



# Teamcenter Enterprise: Guidelines for Setup and Configuration

Volume ITI-MTI-B

Written by

**Erich Brauchli**

March 2, 2008

**Version 9.24**

Reflecting Teamcenter<sup>®</sup> Enterprise 2005

Email: [erich@brauchli.info](mailto:erich@brauchli.info)

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

(Total 41 pages)

**Entire List of Teamcenter Enterprise Documents:**

ITI-MTI-A	Guidelines for Planing 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

<b>1</b>	<b>A small history of Teamcenter Enterprise</b> .....	<b>3</b>
<b>2</b>	<b>Distributed Teamcenter Enterprise</b> .....	<b>3</b>
2.1	Oracle.....	3
2.2	OS-Username – “trusted User” .....	4
2.3	IP-Port .....	4
2.4	License File / Port.....	4
<b>3</b>	<b>Distributed Services</b> .....	<b>5</b>
3.1	Service topology .....	5
3.2	Service configuration parameters.....	7
3.2.1	Service name .....	7
3.2.2	DSP arguments.....	7
3.2.3	Service path .....	9
3.2.4	Service arguments .....	9
3.3	Configuration recommendations .....	10
3.3.1	Reasonable starting values .....	10
3.3.2	Observation and correction methods.....	10
3.4	Special Services.....	11
<b>4</b>	<b>Sample Configuration</b> .....	<b>12</b>
4.1	Sample „dbtop.prd“.....	12
4.2	Sample of „mtlinks.sql“ .....	13
4.3	„config.cfg“.....	14
4.4	Sample “toccus.dat” file .....	15
4.5	Sample “cfgcus.dat” file .....	16
4.5.1	Special Services.....	18
4.6	Platform specific .....	19
4.6.1	Windows: .....	19
4.6.2	AIX .....	19
<b>5</b>	<b>European Language Support in Oracle</b> .....	<b>20</b>
<b>6</b>	<b>Mail / Notifications</b> .....	<b>20</b>
6.1	Variables in Config File .....	20
<b>7</b>	<b>Hostnames and Communication</b> .....	<b>21</b>
<b>8</b>	<b>File System access and Shared Disks</b> .....	<b>22</b>
8.1	File Systems – Locations - Files.....	24
8.1.1	Bulk Data Files.....	24
8.1.2	File System part of other “File System Items”.....	24
8.1.3	Meta Data .....	25
8.1.4	File System Item Class Structure .....	25
8.2	Spaces and Locations .....	26
8.2.1	Overview and Identification .....	26
8.2.2	Vaults and Vault Locations.....	26
8.2.3	Work Locations .....	28
8.3	File Systems.....	28
8.3.1	File System Definition.....	29
8.3.2	File System Mounts.....	30
8.3.3	Proxy Host.....	30
8.4	Shared drives versus “Replicated Vault Locations”.....	31
8.4.1	Shared Drives or mounted file systems.....	31
8.4.2	Replicated Vault Locations.....	31
<b>9</b>	<b>Data Integrity Mechanisms</b> .....	<b>32</b>
9.1	Data Integrity Failure Sources with Teamcenter Enterprise.....	32
9.2	Instrumented Integrity Checks.....	33
9.2.1	Utilities .....	33
9.2.2	Reasonable Frequency .....	34
9.3	Non Instrumented Checks.....	35
9.3.1	Database – Vault Location Consistency .....	35
9.3.2	Data Dictionary – Database Schema Consistency .....	35
9.4	Recovery Options .....	35
9.4.1	Full Recovery .....	35
9.4.2	Partial Recovery.....	35
<b>10</b>	<b>Upgrade of Metaphase 3 Instances created under Metaphase 2</b> .....	<b>36</b>
10.1	Configuration File .....	36
10.2	File “toccus.dat”.....	36
10.3	Distributed Databases .....	36
<b>11</b>	<b>Teamcenter Instance Migration / Data Migration</b> .....	<b>36</b>
11.1	Definition of the TERM “migration” .....	36
11.2	Scenarios .....	37
11.3	Upgrade Migration.....	38
11.3.1	Upgrade of a Test System.....	38
11.3.2	Data Migration from a lower release to a higher one .....	38
11.3.3	Productive Upgrade Alternative.....	39

# 1 A small history of Teamcenter Enterprise

1990: the two companies SDRC (Structural Dynamics Research Corporation<sup>1</sup>) and CDC (Control Data Corporation<sup>2</sup>) decided to strengthen their relative CAD support systems<sup>3</sup> (DMCS and EDL<sup>4</sup>) into a common new PLM product in a modern world wide distributed context.

Under the lead of chief design architect Jim Heppelman the new common company Metaphase Technology (owned 50% by each of the two parent companies) did build a PDM software from scratch, in new object oriented architecture, but with a development crew built out engineers of the two parent products.

1994: First Release a revolutionary PDM product Metaphase 2.

1996: SDRC became 100% owner of Metaphase Technology.

2001: SDRC was purchased by EDS subsidiary UGS Corporation.

2004: EDS sold off UGS Corporation.

2007: UGS Corporation was purchased by Siemens AG and is now "Siemens PLM Solutions"

The product was branded as Metaphase 2.0 to 3.3, then Collaboration Foundation 1 to 2, then Teamcenter Enterprise 3.0 and later and will continue in the next years as such original version as well as soon in a new common platform together with Teamcenter Engineering and other Teamcenter application.

## 2 Distributed Teamcenter Enterprise

### 2.1 Oracle

Before you install any Teamcenter Enterprise, check carefully the installed **Oracle** and **SqlNet** installations. On any side log in as Oracle administrator and try to create a database link to any other used Oracle system, then try the access to the remote system doing a

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

If this does not work, do not continue with Teamcenter! Teamcenter's *msqlora* service will only 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 links<sup>5</sup>.

If this verification runs as os user "oracle" but not as the Teamcenter "trusted user", this may have one of the following causes:

- "tnsnames.ora" has no read permission set for "others".
- Oracle Variables or path is not fully set up for the Teamcenter "trusted user".

**Caution under Windows:** do not follow sample connections in SqlNet, which are not TCP connections.

---

<sup>1</sup> ) Founded 1967 by professor Dr. Jason R. Lemon from University of Cincinnati.

<sup>2</sup> ) Founded 1957 by William C. Norris in Minneapolis.

<sup>3</sup> ) Rather CAD drawing management systems.

<sup>4</sup> ) EDL = Engineering Data Library, 1990-1994 branded as Metaphase 1.

<sup>5</sup> ) Except usage of „DOT-Notation“, more details see volume ITI-MIT-C, chapter 3.3 Usage of „DOT notation“.

## 2.2 OS-Username – “trusted User”

If on your system exists already a Teamcenter Enterprise installation: you must install and run the new environment as another os-user. This user is the Teamcenter “trusted user”, the owner of the Teamcenter Enterprise Application. The Teamcenter background processes can only determine to which installation they belong through the **os-username** under which they are started. Any mismatch here may either not work, or corrupt your database.

***Each Teamcenter Enterprise installation runs under its own unique OS-user name; it is recommended to be the same over all servers of the Teamcenter Enterprise instance.***

## 2.3 IP-Port

Obviously each Teamcenter Enterprise installation must run with its own unique **IP-port** within the entire network. This means all software servers and clients on all machines, which communicate with each other as one common Teamcenter Enterprise system must have exactly one IP-port number declared as VC\_MUX\_IPPORT<sup>6</sup>, which may not be used by any other Teamcenter Enterprise environment in your network, nor by any other service in your network. It is recommended to enter the used IP-port in the */etc/services* file, just for documentation purpose (Teamcenter Enterprise does not look up their, because you may run more than one Teamcenter Enterprise system on one host, so they must run different on IP-ports).

***Each Teamcenter Enterprise installation has its own unique IP-Port number; which is mandatory the same over all servers and clients of the Teamcenter Enterprise instance.***

### Note for AIX:

On AIX cluster systems (i.e. SP line) the default Teamcenter Enterprise is in the IBM reserved range for Cluster communication. IBM requests any application port to be higher than 20'000.

## 2.4 License File / Port

The port number on the license file must be different from any config IP-port. All Teamcenter Enterprise installations (environments) sharing the same license manager host must use the same license IP port.

Even in distributed environment you do not need more than one license server host! But the license server host must be running and accessible to create a new rule cache file.

In distributed environment all host running any method service need an exact copy of the license file. The host mentioned in the license file is the master license server. All other need the copy for check of modules allowed to run. These copies do not serve for *FlexLM*; it is read directly by “*lamserv*”.

The numbers users (authors/consumers/explorers) are checked only at rule file generation time. This action uses FlexLM access, thus need a connection to the license server host. Checks for licensed modules are made by dispatcher at startup through “*lamserv*” (“*lamserv*” reads the LOCAL COPY of the license file [exact copy of the file on license server host]).

This may explain, why each host running a method service (“*objserv*”, ...) does need a copy of the license file, and must run “*lamserv*”.

You need to have a license file for the highest release level of Teamcenter you run in the company. The two programs “*lmgrd*” and “*mtid*” must be of this release too. Then it can serve any of you Teamcenter Enterprise installations.

---

<sup>6</sup>) This port is used just to open a connection stream, which then runs under its own free port number.

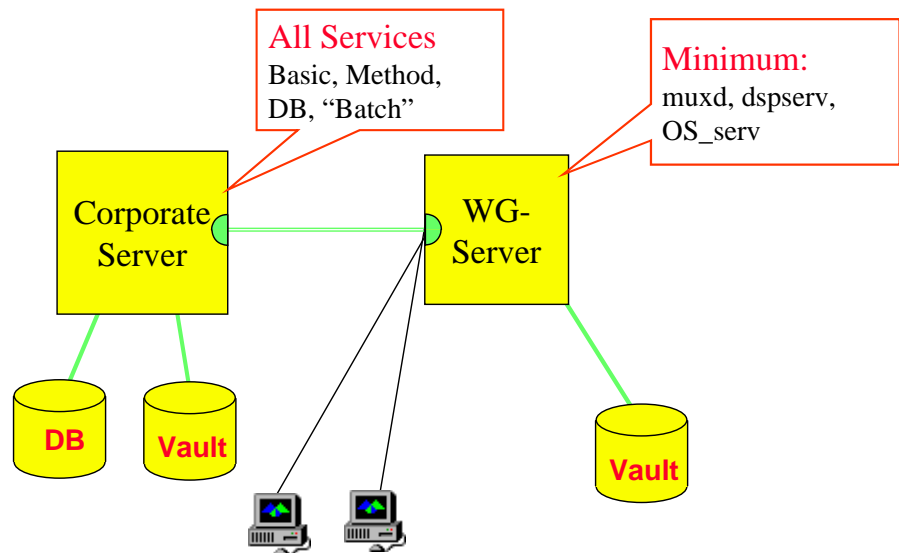
## 3 Distributed Services

### 3.1 Service topology

Workgroup Servers may have different qualities, depending on level of service distribution. This distribution has impact on the hardware sizing.

#### Level 1:

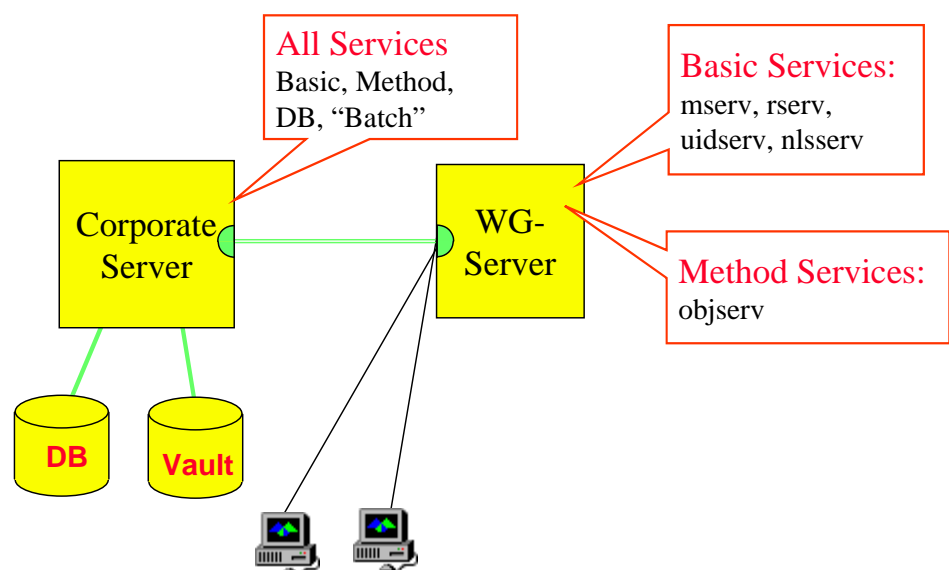
No local processing is delivered by the WG server. But Files can be stored in local Vault Locations.



