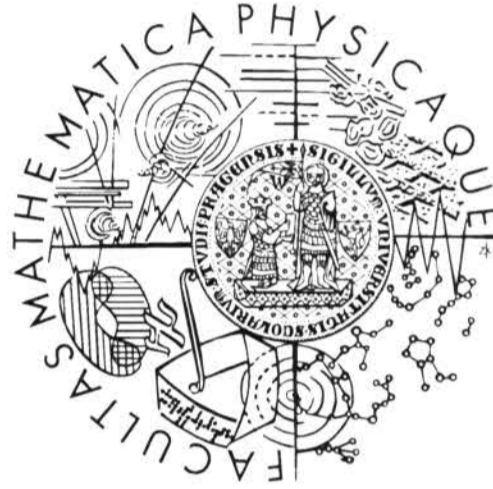


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Eva Ondrášková

Oracle Visual Tools

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. David Hoksza

Studijní program: Informatika, Obecná informatika

2009

Ráda bych na tomto místě poděkovala vedoucímu mé bakalářské práce RNDr. Davidu Hokszoovi za vedení mé práce. Dále děkuji Ing. Davidu Krchovi, Technology Sales Consultant Oracle Czech, za konzultace a vstřícný přístup. A také závěrem patří velké poděkování Ing. Jiřímu Ondráškovi a Ing. Haně Ondráškové za podporu.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce

V Praze dne 6.8.2009

Eva Ondrášková

Eva Ondrášková

Název práce: Oracle Visual Tools
Autor: *Eva Ondrášková*
Katedra (ústav): *Katedra softwarového inženýrství*
Vedoucí bakalářské práce: *RNDr. David Hoksza*
e-mail vedoucího: hoksza@ksi.mff.cuni.cz

*Abstrakt: Cílem bakalářské práce bylo vytvořit pro produkt Oracle Database sadu uživatelsky orientovaných grafických nástrojů pro Export, Import a SQL*Loader a umožnit tak jednoduchý přístup k funkcionalitě těchto nástrojů, která je dostupná pouze na příkazové řádce. Dále pak umožnit ladění SQL příkazů vytvořením monitorovacího nástroje Profiler pro online monitorování SQL příkazů spouštěných v databázi. Program byl vypracován v jazyku C# pomocí nástroje Microsoft Visual Studio 2008 a odladěn na verzi OracleDatabase 10g Express Edition.*

*Klíčová slova: (Oracle, databáze, Export, Import, SQL*Loader, Profiler)*

Title: *Oracle Visual Tools*
Author: *Eva Ondrášková*
Department: *Department of Software Engineering*
Supervisor: *RNDr. David Hoksza*
Supervisor's e-mail address: hoksza@ksi.mff.cuni.cz

*Abstract: The main goal of the bachelor thesis was to create for the product of the Oracle Database a set of user-oriented graphical tools for Export, Import and SQL*Loader offer an easy acces to functionality of these tools that is available only on the command line and then to enable an easier tuning of SQL statements creating monitoring tool Profiler that facilitates the online monitoring of the statements lanuched in the database. The program was created in C# language with Microsoft # Visual Studio 2008 tool and debugged in OracleDatabase 10g Express Edition.*

*Keywords: (Oracle, database, Export, Import, SQL*Loader, Profiler)*

1.	Úvod	5
1.1	Stručné uvedení do problematiky ladění SQL dotazů	6
1.2	Architektura zpracování SQL příkazu	6
2.	Analýza zadaného úkolu	7
2.1	Analýza pro Export, Import a SQL*Loader	7
2.2	Analýza pro monitorování a zobrazování statistik	8
3.	Metody monitorování, shromažďování statistik a jejich srovnání....	9
3.1	Placené nástroje Oracle pro online monitorování	9
3.2	Neplacené nástroje pro online monitorování	9
3.3	Nástroje jež jsou součástí implementace OracleDatbase	9
4.	Varianty řešení a výběr řešení	10
4.1	Export, Import a SQL*Loader	10
4.2	Profiler	10
5.	Řešení.....	11
5.1	Export, Import, SQL*Loader	11
5.2	Profiler	11
6.	Základní omezující podmínky	17
7.	Uživatelská dokumentace	18
7.1	Instalace.....	18
7.2	Nástroje pro originální Import dat a originální Export dat.....	19
7.3	Nástroj SQL*Loader	19
7.4	Popis funkčních okruhů řešených programem	20
8.	Export a Import- popis parametrů	30
8.1	Módy Importu a Exportu	30
9.	Programátorská dokumentace	48
10.	Závěr	51
11.	Seznamliteratury	52

1. Úvod

Pojem databáze se postupně čím dál více dostává do podvědomí široké veřejnosti. Lidé mají potřebu evidovat a shromažďovat informace už od pradávna a databáze se tak stala součástí běžného života moderního člověka. Setkáme se s ní jak v podnikových informačních systémech, tak i ve školství, zdravotnictví, dopravě..

Předchůdcem databází byly papírové kartotéky, které umožňovaly zapsání, zatřídění a následné vyhledání potřebných informací, přičemž veškeré operace prováděl člověk. Postupem času databázová problematika stejně jako jiné technologie prošla vývojem, díky kterému jsme nyní schopni během stále kratší doby získávat jak velké množství dat, tak i jejich specifickou podmnožinu. Vývoj databázových technologií jde stále dobředu, neboť se snaží pokrýt stále rostoucí nároky uživatelů, s čímž jdou ruku v ruce požadavky na výkon.

Pro vytváření a manipulování s daty slouží standardizovaný dotazovací jazyk SQL(Structured Query Language) Práce s databází je tedy umožněna pomocí tohoto jazyku.

Existuje několik druhů databází. Mým úkolem bylo naprogramovat nástroje pro databázi OracleDatabase..

Jednou z primárních potřeb uživatele jsou výkonnostní a s tím související časové nároky. Návrh aplikace a problémy s příkazy SQL způsobují většinu problémů s výkonem v jinak správně navržené a nakonfigurované databázi.

Klíčem pro ladění příkazů jazyka SQL je minimalizovat cesty získávání dat, které aplikace využívá při vyhledávání dat. I jediný dotaz, který není vyladěn na maximum, může mít vliv na výkon celé databáze v případě, že je prováděn dostatečně často. Dotaz jazyka SQL, který generuje málo nebo vůbec žádné V/V operace přesto spotřebovává prostředky procesoru.

Další primární potřebou je i manipulace s daty, jak pro zálohování databáze, tak i pro přenos dat mezi různými databázemi pracujícími na různých systémech. K tomu slouží nástroje Export, Import, Data Pump Export a Data Pump Import. K naplnění databáze již existujícími daty ve formě např.excel tabulek slouží SQL*Loader.

Již existující uživatelská rozhraní pro Import, Export a SQL*Loader neumožňují použití všech parametrů zadávaných v příkazové řádce.

Cílem bakalářské práce tedy bylo vytvořit sadu uživatelsky orientovaných grafických nástrojů pro Export, Import a SQL*Loader a monitorovací nástroj Profiler pro online monitorování SQL příkazů spouštěných v databázi.

1.1 Stručné uvedení do problematiky ladění SQL dotazů

Cílem návrhu databáze je, aby fyzická omezení (propustnost I/O operací, velikost paměti, výkon procesoru) neměly zásadní vliv na výkon aplikace.

Ladění výkonu je částí životního cyklu každé databázové aplikace. Většina problémů s výkonem nevzniká izolovaně, ale jako důsledek špatného návrhu aplikace. Ladění se v takovém případě musí zaměřit na nalezení a odstranění problémů, které vedou k nízkému výkonu. Ladění je posledním krokem vývojového cyklu, které zahrnuje plánování, implementaci a monitorování. Ladění začíná analýzou toho co aplikace/databáze dělá, čím tráví nejvíce času a jaké činnosti spotřebovávají nejvíce jakých zdrojů. Poté se volí co se bude ladit:

- ladění příkazů jazyka SQL
- ladění využití paměti
- ladění využití úložiště dat
- ladění příkazů pro manipulaci s daty
- ladění fyzického a logického úložiště
- ladění síťového provozu

1.2 Architektura zpracování SQL příkazu

Poté, co uživatel spustí SQL příkaz, provede se několik kroků. Prvním krokem je tzv. parsování, kdy se provádí syntaktická a sémantická analýza dotazu. Poté optimalizátor navrhne co nejefektivnější způsob získání výsledku dotazu. Využije jednu ze dvou metod: cost-based (CBO) nebo rule-based (RBO) a vytvoří optimalizační plán. RBO se již dnes neuvádá, nicméně v databázi zůstal. Optimalizační plán poté zpracuje Row Source Generator, který vytvoří exekuční plán pro daný SQL dotaz. Exekuční plán má formu stromu, jehož uzly jsou tvořeny „row sources“ – tzv. zdroji řádků. Vykonáváním jednotlivých řádků exekučního plánu se získá výsledek zadaného SQL příkazu.

Cílem ladění je jak redukovat čas odezvy (response time) pro koncové uživatele systému, tak i redukovat zdroje, které se používají pro opakovanou činnost. Toho se dá dosáhnout několika cestami: snížit zátěž, vyvážit zátěž, paralelní zátěží. Pro snížení zátěže je možné např. využít vhodného indexování, Touto cestou se dají ušetřit zdroje.

Vyvážení zátěže se dá dosáhnout tím, že by se provádění jobů a jiných akcí naplánovalo na noční hodiny, kdy je obvykle vytížení databáze menší.

SQL ladění je jedno z časově nejnáročnějších úkolů jak pro administrátory tak i vývojáře.

Ladění SQL příkazy zahrnuje 3 kroky:

- identifikaci zatěžujících SQL příkazů, které jsou odpovědné za velký podíl na zatížení databáze a systémových zdrojů
- ověření exekučních plánů
- implementace akcí, které napomohou ke generování lepších exekučních plánů pro tyto nízko výkonné SQL dotazy

2. Analýza zadaného úkolu

Oracle je dodáván na trh v několika funkčně i výkonově odlišných verzích a též odlišně licencovaných:

- Oracle Express Edition
- Oracle Standard Edition One
- Oracle Standard Edition
- Oracle Enterprise Edition

Pouze Oracle Express Edition je volně šiřitelnou verzí. Ostatní verze jsou licencovány zpravidla podle počtu současně přistupujících uživatelů a podle počtu CPU. Další dělení Oracle Database je dle čísla verze. V současnosti jsou podporovány verze 9i, 10g a 11g.

Na obrázku je přehled funkčních možností jednotlivých verzí Oracle:

Key Feature Summary	Express Edition 10g	Standard Edition One	Standard Edition	Enterprise Edition
Maximum	1 CPU	2 Sockets	4 Sockets	No Limit
RAM	1GB	OS Max	OS Max	OS Max
Database Size	4GB	No Limit	No Limit	No Limit
Windows	●	●	●	●
Linux	●	●	●	●
Unix		●	●	●
64 Bit Support		●	●	●
Application Development Read More				
Oracle SQL Developer Graphical tool that enhances productivity and simplifies database development tasks. Read More	●	●	●	●
Manageability Read More				
Enterprise Manager Manage and monitor all applications and systems based on the Oracle stack via a single, integrated console. Read More		●	●	●

(zdroj: http://www.oracle.com/database/product_editions.html www.oracle.com).

Z tabulky vyplývá, že všechny verze kromě Oracle Database 10g Express Edition disponují komplexním nástrojem Oracle Enterprise Manager, jehož funkčnost částečně pokrývá zadání bakalářské práce.

2.1 Analýza pro Export, Import a SQL*Loader

Export, Import a SQL*Loader jsou originální nástroje Oracle. Jsou dostupné ve formě souborů exp.exe, imp.exe a sqlldr.exe na tom pc, kde je nainstalovaný klient Oracle. Jsou spustitelné v dávkovém souboru, do kterého se zadá název příslušného souboru exp.exe, imp.exe nebo sqlldr.exe, dále se předvyplní tzv. connection string a parametrický soubor. Přestože jsou spustitelné na příkazové řádce, není vhodné parametry a jejich hodnoty vkládat přímo na jednu řádku za typ nástroje a connection string z důvodu omezení délky příkazové řádky. Nabízí se řešení naplnit parametry z grafického rozhraní a ty pak předat utilitě a spustit v dávkovém souboru s odchyčením chyb a logu zpět do grafického rozhraní.

