SUNCOAST ORACLE USERS' GROUP

How to get/put Oracle dump files?

Max Satula, SCC DBA

Disclaimer

- All information is offered in good faith and in the hope that it may be of use, but is not guaranteed to be correct, up to date or suitable for any particular purpose.
- All licensing questions regarding use of Oracle products should be directed to Oracle representatives

Myself

- IT experience since 1999, Oracle since 2001
- Oracle DB since 8.0.5
- OCP 10g/11g
- Besides DBA, used to program in C/C++, ASP, C#, PL/SQL …
- Worked for private software companies, State Treasury of Ukraine

Agenda

- The problem
- Solutions
- The tool

The problem

- Deployment
  - database model designed
  - saved to dump
  - the dump imported to environments
- Pull (partial) data for investigation
- What is the right tool for that?

Solutions

- Database backup
- DBCA template
- Classic exp/imp
- Data Pump
- SQL*Loader
- SQL*Plus
- 3rd party tools
- Homebrewed software

SQL*Loader and SQL*Plus

- Two options
  - Use SQL Plus only, store DDLs and INSERT statements

```
CREATE TABLE t (n NUMBER, t1 VARCHAR2(100));

INSERT INTO t (n, t1) VALUES (1, 'Hello');
INSERT INTO t (n, t1) VALUES (2, 'world');
COMMIT;
```

  - Use both tools

```
CREATE TABLE t (n NUMBER, t1 VARCHAR2(100));
```

```
1, "Hello"
2, "World"
```

```
LOAD DATA INFILE 'myfile.dat' APPEND INTO TABLE t
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
```

SQL*Loader and SQL*Plus

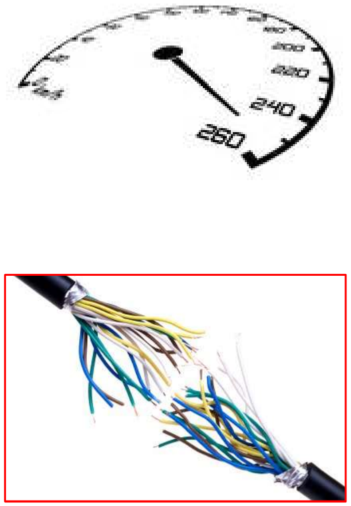
- Well controllable in VCS
- Difficult to manage continuous changes
  - change in both file and database
  - change in file, reload database
  - change in database, regenerate file
- Need to cook source data
  - remember to escape special characters

exp/imp and Data Pump

- Works with all types of objects out-of-box*
- Binary files, non-traceable in VCS
- Changes made directly to the database, then dump recreated

