

**ORACLE®**



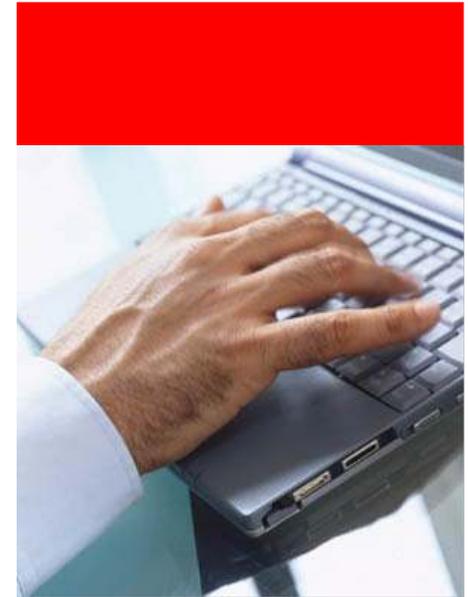
**ORACLE®**

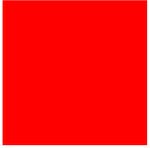
# Обзор Oracle Datapump

**Игорь Мельников**  
старший консультант

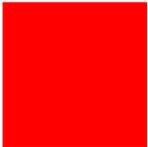
# План

- Что такое Oracle Data Pump ?
- Архитектура и возможности
- Использование
- API для разработчиков
- Новые возможности в Oracle Database 11g
- Демонстрация



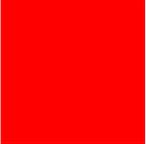


# Что такое Oracle Datapump



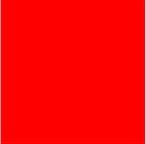
## Недостатки традиционного экспорта/импорта

- Не масштабируется для больших БД
- Медленная выгрузка и загрузка данных, не распараллеливается
- Трудности в мониторинге процессов экспорта/импорта
- Практически невозможно встраивать в приложения (нет API)
- Тяжело сопровождать



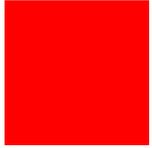
# Что такое Oracle Datapump

- Технология экспорта/импорта данных и метаданных СУБД Oracle Database 10g/11g
- Замена традиционных утилит экспорта/импорта (*exp* и *imp*)
- Новая серверная инфраструктура и утилиты (*expdp* и *impdp*)
- Имеется документированный API для разработчиков
- Много новых возможностей по сравнению с утилитами *exp* и *imp*
- Утилиты *exp/imp* оставлены в целях совместимости !



# Утилиты Oracle Datapump

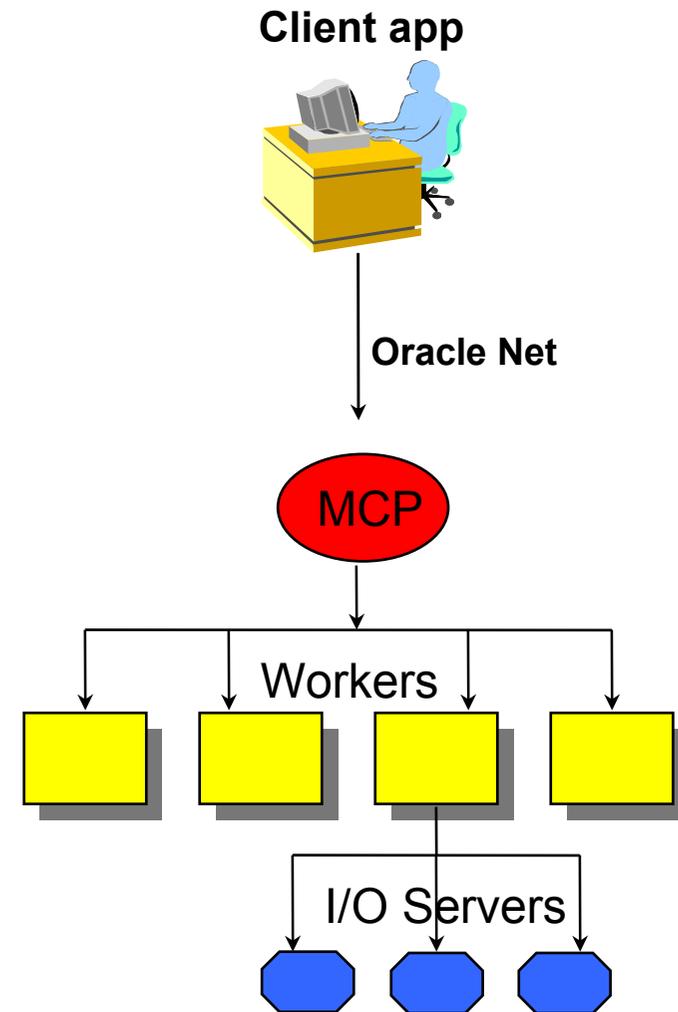
- *expdp* – для экспорта, *impdp* – для импорта
- Расположены в \$ORACLE\_HOME/bin
- Большинство параметров аналогичны параметрам в *exp/imp*
- Формат dmp-файлов не совместим с форматом *exp/imp* !
- Совместимость формата Datapump для всех версий СУБД начиная с 10.1 (10.2 и 11.1)



