Getting Started with SYBASE

NeXT Technical Support Notes

Written by Kate Smith
Edited by Adrienne Wong and Caroline Rose

Contents

Getting Started with SYBASE

To use this guide, you need to be familiar with the NeXT interface, and need to know how to log into the system as **root**. You will also be using UNIX® commands (they're explicitly written out in this guide), which will be typed in a Shell or Terminal window.

The following is a list of files and directories that contain useful information that you should read.

/usr/sybase
SYBASE's home directory. It contains all of the SYBASE®-specific information except for the documentation.

/usr/sybase/ReleaseNote4.0.nr
SYBASE-specific release note for the 4.0 release on all platforms. You'll want to read this if you're familiar with earlier versions of SYBASE—it explains the differences between the current and previous releases. See the section below, "Printing the Documentation," before you print or read it.

/usr/sybase/sample/dblibrary/README
Description of the examples and how to compile and run them.

/NextLibrary/Documentation/NeXT/RelNotes/SybaseNotes.wn
The closest thing to an installation guide. The installation notes are in the "Getting Started" section on page 6 of **SybaseNotes.wn**—they're not in the "Operation" section as stated on page 2. Unfortunately, these notes are buggy. The information below will clarify the operations.

/NextLibrary/Documentation/Sybase
Directory that contains all of the documentation that's normally printed. We recommend adding this directory as a target to the Digital Librarian™ application for reference. (See *The NeXT User's Reference Manual* for more information about adding targets to the Digital Librarian.) An index has already been built for this directory, so there is no need to update it.

/NextLibrary/Documentation/Sybase/README.wn
A file that describes how you print and read SYBASE's documentation. It contains some errors. See the section below, "Printing the Documentation."

/NextLibrary/Documentation/Sybase/SAG
The System Administrator's Guide. If you're installing SYBASE, you're probably going to be the system administrator. You should read at least the first two chapters of this guide. See the section below, "Printing the Documentation," before you print or read it.

Printing the Documentation

Most of the documentation files are in nroff format. This means that the files can't be read until they're formatted. The **README.wn** file describes how to format the files, but there are two problems: None of the pictures exist, so the formatter dies when it finds a reference to a picture within the text; and tables aren't formatted properly because there's a step missing in the process.

To fix the first problem:

1. Log in as **root**.

2. Edit the following files using the Edit application or some other editor. Remove all lines starting with ".Sc".

   /NextLibrary/Documentation/Sybase/COMREF/comref.c
   /NextLibrary/Documentation/Sybase/COMREF/comref.t
   /NextLibrary/Documentation/Sybase/SAG/sag2
   /NextLibrary/Documentation/Sybase/SAG/sag3
   /NextLibrary/Documentation/Sybase/SAG/sag7
   /NextLibrary/Documentation/Sybase/SQLUG/sql01
   /NextLibrary/Documentation/Sybase/SQLUG/sql05
   /NextLibrary/Documentation/Sybase/SQLUG/sql13
   /NextLibrary/Documentation/Sybase/SQLUG/sql14

If you don't edit the above files, you'll only get the first part of each of these files when you format them.

To fix the second problem, in the **README.wn** file, change the line for viewing,

```
ptroff –t –msyb –rT1 > /tmp/tempfile.eps
```

to

```
tbl filename | ptroff –t –msyb > /tmp/tempfile.eps
```

If you want to scan a file in a Shell window, the line should read:

```
tbl filename | nroff –t –msyb
```

If you want to print a file, the line should read:

```
tbl filename | ptroff –msyb
```

There is no table of contents in this documentation. The next release of the SYBASE software will contain the fixes mentioned here, a table of contents, and a script for printing all or part of the documentation.