#### Level 2:

If you add "mloader" to basic services, then you need a local copy of icons, forms, texts and rulecache. Without method services this is often the most efficient WG server for distant sites.

With Method services this delivers all non DB actions locally. Files to and from vaults need still a transfer.



Noticeable reduce of net traffic and frees the central CPU as well as central MUX.

*This level can be combined with level 1.*

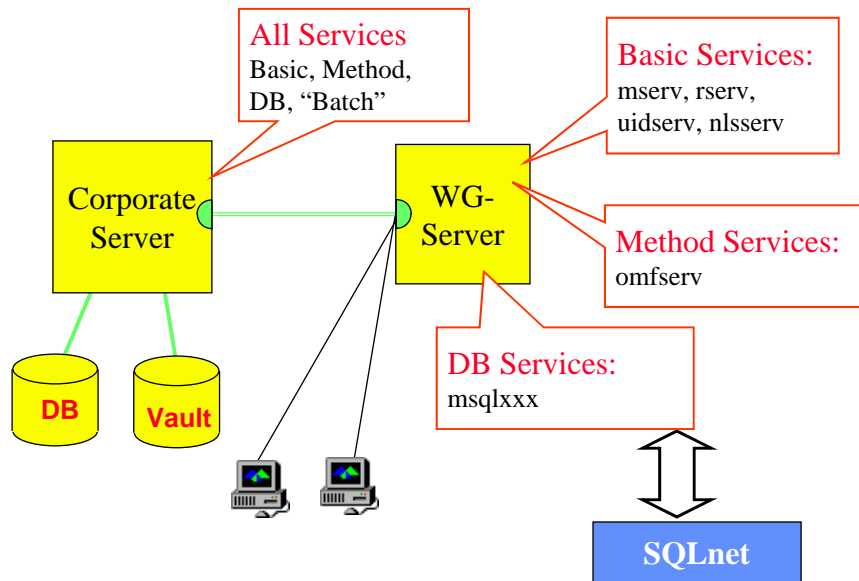
Services like „advancer“, „expireserv“, „schedulserv“ may be considered as batch services.

**Level 3:**

This reduces the traffic through mux and replaces it by SQLnet traffic.

This option is a work-around if the muxd on the corporate server (or relative DB server) is overloaded, but depending on activities it may increase somewhat the network load.

It can be combined with level 1.



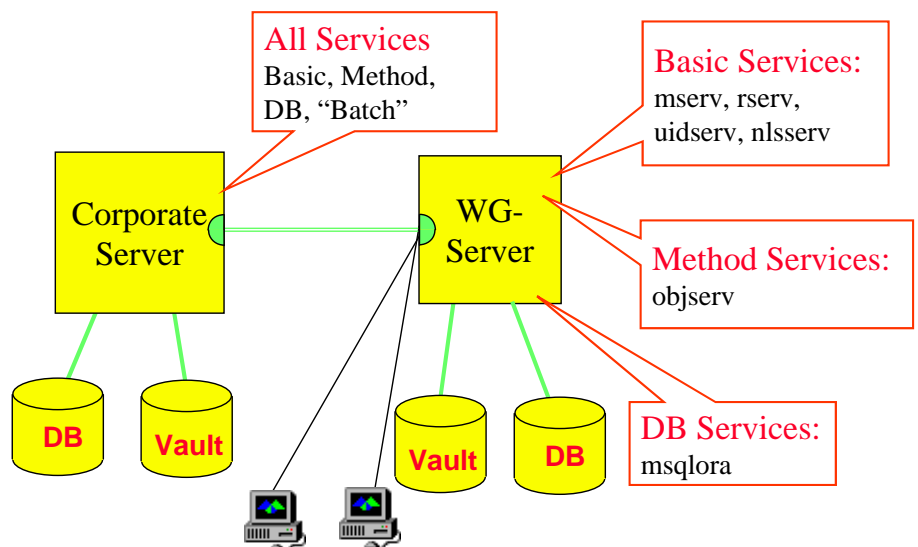
In this environment, the WG server has not a full Oracle installation, but only an Oracle Client.

Vault Locations may be added independently from presence of Method Services.

**Level 4:**

This can be configured so that almost no traffic is seen between MUXes of the servers. Remaining MUX connections are for moves between vaults.

Any DB access is made locally and forwarded through SQLnet rather than Teamcenter protocol.



The difference between a level 4 server and the corporate sever is, that the corporate server contains:

- The Key Database

Some services may be only on a host, where their counterpart application or database is located:

- An MRP Interface service must run only on a server that has MRP access.
- Advancer and Expire service must run on servers that contain an OPS database.

A **registry service** should be established per major site. Note that each registry service must have a different service name (i.e. „rgy\_site“: list rgy\_site in services and service.cfg map of configuration file). Users the should be assigned to both a user database near to their main physical location and to the registry service of the respective major site.

Database distribution options see Volume ITI-MTI-A, chapter Facility Management, and volume ITI-MTI-C, chapter Installation of Distributed Databases.

## 3.2 Service configuration parameters

The config file contains a “service.cfg” map for all services running:

OS_SERV	"0 0 1 20 30 120	\$(@FILEPATH:q \$(PDM_BIN:q) osserv)"	
mserv	"1 1 1 3 3 120	\$(@FILEPATH:q \$(PDM_BIN:q) mserv)"	
rserv	"1 1 1 3 3 120	\$(@FILEPATH:q \$(PDM_BIN:q) rserv)"	
uidserv	"1 1 1 3 3 120	\$(@FILEPATH:q \$(PDM_BIN:q) uidserv)"	
nlsserv	"1 1 1 3 3 120	\$(@FILEPATH:q \$(PDM_BIN:q) nlsserv)"	
omfsvr	"1 2 9 12 12 120	\$(@FILEPATH:q \$(PDM_BIN:q) objserv) -C 500 -S 40"	
fedsvr	"0 0 2 4 4 120	\$(@FILEPATH:q \$(PDM_BIN:q) fedserv) -C 500 -S 40"	
registry_A	"0 1 1 1 1 1	\$(@FILEPATH:q \$(PDM_BIN:q) rgyserv) -sregistry_A"	
registry_B	"0 1 1 1 1 1	\$(@FILEPATH:q \$(PDM_BIN:q) rgyserv) -sregistry_B"	
notifserv	"0 0 1 1 1 1	\$(@FILEPATH:q \$(PDM_BIN:q) notif) -S 30"	
postserv	"0 1 2 3 3 1	\$(@FILEPATH:q \$(PDM_BIN:q) post) -h "	
advancerserv	"0 1 1 1 1 1	\$(@FILEPATH:q \$(PDM_BIN:q) advancer) -S 30"	
lamserv	"1 1 1 1 1 1	\$(@FILEPATH:q \$(PDM_BIN:q) advancer) -C 5000"	
msqlUsrB	"1 3 6 10 10 20	\$(@FILEPATH:q \$(PDM_BIN:q) msqlora) -sn\$(UsrB_PWF) -C 5000"	
<b>service</b>	<b>"dsp-args</b>	<b>path</b>	<b>service args"</b>

### 3.2.1 Service name

Each service must have its own specific name. Two services may run the same program, but having different functions, thus they are not equivalent in all respects, so they have different names. Each service may run in several instances on each server.

The main service is always “omfsvr”. Typically registry and msql servers exist with different names. I.e. a “registry\_A” may take care of all users of one major site, while “registry\_B” cares about all users of another site.

“msql”-servers have to login into exactly one local data base. Thus in a distributed environment there must run an “msql” on each DB host, but each host’s service must have a different name, which then can run in more than one instances.

### 3.2.2 DSP arguments

The arguments for DSP to control the service consist of 6 decimal numbers:

#### 3.2.2.1 Autostart argument

Manuals says: Specifies the threshold for starting new servers. When the system assigns servers, a new server is automatically started if both of the following criteria are met: The total number of idle servers drops to less than the autostart value. The total number of servers for this service is less than the target value. If you specify a nonzero value for AutoStart, the system starts servers during dispatcher initialization until the configured minimum value is reached.

This value is “0” for “no autostart; that means the service is only started, if there is a relative request. Set this value to “1” to get the “minimum” servers started at dispatcher start time.

In the sample above, DSP will start 3 msql, 2 omfsvr and 1 mserv at its own startup, but no advancerserv and no fedsvr..

#### 3.2.2.2 Minimum argument

Manuals says: Specifies the minimum number of servers kept idle or assigned for this service.

This value defines, how many idle servers are granted to be available, as far as the “target” is not yet reached

In the sample above DSP will autostart 3 omfsvr, and more, to have always 3 omfsvrs available (free), unless the total number of omfsvrs reaches 9. Then it may start additional omfsvrs, if requested, up to a total of 12, but never more. There is no advancerserv started automatically; but once started, this server will be kept even if idle for a long time.

### 3.2.2.3 Target argument

Manuals says: Specifies the ideal number of servers, both idle and busy, for this service. When the target value is reached, autostart is not performed.

If this number of servers is running, there no more any autostarts performed. But upon request, DSP will still start servers up to the maximum attribute.

**Recommended value for “omfsvr” is:** *The expected number of really concurrent users served by this server host. On a pure batch server use the number of work streams you desire. (But be aware of memory consumption of each one.)*

**Recommended value for “advancer” is:** *1. Any higher value can cause severe performance problems.*

*On a specific server running only batch services (no interactivity) you can run 2 advancers; more is a risk of dead-locks.*

### 3.2.2.4 Maximum argument

Manuals says: Specifies the absolute limit of servers for this service.

If this number of services is reached, no additional server will be assigned on a new request. The requests will now be queued until an assigned one gets free.

**Recommended value for “omfsvr” is:** *The “target” value multiplied by 1.5. This value can be increased, if “dspstat” reports a value for “queued at server limit” which is more than a very low percentage of “total assignments”.*

**Recommended value for “OS\_SERV” is:** *slightly higher than maximum of “omfsvr”.*

**Recommended value for “advancer” is:** *1. On a pure batch server use 2. Any higher value can cause severe performance problems.*

Be careful with max service and memory calculations. Respect your physical real memory size. Teamcenter Enterprise services should use swap space only in very exceptional cases! Otherwise your performance will be unacceptable.

Memory allocation differs between platforms. Good implementations of “shared libraries” do store the code of a library only once in the memory, while data memory segments are still associated to each process. (Some platform [like AIX] keep such a library in memory until next boot or an explicit library clean).

### 3.2.2.5 Variant Max argument

Manuals says: Specifies the maximum number of servers per variant for the OS\_SERV and GUISRV servers. For example, the operating system servers (OS\_SERV) are managed as separate variants based on the user name each server represents. Most servers are run as a single variant. For these servers, set the value of Maximum and Variant Max to the same value.

This value must be the same as “maximum” for all services except “OS\_SERV”.



**Recommended value for “OS\_SERV” is: maximum of “OS\_SERV” multiplied by 1.5.**

For a file transfer into a Vault usually you need to run two OS\_SERV under different user names: one under your name to read the file from the work location and one under system account to write it in the vault location.

### 3.2.2.6 Time Out argument

Manuals says: Specifies the number of **minutes** that an idle server waits before becoming eligible for termination.

### 3.2.3 Service path

This just defines the full path of the binary to be executed as the service. The “config” format of path description is common all supported platforms (Unix, Windows, Mac; the “:q” means correct support for file names with spaces by quotes).

### 3.2.4 Service arguments

Some services have different set of arguments. For each service you can get them from the manuals or calling the service-binary with argument “-H”. Most servers have a trace or debug flag as well as an argument to define a log file.

Due to problems in memory management in C code, customizations often create memory leaks (see chapter 2, “Memory Management” in volume ITI-MTI-D, “Guidelines for Customizing”). This results in servers growing in memory size over their life time. To get this effect undone, DSP will stop at certain events a service and restart it (according autostart or start rules). We try to do this in a reasonable frequency, because too few is a memory problem and results in swapping and performance degradation. Too much results in performance degradation too, since the startup of a service is usually expensive in CPU time, especially platforms which are not well tuned for shared library load. Lets here talk about the arguments controlling the stop and re-launch of a service:

#### 3.2.4.1 Count argument

The “-C” argument specifies the count of requests (top level) after which a service gets stopped, as soon it gets “free”. The call count is not very significant. One call may treat thousands of objects and accumulate memory leaks for each; or it can perform almost nothing and not accumulate leaks at all. Thus this argument is used rather as a security limit. It is recommended to be used with msqldb and lamserv together with a very high number (5000 to 10000).

#### 3.2.4.2 Size argument

The “-S” argument<sup>7</sup> limits the memory size of a service (in MBs). Thus if a server accumulates more than that size, it will be stopped next time it gets free.