# Архитектура и возможности

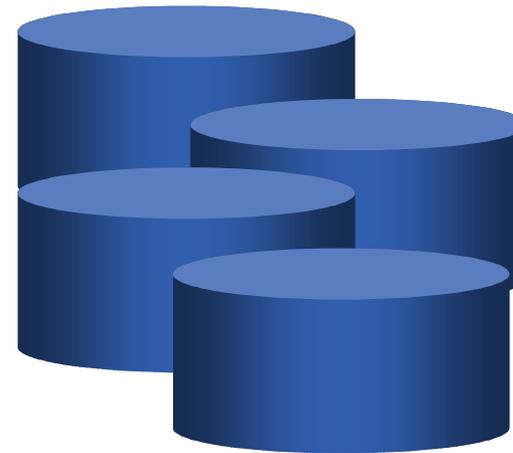
# Архитектура Oracle Dataump

- Инфраструктура для поддержки “тонкого клиента”
- Экспорт/импорт происходит на сервере
- dmp-файл формируется на файловой системе сервера
- Клиент только управляет процессами экспорта/импорта
- Поддержка параллельности
  - Только в Enterprise Edition
  - И для выгрузки, и для загрузки
  - Динамический уровень параллелизма
  - Как на уровне одной таблицы/разделов, так и на уровне нескольких таблиц



# Возможности Oracle Datapump

- Рестарт и останов экспорта/импорта на произвольной точке
  - Master table
  - По команде или по ошибке
  - Старт с пред. точки останов
  - Асинхронный запуск
- Гибкий выбор объекта
- Трансформация DDL при импорте
  - Замена имен схемы, файлов данных, табличных пространств
  - Включать/не включать параметры хранения (storage clause)
  - Генерация нового OID для объектных типов



# Возможности Oracle Datapump

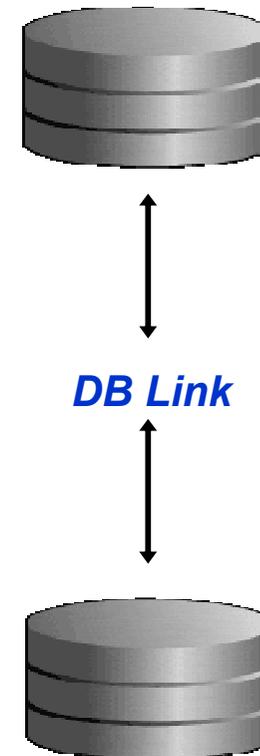
- Предварительная оценка размера dmp-файла
- Детальный мониторинг с любого места
  - % выполнения
  - Детальный лог
- Интерфейс в OEM
- Сжатие (только метаданные)

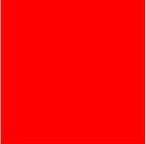


# Возможности Oracle Datapump

## Экспорт/импорт по сети

- Импорт по сети
  - Из одной БД в другую
  - На “лету”: без промежуточного dmp-файла
- Экспорт по сети
  - Из БД в режиме “только для чтения”



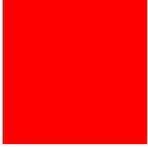


# Внешние таблицы Datapump

- Новый тип внешних таблиц: datapump external tables

```
create table test
  organization external
  (
    type ORACLE_DATAPUMP
    default directory dump_dir
    location ('all_object.dmp')
  )
as select * all_objects;
```

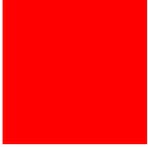
- Возможна запись в таблицы (только на этапе формирования CTAS-таблиц)



# Производительность Oracle Datapump

## *Экспорт*

- 1 GB данных (9.3 млн. строк) – портал крупной компании
- Сервер: 1-CPU Ultra 60, 1 GB RAM, 2 HDD (SCSI 10000 rpm)
- Время экспорта:
  - Утилита exp: 143 сек.
  - Data Pump Export: 92 сек.
  - В общем случае скорость зависит от степени параллелизма, скорости I/O, характеристик CPU ...