2.2 Analýza pro monitorování a zobrazování statistik

Zde se musely řešit dva problémy:

- zachytit veškeré SQL příkazy libovolného aktivního tzv. sezení (session)
- získat statistiky jednotlivých SQL příkazů

3. Metody monitorování, shromažďování statistik a jejich srovnání:

3.1 Placené nástroje Oracle pro online monitorování:

Automatic Workload Repository- nabízí se v rámci Enterprise manager diagnostic Pack Oracle. Poskytuje cenné informace pro rychlé řešení výkonových problémů.

Automatic Database Diagnostic Monitor- pomáhá zjišťovat a diagnostikovat širší spektrum výkonových problémů- analyzuje data AWR, vyhodnocuje např. CPU zátěž, využití paměti, využití V/V operací, SQL příkazy náročné na zdroje, PL/SQL náročné na zdroje a další.

3.2 Neplacené nástroje pro online monitorování:

Nástroj *Statspack*- shromažďuje informace na úrovni celé databáze, umožňuje snímkování stavu databáze v definovaných intervalech, uložení těchto informací a jejich následné vyhodnocení.

Sql Developer- nástroj určený především pro vývoj, umožňuje však také zjistit exekuční plán, trasovat session, profilovat PL/SQL. Především jde o uživatelské prostředí pro jiné nástroje.

3.3 Nástroje jež jsou součástí implementace OracleDatabase

DBMS_PROFILER- nástroj určený pro ladění jazyka PL/SQL, neřeší jazyk SQL. Obsahuje informace o tom, kolikrát byla daná řádka vykonána, celkový čas, který vykonávání zabralo a minimální a maximální čas, který trvalo vykonání pro jednotlivé řádky. Statistické informace jsou uloženy v tabulkách v databázi.

DBMS_SQLTUNE- nástroj určený k usnadnění ladění SQL příkazu a k nalezení optimálního SQL exekučního plánu. Může diagnostikovat SQL cache, AWR, nebo příkazy vložené uživatelem. API pro tento nástroj je Tuning Advisor, který je dostupný pomocí Oracle Enterprise Manager(OEM)

DBMS_STATS- pomocí tohoto balíku lze sledovat a modifikovat statistiky optimalizéru shromažďované pro databázové objekty, nejde tedy o statistiky pro diagnostiku administrátora člověkem, ale o data podle kterých optimalizátor určuje náklady operaci - počet a prům. velikost záznamu, min, max hodnota sloupce, počet unikátních hodnot sloupce, histogram rozložení hodnot...

DBMS_MONITOR- tento nástroj obsahuje procedury pro povolování a zakazování shromažďování statistik. Nabízí funkcionalitu staršího DBMS_SUPPORT. Obsahuje procedury pro trasování podle session id, client identifier, trasování založené na kombinaci module name a action name nebo trasování celé databáze. Je to jediná metoda pro verzi Express Edition, která umožňuje odchytit všechny SQL dotazy.

4. Varianty řešení a výběr řešení

4.1 Export, Import a SQL*Loader

Řešením těchto nástrojů je zadávání příkazů exp.exe, imp.exe a sqlldr.exe do batch souboru, který je následně spolu s připojovacím stringem a parametrickým souborem se seznamem parametrů spuštěn na příkazové řádce.

4.2 Profiler

Trasování SQL příkazů přímo do tabulky v databázi umožňuje pohodlnější manipulaci s daty a operace nad nimi.

Pokud bychom znali SQL příkaz, stačilo by před zadáním zadat start monitoring, provést příkaz, provést stop monitoring a pomocí pohledů v\$statname a v\$sesstat odečíst statistiky před a po vykonání dotazu a tím získat nové statistiky. Získání statistik by se dalo realizovat následujícím dotazem:

```
SELECT ses.username,n.name,stat.*
FROM v$session ses, v$sesstat stat, v$statname n
WHERE ses.sid=stat.sid and stat.statistic#=n.statistic#;
```

Prvním problémem monitorování v ExpressEdition je tedy získat veškeré SQL příkazy uživatelů, neboť tato nejnižší edice neobsahuje tabulky s historií SQL dotazů. Je dostupné pouze System Global Area (SGA), ve kterém není možné pořádkem časový símek.

První myšlenkou k získání SQL příkazů bylo propojit pohledy v\$session a v\$sql následujícím způsobem:

```
select u.sid, u.username s.sql_text from v$sql s,
v$session u where s.hash_value = u.sql_hash_value order
by u.sid;
```

Tímto lze však získat pouze aktuálně prováděný SQL příkaz. V úvahu tedy připadalo nad tímto vytvořit JOB a nastavit sekvence provádění na zadaný časový interval. Pokud bychom se spokojili s tím, že SQL příkazy, které se vykonávají kratší dobu než je specifikovaný časový interval pro nás nejsou podstatné, pak by se o tomto řešení dalo uvažovat, celkově jsem ho ale vyhodnotila za nevyhovující. Úlohu neřeší, neboť nemáme žádnou záruku, že by se podařilo zachytit všechny SQL příkazy. Využití by se spíše našlo ve chvíli, kdy administrátor přijde k nové databázi a chce si udělat obrázek, co se v databázi děje.

Nejlepším řešením se tedy ukázalo být použití balíku *DBMS_MONITOR*. Je dostupný i ve verzi ExpressEdition a jeho výhodou je zachycení všech SQL příkazů vykonaných v databázi a shromažďování základních statistik, pomocí nichž se dají odhalit problematická místa snižující výkon. Monitorovat se dá celá databáze, vybrané sessions, vybraný client_identifier a nebo vybrané module a action name.

5. Řešení

Celou úlohu jsem rozdělila na dva řešené okruhy spojené však do jednoho programu.

První okruh se zabývá naimplementování grafického prostředí na vyplňování parametrických hodnot příkazů Export, Import a SQL*Loader.

Druhou částí je pokus o online sledování vykonávaných SQL příkazů.

Vzhledem k dostupnosti jsem použila OracleDatabase10g Express Edition, vývojový nástroj Microsoft Visual Studio 2008 a jeho programovací jazyk C# s knihovnou tříd Microsoft Windows Forms a s rozšířením o volně šiřitelný zásuvný modul Oracle Data Provider pro přístup k datům v OracleDatabase.

V práci jsem se soustředila na database OracleDatabase10g Express Edition, neboť vyšší verze nabízí balíčky jako rozšíření nad databází, které tyto nástroje podporují, ale přesto stále nenabízí takové parametry, jako originální Export, Import a SQL*Loader, nabízí však grafické rozhraní a rozsáhlé statistiky.

5.1 Export, Import, SQL*Loader

Problém je řešen předáváním parametrů zadaných uživatelem z grafického rozhraní do parametrického.souboru.

V případě pokud to má smysl je provedeno interaktivní zavedení seznamu tabulek a uživatelů přímo z databáze, do které se uživatel připojí.

Před odesláním parametrů je provedena základní syntaktická kontrola a navíc před odesláním do dávkového souboru je možnost editace připravených vygenerovaných parametrů.

S tímto souborem je pak v dávce spuštěna utilita imp.exe, exp.exe nebo sqlldr.exe. Výstupy běžící utility jsou přesměrovány do grafického prostředí programu a jsou viditelné v okně pro log.

Uživatel má možnost sledovat historii provedených činností, ukládat, číst a obnovovat parametry spouštěné dávky ze souboru, který je programem obsluhován.

5.2 Profiler

Jak jsem již zmínila výše, vybrala jsem jako způsob řešení balíček dbms_monitor. Umožňuje trasování podle sessions, client_id, module a action name nebo celé databáze.

Možností dosáhnoutí vytvoření trace souboru a shroaždování statistik je však celá řada:

```
SQL> ALTER SESSION SET sql_trace=TRUE;
SQL> ALTER SESSION SET sql_trace=FALSE;
```

```
SQL> EXEC DBMS_SESSION.set_sql_trace(sql_trace => TRUE);
SQL> EXEC DBMS_SESSION.set_sql_trace(sql_trace => FALSE);
```

```
SQL> ALTER SESSION SET EVENTS '10046 trace name context
forever, level 8';
SQL> ALTER SESSION SET EVENTS '10046 trace name context off';
```

```

SQL> EXEC DBMS_SYSTEM.set_sql_trace_in_session(sid=>123,
serial#=>1234, sql_trace=>TRUE);
SQL> EXEC DBMS_SYSTEM.set_sql_trace_in_session(sid=>123,
serial#=>1234, sql_trace=>FALSE);

SQL> EXEC DBMS_SYSTEM.set_ev(si=>123, se=>1234, ev=>10046,
le=>8, nm=>' ');
SQL> EXEC DBMS_SYSTEM.set_ev(si=>123, se=>1234, ev=>10046,
le=>0, nm=>' ');

```

(zdroj: http://www.dba-oracle.com/plsql/t_plsql_trace.htm)

Formát trace souboru by měl být stejný, ať se trasování spustí přes *DBMS_MONITOR*, `alter session set SQL_TRACE=true`, nebo nastavení *EVENT=10046*... Pomocí *DBMS_MONITOR* se však nemusí na začátku sledované session spouštět speciální příkaz pro zapnutí trasování přímo v dané session a trasování se dá zapnout pro jinou session nebo client identifier, module a action name nebo ceou databázi.

Soustředila jsem se na monitorování jednotlivých sessions jako nejmenších celků oproti monitorování celé databáze.

Monitorování jedné session spustíme příkazem:

```

DBMS_MONITOR.SESSION_TRACE_ENABLE(
  session_id  IN  BINARY_INTEGER DEFAULT NULL,
  serial_num  IN  BINARY_INTEGER DEFAULT NULL,
  waits       IN  BOOLEAN DEFAULT TRUE,
  binds       IN  BOOLEAN DEFAULT FALSE)

```

Monitorování vypneme příkazem:

```

DBMS_MONITOR.SESSION_TRACE_DISABLE(
  session_id  IN  BINARY_INTEGER DEFAULT NULL,
  serial_num  IN  BINARY_INTEGER DEFAULT NULL);

```

Kde *session_id* je číslo session, kterou chceme monitorovat a *serial_num* specifikujeme pro případ, že by session s jedním id zanikla a poté zase vznikla. Pokud specifikujeme *serial_num* na null, pak je monitorována každá session se zadaným *session_id*.

Každý časový okamžik session je monitorován. Jestliže session pracuje, spotřebovává čas CPU, jestliže session čeká, spotřebovává wait time. Wait time se dělí na dva případy. Session buď na něco čeká, nebo je v nečinnosti(idle time). Když Oracle spustí SQL příkaz, tento příkaz někdy musí čekat na specifickou událost- wait event.

Pokud nastavíme parametr *waits* na true, nastavíme sledování wait events. Nastavením parametru *binds* na true umožníme sledovat i proměnné.

DBMS_MONITOR shromažďuje informace do trace říu, defaultně umístěných v adresáři UDUMP v ORACLE_HOME. Cesta k těmto souborům se dá jednoduše zjistit pomocí pohledů v\$session, v\$process, v\$instance a v\$parameter SQL příkazem:

```
SELECT a.trace_path || '\\\ ' || d.instance_name || '_' ||
b.trace_file
FROM (SELECT VALUE trace_path FROM v$parameter
WHERE NAME = 'user_dump_dest') a,
SELECT instance_name FROM v$instance) d,
SELECT 'ora_' || spid || (SELECT NVL2(VALUE, '_' || VALUE,
NULL) AS VALUE FROM v$parameter
WHERE NAME = 'tracefile_identifier') || '.trc' trace_file
FROM v$process p, v$session s
WHERE (p.addr = s.paddr AND s.sid=xxx AND s.serial#=yyy))b
```

Kdy cesta je uložena ve v\$parameter v proměnné 'user_dump_dest', název souboru sestává z názvu instance_ora_id procesu.

Trace soubor je vztažen ke konkrétní session. Pokud vypneme a znovu zapneme trace v rámci jedné session, Oracle bude zapisovat do toho samého souboru. Pokud ho mezitím smažeme, daná session už nezapíše nic.

