

# ***The Multipurpose Virtual Laboratory: Node Administrator Manual***


---

**ELETTRA Software for Measurements – DRAFT**

## **Executive Summary**

The primary goal of this document is to provide a description of the *Multipurpose Virtual Laboratory* (MVL) from the point of view of the node administrator. The document will explain the basic steps needed to install and administer a MVL node. The document refers to the First Version of the MVL prototype.



<b>PAGE:</b> 1 / 13	<b>VERSION:</b> 2.0	<b>DATE:</b> 12 MAY 2005	
<b>DOCUMENT:</b> <i>THE MULTIPURPOSE VIRTUAL LABORATORY: NODE ADMINISTRATOR MANUAL</i>			

### Delivery slip

	<b>Name</b>	<b>Activity</b>	<b>Date</b>	<b>Signature</b>
<b>From</b>	Roberto Pugliese		12/05/05	
<b>Reviewed by</b>				
<b>Approved by</b>	Roberto Pugliese		12/05/05	

### Change log

<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Comment</b>
01.00.00	31/03/05	Roberto Pugliese	First version
02.00.00	12/05/05	Roberto Pugliese	Version based on scripts and webservice

# Table of Contents

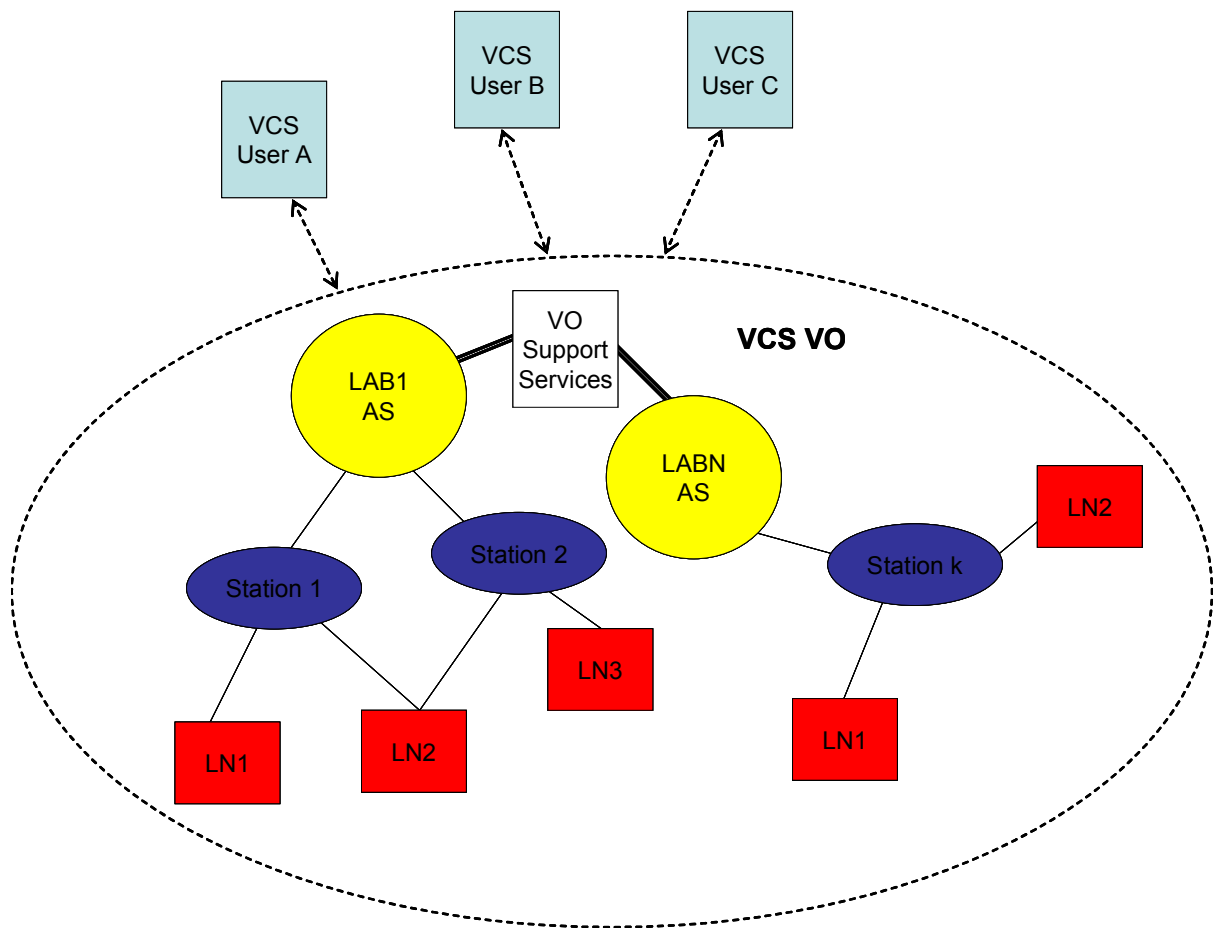
1 The Architecture of the MVL.....	3
2 Installation of a MVL Node.....	5
2.1 Installation of the software on the Application Server.....	5
2.2 Installation of the software in a MVL local node.....	8
3 Personalizing a MVL node: creating MVL stations and MVL Local Nodes.....	9
4 Connecting to the VO.....	13
5 Help desk.....	13