Data Pump

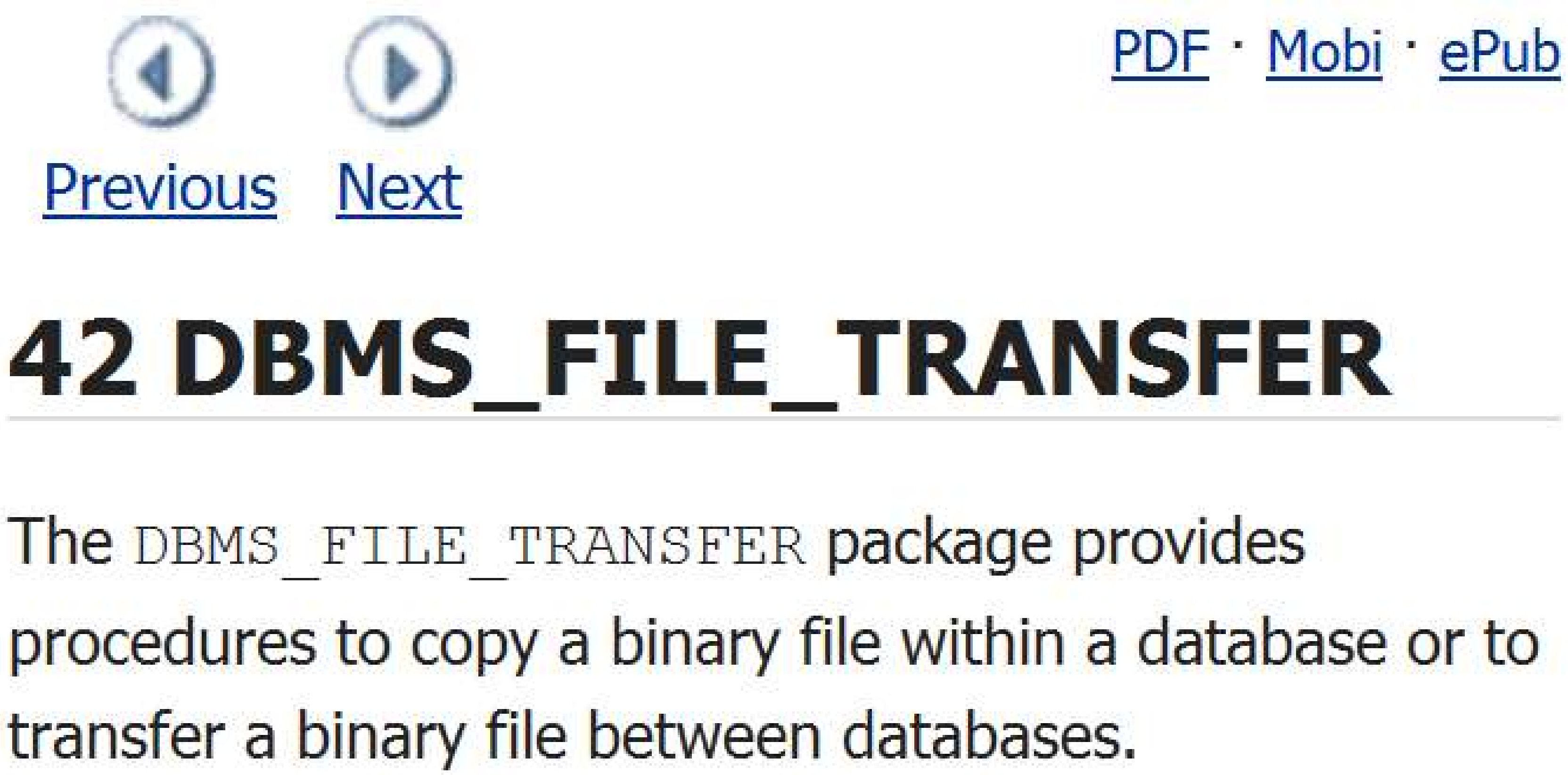
- Stores dump files on server side
- Can run in background
- Advantages
  - Performance
  - Interruption safe
  - Is the only way supported in 11g and up



Data Pump disadvantages

- None?
  - … unless …
    - no physical access to an Oracle directory

DBMS_FILE_TRANSFER



42 DBMS_FILE_TRANSFER

The `DBMS_FILE_TRANSFER` package provides procedures to copy a binary file within a database or to transfer a binary file between databases.

Option 1: Ask

- Deployment specialist asks a DBA to upload a dump file
  - have to repeat each time

Option 2: Do It Yourself

- Console access
- NFS share
- SCP (over SSH)
- FTP
- HTTP (WebDAV)
- Console copy-n-paste
- etc
- Need to ask admin once to get access (see option 1)
- … or …

Option 2a: Do It Yourself, Completely

- SQL*Net, i.e. Oracle connection
  - no additional access required
  - is that possible?

The Tool

- Get
  - Read from Oracle Directory with UTL_FILE
  - Store a portion into RAW bind variable
  - Save to a local file
- Put
  - Read from a local file
  - Pass to Oracle through RAW bind variable
  - Write to Oracle Directory with UTL_FILE
- A simple program written in C uses OCI

Introducing: OCP

- **O**racle **C**o**p**y tool
- GPL v2
- https://github.com/maxsatula/ocp
- Uses GNU autotools

```
$ autoreconf -I && make dist
```

```
$ gunzip < ocp-*.tar.gz | tar xf -
$ cd ocp-…
$ ./configure && make
```

Basic Feature

- The main purpose: copy files to/from Oracle
- No extra permissions required
  - CREATE SESSION
  - [READ|WRITE] ON DIRECTORY
  - EXECUTE ON UTL_FILE
- No extra database objects needed*

```
$ ocp /@mydb DATA_PUMP_DIR:export.dmp .
$ ocp /@mydb mymodel.dmp DATA_PUMP_DIR:seedfile.dmp
```

