

# Nets Installation and Administration Guide

---

Copyright (C) 1999-2005  
Mike McCauley and Hugh Irvine

Installation and Administration Guide for  
the Nets network inventory and  
management system.  
For Nets Revision 2.5

---

## 1.0 Introduction

---

Nets is a graphical source code product providing a flexible and extensible environment for maintaining essential network inventory, configuration and cost information in a platform and database independent manner.

Nets provides a physical model of the network, from which it is simple and straightforward to derive and display any number of textual, logical or graphical views.

Nets is designed by network engineering professionals to address the common problems encountered in network design, management and operation.

In keeping with its platform and database independent design, Nets can be installed and run on a number of different platforms with a number of different databases.

This document describes the installation and administration of the Nets system on two common platforms: UNIX (any version) and Windows 95/98/2000/NT.

Readers of this document are expected to have some knowledge of system administration, SQL databases, software installation and the Perl environment. You will also need to have access to superuser privilege for some parts of the installation process.

---

## **2.0 Overview**

---

### **2.1 Prerequisites**

Nets is a graphical application written in Perl that operates on objects and tables stored in an SQL database. The graphical subsystem used in Nets is Tk, which is available for many environments. As well as network data, Nets also stores user details, including encrypted passwords. Therefore, Nets has the following prerequisites:

1. Perl. Perl is a freely available program for all Unixes, Win95/98, NT and others. Full source code and binaries for Unix are available at <http://www.perl.com>. For Win95/98 and NT, we recommend ActivePerl from ActiveState, which is also freely available at <http://www.activestate.com>.
2. Tk. Tk is a graphical subsystem that is available for Perl. Tk is available from CPAN ([www.perl.com/cpan](http://www.perl.com/cpan)) and ActiveState.
3. MD5. MD5 is used to encrypt Nets user passwords. MD5 is available from CPAN and ActiveState.
4. Perl modules for accessing SQL. The DBI and DBD modules for most free and commercial databases are available from CPAN and ActiveState. You will need to install the DBI module, and the DBD module that corresponds to the SQL database you plan to use.
5. SQL Server. Nets uses an SQL database to store details about your network. Nets will work with a very wide range of free and commercial SQL servers on a wide variety of platforms, including:
  - Oracle ([www.oracle.com](http://www.oracle.com))
  - Sybase ([www.sybase.com](http://www.sybase.com))
  - Microsoft SQL 6.5 and 7.0 ([www.microsoft.com](http://www.microsoft.com))
  - MySQL ([www.mysql.org](http://www.mysql.org))
  - mSQL ([www.hughes.com.au](http://www.hughes.com.au))
  - Postgresql ([www.postgresql.org](http://www.postgresql.org))
  - Informix ([www.informix.com](http://www.informix.com))
  - InterBase ([www.interbase.com](http://www.interbase.com))
  - many others

No matter which SQL database you choose, you will probably have to nominate one of your staff to become expert in maintaining it, especially for backups of your vital network database.

You will need to have a significant amount of disk space available on your SQL server to hold the Nets database. For most systems, you will need at least 100Mb of space for your Nets tables.

### **2.2 Deployment**

A complete Nets system consists of 2 largely independent components:

- Nets itself, running on one or more “User Hosts”
- SQL Server, running on an “SQL Host”

These 2 components may be installed on a single computer, or installed on separate computers, depending on your license and requirements. You will need to decide how to deploy the Nets application on your user’s computers, whether separate installations on multiple workstations or PC’s or possibly a single User Host with remote access using X-Windows. Some of the factors that could influence your decision may be:

- Available hardware and performance.
- Available SQL server licenses and costs.
- The size of your network.
- The size of your user population and their workstation platforms.

For a single user system, it is usual to run both components on the one computer (i.e. User Host and SQL Host are one and the same computer). For larger user populations it is common to run a separate SQL Host, and multiple User Hosts, one for each user.

### 2.3 Licensing

Nets licensing is dependent on the number of concurrent Nets users. If you purchase an unlockable license, it will limit you to particular number of concurrent Nets users. An unlimited license has no such restrictions.

Nets license keys are installed into a Nets configuration file during the installation process.

Nets keeps count of the number of current Nets users in the database, and when all the licenses are used, no more users will be permitted to log in to Nets. If a Nets users computer crashes without disconnecting from the database, their Nets license will be automatically expired after 10 minutes, and their license will again become available for another user.

Therefore, one of the deployment considerations should be the number of users that need to use Nets, and the number of licenses that you have purchased. You can purchase more licenses by contacting [sales@open.com.au](mailto:sales@open.com.au).

## 3.0 Installation Overview

---

No matter how many computers you plan to deploy Nets on, installation and configuration consists of the following basic steps. There are more detailed instructions below.

- Make sure you are (or have access to) a system administrator and someone who understands your SQL database environment.
- Purchase and download Nets.
- Install your selected SQL server on your SQL host.
- Install Perl on your Nets host(s) and your SQL host.

- Install Tk and MD5 on your Nets host(s).
- Install DBI and the DBD module for your selected SQL server on your Nets host and your SQL host.
- Unpack your Nets distribution (see below).
- Create the Nets database and tables (see below).
- Install and configure Nets on your Nets host(s) (see below).
- Test the entire Nets/SQL database system (see below).
- Arrange for backups of your SQL database to be taken periodically.

The detailed steps are somewhat different for Unix and Windows 95/98/NT/2000 systems, so we have given detailed instructions separately for each below. In a mixed environment, you may have to follow instructions for part of the install for Unix and part for Windows.

---

## 4.0 Installation on UNIX

---

You should already have installed your selected SQL server, and be familiar with administering it.

### 4.1 Install Perl and required support modules

Chances are that Perl will already be installed on your system, but you will still be required to compile and install the following modules, which can all be downloaded from CPAN ([www.perl.com/cpan](http://www.perl.com/cpan))

- MD5 (1.7 or better)
- DBI (1.13 or better)
- DBD for your selected SQL server (the latest version)
- Tk (800.022 or better)

In general, you should follow the standard procedures for building perl modules:

1. Download the package
2. unpack it in a working directory with:  

```
gunzip -c filename|tar xvf -
```
3. `cd distributiondir`
4. `perl Makefile.PL`
5. `make`
6. `make test`
7. (as root) `make install`

On Linux platforms you should be able to get RPM packages for each of the above. If so it will be quicker to use those than to build the package. Use `rpm` to install RPM packages like this:

```
rpm -Uvh packagefilename.rpm
```

### 4.2 Unpack your Nets distribution.

#### 4.2.1 All Unix platforms

The Nets distribution is supplied as a tarred and gzipped file called something like Nets-2.1.tgz. You will need to move it to a work area and unpack it.

```
gunzip -c Nets-2.1.tgz | tar xvf -
```

This will create a directory called Nets-2.1 in the current directory.

```
cd Nets-2.1
```

