

# HOW TO CHECK AND LAUNCH A SERVER APPLICATION RUNNING ON LAL VACUUM STATIONS

E. JULES<sup>1</sup> - D. JEHANNO<sup>1</sup>

<sup>1</sup>Laboratoire de l'Accélérateur Linéaire, Service Electronique, Groupe Contrôle-Commande, IN2P3-CNRS et Université Paris-Sud, BP 34, 91898 ORSAY Cedex, FRANCE

REVISION : 1.2 - DATE : February 17, 2004

**ABSTRACT :** The server applications running on LAL Vacuum stations - tubes and towers - are described regarding the end-user needs to allow the complete understanding of their fonctionnalités : list of processes, descriptors, pipes, data modules, environment variables, and explanation of the cleaning and launching methods. The server application configuration methods are not described in the present paper.

## PART I : TUBE STATIONS

### 1- HOW DOES A TUBE SERVER APPLICATION START ?

After the main AC power of the tube pumping station rack has been switched on, the Central Processing Unit board (CPU12A) of the OS9 crate is configured to boot remotely from the 193.205.72.xxx ETHERNET local network. Notice that the OS9 crate does not have any hard disk drive (HDD) installed, meaning that the boot is called "diskless boot". In fact, all the needed boot files are stored in the HDD of the OS9BOOT rack located in the DAQ room of the central building and addressed at 193.205.72.160. No need to say that the OS9BOOT crate has to work properly and has to be connected to the network.

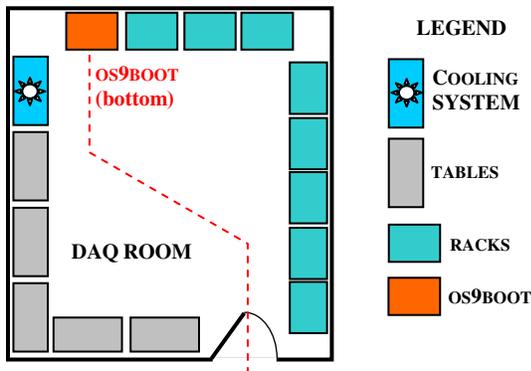


Fig. 1 : OS9BOOT crate location in central Building DAQ Room

After the 1st boot step has been completed, then the 2d boot step occurs. It consists in automatically launching the server application. The main application process is called "TuServerOS9". For now, all the needed server application files are remotely loaded from the

virgoDev/Tu/v1r1/binos9 repertory, but soon, in order to comply to the VIRGO standard, they will be loaded from the new repertory virgoApp/Vacuum/Tu/v1r1/binos9.

### 2- HOW DO WE CONNECT TO A TUBE STATION ?

**METHOD A** You can connect a serial link directly to the station, for instance with a laptop or a VT100 terminal. The dedicated COM1 connector is situated in the front panel of the OS9 crate, as follow, and the configuration is : 9600 bauds, 8 data bits, 1 stop bit, no parity.



Fig. 2 : COM1 connector at OS9 crate front panel

The possible Tube stations are, for the NORTH arm, the following :  
TuN1, TuN2, TuN3, TuN4, TuN5, TuN6

The possible Tube stations are, for the WEST arm, the following :  
TuW1, TuW2, TuW3, TuW4, TuW5, TuW6

After pressing the ENTER key, a command line menu invites you to type the user name and password, as follow :

```
User name?: super ↵
Password: user ↵
```

Then, you should see the following text, depending on the station you are connected to (here on *TuN1*) :

```
Welcome!
*****
**          LAL          VIRGO          G96 - OS9 68040
**
**  Bienvenue sur le chassis diskless OS9 68k VIRGO          **
**                      TUN1
**
**  L'accès a cet ordinateur est reserve aux personnes autorisees
**
**          Unauthorized access to this computer is prohibited
**
*****
----- DISKLESS LOGIN -----
-- Executable directory path --
/r0/CMD5
-- Source directory path --
/r0/SOURCE
February 9, 2004 Monday 13:20:00
Current total free RAM: 448.00 K-bytes
let path = "/r0/cmds"
let hostName = "TuN1.virgo.infn.it"
setenv MYHOST TuN1
setenv PATH /r0/cmds:/virgoApp/Cm/v7r13/OS9:/virgoDev/Tu/v1r1/binos9
```

**METHOD B** You prefer to use a *telnet* command. Then, you have to type the following line on your machine :

```
telnet <station_name>.virgo.infn.it ↵
```

Notice that "*virgo.infn.it*" is not mandatory, because the text name of the tube stations can only be resolved inside the Cascina domain, where *<station\_bench>* is enough to get the IP address.

