Database setup for GATE

DATABASE SETUP FOR GATE **1**

1. INTRODUCTION **3**

2. BEFORE YOU BEGIN **5**

3. DATABASE INSTALLATION INSTRUCTIONS **6**

3.1. ORACLE - BASIC SETUP FOR 8I 6
3.2. ORACLE - ADVANCED SETUP FOR 8I 10
3.3. ORACLE - BASIC SETUP FOR 9I 14
3.4. ORACLE - ADVANCED SETUP FOR 9I 17
3.5. POSTGRESQL - SETUP 20
3.6. POSTGRESQL - ADVANCED SETUP 22

4. ORACLE - GATE SCHEMA SETUP **24**

5. ORACLE - FREQUENTLY ASKED QUESTIONS **26**

5.1. WHERE DO I FIND THE ANSWER TO THE ULTIMATE QUESTION OF LIFE, THE UNIVERSE, AND ORACLE? 26
5.2. WHAT IS THE CHARACTER SET OF MY ORACLE INSTANCE? 26
5.3. HOW TO STARTUP/SHUTDOWN 8I INSTANCE 26
5.4. HOW TO STARTUP/SHUTDOWN 9I INSTANCE 27
5.5. HOW TO CREATE A NET8 SERVICE DESCRIPTION FOR THE DATABASE 28
5.6. HOW TO ANALYSE DATABASE STATISTICS 28
5.7. HOW TO SET ORACLE ENVIRONMENT VARIABLES 29
5.8. HOW TO CHANGE INSTANCE PARAMETERS 29
5.9. HOW TO EXPORT DATA FROM THE DATABASE 30
5.10. HOW TO IMPORT DATA INTO THE DATABASE 33
5.11. HOW TO CREATE COMPRESSED EXPORT FILE 37

6. ORACLE - NOTES FOR DEVELOPERS **38**

6.1. DATABASE CONNECTIONS 38
6.2. USE OF BIND VARIABLES 38
6.3. ROW FETCHING 38
6.4. BULK UPDATES 39
6.5. STATEMENT BATCHING 39
6.6. STATEMENT CACHING 39

APPENDIX A **40**

LIST OF SUPPORTED PLATFORMS AND DATABASE VERSIONS 40

APPENDIX B **41**

ORACLE - SAMPLE EXPORT LOG FILE 41

APPENDIX C **43**

ORACLE - SAMPLE EXPORT PARAMETER FILE 43

APPENDIX D **44**

ORACLE - SAMPLE IMPORT LOG FILE 44

APPENDIX E **45**

ORACLE - SAMPLE IMPORT PARAMETER FILE 45

1. Introduction

This document contains the steps that should be performed in order to have a database support for GATE as well as some instructions for developers. At present GATE supports the following databases to be used as repository for GATE data:

- Oracle **8i**
- Oracle **9i**
- PostgreSQL **7.2**[1]

For most of the supported databases there are two types of setup - **basic** and **advanced**.

The *basic* setup is intended for sites that will not store extensive amounts of data in the database, or for sites that do not have a dedicated Oracle/Postgres instance[2] or have no sufficient privileges to perform the steps of the advanced setup. The basic setup does not demand a dedicated database instance for GATE, it may use an already existing one but this will result in degraded performance and reduced flexibility.

The *advanced* setup will give you the best performance when using GATE and this is the *recommended* type of setup. The advanced setup assumes that a dedicated instance is available for use only by GATE applications, which gives you maximum performance and flexibility.

Note: For any kind of setup, if a database administrator is responsible for the Oracle/Postgres server that will be used with GATE, then all the steps should be reviewed and approved by him/her. The steps and settings we propose are just a recommendation and could be changed to best fit your Oracle/Postgres configuration.

Depending on the task you want to perform, the following chapters should be read:

- configure Oracle8i persistence for GATE:
  - one of {Oracle - Basic setup for 8i, Oracle - Advanced setup for 8i}
  - Oracle - GATE schema setup
  - Oracle - Frequently Asked Questions
- configure Oracle9i persistence for GATE:
  - one of {Oracle - Basic Setup for 9I , Oracle - Advanced Setup for 9I}

[1] Implementation inprogress
[2] *Instance* is intended to mean a *database* (PostgreSQL) or *database instance* (Oracle)

- Oracle - GATE schema setup
- Oracle - Frequently Asked Questions

- configure PostgreSQL persistence for GATE:
  - XXXXXXXXXXXXX
  - XXXXXXXXXXXXX
  -
- make development related to the database persistence:

2. Before you begin

GATE will run on any Oracle8i, Oracle9I or PostgreSQL 7.2 platform including Solaris, Linux and Windows (see the list of supported platforms and database versions in Appendix A).

Note that we assume that you have already successfully installed the Oracle/Postgres server for your platform. Please **do not** contact the GATE team for questions related to Oracle/Postgres installation.

In case of problems related to the installation of the database server please consult the documentation located at:

- (Oracle) : MetaLink - *http://metalink.oracle.com* and TechNet - *http://technet.oracle.com*
- (PostgreSQL) : *http://techdocs.postgresql.org*

3. Database Installation instructions

3.1. Oracle - Basic setup for 8i

NOTE1: this is the setup for **8i** instance, if you have Oracle9i then you should refer to the relevant sections - Oracle - Basic Setup for 9I

NOTE2: the recommended type of setup for 8i is Oracle - Advanced *setup* for 8i

The *basic* setup gives you the choice to either use an already existing Oracle8I (with UTF8 character set) instance with GATE and this instance be used by other applications too, or create a new instance to be used only with GATE but do not perform the advanced setup.

Prerequisites:

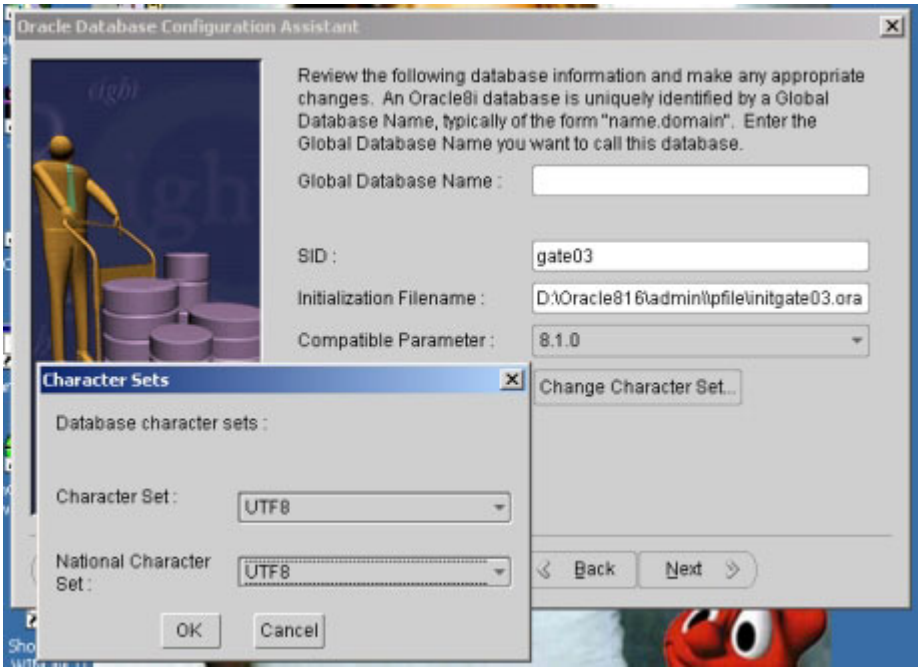
- Existing 8I (Standard Edition or Enterprise Edition) instance with UTF8 character set (or a newly created instance as explained in the following steps). **Note** that you may reuse an existing instance only if it was created with UTF8 character set. If unsure about the character set then see the FAQ (What is the character set of my Oracle instance?)