#### 4.2.2 Linux

Nets is also available as an RPM binary package, and you may prefer to install from the RPM file instead of the standard Unix tar-gzip package. The typical package name is something like Nets-U-2.2-1.i386.rpm. Download the package, place it in a temporary directory and install it (as root) with:

```
rpm -Uvh Nets-U-2.2-1.i386.rpm
```

This will create an installation directory `/usr/local/nets`, install the Perl library files in `/usr/lib/perl5/site_perl/5.6.0` and Nets executables in `/usr/bin/nets*`

```
cd /usr/local/nets
```

### 4.3 Create the Nets database

Instructions for specific databases follow:

#### 4.3.1 mSQL

1. Edit `installMsql.sh`, check and possibly change the configurable variables near the top to suit your site.
2. `sh installMsql.sh`

#### 4.3.2 MySQL

1. Edit `installMysql.sh`, check and possibly change the configurable variables near the top to suit your site.
2. Become root
3. `sh installMysql.sh`

#### 4.3.3 Oracle

1. Edit `installOracle.sh`, check and possibly change the configurable variables near the top to suit your site.
2. `sh installOracle.sh`

#### **4.3.4 Sybase**

1. Edit installSybase.sh, check and possibly change the configurable variables near the top to suit your site.
2. sh installSybase.sh

#### **4.3.5 Postgresql**

1. Edit installPostgresql.sh, check and possibly change the configurable variables near the top to suit your site.
2. sh installPostgresql.sh

#### **4.3.6 Other Databases**

We have not provided installation scripts for other databases. You will need to follow these steps:

1. Following your SQL vendor's instructions, create an empty database.
2. Create a database user that has permission to create tables in the new database.
3. Set a password for the new database user.
4. Edit Nets/Sql.pm, and configure it so that Nets can connect to the SQL database.
5. Try to automate the entire process and send us an installXXX.sh so others can benefit from your experience.

---

## **5.0 Installation on Windows 95/98/2000/NT**

---

You should already have installed your selected SQL server, and be familiar with administering it.

### **5.1 Install Perl and required support modules**

1. Download and install ActiveState Perl from <http://www.ActiveState.com>. During installation, accept all the defaults. Allow setup to reboot your computer.
2. Connect your computer to the Internet so you can download the required Perl modules from ActiveState using PPM in the next step.
3. Double click on c:\perl\5.00502\bin\ppm (the Perl package manager). You will get a command line screen running ppm with a PPM> prompt.
4. Type "install MD5". The MD5 package will be downloaded and installed.
5. Type "install Tk". The Tk package will be downloaded and installed.
6. Type "install DBI". The DBI package will be downloaded and installed.
7. Type "install DBD-ODBC". The DBD-ODBC package will be downloaded and installed. This will allow Nets to connect to any ODBC compliant SQL server.
8. Close the PPM window. Perl is now installed.

## 5.2 Unpack your Nets distribution

The Nets distribution is supplied as a tarred and gzipped file called something like Nets-2.1.tgz. You will need to move it to a work area and unpack it. We recommend you unpack it into a directory called something like c:\Nets-2.1.

The Nets distribution can be unpacked with any recent version of WinZip.

## 5.3 Create the Nets database and tables

Here we will describe how to create a new Nets database on Microsoft SQL 6.5 and 7.0. For other SQL servers, follow the database vendor's instructions.

First, you create an empty database for the Nets tables. To do this, you will use the MS-SQL Enterprise Manager, which will have been installed on your Windows host when you installed MS-SQL.

In order to create a database, you must first create 2 database devices. A database device is the physical storage on which a database resides. You will create 2 devices: one for the database itself, and one for the transaction log. The transaction log is used to store recent database transactions and is very important for database recovery purposes. Do not neglect to make a log device.

You must now decide how big your database device is to be. As a rough guide you will need approximately 1kb per Nets object, and additional space per session to hold audit details. Therefore the database size depends on how many Nets objects you intend to keep in the database, how many users you need to support, how often they log in, and how far back you intend to keep audit information. If you are unsure how big to make your database, just say 100Mb. You can always enlarge it later. The device for the transaction log should be about 20% of the size of your database device.

### 5.3.1 Creating an empty database on MS-SQL 6.5

1. Run SQL Enterprise Manager. You may have to register your SQL server the first time you connect by entering the IP address, the "sa" login and the sa password (which is blank in a newly installed SQL server).
2. Choose Manage->Database Devices in the menubar. Click on "Create New Device". Enter the new device name (we recommend "netsdev") and the database device size you chose above. Enter a suitable location for the device (this is where the database file will be stored).
3. Using the same method, create the transaction log device. We recommend that you name the log device "netslogdev".
4. Choose Manage->Databases in the menubar. Click on "Create New Database". Specify a database name (we recommend "nets"). Make sure you put the database on the database device you created, and the transaction log on the transaction log device that you created. Check the "Truncate Log on Checkpoint" checkbox (if that is present).
5. Click "Create Now".
6. Close Enterprise Manager.

### 5.3.2 Creating an empty database on MS-SQL 7.0

1. Run Enterprise Manager. It should automatically register the new SQL server the first time you run it.
2. In the left hand window, find your server by expanding the SQL Server Group. Double-click on your server name.
3. Right click on the Databases folder. Choose “New Database”. Specify a database name (we recommend “nets”) and the size you chose above.
4. Click on the Transaction Log tab. Specify the size for the transaction log that you chose above.
5. Click on the Options tab. Check the “Truncate Log on Checkpoint” checkbox.
6. Click OK. The database device will be created and sized automatically.
7. Close Enterprise Manager.

### 5.3.3 Create the Nets database user

You must now create a database user that Nets will use to log in to the database you created above.

1. Run the ISQL\_w application (on MS-SQL 6.5) or the Query Analyzer (for MS-SQL 7.0). Login as “sa”, password is blank in a new SQL server.
2. Change to the “nets” database, using the DB: menu near the top.
3. Run these commands as SQL queries:

```
sp_addlogin nets,netspw,nets
go
sp_changedbowner nets,true
go
```

This creates a new database user called “nets”, with a password “netspw”. You may wish to choose a different password, but we recommend that you keep the user name nets. The default database for the “nets” user is set to the “nets” database, which is also owned by nets. Record these names and password, you will need them later when you edit Sql . pm.

4. Close ISQL\_w or Query Analyzer.

### 5.3.4 Create an ODBC System Data Source Name (DSN)

Now you must create an ODBC System DSN so that Nets can connect to your SQL database. You must make it a System DSN, not a User DSN.

1. Double-click on “ODBC” in the Control Panel.
2. Choose the “System DSN” tab.
3. Click on “Add...”.
4. Choose “SQL Server (32-bit)” if it is present, else “SQL Server”. If these are not present, then you have not correctly installed the ODBC drivers for MS-SQL.
5. Click on Finish.

