

Maîtriser l'interface ADR sous Oracle 11g

par Laurent Léturgez (<http://laurent-leturgez.developpez.com/>)
Laurent Léturgez ([Page Personnelle de Laurent](#))

Date de publication : 09/11/2009

Dernière mise à jour : 09/11/2009

I - Evolution des diagnostics.....	3
II - Architecture de l'ADR.....	3
III - L'interpréteur de commandes ADR : ADRCI.....	4
III-A - Présentation.....	4
III-B - Principales commandes :.....	5
III-B-1 - Configuration.....	5
III-B-1-a - SET CONTROL.....	5
III-B-1-b - SET EDITOR.....	6
III-B-1-c - SET ECHO.....	6
III-B-1-d - SET HOME(S).....	6
III-B-1-e - SET HOMEPATH.....	6
III-B-1-f - SET BASE.....	6
III-B-1-g - Configuration du listener dans ADR.....	6
III-B-2 - Analyse et monitoring.....	7
III-B-2-a - SHOW ALERT.....	7
III-B-2-b - SHOW TRACEFILE.....	8
III-B-2-c - SHOW TRACE.....	9
III-B-3 - Packaging d'incidents pour le support.....	9
III-B-3-a - IPS CREATE PACKAGE.....	10
III-B-3-b - IPS ADD FILE.....	11
III-B-3-c - IPS ADD INCIDENT.....	11
III-B-3-d - IPS SHOW FILES.....	11
III-B-3-e - IPS GENERATE.....	11
III-B-4 - Maintenance.....	11
IV - Conclusion.....	13
V - Liens.....	13

I - Evolution des diagnostics

Jusqu'en Oracle 10g, les diagnostics étaient principalement réalisés par le DBA à partir de fichiers de logs (alert.log, logs des background process), de trace utilisateurs, de core dumps etc.

Ces fichiers de logs étaient générés dans un répertoire pointé par les paramètres suivants :

```
other
- BACKGROUND_DUMP_DEST
- USER_DUMP_DEST
- CORE_DUMP_DEST
...
```

A cela, il fallait ajouter les fichiers de log et de trace du listener. Le cas échéant, les logs de l'instance ASM, du clusterWare CRS etc.

En plus de cela, les architectures étaient plus ou moins normalisées ... en fonction de nombreux paramètres (Architecte technique, DBA, client, plate forme, volonté des uns et des autres, etc.)

Évidemment, tout cela ne simplifiait pas la tâche du support Oracle lors de la résolution d'incidents en allongeant leurs durées, notamment lors des premiers échanges entre le client et le support.

Avec Oracle 11g, la nouvelle version intègre un changement majeur dans l'organisation des diagnostics. Désormais, les diagnostics sont centralisés par serveur, ils sont normalisés, et organisés de manière uniforme. Cette nouvelle organisation se présente donc sous le nom d'ADR : Automatic Diagnostic Repository dans laquelle nous retrouverons : les fichiers de logs, de trace, les alert.log, les rapports d'incidents etc.

De plus, il est à noter le changement d'organisation majeur des fichiers de log (alert.log, listener.log etc.). Ceux ci sont désormais au format XML qui leur permettra d'être lus plus facilement par des outils fourni dans ADRCI (voir plus bas).

II - Architecture de l'ADR

ADR est basé sur une architecture de répertoires normalisés. Comme pour la norme OFA, il dispose d'une base : ADR_BASE. Cette base sera le point d'entrée de tous les diagnostics émis par les produits d'un serveur. On y retrouvera les diagnostics des instances de bases de données, de l'instance ASM, du CRS (avant 11gR2), du listener (selon la configuration du listener) etc.

Au sein de cette Base ADR, on va retrouver des "ADR Homes" qui vont contenir les diagnostics d'un composant spécifique hébergé sur le serveur : une instance de base de données, une instance ASM ou le listener par exemple.