Follow the steps:

1. *[optional]* (If you intend to resue an existing UTF8 database, then you should skip this step)
   Create a database instance

   - At database creation time make sure that the database character set is set to UTF8
   - The database block size is set to 8KB (unless you have a special reason for different block size)

Screenshot for database character setup:



2. *[optional]* Modify instance parameters. Edit the configuration file for the instance - init*SID*.ora

(*SID* is the Service ID of the newly created database or the database you will reuse)

The init*SID*.ora file usually resides in $ORACLE_HOME/admin/*SID*/pfile/.
Don't forget to **save a copy** of the original file before editing, just in case something goes wrong.

Modify the configuration file so that:

- *open_cursors* is set to at least 900
- *shared_pool_size* is about 30MB
- *db_block_buffers* is at least 5000 (but so that *db_block_buffers*db_block_size* is less than 30% of the physical memory)
- *processes* is sufficient to handle the number of concurrent users you expect (100 is usually ok)
- *log_buffer* is at least 512K, or better 1MB
- *timed_statistics* = false
- *compatible* = 8.1.0
- *sort_area_size* is 1MB

NOTE: All of the values above should be set in **bytes** in the configuration file

3. *[optional]* If you have performed the actions in **step 2** then the Oracle instance must be restarted (see FAQ - How to startup/shutdown 8i).

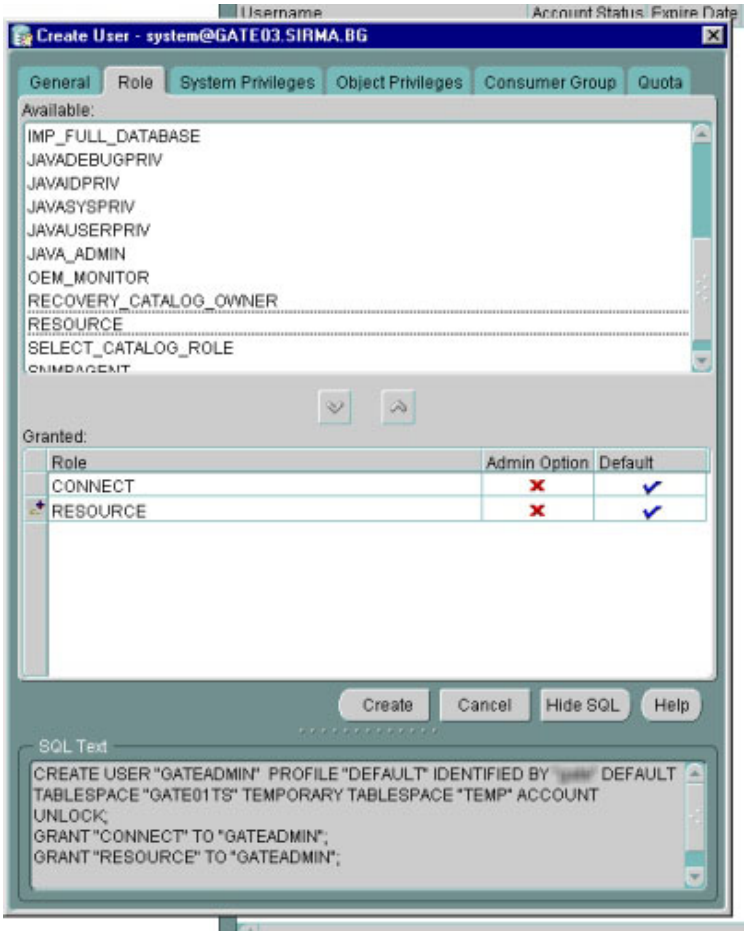
4. Make sure that the number of rollback segments is at least 4 (see Storage --> Rollback Segments info from the DBA Studio, or ask the database administrator)

5. Login in DBA studio (or your favourite tool) as *SYSTEM* (or another privileged user)

6. Create a new **user** GATEADMIN
   Make sure that:

   - In the *General* tab of the user creation dialog make sure the default tablespace for the user **is not** *SYSTEM* but the tablespace that you intend to use for GATE. If necessary create one (see advanced setup for creating a GATEDATA01 tablespace - below)
   - In the *Role* tab, the user is granted the *CONNECT* and *RESOURCE* roles
   - In the *System Privileges* tab, there are **no** system privileges granted

See screenshot for user creation dialog:



7. Create a new **role** GATE_USER_ROLE. Make sure that:

   - There are **no** predefined roles granted to this role
   - The only two system privileges granted to this role are *CREATE SESSION* and *ALTER SESSION*

8. Create a new **user** GATEUSER. Make sure that:

   - The default tablespace for the user **is not** *SYSTEM* but the tablespace that you set for the GATEADMIN user
   - The **only** role granted to this user is *GATE_USER_ROLE* previously created
   - There are **no** system privileges granted to this user.

9. Create a Net8 service on every *client machine* that will access the database with GATE (see FAQ - How to create a Net8 service description for the database)

3.2. Oracle - Advanced setup for 8i

NOTE: this is the setup for **8i** instance, if you have Oracle9i then you should refer to the relevant sections - Oracle - Advanced Setup for 9I

The *advanced* setup for GATE assumes that there is a database instance that will be used only for GATE applications and thus it will be configured in a way that gives maximum performance and flexibility

Follow the steps:

1. Create a new database (use Database Configuration Assistant).

   At database creation time make sure that:

   - The database character set is UTF8 (character set cannot be changed once the database is created) , see - Screenshot for database character setup:
   - database block size is 8192 (unless you have a special reason for different block size)

It is recommended that you do **not** put the instance in archive log mode unless you have a DBA responsible for the database. The performance and management overhead for archiving can be avoided for GATE. A good export/import policy can be used for backup and recovery with the lowest management/performance overhead at the price of reduced safety.

It is also recommended that you use "**dedicated server mode**" rather than "shared server" because of the administration complexities associated with the latter.

2. [optional] Edit the configuration file init*SID*.ora [*SID* is the Service ID of the newly created database). The init*SID*.ora file usually resides in $ORACLE_HOME/admin/*SID*/pfile/.

