

CYBERNET

OpenSkies

Networking Engine

Netmax™/Openskies Guide



www.openskies.net

MMPG
MASSIVE
MULTIPLAYER
ONLINE GAMING

NetMAXOS Setup

The Openskies SDK includes executables, examples, scripts, and configuration files that reside on a PC running the Linux Operating system. Because the Openskies SDK is a networking architecture, we have integrated it in with Cybernet's Netmax software and called this distribution of the NetMAX software NetMAXOS (the OS for Openskies). Installation of the NetMAXOS software included as part of the Openskies SDK contains the elements required to execute one or more of the following servers:

1. ***FedHost*** – This process handles all routing of federation data.
2. ***LobbyManager*** – The lobby manager process accepts, authenticates and mediates player connectivity
3. ***Database Server*** – The database server accepts connections from individual clients (federates), webpages, FedHosts and LobbyManagers. It is used as a central repository for network status information, user management, and authentication. Primary functionality included in the mysql database is to record and report network status. Other elements of the DB server are to be regarded as examples or templates that you can use to build your own Database architecture. They are distributed with full source code.

Source Code

Distribution of the source code and examples for the Openskies server is located in the top directory of the Netmax Openskies distribution CDROM. It is a tape archive (.tar) file. Source code for the Openskies SDK server can be made available to customers as part of the license agreement.

Binary Distribution

Server.pl

NetmaxOS gives you the ability to run any combination of the first three servers above (*FedHost*, *LobbyManager*, or *DB Server*) on a particular machine. It does this using a script that is invoked at runtime. This script is called *server.pl* and it is a perl script that is automatically run upon boot up. A standard NetmaxOS has this file in the following location:

```
/usr/netmax/etc/rc.d/server.pl
```

Openskies Server Directory Tree

\$root is the toplevel for the directory tree below. It is currently set to /usr/openskies on the NetMAX build.

\$root/

- o *FedHost/ - copied from Source*
 - FedHost0/
 - CullingModules/
 - o CullingRulesII.so
 - o CullingRulesIII.so
 - o EoECulling.so
 - o CullingRulesII.ini – *not .example. Copied from .example in src*
 - o CullingRulesIII.ini – *not .example. Copied from .example in src*
 - o EoECulling.ini – *not .example. Copied from .example in src*
 - FEDFiles/
 - o EoE.fed
 - o eoe2.fed
 - o OpenSkies.fed
 - o OpenSkiesNoCulling.fed
 - o Aircraft.fed
 - o Pswitch.fed
 - FedHost - *binary*
 - FedHost.ini.example – *copy of FedHost.ini.example (source)*
 - FedHostSecurity.so
 - o *LobbyManager/ - copied from Source*
 - LobbyManager0/
 - LobbyManager.ini.example – *copy of LobbyManager.ini.example (source)*
 - LobbyManager - *binary*
 - LobbySecurity.so
 - o *DBServer/ - from Source/eoe_db*
 - eoe
 - zipcode.txt

/etc/

- eoedb.sql
- eoedbusers.sql

/usr/netmax/etc

- /rc.d/
 - o server.pl

/var/mnt/Web/external

- index.html
- /cgi-bin/
 - o query.cgi

FedHost Startup

The FedHost process is described in the *Openskies Network Architecture* in more detail. This section describes how to automatically start up FedHosts.

To setup a FedHost first a *FedHost.ini* file must be set up. A description of the FedHost.ini file is located below. The mechanism by which Server script executes FedHosts is by traversing the directory tree (above) and looking for *FedHost.ini* files. The directory *FedHost0*, in the tree above, illustrates how the files should be laid out to run a FedHost. Creating a *FedHost.ini* and placing it in the FedHost0 directory will cause a FedHost to be run upon the next boot. To run more than one FedHost on a computer simply copy the directory structure of FedHost0 to FedHost1 and edit the FedHost.ini in FedHost1 to prevent a resource conflict between FedHosts.

FedHost.ini

The following is the actual content of a FedHost.ini file. Lines that start with “;” are comments.

```
[Repeaters]
;You must specify the IP address of the Ethernet card to use
ConnectorPort = 3500
ConnectorAddress = 192.168.1.10

[FederationHost]
SocketLingerOption = 0
SocketLingerTime = 0
; UDP datagrams are sent to federates directly if this is set to 0.
; If this is set to one, they are forwarded to the FedHosts who are
; connected to the federates.
FwdAttributeSetUpdates = 0
; Authentication code is optional. However,
; if the LobbyManager is using one,
; you have to list one here. If this line is
; absent, authentication is disabled
SecurityModule = FedHostSecurity.so

[Database]
; This is optional. If you have a database server
; for federation information,
; add it here in the form of IP address:port number.
ServerAddress=192.245.33.49:8100
```

LobbyManagers Startup

The LobbyManager process is described in the *Openskies Network Architecture* in more detail. This section describes how to automatically start up a LobbyManager.

To setup a LobbyManager first a *LobbyManager.ini* file must be set up. A description of the *LobbyManager.ini* file is located below. The mechanism by which Server script executes LobbyManagers is by traversing the directory tree (above) and looking for *LobbyManager.ini* files. The directory *LobbyManager0*, in the tree above, illustrates how the files should be laid out to run a LobbyManager. Creating a *LobbyManager.ini* and placing it in the LobbyManager0 directory will cause a LobbyManager to be run upon the next boot. To run more than one LobbyManager on a computer simply copy the directory structure of *LobbyManager0* to *LobbyManager1* and edit the LobbyManager.ini in LobbyManager1 to prevent a resource conflict between LobbyManagers.

