



目 录

目 录.....	2
第 6 章 Tuxedo 配置相关文件.....	3
6.1 几个关键环境变量.....	3
6.2 系统配置文件 UBB 及其内容.....	4
6.2.1 *RESOURCES 段的配置.....	5
6.2.2 *MACHINES 段的配置.....	6
6.2.3 *GROUPS 段的配置.....	7
6.2.4 *NETWORK 段的配置.....	7
6.2.5 *SERVERS 段的配置.....	7
6.2.6 *SERVICES 段的配置.....	8
6.2.7 *ROUTING 段的配置.....	9
6.3 域配置文件 DMCONFIG 及其内容.....	9
6.3.1 域(Domain)简介.....	9
6.3.2 Tuxedo 域划分原则.....	10
6.3.3 域(Domain)的功能.....	10
6.3.4 Tuxedo Domain 配置:	10
6.4 日志文件 ULOG.....	11

第 6 章 Tuxedo 配置相关文件

Tuxedo 作为企业级 OLTP 的产品，一样不例外，需要进行相关配置，但其配置总体来讲比较简单，易学易懂。

6.1 几个关键环境变量

Tuxedo 系统在建立、启动和运行的过程中，要从环境变量中读取一些配置参数。

下面用表格的形式对变量的含义进行说明：

环境变量	含义和用法
TUXDIR	Tuxedo 系统的安装路径，只要是设置 TUXDIR，Tuxedo 的 build 系统命令就会自动到 \$TUXDIR/include 和 lib 目录下去找到头文件和库文件。 例如：TUXDIR=/root/bea/tux; export TUXDIR
APPDIR	应用程序的存放路径。Tuxedo 启动 SERVER 时，会在 \$APPDIR 下查找执行文件。 例如：APPDIR=/root/simpapp; export APPDIR
TUXCONFIG	应用程序的二进制配置文件名。 例如：TUXCONFIG=\$APPDIR/tuxconfig; export TUXCONFIG
PATH	要包含 Tuxedo 的 bin 目录。 例如：PATH=\$TUXDIR/bin:\$JAVA_HOME/bin:\$PATH; export PATH
INCLUDE	仅用于 NT 平台，用于指定 C 语言头文件
CC、CFLAGS	指定 C 编译器和编译选项，如果系统没有使用操作系统的编译器(如 gcc)就需要指定 CC 变量 例如：CC =gcc; export CC
LD_LIBRARY_PATH SHLIB_PATH LIBPATH、LIB	这几个变量的作用是一样的，需要包含 \$TUXDIR/lib 目录及其它公共库路径。 LD_LIBRARY_PATH 用于一般 UNIX 系统； SHLIB_PATH 用于 HP—UNIX 系统； LIBPATH 用于 AIX 系统； LIB 用于 NT 平台。

表 6-1

在 UNIX 平台下安装 Tuxedo 系统后，可以在安装目录中找到一个 tux.env 的脚本，执行这个脚本就可以完成一些最常用的环境变量设置。

Tuxedo 客户端环境变量：

环境变量	含义和用法
TUXCONFIG	本地客户机加入 Tuxedo 系统时，要从这个配置文件中查找公告板信息，仅对本地客户机有意义。
WSNADDR	远程客户机要加入 Tuxedo 系统时，要从当地的 WSNADDR 环境变量中读取服务器上 WSL 进程监听的 IP 和端口，仅对远程客户端有意义。
WSTYPE	一个字符串，用于指定/ws 的机器类型，仅对远程客户端有意义。 当 WSTYPE 和服务器的 TYPE 一致时，将不对远程客户端和服务器的数据传送进行编码/解码，以提高性能。

表 6-2

6.2 系统配置文件 UBB 及其内容

本节讲述配置文件 ubb，描述 Tuxedo 应用程序的运行环境和部署方式。

下面是一个简单的 ubb 的配置模版：

```
# (c) 2003 BEA Systems, Inc. All Rights Reserved.
#ident "@(#) samples/atmi/simpapp/ubbsimple $Revision: 1.7 $"

#Skeleton UBBCONFIG file for the Tuxedo Simple Application.
#Replace the <bracketed> items with the appropriate values.

*RESOURCES
#IPCKEY <Replace with a valid IPC Key>

#Example:
IPCKEY 123456

DOMAINID simpapp
MASTER SITE1
UID 501
GID 1001
PERM 0666
MAXACCESSERS 100
MAXSERVERS 50
MAXSERVICES 100
MODEL SHM
LDBAL N

*MACHINES
```

```
DEFAULT:
    APPDIR="/home/landingbj/simpapp"
    TUXCONFIG="/home/landingbj/simpapp/tuxconfig"
    TUXDIR="/home/tuxedo"

#Example:
#    APPDIR="/home/me/simpapp"
#    TUXCONFIG="/home/me/simpapp/tuxconfig"
#    TUXDIR="/usr/tuxedo"

landingbj    LMID=SITE1

#Example:
#beatux     LMID=simple

*GROUPS
GROUP1
    LMID=SITE1    GRPNO=1
    OPENINFO=NONE  CLOSEINFO=NONE

*SERVERS
DEFAULT:
    CLOPT="-A"

simplserv    SRVGRP=GROUP1  SRVID=1

*SERVICES
TOUPPER
```