6. You will get a new window “Create a New Data Source to SQL Server”. In the Name: field, enter “Nets”. In the Description field, enter “Nets database”. In the Server field, choose or enter the name of the server where your SQL database is running (usually “local”, meaning on the same computer). Click on “Next>”
7. Choose the “With SQL Server authentication using ...” radio button. In Login ID: enter “nets”. In Password, enter “netspw”, or whatever the password you configured for the nets user is. Click on “Next>”
8. Click on “Next>”.
9. Click on “Next>”.
10. Click on “Finish”.
11. Click on “Test Data Source”. This will attempt to connect using ODBC to the Nets database, and log in as the nets user. If it says “TESTS COMPLETED SUCCESSFULLY!” you are ready to move onto the next step. Click on “OK”.
12. Close the ODBC Data Source Administrator.

### 5.3.5 Configure Nets

By now, you should have installed an SQL server, ActiveState Perl, plus the MD5, Tk, DBI and DBD-ODBC perl modules. If so, you can now configure Nets to use the database you created above.

Edit the file Nets\Sql.pm in your Nets distribution directory. This is the file that tells all the Nets programs how to connect to the SQL database. Ensure that it is set up like this:

```
$Nets::config{DBSource}    = 'dbi:ODBC:Nets';  
$Nets::config{DBUsername} = 'nets';  
$Nets::config{DBAuth}    = 'netspw';
```

Where “Nets” is the name of the ODBC System DSN you created above, “nets” is the SQL database login name you created above, and “netspw” is the password for that user.

---

## 6.0 Completing the Nets test installation

---

By this stage, Perl and its supporting modules have been installed, your database server is installed, along with an SQL user for Nets, Nets is configured to use that server, and the SQL tables that Nets needs have been created. Its now time to finish the installation and test Nets.

### 6.1 Create the Nets tables

Create the Nets tables by running

```
perl netscreate.pl
```

This will attempt to log in to the database using the details you entered into Nets/Sql.pm, then create the tables described in Nets/Schema.pm

If the previous step fails, it may be due to errors in your database configuration or setup. If you need to recreate the tables after previously creating them, you should use:

```
perl netscreate.pl -dropfirst
```

This will attempt to drop (delete) the Nets tables before recreating them. **CAUTION:** this will permanently remove any existing data in your Nets database.

### 6.2 Create example Nets data

This step creates a basic database that you can get started with. It includes ADMINISTRATOR and DEFAULT users, basic reports, drawings, and data types, but no Devices, Links, Locations or other site-specific data.

```
perl netsload.pl basicdb.dat
```

If you are intending to just try out Nets at this stage, you can load a sample database, including an example network and data with:

```
perl netsload.pl exampledata.dat
```

You can then run the test suite with:

```
perl test.pl
```

This will exercise and display all the windows in Nets, then remove them all. If the test ends with “ok 1” printed to stdout, then the test suite succeeded.

Later, when you are ready to build your own network, you can revert to a basic starting database which is ready to fill with your specific data, with:

```
perl netsload.pl -clearfirst basicdb.dat
```

### 6.3 Try it out

Nets has now been installed so it can run out of the distribution directory. This will allow you to test and evaluate Nets. You can now start Nets with the command:

```
perl netsmain.pl
```

and explore the features that Nets provides. Login with the user name ADMINISTRATOR, and password ADMIN. Refer to the Nets User Guide for more details.

If you have an unlockable, evaluation or beta version of Nets, the software will stop running some time after your evaluation test period has expired.

If you need to remove this limit to facilitate further testing or evaluation, contact [info@open.com.au](mailto:info@open.com.au) for a new License Key.

## 7.0 Permanent installation

---

By now, you have tested Nets and have decided to install it permanently. In the following installation instructions, the Nets support files will remain in your Nets distribution directory, but the Nets main program, utilities and perl modules will be installed in the usual publicly accessible places for Perl programs.

### 7.1 Unix

1. Purchase a License from Open System Consultants, go to  
`http://www.open.com.au/ordering.html`
2. Optionally edit Site.pm, set the value of NetsDir near the top of Site.pm to the Nets Home Directory as specified in your Makefile.PL (typically /usr/local/nets).
3. (if not using Linux RPM) perl Makefile.PL
4. (if not using Linux RPM) make install
5. cd /usr/local/nets
6. Clear the database and add some comman basic data with  
`netsload.pl -clearfirst basicdb.dat`
7. When you receive your License Key by email, paste it exactly as received into license/license.rc in your Nets distribution

### 7.2 Windows

1. Purchase a License from Open System Consultants, go to  
`http://www.open.com.au/ordering.html`
2. When you receive your License Key by email, paste it exactly as received into license/license.rc in your Nets distribution.
3. Edit Site.pm, set the value of NetsDir near the top of Site.pm to the full path to your Nets distribution.
4. perl Makefile.PL install
5. Clear the database and add some comman basic data with  
`perl netsload.pl -clearfirst basicdb.dat`

Now your Nets programs will be installed in the usual place for perl programs (typically C:\Perl\bin).

## 8.0 Upgrades

---

When upgrading Nets it is imperative that your database be completely backed up in a database independent manner (this is in addition to your regular database backups). This is necessary because new versions of Nets usually contain additional data fields for some existing database objects as well as some

totally new objects. Therefore the database must be recreated with the new object definitions, and the existing Nets data reloaded into the new database.

The best way to proceed with a Nets upgrade is as follows:

1. Ensure that no users are accessing the Nets database.
2. From within the current distribution directory run `./netsdump.pl > dumpfile` (where *dumpfile* is something like `/tmp/database.dat`).
3. Verify the contents of the dumpfile.
4. Change directory to the new Nets distribution.
5. Make any required site-specific changes to the file `"Nets/Sql.pm"` and `"Nets/ Site.pm"`
6. Create the new database with `./netscreate.pl -dropfirst`.
7. Reload the data from step 2 with `./netsload.pl dumpfile`
8. Start the new version of Nets and verify operation with `./netsmain.pl`
9. Distribute and install the new version of Nets to the relevant user machines with "make install" or "perl Makefile.PL install"

Here is an example (on UNIX):

```
cd /usr/local/projects/Nets/Nets-U-2.1
./netsdump.pl > /tmp/database.dat
less /tmp/database.dat
cd /usr/local/projects/Nets/Nets-U-2.2
vi Nets/Sql.pm
vi Nets/ Site.pm
./netscreate.pl -dropfirst
./netsload.pl basicdb.dat
./netsload.pl /tmp/database.dat
./netsmain.pl -u .... -p .....
```

**Caution:** Failure to observe the above procedure may result in data loss.

---

## 9.0 Theory of Operation

---

Nets and its utilities are all written in the Perl programming language ([www.perl.com](http://www.perl.com)). This means that Nets runs on any platform supported by Perl, which in practice means most computer systems, including all Unix platforms, Windows 95/98/NT/2000, Mac OSX, Open VMS and others.