## 1 THE ARCHITECTURE OF THE MVL

The MVL can be used to implement a widely distributed Virtual Organisation (VO) connecting all the stations and laboratories involved in a multi-institutional collaboration. The MVL presents a multi tier architecture: one MVL node for each institution plus some centralized service to support the connection of the different MVL nodes in the VO. This document focus on the installation of a single MVL node; the centralised services are considered already working and installed in some highly available configuration. The installation of a MVL node can be divided into two main components:

- the Application Server (AS)
- a set of local nodes (LN)

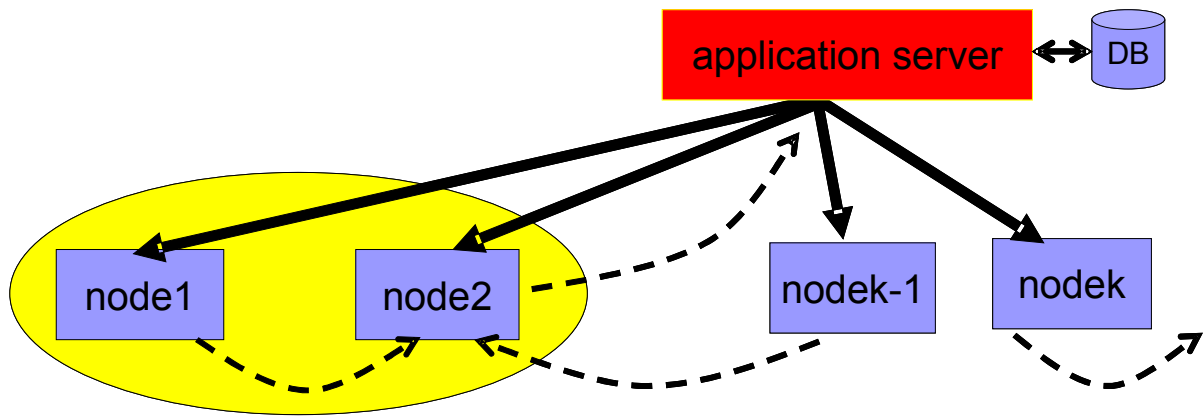
The AS is usually installed on a single host, running the portal application, the user and project database. The AS manages a (possibly) infinite number of nodes, and activates actions implemented by agents running in the local nodes.



There are many categories of nodes according to their function:

- data collection nodes,
- data storage nodes,
- computing nodes,
- data backup nodes

and new categories can be also created. A single computer can host one or more nodes. A station is associated with a group of nodes. Communications between the application server and the nodes agents is via a sort of web service implemented via secure http requests to webmin. The agents in turn communicate with the application server via database operations.