示例 6-1

6.2.1 *RESOURCES 段的配置

*RESOURCES

(1) IPCKEY 123456

IPCKEY 是 IPC 资源的标识符, 它取值在 32768~262143 之间的整数。

单机环境 (SHM) 下, 标识公告板的入口地址。多机环境下 (MP) 它标识着 DBBL 进程的消息队列名, 同一台主机的任何两个应用系统, IPCKEY 值不能相同。

(2) MASTER SITE1

MASTER 指定了做为主节点的逻辑主机标识。因为例子是单机模式, 所以只有一个 SITE1。如果是多机模式, 即下面的属性 MODEL 为 MP 时, 可以写为 “MASTER SITE1, SITE2”, 这里 SITE1 为主节点, SITE2 为备份节点。

(3) MODEL SHM

单机模式 (SHM) 还是多机模式 (MP)。单机配置指当前配置中只能有一台物理主机, 即 MACHINES 段中只能有一个主机定义; MP 指当前配置中有两台以上的物理主机, 即 MACHINES 段中包含两个以上主机。

MP 模式下会把 MASTER 节点的配置复制到非 MASTER 节点主机上, 并实行统一管理。

(4) UID 501 GID 1001 PERM 0666

这三个参数控制着 Tuxedo 应用程序对系统 IPC 资源的存取权限。

UID 和 GID 是可以对该 Tuxedo 应用系统进行管理的用户 ID 和组 ID。如果不指定, 默认为执行 `tmloadcf` 的用户 ID 和组 ID。

PERM 定义了 Tuxedo 应用程序对系统 IPC 资源的访问权限, 默认值为 0666, 即任何用户都具有对 Tuxedo 为该应用创建的 IPC 资源进行读写的权限。

(5) MAXACCESSERS 100

定义了公告板的最大容量, 即同时可以容纳的客户机和服务器的数量, 取值范围 0~32768. 默认值为 50。

(6) MAXSEREVERS 50

指定公告板可以容纳的最大服务器数量, 取值在 0~8192 之间。

(7) MAXSERVICES 100

指定公告板可以容纳的最大服务数量, 取值在 0~32768 之间。

(8) LDBAL N

指定是否启用负载均衡算法。

列举完上例中的相关参数后, 其实在*RESOURCES 段还有很多隐含的默认参数, 如果有兴趣的话, 用“`tmunloadcf`”命令可以看到完整的列表, 下面对这些参数主要以默认值为例进一步继续阐释:

(9) MAXGROUPS 100

指定公告板可以配置的组的数量, 取值在 100~32768 之间。

(10) MAXGTT 100

定义最大并发全局事务数. 取值在 0~32768 之间。

(11) MAXCONV 64

定义了最大并发会话数, 取值在 1~32768 之间。

(12) MAXNETGROUPS

指定公告板中可以配置的网络组的数量。

(13) SYSTEM_ACCESS FASTPATH

可取值 FASTPATH 或 PROTECTED 默认值为 FASTPATH。

指示连接到服务器进程中的 Tuxedo 的内部库将以什么样的方式访问 Tuxedo 的内部表。

(14) SCANUNIT 10

设置系统健康检查的时间单位。

(15) SANITYSCAN 12

设置 BBL 对系统做健康检查的时间间隔. 即每隔 `SCANUNIT * SANITYSCAN` 秒做一次健康检查。

(16) BBLQUERY 30

用于设置 DBBL 定期检查 BBL 状态的时间间隔。

(17) BLOCKTIME 6

设置客户端发出请求后, 在得到响应之前, 可以等待的最长时间为 BLOCKTIME * SCANUNIT, 默认值为 6, 即 60 秒。

6.2.2 *MACHINES 段的配置

(18) landingbj LMID=SITE1

定义了主机名和逻辑主机标识. 逻辑主机名和主机名一一对应, 此逻辑主机名会被 *RESOURCES 段中的 MASTER 参数和 *GROUPS 段中的 LMID 参数引用到。

