, the absolute paths of these two ADR homes are the following:

```
/u01/app/oracle/diag/rdbms/orclbi/orclbi1/
/u01/app/oracle/diag/rdbms/orclbi/orclbi2/
```

You use the `SET HOMEPATH` command to set one or more ADR homes to be current. If ADR base is /u01/app/oracle and you want to set the homepath to /u01/app/oracle/diag/rdbms/orclbi/orclbi2/, you use this command:

```
adrci> set homepath diag/rdbms/orclbi/orclbi2
```

When ADRCI starts, the homepath is null by default, which means that all ADR homes under ADR base are current. In the previously cited example, therefore, the ADR homes for both Oracle RAC instances would be current.

```
adrci> show homes
ADR Homes:
diag/rdbms/orclbi/orclbi1
diag/rdbms/orclbi/orclbi2
```

In this case, any ADRCI command that you run, assuming that the command supports more than one current ADR home, works with diagnostic data from both ADR homes. If you were to set the homepath to /diag/rdbms/orclbi/orclbi2, only the ADR home for the instance with SID `orclbi2` would be current.

```
adrci> set homepath diag/rdbms/orclbi/orclbi2
adrci> show homes
ADR Homes:
diag/rdbms/orclbi/orclbi2
```

In this case, any ADRCI command that you run would work with diagnostic data from this single ADR home only.

See Also:

- *Oracle Database Administrator's Guide* for more information about the structure of ADR homes
- ADR Base
- ADR Home
- Homepath
- SET HOMEPATH
- SHOW HOMES

23.5 Viewing the Alert Log

To view the ADR Command Interpreter alert log (ADRCI), use this procedure to view the alert log in your default editor.

The alert log is written as both an XML-formatted file and as a text file. You can view either format of the file with any text editor, or you can run an ADRCI command to view the XML-formatted alert log with the XML tags omitted.

By default, ADRCI displays the alert log in your default editor. You can use the `SET EDITER` command to change your default editor.

To view the alert log with ADRCI:

1. Start ADRCI in interactive mode.
2. (Optional) Use the `SET HOMEPATH` command to select (make current) a single ADR home.

   You can use the `SHOW HOMES` command first to see a list of current ADR homes. See Homepath and Setting the ADRCI Homepath Before Using ADRCI Commands for more information.
3. At the ADRCI prompt, enter the following command:

   ```
   SHOW ALERT
   ```

   If more than one ADR home is current, you are prompted to select a single ADR home from a list. The alert log is displayed, with XML tags omitted, in your default editor.
4. Exit the editor to return to the ADRCI command prompt.

The following are variations on the `SHOW ALERT` command:

```
SHOW ALERT -TAIL
```

This displays the last portion of the alert log (the last 10 entries) in your terminal session.

```
SHOW ALERT -TAIL 50
```

This displays the last 50 entries in the alert log in your terminal session.

```
SHOW ALERT -TAIL -F
```

This displays the last 10 entries in the alert log, and then waits for more messages to arrive in the alert log. As each message arrives, it is appended to the display. This command enables you to perform *live monitoring* of the alert log. Press CTRL+C to stop waiting and return to the ADRCI prompt.

```
SPOOL /home/steve/MYALERT.LOG
SHOW ALERT -TERM
SPOOL OFF
```

This outputs the alert log, without XML tags, to the file `/home/steve/MYALERT.LOG`.

```
SHOW ALERT -P "MESSAGE_TEXT LIKE '%ORA-600%'"
```

This displays only alert log messages that contain the string 'ORA-600'. The output looks something like this:

```
ADR Home = /u01/app/oracle/product/11.1.0/db_1/log/diag/rdbms/orclbi/orclbi:
*************************************************************************
01-SEP-06 09.17.44.849000000 PM -07:00
AlertMsg1: ORA-600 dbgris01, addr=0xa9876541
```

Related Topics

- SHOW ALERT
  The ADRCI `SHOW ALERT` command shows the contents of the alert log in the default editor.

> **See Also:**
>
> - SHOW ALERT
> - SET EDITOR
> - *Oracle Database Administrator's Guide* for instructions for viewing the alert log with Oracle Enterprise Manager or with a text editor

23.6 Finding Trace Files

ADRCI enables you to view the names of trace files that are currently in the automatic diagnostic repository (ADR).

You can view the names of all trace files in the ADR, or you can apply filters to view a subset of names. For example, ADRCI has commands that enable you to:

- Obtain a list of trace files whose file name matches a search string.
- Obtain a list of trace files in a particular directory.
- Obtain a list of trace files that pertain to a particular incident.

You can combine filtering functions by using the proper command line parameters.

The `SHOW TRACEFILE` command displays a list of the trace files that are present in the trace directory and in all incident directories under the current ADR home. When multiple ADR homes are current, the traces file lists from all ADR homes are output one after another.

The following statement lists the names of all trace files in the current ADR homes, without any filtering:

```
SHOW TRACEFILE
```

The following statement lists the name of every trace file that has the string `mmon` in its file name. The percent sign (%) is used as a wildcard character, and the search string is case sensitive.

```
SHOW TRACEFILE %mmon%
```

This statement lists the name of every trace file that is located in the /home/steve/temp directory and that has the string `mmon` in its file name:

```
SHOW TRACEFILE %mmon% -PATH /home/steve/temp
```

This statement lists the names of trace files in reverse order of last modified time. That is, the most recently modified trace files are listed first.

```
SHOW TRACEFILE -RT
```

This statement lists the names of all trace files related to incident number 1681:

```
SHOW TRACEFILE -I 1681
```

> **See Also:**
>
> - SHOW TRACEFILE
> - *Oracle Database Administrator's Guide* for information about the directory structure of the ADR

23.7 Viewing Incidents

The ADRCI `SHOW INCIDENT` command displays information about open Oracle Database incidents.

When you submit a `SHOW INCIDENT` command, the ADRCI report shows the incident ID, problem key, and incident creation time for each incident. If you set the **homepath** (a directory within the ADR base hierarchy) so that there are multiple current ADR homes within that hierarchy location, then the report includes incidents from all of the ADR homes. See "Definitions for Oracle Database ADRC" for more information about homepath and other ADRCI terms.

1. Start ADRCI in interactive mode, and ensure that the homepath points to the correct directory within the ADR base directory hierarchy.
2. At the ADRCI prompt, enter the following command:

   ```
   SHOW INCIDENT
   ```

   ADRCI generates output similar to the following:

   ```
   ADR Home = /u01/app/oracle/product/11.1.0/db_1/log/diag/rdbms/orclbi/orclbi:
   ********************************************************************************
   INCIDENT_ID       PROBLEM_KEY               CREATE_TIME
   ----------------- ------------------------- ---------------------------------
   3808              ORA 603                   2010-06-18 21:35:49.322161 -07:00
   3807              ORA 600 [4137]            2010-06-18 21:35:47.862114 -07:00
   3805              ORA 600 [4136]            2010-06-18 21:35:25.012579 -07:00
   3804              ORA 1578                  2010-06-18 21:35:08.483156 -07:00
   4 rows fetched
   ```

The following are variations on the `SHOW INCIDENT` command:

```
SHOW INCIDENT -MODE BRIEF
SHOW INCIDENT -MODE DETAIL
```

These commands produce more detailed versions of the incident report. For example, to see a detailed incident report for incident 1681, enter the following command:

```
SHOW INCIDENT -MODE DETAIL -P "INCIDENT_ID=1681"
```

Related Topics

- ADRCI Command Reference
  Learn about the commands you can use with the Automatic Diagnostic Repository Command Interpreter (ADRCI).
- Definitions for Oracle Database ADRC
  To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.

23.8 Packaging Incidents

You can use ADRCI commands to *package* one or more incidents for transmission to Oracle Support for analysis.

Background information and instructions are presented in the following topics:

- About Packaging Incidents
  Packaging ADR Command Interpreter (ADRCI) incidents is a three-step process.
- Creating Incident Packages
  The following topics describe creating incident packages.

23.8.1 About Packaging Incidents

Packaging ADR Command Interpreter (ADRCI) incidents is a three-step process.

Step 1: Create a logical incident package.

The incident package (package) is denoted as logical, because it exists only as metadata in the automatic diagnostic repository (ADR). It has no content until you generate a physical package from the logical package. The logical package is assigned a package number, and you refer to it by that number in subsequent commands.

You can create the logical package as an empty package, or as a package based on an incident number, a problem number, a problem key, or a time interval. If you create the package as an empty package, then you can add diagnostic information to it in step 2.

Creating a package based on an incident means including diagnostic data—dumps, health monitor reports, and so on—for that incident. Creating a package based on a problem number or problem key means including in the package diagnostic data for incidents that reference that problem number or problem key. Creating a package based on a time interval means including diagnostic data on incidents that occurred in the time interval.

Step 2: Add diagnostic information to the incident package

If you created a logical package based on an incident number, a problem number, a problem key, or a time interval, this step is optional. You can add additional incidents to the package or you can add any file within the ADR to the package. If you created an empty package, you must use ADRCI commands to add incidents or files to the package.

Step 3: Generate the physical incident package

When you submit the command to generate the physical package, ADRCI gathers all required diagnostic files and adds them to a zip file in a designated directory. You can generate a complete zip file or an incremental zip file. An incremental file contains all the diagnostic files that were added or changed since the last zip file was created for the same logical package. You can create incremental files only after you create a complete file, and you can create as

many incremental files as you want. Each zip file is assigned a sequence number so that the files can be analyzed in the correct order.

Zip files are named according to the following scheme:

packageName_mode_sequence.zip

where:

- *packageName* consists of a portion of the problem key followed by a timestamp
- *mode* is either COM or INC, for complete or incremental
- *sequence* is an integer

For example, if you generate a complete zip file for a logical package that was created on September 6, 2006 at 4:53 p.m., and then generate an incremental zip file for the same logical package, you would create files with names similar to the following:

```
ORA603_20060906165316_COM_1.zip
ORA603_20060906165316_INC_2.zip
```

23.8.2 Creating Incident Packages

The following topics describe creating incident packages.

The ADRCI commands that you use to create a logical incident package (package) and generate a physical package are:

- Creating a Logical Incident Package
  You use variants of the IPS CREATE PACKAGE command to create a logical package (package).
- Adding Diagnostic Information to a Logical Incident Package
  After you have an existing logical package (**package**) configured for packaging incidents, you can add diagnostic information to that package.
- Generating a Physical Incident Package
  When you generate a package, you create a physical package (a zip file) for an existing logical package.

> **See Also:**
>
> About Packaging Incidents

23.8.2.1 Creating a Logical Incident Package

You use variants of the IPS CREATE PACKAGE command to create a logical package (package).

1. Start ADRCI in interactive mode, and ensure that the **homepath** (a directory within the ADR base hierarchy) points to the correct directory within the ADR base directory hierarchy for the database for which you want to create a logical package.

   See "Definitions for Oracle Database ADRC" for more information about homepath and other ADRCI terms.
2. At the ADRCI prompt, enter the following command:

```
IPS CREATE PACKAGE INCIDENT incident_number
```

For example, the following command creates a package based on incident 3:

```
IPS CREATE PACKAGE INCIDENT 3
```

ADRCI generates output similar to the following:

```
Created package 10 based on incident id 3, correlation level typical
```

The package number assigned to this logical package is 10.

The following are variations on the `IPS CREATE PACKAGE` command:

```
IPS CREATE PACKAGE
```

Entering the command without specifications creates an empty package. To add diagnostic data to the package before generating it, you then must use the `IPS ADD INCIDENT` or `IPS ADD FILE` commands.

```
IPS CREATE PACKAGE PROBLEM problem_ID
```

This command creates a package, and includes diagnostic information for incidents that reference the specified problem ID. (Problem IDs are integers.) You can obtain the problem ID for an incident from the report displayed by the `SHOW INCIDENT -MODE BRIEF` command. Because there can be many incidents with the same problem ID, ADRCI adds to the package the diagnostic information for the first three incidents (**early incidents**) that occurred and last three incidents (**late incidents**) that occurred with this problem ID, excluding any incidents that are older than 90 days.

> **Note:**
>
> The number of early and late incidents, and the 90-day age limit are defaults, which you can change. See IPS SET CONFIGURATION.

ADRCI may also add other incidents that correlate closely in time or in other criteria with the already added incidents.

```
IPS CREATE PACKAGE PROBLEMKEY "problem_key"
```

This command creates a package, and includes diagnostic information for incidents that reference the specified problem key. You can obtain problem keys from the report displayed by the `SHOW INCIDENT` command. Because there can be many incidents with the same problem key, ADRCI adds to the package only the diagnostic information for the first three early incidents, and the last three late incidents with this problem key, excluding incidents that are older than 90 days.

> **Note:**
>
> The number of early and late incidents, and the 90-day age limit are defaults, which you can change. See IPS SET CONFIGURATION.

ADRCI may also add other incidents that correlate closely in time or in other criteria with the already added incidents.

The problem key must be enclosed in single quotation marks (') or double quotation marks (") if it contains spaces or quotation marks.

```
IPS CREATE PACKAGE SECONDS sec
```

This creates a package and includes diagnostic information for all incidents that occurred from *sec* seconds ago until now. *sec* must be an integer.

```
IPS CREATE PACKAGE TIME 'start_time' TO 'end_time'
```

This command creates a package and includes diagnostic information for all incidents that occurred within the specified time range. *start_time* and *end_time* must be in the format 'YYYY-MM-DD HH24:MI:SS.FF TZR'. This string is a valid format string for the NLS_TIMESTAMP_TZ_FORMAT initialization parameter. The fraction (FF) portion of the time is optional, and the HH24:MI:SS delimiters can be either colons or periods.

For example, the following command creates a package with incidents that occurred between July 24th and July 30th of 2010:

```
IPS CREATE PACKAGE TIME '2010-07-24 00:00:00 -07:00' to '2010-07-30 23.59.59 -07:00'
```

Related Topics

- ADRCI Command Reference
  Learn about the commands you can use with the Automatic Diagnostic Repository Command Interpreter (ADRCI).
- Definitions for Oracle Database ADRC
  To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.
- IPS CREATE PACKAGE
  The ADRCI IPS CREATE PACKAGE command creates a new package. ADRCI automatically assigns the package number for the new package.