Then, you will have to follow the same procedure as for method A, starting with the user name and password. Please refer to it.

### 3- CHECKING METHOD FOR THE TUBE SERVER

Generally, the 1st boot step works properly. But it is possible that the main application process *TuServerOS9* does not start or stops abnormally.

Once you are connected to the tube station, with one of the above methods A or B, you can follow a *5-step verification* to check the proper launching of the server.

This paragraph describes the 5-step verification only. The next one will give you the appropriate *solutions* to solve all the discrepancies you may encounter.

#### STEP 1 - You check that the virtual disks are correctly mounted

```
mount -d ↵
```

The mount status should be :

Device	System	Map	Point
<i>/virgoDev</i>	virgodev	yes	<i>/virgoDev</i>
<i>/virgoApp</i>	virgoapp	yes	<i>/virgoApp</i>

Notice that the */virgoDev* will soon not be mounted any more. Only */virgoApp* will remain.

#### STEP 2 - You check the environment variables declaration

```
printenv ↵
```

The needed environment variables are listed below (*ie TuN1*):

```
PATH      = /r0/cmds:/virgoApp/Cm/v7r13/OS9:/virgoDev/Tu/v1r1/binos9
CMMGR     = /virgoApp/Cm/v7r13/OS9
CMDOMAIN  = Cascina
MYHOST    = TuN1
```

#### STEP 3 - You check that the 11 processes are running

```
procs -e ↵
```

The 11 normal processes are listed below :

```
TuMidivacPumpProg  TuPumpProg      TuServerOS9
TuMaxiGaugeProg   TuTitanePumpProg TuTRGaugeProg
TuRGAProg         TuValveProg     TuTMU1600PumpProg
TuSensorProg      TuIOProg
```

#### STEP 4 - You check that the 8 descriptors are loaded in memory

```
mdir ↵ (or mdir -e ↵ for details)
```

The 8 normal descriptors are listed below :

```
inp1      cio2
sio4_1p2  sio4_1p3      sio4_1p4      sio4_1p5
sio4_2p4  sio4_2p5
```

Furthermore, you should also see the name of the following *data module* loaded in memory :

```
IOCONFMod
```

#### STEP 5 - You check that the 20 pipes are present

```
dir /pipe ↵ ou dir -e /pipe ↵ (for details)
```

The normal pipes are listed below :

```
TuIOProg_r      TuMidivacPumpProg_r  TuTRGaugeProg_r
TuMaxiGaugeProg_r  TuIOProg_w          TuPumpProg_r
TuRGAProg_r     TuSensorProg_r      TuTMU1600PumpProg_r
TuTitanePumpProg_r  TuValveProg_r

TuTRGaugeProg_w  TuMaxiGaugeProg_w  TuMidivacPumpProg_w
TuPumpProg_w     TuRGAProg_w        TuSensorProg_w
TuTMU1600PumpProg_w  TuTitanePumpProg_w  TuValveProg_w
```

### 4- MANUAL LAUNCHING OF THE SERVER

In case of any discrepancy, the tube server may not work properly. Thus, it has to be properly *re-launched* in order to restore the normal status. The solutions are listed below :

**SOLUTION 1a** - If you are connected *directly* to the tube station with a local link, you can switch off, then on, the main AC power, and redo the tests described in §2 and §3.

**SOLUTION 1b** - Or, if you do not want to switch the main AC power off for external reasons (*ie* : pumps active), you can simply *reset* the CPU12A board of the OS9 crate. The CPU12A is located at the

right of the crate, and the RESET button is the *red one* on the bottom, as follow :

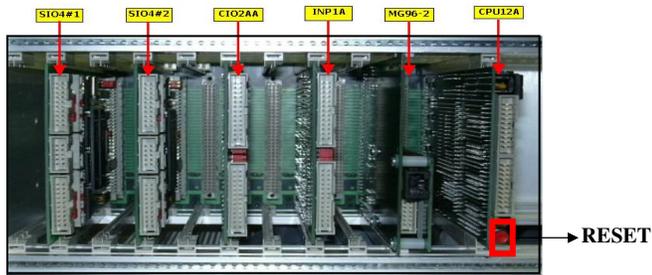


Fig.3 : RESET button position in the Tube crate

After having pressed the reset button, redo the tests described in §2 and §3.