Nets uses the Perl DBI interface to connect to and access an SQL database which contains information about the network Nets is being used to manage. The Perl DBI interface allows Nets to access most free and commercial SQL databases on most platforms.

The SQL database may reside on the same host as where the Nets interface is running, or on a remote host. In any case, multiple instances of Nets can all connect to the same database, allowing multiple Nets users (even in separate departments in an organization, and on a wide range of platforms) to see a single unified description of the network.

Nets uses the Perl Tk module to display its user interface. Tk provides a standard graphical user interface for Nets on Unix and Windows platforms. On Unix the X Window System is used, allowing Nets to be used from any X Windows system. On Windows, the native Windows system is used.

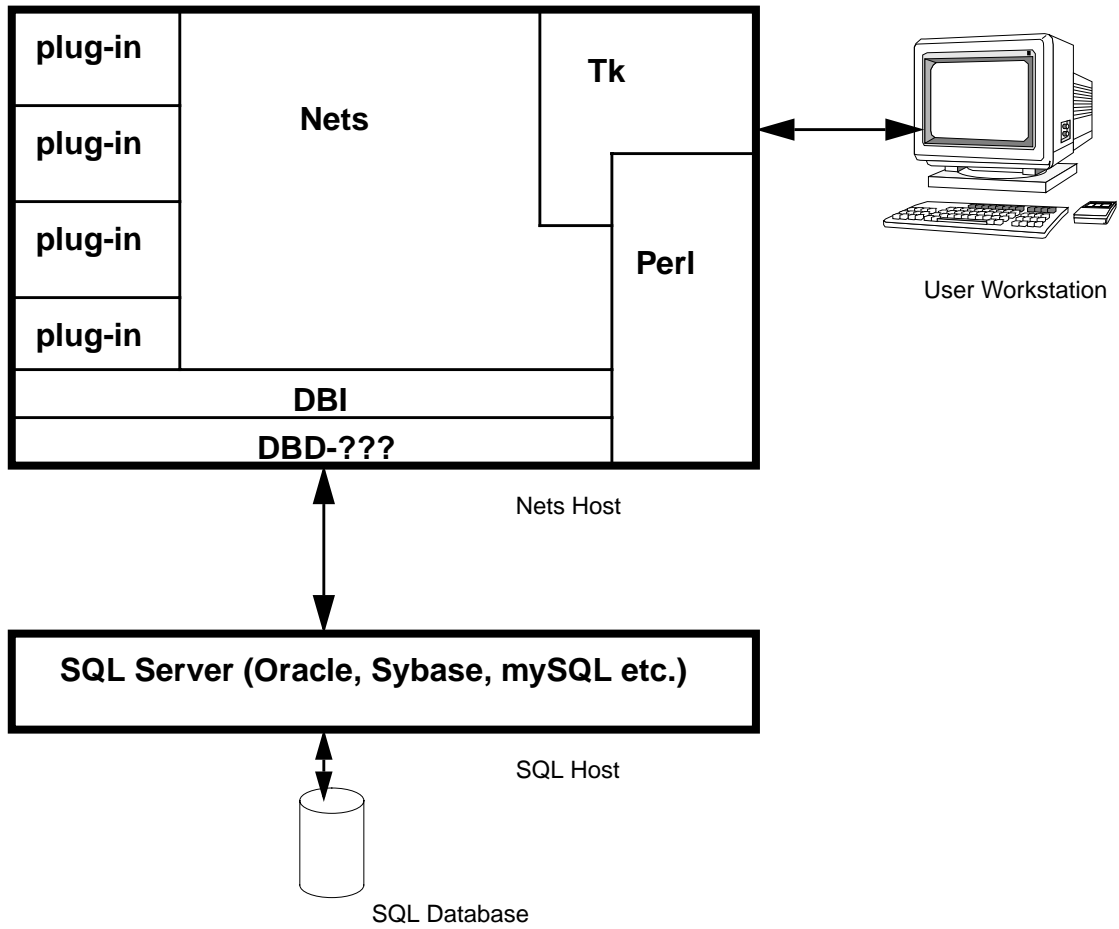
The SQL database tables and indexes are defined by a data structure in Schema.pm. This data structure defines all the tables, columns, datatypes and indexes in a database independent way, allowing Nets to be easily installed and operated on most SQL database systems. Further it allows Nets administrators to easily alter the database schema, enabling any amount of site-specific data to be stored in the database, along with the standard data that Nets needs.

The editor interface for most Nets database objects is also defined by a data structure in UIDesc.pm. This structure defines the editable fields and data types for each SQL table Nets uses. This system allows Nets and its user interface to be very easily extended to include new fields in existing tables, or completely new tables. The resulting user interfaces are completely integrated with the rest of the Nets user interface system. It is also easy to add object-specific tools to the user interface, allowing site-specific actions to be performed on specific types of Nets objects.

In the SQL database, Nets uses a row revisioning system to detect editing collisions between users. Each row in each table contains an OREVISION column which is incremented each time the row is updated. Before each update, the OREVISION is checked to ensure it has not been changed since the data was last fetched from the SQL database.

Nets provides a well-defined system of Plug-Ins for extending its functionality. At start-up, Nets loads any .npm files from the plug-ins directory into the Nets main program. A plug-in becomes a first class component of Nets, and can access any data or code that is part of the Nets system. Plug-ins can therefore be used to extend the database schema or user interface, intercept database activity, or enable any other operation to suit site-specific needs.

FIGURE 1. Nets software structure



---

## 10.0 Nets Environment

---

When the Nets main user interface (netsmain.pl) starts up it configures itself from a variety of sources. The range of configuration sources allows the Nets administrator to tailor the behaviour of Nets to suit the needs of different groups of Nets users.

This section describes where and how Nets gets its configuration information during start-up. They are listed below in the order in which they are used.

### 10.1 Site.pm and Sql.pm

These Nets files are loaded into Nets as soon as the netsmain.pl is read by Perl. They are read from the Perl site-lib directory (or the local directory if present there). They contain default fallback values for most Nets configuration variables, however most can be overridden using the sources described below. You can consult Site.pm and Sql.pm to see exactly what variables are available for tailoring.

### 10.2 Environment variables

Nets uses a number of environment variables to override some of the Nets configuration variables:

- **NETSDIR**  
Overrides NetsDir in Site.pm, which specifies the base directory for many of the files and directories that Nets uses. Can be overridden with -home on the command line.
- **NETS\_USERNAME**  
Sets a default username to log into Nets. Can be overridden by -u on the command line.
- **NETS\_PASSWORD**  
Sets a default password to log into Nets. Can be overridden by -p on the command line. **CAUTION:** on some operating systems, environment variables can be seen by other users. This may allow others to see your Nets password.
- **NETS\_DBSOURCE**  
Overrides DBSource in Sql.pm (but not -dbsource on the command line).
- **NETS\_DBUSERNAME**  
Overrides DBUsername in Sql.pm (but not -dbusername on the command line).
- **NETS\_DBAUTH**  
Overrides DBAuth in Sql.pm (but not -dbauth on the command line).
- **HOME**  
Overrides SaveReportDir in Site.pm, and also specifies the location of the user-specific configuration file \$HOME/.netsrc.