Obsah trace souboru:

příklad:

```
Dump file c:\oraclexe\app\oracle\admin\xe\udump\xe_ora_2704.trc
Sun Aug 02 15:20:16 2009
ORACLE V10.2.0.1.0 - Production vsnsta=0
vsnsql=14 vsnxtr=3
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
Windows XP Version V5.1 Service Pack 2
CPU : 2 - type 586
Process Affinity : 0x00000000
Memory (Avail/Total): Ph:150M/1022M, Ph+PgF:1119M/2456M, VA:1606M/2047M
Instance name: xe
```

Redo thread mounted by this instance: 1

Oracle process number: 20

Windows thread id: 2704, image: ORACLE.EXE (SHAD)

*** 2009-08-02 15:20:16.481

*** SERVICE NAME:(SYS\$USERS) 2009-08-02 15:20:16.309

*** CLIENT ID:(eva:127.0.0.1) 2009-08-02 15:20:16.309

*** SESSION ID:(31.257) 2009-08-02 15:20:16.309

=====

PARSING IN CURSOR #2 len=37 dep=1 uid=0 oct=3 lid=0 tim=107681468090
hv=1398610540 ad='20ae9760'

select text from view\$ where rowid=:1

END OF STMT

PARSE #2:c=0,e=116477,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=4,tim=107681468075

EXEC #2:c=0,e=110245,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=4,tim=107681798557

FETCH #2:c=0,e=89,p=0,cr=2,cu=0,mis=0,r=1,dep=1,og=4,tim=107681803410

```

STAT #2 id=1 cnt=1 pid=0 pos=1 obj=63 op='TABLE ACCESS BY USER ROWID VIEW$
(cr=1 pr=0 pw=0 time=71 us)'
=====
PARSING IN CURSOR #3 len=43 dep=0 uid=5 oct=3 lid=5 tim=107681847549
hv=1991533359 ad='20a90e1c'
select tablespace_name from dba_tablespaces
END OF STMT
PARSE #3:c=93750,e=545107,p=0,cr=2,cu=0,mis=1,r=0,dep=0,og=1,tim=107681847540
EXEC #3:c=0,e=74,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=1,tim=107681869624
WAIT #3: nam='SQL*Net message to client' ela= 10 driver id=1952673792 #bytes=1 p3=0
obj#=37 tim=107681894384
FETCH #3:c=0,e=157,p=0,cr=7,cu=0,mis=0,r=5,dep=0,og=1,tim=107681899989
*** 2009-08-02 15:20:35.621
WAIT #3: nam='SQL*Net message from client' ela= 18880225 driver id=1952673792
#bytes=1 p3=0 obj#=37 tim=107700785223
STAT #3 id=1 cnt=5 pid=0 pos=1 obj=16 op='TABLE ACCESS FULL TS$ (cr=7 pr=0 pw=0
time=138 us)'
=====

```

Popis trace souboru

V hlavičce se nachází základní informace jako je název trace souboru, datum vytvoření souboru, tedy první spuštění monitorování na dané session, verze Oracle database, jméno instance, service name, client id a session id.

Následuje SQL příkaz s přiděleným cursorem, a dalšími parametry:

len = délka příkazu

dep= hloubka příkazu- pokud příkaz nebyl rekursivní, má hloubku příkazu 0, jinak hodnotu větší než 0

uid = id schématu, pod kterým bylo provedeno parsování SQL dotazu id schématu

oct = typ příkazu: SELECT, INSERT, UPDATE, DELETE

lid = id privilegovaného uživatele

hv = hash id

ad = adresa SQL příkazu pro dohledání v pohledu v\$sqlarea

Následuje parsování příkazu, provedení a získání dat. Statistiky k těmto akcím najdeme na řádcích začínající PARSE, EXECUTE a FETCH. Jejich statistiky jsou následující :

c = čas procesoru

e = uplynulá doba v setinách sekundy

p = počet fyzických čtení

cr = počet bufferů získaných pro CR čtení

cu = počet bufferů získaných v aktuálním nastavení

mis = kurzory ponechané v cache

r = počet zpracovaných řádků

dep = hloubka příkazu- pokud příkaz nebyl rekursivní, má hloubku 0, jinak >0

og = cíl optimizéru: 1=All_Rows, 2=First_Rows, 3=Rule, 4=Choose

tim = časová známka, dá se propojit s pohledem v\$timer

Statistiky STAT:

<CURSOR> = kurzor, ke kterému se vztahují statistiky

id = řádek vysvětluje plán dotazu. Toto je efektivita jednotlivých řádků v "execution tree"

cnt = počet řádků

pid = nadřazené id tohoto řádku

pos = pozice v "explain plan"

obj = id objektu tohoto řádku (pokud se jedná o základní objekt)
op = '.' přístupující operace k řádku

WAIT událost, na kterou se čeká
nam = jméno wait eventu
ela = uplynulý čas
p1 = proměnná p1 pro wait event
p2 = proměnná p2 pro wait even
p3 = proměnná p3 pro wait event

BIND proměnné
Value= hodnota proměnné

Zmiňovane pohledy:

v\$session- pohled na aktuální sessions

v\$sql- pohled obsahující statistiky nad shared SQL area. Shared SQL area je důležité pro vykonání každého SQL příkazu obsahující i exekuční plán.

v\$statname- pohled na seznam statistik

v\$sesstat- pohled na statistiky pro jednotlivé sessions

v\$parameter- pohled na inicializační parametry

v\$instance- pohled na stav současné instance

v\$process- pohled na procesy

6. Základní omezující podmínky

Licenční omezení na databázi a vývojové nástroje, použité operační systémy.

Pro řešení byly vybrány:

operační systém Microsoft Windows 32-bit

verze Oracle Database 10g Express Edition

Microsoft Visual Studio 2008 Professional Edition - studentská licence

programovací jazyk C#

Databáze byla instalována lokálně na notebooku s OS Microsoft Windows Home edition a lokálně byl aplikační program laděn na verzi Oracle Express10g , byl vyzkoušen na jiném PC.

7. Uživatelská dokumentace:

7.1 Instalace:

Minimální podmínky:

OS MS Windows XP SP2 nebo SP3 jak z řady Home tak i Professional

Microsoft .Net Framework 3.5 SP1

OracleClient 11g

Microsoft Installer

Instalace probíhá standardním způsobem pomocí instalačního souboru.

Oinstalace se provede přes ovládací panel Přidat nebo odebrat programy.

Programy pro Import - imp.exe Export - exp.exe a SQL*Loader - sqlldr.exe jsou umístěny v Oracle_Home adresáři klienta nebo databáze, v podadresáři s číslem verze v adresáři Bin. Je nutné mít nastavenou tuto cestu v systému.

7.2 Nástroje pro originální Import dat a originální Export dat

Tyto utility Oracle jsou původními nástroji Oracle Database ještě před nasazením tzv. Exportních a Importních pump a jsou ve všech verzích. Originální nástroje na rozdíl od "Oracle Pump" nepodporují všechny funkční možnosti používané verze databáze.

Nástroj imp.exe se však musí použít pro Import dat provedený spustitelným programem exp.exe nebo pro přenos dat do starší verze databáze.

Originální Export a Import umožňují přenos datových objektů mezi databázemi a to dokonce mezi různými platformami a různým hardware.

Spustitelné soubory jsou součástí jak Oracle Database tak i jejího klienta. Umožňují připojení i do vzdálené databáze a tak spouštět Import a Export objektů a dat z PC uživatele po síti (pokud má příslušná oprávnění).

7.3 Nástroj SQL*Loader

SQL*Loader vkládá data z externích zdrojů do tabulek Oracle databáze. Je to mocný datově-analytický orientovaný produkt s minimálním omezením na formáty dat ve vstupních datových souborech.

Jeho hlavní přednosti jsou:

- Zavádění dat přes síť. To znamená že můžete spustit SQL*Loader klient na jiném systému
- Zavádí data z vícenásobných datových souborů v průběhu jednoho běhu
- Zavádí data do více tabulek v rámci jednoho běhu
- Lze specifikovat znakové sady dat
- Data lze zavádět selektivně podle obsahu záznamu
- S daty lze před zavedením manipulovat užitím SQL funkcí
- Dokáže generovat jedinečné hodnoty vybraných sloupců
- K přístupu do souborů užívá operační systém
- Lze zavádět data z disků, pásek nebo prostřednictvím datových propojení
- Generuje sofistikované chybové zprávy usnadňující nápravu
- Zavádí libovolně složitá objektově-relační data
- Užívá druhotné datové soubory pro zavedení LOBů a kolekcí
- Užívá konvenční nebo přímou cestu pro zavedení dat. Konvenční cesta je velmi flexibilní, přímá cesta naopak může být velmi výkonná
- SQL*Loader se typicky používá ve spojení s vstupním parametrickým souborem řídící celý průběh vkládání dat, výstup je směřovaný do databáze, chybových a logovacích souborů, potencialně též do souborů s odhozenými daty které neprošly vstupním filtrem

7.4 Popis funkčních okruhů řešených programem:

V zásadě jsou řešeny čtyři funkčnosti v souladu se zadáním bakalářské práce : Import, Export, plnění dat přes SQL*Loader a Profiler, jako doplňková podmiňující činnost je Connect.

Funkčnost *Connect*

Vzhled:

Connection Name	Connection Details
o	SYSTEM@//localhost:1521/xe
o2	SYSTEM@//localhost:1521/xe
o3	SYSTEM@//localhost:1521/xe

Connection Name	o
Username	SYSTEM
Password	*****
Role	default
Hostname	localhost
Port	1521
SID	xe

Určení:

Slouží pro připojení do databáze.

Strom připojení ukazuje použitá a již dříve uložená připojení identifikovaná jejich aliasy. Nová připojení můžeme přidat po vyplnění parametrů v pravém okně stiskem tlačítka Add.

Lze se připojit do vzdálené nebo lokální databáze tzv. základní metodou připojení. Připojení TNS přes síťové aliasy není v této verzi podporováno, Pro správnou funkčnost musí být použit OracleClient11g, obsahující potřebné knihovny pro přístup do Oracle Database z API programu.

Pro nové připojení do databáze musíme tedy zadat následující údaje:

- Connection Name: Alias pro připojení do databáze
- Username: uživatelské jméno databázového uživatele připojícího se do databáze. Tento uživatel musí mít přidělená dostatečná oprávnění pro provádění operací exp, imp, sql*loader nebo pro činnosti spojené s monitorováním příkazů sql – zde jako sys
- Password: Heslo spojené s specifikovaným uživatelem
- Role: Nastavuje privilegia spojená s připojením. Pro uživatele kteří mají přidělena práva SYSDBA můžete zadat že se chcete připojit s těmito právy.
- Host Name: jméno nebo IP adresa počítaše hostující databázi Oracle.
- Port: číslo Listener portu, zpravidla 1521
- SID: Jméno databázové instance

Pro připojení lze buď vybrat alias připojení ze stromu nebo v pravém okně, vyplnit heslo a stisknou tlačítko Connect.

Úspěšné připojení je indikováno na informačním řádku zeleným podbarvením textu s verzí připojené databáze.

Odpojení je signalizováno červeným podbarvením textu s verzi připojené databáze.

Současně lze být připojen jen do jedné databáze a připojení je platné pro všechny naprogramované funkčnosti, tedy Export, Import, SQL*Loader i Profiler.

