Everything's possible.

DriveLibrary™
Quick Start Guide
ADVANCED Motion Controls constantly strives to improve all of its products. We review the information in this document regularly and we welcome any suggestions for improvement. We reserve the right to modify equipment and documentation without prior notice.

For the latest revision of this document, visit the company's website at www.a-m-c.com. Otherwise, contact the company directly at:

ADVANCED Motion Controls • 3805 Calle Tecate Camarillo, CA • 93012-5068 USA

This manual is for the exclusive use of ADVANCED Motion Controls. The reproduction, transmission or use of this document or its contents is prohibited without the expressed written permission of ADVANCED Motion Controls.

General Safety

You must install and operate ADVANCED Motion Controls motion control equipment so that you meet all applicable safety requirements. Ensure that you identify the relevant standards and comply with them. Failure to do so may result in damage to equipment and personal injury.

Read this entire manual prior to attempting to install or operate the drive. Become familiar with practices and procedures that allow you to operate these drives safely and effectively. You are responsible for determining the suitability of this product for the intended application. ADVANCED Motion Controls is neither responsible nor liable for indirect or consequential damages resulting from the inappropriate use of this product.

High-performance motion control equipment can move rapidly with very high forces. Unexpected motion may occur especially during product commissioning. Keep clear of any operational machinery and never touch them while they are working.
Keep clear of enclosed units, motor terminals, and transformer terminals when power is applied to the equipment. Follow these safety guidelines:

- Always turn off the main power and allow sufficient time for complete discharge before making any connections to the drive.
- Make sure that the minimum inductance requirements are met. Pulse Width Modulated (PWM) amplifiers deliver a pulsed output that requires a minimum amount of load inductance for proper operation.
- Do not rotate the motor shaft without power. The motor acts as a generator and will charge up the power supply capacitors through the amplifier. Excessive speeds may cause over-voltage breakdown in the power output stage. Note that an amplifier having an internal power converter that operates from the high voltage supply will become operative.
- Do not short the motor leads at high motor speeds. When the motor is shorted, its own generated voltage may produce a current flow as high as 10 times the amplifier current. The short itself may not damage the amplifier but may damage the motor.
- Do not make any connections to any internal circuitry. Only connections to designated connectors are allowed.

**Related Documentation**

- Product datasheet specific for your drive, available for download at www.a-m-c.com

<table>
<thead>
<tr>
<th>Document ID</th>
<th>Revision</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNDGDLQS-01</td>
<td>1.0</td>
<td>1/2018</td>
<td>First Release</td>
</tr>
<tr>
<td>MNDGDLQS-02</td>
<td>1.1.0</td>
<td>8/2018</td>
<td>Updated for DriveLibrary 1.1.0</td>
</tr>
</tbody>
</table>

© 2018 ADVANCED Motion Controls. All rights reserved.
DriveLibrary™ is a high level API interface that defines a collection of control functions to support developers of motion control solutions.

Included in the DriveLibrary installation are a handful of demo solutions to aid developers in understanding the basic concepts of working with DriveLibrary. This document describes the steps to successfully configure Visual Studio, make sure all the necessary files are available, and run a DriveLibrary demo program.

For additional resources and information, the DriveLibrary API Reference Manual is available for download at www.a-m-c.com.

**Prerequisites**

Before beginning, make sure the following items are installed and available:

1. DriveLibrary
2. DriveWare (version 7.4.1 or greater)
3. Microsoft Visual Studio Community 2015 or greater
4. **ADVANCED** Motion Controls’ DigiFlex Performance servo drive
   - Supports CANopen, RS485/232, or Ethernet communication
   - Fully tuned and commissioned
Step 1 - Locating the Demo Project

By default, during the DriveLibrary installation process the demo solutions are saved to:

C:\Users\Public\Documents\DriveLibrary 1.0\Demos\DriveLibrary_Demos.sln

Open this solution in Visual Studio. Included in this Visual Studio Solution are 6 projects:

1. **Hello Drive** - Initializes the drive and reads the active events.
2. **Initialize** - Powers on the drive, reads position on power-up, runs the Homing routine, and finishes by reading the position after Homing.
3. **IO Test** - Initializes the drive, reads the current value of digital input and digital output 1, changes the state of digital output 1, and finally prints the initial read states to the console.
4. **Position Moves** - Powers on the drive, runs the Homing routine, runs the drive through various absolute and relative position moves, and returns the drive to Home and powers down.
5. **Status** - Powers on the drive, checks the current events, puts the drive in a fault state, resets the drive, and displays the events to show it is no longer faulted.
6. **Velocity Moves** - Powers on the drive, starts a velocity move, reads velocity when it reaches the target, runs for 20 seconds, halts the drive, and powers down.