### 10.3 SQL database

After connecting to the database (based on Sql.pm, environment variables and -dbsource, -dbusername and -dbauth on the command line), Nets will then attempt to log the user in, according to NETS\_USERNAME/NETS\_PASSWORD in the environment, -u/-p on the command line and/or the Login dialog. If the user is successfully logged in to Nets, Nets will then consult the user's Preferences from the PREFERENCES table in the database. Any Preferences in the database will override the values from areas described above. The administrator and other suitably privileged Nets users can change the Preferences for a Nets user.

### 10.4 /usr/local/etc/nets.rc

Nets then reads /usr/local/etc/nets.rc, using the contents to override the values of any Nets configuration variables. It looks for lines in the file with the form:

```
name = value
```

and the Nets configuration variable '*name*' will be set to '*value*'. For example:

```
NetsDir = /usr/local/nets/etc
```

will set NetsDir (the base directory for the files and directories that Nets uses) for all Nets users on that host to /usr/local/nets/etc.

Lines beginning with a hash ('#') are ignored.

### 10.5 \$HOME/.netsrc

Nets then reads this user-specific configuration file, looking for lines in the same format as /usr/local/etc/nets.rc (see Section 10.4 on page 16).

### 10.6 License file

Nets will now read the licensing file specified in the 'License' configuration variable (typically <NetsDir>/license/license.rc) looking for lines in the same format as /usr/local/etc/nets.rc (see Section 10.4 on page 16).

This file is typically used for License Keys, but you can actually set any Nets configuration variables in it. As delivered, license/license.rc is basically empty, which means that unlockable versions of Nets will stop working after your evaluation period has expired.

### 10.7 Other RC files

If a Nets RC file is specified with the -c command line flag, it is read now, looking for lines in the same format as /usr/local/etc/nets.rc (see Section 10.4 on page 16).

### 10.8 Command line arguments

Finally, any command line arguments of the form name=value are used to override the values of any Nets configuration variables. For example:

```
netsmain.pl NetsDir=/usr/local/nets/etc
```

will set NetsDir (the base directory for the files and directories that Nets uses) for that session to /usr/local/nets/etc.

---

## 11.0 Running Nets programs

---

### 11.1 netmain.pl

When Nets has been permanently installed, you can run the main Nets user interface with just

netsmain.pl

On Windows, you can also run netsmain.pl by just double-clicking it in Windows Explorer. You can also create a Shortcut to netsmain.pl and double-click on that.

Netsmain.pl understands a range of command line arguments:

```
netsmain.pl [-h] [-d] [-u username] [-p password]
             [-dbsource dbi:xxx:yyy] [-dbusername name] [-dbauth auth]
             [-c rcfile] [-home dir]
             [configname=value] ... [scriptfilename] ...
```

- -h  
Prints some command line help to stdout and exits.
- -d  
Turns debugging on. Messages are printed to stdout. This is equivalent to setting the Nets LogLevel to 4.
- -u username  
Attempts to log in to Nets with the given username. If the username/password combination authenticates, the user will be logged in without presenting a login dialog.
- -p password  
Attempts to log in to Nets with the given password.
- -dbsource dbi:xxx:yyy  
Overrides DBSource in Sql.pm and the NETS\_DBSOURCE environment variable. Specifies the database source name that Nets will use.
- -dbusername name  
Overrides DBUsername in Sql.pm and the NETS\_DBUSERNAME environment variable. Specifies the SQL user name that Nets will use to log in to the database.
- -dbauth auth  
Overrides DBAuth in Sql.pm and the NETS\_DBAUTH environment variable. Specifies the SQL user password that Nets will use to log in to the database.
- -c rcfile  
Reads the named Nets RC file, interpreting lines of the form name=value and using them to set Nets configuration variables.
- -home dir  
Specifies an alternate directory to override the value for NetsDir. Defaults to /usr/local/nets if it exists, else the current directory.
- configname=value  
Overrides configuration variable settings in Site.pm. You can specify an number of variable settings.
- scriptfilename  
Starts the Nets Scripting Language script running. You can specify any number of script file names.

## 11.2 netscreate.pl

Netscreate.pl creates SQL database tables and indexes according to the Nets database description in Schema.pm.

```
netscreate.pl [-h] [-d] [-n]
  [-dbsource dbi:xxx:yyy] [-dbusername name] [-dbauth auth]
  [-dropfirst]
```

- -h  
Prints some command line help to stdout and exits.
- -d  
Turns debugging on. Messages are printed to stdout.
- -n  
Does not actually drop or create tables.
- -dbsource dbi:xxx:yyy  
Overrides DBSource in Sql.pm and the NETS\_DBSOURCE environment variable. Specifies the database source name that Nets will use.
- -dbusername name  
Overrides DBUsername in Sql.pm and the NETS\_DBUSERNAME environment variable. Specifies the SQL user name that Nets will use to log in to the database.
- -dbauth auth  
Overrides DBAuth in Sql.pm and the NETS\_DBAUTH environment variable. Specifies the SQL user password that Nets will use to log in to the database.
- -dropfirst  
Causes netscreate.pl to try to drop each table before it is created. Use this option to replace any existing tables. CAUTION: use of this option will cause permanent loss of data from existing Nets tables.

## 11.3 netsload.pl

Netsload.pl loads SQL data from a flat file into the Nets database tables. The tables and their columns are interpreted according to the Nets database description in Schema.pm. Attempts to load data into tables that are not defined by Schema.pm will be silently ignored. Attempts to load data into columns that are not defined by Schema.pm will be silently ignored. The flat file data format is described in the Nets Developers Guide.

If no files are specified on the command line, stdin will be used. Any '\n' sequences in the data will be translated to a newline character.

Data for 'datetime' type columns can use any of the special datetime strings (such as 'now' for the current time).

Data for ID and foreign key tables can use macros like lastId and idForName as described in the Nets Developers Guide.

```
netsload.pl [-h] [-d] [-n]
  [-dbsource dbi:xxx:yyy] [-dbusername name] [-dbauth auth]
  [-dropfirst] [filename] ...
```

- `-h`  
Prints some command line help to stdout and exits.
- `-d`  
Turns debugging on. Messages are printed to stdout.
- `-n`  
Does not actually clear or insert into tables.
- `-dbsource dbi:xxx:yyy`  
Overrides DBSource in `Sql.pm` and the `NETS_DBSOURCE` environment variable. Specifies the database source name that Nets will use.
- `-dbusername name`  
Overrides DBUsername in `Sql.pm` and the `NETS_DBUSERNAME` environment variable. Specifies the SQL user name that Nets will use to log in to the database.
- `-dbauth auth`  
Overrides DBAuth in `Sql.pm` and the `NETS_DBAUTH` environment variable. Specifies the SQL user password that Nets will use to log in to the database.
- `-clearfirst`  
Causes `netsload.pl` to try to empty all tables before the data is loaded. CAUTION: use of this option will cause permanent loss of data from existing Nets tables.

