



Teamcenter Enterprise: Guidelines for Oracle-Setup and DB-Configuration

Volume ITI-MTI-C

Written by
Erich Brauchli

March 2, 2008

Version 9.28

Reflecting Teamcenter[®] Enterprise 2005
(including some old chapters Metaphase 3.2)

Email: erich@brauchli.info

Internet: <http://www.brauchli.info>

(Total 21 pages)

Entire List of Teamcenter Enterprise Documents:

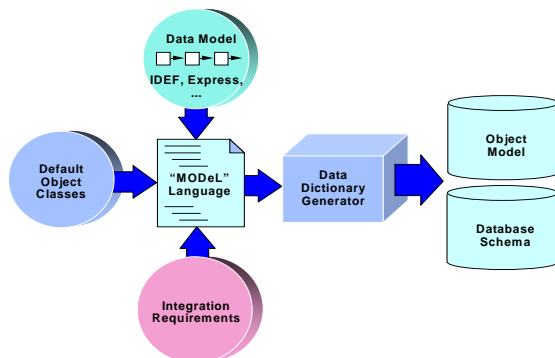
ITI-MTI-A	Guidelines for Planning and Operation
ITI-MTI-B	Guidelines for Setup and Configuration
ITI-MTI-C	Guidelines for Oracle Setup and DB Configuration
ITI-MTI-D	Guidelines for Customizing
ITI-MTI-E	Rules

Content

1	How Teamcenter® Enterprise interacts with Oracle.....	3
2	Installation of Distributed Database	4
2.1	General Remark	4
2.1.1	Oracle	4
2.1.2	Install Distributed System	4
2.2	Install Corporate Server	5
2.2.1	Answers for:.....	5
2.2.2	Result	6
2.2.3	Test.....	6
2.2.4	Installation of Workgroup Server.....	7
2.2.5	Test.....	7
2.3	Add Replication of DB's	7
2.3.1	Goal	7
2.3.2	Procedure	8
2.3.3	Result	8
2.4	Sample Configuration	8
2.4.1	Sample „dbtop.prd“	8
2.4.2	„config.cfg“	9
2.5	Alternatives for DB initialization	9
2.6	Manual Load of Initial Data into Teamcenter Enterprise Databases	10
2.7	Key-Databases	11
3	ORACLE — Database Installation	12
3.1	Oracle Version problems	12
3.2	European Language Support.....	12
3.3	Usage of “DOT-Notation”.....	13
3.4	Database creation (old)	14
3.5	Create a Database with changed Block Size (old).....	14
3.6	Install SqlNet 2 (old)	17
3.6.1	listener.ora	17
3.6.2	sqlnet.ora	18
3.6.3	tnsnames.ora	18
3.7	Get Instances started (old)	19
3.8	Install Users and Tablespaces (Teamcenter Enterprise Databases).....	19
3.8.1	Standard Teamcenter Enterprise Database Creation (old)	19
3.8.2	Alternative shell script “create_user”.....	19
3.8.3	Install Teamcenter Enterprise with one tablespace per DB	20
3.8.4	Install Teamcenter Enterprise with two tablespaces per DB	20
3.9	Other critical ORACLE parameters.....	21

1 How Teamcenter® Enterprise interacts with Oracle

- **A Teamcenter Enterprise “Database” is an Oracle “User”.**
- Teamcenter Enterprise Database Names (Oracle Users) are restricted to 8 characters. They must be case-insensitive unique in an entire Teamcenter Enterprise installation over all servers.
- Even Oracle ignores case in usernames (Teamcenter Enterprise database names), Teamcenter Enterprise requires that they are written within Teamcenter Enterprise always case sensitive as defined in the config editor and stored in the “*dbtop.prd*” file.
- If you use distributed option, then you need SQLnet, even if all databases (Oracle users) are on the same host and in the same Oracle instance, except if you use “DOT-Notation”.
- In the distributed Teamcenter Enterprise instance all databases (Oracle users) must have defined a private **database link** to each of the Teamcenter Enterprise databases. These links must have the same name as the database (Oracle user) they point to. The file `teamcenter Enterprise as6(toe)]T0.0506 T`



The Teamcenter Enterprise Data Dictionary is created out of different resources: Teamcenter Enterprise standard definitions, additional modul definitions and customization definitions by the "MODEL" compiler.

The RDBMS schema is exclusively created and updated out of the Teamcenter Enterprise data dictionary through the Teamcenter Enterprise utility "mschema".

2 Installation of Distributed Database

2.1 General Remark

Today the reasons for using distributed databases may differ from the situation in earlier versions. Today's networks are by far more reliable and efficient than years ago. Thus the DB accesses over distances much less critical, while distribute Vault organization may still be an advantage.

For productive databases use an Oracle DB with a block size of at least 8kB, better 16kB. This parameter can be set only at installation time.

Even thinking about a migration to common platform will probably be easier without distributed and replicated databases.

A reason for distributed architecture (even in one local Oracle instance) may be the size of each database.

The decision about database distributed architecture should be done before starting any installation. It is not easy to change from single DB to distributed architecture nor vice versa.

Remark:

The following chapters were created under Metaphase 3.2 and Oracle 7.3. May be some information is not updated completely to the actual versions.

2.1.1 Oracle

After installing any Teamcenter Enterprise (or if installation fails after or during creation of Oracle Schemas), check carefully the installed **Oracle** and **SqlNet** installations. On any side log in as an Oracle user (Teamcenter DB name) and try to use a database link to any other used Oracle system: try the access to the remote system doing a

```
select * from dba_users @ dblinkname;
```

If this does not work, do not continue with Teamcenter Enterprise! Teamcenter Enterprise's *msqlora* service will login into the Oracle user named in the DDB variable in your config file. All other databases, even local ones, are addressed through the respective database links¹. (More info: see chapter 3.6 Install SqlNet 2)

2.1.2 Install Distributed System

Installing from scratch a distributed database Teamcenter Enterprise system, may be done on a single machine with a single ORACLE instance first. The difference to a single DB installation is mainly, that there is at least one DB (DB user in terms of Oracle) of each type is installed. And a non empty „dbtop.prd“ is generated.