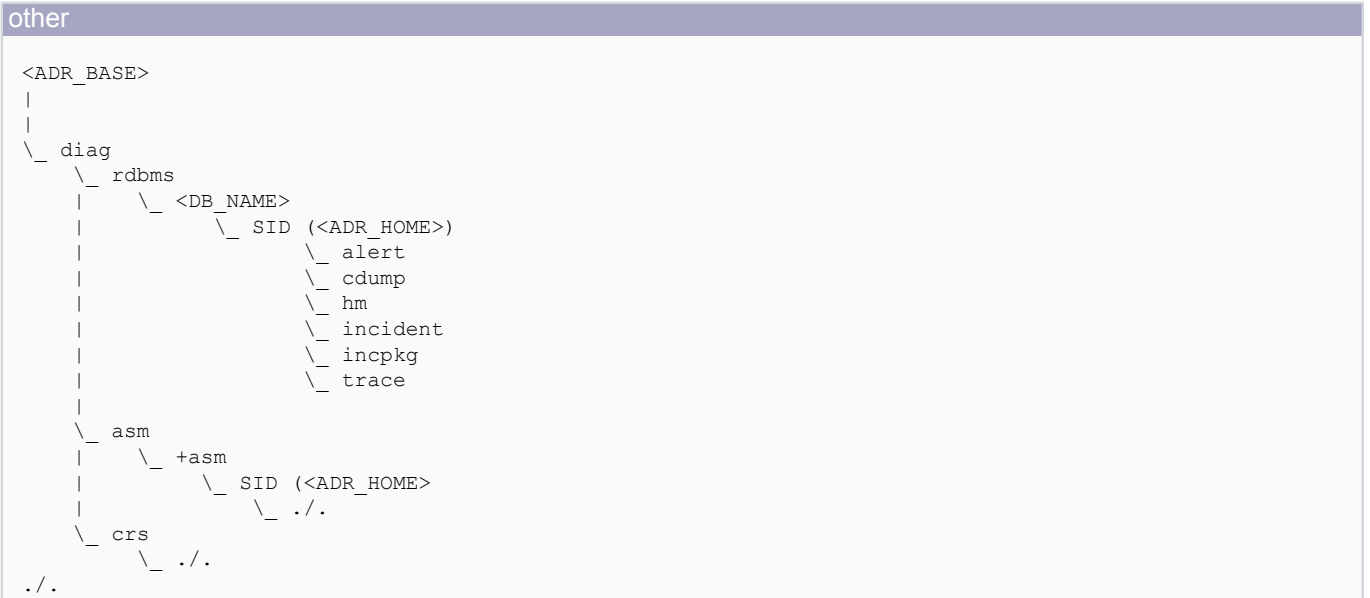
Pour une base de données (ou d'une instance ASM), on peut obtenir les informations de diagnostics avec la requête suivante :

```
other

SQL> select * from gv$diag_info;
INST_ID NAME                                VALUE
-----
1 Diag Enabled                               TRUE
1 ADR Base                                   /u01/app/oracle-product/11.1.0/db_1
1 ADR Home                                   /u01/app/oracle-product/11.1.0/db_1/diag/asm/asm/+ASM
1 Diag Trace                                 /u01/app/oracle-product/11.1.0/db_1/diag/asm/asm/+ASM/trace
1 Diag Alert                                 /u01/app/oracle-product/11.1.0/db_1/diag/asm/asm/+ASM/alert
1 Diag Incident                              /u01/app/oracle-product/11.1.0/db_1/diag/asm/asm/+ASM/
incident
1 Diag Cdump                                 /u01/app/oracle-product/11.1.0/db_1/diag/asm/asm/+ASM/cdump
```

other		
1 Health Monitor		/u01/app/oracle-product/11.1.0/db_1/diag/asm/+asm/+ASM/hm
1 Default Trace File		/u01/app/oracle-product/11.1.0/db_1/diag/asm/+asm/+ASM/trace/
+ASM_ora_31579.trc		
1 Active Problem Count	0	
1 Active Incident Count	0	

En schématisant, l'architecture ADR peut-être représentée comme suit :



Pour configurer une instance, il existe un nouveau paramètre "DIAGNOSTIC_DEST" qui paramètre la base ADR. A partir de cela, comme c'est normalisé, oracle saura se débrouiller pour déposer ses logs et traces

III - L'interpréteur de commandes ADR : ADRCI

III-A - Présentation

ADR a été implémenté dans le but de simplifier et de normaliser la gestion des diagnostics. En 10g, nombre de DBA avait leur "méthode" pour gérer ces fichiers de diagnostics, leur purge, leur rotation, et la création de "package" pour le support, étaient dépendantes de celui qui générerait ces packages.