### 11.4 netsdump.pl

`Netsdump.pl` dumps SQL data from the Nets database into a flat file format to stdout, suitable for reloading by `netsload.pl`. If no specific tables or rows are specified on the command line, every row and every column from every Nets table specified in `Schema.pm` will be dumped. If any specific tables are specified on the command line, only those tables will be dumped. The flat file data format is described in the Nets Developers Guide.

Any newline characters in the data will be translated to `'\n'`.

```
netsdump.pl [-h] [-d] [-n]
             [-dbsource dbi:xxx:yyy] [-dbusername name] [-dbauth auth]
             [[tablename][:where][:colname:subst][:colname:subst]] ...
```

- `-h`  
Prints some command line help to stdout and exits.
- `-d`  
Turns debugging on. Messages are printed to stdout.
- `-n`  
Does not actually clear or insert into tables.
- `-dbsource dbi:xxx:yyy`  
Overrides DBSource in `Sql.pm` and the `NETS_DBSOURCE` environment variable. Specifies the database source name that Nets will use.
- `-dbusername name`

Overrides DBUsername in Sql.pm and the NETS\_DBUSERNAME environment variable. Specifies the SQL user name that Nets will use to log in to the database.

- `-dbauth auth`

Overrides DBAuth in Sql.pm and the NETS\_DBAUTH environment variable. Specifies the SQL user password that Nets will use to log in to the database.

- `-tablename[ ;where][ ;colname ;subst ][ ;colname ;subst ] . . .`

Specifies exactly which table(s) and rows to dump. If none of these are specified, all columns from all rows in all Nets tables will be dumped.

*tablename* is the name of the Nets table.

*where*, if specified, is an SQL where clause that will be used to select rows from the table.

*colname;subst*, if specified, causes netsdump.pl to transform the data from the given column using the Perl regexp substitution given. This technique can also be used to call internal Nets functions and therefore perform very complicated transforms on the data. See the examples below.

**Example:** In this example, some of the rows from the DEVICE table are dumped, and the LOCATION column is transformed from a foreign key ID to the corresponding name.

```
netsdump.pl "DEVICE;RACK=2;LOCATION;s/(.*)/Nets::DBUtil::name-  
ForId('$1','LOCATION')/e"
```

## 11.5 netspkginstall.pl

Netspkginstall.pl validates and installs a Nets Package from a package directory into Nets. The package directory is expected to contain a 'DESCRIPTION' file that describes the contents of the Package. See the Nets Developers Guide for more details about Nets packages.

```
netspkginstall.pl [-h] [-d] [-n]  
[-dbsource dbi:xxx:yyy] [-dbusername name] [-dbauth auth]  
[-dropfirst] [directory] . . .
```

- `-h`

Prints some command line help to stdout and exits.

- `-d`

Turns debugging on. Messages are printed to stdout.

- `-n`

Validate but do not actually install.

- `-dbsource dbi:xxx:yyy`

Overrides DBSource in Sql.pm and the NETS\_DBSOURCE environment variable. Specifies the database source name that Nets will use.

- `-dbusername name`

Overrides DBUsername in Sql.pm and the NETS\_DBUSERNAME environment variable. Specifies the SQL user name that Nets will use to log in to the database.

- `-dbauth auth`

Overrides DBAuth in Sql.pm and the NETS\_DBAUTH environment variable. Specifies the SQL user password that Nets will use to log in to the database.

- **directory**

The directory that contains the package to be installed. Any number of directories may be specified, and a Package will be installed from each one.

---

## **12.0 Nets Directories and Files**

---

The standard Nets distribution includes a number of files and directories, and this section describes the major directories included and how they are used.

### **12.1 Main distribution directory**

When a Nets distribution file is unpacked (using tar/gzip or WinZip, depending on your platform), it creates a single directory containing the entire Nets software tree. The top of this distribution tree will be named something like `Nets-2.1`, depending on exactly which version of Nets you have. This top level directory contains the Nets executable programs, some data files, and a number of important sub-directories described below.

The executable programs include `netsmain.pl` (the main Nets user program), plus some utility programs, `netscreate.pl`, `netsload.pl`, `netsdump.pl` and `netspkinstall.pl`. When Nets is installed, these executable programs are copied to the standard Perl binary directory (typically `/usr/local/bin` on Unix, or `C:\Perl\bin` on Windows).

In the following sections, `<NetsDir>` means the directory at the top of your Nets distribution.

### **12.2 `<NetsDir>/Nets`**

This directory includes all the Nets Perl modules required by the Nets programs mentioned above. Each module has a `.pm` extension. When Nets is installed, these modules are copied to the standard Perl site library directory (typically `/usr/local/lib/perl5/site_perl` on Unix, or `C:\Perl\site` on Windows). You can add your own modules to Nets by adding them to this directory and reinstalling Nets.

### **12.3 `<NetsDir>/images/icons`**

Contains the standard Nets icons, which are using on Nets Drawings and other parts of the Nets user interface.

### **12.4 `<NetsDir>/images/drawings`**

Contains the background images for Nets Drawings.

### **12.5 `<NetsDir>/templates`**

Contains the standard templates, used for creating collections of interrelated Nets Objects.

**12.6 <NetsDir>/plugins**

Contains the standard Plug-ins provided as part of Nets.

**12.7 <NetsDir>/scripts**

Contains the standard Nets Scripting Language scripts provided with Nets.

**12.8 <NetsDir>/goodies**

A collection of contributed and unsupported files that developers may be interested in. See the README file for a brief description of each file.

---

**13.0 Configuring Nets**

---

This section describes how to change basic Nets configuration options.

**13.1 Nets/Sql.pm**

This file contains the information that Nets uses to connect to and log in to its SQL database. On Unix, Sql.pm is set up automatically, depending on which database install script you ran. In Windows, you will need to edit it yourself. In any case, you may need to change Sql.pm whenever your SQL database host is changed or moved.

Sql.pm defines 3 variables:

**13.1.1 DBSource**

This is the DBI data source name that determines what sort of SQL database is to be used, and how to connect to it. The general form is of a data source name is:

```
dbi:databasetype:connectiondetails
```

For example:

```
$Nets::config{DBSource} = 'dbi:mysql:nets';
```

will connect to a MySQL database called 'nets' on the local host.

The exact form to use in the data source name depends on your SQL database, and is determined by the Perl DBD module for your database. If in doubt, consult the documentation for your DBD module.

**13.1.2 DBUsername**

For most database types, this specifies the SQL user name that will be used to log in to the SQL database.

**13.1.3 DBAuth**

For most database types, this specifies the SQL password that will be used to log in to the SQL database.