The following sample show how to install a first level system with distribution on a „corporate“ server and a Local Area server

Refer to standard Teamcenter Enterprise Manuals respective to this activity. Important information is in „Installation guide for UNIX and NT“ and in „Teamcenter Enterprise Distributed Database Implementation“.

Assumption:

Two servers: Alpha (as corporate server) and Beta (as Local Area server), both having an Oracle instance installed and a listener running. Use IP-Port 7002.

Alpha will host following DB's:

- *KeyDB*
- *SuDb*
- *AdmAlph*
- *UsrAlph*

Beta will host following DB's:

- *AdmBeta*
- *UsrBeta*

2.2 Install Corporate Server

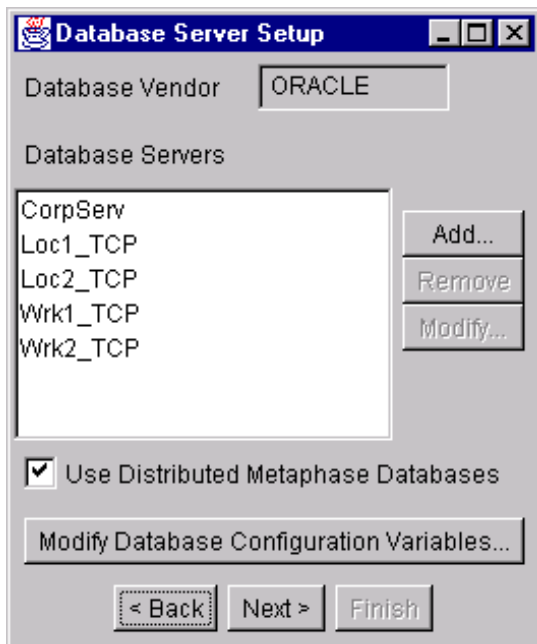
Standard Teamcenter Enterprise installation on server 'Alpha'. Reply to all questions the correct answer.

Remember: A Teamcenter Enterprise Database is an Oracle user.

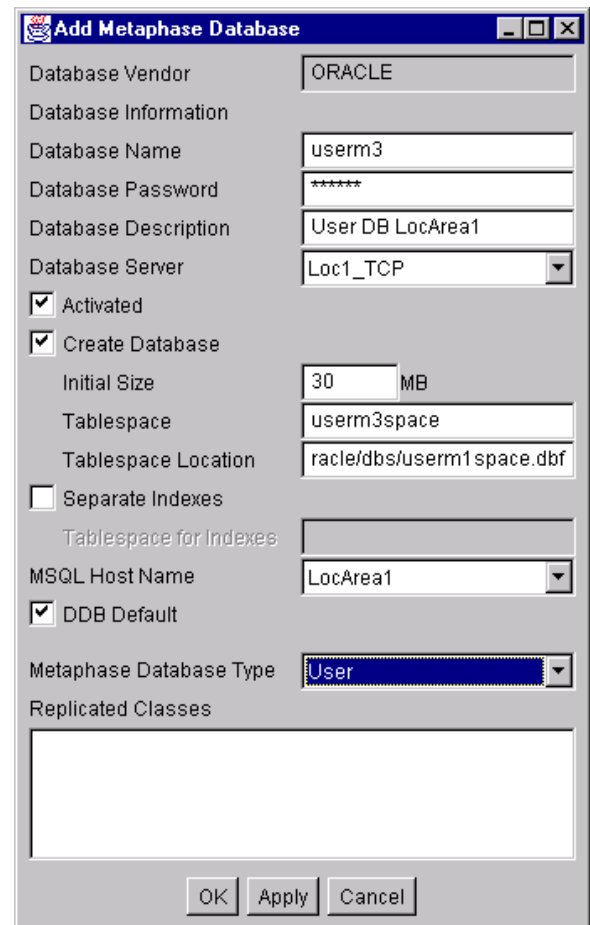
If you have not yet created the Oracle user and Tablespace, then Teamcenter Enterprise will do it for you, but with default size and storage parameters. You have more flexibility in this, when you create user and tablespace beforehand.

2.2.1 Answers for:

- Is this the Corporate Server: **Y**
- IP-Port: [7032] **7032** *(a CR alone will use the default in [brackets])*
- Database Server: **Y**
- Upgrade Installation: **N**



The database have to be defined within the config editor of Teamcenter Enterprise. **The very first decision** to make is, whether you want to use a distributed environment or a single database. After this you have to give correct values for the single database or for each required database.



Do not forget to give the name of the index space, if it is not the same space as for the tables. If you click on the create button, then the installation will drop an eventual already existing Oracle schema and recreate it. If you do not click on the create button, Teamcenter Enterprise assumes, that you have created the Oracle user and table space (with no tables yet).

If you created a data and an index tablespace beforehand, then you must select the "Create database" toggle, fill in the two space names, and then turn off the "Create database" toggle again, so that Teamcenter schema creation may remember the locations for tables and indexes.

2.2.2 Result

The installation routine installs all local databases (if you have created Oracle user and tablespace beforehand, then Teamcenter Enterprise only installs the tables and indexes): KeyDB, OpsAlph, SuDb, UstrAlph. In all databases dblinks are created to all, even to the defined remote one „UrsBeta“. But the remote database is not yet touched at all. This one will be created upon the installation on the workgroup server.

