

ACCESSING THE DDE SERVER VIA A C PROGRAM

Accessing the DDE server from a C program is quite easy. For a complete list of commands consult the help manuals for your specific C program. Here is a sample of what you should be able to find from Borland C++.

DDE functions

<u>DdeAbandonTransaction</u>	Abandons an asynchronous transaction
<u>DdeAccessData</u>	Accesses a DDE global memory object
<u>DdeAddData</u>	Adds data to a DDE global memory object
<u>DdeClientTransaction</u>	Begins a DDE data transaction
<u>DdeCmpStringHandles</u>	Compares two DDE string handles
<u>DdeConnect</u>	Establishes a conversation with a server application
<u>DdeConnectList</u>	Establishes multiple DDE conversations
<u>DdeCreateDataHandle</u>	Creates a DDE data handle
<u>DdeCreateStringHandle</u>	Creates a DDE string handle
<u>DdeDisconnect</u>	Terminates a DDE conversation
<u>DdeDisconnectList</u>	Destroys a DDE conversation list
<u>DdeEnableCallback</u>	Enables or disables one or more DDE conversations
<u>DdeFreeDataHandle</u>	Frees a global memory object
<u>DdeFreeStringHandle</u>	Frees a DDE string handle
<u>DdeGetData</u>	Copies data from a global memory object to a buffer
<u>DdeGetLastError</u>	Returns an error value set by a DDEML function
<u>DdeInitialize</u>	Registers an application with the DDEML
<u>DdeKeepStringHandle</u>	Increments the usage count for a string handle
<u>DdeNameService</u>	Registers or unregisters a service name
<u>DdePostAdvise</u>	Prompts a server to send advise data to a client
<u>DdeQueryConvInfo</u>	Retrieves information about a DDE conversation
<u>DdeQueryNextServer</u>	Obtains the next handle in a DDE conversation list
<u>DdeQueryString</u>	Copies string-handle text into a buffer
<u>DdeReconnect</u>	Reestablishes a DDE conversation
<u>DdeSetUserHandle</u>	Associates a user-defined handle with a transaction
<u>DdeUnaccessData</u>	Frees a DDE global memory object
<u>DdeUninitialize</u>	Frees DDEML resources associated with an application