The reasonable values for this argument may differ depending on several facts:

- ◆ The manual says, this is the “resident memory” of a service. But on some platforms it reflects the total size.
- ◆ Different platforms do not create the same size code for the same program source.

---

<sup>7</sup>) The size argument has only an effect on UNIX systems. It is ignored on other platforms.

- ◆ Think about the total memory used by “maximum” services at memory limit, and add OS and database requirements to it. The more the system swaps, the less it is performing.
- ◆ If your S-argument is lower than the initial size of the service, then it is constantly stopped and re-loaded (where there is usage or not in case of autostart), which is for sure not the intension.

### **3.3 Configuration recommendations**

There is not a best case for every installation. The optimum depends on many conditions, on platforms, on hardware sizing, on customization and strongly on the most frequent type of usage, including user behavior. Best practice is to start with reasonably estimated values, and then use the appropriate tools to watch the behavior and to react on observations.

#### **3.3.1 Reasonable starting values**

For mloader, mserv, rserv, nlsserv, uidserv and postserv use (1,1,1,1,1,120), except on a specific “batch” server do not autostart uidserv. If necessary you may increase maximum and variant max to “2”.

For registry, notifyserv and expireserv use (0,0,1,1,1,120).

For lamserv use (1,1,1,1,1,10000).

For omfsvr use (1,a,b,c,c,120), where b is the expected real number of concurrent users<sup>8</sup> or slightly higher.  $a = b * 0.5$ ;  $c = b * 2$ .

For msq use between  $\frac{1}{2}$  and 1 times the omfsvr values, with double timeout. Be aware that you need the same number of Oracle processes.

For advancer on a server with interactive users use (0,0,1,1,1,1,120); any other value will degrade your system. On a “batch only” server we recommend (1,1,1,2,2,120).

Custom services, which call “...AtParent” should run as dynamic linked service with omfsvr. Services, which depend on not commonly available resources, should run as separate services. Tread them like omfsvr, but estimate about concurrent usage of the specific service. If such a service connects to a legacy system, verify the number of connections, which are allowed in parallel.

#### **3.3.2 Observation and correction methods**

The size arguments may easily be controlled by simple “ps” commands on UNIX. Adjust them according usually seen size and available physical memory.

To control the number of services, use the Teamcenter command “dspstat”; it show following information for each service running in the system:

---

<sup>8</sup>) Concurrent users are not the ones concurrently logged in into Teamcenter, but rather the ones concurrently using an omfsvr.

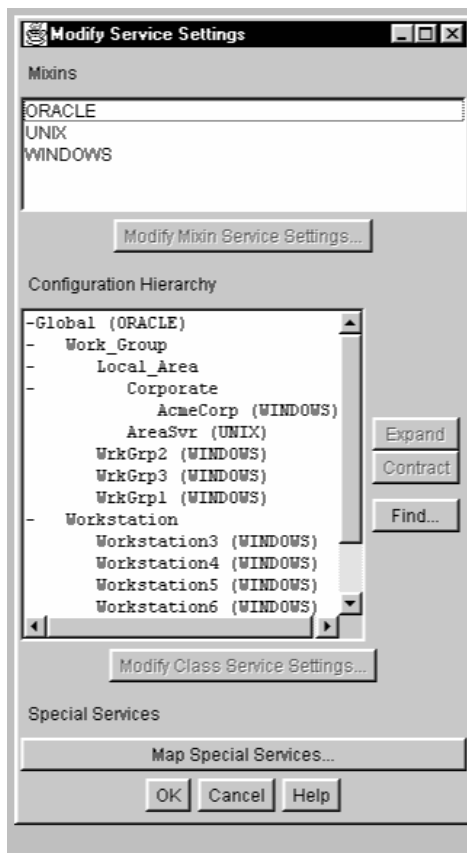
```

Service: omfsvr
Configuration:
  Path: C:\Program_Files\Teamcenter\OMF31\bin\objserv
  Args: -C 250 -S 30
  Autostart: 1 Min: 2 Target: 3 Max: 10 Var Max: 10 Timeout: 5
  Assign calls: 3528
    immediate assign: 3389
    queued assign: 139
      queued for new server: 136
      queued at server limit: 3
Total: 4
  Available: 2
  Variants: 2
    
```

Most information is accumulated since start of dispatcher, except the number of available, variants and total services, which are instant pictures.

The above sample means, that actually there are 2 concurrent users (even if 100 may be logged in; that is reported by “rgystat”), while 2 free services are available for immediate assignment. Since start of DSP 3528 times an omfsvr has been assigned. 3389 assignment could be done immediately, an available (free) server was ready. In 136 cases a new omfsvr had to be started. Only 3 times a user had to wait for a server getting free.

**Conclusion:** In 3% of the user calls, the user had to wait for a server being started; and in 1% a user had to wait for a service becoming free. This looks like very reasonable.



The target should be near to the average “total – available” observed on a regular basis.

### 3.4 Special Services

Services are communicated downwards through the server hierarchy. This is fine as long as the corporate server hosts all services. In this case the corporate server finds all services on itself; and this is the default server host for all others. Whenever a server runs on a lower server too, this is known from there on, downwards.

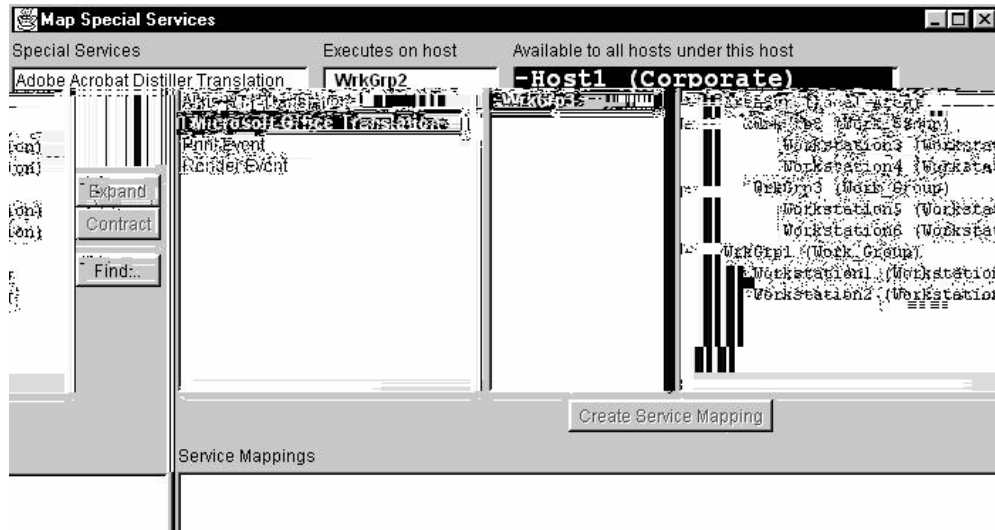
Some servers may run on a specific platform only (like some translation services), which is not the platform of your corporate server. Or a service may be bound to a license not available on the corporate.

Such services have to be defined as “special services”. They may run at a lower level, but this must be known in the hierarchy above them.

Special services have to be defined in the “cfgcus.dat” (going into “\$MTI\_ROOT/install”) in the following format:

A “Batch Server” running LCM processes would require to run Advancer and others as “Special Services” (see 4.5 Sample “cfgcus.dat” file)

```
#####
#@ linkinDef advancerserv OnOS "AIX", "NT"
   insert service.cfg
       sepc-serv "1 1 2 2 2 5 $(@FILEPATH:q $(PDM_BIN:q) spec.exe) -C 800"
   ;
#@ auxEnd
```



Only if this is present in the install directory, the config editor will allow you to maintain correctly all relative maps and variables. Teamcenter Enterprise config editor supports the special services under the “Server settings” tab.

The figure to the left allows to “Create Service Mapping”, which means the correct entry in service.cfg to launch the Translation service on “WrkGrp3” host. And it mentions “WrkGrp3” for translation service in the “services” map even on “Host1”.

## 4 Sample Configuration

Since Release 3.1 of Metaphase these files are generated and maintained by the **Configuration Editor**. Do not edit them manually. The comments shown in this sample is not inserted by the Configuration Editor.

### 4.1 Sample „dbtop.prd“

Be aware: this file is no more manually maintained. It is under control of config editor. Do not change it. All input for it is got from tocXXX.dat and the editor and DB info got into editor.

```
/*
 * CORPORATE DATABASES (if Distributed)
 *
 * Database
 * Name          Class          Description
 * =====
 * keymti        KeyInfo        Key Data
 * adminmti      Admin           Administrative Data (R)
 *               Process        LCM Process Definition Data (R)
 *               LCStep         LCM Lifecycle Step Definition Data (R)
 * sumti         Corporate Super User Account
 *               Operate        and Corporat Events and Logs
 *
 * (R) - replicated database
 */

define database KeyDB ;          /* Key DB */
define database AdmAlpha ;      /* Admin DB Alpha */
```

```
define database AdmBeta ;      /* Admin DB Beta */
define database SuDb ;       /* User DB for super user(s) and Corp Operate*/

/*
 * USER DATABASES
 *
 * Database
 * Name          Class      Description
 * =====      =====  =====
 * UsrAlpha      Data of Alpha users and vaults
 * UsrBeta       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 ;

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

display KeyDB as "(Coporate Key DB)";
display SuDb as "(Coporate Events)";
display AdmAlpha as "(Corporate Admin DB)";
display UsrAlpha as "Corporate Data Base";
display AdmBeta as "(Beta Admin DB)";
display UsrBeta as "Beta Data Base";
```

## 4.2 Sample of „mtlinks.sql“

Be aware: this file is no more manually maintained. It is under control of config editor. Do not change it. All input for it is got from tocXXX.dat and the editor.

```
/* Sample mtilinks.sql */
/* ----- */

/* Oracle Script to create KeyDB user */
drop database link KeyDB;

create database link KeyDB
  connect to KeyDB identified by oracle
  using 'TNS_Alpha' ;

commit;

/* Oracle Script to create SuDb user */
drop database link SuDb;

create database link SuDb
  connect to SuDb identified by oracle
  using 'TNS_Alpha';

commit;

/* Oracle Script to create AdmAlpha user */
drop database link AdmAlpha;

create database link AdmAlpha
  connect to AdmAlpha identified by oracle
  using 'TNS_Alpha';

commit;

/* Oracle Script to create UsrAlpha user */
drop database link UsrAlpha;

create database link UsrAlpha
  connect to UsrAlpha identified by oracle
  using 'TNS_Alpha';

commit;

/* Oracle Script to create AdmBeta user */
drop database link AdmBeta;

create database link AdmBeta
  connect to AdmBeta identified by oracle
  using 'TNS_Beta';

commit;

/* Oracle Script to create UsrBeta user */
drop database link UsrBeta;

create database link UsrBeta
  connect to UsrBeta identified by oracle
  using 'TNS_Beta';

commit;

EXIT;
```

### **4.3 „config.cfg“**

The configuration file “config.cfg” is no more maintained directly. **It is even forbidden to be edited manually** (except in customization contexts). Future “config.cfg” will no more be an ASCII file!

The entire “config.cfg” is maintained exclusively by the configuration editor. Manual addition can be made in “site.cfg”, which is included in “config.cfg” by default. Variable and map settings have to be inserted through the configuration editor for the entire instance of Teamcenter. And the configuration editor writes then the configuration files for each host in your configuration. You have to copy them manually to each host!

The initial structure of the configuration file is depending on each module’s “cfg<MOD>.dat” file. Additional information for the configuration editor is delivered by “toc<MOD>.dat”. As this info (opposite to earlier versions) is now taken in account seriously.