### 13.2 Nets/Site.pm

This file contains the initial values of various configuration variables that can be set within Nets. The Nets configuration in Site.pm can be overridden by a range of per-user and per-host sources, such as configuration files, command-line arguments, and user preferences in the database. Changing the configuration variables allows you configure the default behaviour of Nets. Changing them in Site.pm makes those changes apply to all users who run Nets on the hosts where your Site.pm is installed, but still allows them to be overridden on a per-user or per-site basis.

Site.pm is amongst the last of the Nets support modules loaded when Nets starts up. It can therefore be used to override most functions and data within Nets.

You can also add site-specific code to Site.pm, and use that code to establish Hooks, change the user interface definitions or schema or change or override any code that is part of Nets. However, unless special conditions exist, we recommend that site-specific code be put in a Nets Plug-in. See the Nets Developers Guide for more information.

All of the configuration variables in Site.pm can be overridden by Nets RC files, command line arguments and user Preferences in the database. See below for more details.

#### 13.2.1 NetsDir

One of the most important configuration variables in Site.pm is NetsDir. Nets uses the NetsDir configuration variable to find the images, templates, plug-ins etc. that it needs. The default value set in Site.pm is '.', which means that Nets will look for images, templates and plug-ins directories in the *current* directory. While this is suitable for initial testing of Nets, you will probably want to set up a more permanent (and publicly accessible) location for those support files and directories. If you create a permanent NetsDir you should also change NetsDir in Site.pm to point to the new location.

The NETSDIR environment variable overrides NetsDir in Site.pm.

---

## 14.0 Administration

---

The Nets Administrator is responsible for installing Nets and its database and the initial Nets configuration as described in the previous sections of this Guide.

The Nets Administrator is also responsible for administering elements of Nets that other Nets users are not able to do for themselves. This section describes some administration activities the Nets Administrator is responsible for.

### 14.1 Users

Nets provides a system of user names and logins, allowing various levels and ways of controlling access to Nets and its data.

Nets is delivered with 2 standard Nets users:

- ADMINISTRATOR

By default, this user has access to all parts of the Nets system and can change anything. The default password is ADMIN (this should be changed as soon as possible after installation).

- **DEFAULT**

This is the default user name for any user who does not have their own Nets user-name. It will be used if there is not a specific Nets user for the logged in user name. The default password is DEFAULT.

When the Nets main program (netsmain.pl) starts up, it logs in to the Nets system using the following logic:

1. On Unix, the user name defaults to their Unix username. On Windows, it defaults to 'Windows User'.
2. If NETS\_USERNAME and/or NETS\_PASSWORD environment variables are set, they are used.
3. If -u and/or -p arguments are specified on the command line, they will be used as the user name and password.
4. If there is both a user name and password resulting from the above steps, Nets will attempt to log them in silently (i.e. without presenting a Login dialog). If the named user (or DEFAULT) exists and the password is correct, the user will be logged in to Nets, else the next step is followed.
5. The Login dialog is presented, allowing the user to enter their Nets username and password.
6. If the entered user name (or DEFAULT) exists and the password is correct, the user will be logged in to Nets.
7. After logging in, the user's details will be added to the Nets CURRENT\_USERS table for licensing purposes.

As can be seen from the above, Nets will try to log users in with a user name, and will fall back to the DEFAULT user name (if it exists). This means that you can have specific user names and privileges for some users, and a generic login for other users.

As the Nets administrator, you will have to consider the most appropriate user policy for your site. The best and most secure policy is to have a separate login for each Nets user, and *no* DEFAULT user. If you do elect to have a DEFAULT user, you should make sure that they have no permissions to change system data, and you should also change the password for the DEFAULT user regularly.

A common setup would be to have specific user names for your Network Operations Centre staff, who have permission to change the description of your network, and a DEFAULT user, for use by all other staff who can see the network details, but cannot change any data.

In any case, the ADMINISTRATOR user should always be present, and should always have all permissions. Do not reveal the ADMINISTRATOR password to anyone else.

Note: you should not confuse the Nets user name with the username and password that Nets uses to log in to the SQL database.

## 14.2 Permissions

Each registered Nets user has a set of permissions that apply to them. The permissions are loaded from the Nets database when netsmain.pl starts up, and they limit what actions the user can undertake for the rest of the session.

Nets permissions are administered with the User editing form under Edit->Nets User. You must have EDIT\_ANYTHING or EDIT\_USER permission to edit User details (as delivered, ADMINISTRATOR has that permission).

Some of the more important permissions are:

- **Log in to Nets**  
Permits the user to log in to Nets. You might disable this to temporarily remove permission for a user to use Nets, without deleting their user details.
- **Install Packages**  
User can use File->Install a Nets Package to install a package.
- **Edit all objects**  
User can edit, change and delete any and all Nets data objects.
- **View all objects**  
User can view the details of any and all Nets data objects.
- **Edit Basic**  
User can edit, change and delete basic Nets data, such as ADDRESSTYPE, CONNECTORTYPE etc. It does not include network description data such as DEVICE, LINK, INTERFACE, CONNECTOR, RACK, ADDRESS, BUNDLE, ENTITY, LOCATION, RACK, SOFTWARE.
- **View Basic**  
User can view basic Nets data.
- **Edit Network**  
User can edit, change and delete network data such as DEVICE, LINK, INTERFACE, CONNECTOR, RACK, ADDRESS, BUNDLE, ENTITY, LOCATION, RACK, SOFTWARE.
- **View network**  
User can view network data.
- **Edit User**  
User can edit, change and delete User details.
- **View User**  
User can view User details. CAUTION: this includes the users password.
- **Run Scripts**  
User can run any Nets Scripting Language script.
- **Import**  
User can import data into the Nets database from a flat file using Edit->Import...
- **Export**

User can export data from the Nets database to a flat file.

- Save Window Layout

User can save their current window layout to their user profile.

- Restore Window Layout

User can restore their default window layout from their user profile.

- Use templates

User can use the Edit->Create from Template menu to create a group of related objects using a Nets template.

- Run Reports

User can run any and all reports.

- Edit Drawing

User can edit, change and delete any Drawing.

- View Drawings

User can view any Drawing.

### **14.3 Preferences**

Nets users have a set of Preferences that apply to them. The Preferences are loaded from the Nets database when the user is logged in to Nets. Preferences set new user-specific values for Nets configuration variables, and override the default ones in Site.pm.

Nets preferences are administered with the User editing form under Edit->Nets User. You must have EDIT\_ANYTHING or EDIT\_USER permission to edit User details (as delivered, ADMINISTRATOR has that permission).

### **14.4 Audit Trail**

Nets maintains a complete Audit Trail of all user activities. Whenever any user logs in, adds, modifies or deletes an object, an entry is added to the Nets Audit Trail. The Nets Administrator may be required from time to time to investigate changes to data in the system, and who made the changes.