The installation process created a *dbtop.prd* file in *\$MTI_ROOT/meta* to reflect the database topology, which then is used to create the data dictionary (the content of *dbtop.prd* is stored in the dictionary).

Remark:

Earlier Versions of Teamcenter had still a so called LCM database, which was usually replicated like the Admin DB. For performance reasons its content is today in the Admin DB. If your site did not perform the relative migration, then you should do it soon. It solves a lot of Scope and performance problems.

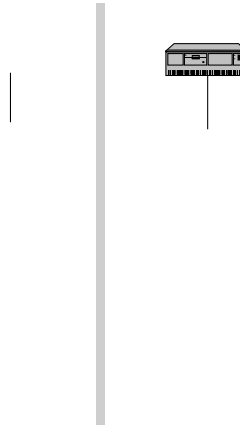
2.2.3 Test

After this installation, try out and do first usage test. It is a good test to try to register a first File System Item on this host.

2.2.4 Installation of Workgroup Server

Before you start the installation on the workgroup server, it is recommended to register that host as a Teamcenter Enterprise host in Teamcenter Enterprise and validate the *super user* on it. Any server must be defined in config editor before hand.

Typical situation of “Alpha” = Corporate Server, “Beta” = WorkGroup server with own DB:



First define the new AdminDB in config editor on corporate host (config editor run always and only there).
Config editor will automatically update dbtop.prd and mtilinks.sql.

2.3.2 Procedure

- a) On both servers stop Teamcenter Enterprise.
- b) Regenerate the data dictionary with "model -b tmti.prd" and push it toh Beta.
- c) On Beta create Oracle user *AdmBeta* and its tablespace.
- d) On Beta call `dbenc AdmBeta/oracle > $MTI_ROOT/config/AdmBeta.pwf`
- e) On Alpha start Teamcenter Enterprise (no users).
- f) Apply mtilinks.sql to all DBs on Alpha ("dblkora -n config/<DB>.pwf -f install/mtilinks.sql").
- g) On Alpha "objdump -d <admindb> -o adm.dat Root".
- h) Create the new DB by "credbora ...".
- i) Apply mtilinks.sql to all DBs on Beta ("dblkora -n config/<DB>.pwf -f install/mtilinks.sql") including the new one.
- j) Start mux and dispatcher on Beta.
- k) On Alpha (or Beta if transferred) load the dump file: "objload -d <newDB> -nrf ad.dat -C100".
- l) Stop and restart Teamcenter Enterprise on Alpha and Beta (with users).

2.3.3 Result

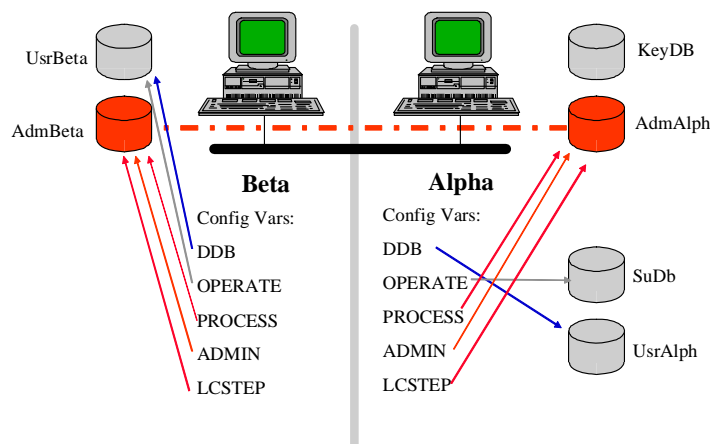


Figure 2.2

2.4 Sample Configuration

Here two files that describe the preceding sample configuration:

2.4.1 Sample „dbtop.prd“

Be aware, that this file is maintained exclusively by config editor:

```

/*
* CORPORATE DATABASES (if Distributed)
*
* Database
* Name          Class          Description
* =====
* keymti        KeyTbl         Key Data
* adminmti      Admin           Administrative Data (R)
* sumti         Root            Corporate Super User Account
* opsmti        Operate        Notification Data (EventQ & Log)
*
* (R) - replicated database

```



```

*/
define database KeyDB ;          /* Key DB */
define database AdmAlpha ;      /* Admin DB Alpha */
define database AdmBeta ;      /* Admin DB Beta */
define database OpsBeta ;      /* Operational DB Beta */
define database SuDb ;         /* User DB for super user(s) */

/*
* USER DATABASES
*
* Database
* Name          Class      Description
* =====
* UsrAlpha      Root       Data of Alpha users and vaults
* UsrBeta       Root       Data of Beta users and vaults
* etc....
*
*/

define database UsrAlpha ;      /* User DB Alpha */
define database UsrBeta ;      /* User DB Beta */

/*
* Replication for Key Database.
*/
define replica set KeyTblRS across KeyDB ;
replicate KeyInfo in KeyTblRS ;

/*
* Replication for Administrative Database.
*/
define replica set AdminRS across AdmAlpha , AdmBeta ;
replicate Admin in AdminRS ;
replicate Process in AdminRS ;
replicate LCStep in AdminRS ;
replicate LCBrch in AdminRS ;

/*
* Connection for all databases.
* (this will create ORACLE database links )
*/
define connection msql to
    KeyDB
    , SuDb
    , AdmAlpha
    , UsrAlpha
    , AdmBeta
    , OpsBeta
    , UsrBeta
    ;

```

2.4.2 „config.cfg“

Do not touch manually. Maintained by config editor.

2.5 Alternatives for DB initialization

Database initialization is usually done by

```
initdb -n <DBname> -s <Hostname> -p <DBname>.pwf
```

If you have specific parameters to set, that are not possible through MODEL statements², you might intend to create your database (Oracle user) manually:

²) Metaphase 2.3 and later allows specifying some of the storage values within MODEL.