Ovládací prvky:

- Add - pro přidání připojení do stromu connect
- Delete - pro odstranění připojení ze stromu connect
- Clear - vymaže pole s údaji o připojení
- Connect - přečte parametry a pokusí se o připojení, zobrazí se úspěch či neúspěch
- Cancel - přeruší provádění připojení a zavře okno

Funkčnost *Export*

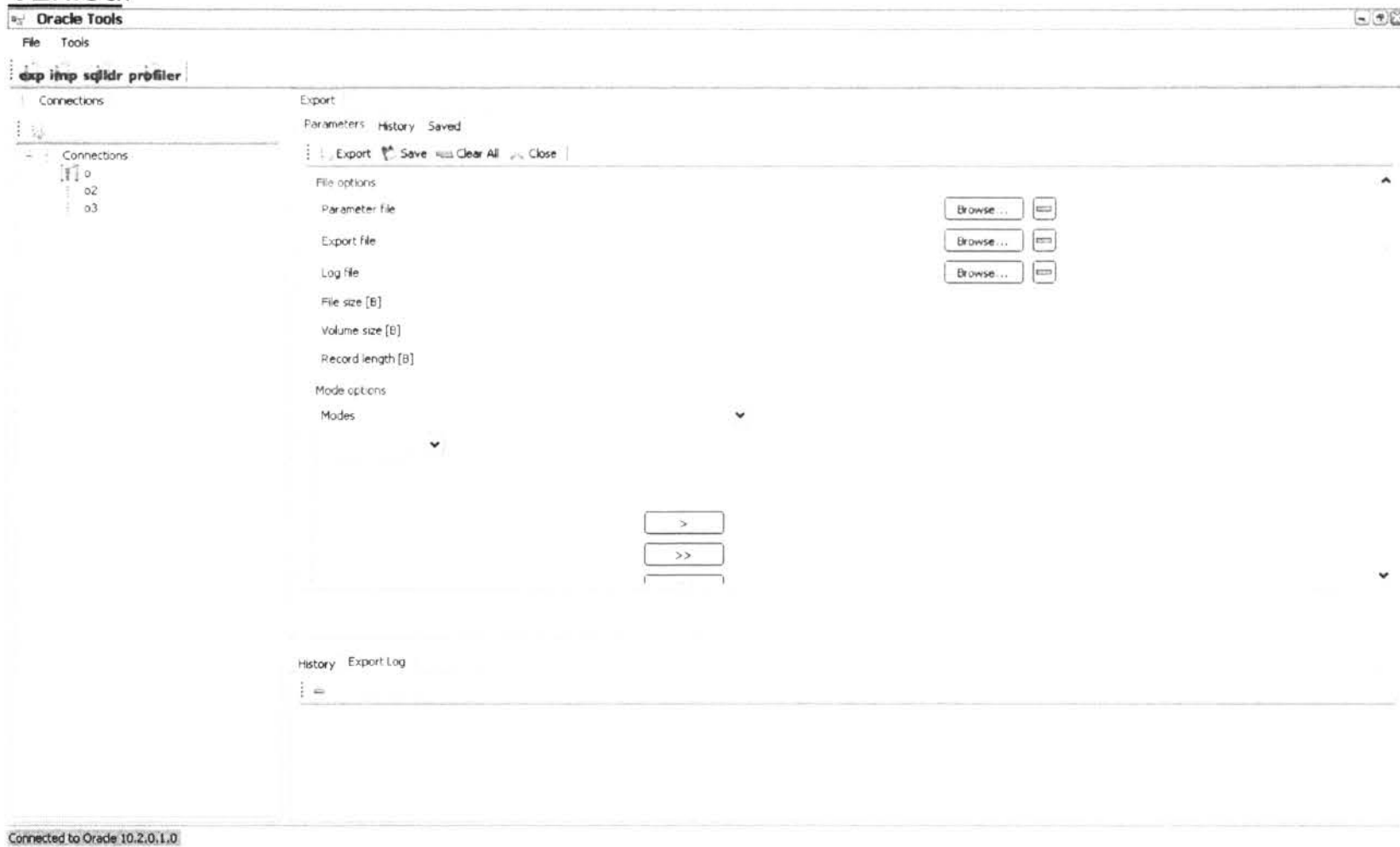
Tuto funkčnost lze spustit vybráním z menu Tools nebo kliknutím na ikonu. Pokud jsem již v databázi připojení zobrazí se přímo ovládací okno pro Export, jinak jsme požádání o připojení cestou viz „funkčnost Connect“

V záhlaví okna jsou tři záložky :

- Parameters - pro zadávání hodnot parametrů pro realizaci exportu
- History - pro prohlížení historie příkazů exportu
- Saved - umožňuje prohlížet , vybrat a použít již dříve zadané parametry

Záložka Parameters:

Vzhled:



Určení:

Slouží k zadávání parametrů pro Export, význam parametrů je uveden v kapitole „Popis parametrů pro Import a Export“ . U relevantních parametrů lze podle módu Exportu vybírat hodnoty pro Export z pohledu do databáze (pro mode Owners, Tables, Tablespaces). Požadované objekty lze postupně zadávat opakovaným výběrem. Po vyplnění požadovaných parametrů se tlačítkem Export spustí vykonávání příkazu. Před tím je však zobrazeno okno s hodnotami generovanými do parametrického souboru Exportu pro kontrolu a případnou editaci, pokud by uživatel nebyl spokojený s generovaným výsledkem. Po stisknutí tlačítka OK se vzkoná Export, po stisknutí tlačítka Cancel se provádění Exportu zruší.

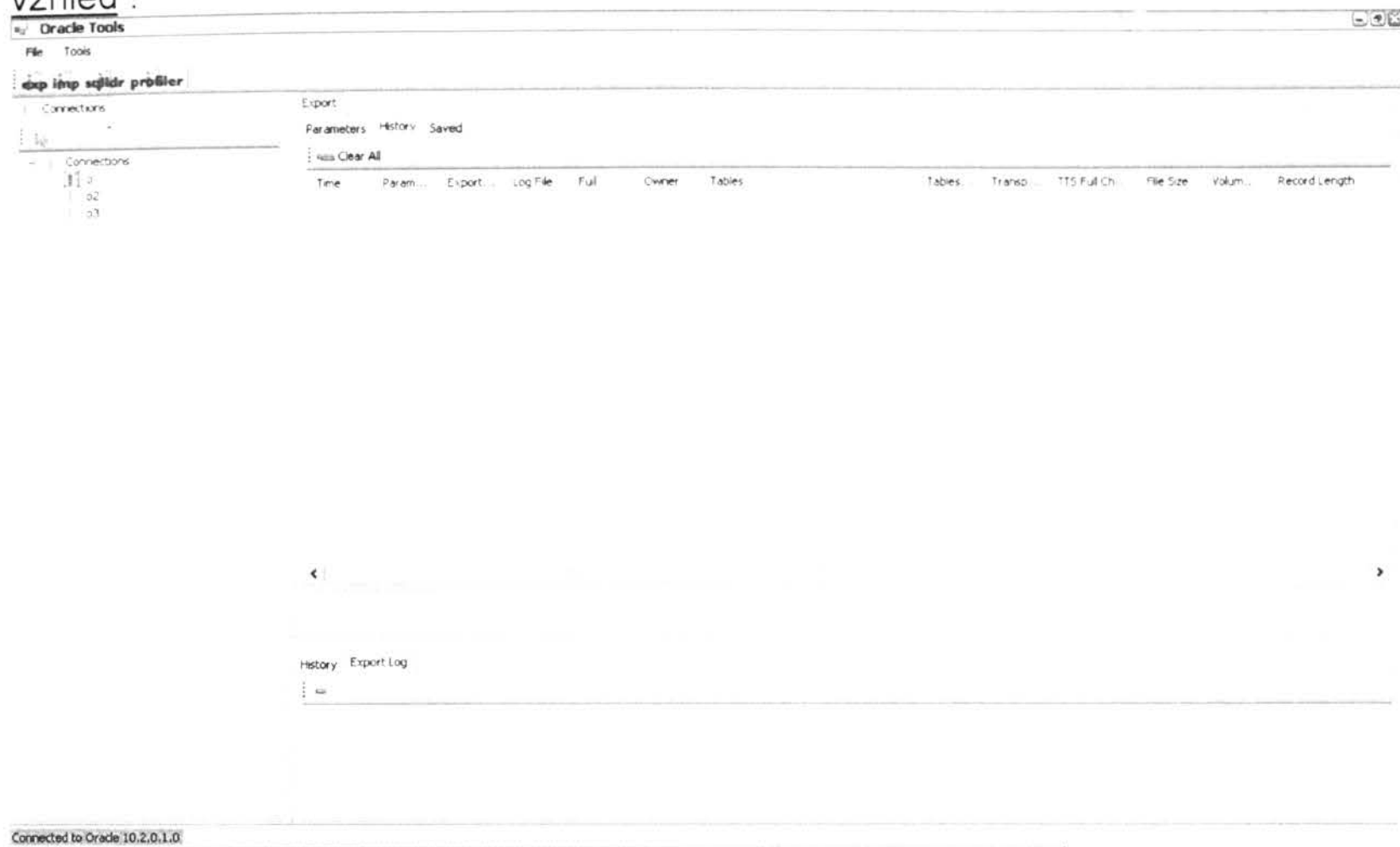
O průběhu Exportu jsme informováni v informačním okně se záložkou Export Log. Po provedení se nám zobrazí okno „SUMMARY“, jehož obsah můžeme uložit do řídicího souboru programové aplikace jehož obsah je vidět na záložce „Saved“ odkud jej můžeme znovu použít příkazem Restore

Ovládací prvky:

Uvnitř okna obsahuje řadu ovládacích prvků spojených se zadáváním hodnot parametrů, ovládání je navrženo intuitivní.

- Export - předá hodnoty do parametrického souboru a po zobrazení a schválení spustí Export
- Save - uloží zadané hodnoty parametru do souboru (jsou viditelné v záložce Saved)
- Clear All - vymaže všechny hodnoty
- Close - uzavře okno se záložkou , nic nevykoná, nic neuloží

Záložka *History*: vzhled :



Určení:

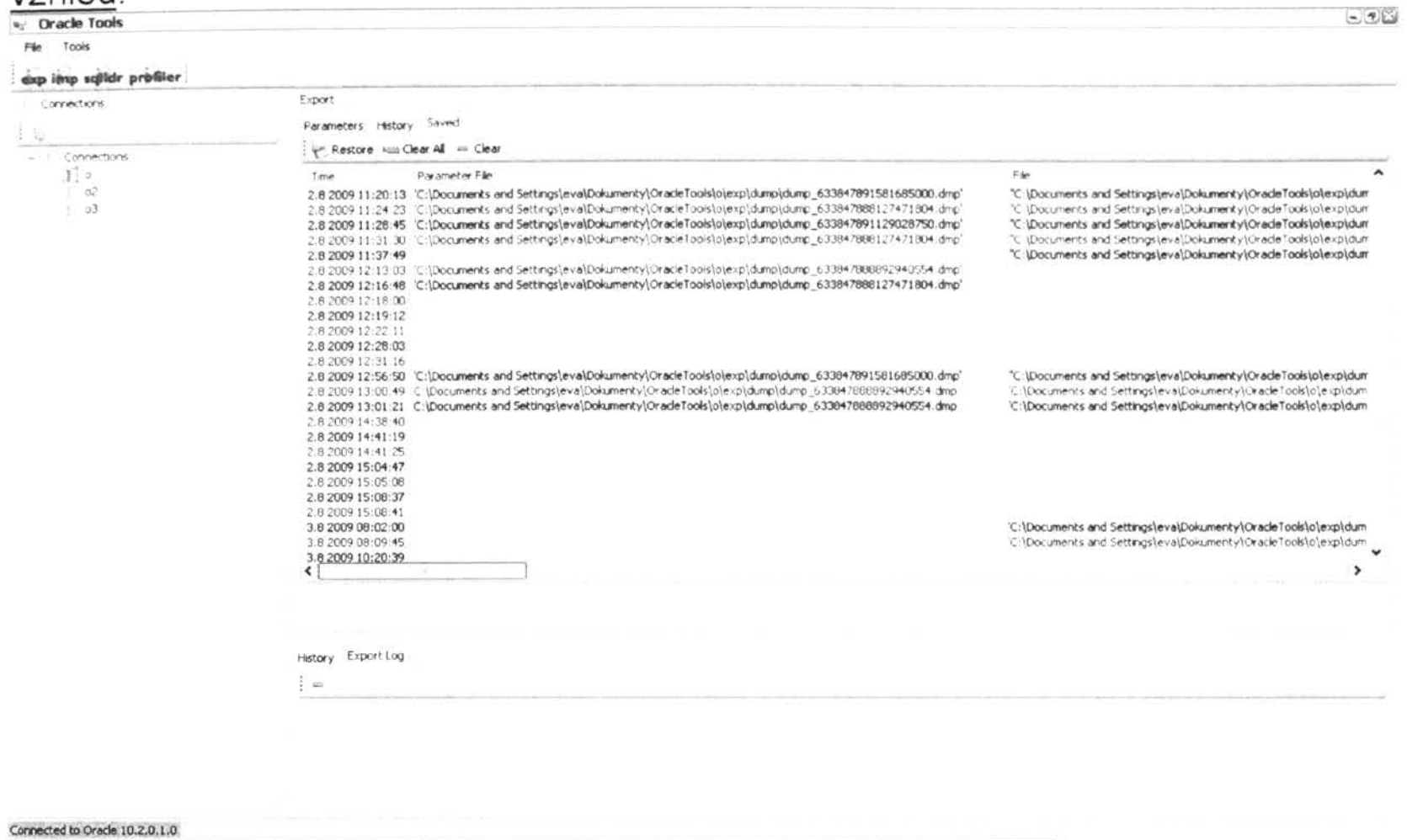
Slouží pro zobrazování historie provedených Exportů, její obsah je jen dočasný a je dostupný jen po dobu spuštění programu. Lze jej pouze prohlížet a nebo mazat. Plnění záznamů je prováděno vždy po spuštění Exportu

Ovládací prvky:

- Clear All - použitím vymažeme obsah okna historie

Záložka Saved

vzhled:



Určení:

Slouží pro prohlížení uložených konfiguračních hodnot Exportu a jejich načtení do okna záložky Parameters tlačítkem Restore pro znovu použití

Ovládací prvky:

- Restore - načte parametry do záložky Parametrs
- Clear All - vymaže obsah souboru
- Clear - vymaže vybrané řádky

Záložka History:

- zobrazuje historii vzkonávání Exportu, Importu, SQL*Loaderu- čas, nástroj a zda byla operace úspěšná či nikoliv.