23.8.2.2 Adding Diagnostic Information to a Logical Incident Package

After you have an existing logical package (**package**) configured for packaging incidents, you can add diagnostic information to that package.

Adding diagnostic information to a logical package enables you to add incident information after the package is created, such the following:

- All diagnostic information for a particular incident
- A named file within the Automatic Diagnostic Repository (ADR).

1. Start ADRCI in interactive mode, and ensure that the **homepath** (a directory within the ADR base hierarchy) points to the correct directory within the ADR base directory hierarchy for the diagnostic information that you want to add.

   See "Definitions for Oracle Database ADRC" for more information about homepath and other ADRCI terms.

2. At the ADRCI prompt, enter the command for the diagnostic information that you want to add:

   ```
   To add all diagnostic information:

   IPS ADD INCIDENT incident_number PACKAGE package_number

   To add a file in the ADR to an existing package:
   ```

- At the ADRCI prompt, enter the following command:

  ```
  IPS ADD FILE filespec PACKAGE package_number
  ```

  filespec must be a fully qualified file name (with path). Only files that are within the ADR base directory hierarchy may be added.

Related Topics

- ADRCI Command Reference
  Learn about the commands you can use with the Automatic Diagnostic Repository Command Interpreter (ADRCI).
- Definitions for Oracle Database ADRC
  To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.

23.8.2.3 Generating a Physical Incident Package

When you generate a package, you create a physical package (a zip file) for an existing logical package.

1. Start ADRCI in interactive mode, and ensure that the homepath (a directory within the ADR base hierarchy) points to the correct directory within the ADR base directory hierarchy.

   See "Definitions for Oracle Database ADRC" for more information about homepath and other ADRCI terms.

2. At the ADRCI prompt, enter the command for the package information that you want to generate (complete or incremental):

   `To generate a complete physical package:`
   The following command generates a complete physical package (zip file) in the path you designate:

   ```
   IPS GENERATE PACKAGE package_number IN path
   ```

   For example, the following command creates a complete physical package in the directory `/home/steve/diagnostics` from logical package number 2:

   ```
   IPS GENERATE PACKAGE 2 IN /home/steve/diagnostics
   ```

   `To generate an incremental physical package`
   You can also generate an incremental package containing only the incidents that have occurred since the last package generation. At the ADRCI prompt, enter the following command:

   ```
   IPS GENERATE PACKAGE package_number IN path INCREMENTAL
   ```

Related Topics

- About Packaging Incidents
  Packaging ADR Command Interpreter (ADRCI) incidents is a three-step process.
- ADRCI Command Reference
  Learn about the commands you can use with the Automatic Diagnostic Repository Command Interpreter (ADRCI).
- Definitions for Oracle Database ADRC
  To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.

23.9 ADRCI Command Reference

Learn about the commands you can use with the Automatic Diagnostic Repository Command Interpreter (ADRCI).

There are four command types in ADRCI:

- Commands that work with one or more current ADR homes
- Commands that work with only one current ADR home, and that issue an error message if there is more than one current ADR home
- Commands that prompt you to select an ADR home when there are multiple current ADR homes
- Commands that do not need a current ADR home

All ADRCI commands support the case where there is a single current ADR home.

> **Note:**
>
> Unless otherwise specified, all commands work with multiple current ADR homes.

- CREATE REPORT
  The ADRCI `CREATE REPORT` command creates a report for the specified report type and run ID, and stores the report in the ADR.
- ECHO
  The ADRCI `ECHO` command prints the input string.
- EXIT
  The ADRCI `EXIT` command exits the ADRCI utility.
- HOST
  The ADRCI `HOST` command runs operating system commands without leaving ADRCI.
- IPS
  The ADRCI `IPS` command calls the Incident Packaging Service (IPS).
- PURGE
  The ADRCI `PURGE` command purges diagnostic data in the current ADR home, according to current purging policies.
- QUIT
  The ADRCI `QUIT` command is a synonym for the `EXIT` command.
- RUN
  The ADRCI `RUN` command runs an ADR Command Interpreter (ADRCI) script.
- SELECT
  The ADRCI `SELECT` command and its functions retrieve qualified diagnostic records for the specified incident or problem.
- SET BASE
  The ADRCI `SET BASE` command sets the ADR base to use in the current ADRCI session.
- SET BROWSER
  The ADRCI `SET BROWSER` command sets the default browser for displaying reports.

- SET CONTROL
  The ADRCI `SET CONTROL` command sets purging policies for Automatic Diagnostic Repository (ADR) contents.
- SET ECHO
  The ADRCI `SET ECHO` command turns command output on or off. This command only affects output being displayed in a script or using the spool mode.
- SET EDITOR
  The ADRCI `SET EDITOR` command sets the editor for displaying the alert log and the contents of trace files.
- SET HOMEPATH
  The ADRCI `SET HOMEPATH` command makes one or more ADR homes current. Many ADR commands work with the current ADR homes only.
- SET TERMOUT
  The ADRCI `SET TERMOUT` command turns output to the terminal on or off.
- SHOW ALERT
  The ADRCI `SHOW ALERT` command shows the contents of the alert log in the default editor.
- SHOW BASE
  The ADRCI `SET EDITOR` command shows the current ADR base.
- SHOW CONTROL
  The ADRCI `SHOW CONTROL` command displays information about the Automatic Diagnostic Repository (ADR), including the purging policy.
- SHOW HM_RUN
  The ADRCI `SHOW HM_RUN` command shows all information for Health Monitor runs.
- SHOW HOMEPATH
  The ADRCI `SHOW HOMEPATH` command is identical to the `SHOW HOMES` command.
- SHOW HOMES
  The ADRCI `SHOW HOMES` command shows the ADR homes in the current ADRCI session.
- SHOW INCDIR
  The ADRCI `SHOW INCDIR` command shows trace files for the specified incident.
- SHOW INCIDENT
  The ADRCI `SHOW INCIDENT` command lists all of the incidents associated with the current ADR home. Includes both open and closed incidents.
- SHOW LOG
  The ADRCI `SHOW LOG` command shows diagnostic log messages.
- SHOW PROBLEM
  The ADRCI `SHOW PROBLEM` command shows problem information for the current ADR home.
- SHOW REPORT
  The ADRCI `SET EDITOR` command shows a report for the specified report type and run name.
- SHOW TRACEFILE
  The ADRCI `SHOW TRACEFILE` command lists trace files.
- SPOOL
  The ADRCI `SET EDITOR` command directs ADRCI output to a file.

23.9.1 CREATE REPORT

The ADRCI `CREATE REPORT` command creates a report for the specified report type and run ID, and stores the report in the ADR.

Purpose

Creates a report for the specified report type and run ID, and stores the report in the ADR. Currently, only the `hm_run` (Health Monitor) report type is supported.

> **Note:**
>
> Results of Health Monitor runs are stored in the ADR in an internal format. To view these results, you must create a Health Monitor report from them and then view the report. You need create the report only once. You can then view it multiple times.

Syntax and Description

```
create report report_type run_name
```

The variable *report_type* must be `hm_run`. *run_name* is a Health Monitor run name. Obtain run names by using the command `SHOW HM_RUN`.

If the report already exists, then it is overwritten. To view the report, use the command `SHOW REPORT`.

This command does not support multiple ADR homes.

Example

This example creates a report for the Health Monitor run with run name `hm_run_1421`:

```
create report hm_run hm_run_1421
```

> **Note:**
>
> `CREATE REPORT REPORT` does not work when multiple ADR homes are set. To set a single ADR home as the target of the command, set the ADRCI home path before using the command.

Related Topics

- SHOW HM_RUN
  The ADRCI `SHOW HM_RUN` command shows all information for Health Monitor runs.
- SHOW REPORT
  The ADRCI `SET EDITOR` command shows a report for the specified report type and run name.

- Setting the ADRCI Homepath Before Using ADRCI Commands
  When diagnosing a problem, you may want to work with diagnostic data from multiple database instances or components, or you may want to focus on diagnostic data from one instance or component.

23.9.2 ECHO

The ADRCI `ECHO` command prints the input string.

Purpose

Prints the input string. You can use this command to print custom text from ADRCI scripts.

Syntax and Description

`ECHO quoted_string`

The string must be enclosed in single or double quotation marks.

This command does not require an ADR home to be set before you can use it.

Example

These examples print the string "Hello, world!":

```
ECHO "Hello, world!"


ECHO 'Hello, world!'
```

23.9.3 EXIT

The ADRCI `EXIT` command exits the ADRCI utility.

Purpose

Exits the ADRCI utility.

Syntax and Description

```
exit
```

`EXIT` is a synonym for the `QUIT` command.

This command does not require an ADR home to be set before you can use it.

23.9.4 HOST

The ADRCI `HOST` command runs operating system commands without leaving ADRCI.

Purpose

Runs operating system commands without leaving ADRCI.

Syntax and Description

`host ["host_command_string"]`

Use `host` by itself to enter an operating system shell, which allows you to enter multiple operating system commands. Enter `EXIT` to leave the shell and return to ADRCI.

You can also specify the command on the same line (*`host_command_string`*) enclosed in double quotation marks.

This command does not require an ADR home to be set before you can use it.

Examples

```
host


host "ls -l *.pl"
```

23.9.5 IPS

The ADRCI `IPS` command calls the Incident Packaging Service (IPS).

Purpose

Calls the Incident Packaging Service (IPS). The IPS command provides options for creating logical incident packages (packages), adding diagnostic data to packages, and generating physical packages for transmission to Oracle Support.

> **Note:**
>
> IPS commands do not work when multiple ADR homes are set. For information about setting a single ADR home, see Setting the ADRCI Homepath Before Using ADRCI Commands.

- Using the <ADR_HOME> and <ADR_BASE> Variables in IPS Commands
  The ADRCI `IPS` command set provides shortcuts for referencing the current ADR home and ADR base directories.
- IPS ADD
  The ADRCI `IPS ADD` command adds incidents to a package.
- IPS ADD FILE
  The ADRCI `IPS ADD FILE` command adds a file to an existing package.
- IPS ADD NEW INCIDENTS
  The ADRCI `IPS ADD NEW INCIDENTS` command finds and adds new incidents for all of the problems in the specified package.
- IPS COPY IN FILE
  The ADRCI `IPS COPY IN FILE` command copies a file into the ADR from the external file system.
- IPS COPY OUT FILE
  The ADRCI `IPS COPY OUT FILE` command copies a file from the ADR to the external file system.

- IPS CREATE PACKAGE
  The ADRCI `IPS CREATE PACKAGE` command creates a new package. ADRCI automatically assigns the package number for the new package.
- IPS DELETE PACKAGE
  The ADRCI `IPS DELETE PACKAGE` command drops a package and its contents from the ADR.
- IPS FINALIZE
  The ADRCI `IPS FINALIZE` command finalizes a package before uploading.
- IPS GENERATE PACKAGE
  The ADRCI `IPS GENERATE PACKAGE` command creates a physical package (a zip file) in a target directory.
- IPS GET MANIFEST
  The ADRCI `IPS GET MANIFEST` command extracts the manifest from a package zip file and displays it.
- IPS GET METADATA
  The ADRCI `IPS GET METADATA` command extracts ADR-related metadata from a package file and displays it.
- IPS PACK
  The ADRCI `IPS PACK` command creates a package, and generates the physical package immediately
- IPS REMOVE
  The ADRCI `IPS REMOVE` command removes incidents from an existing package.
- IPS REMOVE FILE
  The ADRCI `IPS REMOVE FILE` command removes a file from an existing package.
- IPS SET CONFIGURATION
  The ADRCI `IPS SET CONFIGURATION` command changes the value of an IPS configuration parameter.
- IPS SHOW CONFIGURATION
  The ADRCI `IPS SHOW CONFIGURATION` command displays a list of IPS configuration parameters and their values.
- IPS SHOW FILES
  The ADRCI `IPS SHOW FILES` command lists files included in the specified package.
- IPS SHOW INCIDENTS
  The ADRCI `IPS SHOW INCIDENTS` command lists incidents included in the specified package.
- IPS SHOW PACKAGE
  The ADRCI `IPS SHOW PACKAGE` command displays information about the specified package.
- IPS UNPACK FILE
  The ADRCI `IPS UNPACK FILE` command unpacks a physical package file into the specified path.

> **See Also:**
>
> Packaging Incidents for more information about packaging

23.9.5.1 Using the <ADR_HOME> and <ADR_BASE> Variables in IPS Commands

The ADRCI `IPS` command set provides shortcuts for referencing the current ADR home and ADR base directories.

To access the current ADR home directory, use the `<ADR_HOME>` variable. For example:

```
ips add file <ADR_HOME>/trace/orcl_ora_13579.trc package 12
```

Use the `<ADR_BASE>` variable to access the ADR base directory. For example:

```
ips add file <ADR_BASE>/diag/rdbms/orcl/orcl/trace/orcl_ora_13579.trc package
12
```

> **Note:**
>
> Type the angle brackets (`< >`) as shown.

23.9.5.2 IPS ADD

The ADRCI `IPS ADD` command adds incidents to a package.

Purpose

Adds incidents to a package.

Syntax and Description

```
ips add {incident first [n] | incident inc_id | incident last [n] |
     problem first [n] | problem prob_id | problem last [n] |
     problemkey pr_key | seconds secs | time start_time to end_time}
     package package_id
```

The following table describes the arguments of `IPS ADD`.

Table 23-2 Arguments of IPS ADD command

Argument	Description
`incident first [n]`	Adds the first *n* incidents to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the first five incidents are added. If *n* is omitted, then the default is 1, and the first incident is added.
`incident inc_id`	Adds an incident with ID *inc_id* to the package.
`incident last [n]`	Adds the last *n* incidents to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the last five incidents are added. If *n* is omitted, then the default is 1, and the last incident is added.

