

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Oracle. Narzędzia. Leksykon kieszonkowy

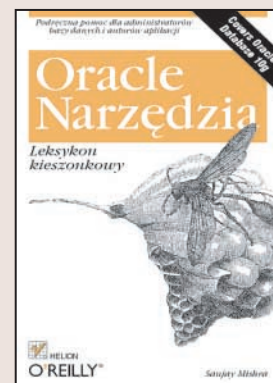
Autor: Sanjay Mishra

Tłumaczenie: Zbigniew Banach

ISBN: 83-7361-826-0

Tytuł oryginału: [Oracle Utilities Pocket Reference](#)

Format: B5, stron: 144



Podręczna pomoc dla administratorów bazy danych i autorów aplikacji

Instancja Oracle'a to nie tylko tabele, ale także wiele procesów działających w tle i sterujących bazą. Dostarczane wraz z Oraclem narzędzia administracyjne pozwalają na monitorowanie tych procesów, kontrolowanie ich pracy i zarządzanie tabelami znajdującymi się w bazie. Każde z nich jest niezbędne administratorowi aplikacji lub serwera bazy danych. Wszystkie narzędzia działają w trybie tekstowym, więc konieczna jest znajomość składni i parametrów poleceń.

„Oracle. Narzędzia. Leksykon kieszonkowy” to podręczne zestawienie opisów wszystkich narzędzi dołączanych do bazy danych Oracle. W książce opisuje się składnię i zastosowanie programów działających w linii poleceń, dostarczanych z najnowszą wersją bazy – Oracle 10g. Niektóre z przedstawianych narzędzi są dobrze znane i udokumentowane, tymczasem o innych praktycznie mało kto słyszał. Część z nich jest rozwijana od pierwszych wersji Oracle'a, a niektóre narzędzia zostały dołączone do jego najnowszej wersji. Wszystkie jednak mogą okazać się przydatne, dlatego warto mieć tę książkę w swojej bibliotece.



Spis treści

Wstęp	7
Narzędzia	9
csscan	12
dbfszize	19
dbhome	20
dbshut	21
dbstart	22
dbv	24
dropjava	29
emctl	31
exp	33
expdp	42
imp	50
impdp	55
isqlplusctl	60
loadjava	61
lsnrctl	68
maxmem	72
nid	73
oerr	77
oradebug	78
orakill	84

orapwd	85
relink	87
sqlldr	88
sqlplus	122
tkprof	125
tnsping	130
trcroute	131
tstshm	132
wrap	133
Skorowidz.....	137

csscan

Narzędzie *csscan* (Character Set Scanner) pozwala ustalić wpływ, jaki na bazę danych miałyby przejście na inny zestaw znaków. Narzędzie to skanuje dane w bazie i generuje raport przedstawiający ewentualne problemy związane ze zmianą zestawu znaków.

Składnia

```
csscan [konto_użytkownika] [wartparam [wartparam...]]
```

```
konto_użytkownika ::=  
nazwa_użytkownika[/hasło][@usługa_sieciowa]
```

```
wartparam ::= {parametr=pojedyncza_wartość  
                |parametr=(wartość1[,wartość2...])}
```

```
parametr ::= {  
userid=nazwa_użytkownika/hasło[@usługa_sieciowa]  
|full={y|n}  
|user=właściciel_tabel  
|table=tabela1[,tabela2[,tabela3...]]  
|exclude=tabela1[,tabela2[,tabela3...]]  
|tochar=nowy_zestaw_znaków  
|fromchar=obecny_zestaw_znaków  
|tonchar=nowy_zestaw_znaków_narodowych  
|fromnchar=obecny_zestaw_znaków_narodowych  
|array=[rozmiar_w_bajtach]  
|process=p  
|maxblocks=maks_ilość_bloków  
|capture={y|n}
```

```
|supress=s
|feedback=f
|boundaries=granica1[,granica2[,granica3...]]
|lastrpt={y|n}
|log=plik_raportu
|parfile=plik_parametrów
|preserve={y|n}
|lcsd={y|n}
|lcsdata={lossy|truncation|convertible|changeless|all}
|help={y|n} }
```

Parametry

Wykonanie skanowania zestawu znaków wymaga określenia, czy skanowana ma być cała baza (`full=y`), tabele należące do konkretnego użytkownika (`user=właściciel_tabel`) czy też wskazane tabele (`table=nazwa_tabeli...`).