Setting the SYBASE Login (as root)

The user **sybase** on the system has a password of *****, which means that no one can log in as **sybase** until the password is changed, the files within the **/usr/sybase** directory are not writable by the **sybase** user, and the library that contains the **dblib** routines needs to be **ranlib**'d so that you can link it into a program. You'll also need to create a directory for your databases—many of the scripts depend on this directory. Here's how to fix all the above problems:

1. Log in as **root**.

2. Start up the Shell application, located in **/NextApps**.

3. In a Shell window, type the following commands (what you type is shown in bold):

```
# mkdirs /usr/sybase/database
# chown -R sybase /usr/sybase
# ranlib -t /usr/sybase/lib/libsybdb.a
```

4. Start up the UserManager application, located in **/NextAdmin**.

5. Choose Set Domain from UserManager's main menu. This brings up the Set Domain panel. Select the correct domain.

6. Choose the View command from the main menu, and choose Long from the View menu. This changes the style of the data in the UserManager window.

7. Select **sybase** from the list of user names on the right in the UserManager window. Type a new password in the password field and click Modify. This changes the password for **sybase** and enables you to log in as **sybase**. Remember your password!

See the NeXT *Network and System Administration* manual for specific details about using the UserManager application.

Creating the Master Database (as sybase)

The following instructions are based on the notes in the file

/NextLibrary/Documentation/NeXT/RelNotes/SybaseNotes.wn

starting on page 6.

Note: Page 1 of the above file lists the file

/NextLibrary/Documentation/Sybase/ReleaseNote4.0.nr

but it doesn't exist; it's really

/usr/sybase/ReleaseNote4.0.nr

Note: The scripts **showserver** and **stopserver** are in **/usr/sybase/Install**.

Now you're ready to start creating databases.

1. Log in as **sybase**.

If you're going to have more than one server on your network, you need to modify **/usr/sybase/interfaces** as well as **/usr/sybase/scripts/SetVars**. If you're only going to have one server, you can skip steps 2 and 3 below. However, it won't hurt to make the changes.

2. (Optional) Edit **/usr/sybase/interfaces** on the machine that will be the database server machine for the network. See the Appendix for a sample **/usr/sybase/interfaces** file.

a. Copy the block of text

```
localhost
    master tcp next-ether localhost 3696
    query tcp next-ether localhost 3696
    console tcp next-ether localhost 4696
    debug tcp next-ether localhost 5696
    trace tcp next-ether localhost 6696
```

and paste it at the end of the file.

b. In the title, change the name of the server from

```
localhost
```

to your server name.

c. Change the name of the host to the server's host machine on each line in that section (from localhost to your host). For example,

```
myServer
    master tcp next-ether myHost 3696
    query tcp next-ether myHost 3696
    console tcp next-ether myHost 4696
    debug tcp next-ether myHost 5696
    trace tcp next-ether myHost 6696
```

d. Repeat steps **a** through **c** for each server.

After you work on the first server you'll need to change the port numbers of the other servers as well. Your file should look something like the one in the Appendix. Note that the server name (port number) can be any name (number) as long as they are consistent among all machines on the network. This means that all machines need to have identical **interfaces** files. Once you've started your server and copied **interfaces** and **SetVars** to another machine, you can log into that machine and run **isql** to talk with your server. For example, type the following in a Shell window:

```
isql -Usa -P -SmyServer
```

3. (Optional) Edit **/usr/sybase/scripts/SetVars**.

a. Modify DBLISTEN and DBQUERY to the name of the server you want to use (change SYBASE to your server's name).

You can also change the typos in the message by changing the quotes to the following **/usr/sybase/scripts/SetVars** example:

```
#
echo "If you didn't type 'source SetVars' then the variable
settings won't take effect."
setenv SYBASE /usr/sybase
setenv DSQUERY myServer
setenv DSLISTEN myServer
setenv PATH $PATH\:.:/usr/sybase/bin
```

4. Edit the script **/usr/sybase/Install/startserver** so that it doesn't open up a Terminal window.

a. Remove "Terminal -Shell" and the double quotes in the command from the two lines where they exist. This solves the problem of not being able to run the script from a remote machine.