The Audit Trail for a particular object can be accessed from the Editor for that object. The entire contents of the Audit Trail can be searched and displayed by using View->Search for Object window.

### **14.5 NetsDir**

One of the most important Nets configuration variables is NetsDir. NetsDir specifies the base directory where Nets looks for the files and directories it needs to operate. The default value for netsDir is set in Site.pm (but you can override it using any of the methods described in Section 10.0 on page 14). As delivered, NetsDir is set to '.' (i.e. the current directory) which allows Nets to be tested and evaluated entirely within the distribution directory.

When you install Nets permanently, you will need to choose a publicly accessible location for NetsDir, and make sure NetsDir is set in Site.pm (or some other global configuration file, such as /usr/local/etc/nets.rc). See Section 7.0 on page 11 for more details.

## 14.6 Icons

Nets uses a number of icons, both as part of its standard user interface, and also to represent various types objects on Nets drawings.

Whenever Nets needs an icon, it looks in the directory given by the IconPath configuration variable. IconPath defaults to '%N/images/icons' (i.e. to images/icons in the NetsDir directory).

You can change the location of IconPath using any of the configuration methods described in Section 10.0 on page 14. Also you can add new icons to that directory at any time, and they will become available immediately to Drawing editors.

Images in the IconPath directory must be in GIF, XBM or XPM formats. Other image formats can be supported by installing additional Tk modules from CPAN.

## 14.7 Packages

Nets provides methods for users to easily install bundles of software into Nets. Such Packages can be used to extend and enhance the standard features provided by Nets. Nets packages will be made available by us and also by third parties.

It is usual to arrange permissions so that only ADMINISTRATOR can install packages. In that case ADMINISTRATOR would be required to install any new packages, using either File->Instal a Nets Package... or with the `netspkginstall.pl` utility (see Section 11.5 on page 20).

## 14.8 Access to Multiple databases

In some circumstances, you may wish to operate multiple independent Nets databases. For example, you may have one main database that contains your current network description, and another one for conducting training of new staff, and perhaps another one for the development of changes or plug-in software.

You can control which database any Nets utility will connect to in a number of ways, as described above. In summary, they are:

- DBSource, DBUsername, DBAuth in Sql.pm.
- NETS\_DBSOURCE, NETS\_USERNAME, NETS\_DBAUTH environment variables.
- -dbsource, -dbusername, -dbauth command line arguments.

The most common arrangement is to put the connection details for the database that most people will use into Sql.pm. Then tell users who need to connect to different databases to use the command line arguments:

```
netsmain.pl -dbsource dbi:Oracle:othersid -dbusername \  
fred -dbauth xxx
```

## 14.9 Licenses

Nets is generally licensed to end users based on the number of concurrent users. Concurrent users means the number of Nets users all logged in and using Nets at the same time. This policy is enforced in unlockable versions of Nets. Unlimited versions of Nets have no licensing limitations built in, and this section may be ignored.

Nets keeps track of the users who are currently logged in to Nets. Whenever a Nets user logs into Nets, Nets adds an entry to the CURRENT\_NETSUSER table. When the user logs out by closing the main window, or selecting File->Close, the entry is removed from the table. While the user is logged on, the entry is refreshed every 5 minutes.

If a Nets host computer crashes while a Nets user is logged in (or if the Nets program is cancelled with Control-C), the CURRENT\_NETSUSER table will still hold an entry for the now non-existent user. This stale entry will be expired automatically after 10 minutes, but in the meantime, the user may not be able to log in if all other license are in use.

Inquisitive engineers should note that the unlockable versions of Nets includes sophisticated tests to prevent license tampering by manipulating the CURRENT\_NETSUSER table.

### 14.9.1 License management

As delivered, unlockable, evaluation and beta versions of Nets have runtime limits on the number of concurrent users, and an expiry date. In order to remove these limitations from your Nets system, you must purchase a license from Open System Consultants Pty Ltd ([www.open.com.au/nets](http://www.open.com.au/nets)). When you purchase a license, we will send you by email a key allowing you to unlock the software according to the type of license you have purchased.

Licenses may be upgraded at any time, and a new key will then be issued by email.

### 14.9.2 Key installation

A Nets key consists of a number of lines of text that must be placed in a Nets RC file. The key will be sent to you when you purchase your license. The key must be entered into the configuration file *exactly* as sent to you. we recommend that you cut-and-paste it, rather than re-typing it.

The key may be added to any of the Nets RC files supported by Nets:

- /usr/local/etc/nets.rc
- \$HOME/.netsrc
- <NetsDir>/license/license.rc
- any file specified with -c on the netsmain.pl command line.

The normal and preferred place is `<NetsDir>/license/license.rc`. In any case, keep a record of where you put it, as you may want to change it later if you upgrade your license.

### 14.9.3 Upgrading keys

If you purchase an upgraded key (perhaps with a larger concurrent users limit), a new key will be sent to you by email. In order to start using the key, you must replace the old key in the configuration file.

Unlimited versions of Nets will ignore any keys found in a configuration file. Therefore if you upgrade from an unlockable to an unlimited version, you do not have to remove the previous key, but you will have to reinstall the Nets software (or at least `nets-main.pl`).

### 14.10 Debugging

Nets includes facilities for Nets developers and database administrators to investigate any problems that might occur.

You can run any Nets utility with the `-d` flag, which turns on debugging level messages. You can also specify the `LogLevel` configuration variable for `netsmain.pl`.

You can also pop up the Message window with `View->Message Log...` which will show all new messages at or above the selected logging level from that time on.

At Debug level (`LogLevel 4`) you will see the details of every SQL query that Nets issues to the database. This can be very helpful for tracking down many types of problems.

If you intend to report a problem, or request assistance from the Nets mailing list you should include as much of the message log at Debug level as possible.

---

## 15.0 Getting Help

---

If you need help with Nets installation or administration tasks, you should consider using the Nets mailing list. If you have a support contract you can also send email to the Nets support email address.

However, before posting to the list requesting information or help, try the following:

- Read and reread the manuals and guides, especially this Installation and Administration Guide.
- Examine the Nets message log at Debug level, or use the `-d` flag to any Nets utility. Use this information to try to figure out what is going on.
- Examine the Nets source code, looking for clues and documentation.
- If you have a question or problem with Perl coding or syntax, you probably won't get much help from the Nets mailing list. Try a Perl book or an appropriate Perl mailing list.

If you do post to the mailing list, include a very detailed description of the problem or question, along with as much of the message log at debug level (LogLevel 4) as is relevant. The more information you can provide the faster and better the response will be.

If you solve your problem, consider posting a summary and the solution, so we can add it to the Nets FAQ.

If you find a bug, please let us know so we can fix it. The best way to let us know is to post to the Nets mailing list.

To subscribe to the Nets mailing list, send email with the single word `subscribe` in the body (not in the subject line) to:

`nets-request@open.com.au`