Wskazówka

Przed przeniesieniem bazy danych na nowy zestaw znaków wskazane jest wykonanie pełnego skanowania.

Oto opisy poszczególnych parametrów:

`userid=nazwa_użytkownika/hasło[@usługa_sieciowa]`

Wskazuje nazwę i hasło użytkownika wykonującego skanowanie. Opcjonalnie można też podać nazwę usługi sieciowej.

`full={y|n}`

Określa, czy skanowana ma być cała baza (łącznie ze słownikiem danych).

`user=właściciel_tabel`

Wskazuje użytkownika, którego tabele będą skanowane.

`table=tabela1[,tabela2[,tabela3...]]`

Lista tabel do skanowania. Opcjonalnie można podać właściciela każdej z tabel.

`exclude=tabela1[,tabela2[,tabela3...]]`

Lista tabel, które nie będą objęte skanowaniem. Opcjonalnie można podać właściciela każdej z tabel.

Wskazówka

Parametry `table` i `exclude` wykluczają się nawzajem.

`tochar=nowy_zestaw_znaków`

Nazwa docelowego zestawu znaków dla danych typu CHAR, VARCHAR2, LONG i CLOB.

`fromchar=obecny_zestaw_znaków`

Nazwa bieżącego zestawu znaków dla danych typu CHAR, VARCHAR2, LONG i CLOB. Wartością domyślną jest zestaw znaków ustawiony dla bazy danych.

`tonchar=nowy_zestaw_znaków_narodowych`

Nazwa docelowego zestawu znaków dla danych typu NCHAR, NVARCHAR2 i NCLOB.

`fromnchar=obecny_zestaw_znaków_narodowych`

Nazwa bieżącego zestawu znaków dla danych typu NCHAR, NVARCHAR2 i NCLOB. Wartością domyślną jest zestaw znaków narodowych ustawiony dla bazy danych.

`array=rozmiar_w_bajtach`

Rozmiar bufora pamięciowego używanego do przechowywania pobieranych wierszy. Rozmiar bufora ogranicza ilość wierszy tabeli, jakie może pobrać *csscan*. Wartością domyślną jest 102 400 bajtów.

`process=p`

Ilość współbieżnych procesów skanujących (od 1 do 32). Wartością domyślną jest 1.

`maxblocks=maks_ilość_bloków`

Maksymalna ilość bloków tabeli. Większe tabele będą dzielone na części o rozmiarze *maks_ilość_bloków* i skanowane

jednocześnie przez kilka procesów skanujących (pod warunkiem odpowiedniego ustawienia wartości parametru `process`).

`capture={y|n}`

Określa, czy przechwytywane mają być informacje o poszczególnych wierszach. Ustawienie `capture=y` powoduje zapisywanie informacji o problematycznych wierszach w tabeli `CSM$ERRORS`, co pozwala ustalić, które wiersze trzeba będzie przenosić przy wykorzystaniu operacji eksportu i importu.

Wskazówka

Skanowanie z opcją `capture=y` wymaga dodatkowego czasu i przestrzeni dyskowej.

`supress=s`

Określa maksymalną ilość wyjątków, jakie mają być zapisane dla każdej tabeli. Po przekroczeniu podanej wartości kolejne wyjątki są ignorowane.

`feedback=f`

Powoduje wyświetlenie wskaźnika postępu składającego się z kropek. Każda kropka będzie odpowiadać *f* przeskanowanym wierszom (od 100 do 100 000).

`boundaries=granica1[,granica2[,granica3...]]`

Wartości graniczne długości kolumn dla raportu podsumowującego.

`lastrpt={y|n}`

Nakazuje wygenerowanie raportu na podstawie poprzedniego skanowania, bez wykonywania bieżącego skanu.

`log=plik_raportu`

Nazwa dla plików raportu: podsumowania, błędów i wyjścia ekranowego. Narzędzie *csscan* generuje trzy pliki raportu o takiej samej nazwie, ale z różnymi rozszerzeniami:

.txt dla raportu podsumowania, *.err* dla raportu błędów i *.out* dla raportu wyjścia ekranowego. Domyślną nazwą pliku jest *scan*.

`parfile=plik_parametrów`

Nazwa pliku parametrów dla narzędzia *csscan*. Szczegółowy opis korzystania z tego parametru podany został w przykładach.