- a) Create Oracle user <DBname>.
- b) Create Tablespace(s).
- c) Create sql statements for table and index creation:
- d) in directory \$MTI_ROOT/meta (where is located file *mti.prd*) you call *mschema -g ../mti.prd > gentab.sql* for table generation statements.
- e) in directory \$MTI_ROOT/meta (where is located file *mti.prd*) you call *mschema -G ../mti.prd > genind.sql* for index generation statements.
- f) Edit '*gentab.sql*': replace all ',generic' by 'long'; and what needed.
- g) Edit '*genind.sql*': replace all 'clustered index' by 'index'; and what needed.
- h) Call *sqlplus <DBname>/oracle*

```
alter user <DBname> default tablespace <space for tables>;
commit;
start gentab
alter user <DBname> default tablespace <space for indexes>;
start genind
alter user <DBname> default tablespace <space for tables>;
insert into zdbconfig (OBID,zname,zvakue) values ('00A','dbDb','<DBname>');
insert into zdbconfig (OBID,zname,zvakue) values ('00B','dbSite','<SiteName>');
commit;
exit
```

2.6 Manual Load of Initial Data into Teamcenter Enterprise Databases

Data File	Database	
omfadmdb.dat	---	DO NOT LOAD !
zsuwrksp.dat	---	DO NOT LOAD !
omfinst.dat	admin	is exact sum of omfadmdb and zsuwrksp
imsadmdb.dat	admin	
lcmadmdb.dat	admin	
lcmintdb.dat	admin	
ccmadmdb.dat	admin	
apcadmdb.dat	admin	
psmadmdb.dat	admin	

Note

"*omfinst.dat*" is created at installation time. It is an exact copy of "*omfadmdb.dat*" and "*zsuwrksp.dat*". While "*omfadmdb.dat*" contains standard initial data common for all Teamcenter Enterprise installations, "*zsuwrksp.dat*" is site dependent and contains the necessary information about super user and first host used by the installation. Thus if you want to load these files into another installation, you need to edit "*zsuwrksp.dat*" and then load "*omfadmdb.dat*" and "*zsuwrksp.dat*" both separately instead of "*omfinst.dat*".

All other files need to be loaded in the order the respective modules have been installed. The data will have the same interdependency as the modules themselves.

Sample

```
$ objload -d adminmti -f omfinst.dat -K omfinst.key
$ objload -d keymti -f omfinst.key
$ objload -d adminmti -f lcmadmdb.dat -K lcmadmdb.key
$ objload -d keymti -f lcmadmdb.key
$ objload -d adminmti -f lcmintdb.dat -K lcmintdb.key
$ objload -d keymti -f lcmintdb.key
$ objload -d adminmti -f ccmadmdb.dat -K ccmadmdb.key
$ objload -d keymti -f ccmadmdb.key
$ objload -d adminmti -f psmadmdb.dat -K psmadmdb.key
$ objload -d keymti -f lpsmadmdb.key
```

```
| $ objload -d adminmti -f pfmadmdb.dat -K pfmadmdb.key  
| $ objload -d keymti -f lpfmadmdb.key
```

Distributed Database Environment

Caution: *objload* will load/update only one database. Its replications need to be loaded or updated separately and accordingly. (Key must be generated just once).

Use the “-C 1000” or smaller option for big loads to avoid “rollback segment too small” or “out of memory” problems during *objload*.

Update / Load of replicated database:

```
| $ objload -d adminmti -f omfinst.dat -K omfinst.key -C 1000  
| $ objload -d keymti -f omfinst.key -C 1000  
| $ objload -d admin2 -f omfinst.dat -C 1000  
| $ objload -d admin3 -f omfinst.dat -C 1000  
| $ objload -d admin4 -f omfinst.dat -C 1000
```

Add a replication: Load

```
| $ objdump -d adminmti -t admin.cop  
| $ objload -d admin2 -rnf admin.cop -C 1000
```

2.7 Key-Databases

The key database stores all values needed for uniqueness guarantees. In an environment with several user databases spread over continents, different types of documents (stored in different Oracle tables) the company wants still unique document numbers over all sites and types. Only new revisions of the document or checked out working copies must have the same document number, with different revision or sequence numbers. Thus the functions “*CheckOut*” and “*Revise*” must be able to create a document with same number, but “*Create*” cannot do so.

This beyond standard database capabilities for multiple tables and multiple databases. Teamcenter Enterprise keeps this information in the key database.

Thus the key database is accessed each time you create, delete, revise, checkout or checkin-replace an item that has a key assigned. Generally keys are assigned to Business Items and Admin Items. Customizations may change this. Data Items (generally files) do not have keys assigned and any action on them will not access the key database.

Other unique info is needed for automatic numbering of object (API “*GetUniqueId*”) as well as support of “multiple open revisions”.

Teamcenter Enterprise standard distributed option supports only instances with one single unique Key-Database.

In this standard case no real implications on customizations are to be respected. If you do not crosslink advanced services and ops-databases, there is almost no Teamcenter Enterprise dialog between sites having a full set of databases. Whatever communication is needed, will be done through SQLnet.

We measured a delay of about 3 seconds in a correctly setup environment, if a create document was issued in US, while the key database was in France.

Whenever this is a performance issue, there is a way of using several key databases. But its usage restricts other features:

- ◆ The customization must grant that each site with a separate key DB generates never a same key as any other site (i.e. using a unique prefix).
- ◆ The functions “*Delete*”, “*CheckOut*”, “*CheckIn-Replace*” and “*Revise*” (implicit or explicit) may not be executed on another site than the one on which it was created, or this functions must be deviated to a host of that site. Both must be realized through the customization.
- ◆ Caution is required (in customization) for use of “*GetUniqueId*”.

