

April 24th, 2006

ORACLE 10g DATA PUMP

Inderpal S. Johal
Principal Consultant
Data Softech Inc.



AGENDA

- Oracle Data Loading/Unloading options
- Why we use Data pump or exp/imp
- Introduction to Data Pump
- Data Pump Architecture & Implementation
- Benefits of Data Pump Utility
- Data Pump vs. Exp/Imp
- DBMS_DATAPUMP package
- Questions & Answers

Oracle Data Loading/Unloading Options

- **INSERT** Sql Command
- **MERGE** Sql Command
- **Exp/Imp** utility
- **PL/SQL** API **UTL_FILE**
- **SQL*Loader**
- **External Tables**
- **PL/SQL Bulk Collect ... ForAll**
- **Data Pump**

Why Export/Import or DataPump

- For Moving Data between Database

Development ↔ QA ↔ Production

- Taking Logical Backup
- Archiving Data for future Reference
- Migrating Database
 - ✓ Operating System[MS] ↔ Operating System [Linux]
 - ✓ Oracle Version [9i] ↔ 10g
 - ✓ One Character Set ↔ Other Character Set

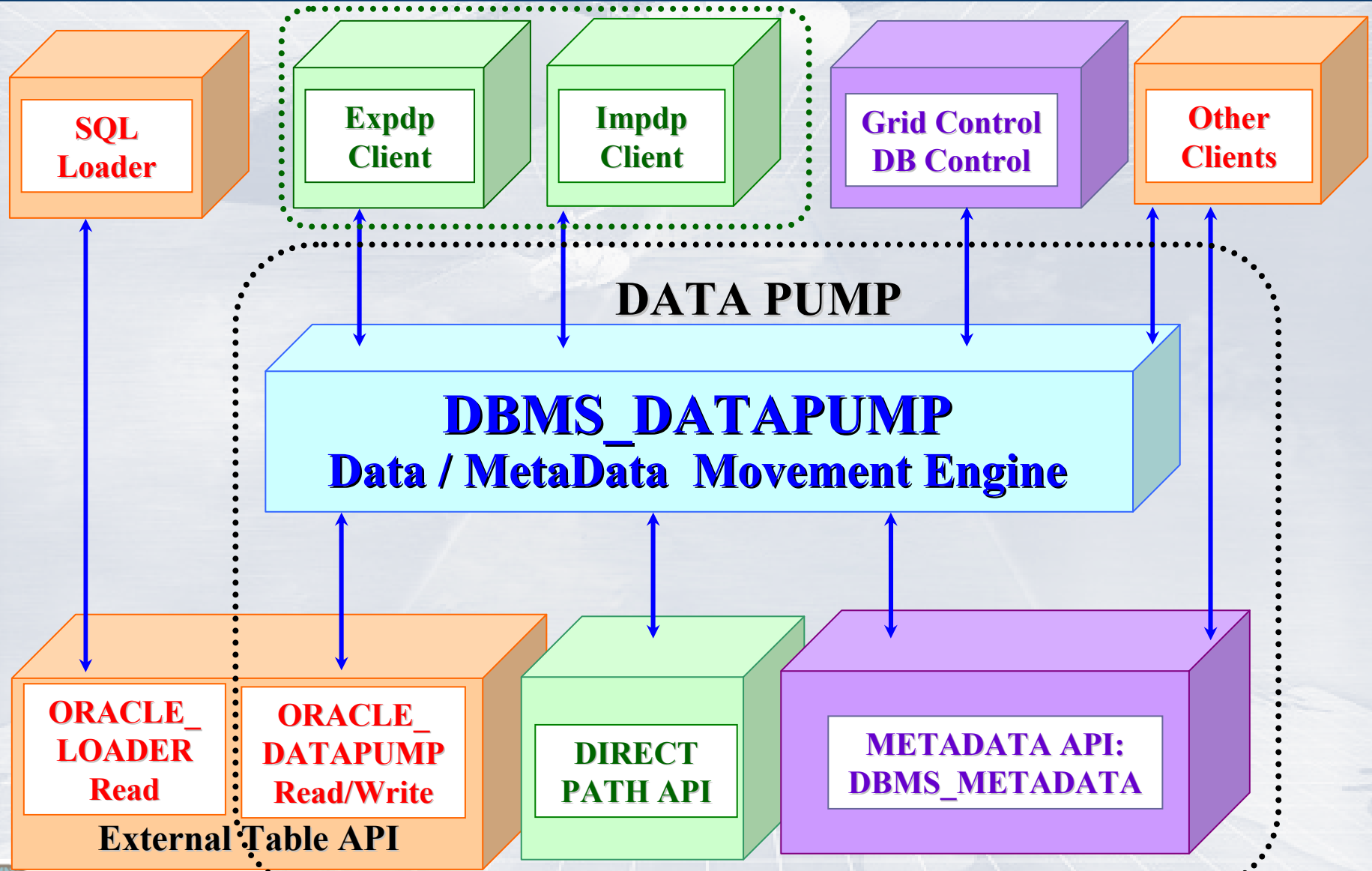
Export/Import Restrictions

- Cannot use for Large Databases
- Slow Data Loading and Unloading
- Limited fine-grained object selection
- Limited Job Monitoring
- No callable API
- Limited Maintenance