`preserve={y|n}`

Powoduje zachowanie (wartość *y*) lub nadpisanie (wartość *n*) statystyk zebranych podczas poprzedniego skanowania bazy.

`lcsd={y|n}`

Określa, czy do danych ma być stosowane automatyczne wykrywanie języka i zestawu znaków w zakresie wskazanym przez parametr *lcsdata*.

`lcsdata={lossy|truncation|convertible|changeless|all}`

Określa zakres automatycznego wykrywania języka i zestawu znaków: stratny (*lossy*), odcięcie (*truncation*), z konwersją (*convertible*), bez zmian (*changeless*) lub wszystkie (*all*).

`help={y|n}`

Wyświetla listę parametrów polecenia *csscan* bez wykonania skanowania.

Wymagania

Przed skorzystaniem z narzędzia *csscan* należy zalogować się do bazy jako SYS i uruchomić skrypt `$ORACLE_HOME/rdbms\admin\csminst.sql`. Skrypt ten tworzy użytkownika bazy danych CSMIG oraz tabele systemowe wymagane przez *csscan*.

Wskazówka

Domyślną przestrzenią tabel dla użytkownika CSMIG jest SYSTEM. Oracle zaleca zmianę domyślnej przestrzeni tabel tego użytkownika w skrypcie *csmnst.sql* przed jego uruchomieniem.

Przykłady

Poniższy przykład pokazuje wywołanie polecenia *csscan* powodujące skanowanie tabeli emp ze schematu użytkownika scott. Celem skanowania jest ustalenie, czy możliwa jest konwersja zestawu znaków z WE8MSWIN1252 na UTF8.

```
C:\>csscan system/manager tabel=scott.emp
fromchar=WE8MSWIN1252 tochar=UTF8 log=scan_emp array=102400
process=1
```

```
Character Set Scanner v2.0 : Release 10.1.0.2.0 - Production
on Tue Aug 31 21:45:39 2004
```

```
Copyright (c) 1982, 2004, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 -
Production
With the Partitioning, OLAP, and Data Mining options
```

```
Enumerating table to scan...
. process 1 scanning SCOTT.EMP[AAAMVBAAEAAAAZAAA]
```

```
Creating Database Scan Summary Report...
Creating Individual Exception Report...
Scanner terminated successfully.
```

Narzędzie *csscan* tworzy trzy pliki wyjściowe: raport podsumowujący (z rozszerzeniem *.txt*), raport błędów (z rozszerzeniem *.err*) i zapis wyjścia ekranowego (z rozszerzeniem *.out*). Wszystkie pliki mają taką samą nazwę (podaną jako wartość parametru *log*), ale różnią się rozszerzeniami.

Pliki parametrów i tryb interaktywny

Parametry dla narzędzia *csscan* można podawać na jeden z trzech sposobów: w linii poleceń, w pliku parametrów lub w trybie interaktywnym. Poprzedni przykład pokazał sposób podawania parametrów w linii poleceń. Inną możliwością jest podanie pożądanych wartości w tekstowym pliku parametrów, a następnie wskazanie tego pliku w parametrze *parfile* polecenia *csscan*, zgodnie z następującym przykładem:

```
csscan parfile=csscan.par
```

Aby osiągnąć taki sam efekt, jak w poprzednim przykładzie z linią poleceń, plik parametrów powinien zawierać następujące wpisy:

```
userid=system/manager  
table=scott.emp  
fromchar=WE8MSWIN1252  
tochar=UTF8  
log=scan_emp  
array=102400  
process=1
```

Trzecią możliwością jest skorzystanie z trybu interaktywnego poprzez wywołanie narzędzia *csscan* bez żadnych parametrów. Program poprosi wtedy o podanie nazwy użytkownika, trybu skanowania (pełne, dla użytkownika lub dla tabeli) oraz wartości parametrów *tochar*, *array* i *process*.

Wskazówka

Podanie niedostatecznej ilości parametrów w linii poleceń lub w pliku parametrów sprawi, że program zażąda uzupełnienia brakujących wartości w trybie interaktywnym.

dbfsize

Narzędzie *dbfsize* pozwala określić rozmiar bloku i ilość bloków we wskazanym pliku bazy danych. *dbfsize* można używać dla plików danych, plików dziennika powtórzeń i plików kontrolnych.

