

## DAPIO32 Interface for Visual Basic .NET and C#

The DAPIO32 interface for VB.NET and C# are in the files DAPIO32.VB and DAPIO32.CS, respectively. The interface files and examples are installed as part of the DAP Development Software version 3.00 or later. The latest version of Visual Studio tested with the interface files is Microsoft Visual Studio 2010.

### Visual Basic .NET

The latest version of DAPIO32.VB, version 3.00, has some minor changes from the previous support for VB.NET. It also has some differences from Visual Basic. If you are updating the interface file in an existing application in .NET or converting an old application from Visual Basic, note the differences described below.

- Changes from previous VB.NET support
  - DAP handle type changes from int to IntPtr.
  - Query result field type in the TDapHandleQuery structure changes from String to IntPtr.
  - The file DAPIOERR.VB is retired. The function is merged into DAPIO32.VB.
- Changes from Visual Basic support
  - A variable of type Short in VB.NET is the same as one of type Integer in previous versions of VB. In VB.NET a Short is 2-byte and an Integer or a Long is 4-byte.
  - The name of the buffer array, instead of the first element of the array, should be passed to the DAPIO32 functions that require one. For example, in previous versions of VB, the function call to DapBufferGet might be DapBufferGet(hDapBinGet, BytesGet, BufData(0)). In VB.NET, the last parameter should be simply BufData.

### Examples

Two examples, DVM and QUERY, for VB.NET are included in the DAP Development Software. Open an example with the file of type "Microsoft Visual Studio Solution". The examples can be compiled as a 32-bit or 64-bit application. To compile an example as a 64-bit application, configure Visual Studio to build for x64 platform.

The DVM example shows a simple digital voltmeter where the application downloads the DAPL configuration in DVM.dap to the DAP to generate a sinewave and send it back to the application. The application program reads a block of data from the DAP board on a one-second timer and displays the last value from the data block. The user starts and stops the program via a button at the program interface.

The QUERY example illustrates the various ways to use the DapHandleQuery function. DapHandleQuery can return query results either in the QueryResult field of the TDapHandleQuery structure as a 32-bit integer or in a user-provided buffer as a scalar of any type, an ASCII string or a double null-terminated multiple ASCII string list. This example shows how to use each of the storage mechanism. This example can be compiled as a 32-bit or 64-bit console program.

## C#

The latest version of DAPIO32.CS, version 3.00, has some minor changes from the previous support for C#. If you are updating the interface file in an existing application in C#, the changes below might require modifying your source code.

- Changes from previous C# support
  - DAP handle type changes from int to IntPtr.
  - Query result field type in the TDapHandleQuery structure changes from String to IntPtr.
  - Query result field name in the TDapHandleQuery structure changes from psz to QueryResult.

### Examples

Two examples, DIOTEST and QUERY, for C# are included in the DAP Development Software. Open an example with the file of type “Microsoft Visual Studio Solution”. The examples can be compiled as a 32-bit or 64-bit application. To compile an example as a 64-bit application, configure Visual Studio to build for x64 platform.

The DIOTEST example shows how each function in the DAPIO32 interface is called in C#. The program invokes one function when the user clicks on a button at the application interface.

The QUERY example in C# performs the same function as the QUERY example in VB.NET – it illustrates the various ways to use the DapHandleQuery function. DapHandleQuery can return query results either in the QueryResult field of the TDapHandleQuery structure as a 32-bit integer or in a user-provided buffer as a scalar of any type, an ASCII string or a double null-terminated multiple ASCII string list. This example shows how to use each of the storage mechanism. This example can be compiled as a 32-bit or 64-bit console program.