DATA PUMP - Introduction

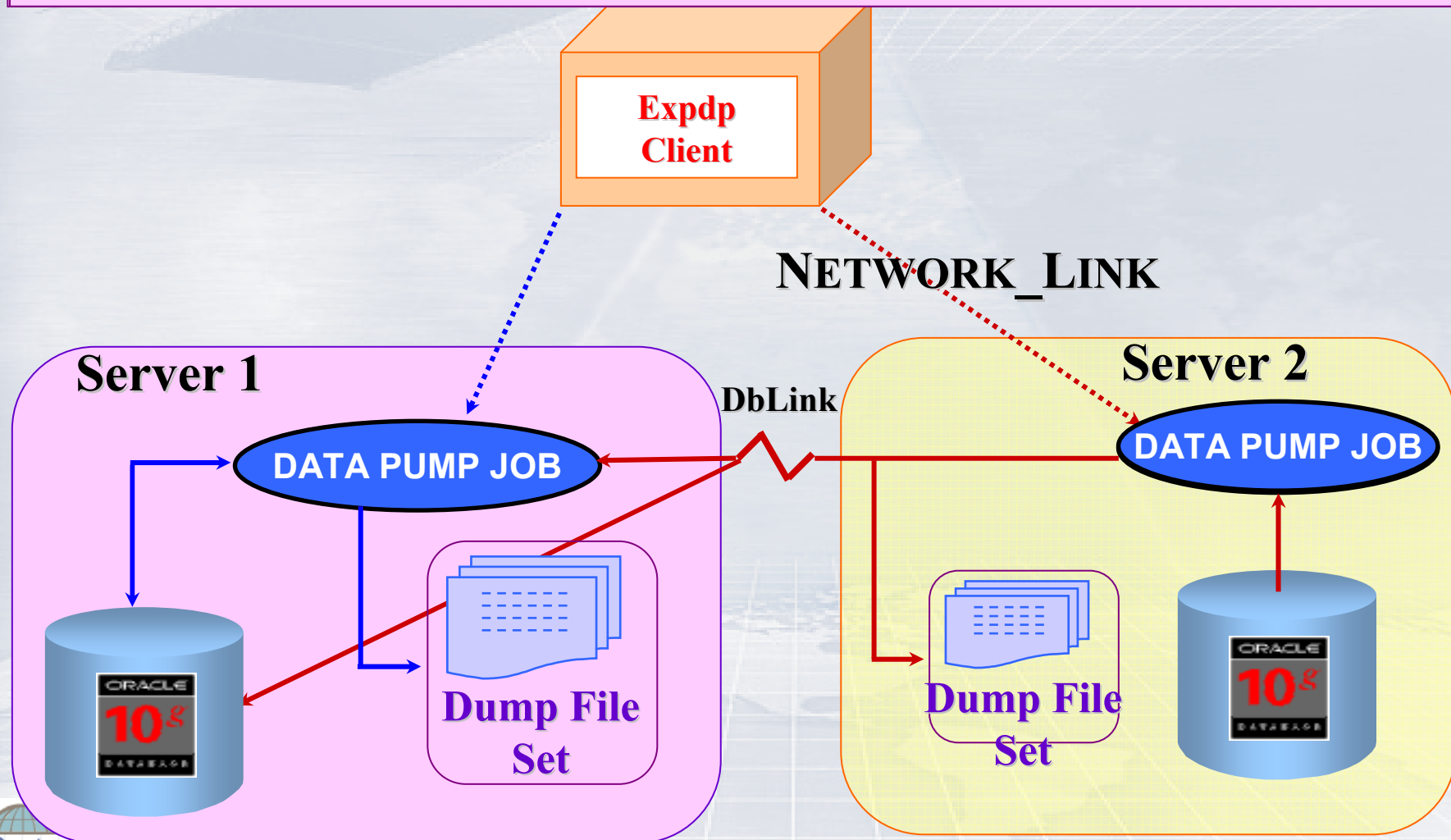
- Oracle Data/Metadata Loading/Unloading utility
- Use PL/SQL API
 - DBMS_DATAPUMP for Data written in direct path stream format
 - DBMS_METADATA for Metadata written in XML format
- New client expdp and impdp
- High speed server based utility
- Create platform independent dump file in binary proprietary format
- Support parallel, bulk data & metadata movement
- Available with all Oracle 10g version for all supported platform
- Can be monitored remotely
- Stop and Restart anytime
- Allow Data movement thru database links

DATA PUMP Architecture



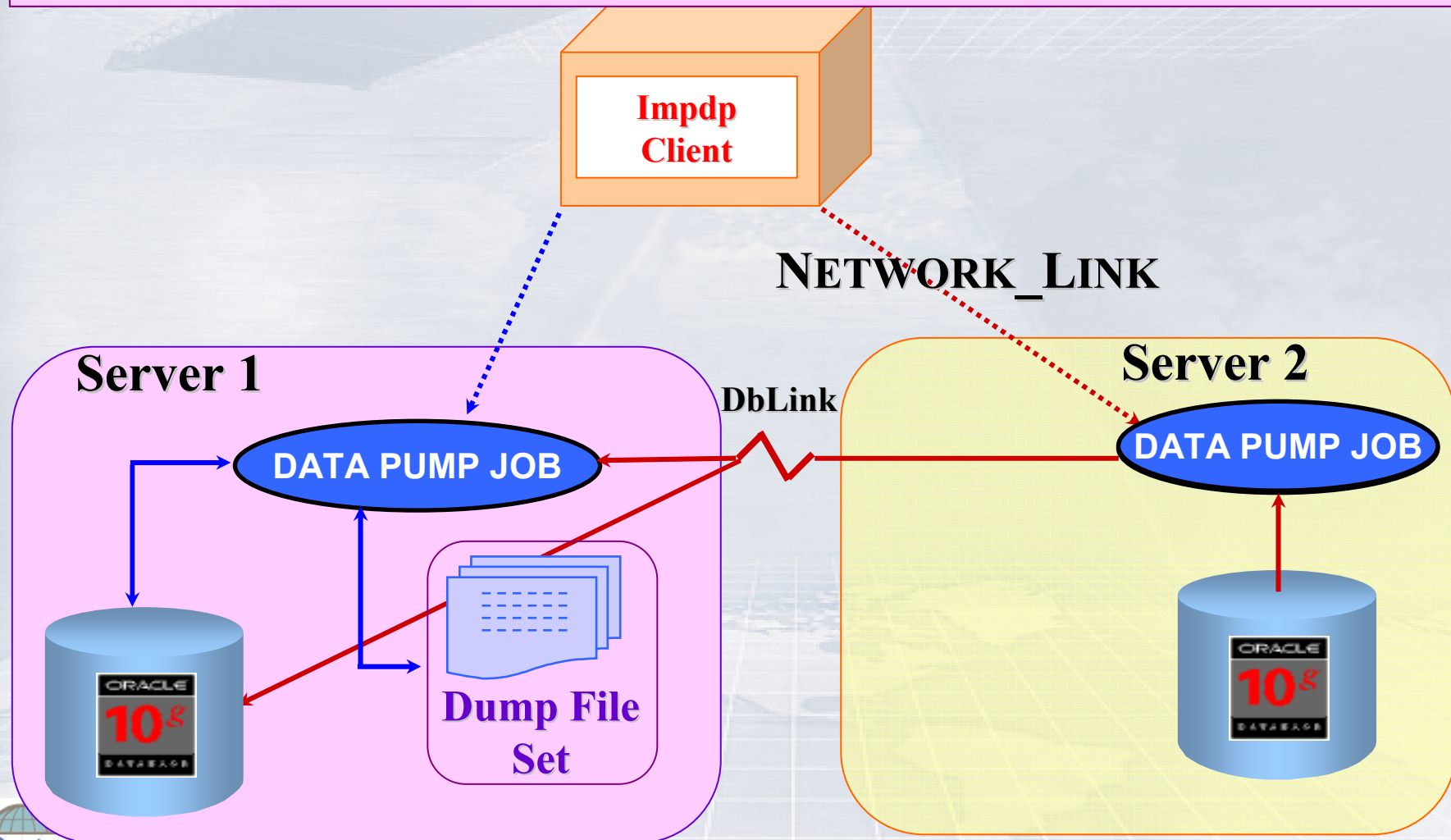
DATA PUMP – New Export Client

```
Server2>expdp indy/indy network_link=server1 dumpfile=indy.dmp  
logfile=indy.log full=y
```

