

BLink.DLL

DLL LOAD “BLink”

DLL GET “SHORT BLink:Setlink” [AS *alias*]

DLL GET “SHORT BLink:Getlink” [AS *alias*]

DLL GET “VOID BLink:Startlink” [AS *alias*]

DLL GET “VOID BLink:Stoplink” [AS *alias*]

[*Variable* = FN]Setlink(*LinkID*,*pData*,*type*,*direction*)

This function creates one side of a 2 - 4 sided link.

Variable

Basic Type: INTEGER

C Type: short

- 0 Success
- 1 Link limit reached
- 2 Invalid type
- 3 Invalid Direction
- 4 Invalid Link ID
- 5 Link ID already exists
- 6 Server with this ID already exists
- 7 Unable to secure data
- 8 Unable to create link object

LinkID

Basic Type: INTEGER

C Type: short

This is a numeric value greater than zero that uniquely identifies the link.

pData

Basic Type: Basic variable by reference

C Type: void *

This is the memory location of the variable to be linked.

type

Basic Type: String\$

C Type: char *

The Basic type of the Variable as a string. valid strings are:

STRING
INTEGER
LONG
REAL

direction

Basic Type: String\$
C Type: char *

Use a literal string or string variable containing “IN” or “OUT”. If “OUT” is specified it indicates that this instance is the server for this link. The Server for a link is the instance that will distribute the data contained in the variable to the other linked instances. Only one side of a 2 - 4 sided link should specify “OUT”.

[*Variable* = FN]Getlink(*LinkID*,*option*)

This function is called by the non-server instances to request an update of the linked variables.

Variable

Basic Type: INTEGER
C Type: short

-1	Invalid Option Specified
0	no updates occurred
positive number	count up successful updates

LinkID

Basic Type: INTEGER
C Type: short

This is a numeric value greater than zero that uniquely identifies the link to be updated. This value may be ignored depending on the *option* parameter.

option

Basic Type: INTEGER
C Type: short

If a 0 is specified the LinkID parameter is ignored and all linked variables will request an update. If a 1 is specified only the link identified by the *LinkID* parameter will request an update.

Startlink()

This function activates the server thread for answering requests from non-server instances. A Server will not answer requests until the server thread is running.

Stoplink()

This function stops the server thread that answers requests from non-server instances. This function should be called before unloading the DLL or exiting HTBasic.