Záložka ExportLog:

- zobrazuje probáhaní Exportu spolu s chybovými výstupy

Grafické rozhraní pro Import a SQL*Loader a ovládání tlačítek jsou analogické jako u Exportu, proto je nebudu dále popisovat.

Funkčnost Profiler

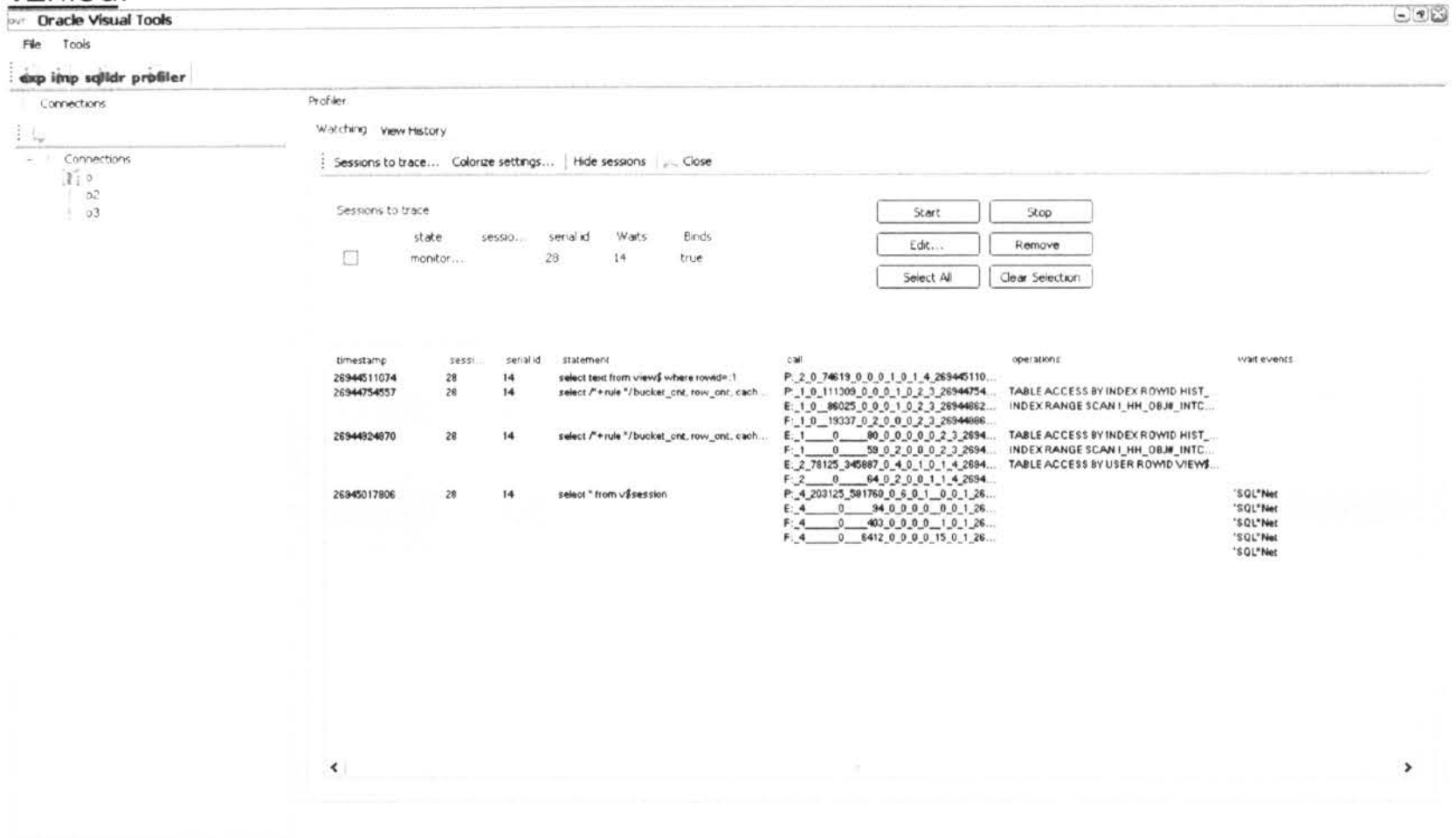
Jde v zásadě o úplně jinou funkčnost než v případě Importu, Exportu a SQL*Loaderu. Je určena pro monitorování právě prováděných dotazů SQL v databázi v závislosti na vybraných session ID. Pokud nejsme dosud připojeni do databáze, musíme se připojit a to s právy jako SYS.

V záhlaví okna jsou záložky :

- Watching- slouží pro zadávání parametrů pro sledování a jejich vyhodnocení
- ViewHistory - pro sledování historie

Záložka Watching

vzhled:



Určení:

Záložka slouží pro ovládání a sledování monitorování

Ovládací prvky :

Session to Trace- zobrazí okno pro výběr sledovaných sessions, po zadání výběru bude sessions připravena k monitorování

- Colorize Setting - pro nastavení barev prostředí
- Hide sessions - pro ovládání viditelnosti okna s seznamem sessions pro monitorování
- Close - uzavře záložku.

Fukčnost:

Nejprve se zobrazuje okno s seznamem aktivních sessions s zobrazením jejich zásadních sloupců s orientací na rozhodování zda sessions sledovat či nikoli. Výběr session provedeme v zaškrťávacím poli trace. Pomocí ovládacích tlačítek provádíme nad výběrem/výběry činnosti Refresh, Clear Selected,, Cancel a OK

V hlavním okně záložky jsou vidět sessions k monitorování, které pomocí přidružených tlačítek mohou ovládat

- Start - zapne monitorování na vybrané session
- Stop - vypne monitorování vybrané session
- Edit - u vybrané session ovládá nastavení parametrů Waits a Binds
- Remove - vymaže vybranou session z monitorování, předtím zastaví monitorování
- Select All - vybere všechny session v okně session to trace
- Clear Selection - opak Select All

Informace z monitorovaných session jsou zobrazovány v monitorovacím okně, které je plněno odspodu (starší informace v řádcích nahoře, novější informace v řádcích dole)

Zobrazené informace jsou odlévány do jednoho souboru souboru dokud nějaké monitorování běží. Takto vytvořené soubory lze možno prohlížet v záložce View History.

Ovládací prvky :

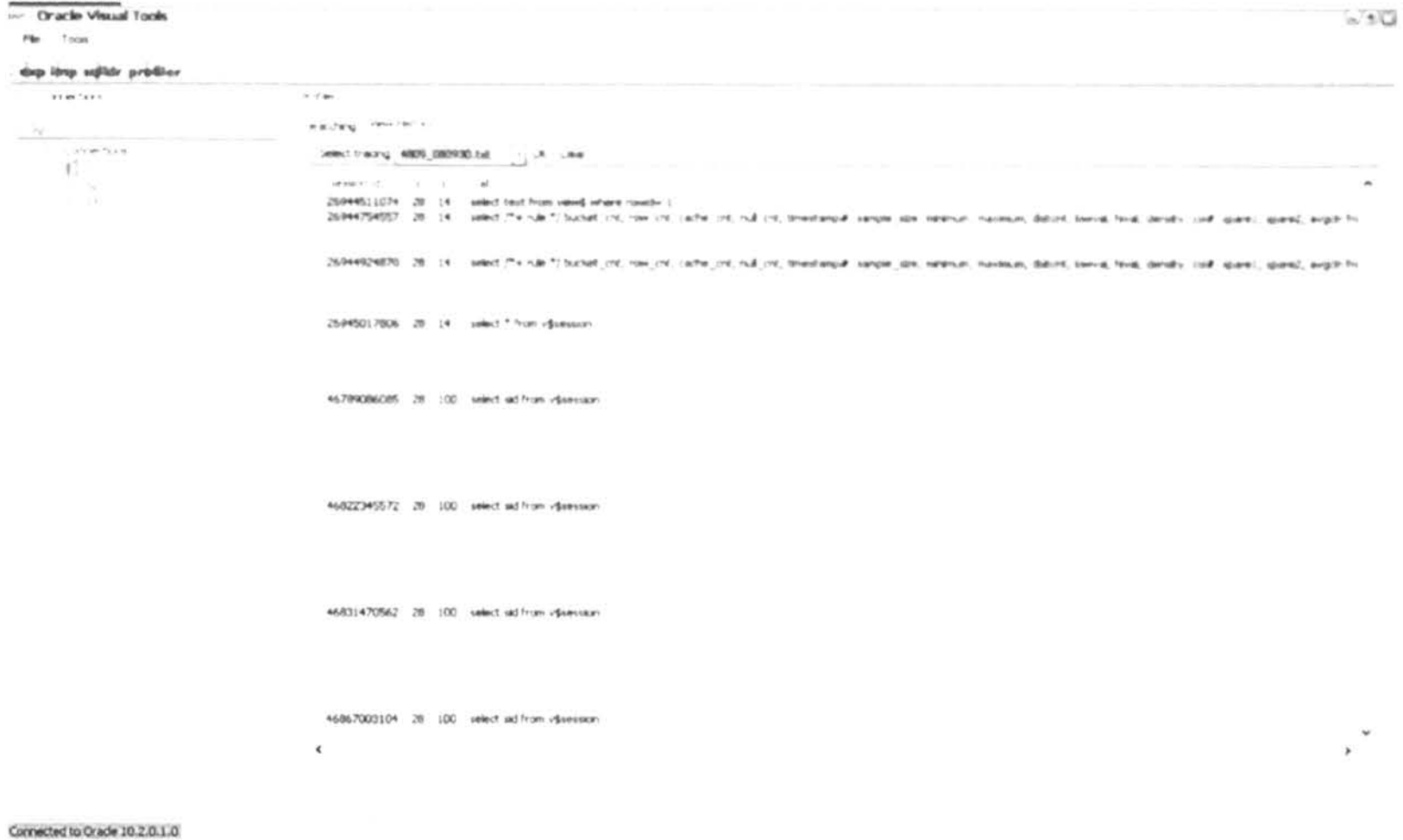
V záhlaví záložky :

- Session to Trace - zobrazí okno pro výběr sledovaných sessions, po zadání výběru bude session připravena k monitorování
- Colorize Setting pro nastavení barev prostředí

- Hide sessions - pro ovládání viditelnosti okna s seznamem sessions pro monitorování
- Close - uzavře záložku

Záložka *View History*

Vzhled



Funkčnost:

Zobrazuje statické informace uložené v monitorovacích souborech naplněných přes záložku Watching. Prostřednictvím ovládacích prvku v záhlaví okna lze vybírat soubory k zobrazení.