#### 4.4 Sample “toccus.dat” file

This is a required INPUT for each module, even custom ones.

```
// $Id: instoc.dat,v 2.7.2.3 1999/01/31 06:03:47 smdw Exp $
//bcprt
// Copyright (c) 1994, 1995, 1996, 1997, 1998, 1999 Structural Dynamics
// Research Corporation (SDRC).
// All Rights Reserved.
// SDRC and Teamcenter are registered trademarks, and Metaphase Enterprise,
// e!Vista, MetaChange, MetaDM, MetaSM, MetaSource, and MetaWeb are
// trademarks of Structural Dynamics Research Corporation.
// Federal Acquisitions: Commercial Computer Software - Use Governed by
// Terms of SDRC's Software License and Service Agreement.
//ecprt
//
// Installation Table of Contents
//
// Some possible fields for this file:
// -----
// ASYNC_LOC indicates that the module is released separately from th
// standard Teamcenter product.
// The insmenu script will request a new CD-ROM location to look for
// this module's files. If the value is set to N or the entry is not
// present in the file, the insmenu script does not request a new CD_ROM
// location. This alternate location may be any disk.
// Default value is N.
// REPSET Indicates that this module has classes to replicate in the indicated
// replica set. Possible replica sets are AdminRS, KeyInfoRS, and
// LCMRS.
// CLASS Indicates what classes to replicate for this module in the indicated
// replica set. Used in conjunction with REPSET.
// INSMET Indicates if this module has an tstins.met file.
// Default value is N.
// TXT Indicates if this module has an tsten_us.txt file.
// Default value is Y.
// FRM Indicates if this module has an tsten_us.frm file.
// Default value is Y.
// CUST Indicates if this is a customized module.
// Default value is N.
// DEF Indicates if this is a "default" Teamcenter out-of-the-box module.
// Default value is N.
// PRQ Indicates the prerequisite modules for this module.
// NAME Indicates the long name for your customization to be displayed in
// the Configuration Editor.//
TAPE cus
OSS OMF_OSS.TZ
DBS CUS_DBS.TZ
OBJ CUS _OBJS.TZ
MOD CUS _MODL.TZ
SPT CUS _SPRT.TZ
INSMET Y
DEF Y
CUST Y
SEPDB N
```

```
ASYNC_LOC Y
CLASS      J9Rep
REPSET     AdminRS
// class J9Rep and subclasses will be added to the admin replica set
NAME      "Company: local Customization"
CUST      Y

// The SOLUTION* fields are for "Server Solution" groupings in Config Editor
SOLUTION_NAME      "Local customization"
SOLUTION_MODULES   "cus"
PRQ                OMF LCM APC PFM CCF
```

## 4.5 Sample “cfgcus.dat” file

This is a required INPUT for each module, even custom ones.

The special services here are a sample for a “Batch server” running Life Cycle processes.

```
//bcprt
// Copyright (c) 1994, 1995, 1996, 1997, 1998 Structural Dynamics
// Research Corporation.
// All Rights Reserved.
// SDRC and Teamcenter are registered trademarks, and Metaphase Enterprise
// and e!Vista are trademarks of Structural Dynamics Research Corporation.
// Federal Acquisitions: Commercial Computer Software - Use Governed
// by Terms of SDRC's Software License and Service Agreement.
//ecprt

#####
#@ classDef Global

#$ Dispatcher Timeout at SBT
set DSPREADY_TIMEOUT "400";
#$ Max Components at SBT
set MAXCOMPONENTS "1000";
#$ Turn off Test output to status window
set TEST_DISPLAY "0";
#$ UID Preload Definition
set UIDSERV_CACHE_LIMIT "80";
#$ UID Preload Definition
set UIDSERV_PRELOAD_CLASSES "DDmtQry DPrtQry DDmtCre DPrtCre DDmtUpd DDmtGetI";

#$ Default Site at SBT
set LG_SITE "Zug";

#$ Default Vault for Register TIFF
set LG_VAULT "Archiv DDA";
#$ Default Vault for Register TIFF
set LG_VAULT_LOC "change me";

#$ Inncoscan Print Tool Parameter
set PSPRINT_WHOLE_COMMAND "1";

#$ Default application for file types
set ABCTool "unknown";
#$ Default application for file types
set WordTool "unknown";
#$ Default application for file types
set ViTool "none";
#$ Default application for file types
set none "none";
#$ Default application for file types
set PowerPnt "unknown";
#$ Default application for file types
set Designer "unknown";
#$ Default application for file types
set ExcelToo "unknown";
#$ Default application for file types
set MSProjec "unknown";
#$ Default application for file types
set PcadTool "unknown";
```



```

#$ Default application for file types
set TIFF "unknown";

#@ auxEnd

#####
#@ linkinDef advancerserv OnOS "AIX", "NT"
insert service.cfg
advancerserv "1 1 2 2 2 5 $(@FILEPATH:q $(PDM_BIN:q) advancer) -C 800"
;
#@ auxEnd
#@ linkinDef expireserv OnOS "AIX", "NT"
insert service.cfg
expireserv "0 1 1 1 1 1 $(@FILEPATH:q $(PDM_BIN:q) expire)"
;
#@ auxEnd
#@ linkinDef repserv OnOS "AIX", "NT"
insert service.cfg
repserv "0 0 0 1 1 5 $(@FILEPATH:q $(PDM_BIN:q) replic8r)"
;
#@ auxEnd
#@ linkinDef scheduleserv OnOS "AIX", "NT"
insert service.cfg
scheduleserv "1 1 1 1 1 1 $(@FILEPATH:q $(PDM_BIN:q) schedule)"
;
#@ auxEnd

#####
#@ mixinDef UNIX
#$ Default application for file types
set TIFF "InnoScan";
#@ auxEnd

#####
#@ mixinDef WINDOWS

#$ Obj Max Windows
set PDM_MAX_QUERY_OBJS "500";

#$ Default application for file types
set ABCTool "ABC_6_0";
#$ Default application for file types
set WordTool "MSWD_97";
#$ Default application for file types
set PowerPnt "MSPP_97";
#$ Default application for file types
set Designer "DESG_6_0";
#$ Default application for file types
set ExcelToo "EXCL_97";
#$ Default application for file types
set MSProjec "MSPJ_4_0";
#@ auxEnd

#####
#@ classDef Work_Group extends Global
#@ auxEnd

#####
#@ classDef Local_Area extends Work_Group
#@ auxEnd

#####
#@ classDef Corporate extends Local_Area
#@ auxEnd

#####
#@ classDef Workstation extends Global
#@ auxEnd

```

**You have to add in “site.cfg” all eventual “defines”:**

```
define LG_SITE ;
define LG_VAULT ;
define LG_VAULT_LOC ;
define PSPRINT_WHOLE_COMMAND;

define WordTool,
    AbcTool,
    PcadTool,
    ViTool,
    none,
    PowerPnt,
    Designer,
    MSProjec,
    TIFF;

# SBT - Special Values

define CentralServers
    "pdmcorp"
    "pdmap"
    "pdmdb"
    "chzugw3675"
    "chzugw1313"
;

set LG_VAULT host={view_HQ_Servers}    "PDM Daten";
set LG_VAULT host={view_SBT_Zug}      "Building System Zug";
set LG_VAULT host={view_SBT_Staefa}   "Staefa";
set LG_VAULT host={view_Rastatt}      "Building System Zug";
set LG_VAULT host={CentralServers}    "PDM Daten";

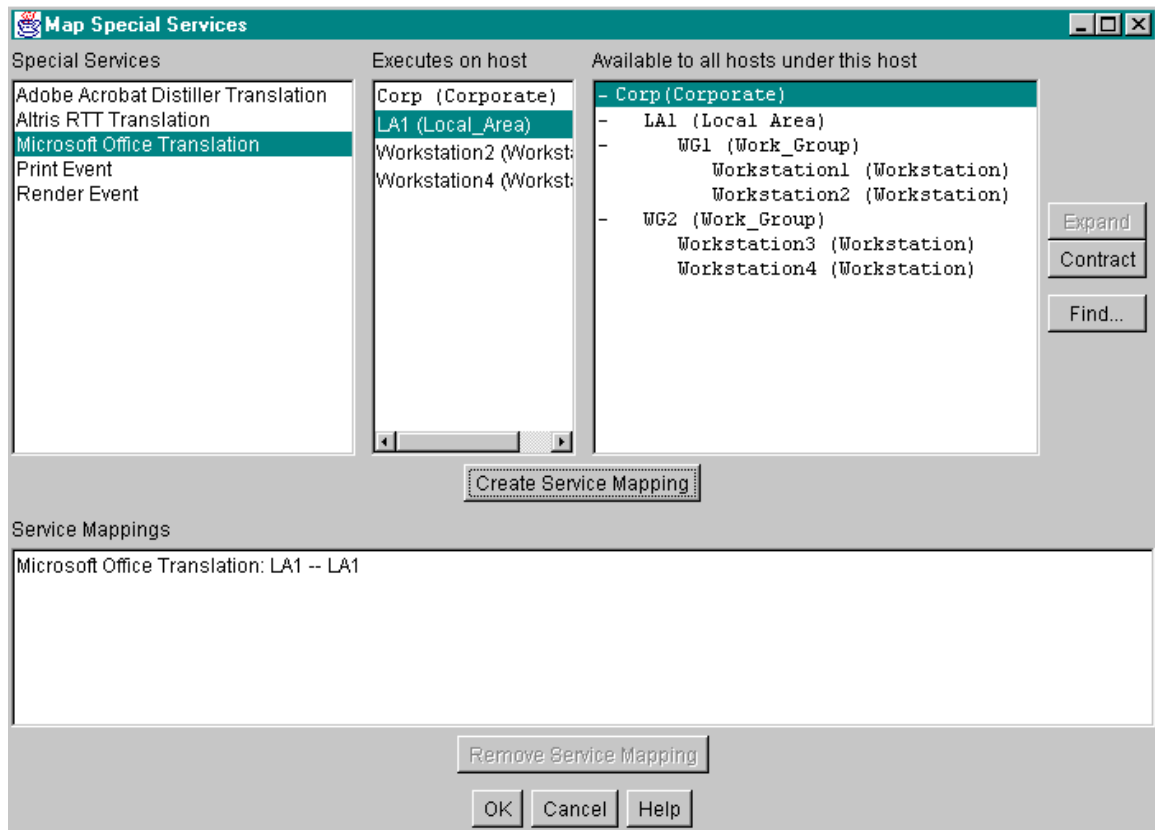
##### CHANGE LOCATIONS !! #####

set LG_VAULT_LOC host={view_HQ_Servers}    "PDM Daten";
set LG_VAULT_LOC host={view_SBT_Zug}      "Building System Zug";
set LG_VAULT_LOC host={view_SBT_Staefa}   "Staefa";
set LG_VAULT_LOC host={view_Rastatt}      "Building System Zug";
set LG_VAULT_LOC host={CentralServers}    "PDM Daten";

set LG_SITE host={view_HQ_Servers}        "Zug";
set LG_SITE host={view_SBT_Zug}          "Zug";
set LG_SITE host={view_SM_Zug}           "Zug";
set LG_SITE host={view_Rastatt}          "Rastatt";
set LG_SITE host={view_SBT_Staefa}       "l0Staefa";
# set LG_SITE host={view_Huddingen}      "Hudd";
# set LG_SITE host={view_Buffalo}        "BufGro";
# set LG_SITE host={view_Frankfurt}     "FRA";
# set LG_SITE host={view_Muehlhausen}   "MUH";

insert pdmuser
    {cls_Global}        "admini"
    {mix_WINDOWS}      "NTusr"
    "chzugw1313"       "PCone"
    "ebrl"              "abc"
;
```

The yellow marked lines above are those needed to define “special services”. In this case we defined some services to run on a “batch server” rather than the corporate. But even corporate server needs to know where these services run. It works exactly as the special services for different translations and rendering, which have to run on a specific server (translation server, NT).



Sample of Special Service Assignment in config editor

## 4.6 Platform specific

### 4.6.1 Windows:

If you change or create new system variables, which must be used in Windows<sup>9</sup> service startup for Teamcenter, you need to boot the Windows machine. Otherwise the server starting utility does not yet know about new variables and values. Thus some of the services may hang.

Typically “lamserv” hangs completely, if it cannot get the license file.

### 4.6.2 AIX

When changing any shared library, then you need to reboot the system or use the “slibclean” utility (available only as “root”).

<sup>9</sup>) Windows XP is not validated as any server, only as client. Use rather Windows 2000 or 2003.

## 5 European Language Support in Oracle

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).

UTF-8 character set is supported, but restricts classic client to English only. UTF-8 can be used only, if the underlying database is installed in UFT-8 mode. This mode can actually not be turned on in an existing database. And migrating existing data into UTF-8 is very hard to do. For more details read carefully "Teamcenter Enterprise Administrator's Manual for Parts and Documents" on this topic.

