www.project-open.com

]project-open[V3.1 Advanced Unix Installation Guide

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

Frank Bergmann,
V2.1, 2006-06-10

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

INDEX

1 ABOUT THIS DOCUMENT 4
1.1 VERSION 4
1.2 SCOPE 4
1.3 AUDIENCE 4
1.4 TODO'S 4

2 SYSTEM OVERVIEW 5
2.1 CONTEXT OVERVIEW 5
2.2 SIZING AND PREREQUISITES 6
2.3 DEVELOPMENT AGAINST PRODUCTION SERVER 7
2.4 CONFIGURATION VARIANTS 7

3 LINUX BINARY INSTALLER 9

4 POSTGRESQL INSTALLATION 10

5 ORACLE INSTALLATION 11

6 AOLSERVER INSTALLATION 13

7 RESPAWN CONFIGURATION 14
7.1 DAEMONTOOLS INSTALLATION 14
7.2 INITTAB INSTALLATION 14

8 POUND 16
8.1 POUND INSTALLATION 16
8.2 POUND CONFIGURATION 18

9 VIRTUAL SERVER BASIC INSTALLATION 19
9.1 SETUP A DNS ENTRY 19
9.2 CREATE A NEW ORACLE DATABASE 20
9.3 CREATE A NEW UNIX USER, GROUP AND DIRECTORIES 20
9.4 CREATE A NEW SERVER DIRECTORY AND EXTRACT THE OPENACS CODE 20

9.5 Edit the /web/<server>/etc/config.tcl 20
9.6 Create a Log-Directory 21
9.7 Add a /sbin/services/<server> entry 21
9.8 Manually Start the Server 22
9.9 Basic Online Server Configuration 22
9.10 Configure Daemontools 23

10 BACKUP 24
10.1 Concept, Tradeoffs and Decisions 24
10.2 Full Backup 25
10.3 Application Backup 25
10.4 Database Backup 25

11 FAILURE RECOVERY 29
11.1 Concept, Tradeoffs and Decisions 29
11.2 Failure Types 30
11.3 Linux Operating System Recovery 32
11.4 Installation From Scratch 32
11.5 Oracle and ACS Update Procedures 32
11.6 Informing the Users 33

12 SECURITY 35
12.1 Concept, Tradeoffs and Decisions 35
12.2 Implementation 35
12.3 Configuration Details 36

13 MAINTENANCE 39
13.1 Directories to watch for it's size 39
13.2 Big Brother 39
13.3 Adding a new Virtual Server (ToDo: update) 39

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

1 About this Document

1.1 Version

Version: 2.1, 2006-06-10

Author: Frank Bergmann

Status: Advanced Draft

1.2 Scope

This manual describes the installation of professional ASP hosting and development services of **]project-open[** V3.x on a Linux server with SuSE, Debian or Red Hat for professional hosting or development.

Please note:

- *This manual does not deal with day-to-day operations. Please see the "PO-Operations-Maintenance-Guide" for updates, maintenance, backup and recovery.*
- *This manual does not deal with a standard Linux installation. Please see the "INSTALL" file for details that comes as part of the* ProjectOpen-3.1.2.0.LinuxInstaller.tgz *installer on SourceForge.*

1.3 Audience

The manual is written for Unix system administrators.

1.4 ToDo's

- Explain the Apache setup, including the “excuse” screen and BigBrother.
- Protect BigBrother directory with password.
- Check for tmp symlink attack at the Oracle backup script.
- Update the firewall policies and rules to reflect recent (010510) changes.
- AOLServer SSL Certificate configuration.

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

2 System Overview

2.1 Context Overview

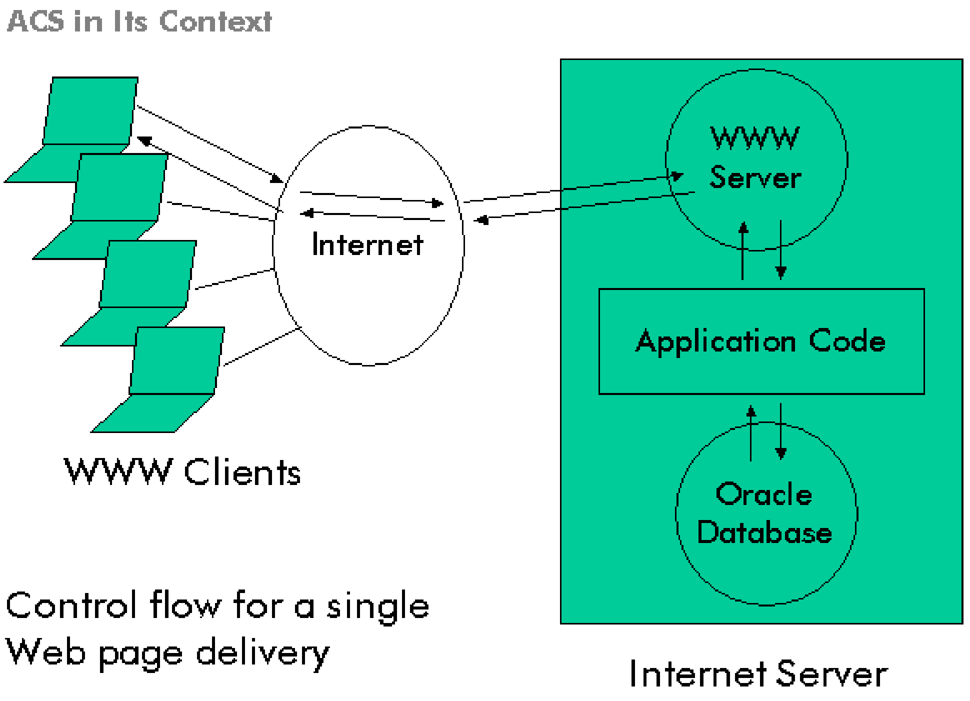


Figure 2-1

An OpenACS server in general consists of the following elements:

- a web server (AOL Server),
- the OpenACS application code and
- the database (PostgreSQL 7.4.x or 8.0.1 or Oracle 8.x – 10.x)

A Unix/Linux server hosts these processes.

2.2 Sizing and Prerequisites

2.2.1 Hardware Sizing

The following table gives an estimate of hardware requirements for running a single operative OpenACS server:

Concurrent Users	Computer	Memory	Hard Disk
1-20	1 Processor Intel Server 1GHz	1GByte	EIDE RAID 1
20-200	4 Processor Intel Server 2GHz	4GByte	SCSI RAID
200-2000	16 Processor Solaris Server	16Gbyte	RAID Arrays
2000-…	To be determined		

Table 2-1

Please note the term "concurrent users". This refers to users who are trying to access the server continuously and concurrently during peak times (for example in a training class). For every "concurrent user" you can have 5-10 "active users" who are using the system less intensively. Examples include a project manager who is writing emails while using the system.

2.2.2 DB

]project-open[is usually run with PostgreSQL 7.4.x or 8.0.11. Other versions are not supported at the time of writing and actually don't work.

]project-open[also supports Oracle 8.x-10.x on request. Please contact us for more information.

2.2.3 AOLServer

AOLServer 3.3 is a special Web server developed by AOL. The version used for OpenACS has a TCL interpreter and an Oracle driver compiled into the web server.

2.2.4 OpenACS

The OpenACS system itself consists only of “passive” TCL code that has to be extracted into a file system directory and SQL code that has to be loaded into the database.

2.2.5 Operating System

]project-open[environment runs best under Linux 2.6.x environments.

]project-open[is also available for Windows (2000, 2003). However, Windows does not reach the same uptime and stability as Linux.

2.3 Development Against Production Server

A "Development Server" is used for software developers in order to develop new software for **]project-open[**. A "Production Server" is meant to run a company's operations.