Ovládací prvky:

- Select tracing - umožní vybrat relevantní soubor monitorovacích informací
- OK - provede načtení dat z vybraného souboru do okna
- Clear - vymaže data z okna

8. Export a Import- popis parametrů

Tyto utility Oracle jsou původními nástroji Oracle Database ještě před nasazením, tzv. Exportních a Importních pump a jsou ve všech verzích. Originální nástroje na rozdíl od "Oracle Pump" nepodporují všechny funkční možnosti používané verze databáze.

Nástroj imp.exe se však musí použít pro Import dat provedený spustitelným programem exp.exe nebo pro přenos dat do starší verze databáze.

Originální Export a Import umožňují přenos datových objektů mezi databázemi a to dokonce mezi různými platformami a různým hardware.

Spustitelné soubory jsou součástí jak Oracle Database tak i jejího klienta. Umožňují připojit se do vzdálené databáze a tak spouštět Import a Export objektů a dat z PC uživatele po síti (pokud má příslušná oprávnění).

Způsoby spouštění a zadávání parametru pro Import a Export:

- zadáním z příkazového řádku
- použitím parametrických souborů
- zadáváním příkazu a parametru v interaktivním režimu

Pro vytvářený program byla zvolena metoda ad 2), kdy parametrický soubor je generován z údajů zadaných uživatelem v grafickém prostředí a následně je použit.

8.1 Módy Importu a Exportu:

Full:

Exportuje/Importuje celou databázi, pouze však uživatelem který má přidělenou roli pro full Export a full Import

Tablespace:

umožňuje privilegovanému uživateli přesouvat celé tabulkové prostory mezi databázemi

User:

Exportuje /Importuje objekty zadaného seznamu uživatelů, pokud k tomu má uživatel oprávnění

Table:

Exportuje/importuje specifikované tabulky a oddíly. Privilegovaný uživatel může kvalifikovat tabulky specifikováním schématu, který je obsahuje.

<i>Object</i>	<i>Table Mode</i>	<i>User Mode</i>	<i>Full Database Mode</i>	<i>Tablespace Mode</i>
<i>Analyze cluster</i>	No	Yes	Yes	No
<i>Analyze tables/statistics</i>	Yes	Yes	Yes	Yes
<i>Application contexts</i>	No	No	Yes	No
<i>Auditing information</i>	Yes	Yes	Yes	No
<i>B-tree, bitmap, domain function-based indexes</i>	Yes	Yes	Yes	Yes
<i>Cluster definitions</i>	No	Yes	Yes	Yes
<i>Column and table comments</i>	Yes	Yes	Yes	Yes
<i>Database links</i>	No	Yes	Yes	No
<i>Default roles</i>	No	No	Yes	No
<i>Dimensions</i>	No	Yes	Yes	No
<i>Directory aliases</i>	No	No	Yes	No
<i>External tables (without data)</i>	Yes	Yes	Yes	No
<i>Foreign function libraries</i>	No	Yes	Yes	No
<i>Indexes owned by users other than table owner</i>	Yes (Privileged users only)	Yes	Yes	Yes
<i>Index types</i>	No	Yes	Yes	No
<i>Java resources and classes</i>	No	Yes	Yes	No
<i>Job queues</i>	No	Yes	Yes	No
<i>Nested table data</i>	Yes	Yes	Yes	Yes
<i>Object grants</i>	Yes (Only for tables and indexes)	Yes	Yes	Yes
<i>Object type definitions used by table</i>	Yes	Yes	Yes	Yes
<i>Object types</i>	No	Yes	Yes	No
<i>Operators</i>	No	Yes	Yes	No
<i>Password history</i>	No	No	Yes	No
<i>Postinstance actions and objects</i>	No	No	Yes	No
<i>Postschema procedural actions and objects</i>	No	Yes	Yes	No
<i>Posttable actions</i>	Yes	Yes	Yes	Yes

<i>Object</i>	<i>Table Mode</i>	<i>User Mode</i>	<i>Full Database Mode</i>	<i>Tablespace Mode</i>
<i>Posttable procedural actions and objects</i>	Yes	Yes	Yes	Yes
<i>Preschema procedural objects and actions</i>	No	Yes	Yes	No
<i>Pretable actions</i>	Yes	Yes	Yes	Yes
<i>Pretable procedural actions</i>	Yes	Yes	Yes	Yes
<i>Private synonyms</i>	No	Yes	Yes	No
<i>Procedural objects</i>	No	Yes	Yes	No
<i>Profiles</i>	No	No	Yes	No
<i>Public synonyms</i>	No	No	Yes	No
<i>Referential integrity constraints</i>	Yes	Yes	Yes	No
<i>Refresh groups</i>	No	Yes	Yes	No
<i>Resource costs</i>	No	No	Yes	No
<i>Role grants</i>	No	No	Yes	No
<i>Roles</i>	No	No	Yes	No
<i>Rollback segment definitions</i>	No	No	Yes	No
<i>Security policies for table</i>	Yes	Yes	Yes	Yes
<i>Sequence numbers</i>	No	Yes	Yes	No
<i>Snapshot logs</i>	No	Yes	Yes	No
<i>Snapshots and materialized views</i>	No	Yes	Yes	No
<i>System privilege grants</i>	No	No	Yes	No
<i>Table constraints (primary, unique, check)</i>	Yes	Yes	Yes	Yes
<i>Table data</i>	Yes	Yes	Yes	Yes
<i>Table definitions</i>	Yes	Yes	Yes	Yes
<i>Tablespace definitions</i>	No	No	Yes	No
<i>Tablespace quotas</i>	No	No	Yes	No
<i>Triggers</i>	Yes	Yes	Yes	Yes
<i>Triggers owned by other users</i>	Yes (Privileged users only)	No	No	No
<i>User definitions</i>	No	No	Yes	No
<i>User proxies</i>	No	No	Yes	No
<i>User views</i>	No	Yes	Yes	No
<i>User-stored procedures,</i>	No	Yes	Yes	No

<i>Object</i>	<i>Table Mode</i>	<i>User Mode</i>	<i>Full Database Mode</i>	<i>Tablespace Mode</i>
<i>packages and functions</i>				

Zdroj Oracle® Database Utilities 10g Release 2 (10.2) B14215-01 June 2005

8.2 Společné parametry pro Export a Import

BUFFER

Default: závislé na operačním systému

Specifikuje velikost buffferu který je použit pro přenos řádku (fetch) a to v bytech. Výsledkem je, že určuje maximální počet řádku v poli přenesených edxportem najednou.

Výpočet:

$buffer_size = rows_in_array * maximum_row_size$

Pokud je uvedena nula, Export přenesse najednou jen jeden řádek.

Tabulky s sloupci typu LOBs, LONG, BFILE, REF, ROWID, LOGICAL ROWID, or DATE jsou přenášeny po jednom řádku.

Poznámka. Používá se jen pro konvenční cestu přenosu Exportem, nemá vliv na přímou cestu Exportu

CONSTRAINTS

Default: y

Specifikuje zda jsou či nejsou exportem přenášeny omezení (constrains)

FEEDBACK

Default: 0 (zero)

Specifikuje periodu ve formě počtu řádků po kterých se má zobrazit stav zpracování.

FILE

Default: expdat.dmp

Specifikuje jména dump souborů do kterých je prováděn výstup exportu. Přípona dmp je nepovinná, můžete použít libovolnou. Export podporuje vícesouborový výstup, proto můžete použít více jmen souborů, oddělených čárkou. Názvy budou pak sekvenčně použity pro případ, že soubor dosáhne maximální povolené velikosti. Pokud bude seznam v průběhu exportu vyčerpán, budete požádáni o zadání názvu souboru.

FILESIZE

Default: Data jsou zapisována do jednoho souboru, jehož velikost je viz tabulka 19-5

Table 19-5 Maximální velikost dump souboru

Operační systém	Release Oracle Database	Maximální velikost
Any	Prior to 8.1.5	2 gigabytes
32-bit	8.1.5	2 gigabytes
64-bit	8.1.5 and later	Unlimited
32-bit with 32-bit files	Any	2 gigabytes
32-bit with 64-bit files	8.1.6 and later	Unlimited

Zdroj : Oracle® Database Utilities 10g Release 2 (10.2) B14215-01 June 2005

Export podporuje zápis do vícenásobných souborů a Import je umí číst. Pokud specifikujete v bytech maximální velikost výstupního souboru, Export bude zapisovat soubory o této maximální velikosti. Jména exportních souborů mohou být zadána v parametru FILE, pokud ne a nebo zásoba jmen je vyčerpána pak jste požádáni zadání názvu souboru.

FULL

Default: n

Indikuje, že bude proveden úplný Export celé databáze. Uživatel musí mít přidělenou roli EXP_FULL_DATABASE. Tzv. full Export je dobrá cesta pro replikaci nebo pročištění databáze, musíme však dbát jistých pravidel, které jsou podrobně popsány v Oracle dokumentaci

GRANTS

Default: y

Specifikuje zda Export přenáší/nepřenáší přístupová práva (grants) spojená s objekty. Přístupová oprávnění jsou přenášena v závislosti na tom zda provádíme full mode Export nebo user mode Export. V full mode exportu jsou přenášena všechna přístupová oprávnění nad tabulkou v user mode jen přístupová práva poskytovaná vlastníkem tabulky. Systémová privilegia jsou přenášena vždy.

HELP

Default: none

Zobrazí popis parametrů Exportu pokud uvedeme Help=y

INDEXES

Default: y

Specifikuje , zda budou exportovány indexy

LOG

Default: none

Specifikuje název souboru do kterého bude zasilány informační a chybové zprávy

PARFILE

Default: none

Specifikuje jméno parametrického souboru obsahující seznam parametrů použitých pro Export

RECORDLENGTH

Default: závislé na operačním systému

Obsahuje délku záznamu v bytech. Parametr je nutný pro přenos výstupního souboru do jiného operačního systému používající jinou standardní hodnotu délky záznamu. Není-li hodnota zadána, použije se velikost dle použitého operačního systému.

RESUMABLE

Default: n

Parametr je použit pro aktivování/deaktivování opětné (RESUM) alokaci prostoru. Jeho nastavení souvisí s parametry *RESUMABLE_NAME* and *RESUMABLE_TIMEOUT*. Využívá se pro pozastavení exportu z důvodu nedostatku prostoru

RESUMABLE_NAME

Default: 'User USERNAME (USERID), Session SESSIONID, Instance INSTANCEID'

Hodnota tohoto parametru indikuje, který statement bude opětně alokovatelný.

RESUMABLE_TIMEOUT

Default: 7200 seconds (2 hours)

Hodnota parametru udává čas v sekundách, během kterých musí být chyba nedostatku prostoru vyřešena. Pokud není problém po uplynutí doby vyřešen, Export statementu je přerušeno

ROWS

Default: y

Určuje zda budou/nebudou exportovány data z řádků tabulek

TABLESPACES

Default: none

Parametr definuje, že budou exportovány všechny tabulky ze seznamu specifikovaných tabulkových prostorů včetně tabulek mající tam některý oddíl. Uživatel provádějící tento Export musí mít přidělenou roli EXP_FULL_DATABASE. Je-li parametr užit ve spojení s parametrem TRANSPORT_TABLESPACE=y můžete exportovat omezený počet tabulkových prostorů.

USERID (username/password)

Default: none

Specifikuje *username/password* (a volitelně *connect_string*) uživatele provádějícího Export. Podrobnější informace v kapitole o připojení k databázi.

VOLSIZE

Default: none

Definuje maximální počet bytu na každé výstupní pásce. Maximální hodnota je maximum, které může být uloženo do 64 bitů vašeho systému

8.3 Parametry pro Export

COMPRESS

Default: y

Specifikuje jak Export řídí alokaci inicializačních extentů dat tabulek. Jestliže specifikujete default COMPRESS=y, pak způsob inicializace extentu je řízen importem. Pokud specifikuje COMPRESS=n, pak se zapisuje do exportu aktuální hodnoty parametrů extentů exportovaných tabulek

CONSISTENT

Default: y

Specifikuje zda Export používá klausuli SET TRANSACTION READ ONLY pro zajištění konzistence dat v čase a nedovolí jejich změnu během exportu. Parametr by se měl zadávat v případě že víme že během exportu může ke změnám docházet. Použitím consistent=n je sice tabulka přenášena v jedné transakci ale vnořené a související tabulky v jiné samostatné transakci. Pokud je tabulka rozdělena, pak každá část je exportována v samostatné transakci.