(19) APPDIR

定义了 Tuxedo 应用程序的存放路径。

(20) TUXCONFIG

定义了二进制形式的 Tuxedo 配置文件的存放路径。

(21) TUXDIR

定义了 Tuxedo 在当前主机上的安装路径。

列举完上例中的相关参数后, 其实在 *MACHINES 段还有一些其他有意义的参数, 下面对这些参数进一步继续阐释:

(22) MAXWSCLIENTS

指示最多可以多少个工作站客户端连接到当前主机, 取值在 0~32768 之间。

(23) ULOGPFX

指定系统日志在磁盘上的存储位置和文件名。

6.2.3 *GROUPS 段的配置

(24) GROUP1 LIMID = SITE1

指定一个组名, 和对应的逻辑主机, 表明当前组内所有服务器都将部署到这台主机上

(25) GPRNO = 1

为当前组指定一个编号, 任何两个组编号不能相同

(26) TMSNAME

指定事务监控器 (Transaction Manager) 进程的名称

(27) OPENINFO

该串提供了打开 RM 连接所必须的参数

(28) CLOSEINFO

该串指定关闭 RM 时需要的提供的参数

6.2.4 *NETWORK 段的配置

定义了网络相关的配置。如果 *RESOURCES 段中的 MODEL 参数定义为 MP, 并且 OPTIONS 参数中包含 LAN, 则必须使用 *NETWORK 段来对每一台主机的 tlisten 和 BRIDGE 进程监听的端口进行配置。

6.2.5 *SERVERS 段的配置

以示例 6-1 中的 simpsserv 参数为例：

(29) SRVGRP = GROUP1

指定服务进程所属的组，这些组必须在*GROUPS 段中定义过的。

(30) SRVID = 1

指定服务进程的编号，同一个组的两个进程，编号不能相同。

列举完上例中的相关参数后，其实在*SERVERS 段还有一些其他有意义的参数，下面对这些参数继续阐释：

(31) CLOPT = “-A -- -xyz”

为服务进程指定命令行参数：“--”前的部分由服务器进程 main() 函数捕捉处理，后面的部分由 tpsvrinit() 捕捉处理。

“--”前面可配置的选项如下：

-A: 初始化并公告服务进程中的所有服务；

-s: 指定一个或多个服务名；

-e: 指定一个文件名，服务进程输出到 stderr 的信息将被重定向到这个文件中；

-o: 指定一个文件名，服务进程中输出到 stdout 的信息将被重定向到这个文件中；

-p: 告诉 Tuxedo 如何依据 SERVER 负载情况启动新进程，和终止进程；

-r: 这个选项用于指示服务进程，把服务的执行情况记录下来，写到 stderr 中。可以使用 txrpt 命令(txrpt -n s1114Cfm -d06/19 -s00:03 -e14:28 <stderr)来分析这些执行记录；

-t: 用于指定 WSL, GWTDMAN 和其它服务进程与早期版本的兼容性。

(32) MIN MAX

启动服务器的最小和最大进程数。

(33) RQADDR = “Q1”

指定当前进程的请求队列名。

若不指定 Tuxedo 会按照组编号和进程编号的组合为它指定队列名。

(34) CONV = Y

Y 表示会话服务 N 表示不是会话服务。

(35) REPLYQ = Y

是否为当前进程创建单独的响应队列，建议取值 Y，此响应队列主要存放此服务作为客户端调用其它服务，其它服务的响应数据。

6.2.6 *SERVICES 段的配置

一般常见的*SERVICES 段就如上述例子，就是列出了对外发布的 Service 们的名称，其实不列出也没关系，因为每个 Server 启动时，会把自己指定的 Service 发布出来，写不写在这段没有实质影响，但下面的一些额外附加参数(以默认值为例)，决定了这个段落的意义：

(36) LOAD = 50

定义当前服务的负载因子, LOAD 越大, 说明服务的负载越大, 处理时间越长。

(37) PRIO = 50

指定当前服务的优先级因子, PRIO 越大, 说明当前服务的优先级越高, 在服务器请求队列中优先出队的几率就越高。

(38) ROUTING = ACCOUNT_ID(这个例子值的对应关系, 下面很快就会解释)