C'est pour ces raisons qu'oracle a implémenté une interface permettant de réaliser tout cela, de manière normalisée.

Cette interface se nomme ADRCI (ADR Command Interface). Elle se pilote via un binaire situé dans \$ORACLE_HOME/bin/adrci.

Cette interface peut être utilisée, comme Isnrctl, de manière interactive ou en mode batch :

Exemple d'utilisation d'ADRCI en mode interactif :

```

other
$ adrci
ADRCI: Release 11.1.0.6.0 - Beta on Mon Jan 12 16:39:38 2009

Copyright (c) 1982, 2007, Oracle. All rights reserved.

ADR base = "/u01/app/oracle"
adrci> show homes
  
```

other

```
ADR Homes:
diag/asm/+asm/+ASM
diag/rdbms/dbtest/dbtest
adrci> exit
```

Exemple d'utilisation d'ADRCI en mode batch :

other

```
$ adrci exec="SHOW HOMES"
ADR Homes:
diag/asm/+asm/+ASM
diag/rdbms/dbtest/dbtest
```

III-B - Principales commandes :

Dans cette partie, nous allons voir les principales commandes de l'interface ADRCI. Par souci de lisibilité, elles ne seront pas toutes expliquées. Pour plus d'informations sur ces commandes, celles ci sont déclinées dans le "book" Utilities de la documentation Oracle.

Avant de commencer, il existe une commande HELP qui permet d'obtenir de l'aide sur chacune des commandes. Les commandes suivantes donnent la liste des commandes disponibles :

- HELP : donne la liste des commandes standards
- HELP EXTENDED : donne la liste des commandes étendues
- HELP IPS : donne la liste des commandes de génération des packages pour le support.

III-B-1 - Configuration

Les commandes permettant de configurer ADR sont généralement des commandes SET complétées d'un paramètre.

III-B-1-a - SET CONTROL

La commande SET CONTROL permet de définir les politiques standard de purge. Il existe deux politiques :

- une politique courte durée dénommée SHORTP_POLICY. Cette politique est appliquée aux fichiers et aux dumps des incidents.
- une politique longue durée dénommée LONGP_POLICY. Cette politique est appliquée aux métadonnées des incidents.

La commande SET CONTROL permet de définir le nombre d'heures de conservation associé à ces politiques.

Exemple :

other

```
adrci> set control (SHORTP_POLICY=1);
adrci> set control (SHORTP_POLICY=2, LONGP_POLICY=100);
```

Le contrôle de la prise en compte de ces paramètres se fait avec la commande SHOW CONTROL. Cette commande, en plus des politiques donne différentes informations de configuration d'ADR.

NB : On doit spécifier une home ADR pour lancer cette commande

III-B-1-b - SET EDITOR

Cette commande définit l'éditeur utilisé :

```
other
```

```
adrci> set editor vi
```

III-B-1-c - SET ECHO

Cette commande définit l'affichage ou non des sorties de commandes lors de spool ou d'exécution de script.

```
other
```

```
adrci> set echo off
```

III-B-1-d - SET HOME(S)

Cette commande permet de sélectionner une ou plusieurs home ADR de travail.

```
other
```

```
adrci> SET HOME diag/rdbms/dbtest/dbtest  
adrci> SET HOMES diag/asm/+asm/+ASM, diag/rdbms/dbtest/dbtest
```

Les commandes SHOW HOME et SHOW HOMES permettent de visualiser la ou les home(s) ADR de travail actuelle(s).