DIRECT

Default: n

Specifikuje zda je použita přímá cesta exportu. Použitím `direct=y` zajistíme, že data jsou exportována přímo přeskočením vrstvy SQL command-processing. Tato metoda může být mnohem rychlejší než Export konvenční cestou.

FLASHBACK_SCN

Default: none

Specifikuje tzv. system change number (SCN). Exportu pro tzv. FlashBack Export je proveden konzistentně vzhledem k tomuto SCN.

FLASHBACK_TIME

Default: none

Umožňuje specifikovat tzv. časové razítko (timestamp). Export vyhledá SCN, které nejvíce vyhovuje zadanému časovému razítku a Export je proveden konzistentně k tomuto SCN.

OBJECT_CONSISTENT

Default: n

Specifikuje zda Export utilita používá/nepoužívá `SET TRANSACTION READ ONLY` instrukci. Jestliže je nastaven `object_consistent = y`, každý objekt je exportován v konzistentním stavu ve své vlastní read-only transakci, dokonce i tehdy pokud je segmentovaný. Naopak jestliže použijete parametr `CONSISTENT`, pak existuje jediná read-only transakce.

OWNER

Default: none

Označuje že Export probíhá v user-mode a exportuje objekty dle seznamu uživatelů.

Pokud je inicializátorem exportu databázový administrátor (role DBA), může se exportovat více uživatelů.

QUERY

Default: none

Umožňuje vybrat sadu záznamů z tabulek, které jsou exportovány v table-modu. Hodnota parametru je řetězec, který obsahuje SQL klausuli `WHERE`, jenž bude aplikována na všechny exportované tabulky. Pokud exportovaná tabulka

neobsahuje sloupec z where je signalizovaná chyba a řádky z tabulky nebudou exportovány

Při palikování parametru Query je dodržet jistá omezení uvedená v Oracle dokumentaci

STATISTICS

Default: ESTIMATE

Definuje typ databázového optimizéru statistik pokud budou exportovaná data importována. Hodnoty mohou být ESTIMATE, COMPUTE, NONE Export bude ukládat předkalkulované statistiky do exportovaného souboru

TABLES

Default: none

Určuje , že Export probíhá v table-mode dle zadaného seznamu názvu tabulek. V seznamu můžeme použít SCHEMANAME, TABLENAME, PARTITION_NAME

TRANSPORT_TABLESPACE

Default: n

Jeli nastaven na y, umožňuje Export metadat tzv. přenositelných tabulových prostorů

TRIGGERS

Default: y

Určuje zda budou/nebudou exportovány databázové triggry.

TTS_FULL_CHECK

Default: n

Je-li parametr nastaven na y, Export prověřuje zda exportovaná sada tabulkových prostorů nemá závislé objekty mimo exportovanou sadu

8.4 Parametry jen pro Import

COMMIT

Default: n

Specifikuje zda by Import měl provést transakci potvrzení zápisu po každém vložení pole dat. Implicitně se provede transakce potvrzení zápisu po zavedení tabulky a Import provede rollback transakci při chybě než bude pokračovat v zavádění dalšího objektu. Pokud je tabulka rozdělena nebo obsahuje vnořené tabulky, pak se provede vložení těchto dat v samostatné transakci. Pokud nastavíme parametr na n, provede se commit transakce při každém řádku, čímž omezíme zatěžování a nadměrnému rustu rollback segmentu. Při zavádění tabulek obsahující sloupce typu LOBs, LONG, BFILE, REF, ROWID, or UROWID columns, vkládání pole se neprovádí.

COMPILE

Default: y

Specifikuje se mohou provádět/neprovádět kompilace packages, procedur a funkcí po jejich vytvoření. Pokud je parametr nastaven na n, provede se kompilace až při jejich prvním spuštění.

DATAFILES

Default: none

Pokud je parametr *TRANSPORT_TABLESPACE* nastaven na y, parametr udává seznam datových souborů přenášených do databáze.

DESTROY

Default: n

Specifikuje zda ano/či ne by se měly znovu použít datové soubory existující v databázi. Pokud je parametr nastaven na y, Import vkládá REUSE volbu na datové soubory při vytváření tabulkových prostorů (*CREATE TABLESPACE*), čímž bude zajištěno jejich opětovné použití po předchozím vymazání jejich obsahu. Při default nastavení parametru na n bude inicializován chybový stav pokud existuje datový soubor použitý při vytváření tabulkového prostoru. Pokud potřebujete importovat do existující databáze bez nahrazení existujících datových souborů, musíte nastavit parametr *IGNORE=y*.

FROMUSER

Default: none

Obsahuje seznam schémat pro Import s nimi spojených objektů. parametr je použitelný pouze pro uživatele s *IMP_FULL_DATABASE* rolí. Pokud role není přidělena pak se použije implicitně Import v uživatelském režimu a importují se objekty importujícího uživatele. Parametr se obvykle používá ve spojení s parametrem *TOUSER* kterým zadáme seznam uživatelských jmen do jejichž schémat se má Import provést. Uživatelé z tohoto seznamu již musí v cílové databázi existovat, jinak je vrácena chyba.

IGNORE

Default: n

Specifikuje, jak mají být ošetřeny chyby při vytváření objektu. Pokud použijeme implicitní parametr n, Import loguje nebo zobrazuje chyby a objekt s takto signalizovanou chybou se nenaimportuje. Pokud je parametr nastaven na y, chyby se přehlíží a nesignalizují a Import objektu se provede.

INDEXFILE

Default: none

Parametr specifikuje název souboru do kterého budou ukládány SQL příkazy pro indexování, vytažené z exportního souboru. Vytvořený indexový soubor se může po importu editovat a pak ve formě SQL skriptu spustit pro vytvoření indexů. Výchozí hodnota parametru není nastavena a je použitelný pouze ve spojení s parametry *FULL=y*, *FROMUSER*, *TOUSER*, nebo *TABLES*.

SHOW

Default: n

Pokud je nastaven na y, pak SQL příkazy z exportního souboru jsou prolistovány a zobrazeny, nikoli ale objekty naimportovány. Je použitelný pouze ve spojení s parametry *FULL=y*, *FROMUSER*, *TOUSER*, nebo *TABLES*.

SKIP_UNUSABLE_INDEXES

Default: hodnota databázového parametru *SKIP_UNUSABLE_INDEXES* uvedená v inicializačním parametrickém souboru

Nastavením hodnoty parametru se přepíše nastavení z inicializačního parametrického souboru. Pokud ponese hodnotu y, vytváření indexu nad vloženým řádkem jenž je ve stavu Index Unusable state bude přeskočeno, ostatní indexy budou při vládání řádků aktualizovány. Pokud nebude parametr nastaven, vkládání řádku s indexem ve stavu Index Unusable state selže.

STATISTICS

Default: ALWAYS

Nastavuje co se děje se statistikami při importu. Může nabývat hodnot

- *ALWAYS* - importuje statistiky
- *NONE* - neimportuje ani nepřepočítává
- *SAFE* - importuje statistiku pokud není podezřelá, jinak by se měla přepočítat
- *RECALCULATE* - při importu přepočítává statistiky, neplatí pokud byl při exportu použit parametr *STATISTICS=NONE*

STREAMS_CONFIGURATION

Default: y

Specifikuje zda budou/nebudou z exportního souboru importována obecná metadata datových toků Stream

STREAMS_INSTANTIATION

Default: n

Specifikuje zda budou/nebudou z exportního souboru importována instancovaná metadata datových toků Stream

TABLES

Default: none

Specifikuje, že Import probíhá v table-mode a definuje seznam jmen tabulek a jejich oddílů. Není přednastavena hodnota. Můžete specifikovat následující hodnoty parametrů:

- *Jméno_tabulky* - pokud nespecifikujete *jméno_oddílu*, pak se importuje celá tabulka včetně všech oddílů
- *Jméno_oddílu* - pokud uživatel uvede, bude importován jen vyjmenovaný oddíl

Syntaxe: *jméno_tabulky:název_oddílu*

TOID_NOVALIDATE

Default: none

Nasavením na *y* se potlačí kontrola importního souboru na existenci typových ID (*TOID*) v aktuální databázi

TOUSER

Default: none

Určuje seznam uživatelů jejichž schemata budou příjemcem importu. Uvedením víceprvkového seznamu v *TOUSER* musíte zajistit správné spárování s uživateli v parametru *FROMUSER*. Uživatelé v seznamu *TOUSER* se mohou opakovat

TTS_OWNERS

Default: none

Je-li specifikován parametr *TRANSPORT_TABLESPACE=y* použijte parametr pro uvedení seznamu vlastníků dat v sadě přenositelných tabulkových prostorů

8.5 SQL*Loader – popis parametrů

SQL*Loader vkládá data z externích zdrojů do tabulek Oracle databáze. Je to mocný datově-analyticky orientovaný produkt s minimálním omezením na formáty dat ve vstupních datových souborech.

Jeho hlavní přednosti jsou:

- Zavádění dat přes síť. To znamená že můžete spustit SQL*Loader klient na jiném systému
- Zavádí data z vícenásobných datových souborů v průběhu jednoho běhu
- Zavádí data do více tabulek v rámci jednoho běhu
- Lze specifikovat znakové sady dat
- Data lze zavádět selektivně podle obsahu záznamu
- S daty lze před zavedením manipulovat užitím SQL funkcí
- Dokáže generovat jedinečné hodnoty vybraných sloupců
- K přístupu do souborů užívá operační systém
- Lze zavádět data z disků, pásek nebo prostřednictvím datových propojení
- Generuje sofistikované chybové zprávy usnadňující nápravu
- Zavádí libovolně složitá objektově-relační data
- Užívá druhotné datové soubory pro zavedení LOBů a kolekcí
- Užívá konvenční nebo přímou cestu pro zavedení dat. Konvenční cesta je velmi flexibilní, přímá cesta naopak může být velmi výkonná
- SQL*Loader se typicky používá ve spojení s vstupním parametrickým souborem řídícím celý průběh vkládání dat, výstup je směřovaný do databáze, chybových a logovacích souborů, potenciálně též do souborů s odhozenými daty které neprošly vstupním filtrem

Následující část popisuje parametry zadávané v režimu příkazové řádky a které je možno zadávat také do parametrického souboru (parametr *PARFILE*)

Parametry mohou být zadávány v libovolném pořadí, mohou být oddělené čárkou

ve formátu klíčových slov hodnota klíčovo parametru nemusí být naplněna
Parametry zadané na příkazovém řádku mají přednost před parametry uvedenými
v parametrickém souboru

Popis parametrů:

BAD (bad file)

Default: jméno datového souboru s příponou bad

Parametr specifikuje jméno chybového souboru vytvářeného SQL*Loaderem pro ukládání záznamu s chybami v procesu vkládání dat. Není-li uveden, použije se implicitní jméno s příponou bad spojené s prvním INFILE

BINDSIZE (maximum size)

Default: zjistí se spuštěním sqlldr bez uvedení parametru

Specifikuje maximální velikost BIND pole.

COLUMNARRAYROWS

Default: zjistí se spuštěním sqlldr bez uvedení parametru

Určuje počet řádku alokovaných pro v módu pro přímou cestu. Hodnota není počítána SQL*Loaderem, musíte ji specifikovat nebo použít implicitní

CONTROL (control file)

Default: none

Specifikuje jméno řídicího souboru SQL*Loaderu který popisuje jak mají být data zaváděna. Přípona souboru není specifikována, implicitně je .ctl. není-li jméno souboru uvedeno, SQL*Loader o něj požádá

DATA (datafile)

Default: jméno řídicího souboru s příponou .dat.

Specifikuje jméno datového souboru obsahující vkládaná data. Není-li uveden použije soubor .dat. Soubor uvedený na příkazové řádce je zaveden jako první a první soubor INFILE z parametrického souboru je ignorován. Další datové soupory z parametrického souboru jsou zpracovány.