指定服务使用的 DDR (数据依赖路由), 这个规则必须是 *ROUTING 段中定义的。

(39) SVCTIMEOUT = 60

以秒为单位, 指定服务调用的超时时间, 不指定表示永远不会超时。

(40) AUTOTRAN = Y

表示如果当前服务被调用时不在全局事务内, 则自动开始一个全局事务。

6.2.7 *ROUTING 段的配置

在不同的组中配置相同的服务器, 这样就可以为这些服务器提供的服务配置数据依赖路由, Tuxedo 收到调用请求后, 就会根据请求缓冲区中某个字段的值, 来决定把请求转发给哪个组中的服务器处理。

```
*ROUTING
ACCOUNT_ID FIELD = ACCOUNT_ID BUFTYPE = "FML"
RANGES = "1000-4999:BANKB1, 5000-7999:BANKB2, 8000-109999:BANKB3"
```

示例 6-2

这个例子是说, 依据不同的客户 ID 号, 将从前端过来的请求, 自动路由分配到后台不同的服务组, 比如 1000-4999 的, 分配到 BANKB1 组, 5000-7999 的, 分配到 BANKB2 组, 而 8000-109999 的, 分配到 BANKB3 组。

6.3 域配置文件 DMCONFIG 及其内容

在对一个域的标准 Tuxedo 配置文件 UBB 进行分析之后, 我们再更进一步的看, 如果想让域和域之间通讯, 会涉及到的配置。

6.3.1 域(Domain)简介

随着企业信息化的水平的不断提高, 各类新兴业务的不断涌现, 一个企业内部会有很多计算机应用系统, 如 ERP, CRM, CALL CENTER, 电子商务系统, 大机遗留系统等等, 同时不同的企业之间要实现电子商务, 他们之间的系统也要能够互相调用。在一个复杂的企业应用环境下, 不同厂商的产品, 不同应用系统之间要能够有效的互连, 实现互操作, 为企业构造一个紧密协作、集中管理的, 同时又是一个高可靠、高可用、易于扩展的企业应用环境。

Tuxedo 是一个高度开放的系统, 能够很容易的与别的应用系统实现互操作。为了有效实现与其他系统的互连, Tuxedo 提出了 DOMAIN (域) 的概念, 将由很多台 (几百甚至上千) 服务器共同组成的应用系统按功能或结构划分为不同的域, 每个域独立地完成域内的操作, 域间操作由域网关完成, 从而提高每个域和整个系统的运行效率。

Tuxedo 的域特性把客户/服务器模型扩展到多个独立自治的应用系统。一个域既可以是一组 Tuxedo 的应用程序, 也可以是若干相关的应用服务和配置环境的组合, 域同时也可能是一组运行在另一个非 Tuxedo 环境中的应用程序。

Tuxedo 和其他中间件的互操作也是利用域的概念来实现的。不同的 Tuxedo 应用域中的服务程序可以互相访问对方的服务，并且当一个交易同时执行多个应用域中的服务（即对于分布式事务处理）时，能够确保交易的完整性。同时，Tuxedo 系统可以指定哪些服务是可供外部应用域访问的，并可为这些服务设置访问控制表等安全认证手段，提高整个系统的安全性。

Tuxedo 对域的划分和管理类似于网络应用中划分子网的概念—将复杂的网络应用按功能或地域等因素划分为不同的子网，子网间以路由器联接，不同的网络协议通过网关透明地转换，从而提高网络效率并加强整个网络的可管理性。这一应用模式已被广泛证明是处理大范围、复杂应用的成功经验和事实上的标准，同时也是 Tuxedo 在多年大规模企业用户的实际应用中总结和开发的行之有效的中间件功能。而其他绝大多数中间件厂商尚无力涉足域的划分与管理，从而很难为超大规模的应用提供强有力的支持。

6.3.2 Tuxedo 域划分原则

Tuxedo 应用域即是一个自治的、可独立配置管理运行的 Tuxedo 应用系统。根据应用的规模和企业组织的策略，将一个大的应用系统划分为几个独立的应用系统，不仅能够满足组织管理的需要，而且能够独立调整监控，使系统性能达到最优。

影响应用域划分的因素：

- 应用规模
- 企业组织策略
- 应用安全范围
- 共享资源和服务的用户组

Tuxedo/DOMAIN 为各应用域之间提供了互操作的基础框架。应用域机制也为应用系统提供了与其它 OLTP 应用系统的互操作手段（通过 OSI-TP 标准协议）。

6.3.3 域(Domain)的功能

可扩展性和模块化：把一个系统划分为多个 DOMAIN，可以很容易实现系统的扩展，而且每个 DOMAIN 是互相独立的，对单个 DOMAIN 的修改不会影响其他的 DOMAIN。还可以根据系统的需要，加入新的 DOMAIN。只要通过一些简单的配置，就可以实现 DOMAIN 之间的互相访问。