List Directories and Files

```
$ ocp /@mydb --list-directories
rw DATA_PUMP_DIR                  (/u01/oraexport)
```

```
$ ocp /@mydb --ls=DATA_PUMP_DIR exp\*
Contents of DATA_PUMP_DIR directory
File Name                                    Size     Last Modified
---------------------------------------- ------------ -------------------
expdat.dmp                                     471040 04/11/2015 14:41:46
export.log                                       1749 03/09/2015 17:34:23
---------------------------------------- ------------ -------------------
    2 File(s)                                  472789
```

List Files

- Additional privileges
  - dbms_java.grant_permission('<user>', 'SYS:java.io.FilePermission', '<oracle_directory_path>', 'read')
  - dbms_java.grant_permission('<user>', 'SYS:java.io.FilePermission', '<oracle_directory_path>/*', 'read')

- Supporting objects
  - 2 types (OBJECT and TABLE OF), a Java Source, a function

List Files: Supporting Objects

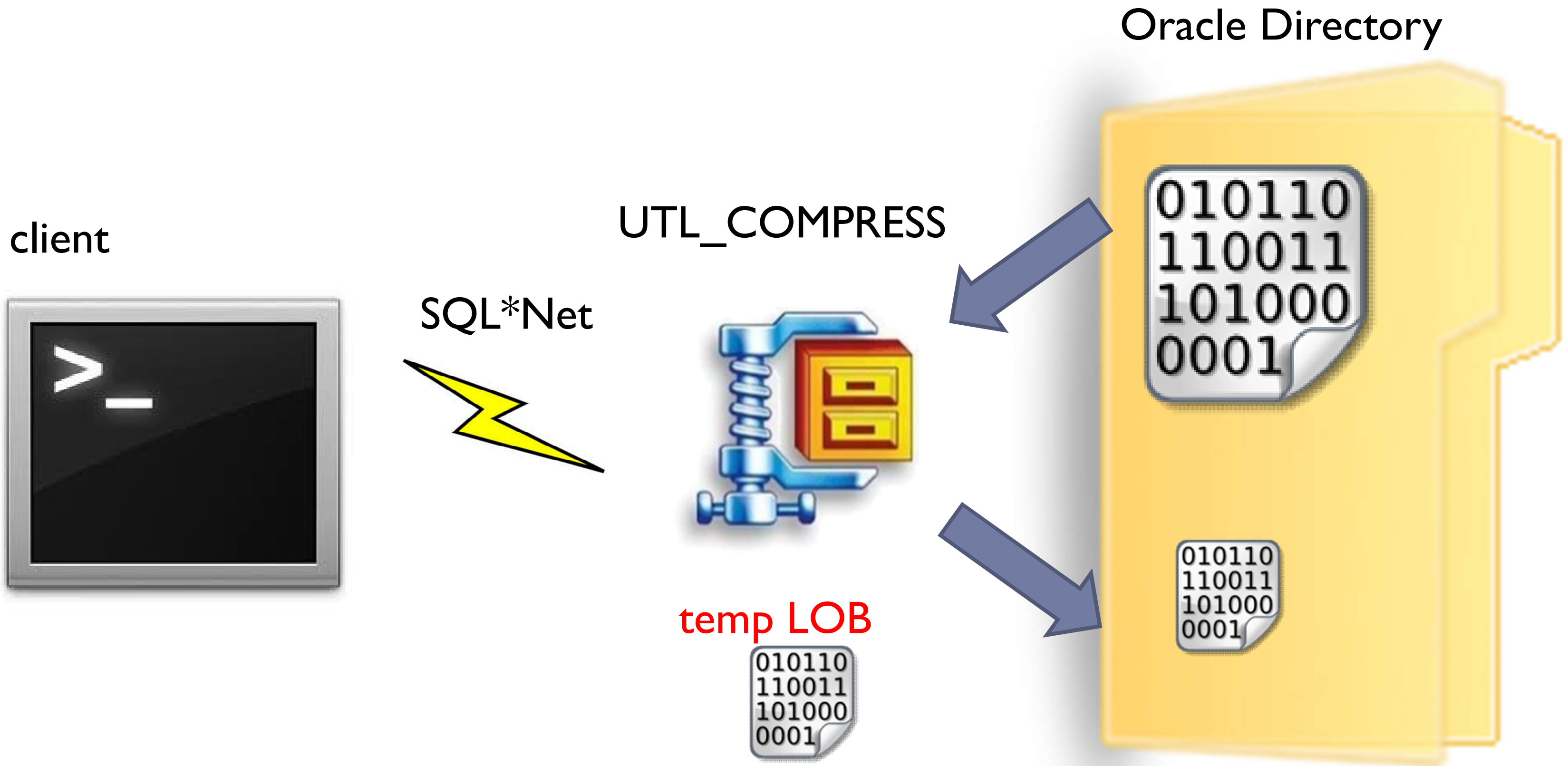
- Additional privileges
  - CREATE TYPE
  - CREATE PROCEDURE

```
$ ocp /@mydb --install
$ ocp /@mydb --deinstall
```