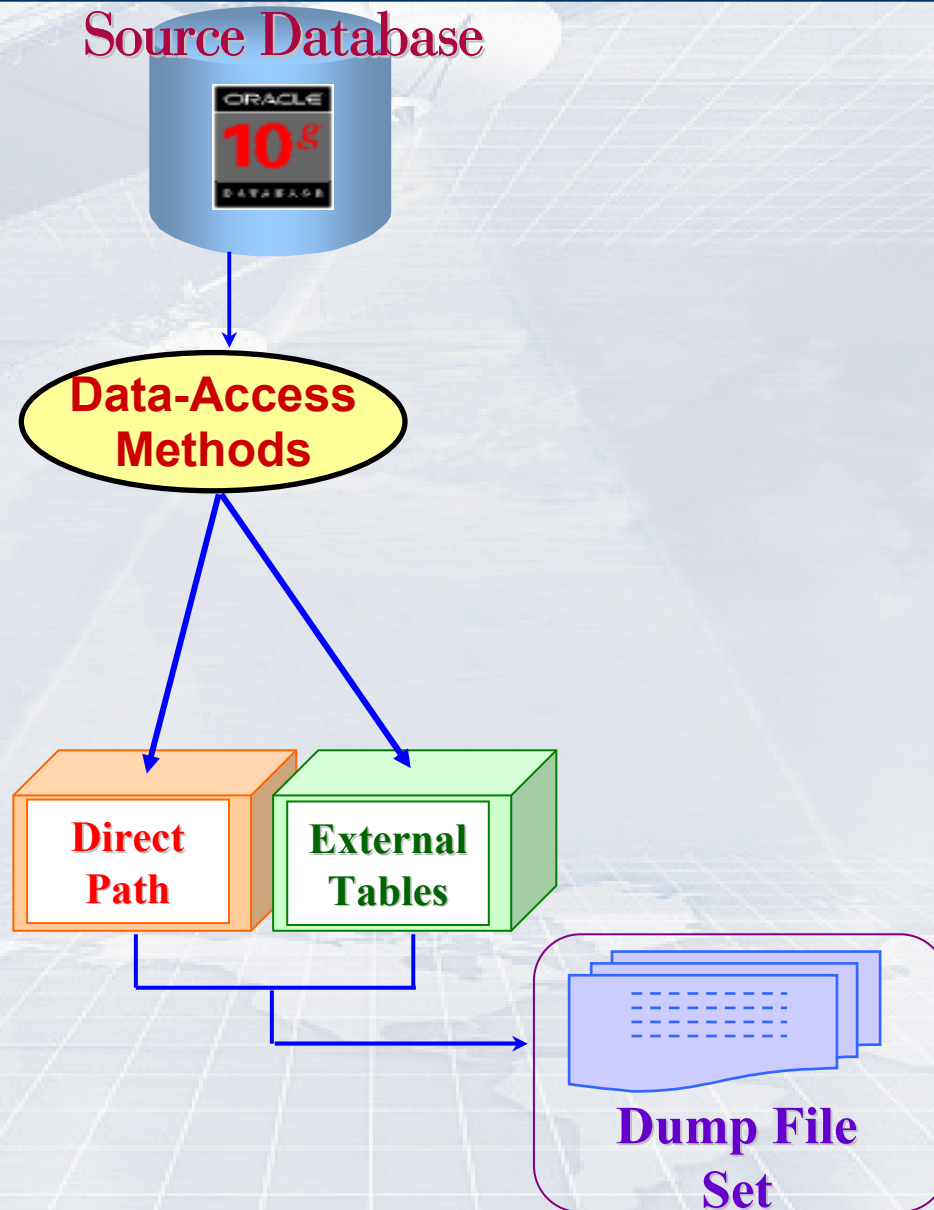


DATA PUMP – New Export Client

```
Server2>impdp indy/indy network_link=server1 dumpfile=indy.dmp  
logfile=indy.log full=y
```



DATA PUMP – Data Unloading Features



When DIRECT Path is not used by DATAPUMP

- Loading tables with Global index on partitioned tables exists
- Loading tables with domain index exists for a LOB column
- Loading Clustered tables
- Loading tables with active Triggers
- Loading and Unloading tables with encrypted columns
- Loading tables with fine-grained access control enabled for inserts
- Loading table containing BFILE or VARRAY columns