The differences between a “productive” and a “development” server are small. They basically consist of added tools and development servers (additional AOLServer instances listening on ports other than 80).

Item	Development	Productive
AOLServer Version	aolserver3.3+ad13	
AOLServer Config	/web/server/etc/config.*.tcl	
AOLServer Restart	Inittab or Daemontools	
Sendmail	Postfix or Sendmail	
Backup	Backup using regular CVS “commits”	Mirroring/redundant setup
Security	▪ inetd enabled for Intranet ▪ access at 3000x port ▪ SSH accessible for Intranet or even Internet	▪ inetd disabled ▪ access to port 80 from the Internet ▪ SSH accessible for Intranet and support only

Table 2-2

2.4 Configuration Variants

This manual deals with the installation of the ACS environment on various Linux distributions and both for development and production environments.

2.4.1 Linux Distribution Differences

We have successfully installed and tested the ACS Installation (Oracle8i and AOLServer) on:

- SuSE
- RedHat
- Debian 3.1 Sarge

The choice of distribution has no effect on the Database and AOLServer code. However, there are some minor differences in system administration such as:

- SuSE enabled/disabled the startup scripts in /etc/rc.config while RedHat uses a separate command for it.
- The locations of the Apache configuration file in SuSE is /etc/httpd/httpd.conf while RedHat uses /etc/httpd/conf/httpd.conf.
- This cannot be an exhaustive list. Please refer to the SuSE and RedHat documentation for details.

3 Linux Binary Installer

The easiest way to install the **]project-open[** is using the binary installer. To download the installer please go to the download zone in SourceForge http://www.sourceforge.net/projects/project-open/ and download the "ProjectOpen-3.1.2.0.LinuxInstaller.tgz".

There is a "INSTALL" file included in the **]project-open[** binary installer. Please see the file and follow the instructions.

The rest of this document deals with specific parts of the installation in more detail. However, you don't need to continue reading this manual if you have successfully installed the ***]project-open[*** *from the binary installer.*

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

4 PostgreSQL Installation

PostgreSQL is part of all major Linux distributions:

- SuSE
- RedHat
- Debian Sarge

In order to install PostgreSQL please use the respective installation managers (YaST, RPM or APT-GET).

The following versions of PostgreSQL are supported:

- 7.4.x (x >= 3) on Windows
- 7.4.x (x >= 8) on Linux

PostgreSQL 7.4.x includes a known bug with the TSearch2 full text search engine extension in the backup recovery. *Please do not install TSearch2 on PG 7.4.x.*

In addition **]project-open[** runs on:

- 8.0.1 on Linux and Windows

However, there are some small issues during the installation process, so we only recommend this version for **]project-open[** users with support contract.

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

5 Oracle Installation

*This chapter is only relevant for servers based on Oracle. Most installations of **]project-open[** today run on PostgreSQL, so you can ignore this chapter.*

This procedure defines how to reinstall the Oracle database in case of a complete loss of the /opt/oracle directory, for example after a complete reinstallation.

Please note that all Oracle8i files are kept together in the /opt/oracle directory in our current configuration. In particular this means that the tablespaces are also held in the file system, as opposed to a raw devices as usual.

For this reason it's sufficient to restore the /opt/oracle directory to recover the entire Oracle database.

To install Oracle from scratch perform the following steps:

1. Make sure that /etc/rc.d/oracle8i and /etc/rc.d/listener8i are stopped if the database was running before.
2. Rename an existing /opt/oracle directory /opt/oracle.yymmdd.
3. Untar the "oracle.initial.tgz" file (or similar) into the root ("/") directory to create a /opt/oracle directory.
4. Untar the "ora8.misc.tgz" file into root ("/"), thus creating the files:
   ```
   /etc/profile.oracle      # env vars to be included into /etc/profile
   /etc/oratab              # configuration of database SIDs
   /etc/rc.d/oracle8i       # start/stop the database
   /etc/rc.d/listener8i     # start/stop the TCP interface
   ```
5. Copy the content of /etc/profile.oracle into /etc/profile.
6. Make sure the start/stop scripts "oracle8i" and "listener8i" exist in /etc/rc.d/.
7. Create symbolic links in the /etc/rc.d/ directory:
   ./rc2.d/K10oracle8i -> ./oracle8i
   ./rc2.d/S40oracle8i -> ./oracle8i
   ./rc3.d/K10oracle8i -> ./oracle8i
   ./rc3.d/S40oracle8i -> ./oracle8i
   Do **not** create symbolic links for the listener8i file. The listener8i is not necessary for the operation of the OpenACS tool and presents an important security hole.
8. On SuSE edit the /etc/rc.config file and add a "START_ORACLE=yes" line. On RedHat edit the /etc/rc.d/oracle8i script and remove the line that checks for the presence of this variable.
9. Make sure the user "oracle" (59) and the group "oinstall" (54) exist in /etc/passwd and /etc/group and change the ownership of all files in /ora8 to these:
   ```
   chown -R oracle:oinstall /ora8
   ```

10. Logout and login again as root to reload the changes in /etc/profile. Check that the environment variables are OK:
    ```
    # set | grep -i ora
    CLASSPATH=:/ora8/m01/app/oracle/product/8.1.6/jlib
    LD_LIBRARY_PATH=/usr/local/lib:/ora8/m01/app/oracle/product/8.1.6/l
    ib
    ORACLE_BASE=/ora8
    ORACLE_HOME=/ora8/m01/app/oracle/product/8.1.6
    ORACLE_SID=ora8
    PATH=/sbin:/usr/sbin:/usr/local/sbin/:/root/bin:/usr/local/bin:/usr
    /bin:/bin:/ora8/m01/app/oracle/product/8.1.6/bin
    ```
11. Check that SqlPlus is running:
    ```
    # sqlplus intranet/intranet
    SQL*Plus: Release 8.1.6.0.0 - Production on Wed May 9 09:24:53 2001
    ..
    SQL> select email from users;
    EMAIL
    ------------------------------------------------------------
    system
    <some more emails>
    ```
12. Make sure the database backup is running. Please see the backup chapter for the Oracle backup script and the corresponding crontab entries.
    AOLServer and OpenACS Installation

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

6 AOLServer Installation

1. Check if a /home/aolserverXXX directory already exists and rename it.
2. Untar the aolserver.yymmdd.tgz into the directory /home. You should get a new directory /home/aolserver3.1+ad8+nsvhr.
3. Make sure the user “nsadmin” (id=501, homedir=/home/aolserver, group=nsadmin) and the group “nsadmin” (id=102) exist in /etc/passwd and /etc/group. Create a symbolic link /home/aolserver to the /home/ aolserver3.1+ad8+nsvhr directory. Change the ownership of all files in /home/ aolserver3.1+ad8+nsvhr to user and group “nsadmin”:
   ```
   cd /home
   ln -s aolserver3.1+ad8+nsvhr aolserver
   chown -R nsadmin:nsadmin aolserver3.1+ad8+nsvhr
   ```
4. Create a new group “web” and make the user “nsadmin” member of “web”.
5. Untar the web.yymmdd.tgz into /. You should get a new directory /web.
6. Only for development systems: If the /web directory contains subdirectories owned by various users, import those users from the /etc/passwd file of original host into the current /etc/passwd. Make sure there are no duplicate user ids. Change the password for each of the new users to a reasonable default. Check that the ownership of the subdirectories in /web coincide with the names of the subdirectories. Make sure all users have /web/<username> as home directory.

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

7 Respawn Configuration

The “respawn configuration” serves the purpose of restarting AOLServer in the case the process should terminate, either during a manual restart or due to an error.

There are two options for a respawn:

- Daemontools is a specific tools for controlling the status of a server. It give you more freedom and allows non-privileged users to manually restart their servers. This is a good option for both development and production servers.
- You can add AOLServer to /etc/inittab. This is more simple then Daemontools, but is more suitable for production servers.