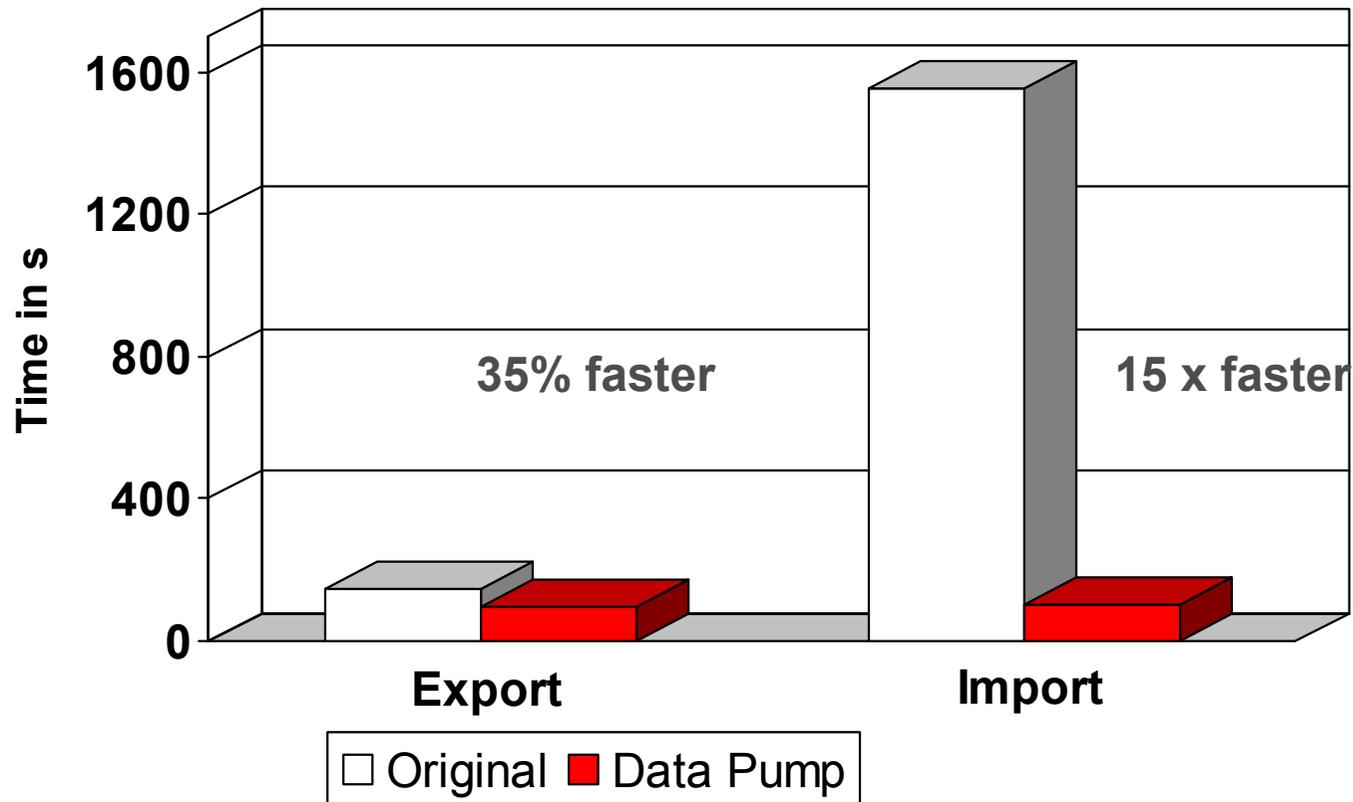
# Производительность Oracle Datapump

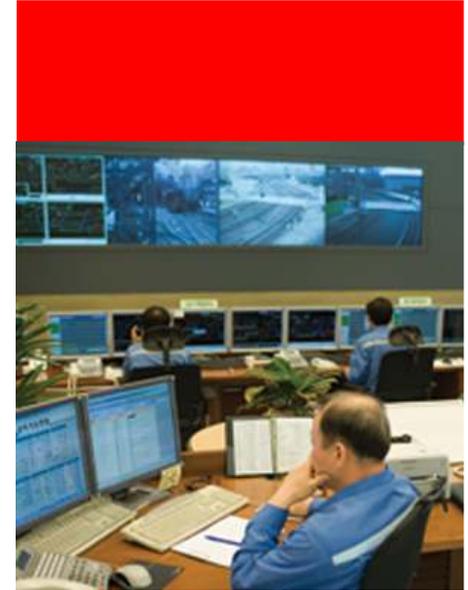
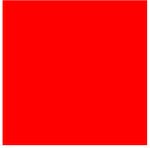
## *Импорт*

- Импорт данных:
  - Утилита imp: 25 мин. 55 сек.
  - Data Pump Import: 1 мин. 41 сек.
  - Data Pump в 15 раз быстрее!
- При использовании старых утилит (exp/imp) практически нет возможностей для настройки производительности
- Data Pump дает готовое производительное решение с возможностью гибкой настройки (степень параллелизма и сжатие)

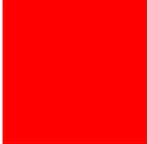
# Производительность Oracle Datapump

## *Сравнение с утилитами exp/imp*





# Использование Oracle Datapump



# Использование Oracle Datapump

## *Экспорт*

- Создаем директорию в СУБД для dmp-файлов

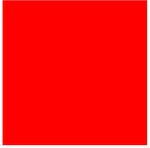
```
SQL> CREATE DIRECTORY dump_dir AS '/backup/oradata';
```

- Выдаем привилегии на директорию

```
SQL> GRANT READ,WRITE ON DIRECTORY dump_dir TO scott;
```

- Выполняем экспорт

```
C:\app>EXPDP scott/tiger@orcl TABLES=employees  
DUMPFILE=dump_dir:employees.dmp NOLOGFILE=y
```



# Oracle Datapump API

# Oracle Datapump API

## *PL/SQL-пакет DBMS\_DATAPUMP*

```
SQL> desc dbms_datapump
```

```
FUNCTION OPEN RETURNS NUMBER
```

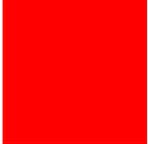
Argument Name	Type	In/Out	Default?
OPERATION	VARCHAR2	IN	
JOB_MODE	VARCHAR2	IN	

```
PROCEDURE START_JOB
```

Argument Name	Type	In/Out	Default?
HANDLE	NUMBER	IN	
SKIP_CURRENT	NUMBER	IN	DEFAULT
ABORT_STEP	NUMBER	IN	DEFAULT

```
PROCEDURE STOP_JOB
```

Argument Name	Type	In/Out	Default?
HANDLE	NUMBER	IN	
IMMEDIATE	NUMBER	IN	DEFAULT
KEEP_MASTER	NUMBER	IN	DEFAULT
DELAY	NUMBER	IN	DEFAULT



# Oracle Datapump API

## *PL/SQL-пакет DBMS\_DATAPUMP*

- Представлено в виде встроенного пакета DBMS\_DATAPUMP
- Позволяет сделать экспорт/импорт прямо из приложения
- Экспорт/импорт возможен как в синхронном, так и в асинхронном режиме
- Возможности по контролю над статусом процессов экспорта/импорта
  - Процент завершения
  - Число полученных байт
  - Состояние задания
  - ....