DATA PUMP – Data Unloading Features

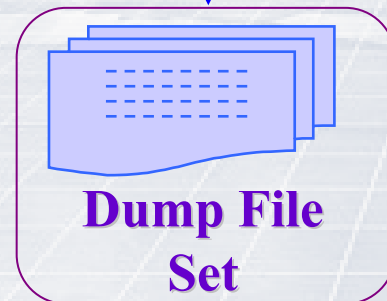
Source Database



Data-Access
Methods

Fine-Grained
Filtering

Procedure/Function
Package/Trigger/views
Grant/Constraints
EXCLUDE/INCLUDE



Include = function

Include = procedure

Include = trigger

Include = view

include = table: **"in ('emp','dept')"**

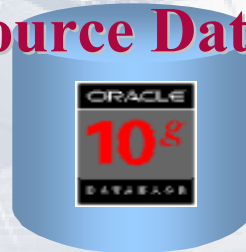
Exclude = index

Exclude = trigger

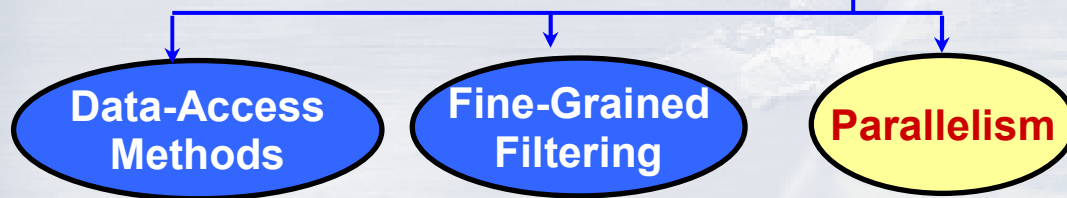
Exclude = view: **"like 'app%'"**

DATA PUMP – Data Unloading Features

Source Database



expdp hr/hr **parallel=3**



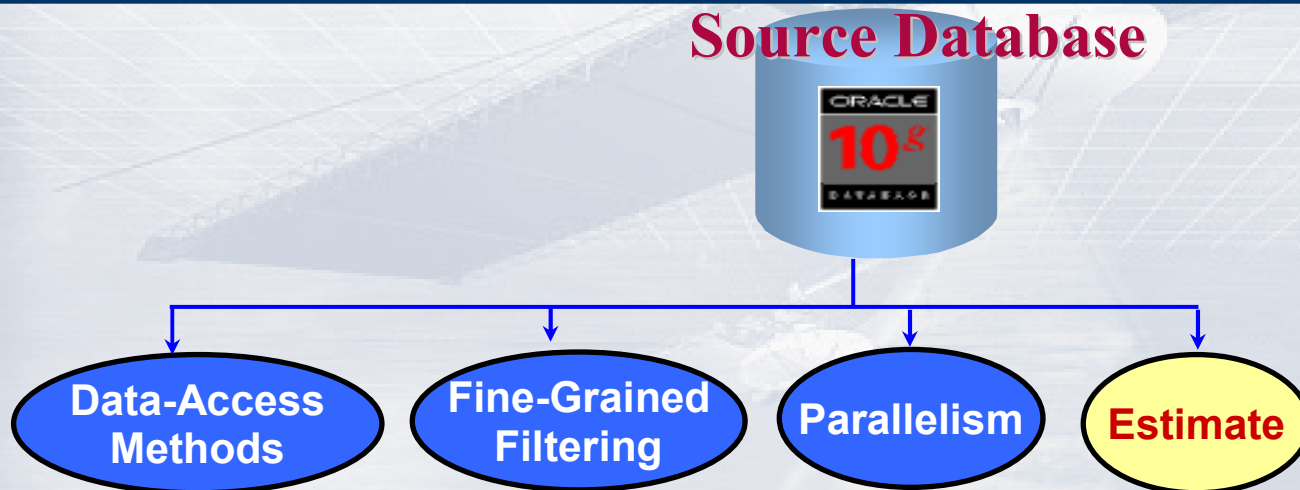
PARALLEL

To specify the number of
Worker processes



Dump File Set

DATA PUMP – Data Unloading Features



ESTIMATE -- Specifies space used by each Table data
 -- Only for Data and not for Metadata
 -- Available in the Logfile or output device

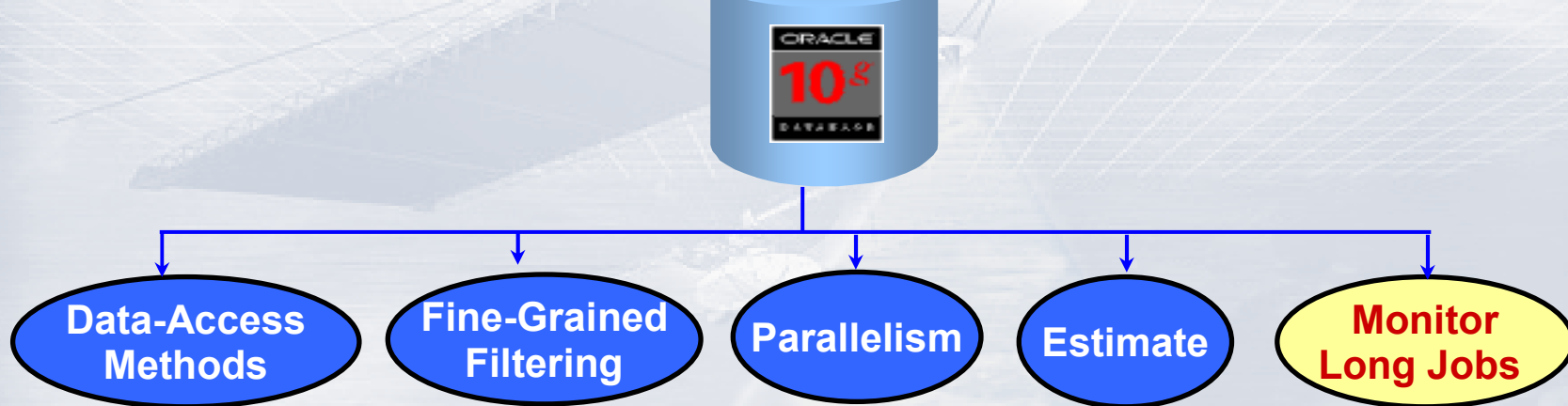
... exported "HR"."PRODUCT" 136.4 MB 649114 rows

ESTIMATE_ONLY -- To determine space required for Export
 -- No Data is actually exported