III-B-1-e - SET HOMEPATH

Cette commande va définir un chemin comme une home ADR :

```
other
```

```
adrci> SET HOMEPATH diag/rdbms/dbtest/dbtest
```

III-B-1-f - SET BASE

Permet de configurer une base qui ne serait pas celle par défaut. Cette commande permet de scanner les homes disponibles dans cette base.

```
other
```

```
adrci> SET BASE /u01/app/oracle
```

III-B-1-g - Configuration du listener dans ADR

Par défaut, un listener en logging ou en trace va générer ses fichiers dans DIAG.

Le logging dans DIAG est conditionné par le paramètre du listener DIAG_ADR_ENABLED_. Ce paramètre est, par défaut, positionné à ON. Si on veut désactiver cela, et revenir à une journalisation "non-ADR", il faut le passer à OFF.

Pour ce qui concerne la mise en trace du listener, les paramètres restent les mêmes (TRACE_LEVEL_, TRACE_DIRECTORY_, TRACE_FILE_ etc.).

Cependant, si DIAG_ADR_ENABLED_ est positionné à ON, et qu'un niveau de trace est défini (USER, ADMIN, SUPPORT), alors le fichier de trace sera positionné dans DIAG.

A partir de la version 11gR2, on peut modifier la BASE ADR pour le listener.

Pour cela, il faut positionner le paramètre ADR_BASE_ dans le fichier listener.ora pour modifier cela.

Exemple pour un listener nommé LISTENER1522 :

```
other
```

```
ADR_BASE_LISTENER1522 = /u01/app/oracle/grid/
```

III-B-2 - Analyse et monitoring

III-B-2-a - SHOW ALERT

Une des commandes des plus importantes dans adrci est la commande SHOW ALERT.

Cette commande va permettre d'exploiter le fichier alert.log version 11g au format XML (elle n'exploite pas l'ancien format qui subsiste pour compatibilité ascendante si le paramètre BACKGROUND_DUMP_DEST est renseigné). Cette commande va également permettre l'exploitation des autres fichiers de trace, par exemple, le fichier de log du listener (qui était bien complexe à exploiter auparavant)

Le prototypage de la commande est le suivant :

```
other
```

```
SHOW ALERT [-p <predicate_string>] [-term] [ [-tail [num] [-f]] | [-file <alert_file_name>] ]
```

- L'option -p permet, via un prédicat de recherche, filtrer le contenu de sortie
- L'option -term, permet d'envoyer la sortie vers le terminal. Si cette option n'est pas spécifiée, l'ouverture se fera par l'éditeur mentionné via la commande SET EDITOR
- L'option -tail, comme sous unix, permet d'obtenir les derniers enregistrements de ce fichier. Avec le complément d'option -f, le rafraîchissement se fait à la volée. Pour terminer cet affichage à la volée, il faut réaliser un "Ctrl+C".
- L'option -file, permet de spécifier un fichier d'alert (au format brut et non XML). Cette option est mutuellement exclusive de l'option tail

NB : La normalisation de cette commande au sein d'ADRCI permettra l'implémentation des commandes unix "grep" (option -p) et "tail" (option -tail) sous Windows.

Exemple 1 : obtenir les cinq derniers événements du fichier d'alerte de l'instance ASM :

```
other
```

```
adrci> show homes
ADR Homes:
diag/asm/+asm/+ASM
diag/tnslsnr/linux1/listener
diag/rdbms/base/sid

adrci> set homepath diag/asm/+asm/+ASM
```

other

```
adrci> show home
ADR Homes:
diag/asm/+asm/+ASM

adrci> show alert -tail 5
```

Exemple 2 : Obtenir les deux derniers événements de la log du listener

other

```
adrci> show homes
ADR Homes:
diag/asm/+asm/+ASM
diag/tnslnsr/linux1/listener
diag/rdbms/base/sid

adrci> set homepath diag/tnslnsr/linux1/listener
adrci> show alert -tail 2
```