Each Visual Studio project inside of the solution will need to be configured individually if you choose to use it. All projects are configured in the same manner. This example will show how to setup **Hello Drive**.
Step 2 - Configuring Visual Studio for the Demo Solutions

The first step is to configure "library" and "include" directories. This is done by right-clicking on the project name in the Solution pane, and selecting Properties.

Navigate to Configuration Properties > General, and select the Windows SDK Version that is installed on the PC.
Once the Windows SDK version is selected, the **Additional Include Directories** parameter must be specified. This is found in **Configuration Properties > C/C++ > General**. Click the dropdown arrow on the far right of the dialog box, and select **Edit**. From the pop-up window that appears, click the **New Line** icon to add the include directory.

The include directory by default is in the same directory as the DriveLibrary installation:

```
C:\Program Files (x86)\Advanced Motion Controls\DriveLibrary 1.0\Include
```

Next, the **Additional Library Directories** parameter must be configured. Go to **Configuration Properties > Linker > General** and click the dropdown arrow on the right of **Additional Library Directories**, and select **Edit**.

In this case, the specified directory will depend on if the final build is for debug or release. The default directory for the debug libraries is:

```
C:\Program Files (x86)\Advanced Motion Controls\DriveLibrary 1.0\Library\Debug
```

If the build is for release, replace "Debug" with "Release" in the directory path.
The next step is to specify the name of the static library file in the **Additional Dependencies** parameter. Go to **Configuration Properties > Linker > Input**, and from the dropdown at the far right of the **Additional Dependencies** line, select **Edit**. In the window that appears, type "DriveLibrary.lib" - this is the static library file used for DriveLibrary during builds.

![Additional Dependencies Window]

The final configuration piece for Visual Studio is to copy all the ".dll" files from:

C:\Program Files (x86)\Advanced Motion Controls\DriveLibrary 1.0\Library\Debug

...to the Debug folder for your project.
Step 3 - Configuring your Drive for DriveLibrary

Create a folder locally that Visual Studio can access. This will be where all the configuration files for the servo drive in use will be stored.

Open DriveWare, and connect to the drive.

**RS485/232**  To configure a drive for RS485/232, open Configuration 0 in the System Browser in DriveWare, and set the Command Source to **DriveLibrary**.

Store this configuration to the drive by selecting the **Store** icon.

Save the drive project configuration file to the folder created on the local machine by using **File > Save Project As...**

**CANopen**  To configure a CANopen drive, go to the **Settings** window in the System Browser in DriveWare, and set the Drive Operation to **Network Mode (CANopen)**.
Open Configuration 0 in the System Browser, and set the Command Source to DriveLibrary.

![System Browser with Configuration 0 and Command Source set to DriveLibrary](image)

Go to Network > Settings, and set the Baud Rate to **1000Kb/s**, the Address to the preferred Drive Address, and the Initial Mode of Operation to **Config 0**.

![Network Settings with Baud Rate set to 1000Kb/s, Address set to Drive Address, and Initial Mode of Operation set to Config 0](image)

Go to Network > RPDO and click the Load Preset button. Select Drive Defaults from the list to load the default COB-ID/Indexes.

![Network RPDO with Load Preset button and Drive Defaults selected](image)

Repeat the last step for Network > TPDO.

Store this configuration to the drive by selecting the Store icon.

![Store icon selected](image)

Save the drive project configuration file to the folder created on the local machine by using **File > Save Project As...**
**Ethernet**  To configure an Ethernet drive, go to the *Settings* window in the System Browser in DriveWare, and set the *Drive Operation* to *Stand-Alone Mode*, and the *Ethernet Protocol* to *Ethernet*.