## 2 INSTALLATION OF A MVL NODE

In order to install the MVL Node you should download the MVL tarball. The MVL requires some prerequisites software components. The components you need to install including the prerequisites are:

- On the Application Server: Java, Apache, Tomcat, MySQL, MVL-AS, VNC
- On the Local Nodes: Python, MVL-LN

### 2.1 Installation of the software on the Application Server

The AS should be connected to the internet and hence secured from the system administration point of view. This machine should be installed behind a firewall or at least with a personal firewall on it. The in-bound open ports should be the standard http, https and a pool of ports dedicated to set up ssh port forwarding tunnels to support the chat server and other specific applications. Further information on specific system administration aspects can be requested via help-desk service.

The AS should be installed on a Linux machine. There is no specific requirement for the hardware configuration but a reasonable choice could be a modern Pentium compatible processor with at least 256MB better 512MB Ram, a fast Ethernet network card.

Here is a list of software packages you need to install on the selected machine in order to be able to run the MVL node.

## Java

JDK 1.4 or 1.5 is required. Please do NOT install JRE only, since Tomcat needs libraries present in JDK only.

## Apache

For manageability and efficiency purposes, it's better if Tomcat doesn't work alone. You should install Apache2.x and connect it to Tomcat with Java Connector (refer to [www.apache.org](http://www.apache.org) and [java.sun.com/j2ee/connector](http://java.sun.com/j2ee/connector) respectively). Normally Apache2.x should be already part of any modern gnu/linux distribution.

## Tomcat

Install a recent version of Tomcat in a standard directory (e.g. /usr/local/tomcat). Modify the tomcat startup scripts in order to load the Httpunit and Jsse java libraries. In the MVL tarball AS/script directory you can find the tomcat script that updated according to your installation can be copied in the /etc/init.d directory and used to start tomcat. Before running tomcat you should modify the server.xml configuration file in order to define the VO organisation parameters and connect to the MVL centralised services. In the MVL tarball AS/scripts you can find a sample server.xml file.

## JSEE 1.0.2

Download the Jsse java library from [java.sun.com/products/jsse/](http://java.sun.com/products/jsse/) and install it in a standard directory (e.g. /usr/local/jsse1.0.2).

Remember to define an environment variable used in the Tomcat startup scripts in order to match the Jsse installation (e.g. JSSE\_HOME=/usr/local/jsse1.0.2).

## MySQL

A recent version (3.0.x-5.0.x) of MySQL database engine will work fine. Create a user/password to the database as you like and configure other components

accordingly (refer to Web application configuration below). Remember to set permissions/security in your MySQL configuration in order to allow access to the database from your machine (e.g. The localhost).

Use the mvl.sql file in the MVL tarball AS/scripts directory to create the tables needed to run the MVL application. The mvl.sql script will create and populate the database. In the table actions modify the webminuser and webminpass according to your preference.

## MVL Web application

Install mvl.war in the MVL tarball AS directory in the webapp directory of Tomcat. You can manually unjar the mvl.war and then modify in mvl/WEB-INF/struts.xml the section related to database connection parameters accordingly to your MySQL installation before starting Tomcat.

## MVL chat server

Install the mvlchat.jar in the MVL tarball AS directory in a directory of your choice and update the mvlchat script in the AS/scripts directory. The script can be copied in your /etc/init.d directory and used to start the chat server.

## MVL Local Node certificates

In order to let the AS communicate securely with the Local Nodes where the resources (storage, computing, data collection, etc) reside you need to import the MVL local node certificates. If the name of the machine where you run a local node is *<node>* you have to import the certificate *<node>.pem* created by openssl and already installed in the host *<node>*. In your java installation go to the directory *jr/lib/security* and run the command:

```
keytool -import -alias <node> -file <node>.pem -keystore /usr/local/tomcat/mvl_certs/truststore
```

You have to repeat this procedure for every local node.

## MVL Status Aligner

Install the StatusAligner you can find in the AS directory of the MVL tarball. This program keeps the MVL Nodes aligned even if due to network errors some node is not reachable during critical operations.

Configure the cron daemon in order to run the StatusAligner periodically (every 5 mins).