# Oracle Datapump API

## Пример экспорта

*-- определяем задание на экспорт*

```
v_xHandle := dbms_datapump.open(operation => 'EXPORT',  
                                job_mode  => 'SCHEMA',  
                                job_name  => 'MY_EXPORT');
```

*-- указываем dmp-файл*

```
dbms_datapump.add_file(handle    => v_xHandle,  
                        filename  => 'scott.dmp',  
                        directory => 'dump_dir',  
                        filetype  =>  
                            dbms_datapump.KU$_FILE_TYPE_DUMP_FILE);
```

*-- указываем log-файл*

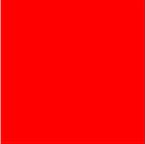
```
dbms_datapump.add_file(handle    => v_xHandle,  
                        filename  => 'scott_dmp.log',  
                        directory => 'dump_dir',  
                        filetype  =>  
                            dbms_datapump.KU$_FILE_TYPE_LOG_FILE);
```

*-- запускаем задание*

```
dbms_datapump.start_job(v_xHandle);
```

*-- уничтожаем задание*

```
dbms_datapump.detach(v_xHandle);
```



# Oracle Datapump API

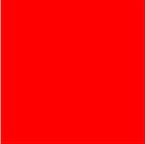
## *Пример импорта*

```
v_xHandle := dbms_datapump.open(operation => 'IMPORT',
                                job_mode  => 'FULL',
                                job_name  => 'MY_IMPORT');

dbms_datapump.add_file(handle    => v_xHandle,
                       filename  => 'scott.dmp',
                       directory => 'dump_dir');

-- импортируем данные в другое табличное пространство !
dbms_datapump.metadata_remap(handle    => v_xHandle,
                              name     => 'REMAP_TABLESPACE',
                              old_value => 'USERS',
                              value    => 'SCOTT_DATA');

dbms_datapump.start_job(v_xHandle);
dbms_datapump.detach(v_xHandle);
```

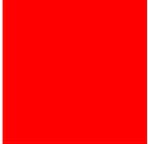


# Oracle Datapump API

## *Контроль выполнения экспорта/импорта*

```
declare
    v_xJobState    varchar2(32);
    v_xJobStatus   ku$_JobStatus;
    v_xStatus      ku$_Status;
begin
    dbms_datapump.get_status(v_xHandle,
                            dbms_datapump.ku$_status_job_status,
                            -1,
                            v_xJobState,
                            v_xStatus);
    v_xJobStatus := v_xStatus.job_status;

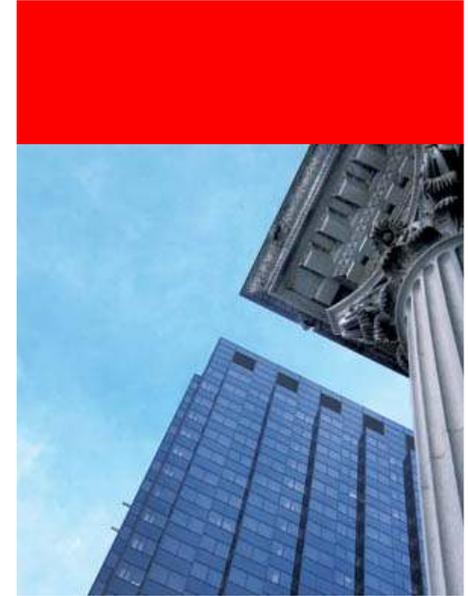
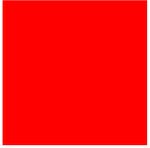
    dbms_output.put_line('Status => ' || v_xJobState);
    dbms_output.put_line('Done % => ' || v_xJobStatus.percent_done);
end;
```



# Oracle Datapump API

## *Резюме*

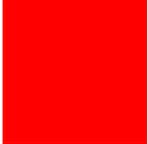
- Позволяет встраивать экспорт/импорт прямо в приложения
- Нет необходимости разрабатывать собственные механизмы экспорта/импорта
- Полный контроль над процессами экспорта/импорта
- Возможно копирование dmp-файла с сервера на клиент, или с клиента на сервер (как BFILE blob) через OCI, ODP.NET, JDBC