- ◆ No use of “multiple open revisions” feature.

🌸 Before planing a multi key DB installation you are advised to contact some Teamcenter Enterprise DB specialist.

3 ORACLE — Database Installation

3.1 Oracle Version problems

Caution with Oracle Versions: use the Oracle version validated with the relative Teamcenter version. Teamcenter Enterprise 2005 (even SP1) is validated only with Oracle 9.2. The release bulletin mentioned initially even the Oracle 10i version. This was cancelled later.

Oracle 10 will return scrambled results on some queries. And a way back to version 9.2 does not exist. If unfortunately, you have already upgraded the database to Oracle 10, you need to organize the access through an Oracle 9 client. This is very efficient, but it works. Teamcenter Enterprise 2007 will probably work correctly with Oracle 10.

Your Oracle instance does not need all the recent security patches, which address the WEB access. An Oracle instance containing Teamcenter DB's is **not allowed** for access other than through the Teamcenter Application. Thus: **do just not start the Oracle WEB access services or daemons** for the relative Oracle instances.

3.2 European Language Support

Teamcenter Enterprise supports the full ASCII character set. This means that you can use all 256 characters of the IBM2 character set in texts as well as in display and translate statements of your custom model. This will be displayed correctly in the Neuron Data client of all platforms (with exception of window titles on HP-UX unless you change the HP default character set).

Even on many UNIX stations you do not have the keys for accentuated characters, you can enter them at least through cut and past. Another option is to edit these texts finally on a PC with your standard office keyboard.

But in many installations then you are blocked, since your European characters entered in input fields will come back from the data base changed to something else, depending on your Oracle setup. If these characters come back as a question mark (“?”), then your Oracle is not set up for 8 bit characters.

- Oracle must be installed with character set “*WE8ISO8859P1*”.³

If your European characters come back from the database as just another character (not question mark), then the problem is not on the Oracle side. You need to include a variable for each recipient of Oracle data – thus each Teamcenter Enterprise user: trusted and normal user – in his setup. Best is to include it in the “*pdmsetup*” file.

In csh:

```
| setenv NLS_LANG AMERICAN_AMERICA.WE8ISO8859P1
```

In sh or ksh:

```
| NLS_LANG = AMERICAN_AMERICA.WE8ISO8859P1; export NLS_LANG
```

In Windows define a System environment variable:

```
| Name=NLS_LANG, Value=AMERICAN_AMERICA.WE8ISO8859P1
```

If this is omitted, the system may work correctly a while. But the first time a user or load routine stores a value into an attribute of type “float” (or “PfmFloat”) you get an Oracle error number 1722 “Illegal number”. The reason is very simple: depending on some parameters Oracle installed with some European languages expects a “comma” as decimal point, while Teamcenter delivers a period (“.”).

³) WE8ISO8859P1 : all upper case letters, digits are underlined here.

3.3 Usage of “DOT-Notation”

The use of “DOT-Notation” is recommended in most cases with distributed database schema, especially, if all databases are member of the same ORACLE DB.

To use this feature the variable “MSQL_DOT_NOTATION” to the value “1” in config file. In this case Teamcenter Config Editor will not create the file “mtlinks.mti”, resp. delete it if it exists already in then install subdirectory.

You need to establish zones and DB links between zones⁴. This is done by adding a map “db.cfg” in the Config file. And the add on your own the DB links between zones through ORACLE directly.

The “db.cfg” map looks like:

```
insert "db.cfg" host={"mix_ORACLE"}
#$ All zone 1 DBs
  "zone1" "DB1 DB2 DB3"
#$ All zone 2 DBs
  "zone2" "DB8 DB9 DB10"
;
```

For the ability to switch on and off this feature, it is recommended to start with feature turned off, let Teamcenter create the DB links, then turn on the feature and use one of the contained DBs name as name of the zone. Then the links will fit for both situations. Here a sample:⁵

```
insert "db.cfg" host={"mix_ORACLE"}
#$ All Madrid DBs
  "UsrSpain" "admSpain UsrSpain sudb keydb"
#$ All New York DBs
  "UsrNY" "UsrNY admNY"
;
```

Anyway it requires testing, whether the feature really increases the overall Oracle performance. We have seen cases, where the feature did decrease the performance rather. It may even be different, depending on which DB name is used as zone name.

Effect of DOT-Notation:

Compare some Teamcenter generated SQL statements assuming the DDB variable points to “UsrNY” on US hosts, issued on a US host:

With DOT-Notation

```
select * from admNY.Usr where ...;
select * from keydb.KeyTbl @ UsrSpain where ...;
```

Without DOT-Notation:

```
select * from Usr @ admNY where ...;
select * from KeyTbl @ keydb where ...;
```

For each query containing an “@” sign in the from clause SQLnet will be involved, which means passing an additional process.

Same Teamcenter generated SQL statements assuming the DDB variable points to “UsrSpain” on Spain hosts, issued on a Spain host:

With DOT-Notation

```
select * from admSpain.Usr where ...;
select * from keydb.KeyTbl where ...;
```

⁴) A zone contains all Teamcenter DBs being in the same ORACLE DB, thus Oracle users of same ORACLE DB.

⁵) Lines starting with „#\$“ in the config file represent the comment for the following line, as shown in the config editor.

Without DOT-Notation:

```
select * from Usr @ admSpain where ...;
select * from KeyTbl @ keydb where ...;
```

3.4 Database creation (old)

Remark:

The rest of this chapter was created under Metaphase 3.2 and Oracle 7.3. May be some information is not updated completely to the actual versions. The chapter is left in for information, which eventually may be adapted upon occurrence of problems.

For bigger database the standard Oracle parameters are no good. The global system area becomes too small and the maximum of extents per table is 121, when Block-Size is left on 2k.