7.1 Daemontools Installation

Install the “daemontools” from http://cr.yp.to/daemontools.html. We have made negative experience with version 0.76, and prefer version 0.70. You can basicly follow the installation instructions on this page:

Untar the daemontools_7.0.tgz into /usr/src, enter into the directory and execute “make” and “./install”. You need to have the standard Linux development tools such as gcc and make installed.

Make sure the daemontools are started up by the operating system after the database has started by editing the Sysetem V init scripts using you favourite runlevel editor.

Make sure that one “supervise” process starts up for each subdirectory of /sbin/service/ by using:

```
# ps auxw | grep supervise
```

For the configuration of a specific virtual server please see the next chapters.

7.2 Inittab Installation

Adding the following line anywhere to your /etc/inittab file will make AOLServer start during system boot and will make it restart after a shutdown:

```
nsd1:34:respawn:/usr/local/aolserver/bin/nsd-oracle -i -u projop -g projop -t
/web/projop/etc/projop.tcl
```

Or for PostgreSQL:

```
nsd1:34:respawn:/usr/local/aolserver/bin/nsd-postgres -i -u projop -g projop -t
/web/projop/etc/projop.tcl
```

Please test this new configurations manually before putting it in production use: Copy the line from /etc/inittab into your command line and replace the “-i” option (Inittab-mode) by the “-f” option (Foreground mode). You will get a line like this:

```
/usr/local/aolserver/bin/nsd-oracle -f -u projop -g projop -t
/web/projop/etc/projop.tcl
```

Please note that this entry calls `nsd-oracle` or `nsd-postgres` instead of plain `nsd` in order to load the Oracle or PostgreSQL environment variables. Here are the respective files:

```
#!/bin/sh
source /etc/profile.d/oracle.sh
exec /apps/aol408/bin/nsd $*
```

Or for PostgreSQL:

```
#!/bin/sh
source /etc/profile.d/postgres.sh
exec /apps/aol408/bin/nsd $*
```

In order to force init to reload the /etc/inittab and to activate the new respawn server please use:

```
# killall -HUP init
```

A sample /etc/profile.d/oracle.sh file could look like this:

```
#!/bin/sh
export NS_DAEMON="nsd"
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/8.1.7
export ORACLE_SID=ora8
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_PATH=$ORACLE_HOME/bin
export ORAENV_ASK=NO
export TNS_ADMIN=$ORACLE_HOME/network/admin
export NLS_LANG=.UTF8
export NLS_DATE_FORMAT="YYYY-MM-DD"
unset LANG
```

8 Pound

8.1 Pound Installation

Install “Pound” from http://www.apsis.ch/pound/. Just take the latest version and follow the installation instructions.

Pound requires the OpenSSL libraries from http://www.openssl.org/source/. OpenSSL is know to have difficulties compiling under a Linux 2.4.x kernel.

You need to automate the startup of Pound during the computer bootup. Here is a RedHat pound startup file in /etc/init.d/pound:

```
#!/bin/sh
#
# chkconfig: - 91 35
# description: Starts and stops the pound daemon \
#
# config:  /etc/pound/pound.cfg


# Source function library.
if [ -f /etc/init.d/functions ] ; then
  . /etc/init.d/functions
elif [ -f /etc/rc.d/init.d/functions ] ; then
  . /etc/rc.d/init.d/functions
else
  exit 0
fi

# Check that pound.cfg exists.
[ -f /etc/pound/pound.cfg ] || exit 0

POUND_BIN=/usr/local/sbin/pound
[ -f $POUND_BIN ] || exit 5

RETVAL=0
start() {
        KIND="POUND"
        echo -n $"Starting $KIND services: "
        daemon $POUND_BIN -f /etc/pound/pound.cfg
        RETVAL=$?
        echo
        return $RETVAL
}
stop() {
        KIND="POUND"
        echo -n $"Shutting down $KIND services: "
        killproc pound
        RETVAL=$?
        echo
        return $RETVAL
}
restart() {
        stop
        start
}
reload() {
        echo -n $"Reloading pound.conf file: "
        killproc pound -HUP
        RETVAL=$?
        echo
        return $RETVAL
}
rhstatus() {
        status pound
}
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  reload)
        reload
        ;;
  status)
        rhstatus
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart|reload|status}"
        exit 1
esac
exit $?
```

And here the equivalent for SuSE > 8.2:

```
#!/bin/sh
# Copyright (c) 1996, 2000 S.u.S.E. GmbH Fuerth, Germany.  All rights reserved.
#
#         Frank Bergmann <frank.bergmann@project-open.com>
#
# /etc/init.d/pound
#
### BEGIN INIT INFO
# Provides:       pound
# Required-Start: $network
# Required-Stop:
# Default-Start:  3 5
# Default-Stop:
# Description:    Pound Reverse Proxy
### END INIT INFO

. /etc/rc.status

POUND_BIN=/usr/local/sbin/pound
test -x $POUND_BIN || exit 5

# First reset status of this service
rc_reset

if [ ! -f /etc/pound/pound.cfg ] ; then
    echo "can't find /etc/pound/pound.cfg"
    # program is not configured
    exit 6
fi

case "$1" in
    start)
        echo -n "Starting Pound Reverse Proxy"
        if checkproc $POUND_BIN; then
                echo -n "Pound Reverse Proxy"
                rc_status -v
                exit
        fi
        startproc -l /var/log/fs-errors $POUND_BIN -f /etc/pound/pound.cfg
        rc_status -v
        echo
        ;;
    stop)
```

```
        echo -n "Shutting down Pound Reverse Proxy"
        killproc -TERM $POUND_BIN > /dev/null
        rc_status -v
        echo
        ;;
    restart)
        $0 stop
        $0 start
        rc_status
        ;;
    status)
        echo -n "Checking for Pound Reverse Proxy: "
        checkproc $POUND_BIN
        rc_status -v
        ;;
    *)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
        ;;
esac

rc_exit
```

8.2 Pound Configuration

Below an excerpt from the Pound configuration files /etc/pound/pound.cfg:

```
# *****************************************************
#       /etc/pound/pound.cfg
#       030928 Frank Bergmann <fraber@project-open.com>
# *****************************************************

LogLevel 3
ListenHTTP 0.0.0.0,80
User wwwrun
Group www
# RootJail /var/pound

# Values for a busy network:
# Overload is handled at the ACS server, not Pound, so
# wait _really_ long.

#Server 5
#Client 20
#Err500
#Err501
Err503 "/etc/pound/503.html"

UrlGroup ".*"
HeadRequire Host ".*podemo.*"
BackEnd 127.0.0.1,30001,1
EndGroup
```

9 Virtual Server Basic Installation

This chapter describes how to setup an OpenACS “virtual” server.

A virtual server is a specific instance of Project/Open, normally associated with a specific company. It is called a “virtual” server because one physical computer with a single IP address can host several virtual servers.

Each virtual server requires some 50-150Mbyte of RAM, depending on the number of users, so please make sure your computer is sufficiently equipped.