## VNC, TightVNC and xVNC

MVL uses Virtual Network Computing technology in order to remote displays and integrate easily existing legacy code which is not easily portable to the web. Download the VNC software from the VNC website (<http://www.tightvnc.com>).

### 2.2 Installation of the software in a MVL local node

The resources that will be made available to the VO are distributed in the MVL local nodes. In each local node or resource node you need to install the MVL-LN software and configure it properly in order to use these resources through the MVL. The prerequisite software you need to install in the MVL node is Python ([www.python.org](http://www.python.org)) which is indeed optional and openssl at least 0.9.7 ([www.openssl.org](http://www.openssl.org)) which is mandatory.

Download these packages from the network and install them following the instructions of each package. These packages are needed to run the MVL-LN agent, create certificates and connect securely the local node to the AS.

## Python

Install any recent distribution of python. Normally an acceptable version of Python should be already part of any modern gnu/linux distribution.



## MVL Local Node

Install in the `/usr/local/mvl` directory the MVL tarball directory `LN/mvl`. In this folder you will find the MVL-LN software and some convenience scripts.

## MVL Local Node certificate

All the actions requested by the AS to the Local Nodes are invoked as script actions. In order to secure the communication between AS and LNs there is a mutual authentication procedure based on X.509 certificates. The AS have to know the certificates of each LNs and each LNs have to know the certificate of the AS.

In order to make simpler the installation procedure a couple of convenience script have been developed and included in the MVL tarball.

Use the script `root.sh` to generate the LN certification authority and `cert.sh <node>` to generate the LN certificate and private key. The generated certificate `<node>cert.pem` should be imported in the AS `truststore`.

The AS certificate and private key should be generated on the AS using the following keytool commands:

```
keytool -keygen -keyalg rsa -alias <as> -keystore /usr/local/tomcat/mvl_certs/keystore
```

```
keytool -export -rfc -alias <as> -keystore /usr/local/tomcat/mvl_certs/keystore -file <as>.cer
```

The file `<as>.cer` should be appended to the LN `cacerts.pem` file in order to complete the setup.

### 3 PERSONALIZING A MVL NODE: CREATING MVL STATIONS AND MVL LOCAL NODES

A MVL Node groups a set of MVL stations each implemented by a possibly overlapping set of Local Nodes where the real resources reside. In the MVL everything is configurable and hence stored in a database table. Besides this flexibility there are some basic rules. MVL users register using their email and specifying a password and the MVL associates to each user a unix login name with

the specified password. Users work by collaborating to projects on specific stations. Each project is associated with some storage and moreover with a unix group account. In order to support this each MVL Node must have at least one mechanism to authenticate users (passwd files, ypserver, ldap server, etc) and at least one storage server where the user unix account are created. Moreover each station will be associated with one storage node where the workspace of each project is created.

## The Stations table


When you create a new station you should insert a new line in the stations table. The fields of this table let you specify the station id, the station name, the associated storage node and storage path and a quota parameter.

## The Actions table

The actions table is where most of the configuration information resides. Each action has a name, the station id used to bind the action to a particular station, the handle which can be described as a URL invoked when the action is executed (i.e. the end point of the associated webservice) and a field for static parameters of the action; moreover there is a type field that can be 'sys' if the action holds system configuration information, 'tool' if the action holds the configuration of a tool associated with a particular station and 'backup' if the action defines a backup station.

The scripts associated with the actions are stored in the database and injected and hence executed in the Local Nodes via webservice. The ref\_station field, together with the action name is used to associate the action with a specific script.

Here is an example of a script in the scripts table.

PAGE: 10 / 13	VERSION: 2.0	DATE: 12 MAY 2005	
DOCUMENT: <i>THE MULTIPURPOSE VIRTUAL LABORATORY: NODE ADMINISTRATOR MANUAL</i>			

name	station s_id	scripts	vorg	md5dig	dt_update	description
addUser	3	#!/usr/bin/python import crypt import sys imp...	mvl	...	2005-04-28 10:32:43	Parameters: user, password, userid userid i...