   Don't forget to **save a copy** of the original file before editing, just in case something goes wrong. Make sure that:

   - *open_cursors* is set to at least 900
   - *shared_pool_size* is about 30MB
   - *db_block_buffers* is at least 5000 (but so that *db_block_buffers*db_block_size* is less than 30% of the physical memory)
   - *processes* is sufficient to handle the number of concurrent users you expect (100 is usually ok)

- *log_buffer* is at least 512K, or better 1MB
- *timed_statistics* = false
- *compatible* = 8.1.0
- *sort_area_size* is 1MB

NOTE: All of the values above should be set in **bytes** in the configuration file

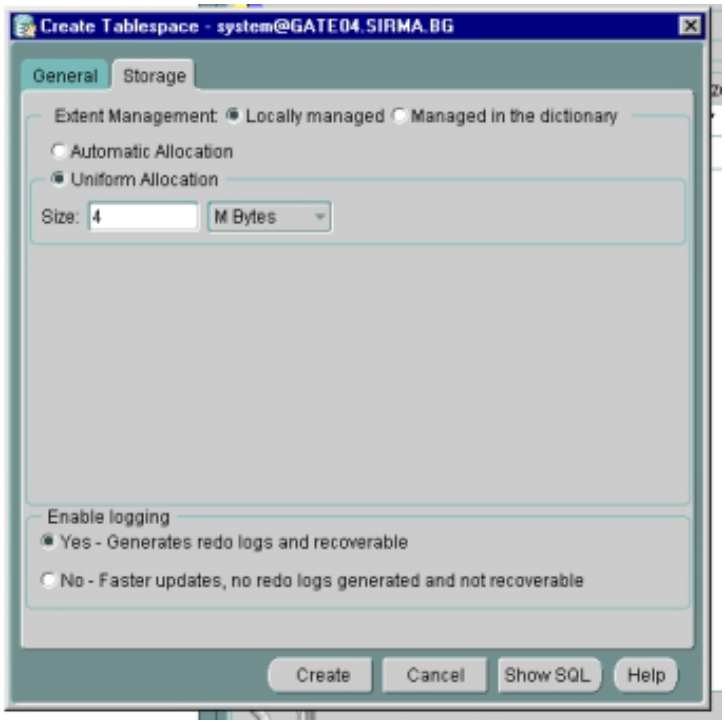
3. [optional] If you have performed the actions in **step 2** then the Oracle instance must be restarted (see FAQ - How to startup/shutdown 8i).

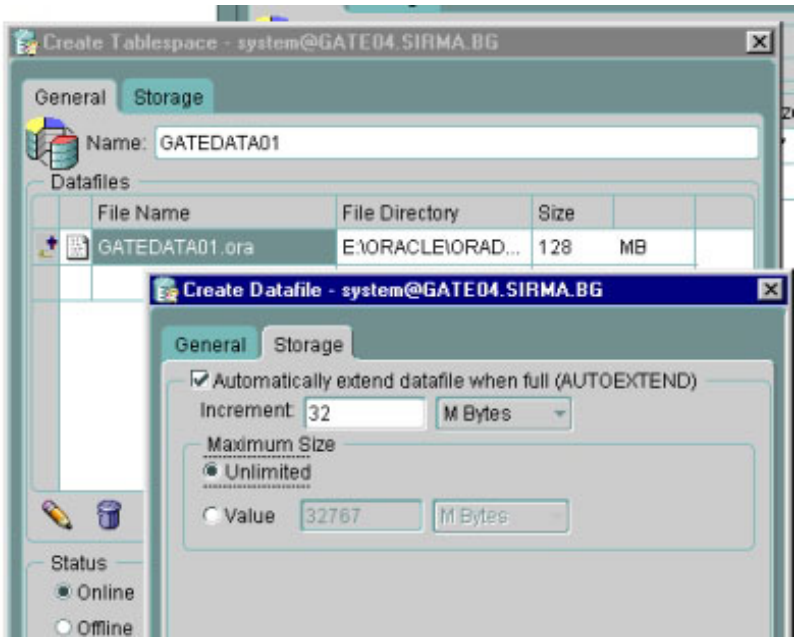
4. Login in DBA studio (or your favourite tool) as *SYSTEM* (or another privileged user)

5. Create a new tablespace called *GATEDATA01* (for data). Make sure that:

   - In the *Storage* tab "uniform allocation" is the chosen allocation method
   - In the *Storage* tab "locally managed" is the chosen extent management method
   - 4MB is the extent size
   - the datafiles of the tablaspace are set so that *AUTOEXTEND* is **on** and the *increment* is sufficient (32MB should be ok)

   See screenshot for tablespace and datafile setup:





6. Create a new tablespace called *GATEINDEX01* (for indexes). The physical settings for the index tablespace should be similar to GATEDATA01 except that the extent size should be smaller - 256KB is usually ok

7. Create a new tablespace called *GATELOB01* (for large objects). The physical settings for the index tablespace should be similar to GATEINDEX01

8. Create a new user GATEADMIN. Make sure that:

- In the *General* tab of the user creation dialog, the default tablespace for the user **is not** *SYSTEM* but *GATEDATA01*
- In the *Role* tab of the user creation dialog, the user is granted the *CONNECT* and *RESOURCE* roles - see screenshot for user creation dialog
- In the *System Privileges* tab of the user creation dialog, there are **no** system privileges granted

9. Create a new role GATE_USER_ROLE. Make sure that:

    - There are **no** predefined roles granted to this role
    - The **only** two system privileges granted to this role are *CREATE SESSION* andt *ALTER SESSION*

10. Create a new user GATEUSER. Make sure that:
    - In the *General* tab of the user creation dialog, the default tablespace for the user **is not** *SYSTEM* but *GATEDATA01*
    - In the *Role* tab of the user creation dialog, the **only** role granted to this user is *GATE_USER_ROLE* previously created
    - In the *System Privileges* tab of the user creation dialog, there are **no** system privileges granted to this user.

11. Create a Net8 service on every machine accessing the new database (see FAQ - How to create a Net8 service description for the database)

3.3. Oracle - Basic Setup for 9I

NOTE1: this is the setup for **9i** instance, if you have Oracle8i then you should refer to the relevant sections - Oracle - Basic setup for 8i

NOTE2: the recommended type of setup for 9i is Oracle - Advanced Setup for 9I

The *basic* setup gives you the choice to either use an already existing Oracle9I (with UTF8 character set) instance with GATE and this instance be used by other applications too, or create a new instance to be used only with GATE but do not perform the advanced setup.

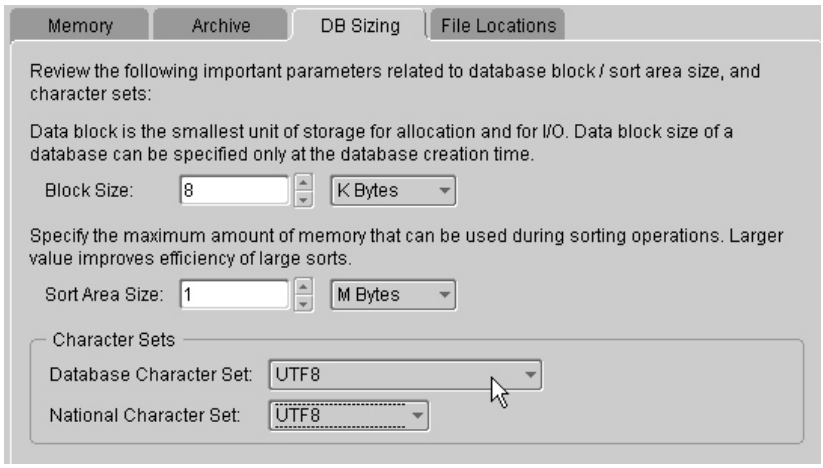
Prerequisites:

- Existing Oracle 9I (Standard Edition or Enterprise Edition) instance with UTF8 character set (or a newly created instance as explained in the following steps). **Note** that you may reuse an existing instance only if it was created with UTF8 character set. If unsure about the character set then see the FAQ (What is the character set of my Oracle instance?)