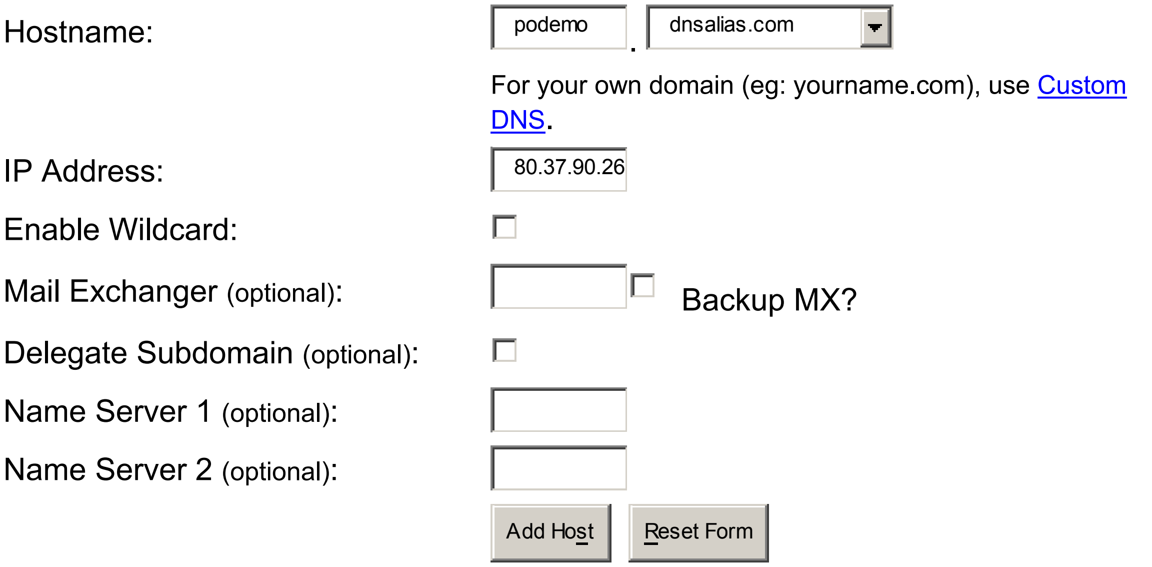
Virtual server hosting works using “Pound” (see installation chapter). Pound is a “reverse proxy” server listening on port 80. Pound looks at the HTTP request header of incoming connections and routes the request to the corresponding virtual server.

The virtual servers are running on user ports > 1024, by default named 30000, 30001, etc…

9.1 Setup a DNS Entry

The first step is to setup an **external** DNS entry for the new virtual server, for example “podemo.dnsalias.com”. Project/Open uses DynDNS.org to host most of its addresses, so this is our default. However, you can use any other domain or provider to obtain an external name for your virtual server.

Below please find a screenshot from a DynDNS.org formular.



Hostname: podemo . dnsalias.com ▾

For your own domain (eg: yourname.com), use Custom DNS.

IP Address: 80.37.90.26

Enable Wildcard: □

Mail Exchanger (optional): □ Backup MX?

Delegate Subdomain (optional): □

Name Server 1 (optional):

Name Server 2 (optional):

9.2 Create a new Oracle database

The new virtual server needs a separate Oracle user and tablespace for security and administrative reasons. The following commands create a suitable environment for a sample server called “podemo”:

```
# sqlplus system/manager
SQL > create tablespace podemo datafile '/opt/oracle/oradata/ora8/podemo01.dbf' size 5m
autoextend on default storage ( pctincrease 1);
SQL > create user podemo identified by podemo default tablespace podemo temporary tablespace
temp quota unlimited on podemo;
SQL > grant connect, resource, ctxapp, javasyspriv, query rewrite to podemo;
SQL > revoke unlimited tablespace from podemo;
SQL > alter user podemo quota unlimited on podemo;
SQL > quit
#
```

9.3 Create a new Unix user, group and directories

Every virtual server needs to be run with it’s own user and group for security reasons. The following commands create a new user under Linux:

```
# groupadd podemo
# useradd -g podemo -s /bin/false -d /web/podemo/filestorage podemo

# mkdir /web/podemo
# chwon podemo:podemo /web/podemo
# chmod go-rwx /web/podemo
```

9.4 Create a new server directory and extract the OpenACS code

The main location of the virtual server is /web/<server>/:

```
# cd /web
# tar xzf /tmp/openacs-5.1.0.tar.tgz
# mv openacs-5.1.0 podemo
# mkdir /web/podemo/filestorage

# chwon -R podemo:podemo /web/podemo
# chmod -R go-rwx /web/podemo
```

9.5 Edit the /web/<server>/etc/config.tcl

Edit the /web/<server>/etc/config.tcl file. Below you find some important variables. Please use double quotes where specified below, except if you know better.

```
set httpport <httpport> (same as in pound.cfg)
set httpsport <httpsport> (you can leave the default=443).
set hostname “<server>.dnsalias.com” (double quotes required)
set server “<server>”
set servername $server
set serverroot “/web/$server”
set database oracle
set db_password $server
set homedir “/usr/local/aolserver” (the right AOLServer path)
…
set pageroot /web/${server}/www
ns_param serverlog “/var/log/${server}/${server}-error.log
…
ns_param file /var/log/${server}/${server}.log
…
```

Make sure you have not changed the ownership or permissions of the config.tcl file, for example when you edit the file as root…

9.6 Create a Log-Directory

- Create a new log directory /var/log/<server> and change user and permissions to rw for <server>

```
# mkdir /var/log/podemo
# chown podemo:podemo /var/log/podemo
# chmod go-rwx /var/log/podemo
```

9.7 Add a /sbin/services/<server> entry

This entry is later going to be the entry point for the Daemontools process monitoring. However, you can also start the server manually by calling the “run” file.

- Create the subdirectories

```
# mkdir /sbin/services/podemo
# cd /sbin/services/podemo
# mkdir /sbin/services/podemo/supervise
# emacs –nw run
```

Please edit the “run” file so that it contains the following code (you need to replace the server name “podemo” by your server name):

```
#!/bin/sh

source /etc/profile.d/oracle.sh

exec /usr/local/aolserver3.3oacs/bin/nsd -it /web/podemo/etc/config.oracle.tcl -u podemo -g
podemo
```

9.8 Manually Start the Server

```
# /sbin/services/<server>run &
```

Don’t forget the “&” after the line. The AOLServer won’t allow you to interrupt its execution (Ctrl-C) or to suspend execution (Ctrl-Z).

- You should see the following lines

  ```
  [01/Jun/2004:14:16:12][12087.16384][-main-] Notice: nsd.tcl:
  starting to read config file...
  [01/Jun/2004:14:16:12][12087.16384][-main-] Notice: nsd.tcl:
  finished reading config file.
  ```

- Check for lines containing “Error” in the log file:

  ```
  # cd /var/log/<server>
  # less <server>-errors.log
  ```

- The last three lines of the file should look like:

  ```
  Notice: nssock: listening on 0.0.0.0:30021
  Notice: nssock: starting
  Notice: nssock: accepting connections
  ```

- Check that the server is listening on the right IP address by checking for a line containing the server port (300xx) in the output of:

  ```
  # netstat -nlp
  ```

- Connect to the server via a browser using a URL with an explicit port. You should see the initial configuration page (see next chapter)

  ```
  # http://<hostname>:<httpport>/
  ```

- Now you should be able to connect to your new virtual server via a web browser and continue the installation graphically.

9.9 Basic Online Server Configuration

In order to proceed with this chapter we assume that OpenACS Installation form is available in a web browser. The page is called “OpenACS Installation: Welcome” and should some 14 formula entries, starting with “Email”, “Username” etc. Below are some sample values that we used for our own server:

Email:	“frank.bergmann@project-open.com”
Username:	„fraber“ (some short form of the users name)
First Name:	„Frank“
Last Name:	„Bergmann“
Password:	„secret“
Password (again):	„secret“
System URL:	„http://projop.dnsalias.com/“
System Name:	„Project/Open Demo Server“
Publisher Name:	“Project/Open” (your organization, for legal purposes)
System Owner:	… (the system should have filled the field with the “Email”.

Press “Start Installation”. The process will take some 2-10 minutes, depending on your hardware. Don’t stop the process. You can follow the installation process by “tailing” the log file:

```
# tail -f /var/log/<server>/<server>-errors.log
```

Check the output for occurrences of “ERROR” (search with case sensitive search). There should be no error lines. If you do find them, please check in the OpenACS community (http://www.openacs.org/) or with Project/Open if you have a support contract.