# НОВЫЕ ВОЗМОЖНОСТИ В Oracle Database 11g

ORACLE®  
DATABASE 11<sup>g</sup>

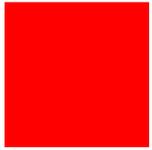
ORACLE ORACLE C



# Oracle Datapump 11g

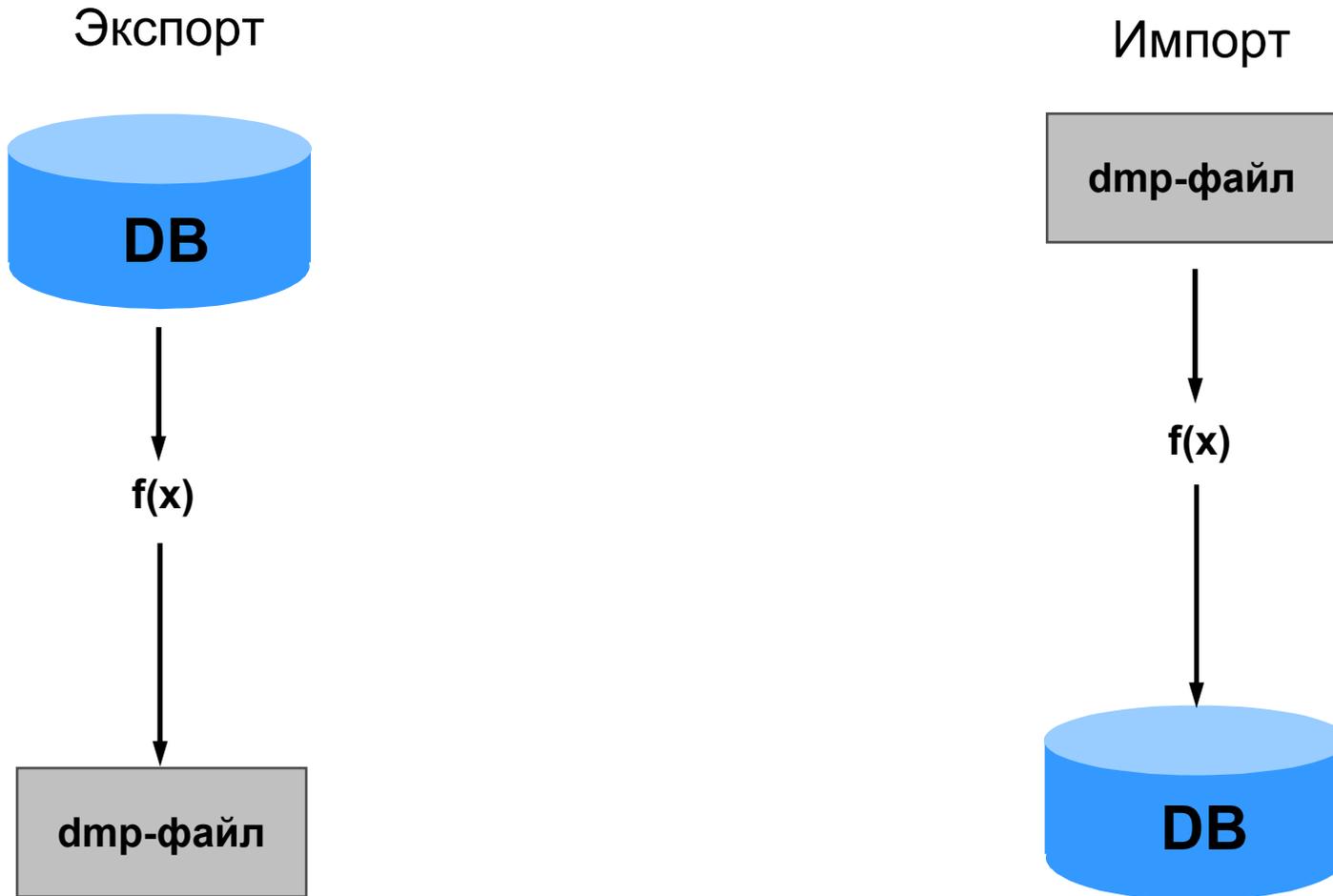
## *Новые возможности в 11g*

- Сжатие данных в ходе экспорта (в 10g - только метаданные)
- Обработка разделов (partitions)
  - PARTITION\_OPTIONS = {none | departition | merge}
- Шифрование dmp-файла в ходе экспорта
  - Алгоритм (AES128 | AES192 | AES256)
  - Параметры (ENCRYPTION\_ALGORITHM и ENCRYPTION\_MODE)
- Переименование таблиц в ходе импорта
  - Параметр REMAP\_TABLE
- Множество улучшений (REUSE\_DUMPFILES, XMLTYPE конвертация и д.р.)



# Oracle Dataump 11g

*Преобразование данных при экспорте/импорте*

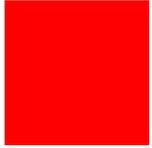




# Oracle Datapump 11g

## *Преобразование данных при экспорте/импорте*

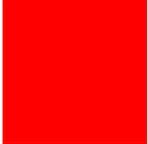
- Преобразование данных при экспорте/импорте
  - Параметр REMAP\_DATA
  - Значение: `[schema.]tablename.column_name:[schema.]pkg.function`
- Определяется PL/SQL-функция преобразования
  - Произвольная PL/SQL – функция
  - Принимает на вход значение столбца и возвращает новое значение столбца
- При экспорте значения столбцов в dmp-файле будут преобразованы (на “лету”)
- При импорте – данные перед записью в БД будут на “лету” преобразованы