Follow the steps:

1. *[optional]* Create a new database instance (you may skip this step if you will reuse existing UTF8 instance). At database creation time make sure that:
    - features such as *Label Security*, *Ultra Search*, *Oracle Spatial*, *Oracle OLAP Services*, *Example Schemas*, *Oracle JVM* and *Oracle InterMedia* are disabled since GATE does not need them (the last two are availablefrom the "Additional Database configurations..." panel)
    - The database character set is UTF8
    - database block size is 8192 bytes (unless you have a special reason for different block size)
    - Redo Log files are properly sized (we recommend a size of at least 50MB)
    - *compatible* is set to 9.0.0 (or the latest number for your instance)
    - *db_cache_advice* is set to OFF
    - *db_cache_size* is at least 30MB
    - *java_pool_size* is set to the minimum allowed (30MB for 9.0.0)
    - *log_buffer* is around 1MB
    - *open_cursors* is at least 900
    - *processes* is at least 150
    - *shared_pool_size* is around 30MB
    - *sort_area_size* is at least 1MB
    - *sort_area_retained_size* equals sort_area_size

Screenshot with 9I database settings:



2. *[optional]* Login in Oracle Enterprise Manager Console (or your favourite tool) as *SYSTEM* (or another privileged user)

3. Create a new user GATEADMIN.
   Make sure that:

   - In the *General* tab of the user creation dialog make sure the default tablespace for the user **is not** *SYSTEM* but the tablespace that you intend to use for GATE. If necessary create one (see advanced setup for creating a GATEDATA01 tablespace - above)
   - In the *Role* tab, the user is granted the *CONNECT* and *RESOURCE* roles
   - In the *System Privileges* tab, there are **no** system privileges granted

4. Create a new **role** GATE_USER_ROLE. Make sure that:

   - There are **no** predefined roles granted to this role
   - The only two system privileges granted to this role are *CREATE SESSION* and *ALTER SESSION*

5. Create a new **user** GATEUSER. Make sure that:

   - The default tablespace for the user **is not** *SYSTEM* but the tablespace that you set for the GATEADMIN user
   - The **only** role granted to this user is *GATE_USER_ROLE* previously created
   - There are **no** system privileges granted to this user.

6. Create a Net8 service on every *client machine* that will access the database with GATE (see FAQ - How to create a Net8 service description for the database)

3.4. Oracle - Advanced Setup for 9I

NOTE: this is the setup for **9i** instance, if you have Oracle8i then you should refer to the relevant sections - Oracle - Advanced setup for 8i

The *advanced* setup for GATE assumes that there is a database instance that will be used only for GATE applications and thus it will be configured in a way that gives maximum performance and flexibility

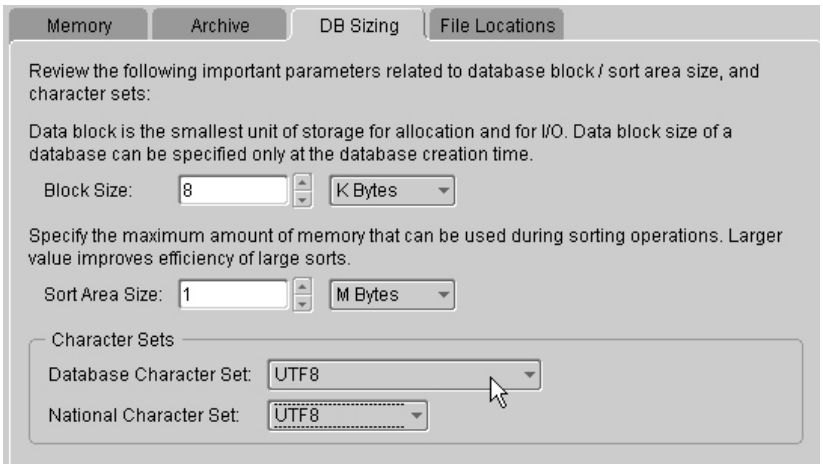
This section contains instructions for:

- Creating the database instance for GATE
- Setting important instance parameters for optimal performance
- Creating tablespaces
- Creating Oracle users
- Creating the database schema

Follow the steps:

1. *[optional]* Create a new database instance (you may skip this step if you will reuse existing UTF8 instance). At database creation time make sure that:
    - features such as *Label Security*, *Ultra Search*, *Oracle Spatial*, *Oracle OLAP Services*, *Example Schemas*, *Oracle JVM* and *Oracle InterMedia* are disabled since GATE does not need them (the last two are available from the "Additional Database configurations..." panel)
    - The database character set is UTF8
    - database block size is 8192 bytes (unless you have a special reason for different block size)
    - Redo Log files are properly sized (we recommend a size of at least 50MB)
    - *compatible* is set to at least 9.0.0
    - *db_cache_advice* is set to OFF
    - *db_cache_size* is at least 30MB
    - *java_pool_size* is set to the minimum allowed (30MB for 9.0.0)
    - *log_buffer* is around 1MB
    - *open_cursors* is at least 900
    - *processes* is at least 150
    - *shared_pool_size* is around 30MB
    - *sort_area_size* is at least 2MB
    - *sort_area_retained_size* equals sort_area_size

Screenshot with 9I database settings:



Parameters that will additionally improve performance are:

- set *cursor_space_for_time* to TRUE
- since GATE is not an critical OLTP application, you can disable features such as "fast-start recovery", so you should set *fast_start_mttr_target* to zero
- eliminate interval checkpoints by setting *log_checkpoint_interval* to zero
- eliminate time-based checkpoints by setting *log_checkpoint_timeout* to zero
- it is recommended that you do **not** put the instance in archive log mode unless you have a DBA responsible for the database. The performance and management overhead for archiving can be avoided for GATE. A good export/import policy can be used for backup and recovery with the lowest management/performance overhead at the price of reduced safety.
- It is also recommended that you use "**dedicated server mode**" rather than "shared server" because of the administration complexities associated with the latter.
- Set *session_cached_cursors* to around 100
- If using automated undo management then reducing *undo_retention* to around 300 (default is 900) will relieve undo overhead without impact n the GATE system.

2. Login in Oracle Enterprise Manager Console (or your favourite tool) as *SYSTEM* (or another privileged user)

3. Create a new tablespace called *GATEDATA01* (for data). Make sure that:

- In the *Storage* tab "uniform allocation" is the chosen allocation method
- In the *Storage* tab "locally managed" is the chosen extent management method
- 4MB is the extent size
- the datafiles of the tablaspace are set so that *AUTOEXTEND* is **on** and the *increment* is sufficient (32MB should be ok)

4. Create a new tablespace called *GATEINDEX01* (for indexes). The physical settings for the index tablespace should be similar to GATEDATA01 except that the extent size should be smaller - 256KB is usually ok

5. Create a new tablespace called *GATELOB01* (for large objects). The physical settings for the index tablespace should be the same as for GATEINDEX01

6. Create a new user GATEADMIN.
   Make sure that:

   - In the *General* tab of the user creation dialog make sure the default tablespace for the user **is not** *SYSTEM* but *GATEDATA01*
   - In the *Role* tab, the user is granted the *CONNECT* and *RESOURCE* roles
   - In the *System Privileges* tab, there are **no** system privileges granted

7. Create a new **role** GATE_USER_ROLE. Make sure that:

   - There are **no** predefined roles granted to this role
   - The only two system privileges granted to this role are *CREATE SESSION* and *ALTER SESSION*

8. Create a new **user** GATEUSER. Make sure that:

   - The default tablespace for the user **is not** *SYSTEM* but *GATEDATA01*
   - The **only** role granted to this user is *GATE_USER_ROLE* previously created
   - There are **no** system privileges granted to this user.