In total, there should be some 2000 lines in the output. The last lines of the file should be like: “**Installation finished** The server has been shut down…”.

9.10 Configure Daemontools

Now that the Project/Open base module is running, you need to automate the startup of the server with Daemontools. There are alternatives to Daemontools, namely entering a reference to /sbin/service/<server>/run in the /etc/inittab, but Daemontools are considerably more flexible.

- We assume that you have already installed the Daemontools binaries.
- We assume that you have already created the /sbin/service/<server>/ directory with the file “run”.
- Shut down the virtual server using:

```
# svc -d /sbin/service/podemo/
```

- Restart the virtual server

```
# svc -u /sbin/service/podemo/
```

Watch for the /var/log/<server>/<server>-error.log file. The server should come up without problems.

To test Daemontools monitoring shutdown the virtual server “the hard way” and watch the error.log file if it comes up again:

```
# killall -9 nsd
```

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

10 Backup

10.1 Concept, Tradeoffs and Decisions

The backup and recovery concept is structured in three stages to construct the working application:

1. Operating System (Hardware, Solaris, SSH, …)
2. Applications (Oracle and AOLServer) and
3. Data and code

Both, the backup and recovery processes are oriented around these three basic steps and the alternative pathes that lead to a successful recovery.

	Data	Backup	Restore
Operating System	2nd Disk backup	Copy the entire disk	Swap disks
	Tape backup	Copy the OS to tape	Restore tape from running minimal installation
	Solaris install from scratch	Original installation media	Install
Applications	Oracle Tar	Tar the /ora directory to tape or disk	Restore the Tar
	Oracle install from scratch	Original installation media	Install
	AOLServer Tar	Tar the /usr/local/aol* to tape or disk	Restore the Tar
	AOLServer from scratch	Original installation media	Install
Data/ Code	Oracle Dump	Write Oracle dump	Restore Oracle dump
	Oracle Full Backup	Tar the /ora directory to tape or disk	Restore the Tar
	Application Code	Tar the /web directory to tape or disk	Restore the Tar

The main challenge in this environment is to make sure that a specific version of the application code is working together with a specific data model.

10.1.1 Tradeoffs and Decisions

Database backup is a standard process covered in detail by the Oracle Administrators Manual. Various options exist.

- Oracle Hot Backup:
  The productive server is backed up using a “dump” database export that saves the content of the entire database into a flat file. This file is moved to a backup location.

This process has the effect of slowing down the database operations, but does not affect the availability, so that several daily backups are possible, preferably during low traffic hours (morning, lunch, night).

- Full Backup:
  In addition, the productive server is backed up once a week fully.

The reason to choose this procedure is that the size of the productive database is relatively small (compressed less then 100mByte) and that we want to backup the DB several times a day. So an online backup is necessary, and the “dump” procedure is much easier to handle than a backup based on redo logs.

To avoid security holes, all database dumps are encrypted using PGP at the productive server and are stored in encrypted form at the backup server.

10.2 Full Backup

A full backup needs to be made every week with the predominant spare disk configuration, copying the entire hard disk to the spare disk. This procedure includes the shutdown of the Oracle database because it includes a normal shutdown of the computer.

An excerpt from the root crontab:

```
37 1 * * 7 /root/.backup/pre_backup.pl
```

10.3 Application Backup

Both Oracle and AOLServer are slowly changing data, being updated every few months. Both application are updated every few months from installation media.

10.3.1 Oracle 8i

Tar the content of the /opt/oracle directory to a tape or a CD. In addition, the following files have to be backed up:

```
/etc/oratab
/etc/oraInst.loc
/etc/profile/oracle/*
```

10.3.2 AOLServer

The AOLServer installation has similar characteristics as Oracle. The following files have to be backed up for a full recovery:

```
/usr/local/aol*
/sbin/services/*
```

10.4 Database Backup

The backup implementation is based on two scripts:

- One script (“export-oracle”) creates the backup “dump” and sends it to a backup server via Email.

- The other script (“dump_backup.perl”) receives the backup “dump” from Email and stores it into a specific directory.

An entry in the root crontab at the productive server starts the “export-oracle” script:

```
10 */3 * * * /usr/sbin/export-oracle 2>&1 | tee /var/log/oraback/oraback.`/bin/date
+\%Y\%m\%d.\%H\%M`.log | egrep -i "warning|error|fatal" | grep -iv "without warnings"
```

An entry in the root crontab at the receiving server (Barna) deletes entries that are more then 7 days old:

```
20 1 * * *      find /mnt/megaraid3/cluster/Tech/dump_backup -ctime +7 -name '*dmp.gz*' -
exec rm {} \; >> /var/log/temporary_internet_files.log 2>&1
```

10.4.1 Export-oracle

This export script is called by the “crontab” of the productive server:

```
[root@www1 /root]# cat /usr/sbin/export-oracle
#
# /usr/sbin/export-oracle
# V1.1, 010412 Frank Bergmann <frank.bergmann@project-open.com>
#
# V1.0 -> V1.1:
#       - Now using COMPUTER_NAME as first part of dump
#       - Now encrypting using PGP
#       - Now sending out to centralized backup server
#

HOME=/home/oracle
HZ=
LOGNAME=oracle
ORACLE_BASE=/opt/oracle/app/oracle
ORACLE_HOME=$ORACLE_BASE/product/8.1.6
PATH=$PATH:$ORACLE_HOME/bin
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
ORA_OWNER=oracle
ORACLE_SID=ora8
ORACLE_TERM=vt100
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
PATH=$ORACLE_HOME/bin:$ORACLE_HOME/lib:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/loc
al/bin:/usr/sbin
SHELL=/bin/sh
TERM=vt100
TZ=US/Eastern
CURRENT_TIME=`/bin/date +\%Y\%m\%d.\%H\%M`

# Change these!
COMPUTER_NAME=<computer_name>
SERVICE_NAME=<service_name>
DATABASE_PASSWORD=<database_password>
BACKUP_EMAIL=<backup_user_email>

exportdir=/var/log/oraback
file_body=$COMPUTER_NAME.$SERVICE_NAME.$CURRENT_TIME.dmp
file=$exportdir/$file_body

echo "exporting to $file"
su - $ORA_OWNER --command="exp $SERVICE_NAME/$DATABASE_PASSWORD file=$file
owner=$SERVICE_NAME consistent=Y"

echo "zipping and encrypting"
gzip $file
pgp -e $file.gz $BACKUP_EMAIL
```

```
# archieve dump at backup server
uuencode $file.gz.pgp $file_body.gz.pgp | mail $BACKUP_EMAIL
```

The following script is called by the QMail mail system working at the backup server whenever a mail arrives for the backup_user. Check that it has write permissions in the destination directory:

```
# cat ~backup_user/.qmail
| ~backup_user/bin/dump_backup.perl >> /var/log/dump_backup/dump.`/bin/date +"%Y%m%d"`.log
2>&1
```

When working with sendmail, the script has to be named ".forward".