Table 23-2 (Cont.) Arguments of IPS ADD command

Argument	Description
`problem first [n]`	Adds the incidents for the first *n* problems to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the incidents for the first five problems are added. If *n* is omitted, then the default is 1, and the incidents for the first problem is added. Adds only the first three early incidents and last three late incidents for each problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problem prob_id`	Adds all incidents with problem ID *prob_id* to the package. Adds only the first three early incidents and last three late incidents for the problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problem last [n]`	Adds the incidents for the last *n* problems to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the incidents for the last five problems are added. If *n* is omitted, then the default is 1, and the incidents for the last problem is added. Adds only the first three early incidents and last three late incidents for each problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problemkey pr_key`	Adds incidents with problem key *pr_key* to the package. Adds only the first three early incidents and last three late incidents for the problem key, excluding any older than 90 days. (Note: These limits are defaults and can be changed.)
`seconds secs`	Adds all incidents that have occurred within *secs*- seconds of the present time.
`time start_time to end_time`	Adds all incidents between *start_time* and *end_time* to the package. Time format is 'YYYY-MM-YY HH24:MI:SS.FF TZR'. Fractional part (FF) is optional.
`package package_id`	Specifies the package to which to add incidents.

Examples

This example adds incident 22 to package 12:

```
ips add incident 22 package 12
```

This example adds the first three early incidents and the last three late incidents with problem ID 6 to package 2, exuding any incidents older than 90 days:

```
ips add problem 6 package 2
```

This example adds all incidents taking place during the last minute to package 5:

```
ips add seconds 60 package 5
```

This example adds all incidents taking place between 10:00 A.M. and 11:00 P.M. on May 1, 2020:

```
ips add  time '2020-05-01 10:00:00.00 -07:00' to '2020-05-01 23:00:00.00 -07:00'
```

23.9.5.3 IPS ADD FILE

The ADRCI `IPS ADD FILE` command adds a file to an existing package.

Syntax and Description

```
ips add file file_name package package_id
```

file_name is the full path name of the file. You can use the `<ADR_HOME>` and `<ADR_BASE>` variables if desired. The file must be under the same ADR base as the package.

package_id is the package ID.

Example

This example adds a trace file to package 12:

```
ips add file <ADR_HOME>/trace/orcl_ora_13579.trc package 12
```

Related Topics

- Using the <ADR_HOME> and <ADR_BASE> Variables in IPS Commands
  The ADRCI `IPS` command set provides shortcuts for referencing the current ADR home and ADR base directories.

23.9.5.4 IPS ADD NEW INCIDENTS

The ADRCI `IPS ADD NEW INCIDENTS` command finds and adds new incidents for all of the problems in the specified package.

Syntax and Description

```
ips add new incidents package package_id
```

package_id is the ID of the package to update. Only new incidents of the problems in the package are added.

Example

This example adds up to three of the new late incidents for the problems in package 12:

```
ips add new incidents package 12
```

> **Note:**
>
> The number of late incidents added is a default that can be changed.

Related Topics

- IPS SET CONFIGURATION
  The ADRCI `IPS SET CONFIGURATION` command changes the value of an IPS configuration parameter.

23.9.5.5 IPS COPY IN FILE

The ADRCI `IPS COPY IN FILE` command copies a file into the ADR from the external file system.

Purpose

To edit a file in a package, you must copy the file out to a designated directory, edit the file, and copy it back into the package. For example, you can use this command to delete sensitive data in the file before sending the package to Oracle Support.

Syntax and Description

```
ips copy in file filename [to new_name][overwrite] package package_id
     [incident incid]
```

Copies an external file, *filename* (specified with full path name) into the ADR, associating it with an existing package, *package_id*, and optionally an incident, *incid*. Use the `to new_name` option to give the copied file a new file name within the ADR. Use the `overwrite` option to overwrite a file that exists already.

Example

This example copies a trace file from the file system into the ADR, associating it with package 2 and incident 4:

```
ips copy in file /home/nick/trace/orcl_ora_13579.trc to <ADR_HOME>/trace/
orcl_ora_13579.trc package 2 incident 4
```

Related Topics

- Using the <ADR_HOME> and <ADR_BASE> Variables in IPS Commands
  The ADRCI `IPS` command set provides shortcuts for referencing the current ADR home and ADR base directories.
- IPS SHOW FILES
  The ADRCI `IPS SHOW FILES` command lists files included in the specified package.

23.9.5.6 IPS COPY OUT FILE

The ADRCI `IPS COPY OUT FILE` command copies a file from the ADR to the external file system.

Purpose

To edit a file in a package, you must copy the file out to a designated directory, edit the file, and copy it back into the package. You may want to do this to delete sensitive data in the file before sending the package to Oracle Support.

Syntax and Description

```
ips copy out file source to target [overwrite]
```

Copies a file, *source*, to a location outside the ADR, *target* (specified with full path name). Use the `overwrite` option to overwrite the file that exists already.

Example

This example copies the file `orcl_ora_13579.trc`, in the trace subdirectory of the current ADR home, to a local folder.

```
ips copy out file <ADR_HOME>/trace/orcl_ora_13579.trc to /home/nick/trace/
orcl_ora_13579.trc
```

Related Topics

- Using the <ADR_HOME> and <ADR_BASE> Variables in IPS Commands
  The ADRCI `IPS` command set provides shortcuts for referencing the current ADR home and ADR base directories.
- IPS SHOW FILES
  The ADRCI `IPS SHOW FILES` command lists files included in the specified package.

23.9.5.7 IPS CREATE PACKAGE

The ADRCI `IPS CREATE PACKAGE` command creates a new package. ADRCI automatically assigns the package number for the new package.

Purpose

Creates a new package. ADRCI automatically assigns the package number for the new package.

Syntax and Description

```
ips create package {incident first [n] | incident inc_id |
     incident last [n] | problem first [n] | problem prob_id |
     problem last [n] | problemkey prob_key | seconds secs |
     time start_time to end_time} [correlate {basic |typical | all}]
```

(Optional) You can add incidents to the new package using the provided options.

Table 23-3 describes the arguments for `IPS CREATE PACKAGE`.

Table 23-3 Arguments of IPS CREATE PACKAGE command

Argument	Description
`incident first [n]`	Adds the first *n* incidents to the package, where *n* is a positive integer. For example, if *n* is set to `5`, then the first five incidents are added. If *n* is omitted, then the default is `1`, and the first incident is added.
`incident inc_id`	Adds an incident with ID *inc_id* to the package.

Table 23-3 (Cont.) Arguments of IPS CREATE PACKAGE command

Argument	Description
`incident last [n]`	Adds the last *n* incidents to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the last five incidents are added. If *n* is omitted, then the default is 1, and the last incident is added.
`problem first [n]`	Adds the incidents for the first *n* problems to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the incidents for the first five problems are added. If *n* is omitted, then the default is 1, and the incidents for the first problem is added. Adds only the first three early incidents and last three late incidents for each problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problem prob_id`	Adds all incidents with problem ID *prob_id* to the package. Adds only the first three early incidents and last three late incidents for the problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problem last [n]`	Adds the incidents for the last *n* problems to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the incidents for the last five problems are added. If *n* is omitted, then the default is 1, and the incidents for the last problem is added. Adds only the first three early incidents and last three late incidents for each problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problemkey pr_key`	Adds all incidents with problem key *pr_key* to the package. Adds only the first three early incidents and last three late incidents for the problem key, excluding any older than 90 days. (Note: These limits are defaults and can be changed.)
`seconds secs`	Adds all incidents that have occurred within *secs* seconds of the present time.
`time start_time to end_time`	Adds all incidents taking place between *start_time* and *end_time* to the package. Time format is 'YYYY-MM-YY HH24:MI:SS.FF TZR'. Fractional part (FF) is optional.
`correlate {basic |typical | all}`	Selects a method of including correlated incidents in the package. There are three options for this argument: • `correlate basic` includes incident dumps and incident process trace files. • `correlate typical` includes incident dumps and any trace files that were modified within five minutes of each incident. You can alter the time interval by modifying the `INCIDENT_TIME_WINDOW` configuration parameter. • `correlate all` includes the incident dumps, and all trace files that were modified between the time of the first selected incident and the last selected incident. The default value is `correlate typical`.

Examples

This example creates a package with no incidents:

```
ips create package
```

Output:

```
Created package 5 without any contents, correlation level typical
```

This example creates a package containing all incidents between 10 AM and 11 PM on the given day:

```
ips create package time '2010-05-01 10:00:00.00 -07:00' to '2010-05-01 23:00:00.00
-07:00'
```

Output:

```
Created package 6 based on time range 2010-05-01 10:00:00.00 -07:00 to 2010-05-01
23:00:00.00 -07:00, correlation level typical
```

This example creates a package and adds the first three early incidents and the last three late incidents with problem ID 3, excluding incidents that are older than 90 days:

```
ips create package problem 3
```

Output:

```
Created package 7 based on problem id 3, correlation level typical
```

> **Note:**
>
> The number of early and late incidents added, and the 90-day age limit are defaults that can be changed.

Related Topics

- IPS SET CONFIGURATION
  The ADRCI `IPS SET CONFIGURATION` command changes the value of an IPS configuration parameter.
- Creating Incident Packages
  The following topics describe creating incident packages.

23.9.5.8 IPS DELETE PACKAGE

The ADRCI `IPS DELETE PACKAGE` command drops a package and its contents from the ADR.

Syntax and Description

```
ips delete package package_id
```

package_id is the package to delete.

Example

```
ips delete package 12
```

23.9.5.9 IPS FINALIZE

The ADRCI `IPS FINALIZE` command finalizes a package before uploading.

Syntax and Description

```
ips finalize package package_id
```

package_id is the package ID to finalize.

Example

```
ips finalize package 12
```

> **See Also:**
>
> *Oracle Database Administrator's Guide* for more information about finalizing packages

23.9.5.10 IPS GENERATE PACKAGE

The ADRCI `IPS GENERATE PACKAGE` command creates a physical package (a zip file) in a target directory.

Syntax and Description

```
ips generate package package_id [in path] [complete | incremental]
```

package_id is the ID of the package to generate. Optionally, you can save the file in the directory *path*. Otherwise, the package is generated in the current working directory.

The `complete` option means the package forces ADRCI to include all package files. This is the default behavior.

The `incremental` option includes only files that have been added or changed since the last time that this package was generated. With the `incremental` option, the command finishes more quickly.

Example

This example generates a physical package file in path `/home/steve`:

```
ips generate package 12 in /home/steve
```

This example generates a physical package from files added or changed since the last generation:

```
ips generate package 14 incremental
```

> **See Also:**
>
> Generating a Physical Incident Package

23.9.5.11 IPS GET MANIFEST

The ADRCI `IPS GET MANIFEST` command extracts the manifest from a package zip file and displays it.

Syntax and Description

```
ips get manifest from file filename
```

filename is a package zip file. The manifest is an XML-formatted set of metadata for the package file, including information about ADR configuration, correlated files, incidents, and how the package was generated.

This command does not require an ADR home to be set before you can use it.

Example

```
ips get manifest from file /home/steve/ORA603_20060906165316_COM_1.zip
```

23.9.5.12 IPS GET METADATA

The ADRCI `IPS GET METADATA` command extracts ADR-related metadata from a package file and displays it.

Syntax and Description

```
ips get metadata {from file filename | from adr}
```

filename is a package zip file. The metadata in a package file (stored in the file `metadata.xml`) contains information about the ADR home, ADR base, and product.

Use the `from adr` option to get the metadata from a package zip file that has been unpacked into an ADR home using `IPS UNPACK`.

The `from adr` option requires an ADR home to be set.

Example

This example displays metadata from a package file:

```
ips get metadata from file /home/steve/ORA603_20060906165316_COM_1.zip
```

This next example displays metadata from a package file that was unpacked into the directory /scratch/oracle/package1:

```
set base /scratch/oracle/package1
ips get metadata from adr
```

In this previous example, upon receiving the `SET BASE` command, ADRCI automatically adds to the homepath the ADR home that was created in `/scratch/oracle/package1` by the `IPS UNPACK FILE` command.

> **See Also:**
>
> IPS UNPACK FILE for more information about unpacking package files

23.9.5.13 IPS PACK

The ADRCI `IPS PACK` command creates a package, and generates the physical package immediately

Purpose

Creates a package, and generates the physical package immediately.

Syntax and Description

```
ips pack [incident first [n] | incident inc_id | incident last [n] |
     problem first [n] | problem prob_id | problem last [n] |
     problemkey prob_key | seconds secs | time start_time to end_time]
     [correlate {basic |typical | all}] [in path]
```

ADRCI automatically generates the package number for the new package. `IPS PACK` creates an empty package if no package contents are specified.

Table 23-4 describes the arguments for `IPS PACK`.

Table 23-4 Arguments of IPS PACK command

Argument	Description
`incident first [n]`	Adds the first *n* incidents to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the first five incidents are added. If *n* is omitted, then the default is 1, and the first incident is added.
`incident inc_id`	Adds an incident with ID *inc_id* to the package.
`incident last [n]`	Adds the last *n* incidents to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the last five incidents are added. If *n* is omitted, then the default is 1, and the last incident is added.