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*".<sup>10</sup>

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 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
```

## 6 Mail / Notifications

### 6.1 Variables in Config File

Set MAIL\_USERID "Teamcenter Name";

The Mail user is defined with his **Teamcenter user name**, not OS-Name !

It is recommended to comment out the two lines "SPOOLNOTIFS" and "ASYNCTNOTIFS". But there are no notifications, if MAIL is not turned on. To do so, set the variables "MAIL\_HOST", "MAIL\_USERID" and "MAIL\_LOG".

---

<sup>10</sup>) WE8ISO8859P1 : all upper case letters, digits are underlined here.

## 7 Hostnames and Communication

Host Names may be short or full (like “myhost” or “myhost.mydomain.county”). Upper case and lower case are not relevant for network software, but are relevant to Teamcenter Enterprise. Thus write in always the same way in Teamcenter Enterprise, and write it as **you get it from “hostname” function or VC\_HOSTNAME variable**.

### Facts

Hosts are identified in Teamcenter Enterprise by Host Names. These names have to be same in the registered Host object and in the config file.

Host names must match the names got by the “hostname” function in UNIX or the configuration variable “VC\_HOSTNAME” on Windows.

Any MUX (client or server) must be able to address by name on TCP/IP any other host’s MUX, he might need to contact. These calls go forth and back: client to server for a query, server to client for showing the result, any to any for file transfers.

Any client or server MUX checks incoming calls, whether the source TCP/IP address belongs to a hostname it knows (config file), and which is caller’s name within the content of the call. If either is not the case, the call is refused.

### Consequences:

The Teamcenter Enterprise Host Community must know each other by name and address. This can be achieved through complete “hosts”-files or by available DNS services, where even client hosts are maintained.

If DNS is used, these services must either cover the entire community; or they must be chained correctly. And it is highly important, that DNS is updated instantly upon new address lease.

If the community is split over different subnets, routing parameters must be set correctly on each host, so that it gets to gateways for all subnets. For checking of routing info use on UNIX the command “*netstat -r*”, on Windows “*route print \**” (if Microsoft stack is used).

### Identify communication problems:

Try “ping IP-address”: If it fails, change routing parameter; nothing else can help in any case. If this works, routing is setup correctly; go to next.

Try “ping hostname”: If this fails, name resolving is not ok. You need to change DNS name resolving or “host” file content.

If this works with the short hostname, then the problem must be within Teamcenter configuration (see later).

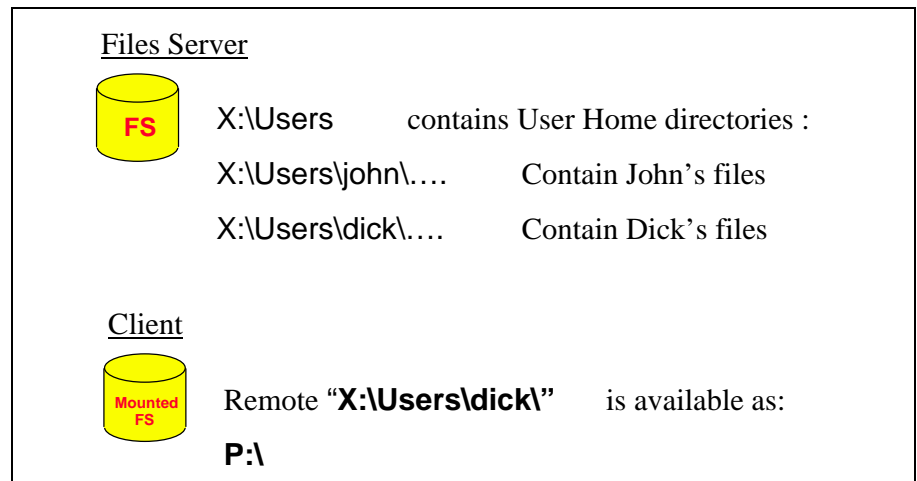
If this works with the fully qualified name only (long name), then there are different ways of corrections possible:

- a) Changes in DNS to server more globally through short names.
- b) Using long names in Teamcenter configuration as network addresses.

If “ping hostname” did work, try “muxping hostname”. If this does not work, then you need to verify Teamcenter configuration.



Basic assumption is:



**Caution:** If the file server is UNIX and the clients are any types of Windows, then you may not register the file system in Teamcenter for the Server. At least solution (A) needs a specific "proxy" host for eventual access, if the user is not logged in on a PC. A small PC, running W2K and Teamcenter MUX and DSP and OS\_SERV can be used for this purpose, if it runs constantly (unattended mode is OK).

Any solution having same drive name for each user, but different content depending on user, needs to differentiate the locations of different users on "P:\". Thus all registered work locations must mandatory be of the form "P:\username\worklocname". Nobody can have a registered directory directly in the P drive!

A better choice is the use if UNC (Uniform Naming Convention), like "\\servername\sharename".

### **Solution A:**

Define File system "P:\:" on a host group containing all clients PCs.

No "proxy" can be defined. Because no host exists, on which one directory can map all users' home directory.

As Teamcenter sees all users' P-drive as one single file system, it will no accept naming conflict. This solution requires that Teamcenter work locations cannot be directly within the home directory, but must be named like "P:\username\locationname" (i.e. "P:\Dick\PDMWorkLoc" for the sample above).

**Disadvantages:** Teamcenter Enterprise requires a proxy. Thus we need to declare one. But we know that each access through the proxy will result in a failure. As accesses through proxy are expected to be rare, we might accept this.

The system cannot prevent users from breaking the work location naming rule, which may result in operational problems for this user. But it can easily be solved by deletion of the relative location and correct re-definition.

**Advantages:** One single file system defined on a whole host group is enough for all PC work locations of that host group. This is a noticeable administrative advantage.

### **Solution B:**

Define a file system for each user as "X:\Users\username" on file server or proxy and as "P:\username" on the host group containing all associated PCs.

**Disadvantages:** Administrator must define two linked file systems (logged in as a super user) before the user can define his/her work location (must be done, when logged in as the user himself). This is a noticeable administrative effort.

**Advantages:** The system will automatically prevent users from breaking the work location naming rule, since no work location can be created outside of any registered and accessible "file system".

Any eventual access through the proxy host is granted perfectly. No exception from standard Teamcenter Enterprise behavior is required.

### **Solution C:**

Define a file system, which is physical base directory of all user's private data on a host group containing the file server (= proxy) and all clients in UNC notation. This is the simplest case for administration.

Sample: Define file system as "\\fileserver\home". User "Dick" may have automatically mounted "\\fileserver\home\dick\$" as his "P:" drive. But he needs to create his work location with path "\\fileserver\home\dick\$\teamcenter" rather than "P:\teamcenter".

Disadvantages: File server must run MUX, DSP and OS\_SERV as an admin account of the file server (having access to all user's data).

Advantages: Every thing runs correctly without additional administrators work for addition of users..

## **8.1 File Systems – Locations - Files**

Objects being "File System Items" (files, directories, bunch of files = aggregats, ...) are managed by Teamcenter Enterprise. They consist within this system of "meta data" (data about the real object, like access path) in a database and the "bulk data", which is the real content in a vault, work or team location.

### **8.1.1 Bulk Data Files**

The real file contents are generally not touched by Teamcenter Enterprise. They are treated as an integral entity, that must be available to some computer applications in order to be visualized, edited, printed, .... Teamcenter Enterprise is not intended to know much about the content of bulk data items. It allows its management, moving them around, check in and out, copying, and vaulting and versioning without name conflicts.

There are a few cases where Teamcenter Enterprise is touching file contents in order to grant correct file accesses across all participating hosts, respectively to deny access where inappropriate:

#### **❑ Files declared as OS depending binaries**

(Class OsDepBin and subclasses) may be accessed only on one specific operating system. When moving or copying a file away from that OS platform, then the valid OS Label is added in front of the file to the data; when it is moved or copied from another platform back to this OS type, then this label is taken away again.

This should grant the usability of the file on the required platform only.

*Typically Word files do not belong to this type of files, because they are usable on several OS types (like Windows and Mac).*

#### **❑ Files declared as ASCII text**

ASCII text files have different internal structures on different platforms. The most obvious difference is the representation of an "End of Line" between Windows and UNIX.

On file transfers Teamcenter will operate the same way as FTP with the ASCII option, i.e. it corrects the representation of "End of Lines" to be correct in the target system. (If non-text files are treated this way, you will loose data; thus never declare files as "Text" if its content must be invariant on different platforms.)

This behavior is often welcome where files, which are visualized on different platforms. But it implies that Teamcenter cannot access the same physical directory through different OS platform types.

### **8.1.2 File System part of other "File System Items"**

#### **❑ Directories**

Directories may be registered as "Directories" within a Work- or Vault-Location, Team-Location or Staging Area. Then they are treated similar like files. They cannot change names or content on moves and copies. These directories are considered to belong to a User, Vault, Team or External System.



Directories (not included in Locations) may be registered as “File Systems” in Teamcenter, or as Work- or Vault-Locations (if within a “File System” object). These types of Directories have some how administrative character (see File System Administration).

❑ **Links (UNIX Links, Windows Shortcuts)**

This type of objects has purely local meaning for Teamcenter Enterprise. They may partially even be registered and deleted. But Teamcenter Enterprise cannot yet deal with them for other actions, as they are very much OS dependent.