**SOLUTION 2** - If you are connected *remotely* to the tube station with a telnet program, you can re-launch the server manually. In order to do that, you have to follow a complete *5-step cleanup procedure*, symmetrical to the 5-step verification mentionned above, and there after a *2-step launching procedure*. Notice that the following solution may also be used *locally*, for instance if you want to use other files for the server.

### 5-step Cleanup Procedure

#### STEP 1 - If necessary, mount the virtual disks manually

```
mount -m -r=1024 -w=1024 virgoDev:/virgoDev /virgoDev ↵ for virgoDev
mount -m -r=1024 -w=1024 virgoApp:/virgoApp /virgoApp ↵ for virgoApp
```

#### STEP 2 - If necessary, initialize the environment variables

```
setenv PATH /r0/cmds:/virgoApp/Cm/v7r13/OS9:/virgoDev/Tu/v1r1/binos9 ↵
setenv CMMGR /virgoApp/Cm/v7r13/OS9 ↵
setenv CMDOMAIN Cascina ↵
setenv MYHOST TuN1 ↵ (ie TuN1)
```

#### STEP 3 - In any case, kill all the processes that are running

```
kill <id1> <id2> <id3> <id4> <id5>...<idn> ↵
```

The <id> are the ID number of each process to be killed. They are given by the *procs* command.

#### STEP 4 - In any case, clean all the descriptors loaded in memory

```
unlink <desc_name1> <desc_name2> ... <desc_namen> ↵
```

The <desc\_name> are the descriptor names given by the *mdir* command. Then re-type the command, and verify the following message, indicating that it has worked :

```
"can't unlink <desc-name>"
```

#### STEP 5 - In any case, clean all the existing pipes

```
list /pipe/* ↵
```

This command actually cleans all the existing pipes.

## 2-step Launching Procedure

### STEP 1 - Verify the TuServerOS9 that will be used

```
which TuServerOS9 ↵
```

This command assures you that the proper server application will be used. It must be in accordance with the *PATH* environment variable.

### STEP 2 - Launch the Tube Server

```
TuServerOS9 <TuBenchName> <>>>/nil & ↵
```

<TuBenchName> can be one of the 12 station names, ie TuN1. Actually, it should correspond to the *MYHOST* environment variable. The nil suffix allows you to redirect the output in a dummy place. The & suffix allows you to restore the user command line.

Notice that the configuration of the server is not described in the present document, as mentionned above. Please refer to the TuUserManual file.

## PART II : HV & UHV TOWER STATIONS

### 1- HOW DOES A TOWER SERVER APPLICATION START ?

*Same description as for Tube* : after the main AC power of the tower pumping station rack has been switched on, the CPU12A board is configured to boot remotely from the 193.205.72.xxx ETHERNET local network. All the needed boot files are stored in the hard disk drive of the OS9BOOT rack located in the DAQ room of the central building and addressed at 193.205.72.160.

After the 1st boot step has been completed, then the 2d boot step occurs. It consists in automatically launching the server application. The main application process is called "ToServerOS9". For now, all the needed server application files are remotely loaded from the *virgoDev/To/v11r1/binos9* repertory, but soon, in order to comply to the VIRGO standard, they will be loaded from the new repertory *virgoApp/Vacuum/To/v11r2/binos9*.

### 2- HOW DO WE CONNECT TO A TOWER STATION ?

**METHOD A** You can connect a serial link directly to the station, for instance with a laptop or a VT100 terminal. The dedicated COM1 connector is situated in the front panel of the OS9 crate, as follow, and the configuration is : 9600 bauds, 8 data bits, 1 stop bit, no parity.

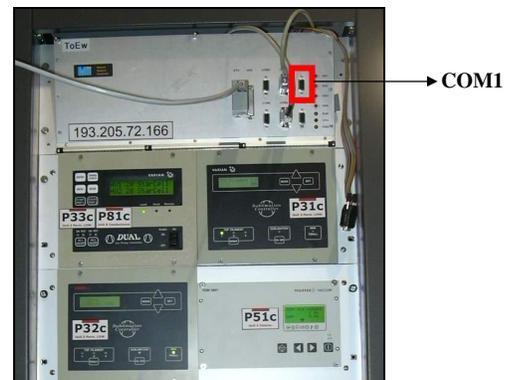


Fig.4 : COM1 connector at OS9 crate front panel (UHV here)

The possible HV Tower station names are listed below :