Table 23-4 (Cont.) Arguments of IPS PACK command

Argument	Description
`problem first [n]`	Adds the incidents for the first *n* problems to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the incidents for the first five problems are added. If *n* is omitted, then the default is 1, and the incidents for the first problem is added. Adds only the first three early incidents and last three late incidents for each problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problem prob_id`	Adds all incidents with problem ID *prob_id* to the package. Adds only the first three early incidents and last three late incidents for the problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problem last [n]`	Adds the incidents for the last *n* problems to the package, where *n* is a positive integer. For example, if *n* is set to 5, then the incidents for the last five problems are added. If *n* is omitted, then the default is 1, and the incidents for the last problem is added. Adds only the first three early incidents and last three late incidents for each problem, excluding any older than 90 days. (Note: These limits are defaults and can be changed. See "IPS SET CONFIGURATION".)
`problemkey pr_key`	Adds incidents with problem key *pr_key* to the package. Adds only the first three early incidents and last three late incidents for the problem key, excluding any older than 90 days. (Note: These limits are defaults and can be changed.)
`seconds secs`	Adds all incidents that have occurred within *secs* seconds of the present time.
`time start_time to end_time`	Adds all incidents taking place between *start_time* and *end_time* to the package. Time format is 'YYYY-MM-YY HH24:MI:SS.FF TZR'. Fractional part (FF) is optional.
`correlate {basic |typical | all}`	Selects a method of including correlated incidents in the package. There are three options for this argument: • `correlate basic` includes incident dumps and incident process trace files. • `correlate typical` includes incident dumps and any trace files that were modified within five minutes of each incident. You can alter the time interval by modifying the `INCIDENT_TIME_WINDOW` configuration parameter. • `correlate all` includes the incident dumps, and all trace files that were modified between the time of the first selected incident and the last selected incident. The default value is `correlate typical`.
`in path`	Saves the physical package to directory *path*.

Example

This example creates an empty package:

```
ips pack
```

This example creates a physical package containing all information for incident 861:

```
ips pack incident 861
```

This example creates a physical package for all incidents in the last minute, fully correlated:

```
ips pack seconds 60 correlate all
```

Related Topics

- IPS SET CONFIGURATION
  The ADRCI `IPS SET CONFIGURATION` command changes the value of an IPS configuration parameter.

23.9.5.14 IPS REMOVE

The ADRCI `IPS REMOVE` command removes incidents from an existing package.

Purpose

Removes incidents from an existing package.

Syntax and Description

```
ips remove {incident inc_id | problem prob_id | problemkey prob_key}
     package package_id
```

After removing incidents from a package, the incidents continue to be tracked within the package metadata to prevent ADRCI from automatically including them later (such as with `ADD NEW INCIDENTS`).

The following table describes the arguments of `IPS REMOVE`.

Table 23-5 Arguments of IPS REMOVE command

Argument	Description
`incident inc_id`	Removes the incident with ID *inc_id* from the package
`problem prob_id`	Removes all incidents with problem ID *prob_id* from the package
`problemkey pr_key`	Removes all incidents with problem key *pr_key* from the package
`package package_id`	Removes incidents from the package with ID *package_id*.

Example

This example removes incident 22 from package 12:

```
ips remove incident 22 package 12
```

Related Topics

- IPS GET MANIFEST
  The ADRCI `IPS GET MANIFEST` command extracts the manifest from a package zip file and displays it.

23.9.5.15 IPS REMOVE FILE

The ADRCI `IPS REMOVE FILE` command removes a file from an existing package.

Syntax and Description

```
ips remove file file_name package package_id
```

`file_name` is the file to remove from package *`package_id`*. The complete path of the file must be specified. (You can use the `<ADR_HOME>` and `<ADR_BASE>` variables if desired.)

After removal, the file continues to be tracked within the package metadata to prevent ADRCI from automatically including it later (such as with `ADD NEW INCIDENTS`). Removing a file, therefore, only sets the `EXCLUDE` flag for the file to `Explicitly excluded`.

Example

This example removes a trace file from package 12:

```
ips remove file <ADR_HOME>/trace/orcl_ora_13579.trc package 12
Removed file <ADR_HOME>/trace/orcl_ora_13579.trc from package 12
ips show files package 12

.
.
.
FILE_ID                 4
FILE_LOCATION           <ADR_HOME>/trace
FILE_NAME               orcl_ora_13579.trc
LAST_SEQUENCE           0
EXCLUDE                 Explicitly excluded
.
.
.
```

See Also:

- IPS GET MANIFEST for information about package metadata
- Using the <ADR_HOME> and <ADR_BASE> Variables in IPS Commands for information about the `<ADR_BASE>` directory syntax
- IPS SHOW FILES

23.9.5.16 IPS SET CONFIGURATION

The ADRCI `IPS SET CONFIGURATION` command changes the value of an IPS configuration parameter.

Syntax and Description

```
ips set configuration {parameter_id | parameter_name} value
```

parameter_id is the ID of the parameter to change, and *parameter_name* is the name of the parameter to change. *value* is the new value. For a list of the configuration parameters and their IDs, use `IPS SHOW CONFIGURATION`.

Example

```
ips set configuration 3 10
```

Related Topics

- IPS SHOW CONFIGURATION
  The ADRCI `IPS SHOW CONFIGURATION` command displays a list of IPS configuration parameters and their values.

23.9.5.17 IPS SHOW CONFIGURATION

The ADRCI `IPS SHOW CONFIGURATION` command displays a list of IPS configuration parameters and their values.

Purpose

These parameters control various thresholds for IPS data, such as timeouts and incident inclusion intervals.

Syntax and Description

```
ips show configuration {parameter_id | parameter_name}]
```

`IPS SHOW CONFIGURATION` lists the following information for each configuration parameter:

- Parameter ID
- Name
- Description
- Unit used by parameter (such as days or hours)
- Value
- Default value
- Minimum Value
- Maximum Value
- Flags

Optionally, you can get information about a specific parameter by supplying a *parameter_id* or a *parameter_name*.

Example

This command describes all IPS configuration parameters:

```
ips show configuration
```

Output:

```
PARAMETER INFORMATION:
   PARAMETER_ID          1
   NAME                  CUTOFF_TIME
   DESCRIPTION           Maximum age for an incident to be considered for
                         inclusion
   UNIT                  Days
   VALUE                 90
   DEFAULT_VALUE         90
   MINIMUM               1
   MAXIMUM               4294967295
   FLAGS                 0

PARAMETER INFORMATION:
   PARAMETER_ID          2
   NAME                  NUM_EARLY_INCIDENTS
   DESCRIPTION           How many incidents to get in the early part of the
range
   UNIT                  Number
   VALUE                 3
   DEFAULT_VALUE         3
   MINIMUM               1
   MAXIMUM               4294967295
   FLAGS                 0

PARAMETER INFORMATION:
   PARAMETER_ID          3
   NAME                  NUM_LATE_INCIDENTS
   DESCRIPTION           How many incidents to get in the late part of the
range
   UNIT                  Number
   VALUE                 3
   DEFAULT_VALUE         3
   MINIMUM               1
   MAXIMUM               4294967295
   FLAGS                 0

PARAMETER INFORMATION:
   PARAMETER_ID          4
   NAME                  INCIDENT_TIME_WINDOW
   DESCRIPTION           Incidents this close to each other are considered
                         correlated
   UNIT                  Minutes
   VALUE                 5
   DEFAULT_VALUE         5
```

```
   MINIMUM                  1
   MAXIMUM                  4294967295
   FLAGS                    0

PARAMETER INFORMATION:
   PARAMETER_ID             5
   NAME                     PACKAGE_TIME_WINDOW
   DESCRIPTION              Time window for content inclusion is from x hours
                            before first included incident to x hours after
last
                            incident
   UNIT                     Hours
   VALUE                    24
   DEFAULT_VALUE            24
   MINIMUM                  1
   MAXIMUM                  4294967295
   FLAGS                    0

PARAMETER INFORMATION:
   PARAMETER_ID             6
   NAME                     DEFAULT_CORRELATION_LEVEL
   DESCRIPTION              Default correlation level for packages
   UNIT                     Number
   VALUE                    2
   DEFAULT_VALUE            2
   MINIMUM                  1
   MAXIMUM                  4
   FLAGS                    0
```

Examples

This command describes configuration parameter `NUM_EARLY_INCIDENTS`:

```
ips show configuration num_early_incidents
```

This command describes configuration parameter 3:

```
ips show configuration 3
```

Configuration Parameter Descriptions

The following table describes the IPS configuration parameters in detail.

Table 23-6 IPS Configuration Parameters

Parameter	ID	Description
CUTOFF_TIME	1	Maximum age, in days, for an incident to be considered for inclusion.
NUM_EARLY_INCIDENTS	2	Number of incidents to include in the early part of the range when creating a package based on a problem. By default, ADRCI adds the three earliest incidents and three most recent incidents to the package.

Table 23-6 (Cont.) IPS Configuration Parameters

Parameter	ID	Description
NUM_LATE_INCIDENTS	3	Number of incidents to include in the late part of the range when creating a package based on a problem. By default, ADRCI adds the three earliest incidents and three most recent incidents to the package.
INCIDENT_TIME_WINDOW	4	Number of minutes between two incidents in order for them to be considered correlated.
PACKAGE_TIME_WINDOW	5	Number of hours to use as a time window for including incidents in a package. For example, a value of 5 includes incidents five hours before the earliest incident in the package, and five hours after the most recent incident in the package.
DEFAULT_CORRELATION_LEVEL	6	The default correlation level to use for correlating incidents in a package. The correlation levels are: • 1 (basic): includes incident dumps and incident process trace files. • 2 (typical): includes incident dumps and any trace files that were modified within the time window specified by INCIDENT_TIME_WINDOW (see above). • 4 (all): includes the incident dumps, and all trace files that were modified between the first selected incident and the last selected incident. Additional incidents can be included automatically if they occurred in the same time range.

Related Topics

- IPS SET CONFIGURATION
  The ADRCI `IPS SET CONFIGURATION` command changes the value of an IPS configuration parameter.

23.9.5.18 IPS SHOW FILES

The ADRCI `IPS SHOW FILES` command lists files included in the specified package.

Purpose

Lists files included in the specified package.

Syntax and Description

```
ips show files package package_id
```

package_id is the package ID to display.

Example

This example shows all files associated with package 1:

```
ips show files package 1
```

Output:

```
   FILE_ID                 1
   FILE_LOCATION           <ADR_HOME>/alert
   FILE_NAME               log.xml
   LAST_SEQUENCE           1
   EXCLUDE                 Included

   FILE_ID                 2
   FILE_LOCATION           <ADR_HOME>/trace
   FILE_NAME               alert_adcdb.log
   LAST_SEQUENCE           1
   EXCLUDE                 Included

   FILE_ID                 27
   FILE_LOCATION           <ADR_HOME>/incident/incdir_4937
   FILE_NAME               adcdb_ora_692_i4937.trm
   LAST_SEQUENCE           1
   EXCLUDE                 Included

   FILE_ID                 28
   FILE_LOCATION           <ADR_HOME>/incident/incdir_4937
   FILE_NAME               adcdb_ora_692_i4937.trc
   LAST_SEQUENCE           1
   EXCLUDE                 Included

   FILE_ID                 29
   FILE_LOCATION           <ADR_HOME>/trace
   FILE_NAME               adcdb_ora_692.trc
   LAST_SEQUENCE           1
   EXCLUDE                 Included

   FILE_ID                 30
   FILE_LOCATION           <ADR_HOME>/trace
   FILE_NAME               adcdb_ora_692.trm
   LAST_SEQUENCE           1
   EXCLUDE                 Included
.
.
.
```

23.9.5.19 IPS SHOW INCIDENTS

The ADRCI `IPS SHOW INCIDENTS` command lists incidents included in the specified package.

Syntax and Description

`ips show incidents package` *`package_id`*

`package_id` is the package ID to display.

Example

This example lists the incidents in package 1:

```
ips show incidents package 1
```

Output:

```
MAIN INCIDENTS FOR PACKAGE 1:
   INCIDENT_ID            4985
   PROBLEM_ID             1
   EXCLUDE                Included

CORRELATED INCIDENTS FOR PACKAGE 1:
```

23.9.5.20 IPS SHOW PACKAGE

The ADRCI `IPS SHOW PACKAGE` command displays information about the specified package.

Syntax and Description

```
ips show package package_id {basic | brief | detail}
```

package_id is the ID of the package to display.

Use the `basic` option to display a minimal amount of information. It is the default when no *package_id* is specified.

Use the `brief` option to display more information about the package than the `basic` option. It is the default when a *package_id* is specified.

Use the `detail` option to show the information displayed by the `brief` option, as well as some package history and information about the included incidents and files.

Example

```
ips show package 12

ips show package 12 brief
```

23.9.5.21 IPS UNPACK FILE

The ADRCI `IPS UNPACK FILE` command unpacks a physical package file into the specified path.

Syntax and Description

```
ips unpack file file_name [into path]
```

file_name is the full path name of the physical package (zip file) to unpack. Optionally, you can unpack the file into directory *path*, which must exist, and muste be writable. If you omit the

path, then the current working directory is used. The destination directory is treated as an ADR base, and the entire ADR base directory hierarchy is created, including a valid ADR home.

This command does not require an ADR home to be set before you can use it.

Example

```
ips unpack file /tmp/ORA603_20060906165316_COM_1.zip into /tmp/newadr
```

23.9.6 PURGE

The ADRCI `PURGE` command purges diagnostic data in the current ADR home, according to current purging policies.

Purpose

Purges diagnostic data in the current ADR home, according to current purging policies. Only ADR contents that are due to be purged are purged.

Diagnostic data in the ADR has a default lifecycle. For example, information about incidents and problems is subject to purging after one year, whereas the associated dump files (dumps) are subject to purging after only 30 days.

Some Oracle products, such as Oracle Database, automatically purge diagnostic data at the end of its life cycle. Other products and components require you to purge diagnostic data manually with this command. You can also use this command to purge data that is due to be automatically purged.

The SHOW CONTROL command displays the default purging policies for short-lived ADR contents and long-lived ADR contents.

Syntax and Description

```
purge [-i {id | start_id end_id} |
  -age mins [-type {ALERT|INCIDENT|TRACE|CDUMP|HM|UTSCDMP}]]
```

The following table describes the flags for `PURGE`.

Table 23-7 Flags for the PURGE command

Flag	Description
`-i {id1 | start_id end_id}`	Purges either a specific incident ID (*id*) or a range of incident IDs (*start_id* and *end_id*)
`-age mins`	Purges only data older than *mins* minutes.