\$ expdp scott/tiger ESTIMATE_ONLY=Y

DATA PUMP – Data Unloading Features

Source Database



JOB_NAME

-- To Specify the Name to the Job

If not given, System generated name is given

expdp hr/hr JOB_NAME=fullScott

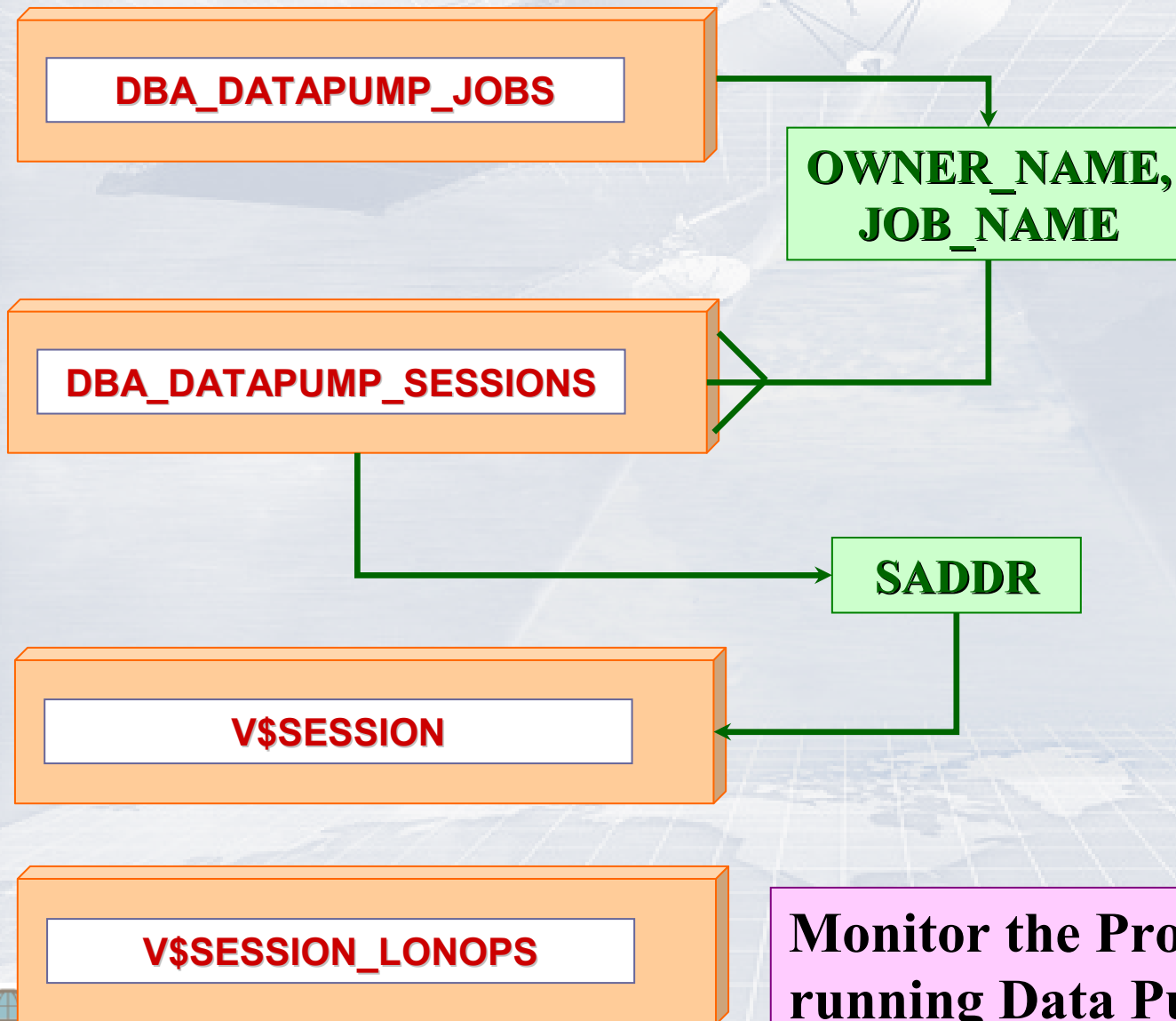
ATTACH

-- To Attach to existing Job

expdp hr/hr ATTACH= fullScott

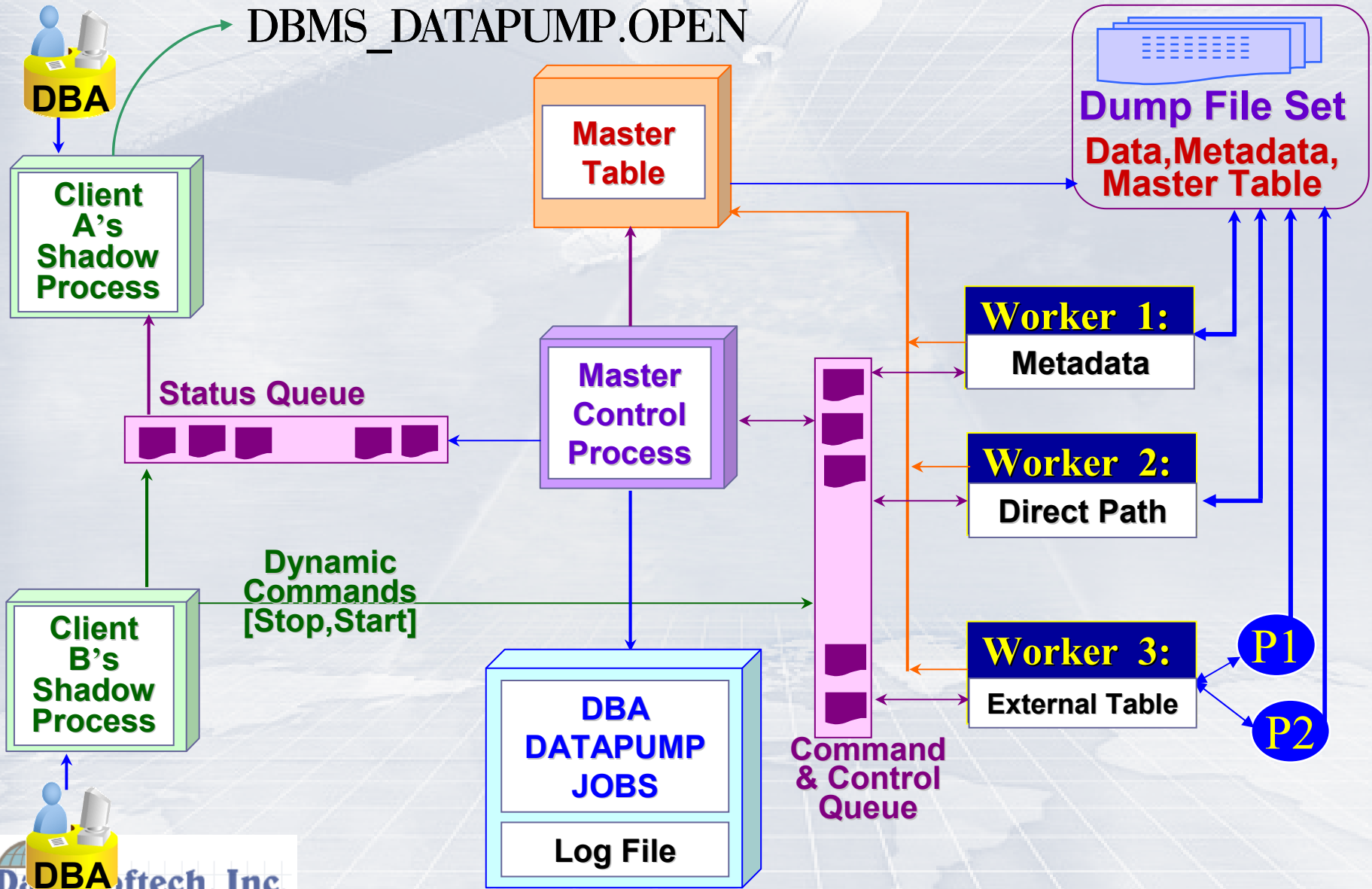
DATA PUMP – Dictionary Views

Monitor
Long Jobs



Monitor the Progress of Long-running Data Pump Job

DATA PUMP IMPLEMENTATION



DATA PUMP Process Components

Client
A's
Shadow
Process

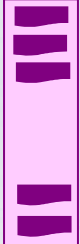
- Foreground process created with datapump calls like `dbms_datapump.open`
- Create a Job based on `JOB_NAME` parameter
- **Create Master Table**
- Create 2 Queues namely
 - Status Queue and
 - Command & Control Queue
- Create MCP
- Give Status to the Client A
- Process goes away when client detach

