

# Remote Procedure Call

Remote Procedure Call (RPC) is a protocol that provides the high-level communications paradigm used in the operating system. RPC presumes the existence of a low-level transport protocol, such as Transmission Control Protocol/Internet Protocol (TCP/IP) or User Datagram Protocol (UDP), for carrying the message data between communicating programs. RPC implements a logical client-to-server communications system designed specifically for the support of network applications.

This chapter provides the following information about programming RPC:

The RPC protocol is built on top of the eXternal Data Representation (XDR) protocol, which standardizes the representation of data in remote communications. XDR converts the parameters and results of each RPC service provided.

The RPC protocol enables users to work with remote procedures as if the procedures were local. The remote procedure calls are defined through routines contained in the RPC protocol. Each call message is matched with a reply message. The RPC protocol is a message-passing protocol that implements other non-RPC protocols such as batching and broadcasting remote calls. The RPC protocol also supports callback procedures and the **select** subroutine on the server side.

A *client* is a computer or process that accesses the services or resources of another process or computer on the network. A *server* is a computer that provides services and resources, and that implements network services. Each network *service* is a collection of remote programs. A remote program implements remote procedures. The procedures, their parameters, and the results are all documented in the specific program's protocol.

RPC provides an authentication process that identifies the server and client to each other. RPC includes a slot for the authentication parameters on every remote procedure call so that the caller can identify itself to the server. The client package generates and returns authentication parameters. RPC supports various types of authentication such as the UNIX and Data Encryption Standard (DES) systems.

In RPC, each server supplies a program that is a set of remote service procedures. The combination of a host address, program number, and procedure number specifies one remote service procedure. In the RPC model, the client makes a procedure call to send a data packet to the server. When the packet arrives, the server calls a dispatch routine, performs whatever service is requested, and sends a reply back to the client. The procedure call then returns to the client.

The RPC interface is generally used to communicate between processes on different workstations in a network. However, RPC works just as well for communication between different processes on the same workstation.

The Port Mapper program maps RPC program and version numbers to a transport-specific port number. The Port Mapper program makes dynamic binding of remote programs possible.

To write network applications using RPC, programmers need a working knowledge of network theory. For most applications, an understanding of the RPC mechanisms usually hidden by the **rpcgen** command's protocol compiler is also helpful. However, use of the **rpcgen** command circumvents the need for understanding the details of RPC.

- **RPC Model**

The remote procedure call (RPC) model is similar to a local procedure call model. In the local model, the caller places arguments to a procedure in a specified location such as a result register.

- **RPC Message Protocol**

This section explains the RPC message protocol.

- **RPC Authentication**

The caller may not want to identify itself to the server, and the server may not require an ID from the caller. However, some network services, such as the Network File System (NFS), require stronger security. Remote Procedure Call (RPC) authentication provides a certain degree of security.

- **RPC Port Mapper Program**

Client programs must find the port numbers of the server programs that they intend to use. Network transports do not provide such a service; they merely provide process-to-process message transfer across a network. A message typically contains a transport address consisting of a network number, a host number, and a port number.

- **Programming in RPC**

Remote procedure calls can be made from any language. Remote Procedure Call (RPC) protocol is generally used to communicate between processes on different workstations. However, RPC works just as well for communication between different processes on the same workstation.

- **RPC Features**

This section explains the Remote Procedure Call (RPC) features.

- **RPC Language**

The Remote Procedure Call Language (RPCL) is identical to the eXternal Data Representation (XDR) language, except for the added program definition.

- **rpcgen Protocol Compiler**

The **rpcgen** protocol compiler accepts a remote program interface definition written in the Remote Procedure Call language (RPCL), which is similar to the C language.

- **List of RPC Programming References**  
This section lists the RPC programming references.
- **Using UNIX Authentication Example**  
This section describes example for using UNIX authentication.
- **DES Authentication Example**  
This section describes example for Data Encryption Standard (DES) authentication.
- **Using the Highest Layer of RPC Example**  
This section explains using the highest layer of RPC example.
- **Using the Intermediate Layer of RPC Example**  
This section explains using the intermediate layer of RPC example.
- **Using the Lowest Layer of RPC Example**  
This section explains using the lowest layer of RPC example.
- **Showing How RPC Passes Arbitrary Data Types Example**  
This section explains two examples to show how Remote Procedure Call (RPC) handles arbitrary data types.
- **Using Multiple Program Versions Example**  
This section explains using multiple program versions example.
- **Broadcasting a Remote Procedure Call Example**  
This section explains broadcasting a Remote Procedure Call (RPC) example.
- **IPv6 concerns for Transport Independent Remote Procedure Call**  
Network Services Library exports Remote Procedure Call (RPC) routines that are both Transport Independent (TI) RPC routines as well as Transport Specific (TS) RPC routines.
- **Using the select Subroutine Example**  
This section explains using the select subroutine example.
- **rcp Process on TCP Example**  
This section explains rcp process on TCP example.
- **RPC Callback Procedures Example**  
This section explains the Remote Procedure Call (RPC) examples.
- **RPC Language ping Program Example**  
This section explains the Remote Procedure Call language (RPCL) ping program example.
- **Converting Local Procedures into Remote Procedures Example**  
This section explains the examples of converting local procedures into remote procedures.
- **Generating XDR Routines Example**  
This section explains the procedure for generating XDR routine example.

**Parent topic:**

→ [Communication Programming Concepts](#)