**ORACLE®**

**Демонстрация**

**Oracle Data Pump**



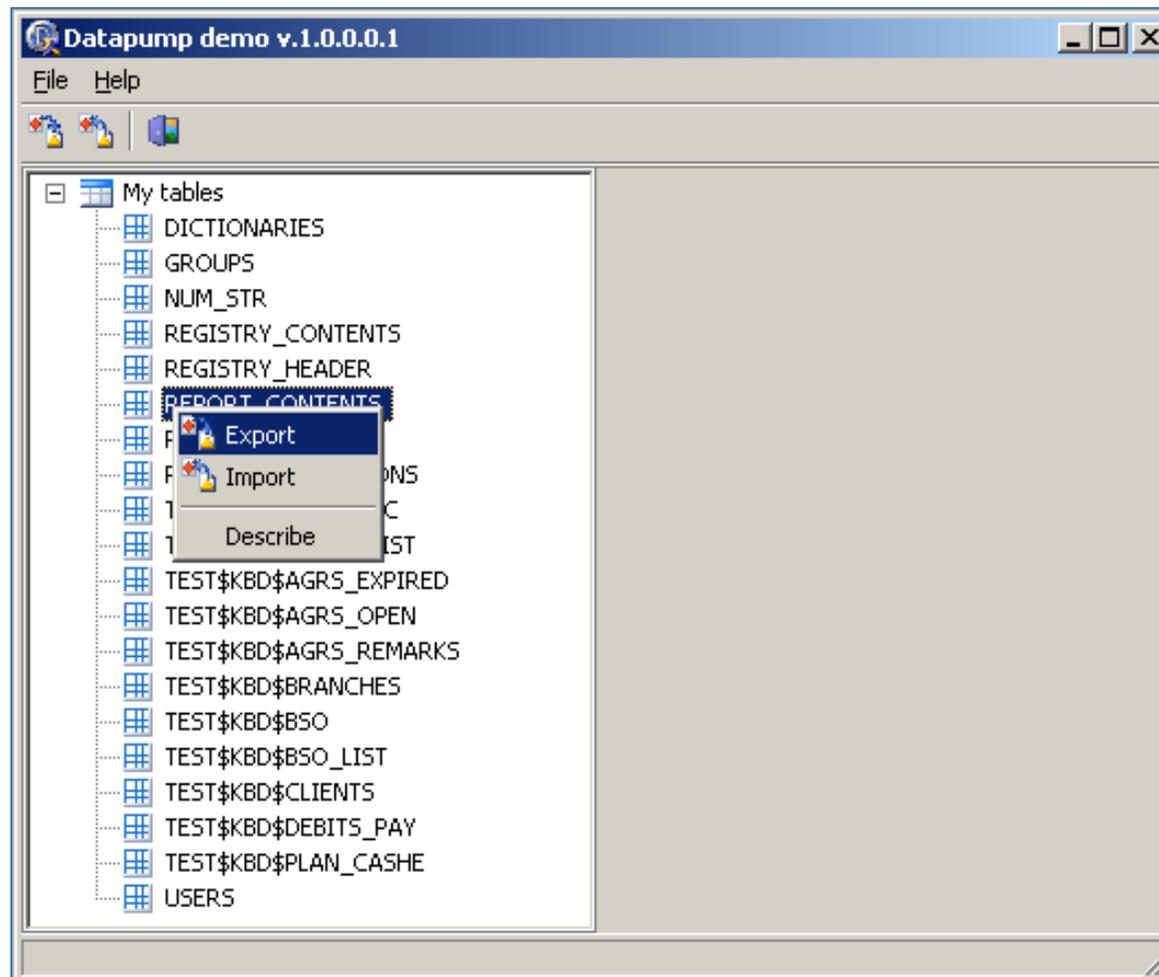
# Демонстрация

## *Встраивание экспорта в приложение*

- Окружение разработки:
  - CodeGear RAD Studio 2007
  - Язык программирования: Object Pascal for .NET
  - Платформа: MS .NET 2.0
  - Oracle Client 10.2.0.3.0
  - Интерфейс доступа: Oracle Data Provider for .NET 10.2.0.3.03
- СУБД
  - Oracle Database 10.2.0.3.11 for MS Windows x86 32bit