Master
Table

Master
Control
Process



Status
Queue



Command
& Control
Queue

DATA PUMP Process Components



**Master
Table**

- Table created within the Schema who initiated Expdp/Impdp
- Table Name = JOB_NAME (If specified)
OR
- SYS_<operation>_<mode>_NN
where NN → 2-digit integer starting at 01
Operation → EXPORT | IMPORT
Mode → FULL | TABLESPACE | SCHEMA
TABLE | TRANSPORTABLE
- Heart of Datapump Containing Dumpfile Set, Job parameters, Status of Exported/Imported database objects, Location of Dumpfile etc

DATA PUMP Process Components



**Master
Control
Process**

- One Background process for each Datapump operation
- Ora_dmNN_ <instanceName>
- Responsible for
 - Assigning Data/Metadata Loading/Unloading to Workers Process
 - Maintain Job State
 - Establish and Maintain Dump file Set information
 - Maintain Restart and dump file set information in Master Table
 - Logging operations
 - Update Status Queue

DATA PUMP Process Components

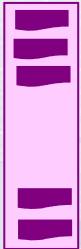


Status Queue

Status Queue

- Populated by MCP
- Consumed by Clients Shadow Process.
- Following information is available for clients
 - Status of the Job
 - Logging Information
 - Errors Encountered
- Queue name like
KUPC\$\$_1_<JobQueTimestamp>
KUPC\$\$_1_20060221135223
- Queue is owned by SYS
- Can check the name in Alert.log file

DATA PUMP Process Components



Command
& Control
Queue

Command & Control Queue

- Used by MCP to send command and control of worker process.
- Used by Client shadow process for executing Abort/Restart request of Datapump job to MCP
- Bidirectional as MCP send and listen to the queue
- Queue name like
KUPC\$C_1_<JobQueTimestamp>
KUPC\$C_1_20060221135223
- Queue is owned by SYS
- Can check the name in Alert.log file

DATA PUMP Process Components

Worker 1: Metadata

- Number of Process depend on the value of PARALLEL parameter.

Worker 2: Direct Path data

- Responsible for Loading/Unloading of Data and Metadata.

Worker 3: External Table

- Background process of the form Ora_DWnn_ <instanceName>
- Describe what object it is working on like Tables or Indexes or views and so on
- Send information about the current status of Database objects like Pending, Completed, Failed and so on

DATA PUMP Process Components



- Represented by DUMPFILE parameter.
- Contains the Data as well as metadata information
- Location depend on the DIRECTORY setting

```
SQL> CREATE DIRECTORY dr1 AS '/oradata/dpdir';
```

```
SQL> GRANT READ ON DIRECTORY dr1 TO hr; → Import
```

```
SQL> GRANT WRITE ON DIRECTORY dr1 TO hr; → Export
```


DATA PUMP Fileset Location Preferences

1

PER- FILE Directory Specifications e.g
DUMPFILE=DR1:ab.dmp

2

Use DIRECTORY Parameter e.g
DIRECTORY=DR1

3

Use DATA_PUMP_DIR Env Setting e.g.
\$ DATA_PUMP_DIR=DR3

4

Use Default DATA_PUMP_DIR Database
Setting e.g.
Default to \$ORACLE_HOME/rdbms/log

DATA PUMP different File Types

DUMPFIL -- Used for creating Dump Files

LOGFILE -- Used for Creating Log file

SQLFILE -- Used for DDL Script File

DATA PUMP Export Interfaces

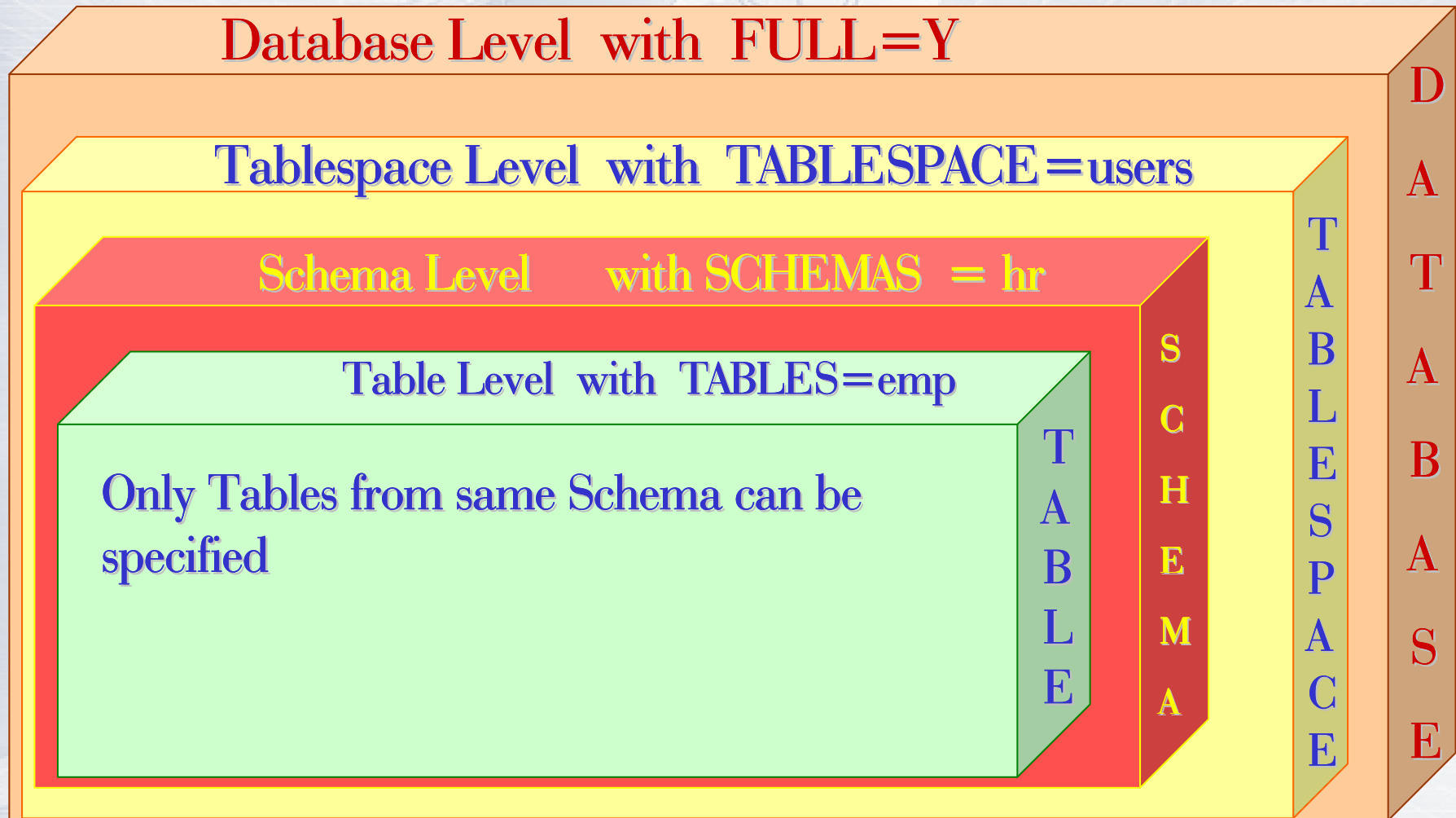
Command Line Interface

```
$ expdp Id/Pass DIRECTORY=abc DUMPFILE=indy.dmp PARALLEL=4
```