位置透明：CLIENT 端不用知道它调用的一个 SERVICE 位于哪里，它可能在 LOCAL DOMAIN 上，或 REMOTE DOMAIN 上，也可以位于 CICS 等其他系统上，或来自 WebLogic 上的一个 EJB。同一个 SERVICE，在 DOMAIN 内外可以有不同的名字，管理员可以定义它们之间的对应关系。

支持跨越 DOMAIN 的全局事务：Tuxedo 支持一个全局事务跨越多个 DOMAIN

安全性：对 LOCAL DOMAIN 中要被 REMOTE DOMAIN 访问的 SERVICE 可以设置访问控制表，DOMAIN 之间的数据传送支持加密和压缩，DOMAIN 之间的连接支持认证。

6.3.4 Tuxedo Domain 配置：

Tuxedo DOMAIN 的配置文件通常称为 DMCONFIG 是文本文件，在应用系统启动前要用 dmloadcf 把它编译成二进制文件 BDMCONFIG。

DMCONFIG 包括以下几个段：

配置参数	含义和用法
DM_RESOURCES	指定总体域配置版本信息。本节中的唯一参数是 Version=string, 其中 string 是一个字段, 用户可以输入当前 DMCONFIG 文件的版本号, 这个字段不由软件检查
DM_LOCAL_DOMAINS	定义该 LOCAL DOMAIN 的 DOMAIN ID(每个 DOMAIN 都有一个 DOMAIN ID), 并对该 DOMAIN 所包含的所有 DOMAIN GATEWAY GROUP 进行定义。 一个 Tuxedo 应用系统(DOMAIN)可以对应多个 DOMAIN GATEWAY GROUP, 不同的 DOMAIN GATEWAY GROUP 可对应不同的 DOMAIN GATEWAY 类型
DM_REMOTE_DOMAINS	定义可以与 DOMAIN 中的 CLIENT, SERVER 进行互操作的 REMOTE DOMAIN, 可以有多个 REMOTE DOMAIN
DM_LOCAL_SERVICES	定义 LOCAL DOMAIN 中可以被其他 REMOTE DOMAIN 访问的 SERVICE
DM_REMOTE_SERVICES	在这里配置 LOCAL DOMAIN 中可以调用的 REMOTE DOMAIN 中的 SERVICE.
DM_ROUTING	在 DOMAIN 之间可以实现数据依赖路由(DDR), 可以通过 DDR 把一个本地的请求路由到 REMOTE DOMAIN 上去。 DOMAIN 之间的 DDR 规则在 DM_ROUTING 中设置, DOMAIN GATEWAY 通过在这里指定的规则把请求发到相应的 REMOTE DOMAIN 中去
DM_ACCESS_CONTROL	为 LOCAL SERVICE 设置访问控制表, 定义哪些 REMOTE DOMAIN 可以访问这些 LOCAL SERVICE

表 6-3

如果要配置 domain 还需要对 UBB 中的配置进行修改, 即在 UBBCONFIG 中要对 DMADM, GWADM, GWTDOMAIN 三个 SERVER 进行配置。

6.4 日志文件 ULOG

User Log(ULOG) 是由 Tuxedo 系统在应用处于运行的时候产生的一个日志文件, 主要记录 Tuxedo 系统运行时候产生的一些有用的信息。应用客户端和服务端把日志记录到这个文件中, 每天产生一个新的文件, 每个机器的都有不同的日志记录。在集群中日志可以被共享给其它的远端系统使用。ULOG 使定位问题的工作变得简单, 直观。你可以用 ULOG 去识别引起系统或者是应用运行失败的原因。通常在日志中比较早产生的日志提供的信息对我们的帮助远大于较晚产生的信息。

ULOG Message Example:

```
151550.landingbj!BBL.28041.1.0: LIBTUX_CAT:262: std main starting
151550.landingbj!BBL.28041.1.0: LIBTUX_CAT:358: reached UNIX limit on semap...
```

```
151550.landingbj!BBL.28041.1.0: LIBTUX_CAT:248: fatal: system init function...
151550.landingbj!BBL.28040.1.0: CMDTUX_CAT:825: Process BBL at SITE1 failed...
151550.landingbj!BBL.28040.1.0: WARNING: No BBL available on site SITE1.
```

示例 6-3

ULOG 日志提供了比较完整的信息，从管理员的角度来说这就是相当于为解决问题指明了方向。

Beijing Landing Technologies