ToLb, ToMc, ToDb

The possible UHV Tower station names are listed below :

ToEn, ToIn, ToEw, ToIw, ToBs, ToSr, ToPr

After pressing the ENTER key, a command line menu invites you to type the user name and password, as follow :

```
User name?: super ↵
Password: user ↵
```

Then, you should see the following text, depending on the station you are connected to (here on ToEn) :

```
Process #32 logged on 04/02/10 10:34:24
Welcome!
*****
** LAL VIRGO G96 - OS9 68030
** Bienvenue sur le chassis diskless OS9 68k VIRGO
** TOEN
** L'accès a cet ordinateur est reserve aux personnes autorisees
** Unauthorized access to this computer is prohibited
*****
----- DISKLESS LOGIN -----
-- Executable directory path --
/r0/CMD5
-- Source directory path --
/r0/SOURCE
February 10, 2004 Tuesday 10:34:25
Current total free RAM: 164.00 K-bytes
let path = "/r0/cmds"
let hostName = "ToEn.virgo.infn.it"
setenv MYHOST ToEn
setenv PATH /r0/cmds:/virgoApp/Cm/v7r13/OS9:/virgoDev/To/v11r2/binos9
```

**METHOD B** You prefer to use a *telnet* command. Then, you have to type the following line on your machine :

```
telnet <station_name>.virgo.infn.it ↵
```

Notice that "virgo.infn.it" it not mandatory, because the text name of the tower stations can only be resolved inside the Cascina domain, where <station\_bench> is enough to get the IP address.

Then, you will have to follow the same procedure as for method A, starting with the user name and password. Please refer to it.

### 3- CHECKING METHOD FOR THE TOWER SERVER

Generally, the 1st boot step works properly. But it is possible that the main application process ToServerOS9 does not start or stops abnormally.

Once you are connected to the tower station, with one of the above methods A or B, you can follow a *5-step verification* to check the proper launching of the server.

This paragraph describes the 5-step verification only. The next one will give you the appropriate *solutions* to solve all the discrepancies you may encounter.

#### STEP 1 - You check that the virtual disks are correctly mounted

```
mount -d ↵
```

The mount status should be :

Device	System	Map	Point
/virgoDev	virgodev	yes	/virgoDev
/virgoApp	virgoapp	yes	/virgoApp

Notice that the */virgoDev* will soon not be mounted any more. Only */virgoApp* will remain.

#### STEP 2 - You check the environment variables declaration

```
printenv ↵
```

The needed environment variables are listed below (*ie ToEn*) :

```
PATH = /r0/cmds:/virgoApp/Cm/v7r13/OS9:/virgoDev/To/v11r2/binos9
CMMGR = /virgoApp/Cm/v7r13/OS9
CMDOMAIN = Cascina
MYHOST = ToEn
```

#### STEP 3 - You check that the processes are running

```
procs -e ↵
```

The normal processes are listed below :

HV Tower (7)	UHV Tower (12)
ToServerOS9	ToServerOS9
ToPumpProg	ToPumpProg
ToRGAProg	ToRGAProg
ToValveProg	ToValveProg
ToITRGaugeProg	ToITRGaugeProg
ToSensorProg	ToSensorProg
ToTMU1600PumpProg	ToTMU1600PumpProg
	ToTMU261PumpProg
	ToMaxiGaugeProg
	ToMaxiGaugeProg
	ToTempSensorProg
	ToIonicPumpProg
	ToTitanePumpProg

#### STEP 4 - You check that the descriptors are loaded in memory

```
mdir ↵ (or mdir -e ↵ for details)
```

The normal descriptors are listed below :

HV Tower (7)	UHV Tower (15)
inp1	inp1
inp2	inp2
cio1	cio1
cio2	cio2
sio4_1p2	sio4_1p2
sio4_1p4	sio4_1p4
sio4_1p5	sio4_1p5
	adi1
	inp3
	sio4_1p3
	sio4_2p2
	sio4_2p3
	sio4_2p4
	sio4_3p4
	sio4_3p5

#### STEP 5 - You check that the pipes are present

```
dir /pipe ↵ ou dir -e /pipe ↵ (for details)
```

The normal pipes are listed below :