Parameter File Interface

```
$ expdp PARFILE=indy.par
```

DATA PUMP – Different Modes



9i Export Vs 10g Datapump Export

Export [exp]	Data Pump Export [expdp]
FILE	DUMPFIL
LOG	LOGFILE
GRANTS	EXCLUDE & INCLUDE
INDEXES	EXCLUDE & INCLUDE
CONSTRAINTS	EXCLUDE & INCLUDE
FEEDBACK	STATUS
OWNER	SCHEMAS
ROWS=N	CONTENT=METADATA_ONLY
ROWS=Y	CONTENT=ALL
CONSISTENT	FLASHBACK_SCN

INCLUDE = GRANTS

9i Export Vs 10g Datapump Export

Export [exp]	Data Pump Export [expdp]
RECORDLENGTH	NONE
RESUMABLE	NONE
RESUMABLE_NAME	NONE
USERID	NONE
VOLSIZE	NONE
COMPRESS	NONE
STATISTICS	NONE
BUFFER	NONE
DIRECT	NONE
NONE	TABLESPACES

DATA PUMP Export- New Parameter

COMPRESSION

-- To Compress Metadata in Dumpfile

METADATA_ONLY | NONE

By Default it is METADATA_ONLY

CONTENT

-- To take export of Data | Metadata | Both

DATA_ONLY | METADATA_ONLY | ALL

KEEP_MASTER

-- To Keep Master Table in the Schema

KEEP_MASTER=Y

DATA PUMP Export- New Parameter

- NETWORK_LINK** -- To specify name of Database Link Name
Enable Network Export
- NOLOGFILE** -- By default Logfile export.log is created
- VERSION** -- To specify the Version of database objects to be exported
- SAMPLE** -- To specify the Percentage of data to be sampled

You can specify the specific table

SAMPLE = 100

SAMPLE = "SCOTT"."DEPT":40

DATA PUMP – Example # 1

```
$ expdp hr/hr FULL=Y PARALLEL=4  
DUMPFILE=DR1:fullhr1%U.dmp,  
DR2:fullhr2%U.dmp,  
DR3:fullhr3%U.dmp,  
DR4:fullhr4%U.dmp,
```

This will Create

- Files as fullhr101.dmp, fullhr201.dmp and so
- Job name as SYS_EXPORT_FULL_01
- Master Table Name as SYS_EXPORT_FULL_01
- Use DR1,DR2,DR3,DR4 Directory defined in Database



DATA PUMP – Example # 2

```
$ expdp hr/hr PARFILE = indy.par
```

```
DIRECTORY=DR1
```

```
DUMPFILE=hr1%U.dmp
```

```
SCHEMAS=HR
```

```
INCLUDE=PROCEDURE
```

```
INCLUDE=PACKAGE
```

```
INCLUDE=VIEW:"like 'VW_H%' "
```

This will export

- All Procedure, Packages and views like VW_H
- No Schema Definition/other System grants are exported
- Master Table Name as SYS_EXPORT_FULL_01
- Use DR1 Directory defined in Database



New Interactive Mode

Monitor
Jobs Remotely

```
$ expdp hr/hr ATTACH = jobname
```

Or

```
$ expdp hr/hr ATTACH
```

... You will be attached to the Job running as HR User

Or

```
$ expdp hr/hr dumpfile=a.dmp logfile=a.log full=y
```

... when Export is started press Cntrl C [^ C]

... You will get EXPORT> prompt

```
EXPORT>
```

```
expdp prsearch/prsearch directory=data_pump_dir dumpfile=prs111.dmp logfile=prs11.log
```

```
Export: Release 10.2.0.1.0 - 64bit Production on Friday, 03 March, 2006 17:10:55
```

```
Copyright (c) 2003, 2005, Oracle. All rights reserved.
```

```
Connected to: Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - 64bit Product
```

```
With the Partitioning, OLAP and Data Mining options
```

```
Starting "PRSEARCH"."SYS_EXPORT_SCHEMA_01": prsearch/***** directory=data_pump_dir
```

```
Estimate in progress using BLOCKS method...
```

```
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
```

```
^C
```

```
Export> help
```

```
The following commands are valid while in interactive mode.
```

```
Note: abbreviations are allowed
```

Command	Description
ADD_FILE	Add dumpfile to dumpfile set.
CONTINUE_CLIENT	Return to logging mode. Job will be re-started if idle.
EXIT_CLIENT	Quit client session and leave job running.
FILESIZE	Default filesize (bytes) for subsequent ADD_FILE commands.
HELP	Summarize interactive commands.
KILL_JOB	Detach and delete job.
PARALLEL	Change the number of active workers for current job. PARALLEL=<number of workers>.