Unfortunately this parameter can only be set before the creation of the first database. And procedure to do so have slightly changed since Oracle 7.1: The 'orainst' script does still allow installing an instance without database. But option 2 (to create later a database object) does no more ask you, whether to create a database.

You will even detect that "ORACLE_HOME"-directory may no more be the "HOME"-directory of UNIX user "oracle". In the following sample we have:

```
Oracle's HOME = "/opt/oracle"
ORACLE_HOME = "/opt/oracle/app/oracle/product/7.3.2"
```

3.5 Create a Database with changed Block Size (old)

- Install the Oracle instance and deny the creation of a database. Select modules as required in Teamcenter Enterprise Installation Guide.
- Do not forget to request Oracle to be relinked.
- In \$ORACLE_HOME copy the file 'init.ora' into 'initMTI.ora' (where MTI is the Oracle Instance Name or "SID"). Edit this file: to reflect your instance; define enough rollback segments (4 are ok for most cases). Sample:

```
ifile                = /opt/oracle/app/oracle/product/7.3.2/dbs/configMTI.ora
rollback_segments   = (r01,r02,r03,r04)
#
# $Header: init.ora,v 1.19 1994/09/19 18:49:08 alingelb Exp $
#
# Copyright (c) 1991 by Oracle Corporation
# NAME
#   init.ora
# FUNCTION
# NOTES
#####
# Example INIT.ORA file
#
# This file is provided by Oracle Corporation to help you customize
# your RDBMS installation for your site.  Important system parameters
# are discussed, and example settings given.
#
# Some parameter settings are generic to any size installation.
# For parameters that require different values in different size
# installations, three scenarios have been provided: SMALL, MEDIUM
# and LARGE.  Any parameter that needs to be tuned according to
# installation size will have three settings, each one commented
# according to installation size.
#
# Use the following table to approximate the SGA size needed for the
# three scenarios provided in this file:
#
#           -----Installation/Database Size-----
#           SMALL           MEDIUM           LARGE
```

```

# Block          2K   4500K           6800K           17000K
# Size           4K   5500K           8800K           21000K
#
# To set up a database that multiple instances will be using, place
# all instance-specific parameters in one file, and then have all
# of these files point to a master file using the IFILE command.
# This way, when you change a public
# parameter, it will automatically change on all instances. This is
# necessary, since all instances must run with the same value for many
# parameters. For example, if you choose to use private rollback segments,
# these must be specified in different files, but since all gc_*
# parameters must be the same on all instances, they should be in one file.
#
# INSTRUCTIONS: Edit this file and the other INIT files it calls for
# your site, either by using the values provided here or by providing
# your own. Then place an IFILE= line into each instance-specific
# INIT file that points at this file.
#####

# db_file_multiblock_read_count = 8                # SMALL
db_file_multiblock_read_count = 16                # MEDIUM
# db_file_multiblock_read_count = 32              # LARGE

# db_block_buffers = 60                            # SMALL
db_block_buffers = 550                            # MEDIUM
# db_block_buffers = 3200                          # LARGE

# shared_pool_size = 3500000                       # SMALL
shared_pool_size = 5000000                         # MEDIUM
# shared_pool_size = 9000000                       # LARGE

log_checkpoint_interval = 10000

# processes = 50                                   # SMALL
processes = 100                                    # MEDIUM
# processes = 200                                  # LARGE

# dml_locks = 100                                  # SMALL
dml_locks = 200                                    # MEDIUM
# dml_locks = 500                                  # LARGE

# log_buffer = 8192                                # SMALL
log_buffer = 32768                                 # MEDIUM
# log_buffer = 163840                              # LARGE

# sequence_cache_entries = 10                     # SMALL
sequence_cache_entries = 30                        # MEDIUM
# sequence_cache_entries = 100                     # LARGE

# sequence_cache_hash_buckets = 10                # SMALL
sequence_cache_hash_buckets = 23                   # MEDIUM
# sequence_cache_hash_buckets = 89                 # LARGE

# audit_trail = true                               # if you want auditing
# timed_statistics = true                          # if you want timed statistics
max_dump_file_size = 10240                          # limit trace file size to 5 Meg each

# Uncommenting the line below will cause automatic archiving if archiving has
# been enabled using ALTER DATABASE ARCHIVELOG.
# log_archive_start = true
# log_archive_dest = disk$rdbms:[oracle.archive]
log_archive_format = "T%TS%S.ARC"

# If using private rollback segments, place lines of the following
# form in each of your instance-specific init.ora files:
# rollback_segments = (name1, name2)

# If using public rollback segments, define how many
# rollback segments each instance will pick up, using the formula
# # of rollback segments = transactions / transactions_per_rollback_segment
# In this example each instance will grab 40/10 = 4:

```

```
# transactions = 40
# transactions_per_rollback_segment = 10

# Global Naming -- enforce that a dblink has same name as the db it connects to
global_names = FALSE

# Edit and uncomment the following line to provide the suffix that will be
# appended to the db_name parameter (separated with a dot) and stored as the
# global database name when a database is created. If your site uses
# Internet Domain names for e-mail, then the part of your e-mail address after
# the '@' is a good candidate for this parameter value.

# db_domain = us.acme.com      # global database name is db_name.db_domain

# FOR DEVELOPMENT ONLY, DEFAULT TO SINGLE-PROCESS
# single_process = TRUE
open_links = 50
db_files = 60
```

- In \$ORACLE_HOME copy the file 'initMTI.ora' into 'initMTI_0.ora' (where MTI is the Oracle Instance Name or "SID"). Edit this file: to reflect your instance; **define no rollback segments** for the initial start up. Sample:

```
ifile                = /opt/oracle/app/oracle/product/7.3.2/dbs/configMTI.ora
rollback_segments   = ( )
#
# $Header: init.ora,v 1.19 1994/09/19 18:49:08 alingelb Exp $
#.....
```