Table 23-7 (Cont.) Flags for the PURGE command

Flag	Description
`-type {ALERT|INCIDENT|TRACE|CDUMP|HM|UTSCDMP}`	Specifies the type of diagnostic data to purge. Used with the `-age` clause. The following types can be specified: • `ALERT` - Alert logs • `INCIDENT` - Incident data • `TRACE` - Trace files (including dumps) • `CDUMP` - Core dump files • `HM` - Health Monitor run data and reports • `UTSCDMP` - Dumps of in-memory traces for each session The `UTSCDMP` data is stored in directories under the trace directory. Each of these directories is named cdmp_*timestamp*. In response to a critical error (such as an ORA-600 or ORA-7445 error), a background process creates such a directory and writes each session's in-memory tracing data into a trace file. This data might be useful in determining what the instance was doing in the seconds leading up to the failure.

Examples

This example purges all diagnostic data in the current ADR home based on the default purging policies:

```
purge
```

This example purges all diagnostic data for all incidents between 123 and 456:

```
purge -i 123 456
```

This example purges all incident data from before the last hour:

```
purge -age 60 -type incident
```

> **Note:**
>
> `PURGE` does not work when multiple ADR homes are set. For information about setting a single ADR home, see "Setting the ADRCI Homepath Before Using ADRCI Commands".

23.9.7 QUIT

The ADRCI `QUIT` command is a synonym for the `EXIT` command.

Related Topics

- EXIT
  The ADRCI `EXIT` command exits the ADRCI utility.

23.9.8 RUN

The ADRCI `RUN` command runs an ADR Command Interpreter (ADRCI) script.

Syntax and Description

```
run script_name

@ script_name

@@ script_name
```

The variable *script_name* is the file containing the ADRCI commands that you want to run. ADRCI looks for the script in the current directory, unless a full path name is supplied. If the file name is given without a file extension, then ADRCI uses the default extension `.adi`.

The `run` and `@` commands are synonyms. The `@@` command is similar to `run` and `@`. However, when used inside a script, `@@` uses the path of the calling script to locate *script_name*, rather than the current directory.

You are not required to have an ADR home set before you can use the `run` command.

Example

```
run my_script

@my_script
```

23.9.9 SELECT

The ADRCI `SELECT` command and its functions retrieve qualified diagnostic records for the specified incident or problem.

Purpose

Retrieves qualified records for the specified incident or problem, to assist with diagnosing the issue.

Syntax and Description

```
select {*|[field1, [field2, ...]} FROM {incident|problem}
  [WHERE predicate_string]
  [ORDER BY field1 [, field2, ...] [ASC|DSC|DESC]]
  [GROUP BY field1 [, field2, ...]]
  [HAVING having_predicate_string]
```

Table 23-8 Flags for the SELECT command

Flag	Description
field1, *field2*, ...	Lists the fields to retrieve. If * is specified, then all fields are retrieved.
incident|problem	Indicates whether to query incidents or problems.

Table 23-8 (Cont.) Flags for the SELECT command

Flag	Description
`WHERE "predicate_string"`	Uses a SQL-like predicate string to show only the incident or problem for which the predicate is true. The predicate string must be enclosed in double quotation marks. SHOW INCIDENT lists the fields that can be used in the predicate string incidents. SHOW PROBLEM lists the fields that can be used in the predicate string for problems.
`ORDER BY field1, field2, ... [ASC|DSC|DESC]`	Show results sorted by field in the given order, as well as in ascending (`ASC`) and descending order (`DSC` or `DESC`). When the `ORDER BY` clause is specified, results are shown in ascending order by default.
`GROUP BY field1, field2, ...`	Show results grouped by the specified fields. The `GROUP BY` flag groups rows but does not guarantee the order of the result set. To order the groupings, use the `ORDER BY` flag.
`HAVING "having_predicate_string"`	Restrict the groups of returned rows to those groups for which the having predicate is true. The `HAVING` flag must be used in combination with the `GROUP BY` flag.

Note:

The `WHERE`, `ORDER BY`, `GROUP BY`, and `HAVING` flags are similar to the clauses with the same names in a `SELECT` SQL statement.

See *Oracle Database SQL Language Reference* for more information about the clauses in a `SELECT` SQL statement.

Restrictions

The following restrictions apply when you use the `SELECT` command:

- The command cannot join more than two tables.
- The command cannot use table aliases.
- The command can use only a limited set of functions, which are listed in this section.
- The command cannot use column wildcard (`"*"`) when joining tables or when using the `GROUP BY` clause.
- Statements must be on a single line.
- Statement cannot have subqueries.
- Statement cannot have a `WITH` clause.
- A limited set of pseudocolumns are allowed. For example, `ROWNUM` is allowed, but `ROWID` is not allowed.

Examples

This example retrieves the `incident_id` and `create_time` for incidents with an `incident_id` greater than `1`:

```
select incident_id, create_time from incident where incident_id > 1
```

The following is an example of output for this query:

```
INCIDENT_ID          CREATE_TIME
-------------------- ---------------------------------------
4801                 2011-05-27 10:10:26.541656 -07:00
4802                 2011-05-27 10:11:02.456066 -07:00
4803                 2011-05-27 10:11:04.759654 -07:00
```

This example retrieves the `problem_id` and `first_incident` for each problem with a `problem_key` that includes `600`:

```
select problem_id, first_incident from problem where problem_key like '%600%'
```

The following is an example of output for this query:

```
PROBLEM_ID           FIRST_INCIDENT
-------------------- --------------------
1                    4801
2                    4802
3                    4803
```

Functions

This section describes functions that you can use with the `SELECT` command.

The purpose and syntax of these functions are similar to the corresponding SQL functions, but there are some differences between SQL functions and the functions used with the ADRCI utility.

The following restrictions apply to all of the ADRCI functions:

- The expressions must be simple expressions.

  See *Oracle Database SQL Language Reference* for information about simple expressions.
- You cannot combine function calls. For example, the following combination of function calls is not supported:

  ```
  sum(length(column_name))
  ```
- No functions are overloaded.
- All function arguments are mandatory.
- The functions cannot be used with other ADRCI Utility commands.
- AVG
  The `AVG` function of the `ADRC SELECT` command returns the average value of an expression.
- CONCAT
  The `CONCAT` function of the `ADRC SELECT` command returns a concatenation of two character strings.
- COUNT
  The `COUNT` function of the ADRC `SELECT` command returns the number of rows returned by a query.
- DECODE
  The `DECODE` function of the `ADRC SELECT` command compares an expression to each search value one by one.

- LENGTH
  The `LENGTH` function of the `ADRC SELECT` command returns the length of a character string using as defined by the input character set.
- MAX
  The `MAX` function of the `ADRC SELECT` command returns the maximum value of an expression.
- MIN
  The `MIN` function of the `ADRC SELECT` command returns the minimum value of an expression.
- NVL
  The `NVL` function of the `ADRC SELECT` command replaces null (returned as a blank) with character data in the results of a query.
- REGEXP_LIKE
  The `REGEXP_LIKE` function of the `ADRC SELECT` command returns rows that match a specified pattern in a specified regular expression.
- SUBSTR
  The `SUBSTR` function of the `ADRC SELECT` command returns a portion of character data.
- SUM
  The `SUM` function of the `ADRC SELECT` command returns the sum of values of an expression.
- TIMESTAMP_TO_CHAR
  The `TIMESTAMP_TO_CHAR` function of the `ADRC SELECT` command converts a value of `TIMESTAMP` data type to a value of `VARCHAR2` data type in a specified format.
- TOLOWER
  The `TOLOWER` function of the `ADRC SELECT` command returns character data, with all letters lowercase.
- TOUPPER
  The `TOUPPER` function of the `ADRC SELECT` command returns character data, with all letters uppercase.

23.9.9.1 AVG

The `AVG` function of the `ADRC SELECT` command returns the average value of an expression.

Purpose

Returns the average value of an expression.

Syntax

See the description of `AVG` in *Oracle Database SQL Language Reference*.

Restrictions

The following restrictions apply when you use the `AVG` function in the `SELECT` command:

- The expression must be a numeric column or a positive numeric constant.
- The function does not support the `DISTINCT` or `ALL` keywords.
- The function does not support the `OVER` clause.

Related Topics

- *Oracle Database SQL Language Reference* AVG

23.9.9.2 CONCAT

The `CONCAT` function of the `ADRC SELECT` command returns a concatenation of two character strings.

Purpose

Returns a concatenation of two character strings. The character data can be of the data types `CHAR` and `VARCHAR2`. The return value is the same data type as the character data.

Syntax

See the description of `CONCAT` in *Oracle Database SQL Language Reference*.

Restrictions

The following restrictions apply when you use the `CONCAT` function in the `SELECT` command:

- The function does not support LOB data types, including `BLOB`, `CLOB`, `NCLOB`, and `BFILE` data types.
- The function does not support national character set data types, including `NCHAR`, `NVARCHAR2`, and `NCLOB` data types.

Related Topics

- *Oracle Database SQL Language Reference* CONCAT

23.9.9.3 COUNT

The `COUNT` function of the ADRC `SELECT` command returns the number of rows returned by a query.

Purpose

Returns the number of rows returned by the query.

Syntax

See the description of `COUNT` in *Oracle Database SQL Language Reference*.

Restrictions

The following restrictions apply when you use the `COUNT` function in the `SELECT` command:

- The expression must be a column, a numeric constant, or a string constant.
- The function does not support the `DISTINCT` or `ALL` keywords.
- The function does not support the `OVER` clause.
- The function always counts all rows for the query, including duplicates and nulls.

Examples

This example returns the number of incidents for which `flood_controlled` is `0` (zero):

```
select count(*) from incident where flood_controlled = 0;
```

This example returns the number of problems for which `problem_key` includes `ORA-600`:

```
select count(*) from problem where problem_key like '%ORA-600%';
```

Related Topics

- *Oracle Database SQL Language Reference* COUNT

23.9.9.4 DECODE

The `DECODE` function of the `ADRC SELECT` command compares an expression to each search value one by one.

Purpose

Compares an expression to each search value one by one. If the expression is equal to a search, then Oracle Database returns the corresponding result. If no match is found, then the database returns the specified default value.

Syntax

See the description of `DECODE` in *Oracle Database SQL Language Reference*.

Restrictions

The following restrictions apply when you use the `DECODE` function in the `SELECT` command:

- The search arguments must be character data.
- A default value must be specified.

Example

This example shows each `incident_id` and whether or not the incident is flood-controlled. The example uses the `DECODE` function to display text instead of numbers for the `flood_controlled` field.

```
select incident_id, decode(flood_controlled, 0, \
  "Not flood-controlled", "Flood-controlled") from incident;
```

Related Topics

- *Oracle Database SQL Language Reference* DECODE

23.9.9.5 LENGTH

The `LENGTH` function of the `ADRC SELECT` command returns the length of a character string using as defined by the input character set.

Purpose

Returns the length of a character string using as defined by the input character set. The character string can be any of the data types `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, `CLOB`, or `NCLOB`. The return value is of data type `NUMBER`. If the character sting has data type `CHAR`, then the length includes all trailing blanks. If the character string is null, then this function returns 0 (zero).

> **Note:**
>
> The SQL function returns null if the character string is null.

Syntax

See the description of `LENGTH` in *Oracle Database SQL Language Reference*.

Restrictions

The `ADRC SELECT` command does not support the following functions: `LENGTHB`, `LENGTHC`, `LENGTH2`, and `LENGTH4`.

Example

This example shows the `problem_id` and the length of the `problem_key` for each problem.

```
select problem_id, length(problem_key) from problem;
```

Related Topics

- *Oracle Database SQL Language Reference* LENGTH

23.9.9.6 MAX

The `MAX` function of the `ADRC SELECT` command returns the maximum value of an expression.

Syntax

See MAX in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `MAX` function in the `SELECT` command:

- The function does not support the `DISTINCT` or `ALL` keywords.
- The function does not support the `OVER` clause.

Example

This example shows the maximum `last_incident` value for all of the recorded problems.

```
select max(last_incident) from problem;
```

23.9.9.7 MIN

The `MIN` function of the `ADRC SELECT` command returns the minimum value of an expression.

Syntax

See MIN in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `MIN` function in the `SELECT` command:

- The function does not support the `DISTINCT` or `ALL` keywords.
- The function does not support the `OVER` clause.

Example

This example shows the minimum `first_incident` value for all of the recorded problems.

```
select min(first_incident) from problem;
```

23.9.9.8 NVL

The `NVL` function of the `ADRC SELECT` command replaces null (returned as a blank) with character data in the results of a query.

Purpose

If the first expression specified is null, then `NVL` returns second expression specified. If first expression specified is not null, then `NVL` returns the value of the first expression.

Syntax

See NVL in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `NVL` function in the `SELECT` command:

- The replacement value (second expression) must be specified as character data.
- The function does not support data conversions.

Example

This example replaces `NULL` in the output for `signalling_component` with the text "No component."

```
select nvl(signalling_component, 'No component') from incident;
```

23.9.9.9 REGEXP_LIKE

The `REGEXP_LIKE` function of the `ADRC SELECT` command returns rows that match a specified pattern in a specified regular expression.

Purpose

In SQL, `REGEXP_LIKE` is a condition instead of a function.

Syntax

See REGEXP_LIKE Condition in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `REGEXP_LIKE` function in the `SELECT` command:

- The pattern match is always case-sensitive.

- The function does not support the `match_param` argument.

Example

This example shows the `problem_id` and `problem_key` for all problems where the `problem_key` ends with a number.

```
select problem_id, problem_key from problem \
  where regexp_like(problem_key, '[0-9]$') = true
```

23.9.9.10 SUBSTR

The `SUBSTR` function of the `ADRC SELECT` command returns a portion of character data.

Purpose

The portion of data returned begins at the specified position and is the specified substring length characters long. `SUBSTR` calculates lengths using characters as defined by the input character set.

Syntax

See SUBSTR in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `SUBSTR` function in the `SELECT` command:

- The function supports only positive integers. It does not support negative values or floating-point numbers.
- The `SELECT` command does not support the following functions: `SUBSTRB`, `SUBSTRC`, `SUBSTR2`, and `SUBSTR4`.