Remove a File

```
$ ocp /@mydb --rm DATA_PUMP_DIR:expdat.dmp
```

Server-Side (De-)Compression



```
$ ocp /@mydb --gzip DATA_PUMP_DIR:bigexport.dmp
$ ocp /@mydb --gunzip DATA_PUMP_DIR:bigexport.dmp.gz
```

Compression Options

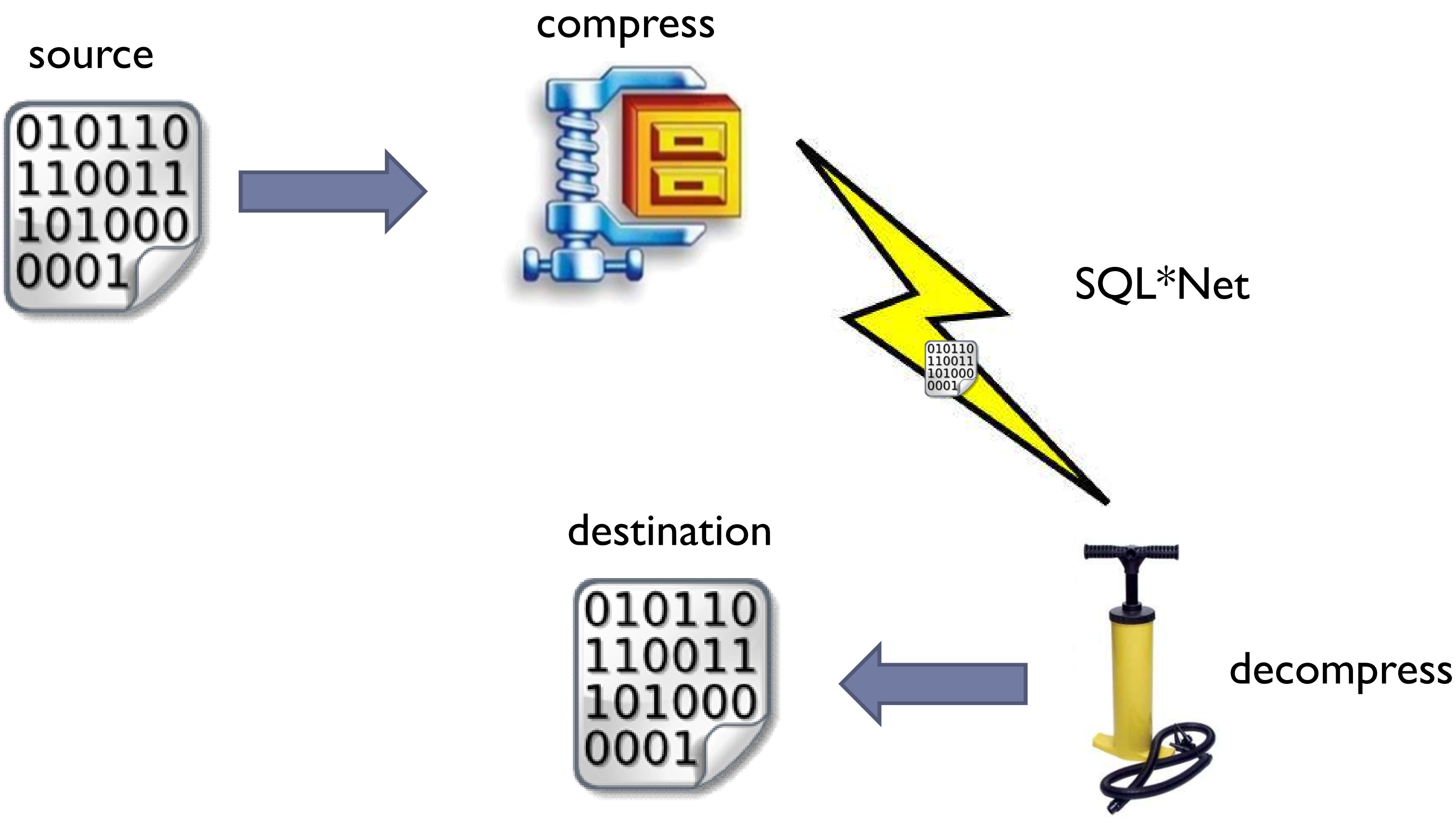
```
Compression options:
  --gzip                    Compress file in Oracle directory
  --gunzip                  Decompress file in Oracle directory
  -1, --fast                Fastest compression method
  -9, --best                Best compression method
  -b, --background          Submit an Oracle Scheduler job and
                            exit immediately
```