Exemple 3 : Afficher les dernières erreurs ORA-19815 (Flash Recovery area trop petite)

other

```
ADR base = "/home/oracle"
adrci> show homes
ADR Homes:
diag/rdbms/dbtest/dbtest

adrci> show alert -p "MESSAGE_TEXT LIKE '%ORA-19815%' " -term

ADR home = /home/oracle/diag/rdbms/dbtest/dbtest:
*****
2009-01-14 15:12:28.549000 +01:00
Errors in file /home/oracle/diag/rdbms/dbtest/dbtest/trace/dbtest_mmon_5486.trc:
ORA-19815: WARNING: db_recovery_file_dest_size of 104857600 bytes is 100.00% used, and has 0 remaining
bytes available.
2009-01-14 15:13:35.383000 +01:00
Errors in file /home/oracle/diag/rdbms/dbtest/dbtest/trace/dbtest_arc2_5782.trc:
ORA-19815: WARNING: db_recovery_file_dest_size of 104857600 bytes is 100.00% used, and has 0 remaining
bytes available.
Errors in file /home/oracle/diag/rdbms/dbtest/dbtest/trace/dbtest_arc3_5784.trc:
ORA-19815: WARNING: db_recovery_file_dest_size of 10485
7600 bytes is 100.00% used, and has 0 remaining bytes available.2009-01-14 15:15:10.870000 +01:00
Errors in file /home/oracle/diag/rdbms/dbtest/dbtest/trace/dbtest_ora_15762.trc:
ORA-19815: WARNING: db_recovery_file_dest_size of 524288000 bytes is 100.00% used, and has 0 remaining
bytes available.
2009-01-14 15:15:17.978000 +01:00
Errors in file /home/oracle/diag/rdbms/dbtest/dbtest/trace/dbtest_mmon_5486.trc:
ORA-19815: WARNING: db_recovery_file_dest_size of 2147483648 bytes is 88.93% used, and has 237809152
remaining bytes available.
```

III-B-2-b - SHOW TRACEFILE

Cette commande permet de visualiser les fichiers de trace générés pour une home ou plusieurs home ADR :

other

```
ADR Homes:
diag/asm/+asm/+ASM
diag/tnslnsr/linux1/listener
diag/rdbms/dbtest/dbtest

adrci> show tracefile
diag/asm/+asm/+ASM/trace/+ASM_ora_1796.trc
```


other

```
diag/asm/+asm/+ASM/trace/+ASM_gmon_1833.trc
diag/asm/+asm/+ASM/trace/+ASM_rbal_1831.trc
diag/asm/+asm/+ASM/trace/alert_+ASM.log
diag/asm/+asm/+ASM/trace/+ASM_vktm_1803.trc
diag/tnslsnr/linux1/listener/trace/listener.log
diag/tnslsnr/linux1/listener/trace/ora_2419_47162629456496.trc
diag/rdbms/dbtest/dbtest/trace/dbtest_ora_21674.trc
diag/rdbms/dbtest/dbtest/trace/dbtest_arc2_5782.trc
diag/rdbms/dbtest/dbtest/trace/dbtest_ora_13896.trc
diag/rdbms/dbtest/dbtest/trace/dbtest_m000_15784.trc
```

La commande `show tracefile` permet également de filtrer les traces associés à un processus d'arrière plan, par exemple :

other

```
adrci> show homes
ADR Homes:
diag/asm/+asm/+ASM
diag/tnslsnr/linux1/listener
diag/rdbms/dbtest/dbtest

adrci> show tracefile %arc0%
diag/rdbms/dbtest/dbtest/trace/dbtest_arc0_5778.trc
```

La visualisation de ces fichiers peut se faire avec un éditeur de texte de manière externe. ADRCI dispose également de la commande `HOST` permettant de débrayer sur le shell système. On peut également utiliser la commande `SHOW TRACE`.

III-B-2-c - SHOW TRACE

Cette commande permet d'afficher un fichier de trace. Soit l'affichage se fait par défaut dans l'éditeur configuré dans `adrci`, soit dans le terminal avec l'option `-term`. Il existe également un certain nombre d'options permettant de filtrer certaines données (plus d'infos dans la commande `HELP SHOW TRACE`)

other

```
adrci> show tracefile %pmon%
diag/rdbms/dbtest/dbtest/trace/dbtest_pmon_5458.trc
adrci> show trace dbtest_pmon_5458.trc
adrci> show trace dbtest_pmon_5458.trc -term
```

III-B-3 - Packaging d'incidents pour le support

ADR permet maintenant de détecter problèmes et incidents, et de réaliser des packages pour le support. Ces packages vont contenir, dans une archive ZIP, l'ensemble des fichiers nécessaires au support pour analyser le problème.