Example

This example shows each `problem_key` starting with the fifth character in the key.

```
select substr(problem_key, 5) from problem;
```

23.9.9.11 SUM

The `SUM` function of the `ADRC SELECT` command returns the sum of values of an expression.

Syntax

See SUM in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `SUM` function in the `SELECT` command:

- The expression must be a numeric column or a numeric constant.
- The function does not support the `DISTINCT` or `ALL` keywords.
- The function does not support the `OVER` clause.

23.9.9.12 TIMESTAMP_TO_CHAR

The `TIMESTAMP_TO_CHAR` function of the `ADRC SELECT` command converts a value of `TIMESTAMP` data type to a value of `VARCHAR2` data type in a specified format.

Purpose

If you do not specify a format, then the function converts values to the default timestamp format.

Syntax

See the syntax of the `TO_CHAR` function (TO_CHAR (datetime)) in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `TIMESTAMP_TO_CHAR` function in the `SELECT` command:

- The function converts only `TIMESTAMP` data type. `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`, and other data types are not supported.
- The function does not support the `nlsparm` argument. The function uses the default language for your session.

Example

This example converts the create_time for each incident from a `TIMESTAMP` data type to a `VARCHAR2` data type in the `DD-MON-YYYY` format.

```
select timestamp_to_char(create_time, 'DD-MON-YYYY') from incident;
```

23.9.9.13 TOLOWER

The `TOLOWER` function of the `ADRC SELECT` command returns character data, with all letters lowercase.

Purpose

The character data can be of the data types `CHAR` and `VARCHAR2`. The return value is the same data type as the character data. The database sets the case of the characters based on the binary mapping defined for the underlying character set.

Syntax

See the syntax of the `LOWER` function (LOWER) in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `TOLOWER` function in the `SELECT` command:

- The function does not support LOB data types, including `BLOB`, `CLOB`, `NCLOB`, and `BFILE` data types.
- The function does not support national character set data types, including `NCHAR`, `NVARCHAR2`, and `NCLOB` data types.

Example

This example shows each `problem_key` in all lowercase letters.

```
select tolower(problem_key) from problem;
```

23.9.9.14 TOUPPER

The `TOUPPER` function of the `ADRC SELECT` command returns character data, with all letters uppercase.

Purpose

The character data can be of the data types `CHAR` and `VARCHAR2`. The return value is the same data type as the character data. The database sets the case of the characters based on the binary mapping defined for the underlying character set.

Syntax

See the syntax of the `UPPER` function (UPPER) in *Oracle Database SQL Language Reference*

Restrictions

The following restrictions apply when you use the `TOUPPER` function in the `SELECT` command:

- The function does not support LOB data types, including `BLOB`, `CLOB`, `NCLOB`, and `BFILE` data types.
- The function does not support national character set data types, including `NCHAR`, `NVARCHAR2`, and `NCLOB` data types.

Example

This example shows each `problem_key` in all uppercase letters.

```
select toupper(problem_key) from problem;
```

23.9.10 SET BASE

The ADRCI `SET BASE` command sets the ADR base to use in the current ADRCI session.

Syntax and Description

```
set base base_str
```

`base_str` is a full path to a directory. The format for *`base_str`* depends on the operating system. If there are valid ADR homes under the base directory, these homes are added to the home path of the current ADRCI session.

This command does not require an ADR home to be set before you can use it.

Example

```
set base /u01/app/oracle
```

Related Topics

- Definitions for Oracle Database ADRC
  To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.

23.9.11 SET BROWSER

The ADRCI `SET BROWSER` command sets the default browser for displaying reports.

Syntax and Description

> **Note:**
>
> This command is reserved for future use. At this time ADRCI does not support HTML-formatted reports in a browser.

```
set browser browser_program
```

browser_program is the browser program name (it is assumed the browser can be started from the current ADR working directory). If no browser is set, then ADRCI displays reports to the terminal or spool file.

This command does not require an ADR home to be set before you can use it.

Example

```
set browser mozilla
```

> **See Also:**
>
> - SHOW REPORT for more information about showing reports
> - SPOOL for more information about spooling

23.9.12 SET CONTROL

The ADRCI `SET CONTROL` command sets purging policies for Automatic Diagnostic Repository (ADR) contents.

Purpose

Sets time limit and size limit controls that manage when ADR repository files are purged.

Syntax and Description

```
set control (purge_policy = value purge_policy = value, ...)
```

In the preceding syntax, the variable *purge_policy* can be `SHORTP_POLICY`, `LONGP_POLICY`, or `SIZEP_POLICY`.

For SHORTP_POLICY and LONGP_POLICY, *value* is the number of hours after which the ADR contents become eligible for purging. The controls SHORTP_POLICY and LONGP_POLICY are not mutually exclusive. Each policy controls different types of content.

For SIZEP_POLICY, *value* is the size limit that you want to set for the ADR home. If you do not set a value, then the ADR home is purged every 24 hours. If you set a value for SIZEP_POLICY, then a MMON task is set that checks the current status of that limit every four hours. When the ADR home size reaches that limit, the ADR home is purged.

This command works with a single ADR home only.

Use SET CONTROL to set the following purge attributes:

Attribute Name	Description
SHORTP_POLICY	Number of hours after which to purge ADR contents that have a short life. Default: 720 (30 days). A setting of 0 (zero) means that all contents that have a short life can be purged. The maximum setting is 35791394. If a value greater than 35791394 is specified, then this attribute is set to 0 (zero). The ADR contents that have a short life include the following: • Trace files, including those files stored in the cdmp_*timestamp* subdirectories • Core dump files • Packaging information
LONGP_POLICY	Number of hours after which to purge ADR contents that have a long life. Default is 8760 (365 days). A setting of 0 (zero) means that all contents that have a long life can be purged. The maximum setting is 35791394. If a value greater than 35791394 is specified, then this attribute is set to 0 (zero). The ADR contents that have a long life include the following: • Incident information • Incident dumps • Alert logs
SIZEP_POLICY	(Optional) Defines the size limit for an Automatic Diagnostic Repository (ADR) home. In Oracle Database 12c Release 2 (12.2) and later releases, you can use SIZEP_POLICY to set a size limit for the AWR. When you set SIZEP_POLICY, the MMON background process collects statistics for the AWR home. By default, the ADR home is purged every 24 hours. If this purge time frame is inadequate, then you can set the SIZEP_POLICY to define a size limit for an ADR home to purge the ADR home when it approaches the purge size threshold. When you set a size limit using SIZEP_POLICY, MMON checks the current status of that limit every four hours. If the size limit is reached, then ADR purges the ADR repository.
PURGE_THRESHOLD	The PURGE_THRESHOLD value is a value at which the SIZEP_POLICY is triggered. If you set SIZEP_POLICY, then by default, the value of PURGE_THRESHOLD is 95 percent of the value of the SIZEP_POLICY. In a multitenant environment, the ADR home is shared, so the PURGE_THRESHOLD size policy is applied to the diagnostics storage location (diag). You can tune PURGE_THRESHOLD independently for each ADR home by setting the value for the PURGE_THRESHOLD column in the ADR_CONTROL_AUX relation . When you tune the PURGE_THRESHOLD, this can assist you with keeping the amount of ADR data below the SIZEP_POLICY limit, even if your ADR home is purged infrequently.

Example

Suppose the ADR purge policy is set to the default values of 720 for short life files (30 days), 8760 for long life files (365 days), and that you have no size-based purge policy set for the ADR repository. In the following example, the ADR short life files purge policy is changed to 360 (15 days), the short life files size limit before a purge is set to 18 gigabytes (G), and the size purge threshold is set to 12G

```
set control (SHORTP_POLICY = 360 SIZEP_POLICY = 18G PURGE_THRESHOLD =12G)
```

23.9.13 SET ECHO

The ADRCI `SET ECHO` command turns command output on or off. This command only affects output being displayed in a script or using the spool mode.

Syntax and Description

```
SET ECHO ON | OFF
```

This command does not require an ADR home to be set before you can use it.

Example

```
SET ECHO OFF
```

Related Topics

- SPOOL
  The ADRCI `SET EDITOR` command directs ADRCI output to a file.

23.9.14 SET EDITOR

The ADRCI `SET EDITOR` command sets the editor for displaying the alert log and the contents of trace files.

Syntax and Description

```
SET EDITOR editor_program
```

editor_program is the editor program name. If no editor is set, then ADRCI uses the editor specified by the operating system environment variable `EDITOR`. If `EDITOR` is not set, then ADRCI uses `vi` as the default editor.

This command does not require an ADR home to be set before you can use it.

Example

```
SET EDITOR xemacs
```

23.9.15 SET HOMEPATH

The ADRCI `SET HOMEPATH` command makes one or more ADR homes current. Many ADR commands work with the current ADR homes only.

Syntax and Description

```
SET HOMEPATH homepath_str1 homepath_str2 ...
```

When diagnosing data, to work with data from other instances or components, you must ensure that all the ADR homes for all of these instances or components are current. The *homepath_strn* strings are the paths of the ADR homes relative to the current ADR base. The `diag` directory name can be omitted from the path. If the specified path contains multiple ADR homes, then all of the homes are added to the home path.

If a desired new ADR home is not within the current ADR base, then you can use `SET BASE` to set a new ADR base, and then use `SET HOMEPATH`.

This command does not require an ADR home to be set before you can use it.

Example

```
SET HOMEPATH diag/rdbms/orcldw/orcldw1 diag/rdbms/orcldw/orcldw2
```

The following command sets the same home path as the previous example:

```
SET HOMEPATH rdbms/orcldw/orcldw1 rdbms/orcldw/orcldw2
```

Related Topics

- Definitions for Oracle Database ADRC
  To understand how to diagnose Oracle Database problems, learn the definitions of terms that Oracle uses for the ADRCI, and the Oracle Database fault diagnosability infrastructure.

23.9.16 SET TERMOUT

The ADRCI `SET TERMOUT` command turns output to the terminal on or off.

Syntax and Description

```
SET TERMOUT ON | OFF
```

This setting is independent of spooling. That is, the output can be directed to both terminal and a file at the same time.

This command does not require an ADR home to be set before you can use it.

> **See Also:**
>
> `SPOOL` for more information about spooling

Example

```
SET TERMOUT ON
```

Related Topics

- SPOOL
  The ADRCI `SET EDITOR` command directs ADRCI output to a file.

23.9.17 SHOW ALERT

The ADRCI `SHOW ALERT` command shows the contents of the alert log in the default editor.

Purpose

Shows the contents of the alert log in the default editor.

Syntax and Description

```
show alert [-p "predicate_string"] [-tail [num] [-f]] [-term]
  [-file alert_file_name]
```

Except when using the `-term` flag, this command works with only a single current ADR home. If more than one ADR home is set, ADRCI prompts you to choose the ADR home to use.

Table 23-9 Flags for the SHOW ALERT command

Flag	Description
`-p "predicate_string"`	Uses a SQL-like predicate string to show only the alert log entries for which the predicate is true. The predicate string must be enclosed in double quotation marks. The table that follows this table lists the fields that can be used in the predicate string.
`-tail [num][-f]`	Displays the most recent entries in the alert log. Use the *num* option to display the last *num* entries in the alert log. If *num* is omitted, then the last 10 entries are displayed. If the `-f` option is given, after displaying the requested messages, the command does not return. Instead, it remains active and continuously displays new alert log entries to the terminal as they arrive in the alert log. You can use this command to perform live monitoring of the alert log. To terminate the command, press CTRL+C.
`-term`	Directs results to the terminal. Outputs the entire alert logs from all current ADR homes, one after another. If this option is not given, then the results are displayed in the default editor.
`-file alert_file_name`	Enables you to specify an alert file outside the ADR. *alert_file_name* must be specified with a full path name. Note that this option cannot be used with the *-tail* option.

Table 23-10 Alert Fields for SHOW ALERT

Field	Type
ORIGINATING_TIMESTAMP	timestamp
NORMALIZED_TIMESTAMP	timestamp
ORGANIZATION_ID	text(65)
COMPONENT_ID	text(65)
HOST_ID	text(65)

Table 23-10 (Cont.) Alert Fields for SHOW ALERT

Field	Type
HOST_ADDRESS	text(17)
MESSAGE_TYPE	number
MESSAGE_LEVEL	number
MESSAGE_ID	text(65)
MESSAGE_GROUP	text(65)
CLIENT_ID	text(65)
MODULE_ID	text(65)
PROCESS_ID	text(33)
THREAD_ID	text(65)
USER_ID	text(65)
INSTANCE_ID	text(65)
DETAILED_LOCATION	text(161)
UPSTREAM_COMP_ID	text(101)
DOWNSTREAM_COMP_ID	text(101)
EXECUTION_CONTEXT_ID	text(101)
EXECUTION_CONTEXT_SEQUENCE	number
ERROR_INSTANCE_ID	number
ERROR_INSTANCE_SEQUENCE	number
MESSAGE_TEXT	text(2049)
MESSAGE_ARGUMENTS	text(129)
SUPPLEMENTAL_ATTRIBUTES	text(129)
SUPPLEMENTAL_DETAILS	text(4000)
PROBLEM_KEY	text(65)

Examples

This example shows all alert messages for the current ADR home in the default editor:

```
show alert
```

This example shows all alert messages for the current ADR home and directs the output to the terminal instead of the default editor:

```
show alert -term
```

This example shows all alert messages for the current ADR home with message text describing an incident:

```
show alert -p "message_text like '%incident%'"
```

This example shows the last twenty alert messages, and then keeps the alert log open, displaying new alert log entries as they arrive:

```
show alert -tail 20 -f
```

This example shows all alert messages for a single ADR home in the default editor when multiple ADR homes have been set:

```
show alert

Choose the alert log from the following homes to view:

1: diag/tnslsnr/dbhost1/listener
2: diag/asm/+asm/+ASM
3: diag/rdbms/orcl/orcl
4: diag/clients/user_oracle/host_9999999999_11
Q: to quit

Please select option:
3
```

Related Topics

- SET EDITOR
  The ADRCI `SET EDITOR` command sets the editor for displaying the alert log and the contents of trace files.