- In \$ORACLE_HOME copy the file 'cnfg.ora' into 'configMTI.ora' (where MTI is the Oracle Instance Name or "SID"). Edit this file: to reflect your instance; define the block_size according your needs (4k is ok for a base with one million documents). Sample:

```
#
# $Header: cnfg.ora 1.1 95/02/27 12:14:25 wyim Osd<unix> $ Copyr (c) 1992 Oracle
#
# cnfg.ora - instance configuration parameters

control_files        = (?/dbs/ctrl1@.ctl,
                       /etc/ctrl2@.ctl,
                       /data/oracle/@/ctrl3@.ctl)
# Below for possible future use...
#init_sql_files      = (?/dbs/sql.bsq,
#                       ?/rdbms/admin/catalog.sql,
#                       ?/rdbms/admin/expvew.sql)
background_dump_dest = /data/oracle/MTI/bdump
core_dump_dest       = /data/oracle/@/cdump
user_dump_dest        = /data/oracle/@/udump
#log_archive_dest    = /data/oracle/@/arch/arch.log
db_block_size        = 4096

db_name               = MTI
```

Be aware, that a Teamcenter Enterprise Database is an Oracle User !

The "@" sign is used in file paths to symbolize the SID. Oracle will replace it by the SID.

The "?" sign is used in file paths to symbolize the ORACLE_HOME. Oracle will replace it by the directory name found in ORACLE_HOME variable.

Use 3 different locations for your control files!

- Now as the Oracle user do call "svrmgrl"⁶. And enter:

```
connect internal
startup nomount;
create database pfile=/usr5/app/oracle/admin/MTICEUX1/pfile/initMTI_0.ora;
@ ?/rdbms/admin/catalog
```

⁶) „svrmgrl“ is the new Oracle „Server Manager“ that replaces old „sqldba“. There is a line mode version „svrmgrl“ and a motif version „svrlmgrm“. But be aware, „svrlmgrm“ does not work correctly yet.


```
@ ?/rdbms/admin/catproc
create tablespace ROLLBACKSPACE
  datafile '/data/oracle/@/rollback_space';
create rollback segment R01 table space ROLLBACKSPACE
  storage (initial 2M next 5M pctincrease 50 minextents 2 maxextents 100)
  online;
create rollback segment R02 table space ROLLBACKSPACE
  storage (initial 2M next 5M pctincrease 50 minextents 2 maxextents 100)
  online;
create rollback segment R03 table space ROLLBACKSPACE
  storage (initial 2M next 5M pctincrease 50 minextents 2 maxextents 100)
  online;
create rollback segment R04 table space ROLLBACKSPACE
  storage (initial 2M next 5M pctincrease 50 minextents 2 maxextents 100)
  online;
shutdown;
startup;
exit
```

- Verify your Oracle instance by using “*sqlplus*” as System.
- **Caution exact scripts may vary between platforms and versions !**

3.6 Install SqlNet 2 (old)

Unlike previous versions, you need to define a domain for any *sqlnet* service (entry in ‘*tnsnames.ora*’). To get Teamcenter Enterprise still running with this, you need to turn off “Global Names” in the ‘*initMTI.ora*’ file.

```
TRACE_DIRECTORY_LISTENER = /data/oracle/MTI/logs  
LOG_DIRECTORY_LISTENER = /data/oracle/MTI/logs  
LOG_FILE_LISTENER = alpha_listener.log
```

3.6.2 sqlnet.ora

```
#####  
## Configuration SQL*Net  
##  
## Application MTI  
## Serveur ALPHA  
## Created 26/05/97  
#####  
  
AUTOMATIC_IPC = OFF  
TRACE_LEVEL_CLIENT = OFF  
TRACE_FILE_CLIENT = alpha_client.trc  
TRACE_DIRECTORY_CLIENT = /data/oracle/MTI/logs  
TRACE_LEVEL_SERVER = OFF  
TRACE_FILE_SERVER = alpha_server.trc  
TRACE_DIRECTORY_SERVER = /data/oracle/ MTI/logs  
SQLNET.EXPIRE_TIME = 0  
NAMES.DEFAULT_DOMAIN = world  
NAMES.DEFAULT_ZONE = world  
SQLNET.AUTHENTICATION_SERVICES = OFF
```

3.6.3 tnsnames.ora

```
#####  
### Configuration SQL*Net v2  
###  
### Application MTI  
### Serveur ALPHA  
### Created 26/05/97  
###  
#####  
  
ALPHA.world =  
  ( DESCRIPTION =  
    ( ADDRESS_LIST =  
      ( ADDRESS =  
        ( COMMUNITY = TCP.world )  
        ( PROTOCOL = TCP )  
        ( HOST = ALPHA.world )  
        ( PORT = 1521 )  
      )  
    )  
  ( CONNECT_DATA =  
    ( SID = MTI )  
    ( GLOBAL_NAME = ALPHA.world )  
  )  
)
```

3.7 Get Instances started (old)

Do not forget to start and stop Oracle and SQLnet automatically at boot time. The way to do this depends slightly on the platform.

UNIX:

There is a file or directory "rc2.d" in "/etc" that must contain commands like:

```
|su - oracle -c (lsnrctl start; dbstart)
```

And

```
|su - oracle -c (dbshut; lsnrctl stop)
```

NT:

Follow Oracle instructions to start Oracle and SQLnet as a service.

