

## APPENDIX B: GUIDELINES FOR PROTOCOL INDEPENDENCE

B

By defining a set of services common to many transport protocols, the Transport Interface offers protocol independence for user software. However, all transport protocols do not support all the services supported by the Transport Interface. If software must be run in a variety of protocol environments, only the common services should be accessed. The following guidelines highlight services that may not be common to all transport protocols.

- In the connection-mode service, the concept of a transport service data unit (TSDU) may not be supported by all transport providers. The user should make no assumptions about the preservation of logical data boundaries across a connection. If messages must be transferred over a connection, a protocol should be implemented above the Transport Interface to support message boundaries.
- Protocol and implementation specific service limits are returned by the **t\_open** and **t\_getinfo** routines. These limits are useful when allocating buffers to store protocol-specific transport addresses and options. It is the responsibility of the user to access these limits and then adhere to the limits throughout the communication process.
- User data should not be transmitted with connect requests or disconnect requests [see **t\_connect(3N)** and **t\_snddis(3N)**]. All transport protocols do not support this capability.
- The buffers in the **t\_call** structure used for **t\_listen** must be large enough to hold any information passed by the client during connection establishment. The server should use the **T\_ALL** argument to **t\_alloc**, which will determine the maximum buffer sizes needed to store the address, options, and user data for the current transport provider.
- The user program should not look at or change options that are associated with any Transport Interface routine. These options are specific to the underlying transport protocol. The user should choose not to pass options with **t\_connect** or **t\_sndudata**. In such cases, the transport provider will use default values. Also, a server should use the options returned by **t\_listen** when accepting a connection.
- Protocol-specific addressing issues should be hidden from the user program. A client should not specify any protocol address on **t\_bind**, but instead should allow the transport provider to assign an appropriate address to the transport endpoint. Similarly, a server should retrieve its address for **t\_bind** in such a way that it does not require knowledge of the transport provider's address

## APPENDIX B: GUIDELINES FOR PROTOCOL INDEPENDENCE

### B

space. Such addresses should not be hard-coded into a program. A name server mechanism could be useful in this scenario, but the details for providing such a service are outside the scope of the Transport Interface.

- The reason codes associated with **t\_rcvdis** are protocol-dependent. The user should not interpret this information if protocol-independence is a concern.
- The error codes associated with **t\_rcvuderr** are protocol-dependent. The user should not interpret this information if protocol-independence is a concern.
- The names of devices should not be hard-coded into programs, because the device node identifies a particular transport provider, and is not protocol independent.
- The optional orderly release facility of the connection-mode service (provided by **t\_sndrel** and **t\_rcvrel**) should not be used by programs targeted for multiple protocol environments. This facility is not supported by all connection-based transport protocols. In particular, its use will prevent programs from successfully communicating with ISO open systems.