23.9.18 SHOW BASE

The ADRCI `SET EDITOR` command shows the current ADR base.

Syntax and Description

```
SHOW BASE [-product product_name]
```

(Optional) You can show the product's ADR base location for a specific product. The products currently supported are `CLIENT` and `ADRCI`.

This command does not require an ADR home to be set before you can use it.

Example

This example shows the current ADR base:

```
SHOW BASE
```

Output:

```
ADR base is "/u01/app/oracle"
```

This example shows the current ADR base for Oracle Database clients:

```
SHOW BASE -product client
```

23.9.19 SHOW CONTROL

The ADRCI `SHOW CONTROL` command displays information about the Automatic Diagnostic Repository (ADR), including the purging policy.

Purpose

Displays metadata values for the ADR. The ADR maintains it's metadata in a repository as relations between controls in the repository. Use `SHOW CONTROL` to see what the current settings are for automatic time-based ADR purging.

Syntax and Description

```
SHOW CONTROL
```

Show control shows the including the following purging policy attributes:

Attribute Name	Description
SHORTP_POLICY	Number of hours after which to purge ADR contents that have a short life. Default: Starting with Oracle Database 23ai, 504 hours (21 days). A setting of 0 (zero) means that all contents that have a short life can be purged. The maximum setting is 35791394. If a value greater than 35791394 is specified, then this attribute is set to 0 (zero). The ADR contents that have a short life include the following: • Trace files, including those files stored in the cdmp_*timestamp* subdirectories • Core dump files • Packaging information
LONGP_POLICY	Number of hours after which to purge ADR contents that have a long life. Default: Starting with Oracle Database 23ai, 504 hours (21 days). A setting of 0 (zero) means that all contents that have a long life can be purged. The maximum setting is 35791394. If a value greater than 35791394 is specified, then this attribute is set to 0 (zero). The ADR contents that have a long life include the following: • Incident information • Incident dumps • Alert logs
SIZEP_POLICY	(Optional) Defines the size limit for an Automatic Workload Repository (AWR) home. In Oracle Database 12c Release 2 (12.2) and later releases, you can use SIZEP_POLICY to set a size limit for the AWR. When you set SIZEP_POLICY, the MMON background process collects statistics for the AWR home. By default, the ADR home is purged every 24 hours. If this purge time frame is inadequate, then you can set the SIZEP_POLICY to define a size limit for an ADR home to purge the ADR home when it approaches the purge size threshold. When you set a size limit using SIZEP_POLICY, MMON checks the current status of that limit every four hours. If the size limit is reached, then ADR purges the ADR repository.
PURGE_THRESHOLD	The PURGE_THRESHOLD value is a value at which the SIZEP_POLICY is triggered. If you set SIZEP_POLICY, then by default, the value of PURGE_THRESHOLD is 95 percent of the value of the SIZEP_POLICY. In a multitenant environment, the ADR home is shared, so the PURGE_THRESHOLD size policy is applied to the diagnostics storage location (diag). You can tune PURGE_THRESHOLD independently for each ADR home by setting the value for the PURGE_THRESHOLD column in the ADR_CONTROL_AUX relation . When you tune the PURGE_THRESHOLD value, this can assist you with keeping the amount of ADR data below the SIZEP_POLICY limit, even if your ADR home is purged infrequently.

> **Note:**
>
> The SHORTP_POLICY and LONGP_POLICY attributes are not mutually exclusive. Each policy controls different types of content.

Example

In the following example, SHOW CONTROL is used to show the purge policy settings for the ADR home in CDB1. Relevant values are highlighted in **Bold** font. The format of the SHOW CONTROL output is slightly altered in this example. Note the following

- The SHORTP_POLICY shows that the ADR automatically purges files that have a short life, such as trace files, after 30 days (720 hours). This is the default setting.
- The LONGP_POLICY shows that the ADR purges contents that have a long life, such as alert files, after 365 days (8760 hours). This is the default setting.
- The SIZEP_POLICY shows that the maximum size limit for the ADR home is set to 18 GB (19,327,352,832 bytes).
- The PURGE_THRESHOLD shows that the threshold is set to 95 percent of the SIZEP_POLICY (the default).

```
ADRID SHORTP_POLICY LONGP_POLICY LAST_MOD_TIME
    LAST_AUTOPRG_TIME LAST_MANUPRG_TIME ADRDIR_VERSION ADRSCHM_VERSION
ADRSCHMV_SUMMARY
    ADRALERT_VERSION CREATE_TIME SIZEP_POLICY PURGE_PERIOD FLAGS
PURGE_THRESHOLD
    . . .
    1481481004 720 8760 2020-03-31...2020-03-31... 1 2 110 1 2020-03-25...
    19327352832 0  0 95
    -07:00 1 rows fetched
```

Certain values in the SHOW CONTROL output are not relevant for managing the ADR, but can be relevant for Oracle Support. Note that you can also query individual results:

```
adrci> select SHORTP_POLICY,LONGP_POLICY,LAST_AUTOPRG_TIME,LAST_MANUPRG_TIME
from ADR_CONTROL;

ADR Home = /home/oracle/diag/rdbms/cdb1/cdb1:
********************************************************************************
SHORTP_POLICY        LONGP_POLICY
LAST_AUTOPRG_TIME
LAST_MANUPRG_TIME
-------------------- --------------------
---------------------------------------
---------------------------------------
720                  8760                2020-01-03 23:17:09.351760
+00:00
1 rows fetched
```

23.9.20 SHOW HM_RUN

The ADRCI `SHOW HM_RUN` command shows all information for Health Monitor runs.

Purpose

Shows all information for Health Monitor runs.

Syntax and Description

```
show hm_run [-p "predicate_string]
```

predicate_string is a SQL-like predicate that specifies the field names that you want to select. The following table displays the list of field names you can use:

Table 23-11 Fields for Health Monitor Runs

Field	Type
RUN_ID	number
RUN_NAME	text(31)
CHECK_NAME	text(31)
NAME_ID	number
MODE	number
START_TIME	timestamp
RESUME_TIME	timestamp
END_TIME	timestamp
MODIFIED_TIME	timestamp
TIMEOUT	number
FLAGS	number
STATUS	number
SRC_INCIDENT_ID	number
NUM_INCIDENTS	number
ERR_NUMBER	number
REPORT_FILE	bfile

Examples

This example displays data for all Health Monitor runs:

```
show hm_run
```

This example displays data for the Health Monitor run with ID 123:

```
show hm_run -p "run_id=123"
```

Related Topics

- About Health Monitor

23.9.21 SHOW HOMEPATH

The ADRCI `SHOW HOMEPATH` command is identical to the `SHOW HOMES` command.

Syntax and Description

```
SHOW HOMEPATH | SHOW HOMES | SHOW HOME
```

This command does not require an ADR home to be set before you can use it.

Example

```
SHOW HOMEPATH
```

Output:

```
ADR Homes:
diag/tnslsnr/dbhost1/listener
diag/asm/+asm/+ASM
diag/rdbms/orcl/orcl
diag/clients/user_oracle/host_9999999999_11
```

Related Topics

- SET HOMEPATH
  The ADRCI `SET HOMEPATH` command makes one or more ADR homes current. Many ADR commands work with the current ADR homes only.

23.9.22 SHOW HOMES

The ADRCI `SHOW HOMES` command shows the ADR homes in the current ADRCI session.

Syntax and Description

```
SHOW HOMES | SHOW HOME | SHOW HOMEPATH
```

This command does not require an ADR home to be set before you can use it.

Example

```
SHOW HOMES
```

Output:

```
ADR Homes:
diag/tnslsnr/dbhost1/listener
diag/asm/+asm/+ASM
diag/rdbms/orcl/orcl
diag/clients/user_oracle/host_9999999999_11
```

23.9.23 SHOW INCDIR

The ADRCI `SHOW INCDIR` command shows trace files for the specified incident.

Syntax and Description

```
show incdir [id | id_low id_high]
```

You can provide a single incident ID (*id*), or a range of incidents (*id_low* to *id_high*). If no incident ID is given, then trace files for all incidents are listed.

Examples

This example shows all trace files for all incidents:

```
show incdir
```

Output:

```
ADR Home = /u01/app/oracle/log/diag/rdbms/emdb/emdb:
*************************************************************************
diag/rdbms/emdb/emdb/incident/incdir_3801/emdb_ora_23604_i3801.trc
diag/rdbms/emdb/emdb/incident/incdir_3801/emdb_m000_23649_i3801_a.trc
diag/rdbms/emdb/emdb/incident/incdir_3802/emdb_ora_23604_i3802.trc
diag/rdbms/emdb/emdb/incident/incdir_3803/emdb_ora_23604_i3803.trc
diag/rdbms/emdb/emdb/incident/incdir_3804/emdb_ora_23604_i3804.trc
diag/rdbms/emdb/emdb/incident/incdir_3805/emdb_ora_23716_i3805.trc
diag/rdbms/emdb/emdb/incident/incdir_3805/emdb_m000_23767_i3805_a.trc
diag/rdbms/emdb/emdb/incident/incdir_3806/emdb_ora_23716_i3806.trc
diag/rdbms/emdb/emdb/incident/incdir_3633/emdb_pmon_28970_i3633.trc
diag/rdbms/emdb/emdb/incident/incdir_3633/emdb_m000_23778_i3633_a.trc
diag/rdbms/emdb/emdb/incident/incdir_3713/emdb_smon_28994_i3713.trc
diag/rdbms/emdb/emdb/incident/incdir_3713/emdb_m000_23797_i3713_a.trc
diag/rdbms/emdb/emdb/incident/incdir_3807/emdb_ora_23783_i3807.trc
diag/rdbms/emdb/emdb/incident/incdir_3807/emdb_m000_23803_i3807_a.trc
diag/rdbms/emdb/emdb/incident/incdir_3808/emdb_ora_23783_i3808.trc
```

This example shows all trace files for incident 3713:

```
show incdir 3713
```

Output:

```
ADR Home = /u01/app/oracle/log/diag/rdbms/emdb/emdb:
*************************************************************************
diag/rdbms/emdb/emdb/incident/incdir_3713/emdb_smon_28994_i3713.trc
diag/rdbms/emdb/emdb/incident/incdir_3713/emdb_m000_23797_i3713_a.trc
```

This example shows all tracefiles for incidents between 3801 and 3804:

```
show incdir 3801 3804
```

Output:

```
ADR Home = /u01/app/oracle/log/diag/rdbms/emdb/emdb:
*************************************************************************
diag/rdbms/emdb/emdb/incident/incdir_3801/emdb_ora_23604_i3801.trc
diag/rdbms/emdb/emdb/incident/incdir_3801/emdb_m000_23649_i3801_a.trc
diag/rdbms/emdb/emdb/incident/incdir_3802/emdb_ora_23604_i3802.trc
diag/rdbms/emdb/emdb/incident/incdir_3803/emdb_ora_23604_i3803.trc
diag/rdbms/emdb/emdb/incident/incdir_3804/emdb_ora_23604_i3804.trc
```

23.9.24 SHOW INCIDENT

The ADRCI `SHOW INCIDENT` command lists all of the incidents associated with the current ADR home. Includes both open and closed incidents.

Syntax and Description

```
show incident [-p "predicate_string"] [-mode {BASIC|BRIEF|DETAIL}] [-orderby field1,
field2, ... [ASC|DSC]]
```

Table 23-12 Flags for SHOW INCIDENT command

Flag	Description
`-p "predicate_string"`	Use a predicate string to show only the incidents for which the predicate is true. The predicate string must be enclosed in double quotation marks. Refer to the table "Incident Fields for SHOW INCIDENT" for a list of the fields that can be used in the predicate string.
`-mode {BASIC|BRIEF|DETAIL}`	Choose an output mode for incidents. `BASIC` is the default. • `BASIC` displays only basic incident information (the `INCIDENT_ID`, `PROBLEM_ID`, and `CREATE_TIME` fields). It does not display flood-controlled incidents. • `BRIEF` displays all information related to the incidents, as described in the table "Incident Fields for SHOW INCIDENT." It includes flood-controlled incidents. • `DETAIL` displays all information for the incidents (as with `BRIEF` mode) as well as information about incident dumps. It includes flood-controlled incidents.
`-orderby field1, field2, ... [ASC|DSC]`	Show results sorted by field in the given order, as well as in ascending (`ASC`) and descending order (`DSC`). By default, results are shown in ascending order.