DATE_CACHE

Default: Enabled (pro 1000 elements)., pro vypnutí datové vyrovnávací paměti uveďte 0.

DATE_CACHE specifikuje velikost datové vyrovnávací paměti v počtech vstupů. Parametr je dostupný pouze pro přímou cestu vkládání. Pokud počet vstupů přesáhne default nebo zadanou hodnotu, je vyrovnávací paměť automaticky vypnuta pro tuto tabulku.

DIRECT (data path)

Default: false

Specifikuje přímou cestu zavádění, pokud je true. false specifikuje konvenční cestu zavádění.

DISCARD (filename)

Default: jméno datového souboru s příponou .dsc

DISCARD specifikuje discard soubor (volitelně) vytvářené SQL*Loaderem pro ukládání záznamu, které nebyly vloženy nebo tabulkou odmítnuty.

DISCARDMAX (integer)

Default: ALL

DISCARDMAX specifikuje počet odmítnutých záznamů, než bude zavedení ukončeno. Pro zastavení zavedení dat na prvním odmítnutém záznamu, zadejte 1.

ERRORS (errors to allow)

Default: zjistí se spuštěním sqlldr bez uvedení parametru

ERRORS udává maximální počet chyb při vkládání. Je-li tento počet překročen, proces vkládání dat je ukončen.

EXTERNAL_TABLE

Default: NOT_USED

EXTERNAL_TABLE určuje, zda bude SQL*Loader zavádět data užitím volby external table. Má tyto možné hodnoty:

- NOT_USED - implicitní hodnota, zavedení dat bude provedeno v modu konvenční nebo přímou cestou.
- GENERATE_ONLY – umístí všechny SQL příkazy potřebné pro zavedení dat přes externí tabulky do log souboru. Tyto záznamy mohou být následně upraveny. Zavedení dat pak může být provedeno později bez použití SQL*Loaderu spuštěním vygenerovaného sql skriptu v SQL*Plus.
- EXECUTE – povolí provedení SQL příkazů potřebných pro zavedení dat použitím externích tabulek.. Při chybě ve vykonání SQL příkazu bude zavedení dat zastaveno.

FILE (file to load into)

Default: none

FILE specifikuje soubory pro databázové extenty. Ma význam jen při paralelním zavádění dat.

LOAD (records to load)

Default: All records are loaded

LOAD specifikuje maximální povolený počet logických záznamů k zavedení

LOG (log file)

Default: jméno řídicího souboru s příponou log

LOG specifikuje log soubory které t SQL*Loader bude vytvářet pro ukládání logovacích informací v průběhu zavádění dat

MULTITHREADING

Default: true u víceprocesorových systému, false na jednoprocessorovém systému. Je dostupný pouze pro přímou cestu zavedení dat.

PARALLEL (parallel load)

Default: false

PARALLEL specifikuje zda v modu přímé cesty zavedení dat může být nakládáno s více souběžnými sessions pro zavedení dat do stejné tabulky

PARFILE (parameter file)

Default: none

PARFILE specifikuje jméno souboru s společně použitelnými parametry

READSIZE (read buffer size)

Default: zjistí se spuštěním sqlldr bez uvedení parametrů

The *READSIZE* parameter je užit jen při čtení dat z datového souboru. Při čtení záznamů z řídicího souboru je použita vždy hodnota 64 KB jako *READSIZE*. Specifikuje v bytech velikost čtecího bufferu, nechcete-li použít implicitní hodnotu. Maximální hodnota je 20 megabytes (MB)

RESUMABLE

Default: false

RESUMABLE parametr je užít pro zapnutí a vypnutí resumable alokaci souborů. Pokud použijete parametry *RESUMABLE_NAME* a *RESUMABLE_TIMEOUT* musíte parametr nastavit na true

RESUMABLE_NAME

Default: 'User USERNAME (USERID), Session SESSIONID, Instance INSTANCEID'

Hodnota specifikuje řetězec s názvy statementu které budou použity s volbou resumable. Pro specifikaci statementu můžete použít pohled *USER_RESUMABLE* nebo *DBA_RESUMABLE*. Parametr je ignorován není-li použito *RESUMABLE true*

RESUMABLE_TIMEOUT

Default: 7200 seconds (2 hours)

Hodnota parametru udává čas v sekundách, během kterých musí být chyba nedostatku prostoru vyřešena. Pokud není problém po uplynutí doby vyřešen, proces zavádění dat je ukončen

Parametr je ignorován není-li použito *RESUMABLE true*

ROWS (rows per commit)

Default: zjistí se spuštěním sqlldr bez uvedení parametru

Při konvenční metodě zavádění dat specifikuje počet řádku v bind poli. Při přímé metodě zavedení dat specifikuje počet řádku které chcete číst z datového souboru před jejich uložením. Implicitně jsou čteny všechny záznamy a ukládány až do konce

SILENT (feedback mode)

V proběhu provádění SQL*Loaderu jsou zobrazovány na obrazovce různé informace a zprávy, které se ukládají také do log souborů

Parametr SILENT umožňuje potlačit zobrazované informace

Lze použít následujících hodnoty parametru::

- *HEADER* – nezobrazuje hlavičky zpráv, ne však v logu.
- *FEEDBACK* – nezobrazuje zprávy typu "commit point reached".
- *ERRORS* – nezobrazuje chybové zprávy .
- *DISCARDS* – nezobrazuje zprávy o záznamech zaznamenávaných do discard souboru (záznamy, vyřazené vstupním filtrem)
- *PARTITIONS* - Vypne zápis per-partition statistik do log souboru při přímém modu zavádění dat
- *ALL* – implementuje funkčnost hodnot: *HEADER*, *FEEDBACK*, *ERRORS*,

- *DISCARDS*, and *PARTITIONS*

SKIP (records to skip)

Default: žádný záznam není vynechán

SKIP specifikuje počet logických záznamů od začátku souboru, které nebudou zavedeny.

SKIP_INDEX_MAINTENANCE

Default: false

SKIP_INDEX_MAINTENANCE parametr zastavuje údržbu indexu pro přímý mód zavádění, ale nemá vliv na konvenční mód. Nastavuje příznak Index Unusable.

SKIP_UNUSABLE_INDEXES

Default: hodnota konfiguračního parametru Oracle databáze SKIP_UNUSABLE_INDEXES

Hodnota zadaná v SQL*Loader přepisuje hodnotu konfiguračního parametru. Hodnota true znamená, že zavádění dat pokračuje, když se objeví příznak Index Unusable na používaném indexu.

STREAMSIZE

Specifikuje velikost v bytech pro datové proudy při přímém modu zavádění.

USERID (username/password)

Default: none

Specifikuje uživatele a jeho heslo. Pokud se připojíte jako SYS, musíte uvést také AS SYSDBA.

9. Programátorská dokumentace

Cílem projektu bylo naprogramovat uživatelské rozhraní pro Export, Import a SQL*Loader a dále Profiler.

Třídy

Program se tedy skládá z následujících hlavních tříd

Hlavní třídy:

Connections

obsluhuje připojení k databázi

OracleTools_Main

základní třída, obsahující základní formulář

z této třídy se volá třída Connections pro připojení k databázi

Export

třída realizující Export databáze

Import

třída realizující Import databáze

SqlLdr

třída realizující SQL*Loader

Profiler

třída realizující monitorování a zobrazování statistik

LogHistory

třída realizující zobrazení výsledků Exportu, Importu a SQL*Loaderu

Pomocné třídy:

app_data

třída obsluhující cesty pro uložené soubory Exportu, Importu, SQL*Loaderu a Profileru

Current

třída s původním záměrem měla držet aktuálním proměnné jako je aktuální připojení do databáze, také proměnné pro Export, Import, Sqlldr, Profiler, ty byly přesunuty do ToolsCurrent

obsahuje Controls formulářů, ke kterým je potřeba mít přístup, jako jsou TabControl, TabPages, Textboxy

proměnné příznaků, např. zda neprobíhá uzavírání některé z TabPage, podrobněji v kódu

třída na získání aktuálního času ve formátu dd.mm rrrr hh:mm:ss

lepší by bylo tuto třídu rozdělit na více tříd se specifickým účelem

FileClass

třída pro práci se soubory

TextClass

třída pro práci s textem

OpenedCommandClass

třída obsahující otevřené nástroje a umožňující přepínání mezi nimi

OpenedToolsSavedAndHistoryControls

třída obsluhující Saved a History ListView jednotlivých nástrojů

Queries

třída pro získání správných SQL dotazů

Třídy obsahující Controls jednotlivých nástrojů a pomocných proměnných:

ToolsCurrent

třída obsahující Controls společné pro Export, Import a SQLLoader jako je Saved a History ListView pro jednotlivé nástroje

Saved ListView

obsahuje uložené parametry Exportu, Importu nebo SQL*Loaderu

History ListView

obsahuje vykonané parametry Exportu, Importu, SQL*Loaderu od spuštění aplikace

ProfilerCurrent

třída obsahující Controls specifické pro Profiler, seznamy sessions podle toho, v jaké jsou fázi vykonávání, vzory pro zvýraznění při monitorování, statistiky vybrané uživatelem (výběr statistik zatím není naimplementován)

Chybové třídy:

ErrorCodes

základní chybové kódy definované pro lepší ladění

Exceptions

definované typy výjimek: základní je BaseException, která dědí od ApplicationException

od BaseException dědí:

- DebugException- výjimka pro ladění
- FileException- výjimka obsluhující práci se soubory
- DbException- výjimka obsluhující databázi
- UserException- výjimka obsluhující chyby na straně uživatele

ShowMessages

třída pro zobrazování chybových hlášení, jak způsobených uživatelem, tak i chybou v kódu

Třídy s konstantami:

Constants

třída pro konstantní hodnoty

DefinedColors

třída s definovanými barvami

Formuláře:

OracleTools_main

hlavní formulář

Export

formulář pro Export

Import

formulář pro Import

SqlLoader

formulář pro SQL*Loader

Profiler

formulář pro Profiler

ConnectionForm

formulář pro připojení k databázi

LogHistory

formulář pro zobrazování logů a historie Exportu, Importu, SQLLoaderu

AreYouSure

formulář dotazující se uživatele, zda si je jistý, že chce danou akci provést

EditFile

formulář umožňující uživateli editaci před vykonáním Exportu, Importu, SQL*Loaderu

MatchingTablesForm

formulář umožňující pattern matching tabulek

SaveForm

formulář zobrazující parametry po vykonání Exportu, Importu nebo SQL*Loaderu s možností uložení do souboru

SessionsToTraceForm

formulář zobrazující aktuální sessions

SessionEditForm

formulář umožňující nastavit wait a binds jednotlivých sessions

ColorizeSettings

formulář pro Profiler- pro zvýrazňování zadaného textu při monitorování

StatisticsForm

formulář připravený pro výběr statistik, které chce uživatel sledovat

10. Závěr:

Zadaný úkol je širokého rozsahu a pokrývající dvě nesourodé oblasti: oblast uživatelské rozhraní nad příkazovými řádky pro zavádění a vyvádění dat z databáze a sledování sql příkazů.

První část je úzce vymezena seznamem parametrů a šlo jen o návrh grafického prostředí a jak vhodně tyto parametry předat k vykonání, případně umožnit uživateli zadané parametry uložit pro případné pozdější použití.

Druhá část pro její dobré zvládnutí předpokládá poměrně rozsáhlé znalosti vnitřního fungování databáze Oracle a umožňují velmi široký rozsah možností řešení. Ladění je nikdy nekončící práce a programy tohoto typu mohou být stále zdokonalovány. Program Oracle Tools může být v budoucnu rozšířen o filtrování statistik a v průběhu monitorování může přímo z databáze získávat další statistické údaje.

11. Seznam literatury:

- [1] Loney K., Bryla B.(2006): Mistrovství v Oracle 10g, Computer Press
- [2] Oracle (2005):Oracle® DatabaseUtilities 10g Release 2 (10.2) B14215-01 June 2005
- [3] Oracle (2007):Oracle® DatabaseUtilities11g Release 1 (11.1)B28319-02September 2007
- [4] Oracle (2008):Oracle® DatabasePerformance Tuning Guide11g Release 1 (11.1)B28274-02 July 2008