

MAXBUFF - Maximum Buffer Size

For static specification, this parameter is available as a keyword subparameter of the NTRPC macro. For dynamic specification, this parameter is available as a keyword subparameter of the profile parameter RPC.

MAXBUFF can be specified on both the client and the server side.

On the server side, it determines the size of the buffer provided by the server to receive the client request including data and to send back the result. The buffer must be large enough to hold the largest of the following two data areas for all client requests:

- the request received by the client,
- the result send back to the client.

If the size of the buffer is too small for a request, a temporary buffer with the required size is allocated and used for this request.

On the client side, it determines the size of the buffer provided for the automatic execution of Natural RPC calls. This buffer is used to build the client request including data and to receive the result from the server. The buffer must be large enough to hold the largest of the following two data areas for all requests sent by the client:

- the request send to the server,
- the result received from the server.

If the size of the buffer is too small for a request, a temporary buffer with the required size is allocated and used for this request.

For further information, see *Stubs and Automatic RPC Execution* in the *Natural Remote Procedure Call (RPC)* documentation.

The size of the data exchanged between the client and server is provided by the stub generation function of the SYSRPC utility. To calculate the size for automatic RPC execution, you may use the SYSRPC CSMASS command; see *Calculating Size Requirements* in the *SYSRPC Utility* documentation.

Possible settings	1-2097147, but smaller than or equal to RPCSIZE-4	Maximum buffer size in KB. The maximum buffer size must be equal to or less than the value (minus 4) specified with the profile parameter RPCSIZE (for the server side, see below).
	0	No buffer is allocated.
Default setting	0	
Dynamic specification	yes	
Specification within session	no	

Dependency on Number of Parameters on Server Side

On the server side, the difference between `RPCSIZE` and `MAXBUFF` depends on the maximum number of parameters n in the PDA and can be calculated as follows:

- If group structures are present:

$$\text{MAXBUFF} = \text{RPCSIZE} - (3 + n/10)$$

- If no group structures are present:

$$\text{MAXBUFF} = \text{RPCSIZE} - (3 + n/20)$$

Example:

If $n=100$ and $\text{RPCSIZE}=128$, then $\text{MAXBUFF}=120$.

Dependency on `ACIVERS` Settings

In case of an EntireX Broker node, special considerations apply if you are using Entire Net-Work as a transport layer. With Entire Net-Work, the receive buffer length passed to the EntireX Broker stub is restricted by the startup parameter `IUBL` and must not exceed 32 KB. Depending on the setting of the Natural profile parameter `ACIVERS`, the receive buffer length is set as follows:

- `ACIVERS=1`: 32000
- `ACIVERS=2`: 30K
- `ACIVERS>2`: the value specified with `MAXBUFF`

For further information, see the *Natural Remote Procedure Call (RPC)* documentation.