HV Tower (14)	UHV Tower (22)
ToITRGaugeProg_r	ToITRGaugeProg_r
ToMaxiGaugeProg_r	ToMaxiGaugeProg_r
ToPumpProg_r	ToPumpProg_r
ToRGAProg_r	ToRGAProg_r
ToSensorProg_r	ToSensorProg_r
ToTMU1600PumpProg_r	ToTMU1600PumpProg_r
ToValveProg_r	ToValveProg_r
	ToIonicPumpProg_r
	ToTMU261PumpProg_r
	ToTempSensorProg_r
	ToTitanePumpProg_r
ToITRGaugeProg_w	ToITRGaugeProg_w
ToMaxiGaugeProg_w	ToMaxiGaugeProg_w
ToPumpProg_w	ToPumpProg_w
ToRGAProg_w	ToRGAProg_w
ToSensorProg_w	ToSensorProg_w
ToTMU1600PumpProg_w	ToTMU1600PumpProg_w
ToValveProg_w	ToValveProg_w
	ToIonicPumpProg_w
	ToTMU261PumpProg_w
	ToTempSensorProg_w
	ToTitanePumpProg_w

#### 4-MANUAL LAUNCHING OF THE SERVER

In case of any discrepancy, the tower server may not work properly. Thus, it has to be properly *re-launched* in order to restore the normal status. The solutions are listed below :

**SOLUTION 1a** - If you are connected *directly* to the tower station with a local link, you can switch off, then on, the main AC power, and redo the tests described in §2 and §3.

**SOLUTION 1b** - Or, if you do not want to switch the main AC power off for external reasons (ie : pumps active), you can simply *reset* the CPU12A board of the OS9 crate. The CPU12A is located at the right of the crate, and the RESET button is the *red one* on the bottom, as follow :

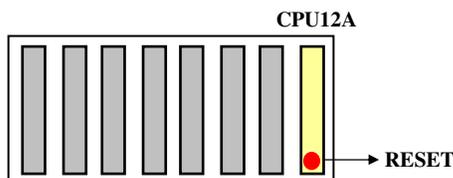


Fig. 5 : RESET button position in the Tower crate

After having pressed the reset button, redo the tests described in §2 and §3.

**SOLUTION 2** - If you are connected *remotely* to the tower station with a telnet program, you can re-launch the server manually. In order to do that, you have to follow a complete *5-step cleanup procedure*, symmetrical to the 5-step verification mentioned above, and there after a *2-step launching procedure*. Notice that the following solution may also be used *locally*, for instance if you want to use other files for the server.

#### 5-step Cleanup Procedure

##### STEP 1 - If necessary, mount the *virtual disks* manually

```
mount -m -r=1024 -w=1024 virgoDev:/virgoDev /virgoDev ↵ for virgoDev
mount -m -r=1024 -w=1024 virgoapp:/virgoApp /virgoApp ↵ for virgoApp
```

##### STEP 2 - If necessary, initialize the *environment variables*

```
setenv PATH /r0/cmds:/virgoApp/Cm/v7r13/OS9:/virgoDev/To/v11r2/binos9 ↵
setenv CMMGR /virgoApp/Cm/v7r13/OS9 ↵
setenv CMDOMAIN Cascina ↵
setenv MYHOST ToEn ↵ (ie ToEn)
```

##### STEP 3 - In any case, kill all the *processes* that are running

```
kill <id1> <id2> <id3> <id4> <id5>... <idn> ↵
```

The <id> are the ID number of each process to be killed. They are given by the *procs* command.

##### STEP 4 - In any case, clean all the *descriptors* loaded in memory

```
unlink <desc_name1> <desc_name2> ... <desc_namen> ↵
```

The <desc\_name> are the descriptor names given by the *mdir* command. Then re-type the command, and verify the following message, indicating that it has worked :

```
"can't unlink <desc-name>"
```

##### STEP 5 - In any case, clean all the existing *pipes*

```
list /pipe/* ↵
```

This command actually cleans all the existing pipes.

#### 2-step Launching Procedure

##### STEP 1 - Verify the *ToServerOS9* that will be used

```
which ToServerOS9 ↵
```

This command assures you that the proper server application will be used. It must be in accordance with the *PATH* environment variable.

##### STEP 2 - Launch the Tower Server

```
ToServerOS9 <ToBenchName> <>>/nil & ↵
```

<ToBenchName> can be one of the 10 station names, ie ToEn. Actually, it should correspond to the *MYHOST* environment variable. The nil suffix allows you to redirect the output in a dummy place. The & suffix allows you to restore the user command line.

Notice that the configuration of the server is not described in the present document, as mentioned above. Please refer to the ToUserManualV1 file.