### 8.1.3 Meta Data

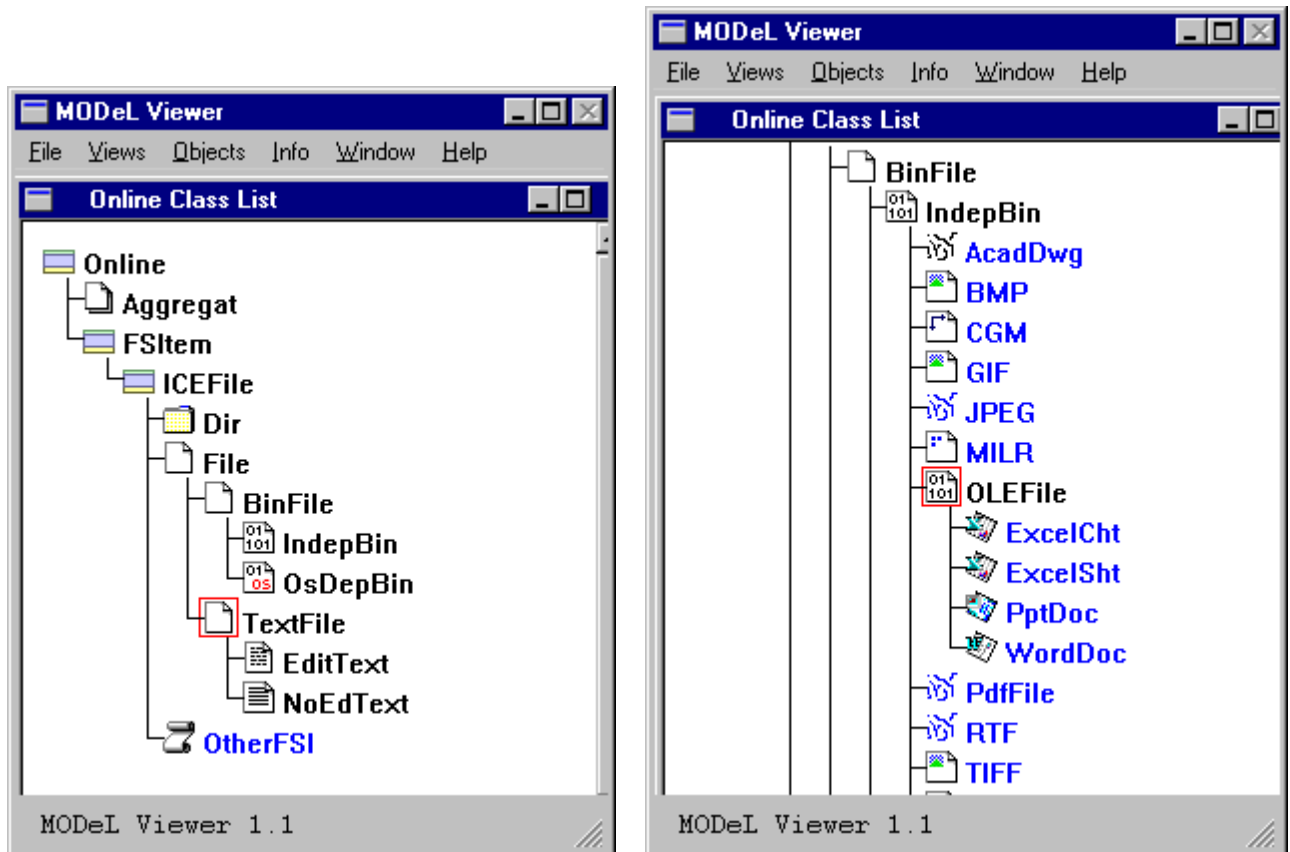
The meta data in the database is used to identify registered file system items, to organize the access within any form of distributed organizations.

Main attributes of this object type are their name, the current relative file path and the file type. The meta data object is used too for relating these files to other objects like documents, parts, projects, ...

Physical access to files registered as objects (except when in private work locations) is not possible otherwise than through Teamcenter Enterprise and its meta data. Thus meta data is used to grant access paths, access rights and confidentiality.

### 8.1.4 File System Item Class Structure

Here after you see the standard partial Teamcenter Enterprise file class trees, illustrating the text above.



## **8.2 Spaces and Locations**

### **8.2.1 Overview and Identification**

Any owned object (business and data items) must have an owner. Objects are either private (owned by a user, person) or they are public and protected (owned by a Vault) or semi-public (owned by a team) or external (owned by an external application, in a staging area). The ownership area is a logical space. Thus all files, document, parts are owned by s space, which is either private (user) or public (Vault) or semi-private or external.

The private space is a workspace; the public space is a Vault. Teamcenter automatically creates a Workspace for each user, when he/she logs first time into the system. Workspaces have always the same name as the user who is the owner. A Workspace is completely owned by its relative user. Standard Teamcenter Enterprise has a rule, which allows each user any actions upon objects he owns. A Vault is not a person as owner, but rather a secure space for permanent storage or public read access. A user may be assigned responsible for a Vault. If none is assigned, the Teamcenter application is the owner, represented by the "super user".

Team Folders are semi-private. A team has access according each user's team role to team's data. Usually the team leader is assigned as representation of the owner "Team-Folder". Upon team creation the administrator must create respective team folder(s) too.

An administrator creates vaults, as they are needed. If a system uses more than one Vault, this may reflect functional structure, geographical structure, departmental structure, progress evolution, data organization or process structures. The owner of a Vault is defined at creation time; normally this is the Teamcenter system itself, thus the "super user"<sup>14</sup>.

In distributed Teamcenter Enterprise each space (Workspace as well as Vault) is assigned to a User-Database, thus all objects in this space will be stored exactly in this database and only in this one, except data classes which are assigned by class to a Replica Set.

To store the real and physical associated files, each space must have at least one Location; Workspaces have Work-Locations, Vaults have Vault-Locations, Team folders have team locations, external systems have staging areas. We call these all "Locations".

Locations are completely owned by their relative space. The Locations can be on any registered host's disk, independent from where the relative database resides.

Locations may be compared to a user's home directory or root directories for data storage. There is no technical limit on the number of Locations that a space (Workspace, Vault, ...) may have assigned. The Location is physically a directory and MUST be owned on OS level by the OS user who represents the owner user in Teamcenter, i.e. Vault-Locations directories must be owned by the trusted user, Work-Locations must be owned by the user himself at OS level too.

### **8.2.2 Vaults and Vault Locations**

Only administrators of the respecting level may create (register) Vaults and Vault-Locations, teams and team folders, ... .

A functional application design has to decide how many Vaults are needed (use as few as possible) and what they are used for.

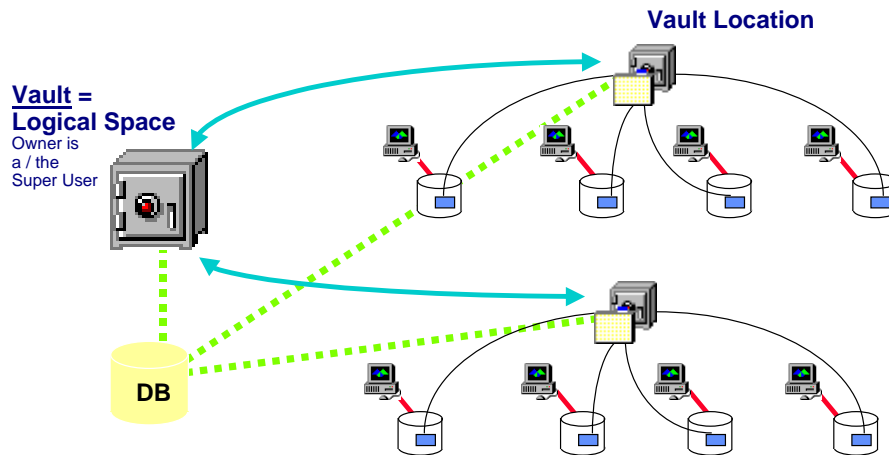
Functional and/or architectonic reasons may define the database each Vault is assigned to.

Functional reasons as well as performance considerations may decide on the number and residence of Vault Locations.

---

<sup>14</sup>) If you use any other name, you need to know what you do. You may get into troubles with accesses.

## Distributed Vaulting



A Vault may have locations on each site of a distributed installation.

Starting with release Metaphase 3.2 **Vault Locations can be replicated**: For each Vault Location an administrator can define, where it must have copies, and what is the replication policy. This means that a file's meta data are kept in just one database, while the file itself (physical bulk data) resides on several sites, and a given client is served by the "nearest" copy. Consistency and invalidation (checkout, checkin) is managed transparently by Teamcenter Enterprise. There is no master copy concept, but rather a peer to peer system.

Vault locations may be created by administrators of respecting level only within earlier defined "File Systems". It is an administrative task to watch over Vault Locations in short periods. Open new Locations (and add rules if necessary) when disk space is low, or when a Vault Location has several thousand files (this can block your backup/recovery procedure).

It is very much recommended to define a naming convention for Vaults and Vault Locations. As users sometimes must select the Vault and/or Vault Location, speaking names are highly recommended. If they do not help the user just by its name, then users will prefer to do overall queries, thus use more and more resources for the same work.

Naming conventions will even help to simplify the rule system (i.e. rule for all Vaults may have a condition like "\$obj.OwnerName like 'Vault\*' " or "\$obj.OwnerName like '\*Archive' ". Be aware that users and Vaults have a common key, thus it is not possible to create a Vault with the same name as a user.

The number of rule and conditions on your database may dramatically influence the performance of your system.

Vault Locations may not accept files until a "selection rule" does permits it. Selection rules have priorities to favors local Vault Locations over remote ones in distributed installations.

Use intelligent site names in the host definitions, so that you can select a priority for these rules based on the actual hosts' site name. The default location for a Vault is the one with the highest priority (lowest number); thus defaults may depend on geography and/or user.

Data access to Vaults and Vault Locations is driven by rules.

### 8.2.3 Work Locations

Each user can only create Work-Locations for himself. Nobody can create explicitly a Workspace. And nobody can create a Work-Location for somebody else than himself!

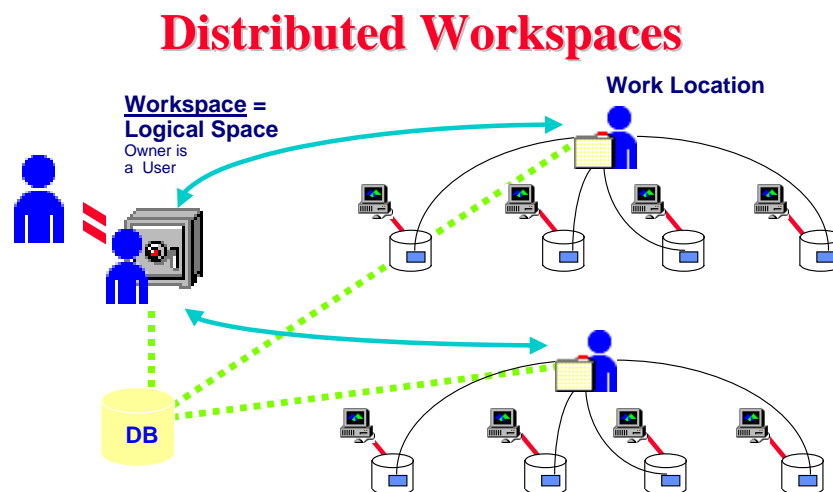
A user can create a work location only within “File Systems”, which were previously defined (registered) by an administrator and which are locally accessible on the actual user’s client host. It cannot be defined elsewhere.

*The designers in new Teamcenter Enterprise project often think, that it is to complicated for users to create their own Work-Locations, and that an administrator should create one for them. After a while all came back to the standard, because users need some times more than one, because administrative solutions are complicated.*

*It is a good advice, to learn the user in his initial training first to create his work location. Users working on different sites of the company may anyway need more than one work location, otherwise they are blocked. Users usually have the right on OS level to make as many subdirectories in their home directory as they want (i.e. to separate different work types), why should they not do the same in Teamcenter Enterprise?*

With the standard rule system nobody has the right (except super user) to deal with objects, which are owned by somebody else.

The user can on his decide, for each one of his work locations, if this location may accept files from other users or not.



The WorkSpace object is hidden in Teamcenter Enterprise, but it exists.

A WorkSpace can contain many Work Locations on different sites. Each location may be accessible from several hosts. (Shared devices)

### 8.3 File Systems

The root directories, where Teamcenter can deal with data, must be registered within Teamcenter as “File Systems”. This may be a physical file system or a directory that Teamcenter should look at as a file system.

File systems may be available (i.e. accessible) on a single host or on several hosts. If a file system is available on a group of hosts with identical file path, then it may be defined directly on a host group in Teamcenter. On host out of the group must be defined as proxy, so that Teamcenter services running on a host not contained in the group can have access (i.e. batch processes, etc.).

File Systems are root paths for locations, which then contain user-registered files and directories.

Due to reasons declared in chapter “8.1. File Systems – Locations - Files”, a file system can only be registered for sets and/or groups of host of same OS platform.

The access path for a file is stored in Teamcenter Enterprise in three parts. The three parts are composed together for each access. Physical access is then made through “OS\_serv”, which is launched under the name of the owner of the relative location.

The three components of the file path are:

- **Hostname and root path of “File System”** (stored in the “FileSys” object)
- **Path of Location relative to “File System”** (stored in the “AccessOn” Relation between FileSys and Location)
- **Relative path of file within location** (stored on the “FSItem” object [meta data of file])

Some samples (in colors as above)