Line 102,

```
Terminal -Shell "/usr/sybase/bin/dataserver -s$SERVER_NAME -
d$MASTER_DEVICE -e$ERROR_LOG" &
```

should be changed to the following (note that there isn't a Return):

```
/usr/sybase/bin/dataserver -s$SERVER_NAME -d$MASTER_DEVICE -
e$ERROR_LOG &
```

Line 121,

```
Terminal -Shell "/usr/sybase/Install/installserver $SERVER_NAME
```

```
$SYBASE/interfaces"
```

should be changed to:

```
/usr/sybase/Install/installserver $SERVER_NAME
$SYBASE/interfaces
```

5. Start up the Shell application.

6. Execute the following commands in a Shell window (what you type is shown in bold). *serverName* is either SYBASE or your server name if you created one. The **isql** commands initialize the database properly for the examples.

```
% source /usr/sybase/scripts/SetVars
If you didn't type 'source SetVars' then the variable settings
won't take effect.
% /usr/sybase/Install/startserver -s serverName
Type startserver help for help.
startserver: Master database device
/usr/sybase/database/master.sdb
            does not seem to be present/writable.
Do you want me to create this one for you (2Mb free space)?
(y/n) y
Starting buildmaster...
Buildmaster/4.0/P/NeXT/0.981/6/ of Thu Aug-31-89 13:07:53
Master device: /usr/sybase/database/master.sdb
writing configuration area
writing the MASTER database
<a lot more messages; this takes quite a while>
Installation complete...
SQL Server is available and ready for use.
% isql -Usa -P < $SYBASE/scripts/installmaster2
<a lot of messages about the rows that are modified>
(return status = 0)
% isql -Usa -P < $SYBASE/scripts/installpubs
<several messages about what has been modified>
(return status = 0)
%
```

SetVars sets the environment variables that tell the scripts which server you're using. **startserver** will ask if you want to create the master database the first time you run it. This is exactly what you want, so type **y**.

SybaseNotes.wn suggests that you use the **CreateMaster** script instead of using **startserver** and the **isql** commands, and either method is OK. Using **startserver** allows you to have control over the server name, but using **CreateMaster** gives you control over the size of the master database. Note that **CreateMaster** installs **master2** and **pubs**, so the last two **isql** steps above are unnecessary.

SybaseNotes.wn tells you to kill the server. This isn't necessary. When the installation is done, SYBASE is already running.

Note that the SYBASE server will be running on your machine until you stop it (by using **shutdown** from isql) or power down. You can check to see if it's running with the **/usr/sybase/Install/showserver** command. Use **startserver** with the **-n** option to restart the server without reinstalling the system services. Type the following in a Shell window:

```
/usr/sybase/Install/startserver –n –s serverName
```

You might want to add the line

```
source /usr/sybase/scripts/SetVars
```

into **~sybase/.cshrc** and your own **~/.cshrc** file so that you always have the environment variables set correctly. Make sure that **/usr/sybase/interfaces** and **/usr/sybase/scripts/SetVars** are identical on all machines that access the same server.

Playing with isql (as sybase)

Use **isql** to test the server installation.

1. Log in as **sybase**.
2. Make sure your environment variables are set correctly. See the last paragraph in the section above.
3. Follow the instructions on page 14 of

   /NextLibrary/Documentation/NeXT/RelNotes/SybaseNotes.wn

   to test the database.
4. Start up the Shell application.

You'll now configure the master database so that the examples will work.

5. In a Shell window, tell the master server that there's a new user: The new user is **sybase**, with a password of *server_password* and privileges to create a database. The password must be *server_password* for the examples to work. (What you type is shown in bold.)

```
% isql –Usa –P
1> sp_addlogin sybase, server_password
2> go
```

```
New login created.
(return status = 0)
1> sp_adduser sybase
2> go
New user added.
(return status = 0)
1> grant create database to sybase
2> go
1> grant create table to sybase
2> go
1> quit
%
```

You can modify or add the password for **sa** (System Administrator) if you want to have some security for that login. Look up **sp_password** in the Sybase Documentation using the Digital Librarian. The entry for **COMREF/SPROCS/sp_password.nr** contains more information about this command. The first time you use this command, a password will be added:

```
% isql -Usa -P
1> sp_password null, NewPassword
2> go
```

Programming with SYBASE (as sybase)

You must be logged in as **sybase** to make the following fixes and run the examples discussed in this section.