10.4.2 dump_backup.perl

The following script is called by the QMail mail system working at the backup server whenever a mail arrives for the backup user (be sure to create ~backup_user/data directory):

```
#!/usr/bin/perl
#
# dump_backup.perl
#
# 010412 Frank Bergmann <frank.bergmann@project-open.com>
# Receives incoming backup file by qmail/sendmail,
# extracts header information, stores them temporarily
# in the data directory and calls uudecode to restore
# them.

$debug = 1;
$home_dir = "/home/backup";
$data_dir = $home_dir."/data";

$now = `/bin/date +"%Y%m%d.%H%M"`; chomp($now);
print "$now: dump_backup.perl: receiving file\n" if ($debug);

# ignore the header information and start writing the
# file with the begin of the UUEncoded data:
#
$file = $data_dir."/incoming.".$now;
open(F,"> $file");
open(G,"> $file.all");
$flag=0;
while ($line = <STDIN>) {
    print G "$line";

    if ($line =~ /^From/) { print "$now: dump_backup.perl: $line"; }
    if ($line =~ /^begin /) {
        $flag = 1;
        print "$now: dump_backup.perl: $line";
    }
    if ($flag) {
        print F "$line";
    }
}
close(F);
close(G);

# uudecode the file
# ToDo: Check for tmp symbolic link attack!!!
$return = system("cd $data_dir; /usr/bin/uudecode $file > /tmp/uuencode.log 2>&1");

$now = `/bin/date +"%Y%m%d.%H%M"`; chomp($now);

if ($return == 0) {
    system("rm -f $file");
    system("rm -f /tmp/uuencode.log");
    print "$now: dump_backup.perl: finished\n" if ($debug);
} else {
    $err_msg = `cat /tmp/uuencode.log`;
    print "$now: dump_backup.perl: error: $err_msg \n";
    system("rm -f /tmp/uuencode.log");
}
```

10.4.3 PGP Configuration

Both sides of the channel use a standard PGP installation to encrypt the database dump. You need to create a PGP public/private key for backup_user@your_company.com. The public key of the backup server has been extracted using:

```
pgp -kx backup_user@your_company.com /tmp/backup.public.key.pgp
```

and imported into the public key ring of the productive server using:

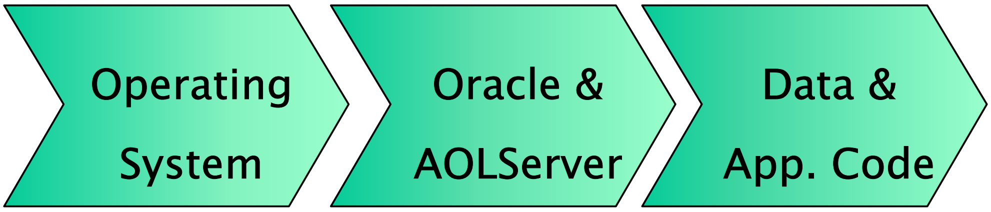
```
pgp -ka /tmp/backup.public.key.pgp
```

11 Failure Recovery

The purpose of this chapter is to describe how to recover a Project/Open application after any kind of incident, ranging from administration mistakes to a complete hardware failure.

11.1 Concept, Tradeoffs and Decisions

The recovery of the system proceeds in 3 major steps that are reflected by the following subchapters:



11.1.1 Assumptions

We assume about the productive server environment:

- That a skilled system administrator is available 24/7 to
  1. Manage the situation,
  2. Communicate with the hosting support and
  3. Perform the necessary SysAdmin recovery steps
- That the hosting support reacts within about an hour to critical situation.
- That the productive server lives in a state of the art hosting center with reasonable infrastructure availability (network & power supplies)
- That the productive server hardware can be replaced in case of a complete failure

11.1.2 Challenges

The main problem with recovery is to avoid a reinstallation of Oracle8i and the AOLServer. In particular Oracle8i causes a considerable delay of 5-10 hours if the 500Mbyte pre-configured TAR has to be transferred through a 128kBit Internet connection.

On the other hand application code and the database content can be restored much faster because:

- The application code is a few MBytes compressed and is maintained by Project/Open, so that it can be regenerated at any time.

- The database content is 3-5Mbyte compressed and is backed up every few hours and sent to the backup server (Barna).

Thus the Oracle installation and the Oracle database content are the critical elements in this backup concept

11.2 Failure Types

All failures that can occur in the productive can be classified into one of the following categories:

1. Primary Hard Disk Failure:
   The main hard disk either fails or a system administrator's failure deletes its content. The secondary hard disk survives.
2. Hardware Failure of both disks (burning server):
   Both hard disks get damaged, for example by a fire or after an unfriendly intrusion.
3. Non-Disk Hardware Failure:
   Some hardware component of the productive server fails such as the power supply or a network card.
4. Hosting Infrastructure or Power Failure:
   Something at the hosting site goes wrong which inhibits the access to the productive server such as power or network shortages.

11.2.1 Recovery Procedures

To deal with the above types of failures, the following procedures are available:

Hosting Center:

1. Swapping the server hard disks:
   The Hosting support can swap the server hard disks on request.
2. Hardware repair:
   The Hosting support can repair defect hardware or replace the entire server by backup hardware within 24 hours.
3. Providing a completely new server hardware

System Administration:

4. Update the Oracle DB data
5. Update the application TCL code
6. Reinstall the Oracle database
7. Reinstall the AOLServer and OpenACS system

The division point between the two companies is that Hosting is responsible to provide a working (Linux) operating system while the System Administrator is responsible to bring the application up and running.

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

11.2.2 Failure Types Against Recovery Procedures

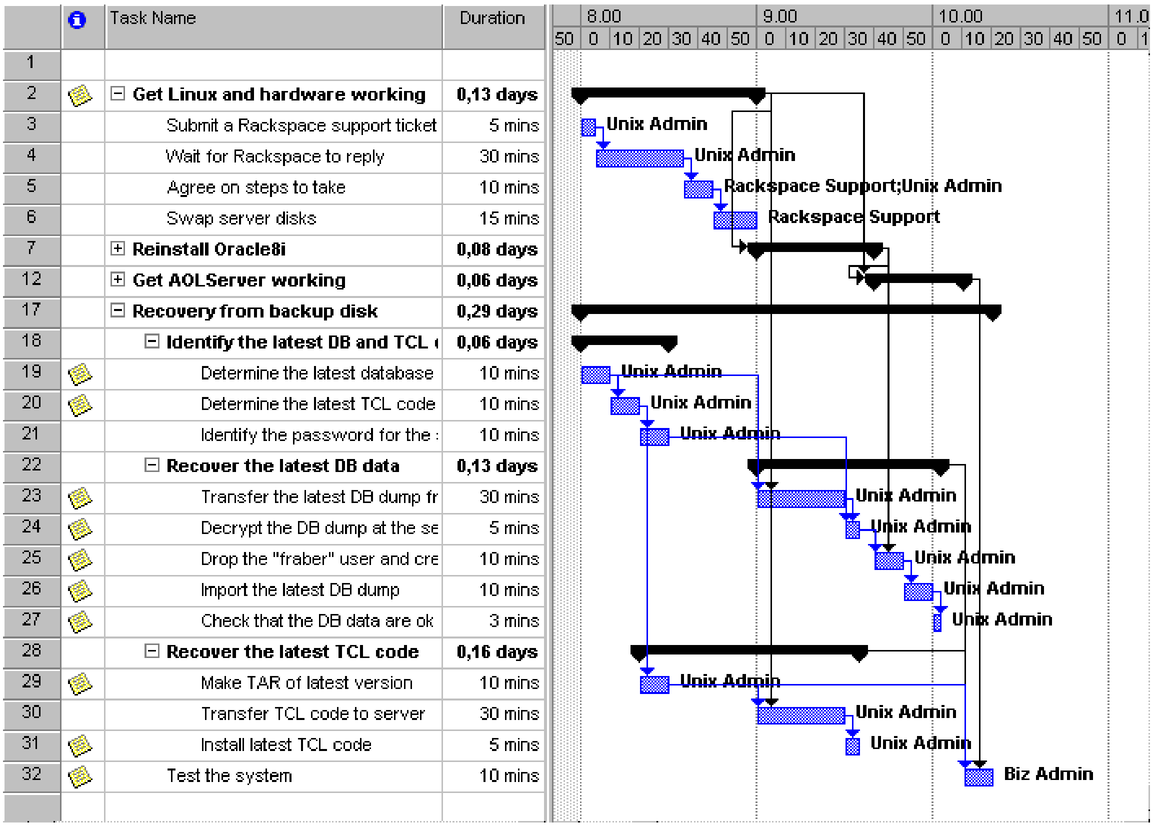
The following table details the steps to be taken in case of a failure:

Failure Type	Recovery Procedures
Primary Hard Disk Failure	1, 4, 5: Fallback to the backup disk and recovery of the latest application data
Hardware Failure of both disks	2, 6,7,4,5: Complete reinstall of the server
Non-Disk Hardware Failure	2: Repair of the server hardware, maintaining the original disks and their content.
Hosting Infrastructure Failure	None: We just have to wait until Hosting gets the servers up again. We might check for an alternative provider in case of a longer failure.