```
Host 1 : /usr3/home/jim/workloc/list.txt
Host 1 : /usr3/home/bob/WL1/sd3/sample.pdf
Host 1 : /pdm/vaults/PDMarchive1/aab1234.txt
PC 12 : C:\Pdm\jim_wl1\samp.xls
NT 13 : \\server1\users\jim\workloc\samp.ppt
NT 14 : \\server1\users\jim\workloc\samp.ppt
PC 15 : \\server1\users\bob\pdm\wloc_1\samp.doc
```

Today many sites have a concept of user's home drives residing on a shared server disk. Teamcenter supports this type of storage being used for work locations: create a host group containing all workstations, which may have physical access (maybe limited by user name) to the shared drive. Then define a file system on this host group rather than on single hosts. Like this work locations on such file systems are accessible from many workstations, just like the usual user's home drives. Maintenance is made through simple add and remove of hosts to/from host group.

### 8.3.1 File System Definition

File Systems like hosts are administrative items. Only an administrator of the respective level can define them. They never are owned by a user or vault. They are just a registered part of the underlying operating system. Thus they never are relative to a user, but relative to a host or a group of hosts.

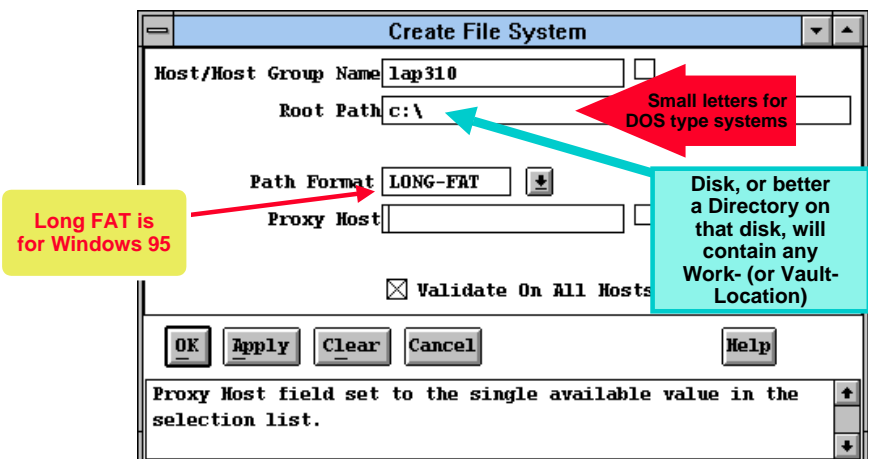
The link between a file and its location is maintained by an attribute on the file object. The link between a file system and a location is maintained by the relation ‘AccessOn’. The link between a File System and a Host or Host Group is stored as an attribute on the File System object. ‘MntFrom’ relation to tells Teamcenter, that two different File Systems are the same, are mounted on each other.

**Caution:** The primary haddisk is for security reasons no more allowed as a Teamcenter file system. You must use a directory within C: (i.e. “C:\pdm”)

Choose correct path format, which corresponds to physical implementation. If it is not correct, you may have access problems.

If a file system is defined on a single host or a host group, then the definition through this dialog is all to be done.

When creating a file system, Teamcenter will at least on the proxy host check the real availability and accessibility of the file system. If the respective check box is selected, the check is made on all hosts of the group. For this check the respective host must have a muxd and dispatcher running. The check is made through muxd and OS\_serv.



### 8.3.2 File System Mounts

A file system may be accessible on different hosts by different paths (typically NFS mounted files systems). This is communicated to Teamcenter Enterprise by the following actions:

Define each file system like stand alone file systems with its correct host name and root path. One of them should be the file system of the host, where the file system physically is located; this host is called the server host, having the server file system; while all the others are clients and client file systems.

Drag and drop all the client file systems on the server file system. This will create the “MntFrom” relations, so that Teamcenter can determine, than a file in such a file system is locally accessible an each of the hosts.

### 8.3.3 Proxy Host

As soon as a file system is accessible from more than one host, Teamcenter asks imperatively for a Proxy Host. It is intended, that this is the “server” host, i.e. the host, where the file system is physically located. To work properly this host must be a multi user system (where different users may be logged in simultaneously) like UNIX or newer Windows and must run a muxd and dispatcher with OS\_serv as trusted user. And OS\_serv must have access to all files and directories within the file system.

Up to version 3.0 of Metaphase the proxy was not used very often. The requirement was always in the manuals.

With release 3.1 this has changed for network performance reason. Now the files are accessed locally on the user’s client only for real local access. Any transfer between two locations (copy, checkin, checkout, move, transfer, shadow copy, ...) is now executed between the two proxies. If the proxies are the real servers, then this grants, that the file goes only once over the net. If this condition is not met, we may have the same situation as in 3.0: the file is going up to tree times over the network for a simple transfer.

For a file system on a single host, this host is automatically the proxy.

A file system defined on a host group requires a value for the proxy host.

On mounted file systems the server gets the proxy. For creation you have to drag the clients on the server, not vice versa.

**Caution:** In earlier versions of the product, if you have a file system on a host group and you enter a host as proxy, which has not or not always access to all work locations (user private areas), then no checkin or checkout was possible, because the system tried to do so on the proxies. Meanwhile an enhancement<sup>15</sup> was realized, which checks before every file access which host delivers the fastest access path. This was necessary to determine even on replicated Vault Locations, which is the fastest available access point.

Starting with release 3.2 an access policy (proxy or client) can be chosen on file system level by an administrator. The system works even with shared files systems without a proxy host (with obvious access limitation, if workstations are turned off).

The user interface does not allow creating File Systems without proxy. But file system may be created by loads without proxy, or the proxy can be cleared out through DB manipulation (be careful about replication). This works if you deny any proxy preference. Access to Work Locations on such File Systems are only possible for client action, generally not for server actions. The proxy must be cleared, before any Work Location is created within the file system, because the proxy will be copied onto AccessOn relations (which are normally replicated over several DBs).

**Caution:** if you use file systems on host groups without proxy host, then the processes cannot checkin (or just touch somehow) files from a location in such a file system, because the server program running on a server with “*advancer*” cannot get access to it. The user (resp. the client program on the client host) must checkin such files before submitting it or related business items into a process.

Vault Locations or Staging Areas **cannot** reside on File Systems without proxy.

---

<sup>15</sup>) Service “netprobe”

## 8.4 Shared drives versus “Replicated Vault Locations”

### 8.4.1 Shared Drives or mounted file systems

This is the standard way of telling Teamcenter, that a file system (or a directory) is accessible from more than one host. There is **only one physical copy** of the relative directory with its files. But this directory is accessible from more than one host.

**Several logical Host-File systems are one physical file system.**

If a file system is accessible from several hosts, but having the same root path on ALL of them, then the file system should be declared on a “Host Group”. This is typically the case for Windows shared drives, as well as some UNIX NFS file systems in CAD context.

If a file system is available from several hosts, but they use different “root paths” (mount points), then you have to declare each root path for a host as a single file system. Through drag and drop of client file system onto the server file system you will tell Teamcenter this mount structure. This is in general a usual UNIX NFS mounted file system.

### 8.4.2 Replicated Vault Locations

This new Teamcenter Enterprise feature allows having files physically replicated over different sites. This is to reduce access time in distributed environments. One logical file may reside as more than one physical copy on the Teamcenter disks. Meta data are kept in one database only (except for replicated file classes, which follow the replica set rules and is not recommended to do) and as one instance of the file.

**One logical-File exists as more than one physical file master copy.**

This feature includes a replication policy and is limited in this first version to entire Vault Locations. Each copy of a Vault Location must be registered on a File System like independent Vault Locations. Then you can create an “Is Replicated On” relation between them. This requires applying an already defined replication policy on that relation.

#### To turn on Vault Location Replication you need to:

- change the condition

**AllowBulkReplicationC := ( 1=1 );**

- Have two file systems ready.
- Recreate the rule cache (rulefile -rd admindb) ! This is mandatory !!!
- Create a Vaultloc "RelLoc1" in "@Vault\_Common" on first File System.
- Select the Vault Location, edit Relationship "Is Accessed From".
- Drop the second file system onto this browser (works only after rulefile !)
- Close this browser.
- Create a Replication Policy.
- Select the VaultLocation "RelLoc1", Edit Relationship "Is Replicated ON".
- Menu Create: "Replication Policy Assignment". Select Policy and first File System, OK.
- Menu Create: "Replication Policy Assignment". Select Policy and second File System, OK.
- If you do not both "Replication Policy Assignment" the replication goes only in one direction.

## 9 Data Integrity Mechanisms

### 9.1 Data Integrity Failure Sources with Teamcenter Enterprise

Generally Integrity Problems with Teamcenter Enterprise are due to behavior of human beings. The overall data management by Teamcenter standard is extremely robust under any conditions.

Stress tests have shown that Teamcenter can be stressed until performance is down to zero. But it is not possible to stress Teamcenter until it creates Data Integrity problems. Thus any such problems are due to either a human action with exceptional rights, or errors in the customization, or by failure of underlying systems and hardware.

#### ❑ **Bad Manipulation by an Administrator.**

Manual operations by operators during cleanups, reconfigurations and upgrades are the most often source for integrity problems: wrong file deleted; miss-typing directory names during disk reconfigurations, manual changes of ownership and access rights.

#### ❑ **MOdel changes through customization without correction of existing data.**

Customization adds required attributes which are relevant to behavior of objects. But existing data does not get these fields initialized.

Development should be required to deliver respective upgrade statements (sql, msql) or processes.

#### ❑ **Inappropriate use of low level DB API's in customization.**

All the API's of form XxxxDblItem (or XxxxDboObject) are at very low level. They mostly deal with an entry in one database. They do not always care about existing replications, relations or keys.

If these API's are used in a customization, they must be carefully controlled. They are the only API's that can create integrity violations.

**It is recommended to forbid their use in customizations!**

#### ❑ **System Failure due to coincident failures of network, hardware and software.**

If during a multi-DB transaction more than one failure of either the network, the hardware, the software or resources occur, then this can lead to a situation, where the auto-recovery can fail. This means that by external failures the system can neither complete a transaction nor roll back.

#### ❑ **Resource Problems.**

Network failures, network service failures, disk full, tablespace full, extends exhausted, SQLNet failure, Oracle failure, OS failure.

#### ❑ **NFS mounting failures.**

If Vault-Locations are accessible from several hosts (servers) via NFS, then it may happen, that the unmount (due to a problem on the NFS server) on an NFS client is not known to the Teamcenter services on the host. If files are now written into the Vault-Location on the NFS-client side, they are written locally in the directory, which normally is just a mount point.

During this time some files of the Vault Location are seen only by the NFS server, others only by the NFS client. Teamcenter cannot handle this exception.

After next boot with "automount" the files in the NFS clients mount point directory are no more seen or accessible. They are logically lost, even the physically exist on a not visible disk.

**Physical accessibility of Vault Locations by users and Teamcenter clients is a noticeable security risk and should not be allowed.**

#### ❑ **Inappropriate restore manipulations.**

Restore of single tablespaces or single site Oracle instance as well as restore of entire single Vault Location in a distributed environment has a high risk of getting data inconsistent.

Following actions generally result in transactions of more than one Teamcenter Database:

- Create, modify and delete of replicated items.
- Create, modify and delete of items with a key.
- Checkin, checkout, revise and transfer of items.
- Create, modify and delete of relations, if at least one of the related items is replicated, or if the two related items are in different bases.



- If the items are files (or other file system items), then the database transactions must be synchronized with the file system actions (Vault Location).  
Any partial restore may bring back only part of the data in a previous state.

## **9.2 Instrumented Integrity Checks**

**Note:** Integrity Check Utilities are included in Teamcenter Enterprise. This section does not apply to any pre 3.1 Instance of Metaphase.

It is customer's responsibility to use these utilities at reasonable frequency.

The utilities are described in the "Network and Database Configuration Guide" manual, chapter 6 "Administering Distributed Databases".

### **9.2.1 Utilities**

The integrity check utilities are DB vendor dependent. This means there is a separate binary for each DB vendor type database. The utilities are called as a concatenation of the utility name and the DB vendor abbreviation.

The utilities are:

<b>kyvldXXX</b>	Key Integrity validation utility
<b>bnvldXXX</b>	Binary data validation utility
<b>rlvldXXX</b>	Relation and Replication data validation utility

where XXX is the DB vendor abbreviation:

ora	Oracle
inf	Informix <sup>16</sup>
mss	Mircosoft SQL Server
syb	Sybase <sup>17</sup>

In the following we will discuss the Oracle version of these utilities. The only difference is the postfix of the utility names.

#### **9.2.1.1 Key Integrity Validation Utility "kyvldora"**

The "kyvldora" utility verifies that for every key in the key table, there is an object in the Teamcenter system that has that key value in its attribute. At the completion of the validation, the "kyvldora" utility writes a report file (keyvalid.rpt) and an SQL script file (keyvalid.sql).

The database administrator can examine the SQL script file and, optionally, run the "msqlora" utility to correct the inconsistencies.

The "kyvldora" utility performs the following checks:

- Compares the key values in the key table with the values of attributes in the class tables to verify they are the same.
- Logs a message to the report file indicating the number of key values in the key table that are different from those in the classes table. Writes an SQL delete or insert statement to the SQL script file.
- Retrieves the key name for the class name specified on the command line. Validates all key values for that key against all classes and against its persistent children that have that key. If no class name is specified, validates all key values of every key.

If the class is clustered and there is an optimized migration file (that is, migration is not complete), the utility searches for rows in both the old class table and the new clustered table. The key table cannot be clustered.

---

<sup>16</sup>) Informix support was dropped on release after Metaphase 3.1.

<sup>17</sup>) Sybase support was dropped with Collaboration Foundation.

In distribute environments “kyvdora” should be run against a user data base. Do not run it against the key data base (in this case we detected the keys of SU-DB to get dropped).

### **9.2.1.2 Binary Data Validation Utility “bnvdora”**

The “bnvdora” utility compares the binary attributes in the input class object (and all of its persistent children class objects) with those stored in the “zbindb” table in the database. From this comparison, the utility generates two files:

- ❑ Report file (binvalid.rpt) describing the results of the validation.
- ❑ Script file (binvalid.sql) containing the SQL statements to correct the discrepancies reported in the report file.

As the database administrator, you can examine the SQL script file and determine whether the statements are correct. If the SQL statements are correct, he can apply it to the database.

The “bnvdora” utility performs the following checks:

- ❑ For each object of the input class and its children, prints out the number of binary attributes defined in the class definition of the MODEl file.
- ❑ For every database object of the class (if there are binary attributes in the class), validates that the “zblob” field is set to plus (+); otherwise, sets the “zblob” field to minus (-). If there are no binary attributes and there are phantom rows in the “zbindb” table, generates the SQL delete statement.
- ❑ Compares the type of binary attribute data with that in the “zbindb” table. If they are inconsistent, writes a message to the report file. This error condition does not generate SQL statements to correct the problem.
