NAME

rdma_getaddrinfo - Provides transport independent address translation.

SYNOPSIS

#include <rdma/rdma_cma.h>

int rdma_getaddrinfo (const char **node*, const char **service*, const struct rdma_addrinfo **hints*, struct rdma_addrinfo ***res*);

ARGUMENTS

node — Optional, name, dotted-decimal IPv4, or IPv6 hex address to resolve.

service — Service name or port number of address.

hints — Reference to an rdma_addrinfo structure containing hints about the type of service the caller supports.

res — A pointer to a linked list of rdma_addrinfo structures containing response information.

DESCRIPTION

Resolves the destination node and service address and returns information needed to establish communication. Provides the RDMA functional equivalent to getaddrinfo.

RETURN VALUE

Returns 0 on success, or -1 on error. If an error occurs, errno will be set to indicate the failure reason.

NOTES

Either node, service, or hints must be provided. If hints are provided, the operation will be controlled by hints.ai_flags. If RAI_PASSIVE is specified, the call will resolve address information for use on the passive side of a connection. If node is provided, rdma_getaddrinfo will attempt to resolve the RDMA address, route, and connection data to the given node. The hints parameter, if provided, may be used to control the resulting output as indicated below. If node is not given, rdma_getaddrinfo will attempt to resolve the RDMA addressing information based on the hints.ai_src_addr, hints.ai_dst_addr, or hints.ai_route.

rdma_addrinfo

ai_flags — Hint flags that control the operation. Supported flags are:

RAI_PASSIVE

Indicates that the results will be used on the passive/listening side of a connection.

RAI_NUMERICHOST
: If specified, then the node parameter, if provided, must be a numerical network address. This flag suppresses any lengthy address resolution.

RAI_NOROUTE
: If set, this flag suppresses any lengthy route resolution.

RAI_FAMILY
: If set, the ai_family setting should be used as an input hint for interpretting the node parameter.

ai_family
: Address family for the source and destination address. Supported families are: AF_INET, AF_INET6, and AF_IB.

ai_qp_type
: Indicates the type of RDMA QP used for communication. Supported types are: IBV_UD (unreliable datagram) and IBV_RC (reliable connected).

ai_port_space
: RDMA port space in use. Supported values are: RDMA_PS_UDP, RDMA_PS_TCP, and RDMA_PS_IB.

ai_src_len
: The length of the source address referenced by ai_src_addr. This will be 0 if an appropriate source address could not be discovered for a given destination.

ai_dst_len
: The length of the destination address referenced by ai_dst_addr. This will be 0 if the RAI_PASSIVE flag was specified as part of the hints.

ai_src_addr
: If provided, the address for the local RDMA device.

ai_dst_addr
: If provided, the address for the destination RDMA device.

ai_src_canonname
: The canonical for the source.

ai_dst_canonname
: The canonical for the destination.

ai_route_len
: Size of the routing information buffer referenced by ai_route. This will be 0 if the underlying transport does not require routing data, or none could be resolved.

ai_route
: Routing information for RDMA transports that require routing data as part of

connection establishment. The format of the routing data depends on the underlying transport. If Infiniband transports are used, ai_route will reference an array of struct ibv_path_data on output, if routing data is available. Routing paths may be restricted by setting desired routing data fields on input to rdma_getaddrinfo. For Infiniband, hints.ai_route may reference an array of struct ibv_path_record or struct ibv_path_data on input.

ai_connect_len
: Size of connection information referenced by ai_connect. This will be 0 if the underlying transport does not require additional connection information.

ai_connect
: Data exchanged as part of the connection establishment process. If provided, ai_connect data must be transferred as private data, with any user supplied private data following it.

ai_next
: Pointer to the next rdma_addrinfo structure in the list. Will be NULL if no more structures exist.

SEE ALSO

rdma_create_id(3), rdma_resolve_route(3), rdma_connect(3), rdma_create_qp(3), rdma_bind_addr(3), rdma_create_ep(3)