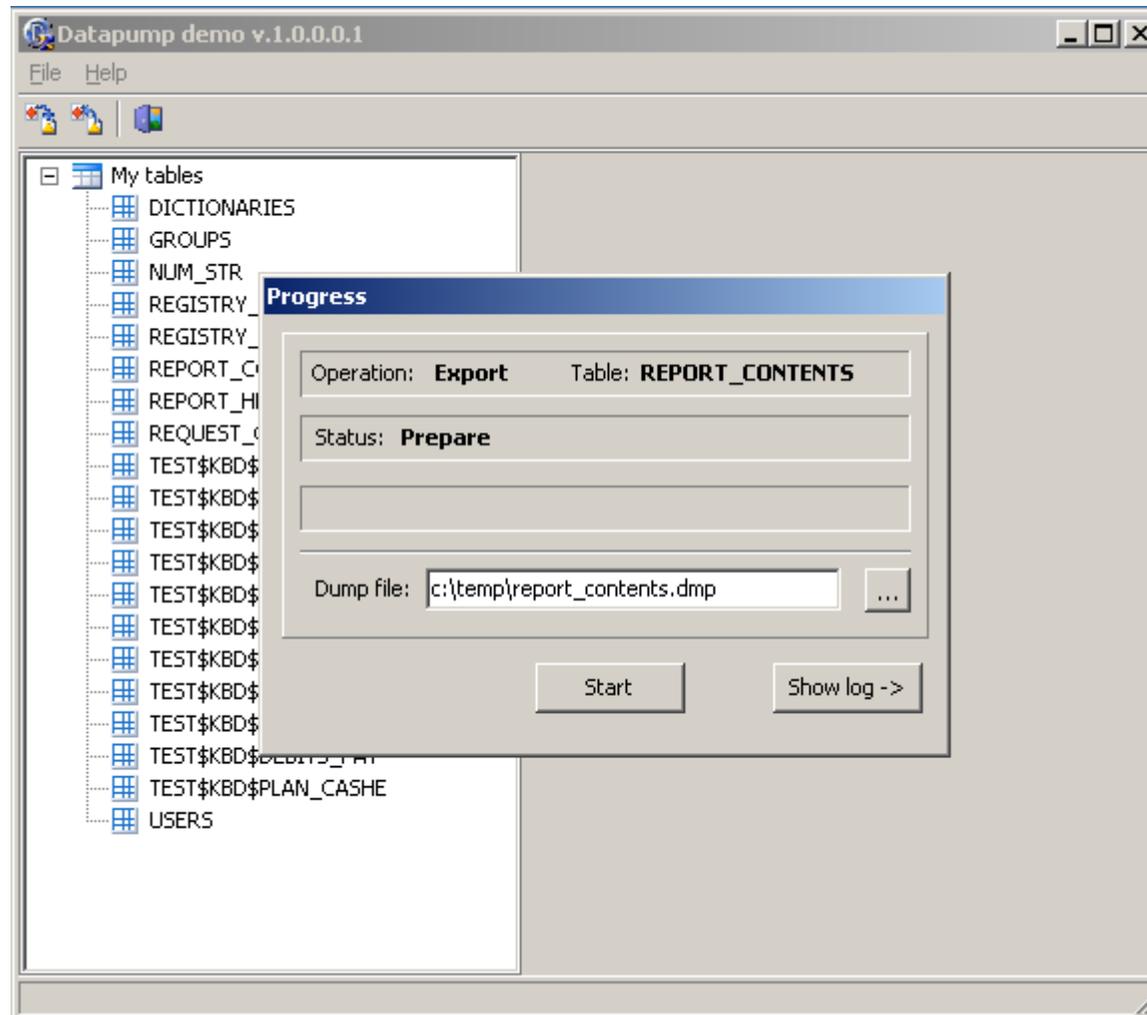
# Демонстрация

## *Выбор таблицы для экспорта*



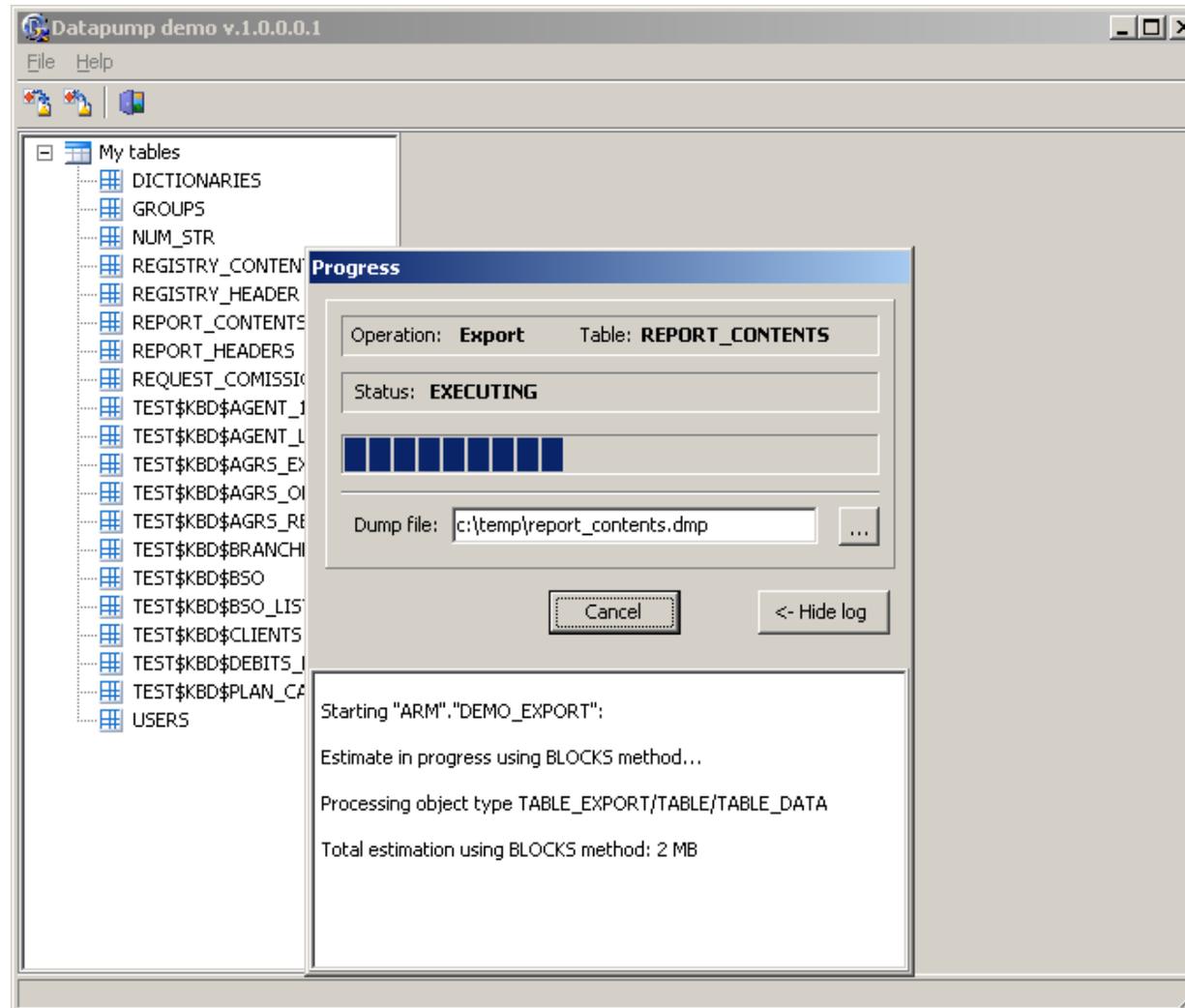
# Демонстрация

*Указываем имя dmp-файла – на клиенте!*



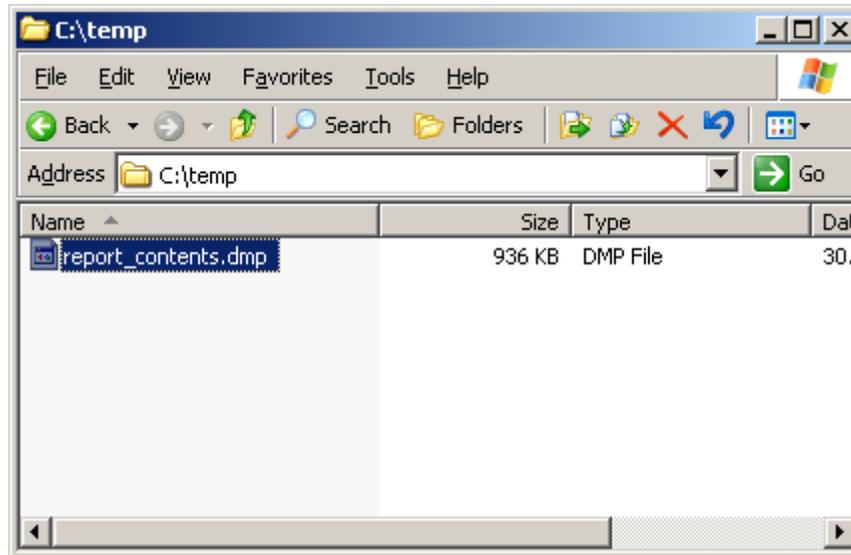
# Демонстрация

## Контроль процесса экспорта



# Демонстрация

*dmp-файл “переброшен” на клиент*



# Демонстрация

## *Как перенести dmp-файл на клиент?*

```
v_xCommand.CommandText := 'select bfilename(:v_xDirectoryName,:v_xServerFileName) as DmpBFile from dual';

v_xCommand.Prepare;
v_xReader := v_xCommand.ExecuteReader;

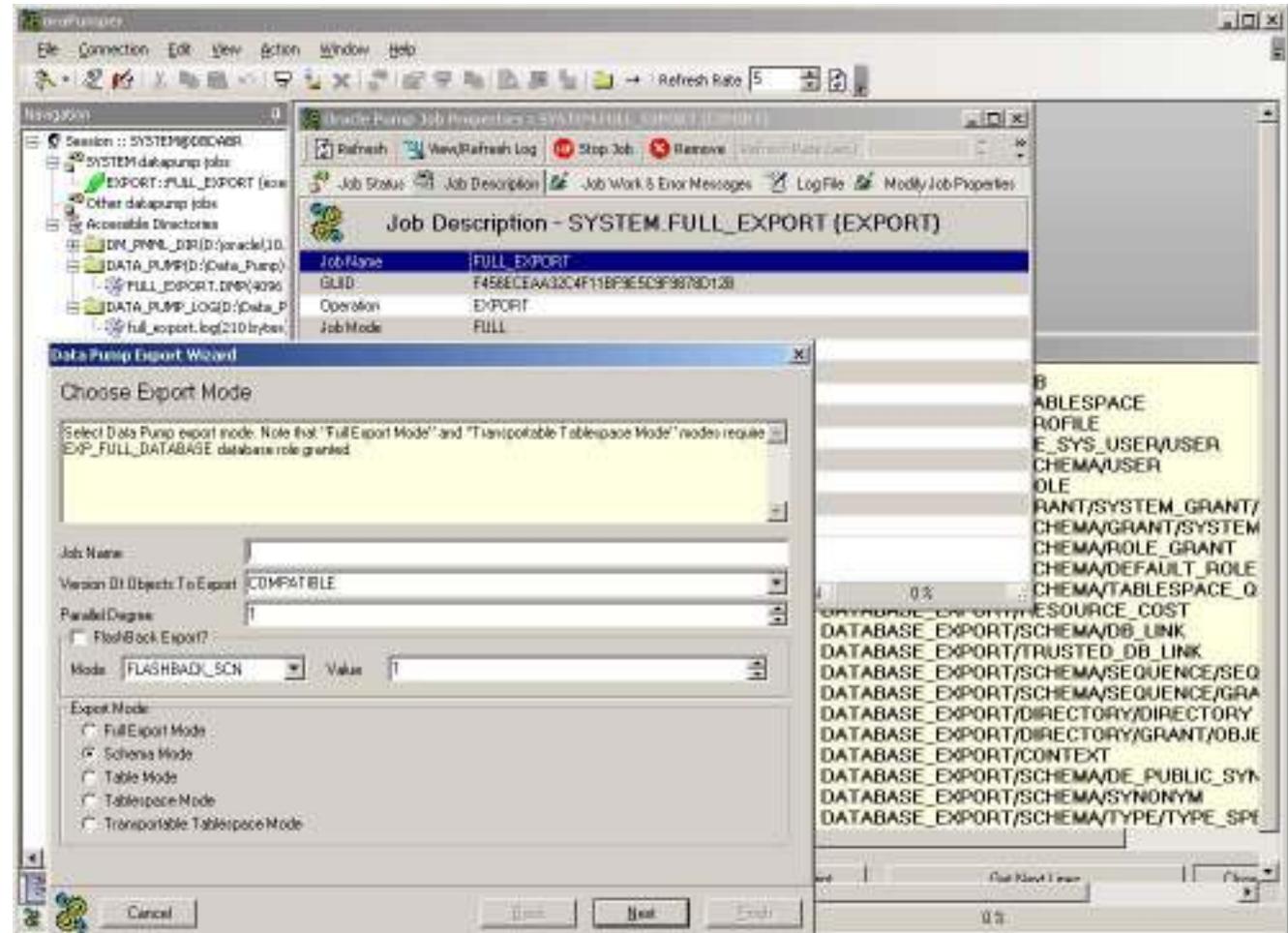
If v_xReader.Read Then
Begin
  v_xMemoryStream := MemoryStream.Create(v_xReader.GetOracleBFile(v_xReader.GetOrdinal('DmpBFile')).Value);

  v_xBuffer := v_xMemoryStream.ToArray();
  v_xBufferStream := MemoryStream.Create(v_xBuffer,0,v_xMemoryStream.Length,true,true);

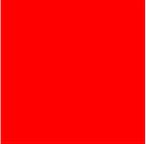
  v_xFileStream := FileStream.Create(v_pClientFullFileName,
                                     System.IO.FileMode.OpenOrCreate);
  v_xWriter := BinaryWriter.Create(v_xFileStream);
  v_xWriter.Write(v_xBufferStream.GetBuffer,0,v_xBufferStream.length);
```

- Обрабатываем dmp-файл как BFile BLOB
- Сохраняем на клиенте в файловый поток (FileStream)
- После удаляем dmp-файл на сервере с помощью UTL\_FILE.REMOVE

# Готовый инструмент для Data Pump *oraPumper*



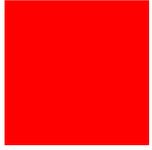
- <http://www.spviewer.com/orapumper.html>



# Oracle Dataump

## *Заключение*

- Развитая технология экспорта/импорта для СУБД Oracle Database
- Высокое быстродействие и масштабируемость
- Мощный API для разработчиков
- Много продвинутых возможностей



*Q & A*



**ORACLE®**

Игорь Мельников  
Старший консультант Oracle СНГ

Email : [Igor.Melnikov@oracle.com](mailto:Igor.Melnikov@oracle.com)

Phone : +7 (495) 641 14 00

Direct: +7 (495) 641 14 42

Mobile: +7 (915) 205 26 27