LobbyManager.ini

The following is the actual content of a FedHost.ini file. Lines that start with “;” are comments.

```
[LobbyManager]
;You must specify the IP address of the Ethernet card to use
LobbyAddress=123.123.12.1
LobbyPort=2000
; Authentication code is optional. If you have a security module,
; list it here. If this line is
; absent, authentication is disabled
SecurityModule=LobbySecurity.so

[Multicast]
; Multicast port and address used by LobbyManager
MCastAddress=224.9.9.1
MCastPort=22500
MCastTTL=5
; Each federation will be assigned with a multicast port.
; The port number starts at McastPort+1, and will not
; exceed MaxPort.
MaxPort=23500

[Repeaters]
; Total number of repeaters connected to this LobbyManager
Count=2
;Entries on the following lines are in the following order:
;max. federate per host, repeater address, repeater port,
;FedHost address, and FedHost port range
Repeater0=300, 192.168.1.10, 3500, 123.123.12.2, 3000, 3400
Repeater1=300, 192.168.1.10, 4500, 123.123.12.3, 3500, 3800

[Database]
; This is optional. If you have a database server for
; federation information,
; add it here in the form of IP address:port number.
ServerAddress=192.245.33.49:8100
```

The structure of this initialization file allows you to configure for multiple network cards on both the LobbyManager machine as well as on each FedHost. The IP address specified in the [LobbyManager] section is the address by which external federates (players/clients) will contact the LobbyManager. Each *Repeater* (FedHost) line of the [Repeaters] section contains two

Address:Port pairs. The first of these specifies the network path that will be used for all *FedHost-FedHost* and *LobbyManager-FedHost* communication. The second address is used for all communication to and from external federates (clients/players) Furthermore, the second actually specifies a port range, which must be wide enough to accommodate the number of, federates specified at the beginning of the line. By specifying different IP addresses in these locations you will be able to make use of multiple Ethernet cards in both your LobbyManger and FedHost, thereby separating strictly FedHost/LobbyManger communication from any communication involving federates (clients/players).

Database Server (DBServer)

Database Creation

Before attempting to use or start up a DB server, the database must be created from a template located in `/etc/eoedb.sql`. It utilizes the file `zipcode.txt` during initial creation.

```
$root/DBServer/zipcode.txt
```

Someone wishing to customize the DB is advised to look at `/etc/eoedb.sql` to view the way in which the EoE sample game database was set up. The file `/etc/eoedbusers.sql` sets up the initial DB users.

A more detailed description of the SQL database is located in *Openskies Database Guide*.

Startup

To start the database server a file `dbserver.conf` be created in the DBServer directory. Creating the file will indicate that the DB should run. The file must contain a hostname. If the hostname is that of the local host, then the computer will act as a database master. The master database will allow other computers to mirror it. To set up the local database as a mirror set the hostname to that of another computer. In this case, the local DB server will act as a slave that will mirror the host named in `dbserver.conf`. If this is the first time the DB has been instructed to run, the file `nodb` will be in `/etc`, after the successful completion of the following `/etc/nodb` will be removed.

System Console Web Access

This `query.cgi` script (described in the *Openskies Database Guide*) allows the user to remotely browse network status information residing in the EoE. Browser access to this Netmax machine will automatically be forwarded to this `cgi`.

In order to enable this interface, which ultimately queries the database, the NetMAX Internet Web Services (Home/Services/WWW) must be enabled. Please refer to the NetMAX documentation for further assistance.

EoE Example Web Server

The NetMaxOS distribution includes some sample webpages and a cgi perl script called `oeo_edit` which queries and updates the database residing on the DB Server (to create and edit player information). In order to create a web server that allows users to add and view their own account information, simply copy the content of `$root/oeo_db/oeo_edit` into a cgi-bin capable directory on the machine you designate as your web server.

A more detailed description of this program and how they function may be found in the *Openskies Database Guide*.

Security/Firewall Considerations

Your system administrator will have to make sure that any firewalls are configured to let through the traffic corresponding to the ports and addresses specified above. Specifically:

For each LobbyManager machine:

- *LobbyManager.ini*
 - [LobbyManager] - external federates (clients/players) must be allowed to connect to this LobbyManager over the address/port specified by `LobbyPort` and `LobbyAddress`.
 - [Repeaters] - for each repeater line corresponding to a FedHost, firewall rules must be made to accommodate the following:
 - `repeater address, repeater port` - This LobbyManager must be able to contact this FedHost using this address and port.
 - [Database] - this LobbyManager must be able to contact the Database server process listening on the specified `ServerAddress`

For each FedHost machine:

- *FedHost.ini*
 - [Repeaters] - The LobbyManager and all other FedHosts must be allowed to connect to this FedHost over the address/port specified by `ConnectorPort` and `ConnectorAddress`.
 - [Database] - this FedHost must be able to contact the Database server process listening on the specified `ServerAddress`
- *LobbyManager.ini*
 - [Repeaters] - for the repeater line corresponding to this FedHost firewall rules must be made to accomedate the following:

- `repeater address, repeater port` – The LobbyManager and other FedHosts must be able to contact this FedHost using this address and port.
- `FedHost address, and FedHost port range` – External federates (players/clients), must be able to contact this FedHost using this address over this port range.

For each Database Machine:

- *FedHost.ini* and *LobbyManager.ini*
 - `[Database]` – all FedHost with this section active (not commented out) must be able to contact this Database server process listening on the specified `ServerAddress`