Open Configuration 0 in the System Browser, and set the *Command Source* to *DriveLibrary*.

Go to *Network > Settings*, and set the *IP Address* to **192.168.100.1**.

Store this configuration to the drive by selecting the *Store* icon.

Save the drive project configuration file to the folder created on the local machine by using *File > Save Project As*...
Step 4 - Editing the Parameter Files

After setting up the project for Visual Studio and configuring the drive for DriveLibrary, the parameter files must be edited to provide the information for DriveLibrary to connect to the drive.

Default config files are provided in:

C:\Program Files (x86)\Advanced Motion Controls\DriveLibrary 1.0\Config

Copy these config files to the same folder that the DriveWare project files were saved to during Step 3.

**RS485/232** Open **Serial_Network_COM4.anf**. For RS485, under [Network] set the type to "6". For RS232, under [Network] set the type to "0".

Next, under [Network], set the baudrate to the same as configured in DriveWare. Specify the port as the same port used to connect DriveWare to the drive.

Under [Axis1], change the address to the same address specified in DriveWare, and the path to the name of the DriveWare ".adf" file saved in Step 3.

Both "name" values can be optionally changed to a value preferred by the user.

```plaintext
# Valid RS232 Network
[Network]
name=Network RS232 Valid
type=0
baudrate=115200
port=COM1

[Axis1]
name=Left Spinner
address=63
pvt_step_size=75
status_interval=200
spooler_interval=20
path=DL Test Project.gradle
```

If you change the name of the ".anf" file, open Serial_System.asf and change the path under [Network1] to the new name of your ".anf" file.
**CANopen**  Open CAN\_Network\_anf. Under [Network] set the type to "3".

Next, under [Network], set the baudrate to the same as configured in DriveWare. Specify the port as the channel number of the CAN device on the system.

Under [Axis1], change the address to the same address specified in DriveWare, and the path to the name of the DriveWare ".adf" file saved in Step 3. The default PVT Step Size value of either 50 or 75 (distance between PVT points in milliseconds) should work for most systems, however this value can be adjusted between 20 and 255 milliseconds.

Both "name" values can be optionally changed to a value preferred by the user.

```
# Valid configuration for CAN network with 1 drives.
[Network]
name=AMC CAN
type=3
baudrate=1000000
port=0

[Axis1]
name=Left Spinner
address=1
pvt_step_size=75
status_interval=100
spooler_interval=20
path=NL\_Test\_Project\_adf
```

---

**Ethernet**  Open Ethernet\_Network\_anf. Under [Network] set the type to "4".

Next, under [Network], set the IP address to the same as configured in DriveWare.

Under [Axis1], change the address to the same address specified when connecting to DriveWare, and the path to the name of the DriveWare ".adf" file saved in Step 3. The default PVT Step Size value of either 50 or 75 (distance between PVT points in milliseconds) should work for most systems, however this value can be adjusted between 20 and 255 milliseconds.

Both "name" values can be optionally changed to a value preferred by the user.
# Valid UDP configuration for Ethernet with 1 drive

[Network]
name=Ethernet Network
type=1

[Axis1]
name=Left Spinner
address=63
ipaddress=192.168.100.1
pvt_step_size=75
status_interval=200
path=Ethernet_config.asf

---

Notice

If you change the name of the "*.anf" file, open Ethernet_System.asf and change the path under [Network1] to the new name of your "*.anf" file.
Step 5 - Modifying the C++ File to Connect to the Drive

The final task before the project can be run is to modify the variables `asfConfigFile` and `axisName` in the Hello Drive C++ code. The variable `asfConfigFile` should be set to the full path of the "asf" file, and `axisName` should be set to the name of the axis specified in the "anf" file.

Once this has been done, rebuild the project in Visual Studio. A number of warnings will appear, however these can be ignored as they are stating that some debug information is missing from the source code.

After the successful build, run the project in Visual Studio by clicking the Local Windows Debugger button.

The Hello Drive program will connect to your drive using the parameters specified, and a console window will appear displaying a list of axis events. By placing a break point at the end of the demo source code, these values can be viewed and cross-referenced with the list of axis events in Appendix E of the DriveLibrary API Manual.