Table 23-13 Incident Fields for SHOW INCIDENT

Field	Type	Description
`INCIDENT_ID`	`number`	ID of the incident
`PROBLEM_ID`	`number`	ID of the problem to which the incident belongs
`CREATE_TIME`	`timestamp`	Time when the incident was created
`CLOSE_TIME`	`timestamp`	Time when the incident was closed
`STATUS`	`number`	Status of this incident
`FLAGS`	`number`	Flags for internal use
`FLOOD_CONTROLLED`	`number` (decoded to a text status by ADRCI)	Encodes the flood control status for the incident
`ERROR_FACILITY`	`text(10)`	Error facility for the error that caused the incident

Table 23-13 (Cont.) Incident Fields for SHOW INCIDENT

Field	Type	Description
ERROR_NUMBER	number	Error number for the error that caused the incident
ERROR_ARG1	text(64)	First argument for the error that caused the incident Error arguments provide additional information about the error, such as the code location that issued the error.
ERROR_ARG2	text(64)	Second argument for the error that caused the incident
ERROR_ARG3	text(64)	Third argument for the error that caused the incident
ERROR_ARG4	text(64)	Fourth argument for the error that caused the incident
ERROR_ARG5	text(64)	Fifth argument for the error that caused the incident
ERROR_ARG6	text(64)	Sixth argument for the error that caused the incident
ERROR_ARG7	text(64)	Seventh argument for the error that caused the incident
ERROR_ARG8	text(64)	Eighth argument for the error that caused the incident
SIGNALLING_COMPONENT	text(64)	Component that signaled the error that caused the incident
SIGNALLING_SUBCOMPONENT	text(64)	Subcomponent that signaled the error that caused the incident
SUSPECT_COMPONENT	text(64)	Component that has been automatically identified as possibly causing the incident
SUSPECT_SUBCOMPONENT	text(64)	Subcomponent that has been automatically identified as possibly causing the incident
ECID	text(64)	Execution Context ID
IMPACT	number	Encodes the impact of the incident
ERROR_ARG9	text(64)	Ninth argument for the error that caused the incident
ERROR_ARG10	text(64)	Tenth argument for the error that caused the incident
ERROR_ARG11	text(64)	Eleventh argument for the error that caused the incident
ERROR_ARG12	text(64)	Twelfth argument for the error that caused the incident

Examples

This example shows all incidents for this ADR home:

```
show incident
```

Output:

```
ADR Home = /u01/app/oracle/log/diag/rdbms/emdb/emdb:
********************************************************************************
INCIDENT_ID          PROBLEM_KEY                                       CREATE_TIME
-------------------- ------------------------------------------------- ---------------------------
3808                 ORA 603                                           2010-06-18 21:35:49.322161 -07:00
3807                 ORA 600 [4137]                                    2010-06-18 21:35:47.862114 -07:00
3806                 ORA 603                                           2010-06-18 21:35:26.666485 -07:00
3805                 ORA 600 [4136]                                    2010-06-18 21:35:25.012579 -07:00
3804                 ORA 1578                                          2010-06-18 21:35:08.483156 -07:00
3713                 ORA 600 [4136]                                    2010-06-18 21:35:44.754442 -07:00
3633                 ORA 600 [4136]                                    2010-06-18 21:35:35.776151 -07:00
7 rows fetched
```

This example shows the detail view for incident 3805:

```
adrci> show incident -mode DETAIL -p "incident_id=3805"
```

Output:

```
ADR Home = /u01/app/oracle/log/diag/rdbms/emdb/emdb:
********************************************************************************

*************************************************************
INCIDENT INFO RECORD 1
*************************************************************
   INCIDENT_ID                   3805
   STATUS                        closed
   CREATE_TIME                   2010-06-18 21:35:25.012579 -07:00
   PROBLEM_ID                    2
   CLOSE_TIME                    2010-06-18 22:26:54.143537 -07:00
   FLOOD_CONTROLLED              none
   ERROR_FACILITY                ORA
   ERROR_NUMBER                  600
   ERROR_ARG1                    4136
   ERROR_ARG2                    2
   ERROR_ARG3                    18.0.628
   ERROR_ARG4                    <NULL>
   ERROR_ARG5                    <NULL>
   ERROR_ARG6                    <NULL>
   ERROR_ARG7                    <NULL>
   ERROR_ARG8                    <NULL>
   SIGNALLING_COMPONENT          <NULL>
   SIGNALLING_SUBCOMPONENT       <NULL>
   SUSPECT_COMPONENT             <NULL>
   SUSPECT_SUBCOMPONENT          <NULL>
   ECID                          <NULL>
   IMPACTS                       0
   PROBLEM_KEY                   ORA 600 [4136]
   FIRST_INCIDENT                3805
   FIRSTINC_TIME                 2010-06-18 21:35:25.012579 -07:00
   LAST_INCIDENT                 3713
   LASTINC_TIME                  2010-06-18 21:35:44.754442 -07:00
   IMPACT1                       0
   IMPACT2                       0
   IMPACT3                       0
   IMPACT4                       0
   KEY_NAME                      Client ProcId
   KEY_VALUE                     oracle@dbhost1 (TNS V1-V3).23716_3083142848
   KEY_NAME                      SID
   KEY_VALUE                     127.52237
   KEY_NAME                      ProcId
   KEY_VALUE                     23.90
   KEY_NAME                      PQ
```

```
   KEY_VALUE                      (0, 1182227717)
   OWNER_ID                       1
   INCIDENT_FILE                  /.../emdb/emdb/incident/incdir_3805/emdb_ora_23716_i3805.trc
   OWNER_ID                       1
   INCIDENT_FILE                  /.../emdb/emdb/trace/emdb_ora_23716.trc
   OWNER_ID                       1
   INCIDENT_FILE                  /.../emdb/emdb/incident/incdir_3805/emdb_m000_23767_i3805_a.trc
1 rows fetched
```

Related Topics

- SHOW INCIDENT
  The ADRCI `SHOW INCIDENT` command lists all of the incidents associated with the current ADR home. Includes both open and closed incidents.

23.9.25 SHOW LOG

The ADRCI `SHOW LOG` command shows diagnostic log messages.

Syntax and Description

```
SHOW LOG [-l log_name] [-p "predicate_string"] [-term] [ [-tail [num] [-f]] ]
```

The following table describes the flags for `SHOW LOG`.

Table 23-14 Flags for SHOW LOG command

Flag	Description
`-l log_name`	Name of the log to show. If no log name is specified, then this command displays all messages from all diagnostic logs under the current ADR Home.
`-p "predicate_string"`	Use a SQL-like predicate string to show only the log entries for which the predicate is true. The predicate string must be enclosed in double quotation marks. The table "Log Fields for SHOW LOG" lists the fields that can be used in the predicate string.
`-term`	Direct results to the terminal. If this option is not specified, then this command opens the results in an editor. By default, it opens the results in the `emacs` editor, but you can use the `SET EDITOR` command to open the results in other editors.
`-tail [num] [-f]`	Displays the most recent entries in the log. Use the *num* option to display the last *num* entries in the log. If *num* is omitted, then the last 10 entries are displayed. If the `-f` option is given, then after displaying the requested messages, the command does not return. Instead, it remains active, and continuously displays new log entries to the terminal as they arrive in the log. You can use this command to perform live monitoring of the log. To terminate the command, press `CTRL+C`.

Table 23-15 Log Fields for SHOW LOG

Field	Type
ORIGINATING_TIMESTAMP	timestamp

Table 23-15 (Cont.) Log Fields for SHOW LOG

Field	Type
NORMALIZED_TIMESTAMP	timestamp
ORGANIZATION_ID	text(65)
COMPONENT_ID	text(65)
HOST_ID	text(65)
HOST_ADDRESS	text(17)
MESSAGE_TYPE	number
MESSAGE_LEVEL	number
MESSAGE_ID	text(65)
MESSAGE_GROUP	text(65)
CLIENT_ID	text(65)
MODULE_ID	text(65)
PROCESS_ID	text(33)
THREAD_ID	text(65)
USER_ID	text(65)
INSTANCE_ID	text(65)
DETAILED_LOCATION	text(161)
UPSTREAM_COMP_ID	text(101)
DOWNSTREAM_COMP_ID	text(101)
EXECUTION_CONTEXT_ID	text(101)
EXECUTION_CONTEXT_SEQUENCE	number
ERROR_INSTANCE_ID	number
ERROR_INSTANCE_SEQUENCE	number
MESSAGE_TEXT	text(2049)
MESSAGE_ARGUMENTS	text(129)
SUPPLEMENTAL_ATTRIBUTES	text(129)
SUPPLEMENTAL_DETAILS	text(4000)
PROBLEM_KEY	text(65)

23.9.26 SHOW PROBLEM

The ADRCI `SHOW PROBLEM` command shows problem information for the current ADR home.

Syntax and Description

```
show problem [-p "predicate_string"] [-last num | -all]
    [-orderby field1, field2, ... [ASC|DSC]]
```

The following table describes the flags for `SHOW PROBLEM`.

Table 23-16 Flags for SHOW PROBLEM command

Flag	Description
`-p "predicate_string"`	Use a SQL-like predicate string to show only the incidents for which the predicate is true. The predicate string must be enclosed in double quotation marks. The table "Problem Fields for SHOW PROBLEM" lists the fields that can be used in the predicate string.
`-last num | -all`	Shows the last *num* problems, or lists all the problems. By default, `SHOW PROBLEM` lists the most recent 50 problems.
`-orderby field1, field2, ... [ASC|DSC]`	Show results sorted by field in the given order (*field1*, *field2*, ...), as well as in ascending (`ASC`) and descending order (`DSC`). By default, results are shown in ascending order.

Table 23-17 Problem Fields for SHOW PROBLEM

Field	Type	Description
PROBLEM_ID	number	ID of the problem
PROBLEM_KEY	text(550)	Problem key for the problem
FIRST_INCIDENT	number	Incident ID of the first incident for the problem
FIRSTINC_TIME	timestamp	Creation time of the first incident for the problem
LAST_INCIDENT	number	Incident ID of the last incident for the problem
LASTINC_TIME	timestamp	Creation time of the last incident for the problem
IMPACT1	number	Encodes an impact of this problem
IMPACT2	number	Encodes an impact of this problem
IMPACT3	number	Encodes an impact of this problem
IMPACT4	number	Encodes an impact of this problem
SERVICE_REQUEST	text(64)	Service request for the problem (entered through Support Workbench)
BUG_NUMBER	text(64)	Bug number for the problem (entered through Support Workbench)

Example

This example lists all the problems in the current ADR home:

```
show problem -all
```

This example shows the problem with ID 4:

```
show problem -p "problem_id=4"
```

23.9.27 SHOW REPORT

The ADRCI `SET EDITOR` command shows a report for the specified report type and run name.

Purpose

Currently, only the `hm_run` (Health Monitor) report type is supported, and only in XML formatting. To view HTML-formatted Health Monitor reports, use Oracle Enterprise Manager or the `DBMS_HM` PL/SQL package.

See *Oracle Database Administrator's Guide* for more information.

Syntax and Description

```
SHOW REPORT report_type run_name
```

report_type must be `hm_run`. *run_name* is the Health Monitor run name from which you created the report. You must first create the report using the `CREATE REPORT` command.

This command does not require an ADR home to be set before you can use it.

Example

```
SHOW REPORT hm_run hm_run_1421
```

Related Topics

- CREATE REPORT
  The ADRCI `CREATE REPORT` command creates a report for the specified report type and run ID, and stores the report in the ADR.
- SHOW HM_RUN
  The ADRCI `SHOW HM_RUN` command shows all information for Health Monitor runs.

23.9.28 SHOW TRACEFILE

The ADRCI `SHOW TRACEFILE` command lists trace files.

Syntax and Description

```
show tracefile [file1 file2 ...] [-rt | -t]
  [-i inc1 inc2 ...] [-path path1 path2 ...]
```

This command searches for one or more files under the trace directory, and all incident directories of the current ADR homes, unless the `-i` or `-path` flags are given.

This command does not require an ADR home to be set unless using the `-i` option.

The following table describes the arguments of `SHOW TRACEFILE`.

Table 23-18 Arguments for SHOW TRACEFILE Command

Argument	Description
file1 file2 ...	Filter results by file name. The `%` symbol is a wildcard character.

Table 23-19 Flags for SHOW TRACEFILE Command

Flag	Description
`-rt | -t`	Order the trace file names by timestamp. `-t` sorts the file names in ascending order by timestamp, and `-rt` sorts them in reverse order. Note that file names are only ordered relative to their directory. Listing multiple directories of trace files applies a separate ordering to each directory. Timestamps are listed next to each file name when using this option.
`-i` *inc1 inc2 ...*	Select only the trace files produced for the given incident IDs.
`-path` *path1 path2 ...*	Query only the trace files under the given path names.

Examples

This example shows all the trace files under the current ADR home:

```
show tracefile
```

This example shows all the `mmon` trace files, sorted by timestamp in reverse order:

```
show tracefile %mmon% -rt
```

This example shows all trace files for incidents 1 and 4, under the path `/home/steve/temp`:

```
show tracefile -i 1 4 -path /home/steve/temp
```

23.9.29 SPOOL

The ADRCI `SET EDITOR` command directs ADRCI output to a file.

Syntax and Description

```
SPOOL filename [[APPEND] | [OFF]]
```

filename is the file name where you want the output to be directed. If a full path name is not given, then the file is created in the current ADRCI working directory. If no file extension is given, then the default extension `.ado` is used. `APPEND` causes the output to be appended to the end of the file. Otherwise, the file is overwritten. Use `OFF` to turn off spooling.

This command does not require an ADR home to be set before you can use it.

Examples

```
SPOOL myfile

SPOOL myfile.ado APPEND

SPOOL OFF

SPOOL
```

23.10 Troubleshooting ADRCI

To assist troubleshooting, review some of the common ADRCI error messages, and their possible causes and remedies.

No ADR base is set
Cause: You may have started ADRCI with a null or invalid value for the `ORACLE_HOME` environment variable.

Action: Exit ADRCI, set the `ORACLE_HOME` environment variable, and restart ADRCI. For more information, see "ADR BASE" in Definitions for Oracle Database ADRC

DIA-48323: Specified pathname *string* must be inside current ADR home
Cause: A file outside of the ADR home is not allowed as an incident file for this command.

Action: Retry using an incident file inside the ADR home.

DIA-48400: ADRCI initialization failed
Cause: The ADR Base directory does not exist.

Action: Check the value of the `DIAGNOSTIC_DEST` initialization parameter, and ensure that it points to an ADR base directory that contains at least one ADR home. If `DIAGNOSTIC_DEST` is missing or null, check for a valid ADR base directory hierarchy in *ORACLE_HOME*/log.

DIA-48431: Must specify at least one ADR home path
Cause: The command requires at least one ADR home to be current.

Action: Use the `SET HOMEPATH` command to make one or more ADR homes current.

DIA-48432: The ADR home path *string* is not valid
Cause: The supplied ADR home is not valid, possibly because the path does not exist.

Action: Check if the supplied ADR home path exists.

DIA-48447: The input path [*path*] does not contain any ADR homes
Cause: When using `SET HOMEPATH` to set an ADR home, you must supply a path relative to the current ADR base.

Action: If the new desired ADR home is not within the current ADR base, first set ADR base with `SET BASE`, and then use `SHOW HOMES` to check the ADR homes under the new ADR base. Next, use `SET HOMEPATH` to set a new ADR home if necessary.

DIA-48448: This command does not support multiple ADR homes
Cause: There are multiple current ADR homes in the current ADRCI session.

Action: Use the `SET HOMEPATH` command to make a single ADR home current.