9. Create a Net8 service on every *client machine* that will access the database with GATE (see FAQ - How to create a Net8 service description for the database)

3.5. PostgreSQL - Setup

Prerequisites:

- Postgres backend installed and running on the server
- Postgres libraries installed on the server
- Postgres tools available: *createdb, createuser, createlang* on the server or on the client machines
- Database cluster (catalog cluster) was initialized with *initdb* (see "Creating a database cluster" chapter from the "PostgreSQL Administration Guide")
-

NOTE: In the instructions below we refer with $PGDATA to the directory where the Postgres databases reside. The exact location depends on your installation, e.g. `/var/lib/pgsql/data`

After you have successfully installed the PostgreSQL server (version 7.2+) you need to follow the steps:

1. Login into the server which hosts the database as the Postgres super user (usually called *postgres*).

2. create a new database with UTF8 encoding:

   ```
   $ createdb -E UNICODE DBNAME
   ```

   …where *DBNAME* is the name for the new database. Make sure that the output of the command is:

   ```
   $ CREATE DATABASE
   ```

   …which means that the database was successfully created.

3. create a new Postgres user (**not** OS user) called GATEADMIN:

   ```
   $ createuser --encrypted --no-createdb --no-adduser gateadmin
   ```

   Make sure the output of the command is "CREATE USER" (which means the user was successfully created). The "`encrypted` " flag means that the newly created user's password will be kept encrypted in the database. The "`no-createdb`" and "`no-adduser`" flags will forbid the new user to create new databases and users respectively (the GATEDAMIN user does not need such privileges). You can change any of above settings since they won't affect GATE anyways.

4. create a new Postgres user (**not** OS user) called GATEUSER in the same way you created the GATEADMIN user:

```
$ createuser --encrypted --no-createdb --no-adduser gateuser
```

5. create a new Postgres group (**not** OS user) called GATE_USER_GROUP:

   - start *psql* (the Postgres client tool) and connect to the *DBNAME* database created in [2] (in this example it is GATE09)

     ```
     # psql gate09
     ```

   - issue the following SQL statement to create the Postgres group:

     ```
     CREATE GROUP gate_user_group;
     ```

     (make sure the output from the execution is "CREATE GROUP")

   - add the GATEUSER user to the GATE_USER_GROUP by issuing the statement:

     ```
     ALTER GROUP gate_user_group ADD USER gateuser;
     ```

     (make sure the output from the execution is "ALTER GROUP")

6. Add PL/pgSQL support to the database. Issue the command:

   ```
   $ createlang plpgsql DBNAME
   ```

   …where DBNAME is the name of the database created in [2]

7. Set TCP/IP support for the Postgres server.

   By default the server process (called *postmaster*) is configured so that it is not accepting TCP/IP client connections. In order to change the default behaviour you should edit the PostgreSQL configuration file `$PGDATA/postgresql.conf`. Set the following parameter:

   tcpip_socket = true

   (default is false). Alternatively in the script that starts the postmaster process make sure you pass the "*-i*" parameter to the *postmaster* process.

8. By default the PostgreSQL server will allow connections to the database only from local users (i.e. processes on the same machine).

   To change this behaviour you should edit the Postgres access control file located in `$PGDATA/pg_hba.conf`. Read the detailed instructions in the beginning of the file.

   The file has format like:

   ```
   # TYPE DBNAME IP_ADDRESS ADDRESS_MASK AUTH_TYPE [AUTH_ARGUMENT]
   ```

   Put a configuration line for each computer/network that will access the database. An example may look like:

   ```
   host  gate09   192.168.128.42 255.255.255.255   md5
   ```

   …which will allow the host 192.168.128.42 to access the GATE09 instance with user passwords being checked against the *pg_shadow* table (remember the "encrypted" parameter in [3]) and passwords encrypted while being sent over the network.

9. Change other important settings in the *postgresql.conf* file:

   shared_buffers = 4096 # number of 8KB pages for cache, default is 32
   sort_mem = 2048 # sort memory in *bytes*

   …this will give you 32MB of buffer cache and 2MB of sort area size. Note that you may additionally increase these values (if the server has enough free memory).

   max_connections =
   wal_buffers =
   checkpoint_timeout =
   commit_delay =

   NOTE: the default server settings (256kb of buffer cache and ?kb of sort size) are totally insufficient for normal operation.

10. dsdsdsds

3.6. PostgreSQL - Advanced Setup

Fgfgf

- Fsync to false

- WAL
- Move log files to diff device
- SHMEM
- Log rotation

-

4. Oracle - GATE schema setup

This section describes the steps necessary for creating the GATE database schema for Oracle, once you have created the database instance. This step can be performed on any client machine that has Oracle client installation or on the Oracle server.

Prerequisites:

- Oracle database instance properly set according to on of 3.1 - 3.4
- Oracle sql tools - sqlplus (resides at $ORACLE_HOME/bin/)
- Net8 description that allows this machine to connect to the Oracle database (step "Create a Net8 service" from sections 3.1 - 3.4)

Follow the steps:

1. Verify that you have a directory $GATE/build/persist/Oracle
   The contents should be:

   $GATE/build/persist/Oracle/advanced:

   - alterIndex.sql
   - createDB.sql
   - createDBDev.sql
   - createIndex.sql
   - createSequence.sql
   - createTable.sql
   - createTriggers.sql
   - createType.sql
   - createView.sql
   - grants.sql
   - initData.sql

   $GATE/build/persist/Oracle/basic:

   - createDB.sql
   - createDBDev.sql
   - createIndex.sql
   - createSequence.sql
   - createTable.sql
   - createTriggers.sql
   - createType.sql
   - createView.sql
   - grants.sql
   - initData.sql

$GATE/build/persist/Oracle/misc:

- migration.sql

$GATE/build/persist/Oracle/packages:

- error.spc
- persist.bdy
- persist.spc
- security.bdy
- security.spc
- test.bdy
- test.spc

2. Depending on the setup that you are performing (*basic* or *advanced*) change the current directory to either *$GATE/build/persist/Oracle/basic* or *$GATE/build/persist/Oracle/advanced*

3. Execute the command:

```
# sqlplus gateadmin/pass@service_name @createDB.sql
```

…where:

- *pass* is the password for the GATEADMIN user (step "Create a new user GATEADMIN " from sections 3.1 - 3.4)
- *service_name* is the Net8 Service name with which you describe the database (step "Create a Net8 service" from sections 3.1 - 3.4)

Make sure that *sqlplus* is in your path (it resides in $ORACLE_HOME/bin/)

4. Check the output of the previous execution - h
Jhjh

5. Oracle - Frequently Asked Questions

5.1. Where do I find the Answer to the Ultimate Question of Life, the Universe, and Oracle?

The Answer to the Ultimate Question of Life, the Universe, and Oracle is **42**.