Pour commencer, nous allons simuler une erreur interne Oracle avec le code PL/SQL suivant :

other

```
declare
ora600 exception;
pragma exception_init(ora600,-600);
begin
raise ora600;
end;
/
```

La visualisation de problème se fait avec la commande SHOW PROBLEM.

La visualisation d'incident se fait avec la commande SHOW INCIDENT.

La relation qui existe entre problème et incident est qu'un PROBLEM peut recenser plusieurs INCIDENTS.

Une fois les incidents identifiés, on utilise IPS (Incident Packaging Service). IPS est intégré à ADRCI.

Pour créer un package, cela se fait en plusieurs étapes :

- créer l'enveloppe du package IPS
- ajouter des fichiers, des incidents à ce package
- réaliser le "packaging" pour envoyer l'archive au support Oracle

III-B-3-a - IPS CREATE PACKAGE

Cette commande sert à créer un package IPS. Elle nécessite d'avoir sélectionné la HOME ADR sur laquelle l'incident est référencé.

```
other
adrci> show homes
ADR Homes:
diag/asm/+asm/+ASM
diag/tnslsnr/linux1/listener
diag/rdbms/dbtest/dbtest
adrci> set homepath diag/rdbms/dbtest/dbtest
adrci> show incident

ADR Home = /HOME/oracle/diag/rdbms/dbtest/dbtest:
*****
INCIDENT_ID      PROBLEM_KEY      CREATE_TIME
-----
21789            ORA 7445         2009-01-15 14:51:14.061784 +01:00
21788            ORA 600          2009-01-15 11:55:37.883993 +01:00
21787            ORA 700 [kgerev1] 2009-01-15 11:55:36.292663 +01:00
3 rows fetched
```

A ce stade, on peut créer un package vide. Il faut veiller à bien noter l'identifiant de ce package qui nous servira durant le processus.

```
other
adrci> ips create package;
Created package 3 without any contents, correlation level typical
```

- On peut également créer un package à partir d'un problème. Tous les incidents référencés dans ce problème seront inclus dans ce package :

```
other
adrci> ips create package problem 2;
Created package 6 based on problem id 2, correlation level typical
```

- On peut également créer un package contenant déjà un incident :

```
other
adrci> ips create package incident 21788
Created package 4 based on incident id 21788, correlation level typical
```

III-B-3-b - IPS ADD FILE

On peut ensuite ajouter d'autres fichiers à ce package :

other

```
adrci> show tracefile %pmon%
diag/rdbms/dbtest/dbtest/trace/dbtest_pmon_5458.trc
adrci> ips add file /home/oracle/diag/rdbms/dbtest/dbtest/trace/dbtest_pmon_5458.trc package 4
Added file /home/oracle/diag/rdbms/dbtest/dbtest/trace/dbtest_pmon_5458.trc to package 4
```

III-B-3-c - IPS ADD INCIDENT

On peut également ajouter un incident au package avec la commande :

other

```
adrci> ips add incident 21787 package 4
Added incident 21787 to package 4
```

III-B-3-d - IPS SHOW FILES

Le contenu du package en cours peut être visualisé avec la commande IPS SHOW FILES :

other

```
adrci> ips show files package 4
```

III-B-3-e - IPS GENERATE

Une fois le package définit, on peut générer l'archive qui sera transmise au support :

other

```
adrci> ips generate package 4 in /var/tmp
Generated package 4 in file /var/tmp/ORA600_20090116112427_COM_1.zip, mode complete
```

Le fichier est prêt à être envoyé au support.

III-B-4 - Maintenance

La maintenance des fichiers de logs et de trace est une nouveauté car, auparavant, il fallait programmer des méthodes de purge de manière externe (sous OEM, scripts adhoc en shell, perl ou autre).