- Background option needs CREATE JOB

```
$ ocp /@mydb --gzip -9 -b DATA_PUMP_DIR:bigexport.dmp
Submitted a job OCP_GZIP_56388
```

- dba_scheduler_job_run_details

On-the-Fly Compression



source

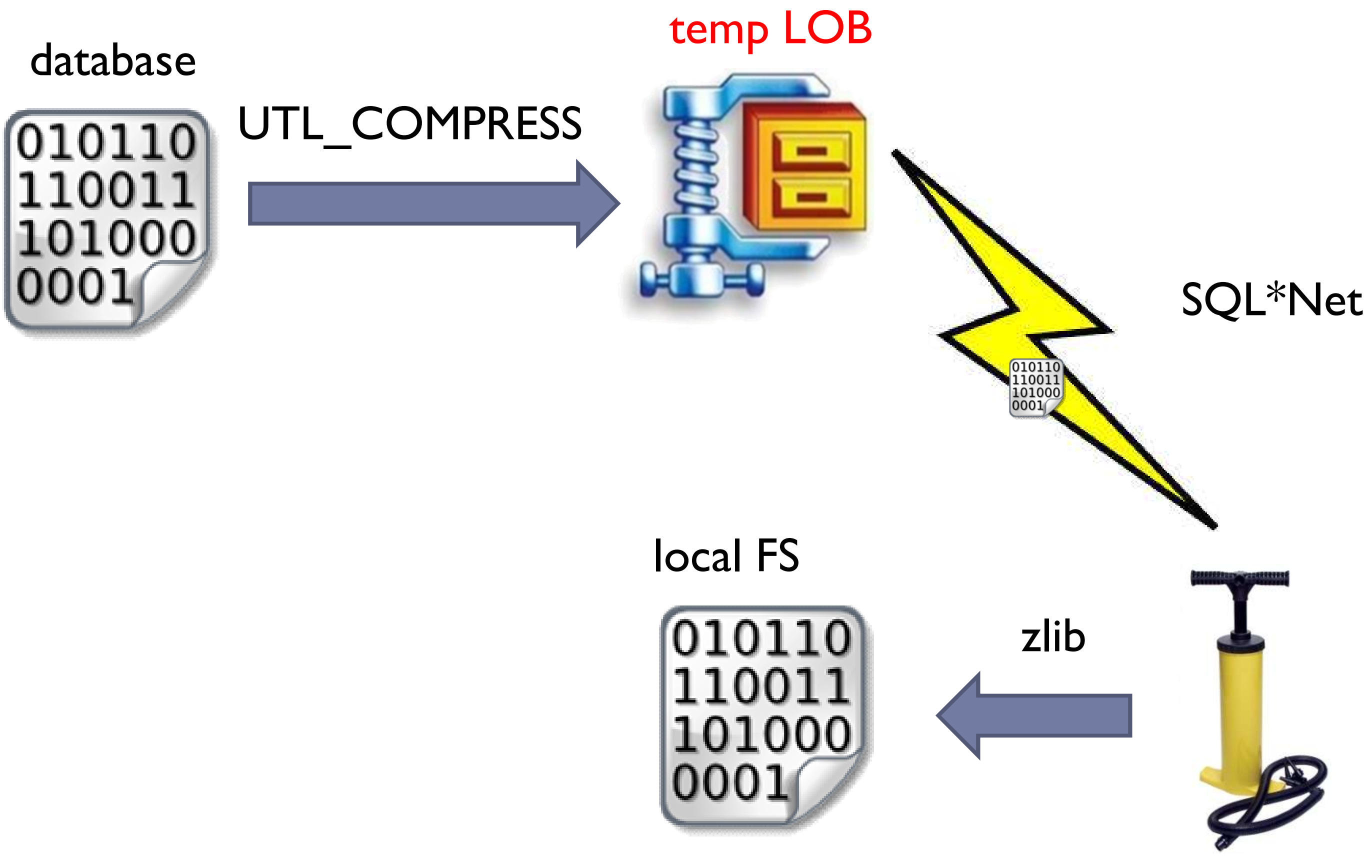
compress

SQL*Net

decompress

destination

On-the-Fly Compression



database

UTL_COMPRESS

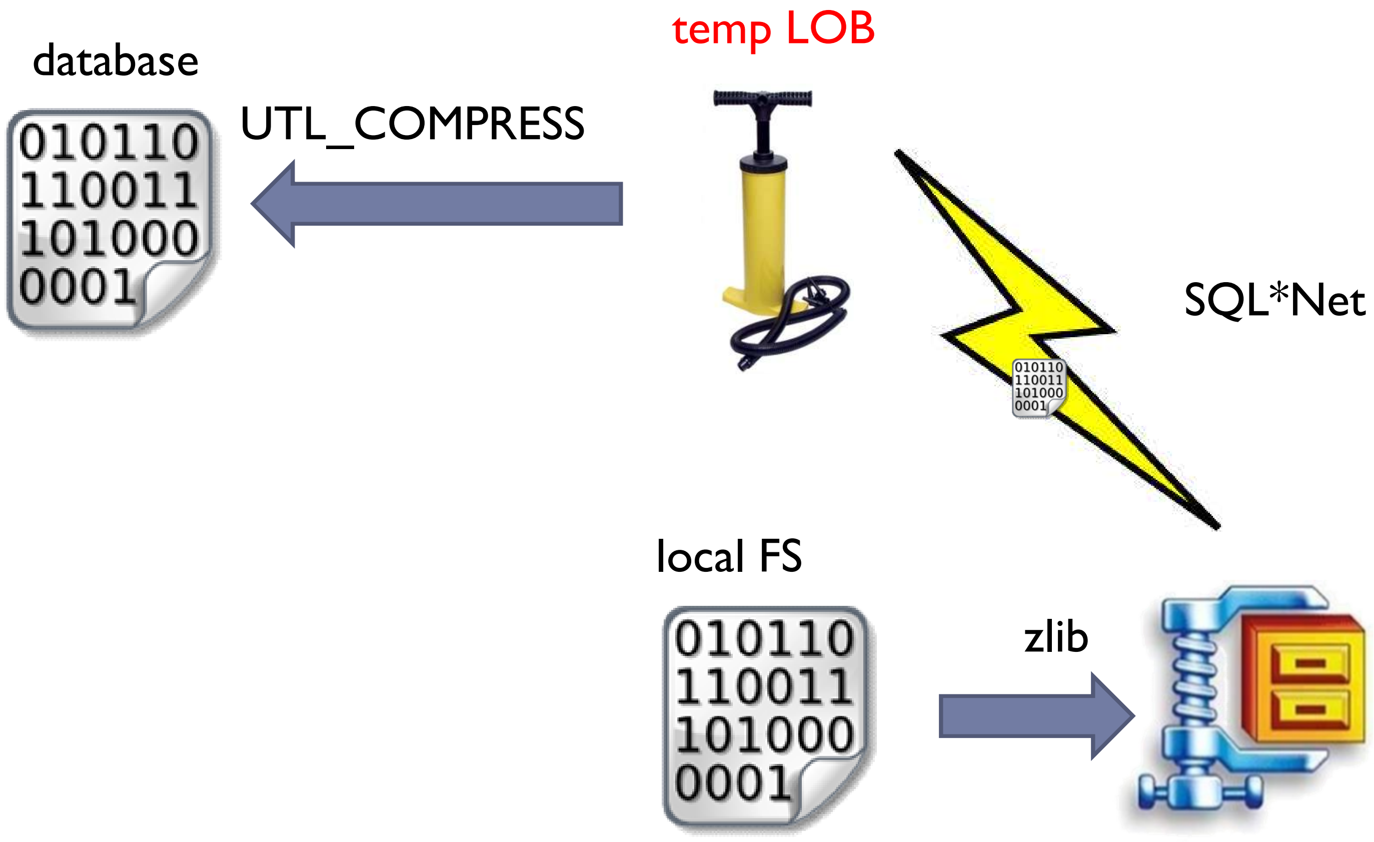
temp LOB

SQL*Net

local FS

zlib

On-the-Fly Compression



temp LOB

database

UTL_COMPRESS

SQL*Net

local FS

zlib

Compression: Summarize

- Server side (remote) (de-)compression
  - UTL_COMPRESS
    - check licensing: Advanced Compression Option?
- On-the-fly compression
  - UTL_COMPRESS and zlib
- Compression uses a temporary LOB of the size of entire compressed file
  - possible TODO: set limit on TEMP usage, or add TEMP monitoring