If this does not solve your problem then you should RTFM shipped with your Oracle server, or better browse the online docs for Tahiti (http://tahiti.oracle.com) or TechNet (http://technet.oracle.com)

5.2. What is the character set of my Oracle instance?

Follow the steps:

1. Login into the Oracle instance as SYSTEM (or another privileged user)
2. Execute the query:

```
select *
from v$nls_parameters
where parameter='NLS_CHARACTERSET';
```

Note: GATE requires UTF8 character set of the database

5.3. How to startup/shutdown 8i instance

- **starting up an 8i instance** (command line)

1. One the machine where the Oracle server resides, set the ORACLE_SID environment variable to the Service ID of the instance you want to start:

   # ORACLE_SID=*SID*

2. start the *svrmgrl* from the Oracle server:

   # svrmgrl

3. Execute the commands:

   ```
   SVRMGRL> CONNECT INTERNAL
   ```
   (you'll be asked for password)

   ```
   SVRMGRL> STARTUP
   ```
   (wait for the "Database opened" message)

- **shutting down an 8i instance** (command line)

1. One the machine where the Oracle server resides, set the ORACLE_SID environment variable to the Service ID of the instance you want to shutdown:

   # ORACLE_SID=*SID*

2. start the *svrmgrl* from the Oracle server:

   # svrmgrl

3. Execute the commands:

   SVRMGRL> CONNECT INTERNAL
   (you'll be asked for password)

   SVRMGRL> SHUTDOWN IMMEDIATE
   (wait for the "ORACLE instance shut down" message)

5.4. How to startup/shutdown 9i instance

- **starting up an 9i instance** (command line)

1. One the machine where the Oracle server resides, set the ORACLE_SID environment variable to the Service ID of the instance you want to start:

   # ORACLE_SID=SID

2. start the *sqlplus* from the Oracle server:

   # sqlplus /nolog

3. Connect to Oracle as a SYSDBA user:

   SQLPLUS> CONNECT *user/password* AS SYSDBA
   (where "user" is a user with SYSDBA role granted, usually SYS)

4. Execute:

   SLPLUS> STARTUP
   (wait for the "Database opened" message)

- **shutting down an 9i instance** (command line)

1. One the machine where the Oracle server resides, set the ORACLE_SID environment variable to the Service ID of the instance you want to shutdown:

   # ORACLE_SID=SID

2. start the *sqlplus* from the Oracle server:

```
# sqlplus
```

3. Connect to Oracle as a SYSDBA user:

   `SQLPLUS> CONNECT user/password AS SYSDBA`
   (where "user" is a user with SYSDBA role granted, usually SYS)

4. Execute:

   `SLPLUS> SHUTDOWN IMMEDIATE`
   (wait for the "Database opened" message)

5.5. How to create a Net8 service description for the database

1. start Net8 Assistant and select Local/Service Naming fron the "Net8 Configuration" tree

2. create a new net8 service so that:
   - the protocol is TCP/IP
   - the hostname is the name or IP address of the Oracle server
   - "Oracle 8i Service Name" is the SID of the database you are accessing (the SID was specified at database creation time)

3. test the service with a user existing in the database (SYSTEM, GATEADMIN, etc). If you get an error message then some of the steps described above was not performed correctly. For 8i read the "Net8 Administrator's Guide", chapter 6 "Configuring Naming Methods" and for 9i read the "Oracle9i Net Services Administrator's Guide", chapter 9, "Configuring Naming Methods" (both are available at http://tahiti.oracle.com/)

5.6. How to analyse database statistics

Analyzing the database regularly is important for the good performance of the Oracle server. You should perform analysis whenever 10-20% of the data in the database is changed. Since such estimate is difficult, it is better to pefrom analysis on a daily/weekly basis.

To analyse **8i** database:

Execute this code from any SQL tool connected to the 8i GATE instance:

```
begin
   dbms_utility.analyze_schema('GATEADMIN','COMPUTE');
end;
/
```

To analyse **9i** database:

Execute this code from any SQL tool connected to the 9i GATE instance:

```
begin
dbms_stats.gather_schema_stats(ownname => 'GATEADMIN',cascade=>true);
end;
/
```

NOTE: There is certain performance overhead for analysis so it is recommended that you do not perform this task when other users are using the GATE database.

5.7. How to set Oracle environment variables

The **NLS_LANG** variable should be set properly for in OS environment of the GATE application.

Add the following lines to the script that starts GATE or to the user profile file if this will not interfere with other applications (if unsure then put it in the scripts that start GATE):

- for Windows:

  set NLS_LANG=AMERICAN_AMERICA.UTF8

- for bash:

  NLS_LANG=AMERICAN_AMERICA.UTF8; export NLS_LANG

5.8. How to change instance parameters

The instance parameters are handled differently in 8i and 9i.

Oracle8i relies on a text file containing the instance settings (init*SID*.ora). It is enough to edit the parameter value in this file and restart the instance in order to see the effect of the change.

Oracle9i relies on a binary file which cannot be edited and changes in the instance settings are performed in a different manner.

The following example shows how to change a parameter for 8i and 9i.:

- For Oracle8I follow the steps:

1. Edit the instance configuration file. It usually resides in $ORACLE_HOME/admin/*SID*/pfile/ where SID is the Service ID for the oracle instance (unique for the server)
2. Set the parameter value (for example *sort_area_size= 2097152*)
3. Restart the instance so that the new settings will take effect - see How to startup/shutdown 8i

- For 9i follow the steps:

1. Login as *SYSTEM* (or another privileged user) into SQLPlus (or any SQL tool)
2. Issue the command:

```
ALTER SYSTEM SET parameter = value SCOPE=BOTH;
```

This will update the system and the system configuration file. No restart is required if the parameter is dynamic.

If the parameter is not dynamic then you will get error issuing the command above - replace `SCOPE=BOTH` with `SCOPE=SPFILE` which will modify only the instance configuration file. Then u need to restart the instance so that the change can take effect

5.9. How to export data from the database

You can use either the OEM Console/DBA Studio or the command line utilities to make export

- for OEM Console/DBA Studio:

1. make sure NLS_LANG is properly set (see How to set Oracle environment variables)

2. start the Export Wizard (Tools --> Database Wizards --> Data Management --> Export)

3. in the "export file" dialog set the proper location for the export file

   NOTE export files are usually twice bigger than the data in the database, so make sure you have enough space on that device/drive or compress the data during the export

4. in the "export type" dialog choose *User*

5. in the "user selection" dialog choose GATEADMIN.
   Deselect any other selected users

6. in the "associated objects" dialog you **should** check:
   - *rows of table data*,
   - *grants associated with database objects*
   - *constraints on tables*

   you could optionally check *indexes on tables* but this is not necessary, since you may always run the index creation scripts to recreate them, so the export file will be smaller if you do **not** export indexes

7. **optionally** from the "General" pane of the "Advanced" setup dialog you may change these settings:
   - check *Compute optimizer statistics when data is imported* (recommended). If statistics are not updated then a substantial degradation in performance will be observed
   - check *Direct Path*

8. **optionally** from the "Tuning" pane of the "Advanced" setup dialog you may change these settings:
   - check *Do **not** export read-consistent view of the data* If the GATE instance is not used by other users in the moment then you may safely export the data without taking care for a read-consistent view of the data. This will speed up export and will relieve the undo management overhead
   - change the value for *Overwrite default buffer size*. The larger the buffer the faster the export will be. A buffer of 2MB is usually enough.

9. When the export is done, look in the log file (usually called export.log and placed in the directory where the export file is dumped). One of the first line should look like:

   Export done in UTF8 character set and UTF8 NCHAR character set

   ...this means that the export peoperly performed without character set conversion. If NLS_LANG was improperly set then a character set conversion may be necessary which is not the desired behaviour.

   Also make sure that the last line of the export log looks like:

   Export terminated successfully without warnings.