- ❑ If the “-r” flag is specified, searches for binary rows in the “zbindb” table that do not have corresponding data rows in the class table and generates an SQL statement to delete the rows.

If the class is clustered and there is an optimized migration file (that is, migration is not complete), the “bnvdora” utility searches for rows in both the old class table and the new clustered table.

**Remark:** In some of the Metaphase 2 and 3 version the standard software did not always delete the binary attribute BLOBs too upon object deletion. The resulting “binvalid.sql” will contain statements to delete such orphan records.

### **9.2.1.3 Relation and Replication Data Validation Utility “rlvdora”**

The “rlvdora” utility verifies that the left and right objects and the relation object itself are kept in the correct databases. It generates a report file (relvalid.rpt) describing the results of the validation. This utility does not produce an SQL script file.

The “rlvdora” utility performs the following checks:

- ❑ Every row in the input relation class table is replicated correctly in the databases of its left and right items.
- ❑ For each relation object, retrieves the left and right objects from the database(s) indicated by the “DBNAME1” and “DBNAME2” fields, respectively, to verify their existence. If the object is replicated (that is, the “DBNAME” field contains an asterisk; the existence of the object is validated on all replicated databases for its class.
- ❑ For a relation object that is in an extended relation network, verify that the relation object exists in all of the replicated databases of the network. It also locates any extended relation object that belongs to the same network as the relation object that might be missing from any of the replicated databases.

If the relation class is clustered and there is an optimized file (that is, migration has not been completed), the utility searches for rows in both the old class table and the new clustered table.

## **9.2.2 Reasonable Frequency**

The utilities described in the previous section are resource consuming. Thus they should not run more often than really required.

If a file is lost in your system, within a week somebody may remember about what it was. If you discover this after 3 years, when you need the information, there is not much hope for a clean recovery. Thus the utilities will have most benefit, if they run in regular periods of a length, where people may remember, what they did.

The Solomonian guess for the period is: AS SHORT AS NECESSARY, AS LONG AS POSSIBLE. Probably a period between a week and a month could be appropriate.

## 9.3 Non Instrumented Checks

### 9.3.1 Database – Vault Location Consistency

A very important regular check, for which there is not yet a utility, must verify the consistency between database and Vault Location content:

- Does every file registered in the database physically exist in the Vault Location?
- Is every file in a Vault Location directory registered in the Database?

It is very wise to check this once a month. Missing files in this context is the most probable case for a successful partial restore.

### 9.3.2 Data Dictionary – Database Schema Consistency

Update of either Teamcenter standard release or Site Customization generally changes the data dictionary. To adapt the data base schemas, a utility calculates schema differences on the base of data dictionary differences.

This has two important impacts:

- Together with each database backup we not only need the Vault Location contents, but the data dictionary file too (file „tmti.prd“)<sup>18</sup>.
- For each upgrade do preserve the old data dictionary file for calculation of schema differences.

If ever you database schema and data dictionary are out of synchronization, it is very hard to get it synchronized again. Some private repair utilities exist for this purpose.<sup>19</sup>

If the „updatedb“ utility creates a file „alterdb.msql“, then apply it to all databases or save it under a different name. It will be overwritten next time and same modification statements will not be recreated.

## 9.4 Recovery Options

### 9.4.1 Full Recovery

**A full recovery consists in full restoration of ALL Teamcenter databases (all sites) of a Teamcenter instance, restoration of the data dictionary (*tmti.prd*) and of ALL Vault Locations.**

You need to decide about the restoration of Work Location contents.

### 9.4.2 Partial Recovery

Partial recovery is advised only as:

- Single file restore, if a registered file is missing in the Vault Location.

---

<sup>18</sup>) It is extremely dangerous, to have more than one file „tmti.prd“ in the context of a Metaphase instance. Depending on the directory, in which they are started, program may use one or the other; this creates system hangs in the best case, and data inconsistencies in the worst case.

<sup>19</sup>) I may deliver you repair scripts.

Be aware that normally file names in Vaults are randomized, while users see in the browsers the original file names.

The attribute "RelativePath" contains the actual physical file name.

The Attribute "Working RelativePath" contains the file name the user sees, and which is given to the file in a Work Location upon CheckOut or ShadowCopy.

- Single file restore, if a user wants back an older version of a file in his Work Location.

All other partial restores are strictly forbidden.

## **10 Upgrade of Metaphase 3 Instances created under Metaphase 2**

### **10.1 Configuration File**

For any upgrade to Metaphase 3.2 or later the config file has to be in new "cfgedit" format. Manual additions are still possible within the "site.cfg". However it is recommended to solve as much as possible within the normal "config.cfg" through "cfgedit2", which helps you to remain upward compatible. Later Teamcenter version were planned to have a binary config file !!!

Create your local variables or maps not by an editor (like vi or notepad) in the config file but rather through the templates in the "toccus.dat" file, as described in Teamcenter manuals, if they are required by the customization. Additional info about your configuration may be added as additional rows in the "cfgedit2".

### **10.2 File "toccus.dat"**

Starting with release Metaphase 3.2 installation and upgrade procedures have been simplified tremendously. But they depend now on correct information in "toccus.dat" for any custom module "CUS". This file includes information about what your customization contains, if it needs a separate database, replica sets, module dependencies as well as information about alternate source location (opposite to standard Teamcenter installation CD). For more see Teamcenter customization manual and the sample "toccus.dat" in this document, chapter «4.4 Sample "toccus.dat" file» on page 15.

### **10.3 Distributed Databases**

Starting with Teamcenter the Distributed Option installs fewer databases. There is no more LCM-DB. Its content goes to AdminRs replica set, thus into the admin DB. On the corporate server no OPS DB is created, the OPERATE-classes go into the SU-DB.

It is recommended to migrate your environment to this concept at a moment of your choice (before or after the upgrade to 3.2 or later). The standard upgrade process anyway does never do this for you (i.e. Teamcenter processes maintains whatever DB topology you have in place).

## **11 Teamcenter Instance Migration / Data Migration**

### **11.1 Definition of the TERM "migration"**

**Create a copy of an entire Teamcenter System.**

This can be

- 1) A restore of a backup system on its origin place.
- 2) Create a new copy of a running system on different Hardware (but same structure!)
- 3) Create a new copy of a running system on same hardware.

We need to distinguish the backup type we have for the Oracle database:

- a) Export
- b) File dump of database and control and init...ora files, when Oracle is shut down.

In the following we discuss each time (a) and (b) separately. The three types have different constraints. The most simple case is no. 1.

## 11.2 Scenarios

Case	a) Export	b) File Dump
1	<ul style="list-style-type: none"> <li>• create the Oracle Table-Spaces and Users with exactly the same names as they were before (datafile organization may be different) on a running Oracle instance with same SID.</li> <li>• import on Oracle the export file.</li> <li>• restore all physical directories containing Vault-Locations or Work-Locations.</li> </ul>	<ul style="list-style-type: none"> <li>• install oracle with no database (if not still there)</li> <li>• restore all the Oracle files as dumped.</li> <li>• start oracle.</li> <li>• restore all physical directories containing Vault-Locations or Work-Locations.</li> </ul>
2	<ul style="list-style-type: none"> <li>• create the Oracle Table-Spaces and Users with exactly the same names as they were before (datafile organization may be different) on a running Oracle instance with same SID or different.</li> <li>• import on Oracle the export file.</li> <li>• Update config file to reflect new hosts.</li> <li>• If SIDs are different and Teamcenter instance is not mono-base, update the DB-links.</li> <li>• restore all physical directories containing Vault-Locations or Work-Locations into a similar file system structure. If file systems do not have same root paths as the original, then the Teamcenter File System object need manually be updated to new root paths as soon as Teamcenter is running again.</li> <li>• Prepare an objload file to load a host object with name of the new corporate server. Load it.</li> <li>• Start Teamcenter, set up any host not yet known as a host object.</li> <li>• Update all file system objects to reflect the new hosts (and eventually root path, if changed in restore step).</li> </ul>	<ul style="list-style-type: none"> <li>• install an oracle instance with no database</li> <li>• restore all the Oracle files as dumped.</li> <li>• start oracle.</li> <li>• If not mono-base, update the DB links.</li> <li>• restore all physical directories containing Vault-Locations or Work-Locations into a similar file system structure. If file systems do not have same root paths as the original, then the Teamcenter File System object need manually be updated to new root paths as soon as Teamcenter is running again.</li> <li>• Prepare an objload file to load a host object with name of the new corporate server. Load it.</li> <li>• Start Teamcenter, set up any host not yet known as a host object.</li> <li>• Update all file system object to reflect the new hosts (and eventually root path, if changed in restore step).</li> </ul>

Case	a) Export	b) File Dump
3	<ul style="list-style-type: none"> <li>• create the Oracle Table-Spaces and Users with exactly the same names as they were before (datafile organization may be different) on a running Oracle instance with a <b>different SID</b>.</li> <li>• import on Oracle the export file.</li> <li>• If the Teamcenter instance is not mono-base, update the DB-links.</li> <li>• restore all physical directories containing Vault-Locations or Work-Locations into a similar file system structure, but on <b>different mount points</b>.</li> <li>• Start Teamcenter, set up any host not yet known as a host object.</li> <li>• Update <b>all file system</b> objects to reflect the new root path (and eventually host, where necessary). If the file system structure is no more similar or Vault/Worklocation directories do no more have same names, then it may be necessary to update the respective paths in the Location Objects.</li> </ul>	<ul style="list-style-type: none"> <li>• Oracle cannot be restored like this !</li> </ul>

**Caution:** Teamcenter registered file systems of different Teamcenter instances may not include each other on the physical disk level. If you do not respect this rule, there is a high risk for corrupt relations between data base and disk storage.

## 11.3 Upgrade Migration

### 11.3.1 Upgrade of a Test System

The recommended way of migration a test environment from one Teamcenter release to another is to do it the same way Teamcenter recommends for productive environments.

This means that you do exactly the same steps to bring your test, training or validation system (including the relative data) to the next release.

If you use the formula of having a “backup dump” with an initial state of such an environment, then bring it back to this initial state first. Then do the standard upgrade and take a new “backup dump” for initial state.

### 11.3.2 Data Migration from a lower release to a higher one

This chapter describes, how you proceed, if you install a new test, training or validation instance of Teamcenter at the newest release level and want to feed in a copy of existing initial data or a copy of your production data, which is at a lower release level.

The starting point is having your new (higher level) system running and functioning; and you need a “Backup dump” as used in 11.2 Scenarios and 11.3.1 Upgrade of a Test System from the system you want to export the data. This new system’s oracle installation must use the same user names (Teamcenter DB names) and table space names.

**Actions to perform on the new system:**

- Stop Teamcenter.
- Save the tmti.prd file as tmti.<new level>.
- Delete all oracle tables on the new system. (By an sql script or the “dropdb” utility of Teamcenter)
- Load all data (Metadata and Files) as mentioned in 11.3.1 Upgrade of a Test System.
- Save the tmti.prd file of the old system as tmti.<old level>.
- Execute “rulefile -r [-d <admindb>”
- *Single base:* do “updatedb -b -o tmti.<old level> -n tmti.<new level> -t <index space name>”;
- *Distributed bases:* create the DB connection file as described in Teamcenter Upgrade Manual, cd to “\$MTI\_ROOT/install/upgrade” and execute “upgrade1 -t1 -s1”.
-

### **11.3.3.1 The Standard**

The Teamcenter standard upgrade strategy allows an upgrade period, which allows a limited time of parallel use of both, the old and the new release level. Thus two Teamcenter instances with **different IP-Port** number and **different data dictionary** (tmti.prd) are active on one same set of data bases.

During this parallel phase you may absolutely not do any other upgrade (i.e. a further upgrade of your customization)! But it allows to upgrade servers and clients in a progressive manner during this phase, without the need of an overall shut down for the whole phase.

An overall shut down is needed only at the final phase, when you turn off the old level and allow the new level to use the full new functionality (to run Upgrade2). A partial shut down is needed on servers and clients to switch over (or add) to use the new release.

#### **Summary:**

**Advantage:** This way of upgrade allows complicated upgrade, during longer time, to do with only two short interruptions at each location.

**Disadvantage:** The new release needs a new trusted user (os user) and a new IP port number.

### **11.3.3.2 The Alternative**

It is possible to upgrade the system in place, at some conditions. In this case you do not need an additional trusted user account, not an additional IP port number. But your 772 he old ansable from start to upgrade to the real end of all procedures. In case of failure you may have to restore the system backup from before the start of upgrade.

It is recommended to well study the release bulletin and upgrade manual before planning of such an upgrade.

#### **Following the main steps on the corporate server:**

- Stop Teamcenter and Oracle.
- Backup data base, \$MTI\_ROOT and all Work/Vaultlocations.
- Start Oracle.
- Cleanup \$MTI\_ROOT, delete what should not be there.
- Save your tmti.prd as "tmti.<old level>".
- Unpack the INSTALL.TZ and CFGEDIT.TZ into \$MTI\_ROOT, respectively \$MTI\_ROOT/install.
- Add your custom module name in "insname.dat".
- Call cfgedit2 and use the "upgrade" option.
- cd to "\$MTI\_ROOT/install/upgrade".
- Start mux and dispatcher.
- Execute "upgrade1 -t1 -s2345<sup>20</sup>"
- Execute "rulefile -r [-d <admindb>]"
- Stop and restart mux and dispatcher.
- Execute "upgrade2 -t1 -s12345<sup>20</sup>"
- Execute "rulefile -r [-d <admindb>]"
- Stop and restart mux and dispatcher.



### **11.3.3.3 CAVEATs at Metaphase Release 3.2**

Read carefully chapter 5 of Upgrade manual. It requires several changes in your config file.

Due to the disappearance of PSM module you need to observe a couple of additional rules. This applies for both the standard and the alternative way.

Content of old PSM is now split into OMF and APC. If you did not have installed APC before, the only correct way is to add under your actual (old) level first the APC module, and do the upgrade later. Not respecting this sequence may lead to problems. If this for some reason is not possible, call the help of SDRC or a Teamcenter expert.

If the alternative method of upgrade is used (see “11.3.3.2 The Alternative”), you need to copy the old file “insdone.sav” into “selected.dat” and “loaded.dat”. It must mention each installed module, except PSM, even if it was installed before.

There are additional cleanups to be made in all files, where PSM is mentioned. Delete all “\*psm\*” files from the install, lib and bin directories, as well as the psm directories in meta and support.