1. Log in as **sybase**.

You'll now make a few fixes.

2. Edit **/usr/sybase/include/sybfront.h**.

a. The following line should be deleted:

```
typedef unsigned char   BOOL;    /* So older programs won't
                                    break yet.*/
```

b. Change the line

```
typedef BOOL    (*BOOLFUNCPTR)();
```

to:

```
typedef DBBOOL  (*BOOLFUNCPTR)();
```

c. Change the Macros section to:

```
#ifndef MAX
#define MAX(a, b)    ((a) > (b) ? (a) : (b))
#endif
#ifndef MIN
#define MIN(a, b)    ((a) < (b) ? (a) : (b))
#endif
#ifndef ABS
#define ABS(a)   ((a) > 0 ? (a) : -(a))
#endif/
```

3. Edit **/usr/sybase/include/sybdb.h**. Change each occurrence of the word BOOL to DBBOOL. Note, there is a macro called **dbSetBool**, do *not* change this one! This will allow you to write programs with or without Objective-C code.

4. Edit **/usr/sybase/sample/dblibrary/example9.c**. Change each occurrence of the word BOOL to DBBOOL.

5. Start up the Shell application.

6. In a Shell window, move to the location of the examples and open the makefiles.

```
% cd /usr/sybase/sample/dblibrary
% open makefile*
```

7. Modify the makefiles. Change the first line in each of the three makefiles from

```
ROOT     =    $(SYBASE)/rel4.0/release
```

to:

```
ROOT     =    $(SYBASE)
```

Now you can compile the examples:

```
% make
```

or individual ones (note that this is wrong in **/usr/sybase/sample/dblibrary/README**):

```
% make exam1
```

To try them out:

```
% exam1
% exam2
. . .
```

If you get the following error for example 1,

```
Msg 4002, Level 14, State 1

        Login failed
DB-Library: Login incorrect.
```

it's because you haven't added the user who's running that example to the master server database correctly.

Note: example8.c will not compile because the file **rpctest** doesn't exist.

Integrating SYBASE and Interface Builder

You can create an interface for your **dblib** programs with the Interface Builder™ application. Once you have an interface, you need to add a file named **Makefile.preamble** to your project directory. This allows you to include the SYBASE **include** files when you compile and the SYBASE library when you link. The file **Makefile.preamble** should contain the following two lines:

```
CFLAGS = -I/usr/sybase/include
OTHER_LIBS = /usr/sybase/lib/libsybdb.a
```

Appendix

/usr/sybase/interfaces

```
DSLISTEN
    master tcp next-ether localhost 3696

DSQUERY
    query tcp next-ether localhost 3696

DSCONSOLE
    console tcp next-ether localhost 4696

SYBASE
    master tcp next-ether localhost 3696
    query tcp next-ether localhost 3696
    console tcp next-ether localhost 4696
    debug tcp next-ether localhost 5696
    trace tcp next-ether localhost 6696

localhost
    master tcp next-ether localhost 3696
    query tcp next-ether localhost 3696
    console tcp next-ether localhost 4696
    debug tcp next-ether localhost 5696
    trace tcp next-ether localhost 6696

myServer
    master tcp next-ether myHost 3696
    query tcp next-ether myHost 3696
    console tcp next-ether myHost 4696
    debug tcp next-ether myHost 5696
    trace tcp next-ether myHost 6696

otherServer
    master tcp next-ether otherHost 3690
    query tcp next-ether otherHost 3690
    console tcp next-ether otherHost 4690
    debug tcp next-ether otherHost 5690
    trace tcp next-ether otherHost 6690
```

Document Revision History

This document is revised from time to time. The following list shows the release date of each version of this document, and the changes from the previous version.

Date	**Changes from Last Version**
December 3, 1989	Created
April 17, 1990	Added information on using **dblib** programs with Interface Builder and setting up networked servers. Major format and editing modifications.
May 30, 1990	Added information about shutdown.