



Technical Notes

Q: When I try to invoke BB^x on my Novell Network, I get the following error:

```
fs load err=70 (70)
Sorry, all user slots are filled. Please try again later.
bbx: unable to initialize file system
```

A: Unfortunately, this is a common problem under Novell. Here's an overview of what's happening:

BB^x keeps a user map in Novell to deal with licensing and locking issues. It does this via a bindery object on the Novell server, which is based on the serial number of the BB^x product. This bindery object can be copied to another server via:

- 1) disk mirroring
- 2) installing the same serial number on another server
- 3) running a local copy of BB^x while logged into another server when:
 - a) you are logged in as supervisor
 - b) the other server is your primary login server

When BB^x is invoked, it queries the other servers on the network to verify that the bindery object doesn't exist on any other server. If a duplicate object is found, you will get the following error:

```
fs load err=70 (70)
Sorry, all user slots are filled. Please try again later.
bbx: unable to initialize file system
```

The solution to this problem is quite simple - make sure that the bindery object exists only on the server on which BB^x is installed. To help you do this, we've provided a stand-alone executable file - **DEACTIV.EXE** - that should be located in the same directory as the BB^x executable. Here are the steps to accomplish the task:

- 1) Find out all the servers that are on your network. You can do this via the 'SLIST' command under Novell.
- 2) Make a note of every server on the network, excluding the server on which BB^x is supposed to be installed.
- 3) Log into each one of the other servers as supervisor (or admin if under NetWare 4.X), specifying that server as your primary server. This can be done via the following command:
login server/supervisor where 'server' is the server to which you are attaching.
- 4) Finally, use the **DEACTIV.EXE** program to remove the bindery object.

DEACTIV BBXSERIALNUMBER

where 'BBXSERIALNUMBER' is your BB^x serial number. Note that you may want to copy the **DEACTIV.EXE** program to a floppy or a local drive, so that it will always be accessible when logged onto different servers. If the bindery object was on that server, you will get a message saying that BB^x has been deactivated. If not, it will say that the serial number was not installed on that server. Once this has been done on all servers (except the one where BB^x is supposed to be located), you should no longer get the fs load err=70.

continued...

Under NetWare 4.1, this error can occur for yet another reason. It stems from the way NetWare 4.1 does its bindery emulation for backwards compatability. In NetWare 3.x each server had its own bindery and never paid any attention to other server's binderies. NetWare 4.x uses NDS (NetWare Directory Services) which replaces the bindery. However, for BB^x to run, it still requires a bindery object for licensing. Unfortunately, in Novel 4.x, although you can turn on bindery emulation, it can cause a problem for BB^x. The bindery context can be set and shared by multiple servers. Therefore if BB^x is installed on one server, it appears to be installed on ALL the servers which is a license violation and the cause for the fs load err=70. One solution is to assign each server its own "bindery context". At the console you type SET BINDERY CONTEXT = <something>. As long as each server has its own context, the emulated binderies will be unique and BB^x's serial number detection won't be fooled. This trick is only practical if you don't depend on the bindery for other things. The new NetWare drivers tend to use Directory Services instead of the bindery, so you should be okay unless you're using older drivers.