Désormais, avec ADR et ADRCI, la commande PURGE va permettre de maintenir les fichiers en fonction des politiques de rétention, ou de paramètres qui lui seront passés.

L'aide de la commande PURGE peut-être obtenu par la commande suivante :

other

```
adrci> HELP PURGE
```

La commande PURGE s'exécute sur une seule Home ADR. De ce fait, il y a nécessité de sélectionner cette home avec les commandes *SET HOME*, *SET HOMEPATH*

Si on utilise la commande purge sans paramètres. Les fichiers seront purgés en fonction des politiques de rétention paramétrées dans ADR.

Pour rappel, il existe deux politiques de rétention :

- une politique courte durée dénommée SHORTP_POLICY. Cette politique est appliquée aux fichiers et aux dumps des incidents.
- une politique longue durée dénommée LONGP_POLICY. Cette politique est appliquée aux métadonnées des incidents.

On obtient leurs valeurs avec la commande "SHOW CONTROL" :

```

other
-----
adrci> SHOW CONTROL
ADRID          SHORTP_POLICY    LONGP_POLICY
-----
3954336691     360              720
.../...
adrci> PURGE
  
```

On peut également spécifier des clauses plus précises pour la purge :

- Purge spécifique d'incidents La purge spécifique d'incidents s'effectue avec l'option -i suivie de la liste des identifiants d'incidents à purger

```

other
-----
adrci> set home diag/rdbms/dbtest/dbtest
adrci> show incident

ADR Home = /home/oracle/diag/rdbms/dbtest/dbtest:
*****
INCIDENT_ID    PROBLEM_KEY      CREATE_TIME
-----
22983          ORA 600          2009-01-20 10:34:35.271448 +01:00
22982          ORA 700 [kgerev1] 2009-01-20 10:34:32.527232 +01:00
2 rows fetched
adrci> purge -i 22982
adrci> show incident

ADR Home = /home/oracle/diag/rdbms/dbtest/dbtest:
*****
INCIDENT_ID    PROBLEM_KEY      CREATE_TIME
-----
22983          ORA 600          2009-01-20 10:34:35.271448 +01:00
1 rows fetched
  
```

- Purge ciblée sur le type et sur l'age;

Cette purge permet de purger une cible particulière (fichier d'alerte (ALERT), fichiers de trace (TRACE), incidents (INCIDENT), core dump (CDUMP), rapport "Health Monitor" (HM))

En complément du type, on purge en fonction d'un age donné en minutes.

Exemple 1 : Purge du fichier d'alerte du listener des événements de plus de 10 jours (14400 minutes) :

```

other
-----
adrci> show homes
ADR Homes:
diag/asm/+asm/+ASM
  
```

other

```
diag/tnslnsr/linux1/listener
diag/rdbms/dbtest/dbtest
adrci>
adrci> set homepath diag/tnslnsr/linux1/listener
adrci> purge -age 14400 -type alert
```

Exemple 2 : Purge des fichiers de trace de la base "dbtest" de plus de 10 heures (600 minutes)

other

```
adrci> show homes
ADR Homes:
diag/asm/+asm/+ASM
diag/tnslnsr/linux1/listener
diag/rdbms/dbtest/dbtest
adrci>
adrci> set homepath diag/rdbms/dbtest/dbtest
adrci> purge -age 600 -type trace
```

NB : noter l'intérêt de pouvoir passer la commande en commande externe pour intégration aux jobs de purge

IV - Conclusion

ADR et ADRCI sont les deux nouveaux outils permettant le diagnostic d'incident.

Ils représentent un changement majeur dans l'organisation des fichiers journaux et de trace de la base de données.

Cependant, ces changements permettent de réelles avancées (stockage XML, packaging d'incidents etc.), un diagnostic et une résolution d'incidents plus rapide qu'auparavant, une gestion plus industrielle des fichiers.

V - Liens

Documentation de l'interpréteur adrci : http://download.oracle.com/docs/cd/E11882_01/server.112/e10701/adrci.htm#BABBHGFC