Actions can be “generic” if they should be invoked for each station. In this case the station id is set to -1.

Here is an example of a generic action.

id	name	stations _id	handle	parameters	type	vorg	ref_sta tion
23	addUser	-1	<a href="https://phase.elettra.trieste.it:10000/">https://phase.elettra.trieste.it:10000/</a>	2	sys	mvl	3

This means that whenever a user is added to the mvl VO the system should invoke and action specified by the handle <https://phase.elettra.trieste.it:10000/> with parameters 2.

Backup nodes can be distinguished besides the specific software and scripts installed by the following actions in the action table.

id	name	statio ns_id	handle	parameters	type	vorg	ref_ stat ion
141	BackupCD-R	8	<a href="https://evsaxs.elettra.trieste.it:10000/">https://evsaxs.elettra.trieste.it:10000/</a>	CD-R	backup	mvl	3

Storage nodes can be distinguished besides the specific software and scripts installed by the 'addProject' action.

id	name	stations_ id	handle	parameters	type	vorg	ref_stat ion
32	addProject	3	<a href="https://evsaxs.elettra.trieste.it:10000/">https://evsaxs.elettra.trieste.it:10000/</a>	/net/evsaxs/store /projects	sys	mvl	3

Other actions are defined but their exhaustive description is out of the scope of this document. Please refer to the help desk service to get the specific details.

## Tools and application integration

The MVL system is also an integration platform. Tools are what is available at the station to collaborate when you create or select a project. In other terms with the tools you configure the tool-bar you find opening each project of a specific station.

All the tools are bound to a specific station. From a system administration point of view the way MVL uses to integrate tools is ssh tunnels which are used to avoid exporting to the Internet the ip addresses of local hosts. An example of tool which uses this technique is the chat.

id	name	stations_id	handle	parameters	type	vorg	ref_station
23	Chat	2	/chat.do	0	tool	mvl	

Available software resource and tools are integrated using ad-hoc tool actions. Tools with a web interface can be simply integrated using links. The Apache proxy module is used for safety reasons in order to avoid exposing local node addresses. Applications without a web interface are integrated using remote display techniques. MVL uses in particular tightVNC which is installed on the application server (in this case xVNC should be configured) or on the local node (this is mandatory if the node is a Windows machine); in any case ssh forwarding tunnels are configured on the AS in order to avoid exposing local node addresses.

Data collection and computing nodes can be distinguished besides the specific software and scripts installed by “Application” actions:

id	name	station_s_id	handle	parameters	type	vorg	ref_station
56	Applications	2	/vnc/index.jsp	HOST=<hn>&PORT=<pt>&WIDTH=1024&HEIGHT=768	tool	mvl	

Data collection nodes in particular have specific software agents which save the

data collected in the storage servers. Data is accessible via the FileManager tool.

Other specific configuration variables can be defined and modified. Please refer to the help desk service to get the specific details.


## 4 CONNECTING TO THE VO

In order to have you node registered and knew to the other nodes belonging to the VO use the help desk TAR service. In the problem description specify that you request a registration of your node in the VO and provide the URL of your MVL application. You will receive a reply when your MVL Node is effectively added to the VO and all the other nodes have been notified.

## 5 HELP DESK

The MVL is a complex distributed system and even if we will try to make the installation and management of a MVL node as simple as possible it's impossible to take into account and sort out in this document all the problem which may arise in the operations of a MVL node.

A practical approach consist of providing a good quality help desk. This service is twofold: we have set up a discussion wiki which includes a faq (<https://ulisse.elettra.trieste.it/wiki/index.php/MVL>) and a Technical Assistance Request (TAR) service. The TAR service is accessible from the portal <http://ulisse.elettra.trieste.it/area> by selecting the TAR link on the left. In order to have you question submitted to the development staff you have to fill a form and select MVL as the target system.

PAGE: 13 / 13	VERSION: 2.0	DATE: 12 MAY 2005	
DOCUMENT: <i>THE MULTIPURPOSE VIRTUAL LABORATORY: NODE ADMINISTRATOR MANUAL</i>			