...this means that no problems were encountered during the export

A sample export log is available here - Oracle - Sample export log file

- for command line utilities:

  1. make sure NLS_LANG is properly set

  2. set the ORACLE_SID environment variable to the Service ID of the GATE instance

     ```
     ORACLE_SID=SID
     ```

  3. locate the Oracle export utility - it is usually *$ORACLE/bin/exp*

  4. create a parameter file for the export session. See the *exp* help (*exp help=y*) for the proper syntax. The parameter file should contain the following information:
     - location and name of the export file
     - location and name of the log file
     - the owner of the schema that will be exported
     - grants option - should grants on objects be exported with the objects *(recommendation = default = **yes**)*
     - index option - should indexes be exported *(recommendation = **no**, default = "yes")*
     - rows option - should indexes be exported *(mandatory - **yes**)*
     - constraints option - should constraints be exported *(mandatory - **yes**)*
     - (optionally) should statistics becomputed/estimated upon import *(recommendation = **compute** , default = "estimate")*
     - (optionally) should *direct path* be used for export *(recommendation = **yes** , default = "no")*
     - (optionally) export buffer size *(recommendation - 2MB)*
     - (optionally) should extents be merged during import *(recommendation = **no**, default = "yes")*

     A sample parameter file (available here - Oracle - Sample export parameter file) looks like:

     ```
     FILE=/tmp/exp-21-feb-02.dmp
     LOG=/tmp/export.log
     OWNER=GATEADMIN
     GRANTS=y
     INDEXES=n
     ```

```
ROWS=y
CONSTRAINTS=y
STATISTICS=compute
DIRECT=y
BUFFER=2097152
COMPRESS=n
```

5. start export by running the command:

   ```
   exp gateadmin parfile=/path_to_parfile/parfile_name
   ```

6. You will be asked for password, enter the GATEADMIN password

7. The export will start, look in the log file (its location was specified in the parameter file). One of the first line should look like:

   ```
   Export done in UTF8 character set and UTF8 NCHAR
   character set
   ```

   ...this means that the export peoperly performed without character set conversion. If NLS_LANG was improperly set then a character set conversion may be necesssary which is not the desired behaviour.

   Also make sure that the last line of the export log looks like:

   ```
   Export terminated successfully without warnings.
   ```

   ...this means that no problems were encountered during the export

   A sample export log is available here - Oracle - Sample export log file

   NOTE: If you encounter error such as:

   ```
   EXP-00008: ORACLE error 6550 encountered
   PLS-00201: identifier 'SYS.LT_EXPORT_PKG' must be
   declared
   EXP-00083: The previous problem occurred when calling
   SYS.LT_EXPORT_PKG.schema_info_exp
   ```

   ...then (Oracle bug:1828996) you should connect as SYS to the instance, and then execute the command:

   ```
   GRANT EXECUTE ON sys.lt_export_pkg TO gateadmin;
   ```

5.10. How to import data into the database

You can use either the OEM Console/DBA Studio or the command line utilities to import data

- for OEM Console/DBA Studio:

    1. make sure NLS_LANG is properly set (see How to set Oracle environment variables)

    2. start the Import Wizard (Tools --> Database Wizards --> Data Management --> Import)

    3. From the "Import File" pane choose the location of the export file and also choose "Manually specify what you want to import"

    4. From the "Import Type" pane choose "User" and specify "GATEADMIN"

    5. From the "User Mapping" pane choose "No"

    6. From the "Associated objects" pane check all boxes, and from the "Advanced" dialog choose:

        - "Overwrite default buffer size" (Tunung) - set it to 2 MB
        - "Execute Analyze statements" (Statistics)

    7. Proceed with the import and check the import log file for error messages. A sample import log is available here - Oracle - Sample import log file

    8. If the export file does not contain indexes, then you should recreate them

        - Depending on the database setup you have performed (basic/advanced) change the current directory to either *$GATE/build/persist/Oracle/basic* or *$GATE/build/persist/Oracle/advanced*

        - Execute the command:

            ```
            # sqlplus gateadmin/pass@service_name @createIndex.sql
            ```

            …where:

            - *pass* is the password for the GATEADMIN user (step "Create a new user GATEADMIN " from sections 3.1 - 3.4)
            - *service_name* is the Net8 Service name with which you describe the database (step "Create a Net8 service" from sections 3.1 - 3.4)

Make sure that *sqlplus* is in your path (it resides in $ORACLE_HOME/bin/)

- If performing **advanced** setup then execute the command:

```
# sqlplus gateadmin/pass@service_name @alterIndex.sql
```

- for command line utilities:

1. make sure NLS_LANG is properly set
2. set the ORACLE_SID environment variable to the Service ID of the GATE instance

   ```
   ORACLE_SID=SID
   ```

3. locate the Oracle import utility - it is usually *$ORACLE/bin/imp*
4. create a parameter file for the export session. See the *imp* help (*imp help=y*) for the proper syntax. The parameter file should contain the following information:
   - location and name of the import file
   - location and name of the log file
   - grants option - should grants on objects be exported with the objects *(recommendation = default = **yes**)*
   - index option - should indexes be exported *(recommendation = **no**, default = "yes")*
   - rows option - should indexes be exported *(mandatory - **yes**)*
   - constraints option - should constraints be exported *(mandatory - **yes**)*
   - (optionally) should precompiled statistics be imported *(recommendation = default = **always**)*
   - (optionally) export buffer size *(recommendation - 2MB)*

A sample parameter file (available here -Oracle - Sample import log file) looks like:

```
FILE=/tmp/exp-21-feb-02.dmp
LOG=/tmp/export.log
GRANTS=y
INDEXES=n
ROWS=y
CONSTRAINTS=y
```

```
STATISTICS=always
BUFFER=2097152
```

5. start import by running the command:

```
imp gateadmin parfile=/path_to_parfile/parfile_name
```

6. You will be asked for password, enter the GATEADMIN password

7. The import will start, look in the log file (its location was specified in the parameter file). One of the first line should look like:

```
Export file created by EXPORT:V09.00.01 via direct path
Import done in UTF8 character set and UTF8 NCHAR
character set
```

...this means that the import will be properly performed without character set conversion. If NLS_LANG was improperly set then a character set conversion may be necessary which is not the desired behaviour.

Also make sure that the last line of the export log looks like:

```
Import terminated successfully without warnings.
```

...this means that no problems were encountered during the export

A sample import log is available here - Oracle - Sample import log file

8. Recreate indexes:

- Depending on the database setup you have performed (basic/advanced) change the current directory to either *$GATE/build/persist/Oracle/basic* or *$GATE/build/persist/Oracle/advanced*

- Execute the command:

```
# sqlplus gateadmin/pass@service_name @createIndex.sql
```

…where:

  - *pass* is the password for the GATEADMIN user (step "Create a new user GATEADMIN " from sections 3.1 - 3.4)
  - *service_name* is the Net8 Service name with which you describe the database (step "Create a Net8 service" from sections 3.1 - 3.4)

Make sure that *sqlplus* is in your path (it resides in $ORACLE_HOME/bin/)

- If performing **advanced** setup then execute the command:

```
# sqlplus gateadmin/pass@service_name @alterIndex.sql
```

5.11. How to create compressed export file

Because the export files are quite huge for big databases, it is recommended that you compress the export file during its generation from the data in the database.

To create a compressed export file on **Unix** platforms one should follow the steps:

- Create a named pipe
- Start a data compression program that reads from the pipe
- Start export from the database into the named pipe

Please note that this method should be thoroughly tested before being implemented.