Wskazówka

Narzędzie *dbfsize* nie jest dostępne na platformach Windows.

Składnia

```
dbfsize nazwa_pliku_bazy_danych
```

Parametry

nazwa_pliku_bazy_danych

Nazwa pliku, która ma być poddany analizie.

Przykłady

```
$ dbfsize control01.ctl
Database file: control01.ctl
Database file type: file system
Database file size: 184 16384 byte blocks
```

```
$ dbfsize redo01.log
Database file: redo01.log
Database file type: file system
Database file size: 20480 512 byte blocks
```

```
$ dbfsize users01.dbf
Database file: users01.dbf
Database file type: file system
Database file size: 768 8192 byte blocks
```

W wyniku działania *dbfsize* zwracane są trzy wartości. Pierwszą z nich jest nazwa pliku podana jako parametr. Drugą wartością jest rodzaj pliku: `file system` (system plików) lub `raw` (surowe

dane). Trzecią częścią wyniku są dwie liczby odpowiadające ilości bloków w pliku i wielkości jednego bloku. Wielkości i ilości bloków mogą być różne dla różnych plików.

dbhome

Narzędzie *dbhome* zwraca katalog Oracle Home używany przez sesję o podanym identyfikatorze SID. Jego działanie polega na odczytaniu wartości parametru ORACLE_HOME z pliku *oratab*.

Wskazówka

Narzędzie *dbhome* nie jest dostępne na platformach Windows.

Składnia

`dbhome [SID]`

Parametry

SID

Identyfikator sesji Oracle. Parametr ten jest opcjonalny. Wywołanie *dbhome* bez jego podania spowoduje zwrócenie katalogu Oracle Home odpowiadającego bieżącej wartości zmiennej systemowej ORACLE_SID.

Przykłady

Poniższe przykłady ilustrują sposób korzystania z *dbhome* z parametrem i bez niego.

```
$ echo $ORACLE_SID
TEST9I
$ dbhome
/u01/app/oracle/product/9.2.0
$ dbhome TEST10G1
/u01/app/oracle/product/10.1.0
```

dbshut

Narzędzie *dbshut* zamyka instancje bazy danych Oracle. Zamknięcie instancji w taki sposób wymaga istnienia odpowiadających im wpisów w pliku *oratab*, przy czym ostatnią wartością wpisu każdej instancji powinno być Y. Polecenie *dbshut* zamyka wszystkie instancje posiadające takie wpisy.

Wskazówka

Narzędzie *dbshut* nie jest dostępne na platformach Windows.

Składnia

dbshut

Przykłady

Przed zamknięciem instancji za pomocą *dbshut* należy się upewnić, że instancja posiada odpowiedni wpis w pliku *oratab* na serwerze obsługującym bazę danych. W systemach Unix i Linux plik ten znajduje się najczęściej w katalogu */etc* lub */var/opt/oracle*.

```
$ echo $ORACLE_SID
TEST10G1
$ grep $ORACLE_SID /etc/oratab
TEST10G1:/u01/app/oracle/product/10.1.0:Y
```

Linia pobrana z pliku *oratab* zawiera trzy różne pola: identyfikator sesji dla instancji, katalog *ORACLE_HOME* oraz wartość Y lub N. Jeśli ostatnie pole zawiera Y, narzędzie *dbshut* (i omawiane w dalszej części książki *dbstart*) uwzględni daną instancję w swych działaniach. Wartość N spowoduje pominięcie instancji przez *dbshut* i *dbstart*.

\$ **dbshut**

SQL*Plus: Release 10.1.0.2.0 – Production on Wed Aug 25
18:45:44 2004

Copyright (c) 1982, 2004, Oracle. All rights reserved.

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> Disconnected from Oracle Database 10g Enterprise
Edition Release 10.1.0.2.0 – Production
With the Partitioning, OLAP and Data Mining options

Database "TEST10G1" shut down.

Wskazówka

Narzędzie *dbshut* dokonuje zamknięcia instancji poleceniem *shutdown immediate*.

Powyższy przykład pokazuje wykorzystanie polecenia *dbshut* w linii poleceń, ale jest ono najczęściej stosowane do automatycznego zamykania instancji Oracle podczas zamykania systemu operacyjnego. Informacje o dokładnym sposobie zastosowania automatycznego zamykania instancji Oracle znajdują się w dokumentacji Oracle dla konkretnej platformy.