3.8 Install Users and Tablespaces (Teamcenter Enterprise Databases)

Note: A Teamcenter Enterprise Database is an Oracle User

3.8.1 Standard Teamcenter Enterprise Database Creation (old)

This is by far the preferred method for database creation:

```
|credbora -n database_name -s syspw [-p database_name.pwf] \  
|          -c tablespacepath -tblspacename=size [-t tmpspacename] \  
|          [-i indexspacename] [-l logfile]
```

Where:

syspw password of oracle system users

size in Megabytes

tmpspacename and *indexspacename* must already exist as tablespaces.

▶ *This is called internally by cfgedit2 now.*

3.8.2 Alternative shell script "create_user"

The following shell-script (csh) lets you define a Teamcenter Enterprise Database outside of Teamcenter Enterprise installation routines. It is called as:

```
create_user DBname TabSize IndexSize
```

where "TabSize" is the initial size of the tablespace you associate to the oracle user (i.e. 50M or 300M), this is used to store the tables only (not indexes), if you set a size for "IndexSize" too

and where "IndexSize" is the initial size of the tablespace you want to assign to indexes.

create_user

```
|#!/bin/csh -f  
|# call create_user username tablesize indexsize  
|# sample ### create_user pdm2 50M 50M  
|if ("-$2" == "--") then  
|  $0 $1 60M 0  
|else  
|sqlplus system/manager << EOF  
|set pages 0 hea off echo on  
|drop user $1 cascade;  
|commit;  
|drop tablespace ${1}_indexsp including contents;  
|commit;  
|drop tablespace ${1}_space including contents;  
|commit;
```

```
create tablespace ${1}_space datafile '/datadisk/@/${1}space.dbf' size $2 reuse
  default storage (initial 2k next 500k pctincrease 0 minextents 1
    maxextents 200);
create tablespace ${1}_indexspace datafile '/datadisk/@/${1}indexspace.dbf'
  size $2 reuse
  default storage (initial 2k next 500k pctincrease 0 minextents 1
    maxextents 200);
commit;

create user $1 identified by oracle
  default tablespace ${1}_space
  temporary tablespace ${1}_space
  quota unlimited on ${1}_space;
  quota unlimited on ${1}_indexspace;
grant connect to $1;
commit;
exit;
EOF
if ("-$3" != "-" && ("-$3" != "-0")) then
  sqlplus system/manager << EOF2
  set pages 0 hea off echo on
  create tablespace ${1}_indexsp datafile '/indexdisk/@/${1}indexsp.dbf'
    default storage (initial 128k next 4M pctincrease 20 minextent 1 maxextent 200)Error! Bookmark not defined.
    size $3 reuse;
  commit;
  alter user $1
    quota unlimited on ${1}_indexsp;
EOF2
Endif
dbenc ${1}/oracle >&! $MTI_ROOT/${1}.pwd
endif
```

3.8.3 Install Teamcenter Enterprise with one tablespace per DB

If you let create your tablespace through Teamcenter Enterprise installation scripts, then your tablespace will not be big enough to hold all tables.

Better practice is to use the routine above with tablespace size not lower than 50M; then reply to Teamcenter Enterprise scripts, that you have already created oracle user and tablespace. In this case call the create_user script above with "IndexSize" 0.

Note: during this process do spell the Dbname everywhere the same way. Teamcenter Enterprise does distinguish upper and lower cases, even if oracle does not care about. Different spelling may bring you in trouble later.

3.8.4 Install Teamcenter Enterprise with two tablespaces per DB

If you prefer to follow Oracle's guidelines and install different tablespaces for tables and indexes, then you should proceed as follows:

1. **Create the Oracle user(s)⁷ with "create_user <DBname> <TabSize> 0"**
2. **Install Teamcenter Enterprise with already created Oracle user(s) and tablespace(s).**
3. **Run:**

```
cd $MTI_ROOT
mschema -g tmti.prd > tabs.msql
mschema -G tmti.prd > indx.msql
sed -e '/generic/s/generic/long/g' tabs.msql > tabs.sql
sed -e '/clustered index/s/clustered index/index/g' indx.msql > indx.sql
rm tabs.msql indx.msql
```

⁷) all if distributed environment.

4. **Run for each <DBname>:**

```
objdump -d DBname -t DBname.dat
create_user DBname TabSize IndexSize
sqlplus DBname/oracle
start tabs.sql
insert into zdbconfig (OBID,zname,zvalue) values
  ('00A','dbDb','DBname', '00B','dbSite','ORA_SID');
commit;
connect system/manager
alter user DBname default tablespace DBname_indexsp;
commit;
connect DBname/oracle
start indx.sql
commit;
connect system/manager
alter user DBname default tablespace DBname_space;
commit;
exit;
objload -d DBname -rnc -f DBname.dat -K DBname.key8
```

From Metaphase 2.3 on you should use the Metaphase / Teamcenter Enterprise standard parameters on all creation and update utilities to indicate the separate table spaces for tables and indexes.

3.9 Other critical ORACLE parameters

Some Oracle parameters must be adapted to the local or total number of Teamcenter Enterprise databases in use. These may be changed, whenever Oracle is down. They are set through the "init<SID>.ora" or its relative config file.

The parameters have very low values in an Oracle standard installation, which is fine for single database, but very soon in a distributed environment you will get some Oracle runtime error⁹.

Recommended values are:

```
max_transaction_branches = 32
distributed_transaction = 32
open_links = 50
db_files = 60
```

max_transaction_branches:	must at least be 2 higher than the number of local Teamcenter Enterprise databases.
distributed_transaction:	must at least be 2 higher than the number of remote Teamcenter Enterprise databases.
open_links:	must at least be 2 higher than the total number of Teamcenter Enterprise databases.
db_files	is the maximum of data files that may be used for the totality of local table spaces (including the system and temporary spaces or spaces of other Oracle users)

⁸) an objload of DBname.key is not needed, as you load in keys anyway too.

⁹) Sample: „Too many local sessions participation in global transaction“