Transfer Options

```
Transfer options:
  -i, --interactive         prompt before overwrite (overrides
                            previous -f –c options)
  -f, --force               force overwrite an existing file
                            (overrides previous -i -c options)
  --keep-partial            if error occurred, do not delete
                            partially transferred file
  -c, --continue            resume transfer (implies
                            --keep-partial)
                            (overrides previous -f -i options)
```

Advanced Features

- Secure Password Store (wallet) support
- Fancy progress meter
  - borrowed from SCP sources
- “-” means STDIN/STDOUT
- v$session.action
- v$session_longops

Known Bugs

- A byte lost when SERVER=SHARED and reading >1 pieces (Oracle bug?)
    - workaround difficulties

Oracle Bug Testcase

- Initial condition

```
declare
  handle utl_file.file_type;
  hexnumbers varchar2(16) := '0123456789ABCDEF';
begin
  handle := utl_file.fopen('DATA_PUMP_DIR', 'allbytes.dmp', 'wb');
  for i in 1..16 loop
    for j in 1..16 loop
      utl_file.put_raw(handle, hextoraw(substr(hexnumbers,i,1)
                                      ||substr(hexnumbers,j,1)));
    end loop;
  end loop;
  utl_file.fclose(handle);
end;
/
```

Oracle Bug Testcase

- **Package**

```
create package bugtest is
  handle utl_file.file_type;
  buf_ raw(16); size_ number := 16;

  procedure open; procedure read; procedure close;
end;
/

create package body bugtest is
  procedure open is begin
    handle := utl_file.fopen('DATA_PUMP_DIR', 'allbytes.dmp', 'rb');
  end;
  procedure read is begin
    utl_file.get_raw(handle, buf_, size_);
    size_ := utl_raw.length(buf_);
  end;
  procedure close is begin utl_file.fclose(handle); end;
end;
/
```

Oracle Bug Testcase

- Single call

```
set serveroutput on
exec bugtest.open;
begin
  bugtest.read;
  dbms_output.put_line(rawtohex(bugtest.buf_));
  bugtest.read;
  dbms_output.put_line(rawtohex(bugtest.buf_));
  bugtest.read;
  dbms_output.put_line(rawtohex(bugtest.buf_));
end;
/
exec bugtest.close;
```

```
000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F
```

Oracle Bug Testcase

- Multiple calls, DEDICATED server

```
set serveroutput on
exec bugtest.open;
exec bugtest.read;
exec dbms_output.put_line(rawtohex(bugtest.buf_));
exec bugtest.read;
exec dbms_output.put_line(rawtohex(bugtest.buf_));
exec bugtest.read;
exec dbms_output.put_line(rawtohex(bugtest.buf_));
exec bugtest.close;
```

```
000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F
```

Oracle Bug Testcase

- Multiple calls, SHARED server

```
set serveroutput on
exec bugtest.open;
exec bugtest.read;
exec dbms_output.put_line(rawtohex(bugtest.buf_));
exec bugtest.read;
exec dbms_output.put_line(rawtohex(bugtest.buf_));
exec bugtest.read;
exec dbms_output.put_line(rawtohex(bugtest.buf_));
exec bugtest.close;
```

```
000102030405060708090A0B0C0D0E0F
1112131415161718191A1B1C1D1E1F20
2122232425262728292A2B2C2D2E2F30
```

Oracle 10g

- Upload works with DEDICATED server only
- SHARED gives an error:

```
ORA-29285: file write error
ORA-06512: at "SYS.UTL_FILE", line 211
ORA-06512: at "SYS.UTL_FILE", line 1184
```

  - or use on-the-fly compression

Security Lesson

- Give me a place to stand, and I will move the world
  - Archimedes

- Give me an Oracle connection, and I will screw your sensitive information

Things TODO

DEMO

Questions?