Table 11-1

11.2.3 Recovery Timing

Please see the attached Gant chart for a timing of the recovery procedures. This chart assumes a Primary Hard Disk Failure:



	ℹ	Task Name	Duration
1			
2	🗒	⊟ **Get Linux and hardware working**	**0,13 days**
3		Submit a Rackspace support ticket	5 mins
4		Wait for Rackspace to reply	30 mins
5		Agree on steps to take	10 mins
6		Swap server disks	15 mins
7		⊞ **Reinstall Oracle8i**	**0,08 days**
12		⊞ **Get AOLServer working**	**0,06 days**
17		⊟ **Recovery from backup disk**	**0,29 days**
18		⊟ **Identify the latest DB and TCL (**	**0,06 days**
19	🗒	Determine the latest database	10 mins
20	🗒	Determine the latest TCL code	10 mins
21		Identify the password for the :	10 mins
22		⊟ **Recover the latest DB data**	**0,13 days**
23	🗒	Transfer the latest DB dump fr	30 mins
24	🗒	Decrypt the DB dump at the se	5 mins
25	🗒	Drop the "fraber" user and cre	10 mins
26	🗒	Import the latest DB dump	10 mins
27	🗒	Check that the DB data are ok	3 mins
28		⊟ **Recover the latest TCL code**	**0,16 days**
29	🗒	Make TAR of latest version	10 mins
30		Transfer TCL code to server	30 mins
31	🗒	Install latest TCL code	5 mins
32	🗒	Test the system	10 mins

Figure 11-1

11.3 Linux Operating System Recovery

There procedures deal with the restoration of the productive server to the point where the Linux operating system is working again

11.3.1 Swapping the server hard disks

On request, Hosting can perform a swap of the server hard disks within 1-2 hours, thus restoring the server state at the time of the last full backup. Please see the section “Full Backup” for details on how the backup is made.

11.3.2 Hardware repair

The Hosting support can repair defect hardware on request.

11.3.3 Providing a completely new server hardware

The Hosting support can replace the entire server by backup hardware within 24 hours.

11.4 Installation From Scratch

These procedures are used to setup Oracle, AOLServer and ACS after a complete crash up to the point where it is possible to upgrade to the latest backup of the application (database data + ACS code).

Please see the installation section for details.

11.5 Oracle and ACS Update Procedures

11.5.1 Recover the Current Application TCL Code

To recover the current version, extract the current version of the application from the CVS system at the development server: ToDo: Where to get the code?

1. Determine the current marketplace version of the productive server: In our server room we have a paper table at the wall with the complete release history. Check the latest release for the productive server.
2. Checkout the latest version from CVS:
   ```
   # cd web
   # cvs co -r version -d directory module
   ```
   with the parameters:
   version: the name of the version to recover (from paper sheet)
   directory: the target directory for the extracted code
   module: the name of the module (also from paper sheet)
3. Example:
   ```
   # cd web
   # cvs co -r eshkol-1-0-0 -d dev productive-acs
   ```
4. Tar the directory, transfer it to the productive server and untar it. That is all; the code is completely “passive” and fits to the version of the data model in Oracle.

11.5.2 Load the Latest Database Backup Dump into Oracle

The following procedure describes how to charge a running Oracle8i database system with the latest backup dump of the database. It assumes that the database is running and the user and the tablespace “intranet” exist.

1. Make sure that Oracle is up and the tablespace and the user exist:
   ```
   # sqlplus intranet/intranet
   SQL> select email from users;
   SQL> quit
   ```
   you should get a list of several emails.
2. Login onto Oracle as “system” and delete the user “intranet” (the default user of the Project/Open application):
   ```
   # sqlplus system/manager
   SQL> drop user intranet cascade;
   SQL> create user intranet identified by intranet default tablespace
   intranet temporary tablespace temp quota unlimited on intranet;
   SQL> grant connect, resource, ctxapp, javasyspriv, query rewrite to
   intranet;
   SQL> revoke unlimited tablespace from jruiz;
   SQL> alter user jruiz quota unlimited on jruiz
   SQL> quit
   # imp system/manager file=www1.intranet.yyyymmdd.hhmm.dmp
   log=/tmp/log.log
   ```
3. Repeat step 1.) to make sure the tablespace is OK and restart the AOLServer (“killall –9 nsd” or svc –t /sbin/service/intranet (when using daemontools). Pray that it´s working now (good hint from a catholic Irish system administrator who used to work here...).
4. Repeat step 1.) to make sure the tablespace is OK and restart the AOLServer (“pkill –9 nsd” or svc –t /sbin/service/intranet (when using daemontools). Pray that it´s working now. (Good hint from a catholic Irish system administrator who used to work with us once upon a time...).

11.6 Informing the Users

It is useful to inform the users about system maintenance when the Linux OS is working, but the ACS is still under maintenance:

11.6.1 “Not Available” Screen during ACS updates

If for any reason the ACS server or the database becomes unavailable, you have to swap the AOLServer (OpenACS) with the Apache web server (static pages).

Normally, AOLServer operates on port 80 (http) and Apache is installed to work on port 81. But during maintenance we take advantage of the preinstalled Apache screen to inform our users about the maintenance.

This is done by starting Apache on port 80 and the AOLServer on port 81:

1. Edit the Apache configuration file at /etc/httpd/httpd.conf and change the line “Port 81” to “Port 80”.

2. Edit the AOLServer configuration file at /home/aolserver/aolconf.tcl and change the line “ns_param Port 80” to “ns_param Port 81”
3. Stop the Apache server: “/etc/rc.d/apache stop” (SuSE 7.x) or “/etc/rc.d/httpd stop” (RedHat).
4. Restart the AOLServer: “killall –9 nsd” (it´s being restarted by Inittab or the daemontools).
5. Start Apache again: “/etc/rc.d/apache start” (SuSE 7.x) or “/etc/rc.d/httpd start” (RedHat).

12 Security

12.1 Concept, Tradeoffs and Decisions

12.1.1 Concept

Security in the context of our Marketplace application can be defined in terms of:

- Loss of Data:
  The chapter about backup deals with the strategy to avoid or limit the loss of data.
- **False creation or alteration of data:**
  This case has to be solved on the application level by introducing strong authentication measures because false creation of data can be due to company members, users or intruders.
  Currently (April 2001), there is no strong authentication build into the application because of user acceptance problems. So this is a political decision.
- **Distribution of Data:**
  Here application administrators, system administrators and intruders are the critical groups. The usual precautions are taken (see below).

12.1.2 Tradeoffs and Decisions

A major decision had to be taken concerning the security/accessibility of the productive database to application developers. Until now (April 2001) and probably during the rest of this year, a lot of changes have to be made to the application, including its data model, so an easy access of the developers to the productive database is essential. This means that the application data cannot be secured (neither access nor change) against application developers and system administrators.

To deal with this situation in a reasonable manner, we are currently (April 2001) introducing a “Bid Printer” to protocol all changes to the database on paper to make changes traceable.

Another decision has been made not to protect the productive server by an additional firewall (just the linux one). Instead, the “usual precautions” are taken to prevent intrusion by closing down al unnecessary Unix services and choosing strong passwords.