DATA PUMP – Interactive Mode

Monitor
Jobs Remotely

ADD_FILE	-- Add file or use %U
CONTINUE_CLIENT	-- Back to Logging Mode
EXIT_CLIENT	-- Exit Interactive Mode Session broke but job continue
KILL_JOB	-- Session broke and Job killed Cannot be restarted
PARALLEL	-- Increase or Decrease Workers
START_JOB	-- Start the job to which attached
STATUS	-- Display the Current Status of Job
STOP_JOB	-- Stop the Current Job Job can be restarted

New Interactive Mode

Monitor
Jobs Remotely

New Parameter in Release 2

FILESIZE --You can restrict the size of file added with
ADD_FILE

In Rel 1, this file size was **UNLIMITED**

Interactive Mode Example

Monitor
Jobs Remotely

```
$ expdp hr/hr attach = export1
```

```
....
```

```
Export> parallel = 5
```

```
Export> add_file = export5.dmp
```

```
Export> start_job
```

```
Export> continue_client
```

9i Import Vs 10g Datapump Import

Import [imp]	Data Pump Import [impdp]
FILE	DUMPFIL
LOG	LOGFILE
GRANTS	EXCLUDE & INCLUDE
INDEXES	EXCLUDE & INCLUDE
CONSTRAINTS	EXCLUDE & INCLUDE
FEEDBACK	STATUS
ROWS=N	CONTENT=METADATA_ONLY
ROWS=Y	CONTENT=ALL
DESTROY	RESUSE_DATAFILES
FROMUSER	REMAP_SCHEMA
TOUSER	REMAP_SCHEMA
INDEXFILE	SQLFILE
IGNORE	TABLE_EXISTS_ACTION

9i Import Vs 10g Datapump Import

Import [imp]	Data Pump Import [impdp]
RECORDLENGTH	NONE
RESUMABLE	NONE
RESUMABLE_NAME	NONE
USERID	NONE
VOLSIZE	NONE
STATISTICS	NONE
BUFFER	NONE
DIRECT	NONE
COMMIT	NONE
FILESIZE	NONE
NONE	TABLESPACES
NONE	SKIP_UNUSABLE_INDEXES
NONE	STREAMS_CONFIGURATION

DATA PUMP Import New Parameter

JOB_NAME | ATTACH | DIRECTORY | NETWORK_LINK | PARALLEL

ENCRYPTION_PASSWORD -- Allow export/import of Encrypted column data

```
$expdp hr/hr dumpfile=dr1:abc.dmp encryption_password=6789
```

```
$impdp hr/hr dumpfile=dr1:abc.dmp encryption_password=6789
```

You cannot use this parameter with NETWORK_LINK

FLASHBACK_SCN -- Valid with NETWORK_LINK to get
FLASHBACK_TIME Data consistent as of this SCN | Time

Both of these parameter are mutually exclusive

DATA PUMP Import New Parameter

- REMAP_DATAFILE** -- Useful when moving data between platform
Change Source file to Target file in DDL's
- REMAP_SCHEMA** -- Load all Source Schema objects to Target
Schema
-- Can create Target Schema if dump file set
contains CREATE USER
`remap_schema(HR,SCOTT)`
- REMAP_TABLESPACE** -- All Source Tablespace are mapped to
destination tablespaces
-- You can specify multiple such parameters
`remap_tablespace(users,indy_data)`

DATA PUMP Import New Parameter

TABLE_EXISTS_ACTION -- What to do if table already exist

SKIP → Leave the table and move to next object [Default]

APPEND → Load rows leaving existing data unchanged

TRUNCATE → Delete existing rows and load rows

REPLACE → Drop existing table and create and load rows

DATA PUMP Import New Parameter

TRANSFORM -- Allow to alter Object Creation DDL before Loading

TRANSFORM = <TrName>:Value[:object_type]

Where <TrName> can be

SEGMENT_ATTRIBUTES = Y | N --Y→Default

-- Y means following are included with DDL

- Physical attributes
- Storage attributes
- Tablespaces
- logging

STORAGE = Y | N --Y→Default

-- Y means Storage clauses are included

OID = Y | N --Y→Default

-- Y means object ID of abstract types is reused

PCTSPACE = Integer value

-- Reduce the space required for tablespace by performing shrink on tablespace storage

DATA PUMP Import New Parameter

```
Impdp hr/hr dumpfile=indy.dmp transform=SEGMENT_ATTRIBUTES:n:index  
----- Ignore Index Storage and Tablespaces Clause
```

DBMS_DATAPUMP API

Use the following steps to create Datapump script

1. Create a handle using DBMS_DATAPUMP.OPEN function

Handle1 :=

```
DBMS_DATAPUMP.OPEN('<operation>', '<modes>', '<dblink>', '<jobname>', '<version>');
```

Operation can be EXPORT | IMPORT | SQL_FILE

Modes can be FULL | SCHEMA | TABLESPACE | TRANSPORTABLE | TABLE

Dblink is the name of database link

Jobname is the name of Datapump job

Version can be LATEST | COMPATIBLE

2. Get the Dumpfile Set information

```
DBMS_DATAPUMP.GET_DUMPFILE_INFO('indy.dmp', 'DATA_PUMP_DIR', 'finfo,ftype');
```

3. Add any Object filtering

```
DBMS_DATAPUMP.METADATA_FILTER(handle1, 'SCHEMA_NAME_EXPR', 'IN("SCOTT")');
```

4. Start the Datapump Job

```
DBMS_DATAPUMP.START_JOB(handle1);
```

Enterprise Manager-DbControl

ORACLE Enterprise Manager 10g Database Control

Database Instance: indydb

[Home](#) [Performance](#) [Administration](#) **Maintenance**

The Administration tab displays links that allow you to administer database objects and initiate database operations inside an Oracle database between or outside Oracle databases.

High Availability

Backup/Recovery

[Schedule Backup](#)
[Perform Recovery](#)
[Manage Current Backups](#)
[Manage Restore Points](#)
[Backup Reports](#)

Backup/Recovery Settings

[Backup Settings](#)
[Recovery Settings](#)
[Recovery Catalog Settings](#)

Data Movement

Move Row Data

[Export to Export Files](#)
[Import from Export Files](#)
[Import from Database](#)
[Load Data from User Files](#)
[Monitor Export and Import Jobs](#)

Move Database Files

[Clone Database](#)
[Transport Tablespaces](#)

Software Deployments

Installed Database Software

[Collection Status](#)

Database Software Patching

[Apply Patch](#)
[View Patch Cache](#)

[Home](#) [Performance](#) [Administration](#) **Maintenance**



QUESTIONS
ANSWERS

Thank You