Follow the steps:

1. Create a unix named pipe:

```
os> mknod /tmp/exp_pipe p
```

2. Start a compressing program that reads from the pipe and writes to a file

```
gzip < exp_pipe > gate.exp.gz &
```

Note that the compression program is started in the background.
You can also use any other compressing tool such as *zip* or *compress*

3. Start the database export - Follow the steps from the "command line utilities" section of How to export data from the database

Make sure the output file specified in the export parameter file is the named pipe created in step 1 (i.e. replace `FILE=/tmp/exp-21-feb-02.dmp` with `FILE=/tmp/exp_pipe`)

6. Oracle - Notes for Developers

This section contains details about the implementation of the Oracle specific persistence and recommendations for developers performing extension of the current functionality

6.1. Database connections

Since there is a certain overhead with opening connection to the database, the current implementation opens one dedicated database connection to the database for each OracleDataStore. In other words, when you `open()` a datastore, a connection to the database is made, which will be closed only when you call `close()` for the datastore. Meanwhile all requests from this datastore to the database will use the single datastore connection.

Since requests using the same JDBC connection are serialized (i.e. if several threads send concurrently JDBC requests for this connection they will **not** be executed concurrently) it is recommended that the same datastore is **not** shared between several threads. If the threads are long lived and they will perform many requests then they should open their own datastores (although to the same database) so that they will have dedicated database connections.

6.2. Use of bind variables

It is recommended that all requests (selects, updates, etc) use bind variables instead of literals. This improves the server performance (better reuse of already executed statements)

It is also recommended that stored procedures are used whenever possible (better performance and reuse)

6.3. Row fetching

It is recommended that the fetch size is not used with the default value (10 for Oracle). If the fetch size is too small (e.g. 10) and you retrieve many rows from the database (e.g. 1000) then you will make redundant round-trips between the client and the server for each fetch (e.g. 100 roundtrips in this case).

The current implementation uses fetch sizes from 30 to 100 depending on the estimate for the number rows that will be retrieved from the database. Do **not** set fetch size too high.

Use the JDBC 2.0 call `setFetchSize()` of the Oracle specific call `setRowPrefetch()` to change the default behaviour.

6.4. Bulk updates

It is recommended that JDBC arrays are used whenever possible so that a single request may be made to the database instead of sending many requests. For example if inserting many entities in the database it is recommended that the values are packed in JDBC arrays and send with a single database request, instead of making a separate request for each entity.

An example for using JDBC arrays in GATE is storing the features of annotation/resource - the set of features are packed in JDBC array, then a single call to the database is made, then the array with features is unpacked by the PL/SQL code in the database and finally individual features are inserted as distinct rows in the proper table.

This approach minimizes the round-trips between the client and the server. The alternative, which is making a new database request to store each individual feature of the annotation/resource, will induce a certain performance overhead because of the increased number of roundtrips and statements being executed by the server.

6.5. Statement Batching

It is recommended that whenever possible batch updates/inserts are performed instead of series of individual ones. Use the `addBatch()/executeBatch()` JDBC functionality to group statements in a batch, and execute the whole bunch together, with a single database call.

Note that the Oracle JDBC drivers do **not** support batching for callable statements, so this approach could not be used at present.

6.6. Statement Caching

It is recommended that statement caching provided by the Oracle driver is used. To employ statement caching call the Oracle specific method `setStmtCacheSize()` after opening the JDBC connection.

The current implementation sets the statement cache to 50 elements.

Appendix A

List of supported platforms and database versions

Database name	version	server platform	JDBC driver
Oracle	8.1.6	Windows NT	8.1.7 thin, 8.1.7 OCI, 9.0.1 thin
Oracle	8.1.7	Windows NT	8.1.7 thin, 8.1.7 OCI, 9.0.1 thin
Oracle	9.0.1	Windows 2000	9.0.1 thin, 9.0.1 OCI, 8.1.7 OCI
Oracle	9.0.1	Linux	9.0.1 thin, 9.0.1 OCI

Appendix B

Oracle - Sample export log file

```
Connected to: Oracle9i Enterprise Edition Release 9.0.1.0.0 -
Production
With the Partitioning option
JServer Release 9.0.1.0.0 - Production
Export done in UTF8 character set and UTF8 NCHAR character set
Note: indexes on tables will not be exported

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user GATEADMIN
. exporting object type definitions for user GATEADMIN
About to export GATEADMIN's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export GATEADMIN's tables via Direct Path ...
. . exporting table      PLAN_TABLE    3 rows exported
. . exporting table      T_ANNOTATION    1832176 rows exported
. . exporting table      T_ANNOTATION_TYPE          81 rows exported
. . exporting table      T_ANNOT_SET       5420 rows exported
. . exporting table      T_AS_ANNOTATION    1879634 rows exported
. . exporting table      T_CORPUS          25 rows exported
. . exporting table      T_CORPUS_DOCUMENT       2703 rows exported
. . exporting table      T_DOCUMENT       2708 rows exported
Table T_DOC_CONTENT will be exported in conventional path.
. . exporting table      T_DOC_CONTENT       2708 rows exported
. . exporting table      T_DOC_ENCODING           1 rows exported
Table T_FEATURE will be exported in conventional path.
. . exporting table      T_FEATURE    6909058 rows exported
. . exporting table      T_FEATURE_KEY          60 rows exported
. . exporting table      T_GROUP           3 rows exported
. . exporting table      T_LANG_RESOURCE       2733 rows exported
. . exporting table      T_LR_TYPE           2 rows exported
. . exporting table      T_NODE    1511415 rows exported
. . exporting table      T_PARAMETER           2 rows exported
. . exporting table      T_USER          6 rows exported
. . exporting table      T_USER_GROUP           8 rows exported
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
```

```
. exporting post-schema procedural objects and actions
. exporting statistics

Export terminated successfully without warnings.
```

Appendix C

Oracle - Sample export parameter file

```
FILE=/tmp/exp-21-feb-02.dmp
LOG=/tmp/export.log
OWNER=GATEADMIN
GRANTS=y
INDEXES=n
ROWS=y
CONSTRAINTS=y
STATISTICS=compute
DIRECT=y
BUFFER=2097152
COMPRESS=n
```

Appendix D

Oracle - Sample import log file

```
Connected to: Oracle9i Enterprise Edition Release 9.0.1.1.1 -
Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production

Export file created by EXPORT:V09.00.01 via direct path
import done in UTF8 character set and UTF8 NCHAR character set
. importing GATEADMIN's objects into GATEADMIN
. . importing table     "T_ANNOTATION"       7150 rows imported
. . importing table     "T_ANNOTATION_TYPE"         56 rows imported
. . importing table     "T_ANNOT_SET"          16 rows imported
. . importing table     "T_AS_ANNOTATION"       7150 rows imported
. . importing table     "T_CORPUS"           4 rows imported
. . importing table     "T_CORPUS_DOCUMENT"           8 rows imported
. . importing table     "T_DOCUMENT"           8 rows imported
. . importing table     "T_DOC_CONTENT"           8 rows imported
. . importing table     "T_DOC_ENCODING"           1 rows imported
. . importing table     "T_FEATURE"      24719 rows imported
. . importing table     "T_FEATURE_KEY"          49 rows imported
. . importing table     "T_GROUP"           3 rows imported
. . importing table     "T_LANG_RESOURCE"          12 rows imported
. . importing table     "T_LR_TYPE"           2 rows imported
. . importing table     "T_NODE"       6603 rows imported
. . importing table     "T_PARAMETER"           2 rows imported
. . importing table     "T_USER"           6 rows imported
. . importing table     "T_USER_GROUP"           8 rows imported
About to enable constraints...
Import terminated successfully without warnings.
```

Appendix E

Oracle - Sample import parameter file

FILE=/tmp/exp-21-feb-02.dmp
LOG=/tmp/import.log
GRANTS=y
INDEXES=n
ROWS=y
CONSTRAINTS=y
STATISTICS=always
BUFFER=2097152