12.2 Implementation

12.2.1 Unix Level Security

The security concept on the Unix level is based on the idea to disable all unnecessary services and to analyzing the remaining services in detail for security. Following closely the “Bugtraq” security mailing list and updating the affected components guarantees the security of the Linux operating system itself.

Below please find a listing of the remaining services. Their respective security is further analyzed in the following chapters.

```
[root@www1 /root]# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:smtp                    *:*                     LISTEN
        (Sendmail)
tcp        0      0 *:www                     *:*                     LISTEN       (OpenACS)
tcp        0      0 *:https                   *:*                     LISTEN       (OpenACS)
tcp        0      0 *:81                      *:*                     LISTEN       (Apache)
tcp        0      0 *:4829                    *:*                     LISTEN       (ToDo:
Check process)
tcp        0      0 *:1984                    *:*                     LISTEN
        (BigBrother)
tcp        0      0 *:ssh                     *:*                     LISTEN       (Secure
Shell)
```

12.2.2 OpenACS Security

The OpenACS application is the most critical security risk in the system because it is under continuous development.

The overall OpenACS architecture consists of:

- **The AOLServer Web server**:
  The least critical component because it is an open source application being used by America Online (AOL) in their own environment. The development group is following closely the “AOLServer” mailing list to check for security exploits.
- **The OpenACS base system**:
  The ACS base system consists of a set of TCL pages that has to be checked for security together with the:
- **Application specific pages for our Marketplace application**:

The security check involves for each page the following criteria:

- Check that all pages require authentication for the appropriate user groups
- Check that all parameter that appear as part of a SQL query are formed appropriately.

Please see the separate document “Security Analysis Marketplace YYMMDD.xls” for a detailed analysis.

12.2.3 Apache Security

Some excerpts from /etc/httpd/conf/httpd.conf:

12.3 Configuration Details

```
# /etc/httpd/conf/httpd.conf
#
Port 81
DocumentRoot "/home/httpd/html"
User nobody
Group nobody
```

```
#
# Limit the load to Apache
#
MinSpareServers 2
MaxSpareServers 20
StartServers 2
MaxClients 50
```

CGI-Directory:

There is only one CGI directory with scripts from “Big Brother”:

```
# /etc/httpd/conf/httpd.conf
#
<Directory "/home/httpd/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

Content of /home/httpd/cgi-bin: These scripts form part of the current “Big Brother” distribution. Big Brother security exploits are reported to the monitored “Bugtraq” security mailing list.

```
bb-ack.sh
bb-hist.sh
bb-histlog.sh
bb-hostsvc.sh
bb-rep.sh
bb-replog.sh
```

12.3.1 SSH & Login Security

The regular telnet login is disabled.

Only the following users should figure in the /etc/passwd:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
www:x:100:101::/home/www:/bin/false
named:x:25:25:Named:/var/named:/bin/false
majordomo:x:91:91:Majordomo List Manager:/usr/lib/majordomo:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
squid:x:23:23::/var/spool/squid:/dev/null
fbergman:x:500:500::/home/fbergman:/bin/bash
rack:x:501:501::/home/rack:/bin/bash
oracle:x:504:504::/home/oracle:/bin/bash
nsadmin:x:505:508::/home/aol30:/bin/bash
webmaster:x:508:511::/home/webmaster:/bin/bash
bb:x:510:507::/usr/local/bb:/bin/sh
```

```
fax:x:10:14:Facsimile Agent:/var/spool/fax:/bin/bash
```

12.3.2 Sendmail Security

It is necessary to run Sendmail because the OpenACS application needs it to send Email. However, the access to the Sendmail port is blocked completely to the outside world.

```
# /etc/sysconfig/ipchains
#
:input ACCEPT
:forward ACCEPT
:output ACCEPT
-A input -s 209.61.155.151/255.255.255.255 -d 209.61.155.151/255.255.255.255 25:25 -p 6 -j
ACCEPT
-A input -s 0.0.0.0/0.0.0.0 -d 209.61.155.151/255.255.255.255 25:25 -p 6 -j REJECT
```

12.3.3 Oracle Security

The Oracle database is not accessible from outside the computer. For this reason, the “lister8i” service has to be disabled.

13 Maintenance

13.1 Directories to watch for it's size

13.1.1 /var/log/aolserver

You can safely delete the "error" log files. They are used for debugging. The rest of log files record who has accessed that server via his browser. You can use them, for example, for statistical purposes, user profile, usage graphs or marketing campaign results.

13.1.2 Core files

The following command will find them:

```
Find / -name core
```

You can safely delete them.

13.1.3 Oracle Files

Check for the size used by Oracle itself (/opt/oracle) and it's databases (installed in /ora/):

/ora/m01

/ora/m02

/ora/m03

13.2 Big Brother

We suggest to use Big Brother for continuous monitoring of table spaces and disk partitions

13.3 Adding a new Virtual Server (ToDo: update)

Summary:

1. Adding the user to the Solaris system

```
# useradd -g nsadmin -m -s /usr/bin/bash <new_user>
# passwd <new_user>
```

2. Creating an Oracle tablespace

```
# su - oracle
# sqlplus system/manager
SQL> create tablespace <new_user> datafile '/ora/m02/oradata
/ora8/<new_user>01.dbf' size <tablespace_size>m autoextend on
default storage ( pctincrease 1);
SQL> create user <new_user> identified by <new_user> default
tablespace <new_user> temporary tablespace temp quota unlimited on
<new_user>;
SQL> grant connect, resource, ctxapp, javasyspriv, query rewrite to
<new_user>;
SQL> revoke unlimited tablespace from <new_user>
SQL> alter user <new_user> quota unlimited on <new_user>;
SQL> quit;
```

Check that it worked:

```
sqlplus <new_user>/<new_user>
SQL> select sysdate from dual;
```

A reasonable value for <tablespace_size> is “5”

3. Providing a copy of the ACS for the new user

```
# cp -r /web/<existing_acs> /web/<new_user>
# chown -R <new_user>:nsadmin /web/<new_user>
```

4. Loading acs into the user's new tablespace

```
# su - oracle
# cd /web/<new_user>/www/doc/sql
# sqlplus <new_user>/<new_user> < load-data-model.sql
```

This command takes several minutes. Please note that you might have to create module specific tables if the developers did not include them into the “load-data-model.sql” file.

5. Adding a new configuration file to /usr/local/aolserver

```
# cd /usr/local/aolserver
# cp aolconf.<org>.tcl aolconf.<new_user>.tcl
# vi aolconf.<new_user>.tcl
  -> Change “<org>” to “<new_user>”
```

6. Define an IP port in /usr/local/aolconf.config

```
# vi aolconf.config
-> Add a lines for “hostname”, “addresses” and “ports” for
<new_user>
```

7. Edit the /web/<new_user>/parameters/*.tcl config file:

   a. Globally replace “podemo” by “<new_user>

8. Edit /etc/inittab and add:

```
# ao:234:respawn:/usr/local/aolserver/bin/nsd-oracle -i -u
<new_user> -s <new_user> -t
/usr/local/aolserver/aolconf.<new_user>.tcl >/dev/msglog 2>&1
```

9. Check whether the server is up and running:

   a. Check for the process list

   ```
   # ps -ef | grep nsd
   ```

   This command should give you the list of running AOLServers (“nsd”). Make sure your new server is there “-s <new_user>”.

   b. Connect to the server using the values from aolconfig.config:

   ```
   # telnet <ip_address> <port>
   ```

   Type in (capital letters!)

   ```
   GET /
   ```

You should get some HTML output consisting of an error message (because the “GET /” did not contail all necessary information).

c. Check for the server log file for notification of problems during startup. You may delete the file and restart the server if you find too many lines:

```
# less /var/log/aolserver/xdomen-error.log
```

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

Ronda Sant Antoní, 51 1° 2a
08011 Barcelona, Spain
Tel.: +34 93 325 0914
Fax.: +34 93 289 0729