

## Re: SAN sample code

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2005-10/msg00608.html>

---

- *From:* "Maxim S. Shatskih" <[maxim@xxxxxxxxxxxxxxxxxxxx](mailto:maxim@xxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 24 Oct 2005 03:00:42 +0400
- 

> 2) Even if you'd manage to write own NDIS protocol driver it would take  
> you years to make it  
> BSD sockets compatible at user level.

Just expose TDI as the upper edge of the protocol and write the WinSock helper DLL. The simplified version of the TCP/IP WinSock helper is in the DDK samples – WSHSAMPLE.

> Ask Max how long did it take them  
> to write BSD sockets  
> library for NT kernel.

Several months – this is with debugging and testing.

> And it was for TCP/IP and not for own protocol.

It is the universal TDI client, but have 1 IP-specific file to handle conversion of sockaddr\_in to TA\_ADDRESS.

> driver for it) – NDIS protocol (it would be you who would write it) –  
> AFD equivalent – User mode DLL

No need in AFD equivalent. If the protocol exposes TDI upper edge – then existing AFD and MSAFD.DLL will be able to operate it.

You can also expose the proprietary IOCTL upper edge and the user-mode WinSock provider DLL which will talk to these IOCTLs – but this means you will need to have things like SO\_SNDBUF/RCVBUF/lingering close/listen backlog to be implemented yourself.

—  
Maxim Shatskih, Windows DDK MVP  
StorageCraft Corporation  
[maxim@xxxxxxxxxxxxxxxxxxxx](mailto:maxim@xxxxxxxxxxxxxxxxxxxx)  
<http://www.storagecraft.com>

- **References:**

- ◆ **SAN sample code**
  - ◇ *From:* sudhir
- ◆ **Re: SAN sample code**
  - ◇ *From:* Maxim S. Shatskih
- ◆ **Re: SAN sample code**
  - ◇ *From:* sudhir
- ◆ **Re: SAN sample code**
  - ◇ *From:* KOOLER

- Prev by Date: **Re: Frameworks [WAS: Cost of developing a device driver]**
- Next by Date: **Re: SAN sample code**
- Previous by thread: **Re: SAN sample code**
- Next by thread: **Re: SAN sample code**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**