Commander User Guide

D-000146 Rev 01

Page intentionally left blank

Commander User Guide

Some Important Links

Further information and brochures	*Commander Online*
Sales and support contact information	*Product, Sales and Service Enquiries*

Document Reference: D-000146 Rev 01

Effective: 29-04-2015

Page intentionally left blank

Chapter Summaries

Table of Contents

1 Introduction

1.1 About this user guide

This guide introduces you to ANCA Motion's Commander software. The guide provides instructions for installing and getting started with Commander, including some core concepts.

In this user guide, the following instructional icons are used:

ⓘ **Note**

✓ **Tip**

⚠ **Caution**

❗ **Warning**

The information contained in this guide was correct at the time of writing, but is subject to change. Please ensure you always refer to the version of the guide corresponding to the Commander version you are using.

1.2 About Commander

ANCA Motion Commander is an application that enables you to build other applications. Commander is particularly suited to building interfaces and industrial control applications for CNCs (Computer Numerical Controls) from ANCA Motion. Rich visuals and logic expressions can be combined intuitively to create powerful user experiences with no programming required.

Using a comprehensive library of pre-defined components you can create interfaces to suit a wide variety of applications. You can customize these pre-defined components or create your own using common tools and technologies.

1.2.1 Features

Commander is built upon state-of-the-art technologies, enabling many key and unique features:

- No coding required
- Live WYSIWYG (What You See Is What You Get) design – no compilation
- Highly customizable
- Made for Touch
- Windows desktop integration
- Modern graphics

1.3 Related documents

CHANGELOG – history of product changes, located in the Commander Application directory following installation

1.4 Terms and abbreviations

UI	User Interface
WYSIWYG	What You See Is What You Get
CNC	Computer Numerical Control
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
URI	Uniform Resource Identifier
OS	Operating System
RAM	Random Access Memory
SSE2	Streaming SIMD Extensions 2
SIMD	Single Instruction Multiple Data
NuGet	NuGet is the package manager for the Microsoft development platform including .NET (see *https://www.nuget.org* for more information).
Widget	User Interface (UI) elements / objects
Layout	Canvas / container for Widgets
Tag	Unique identifier / symbolic name for an I/O point
Tag Table	A collection of Tags
Controller	General term used for connected equipment (PLC, drive, etc.)
Triggers / Actions	Triggers and Actions model cause-and-effect relationships. A Trigger reacts to the cause and invokes one or more Actions.

2 Getting started

2.1 System requirements

Component	Requirement
Computer and processor	1 gigahertz (GHz) or faster x86- or x64-bit processor with SSE2 instruction set
Memory (RAM)	1 gigabyte (GB) RAM (32-bit); 2 gigabytes (GB) RAM (64-bit)
Hard Disk	300 MB available
Display	Graphics hardware acceleration requires a DirectX 9 (or higher) compliant graphics processor – this helps increase the performance of specific features. Most computers available since 2007 meet or exceed this standard.
Operating System	Commander runs on 32-bit and 64-bit versions of Microsoft Windows operating systems. When you run Commander on a 64-bit version of a Windows operating system, the program runs in the 32-bit layer of the Windows operating system. Commander supports the following Windows operating systems: • Windows 8 (32-bit or 64-bit) • Windows 8.1 (32-bit or 64-bit)
.NET version	4.5
Multi-touch	A touch-enabled device is required to use any multi-touch functionality. However, all features and functionality are always available by using a keyboard, mouse, or other standard or accessible input device.
Additional requirements and considerations	Some functionality may vary, based on the system configuration. Some features may require additional or advanced hardware or network connectivity.

Table 2-1 System requirements

Note

Windows 8 includes the .NET Framework 4.5, so you don't have to install it separately. Similarly, Windows 8.1 includes the .NET Framework 4.5.1.

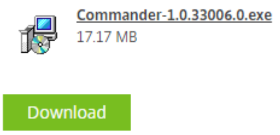
The hard disk system requirement is intentionally larger than the actual disk space usage of the software.

2.2 Installation

This section describes how to install Commander via the Windows desktop or via the command line (allows silent unattended installs). If you are upgrading from a previous version of Commander, the steps are the same. Before installing Commander, make sure that your computer meets the *2.1 System requirements*.

2.2.1 Install via Windows desktop

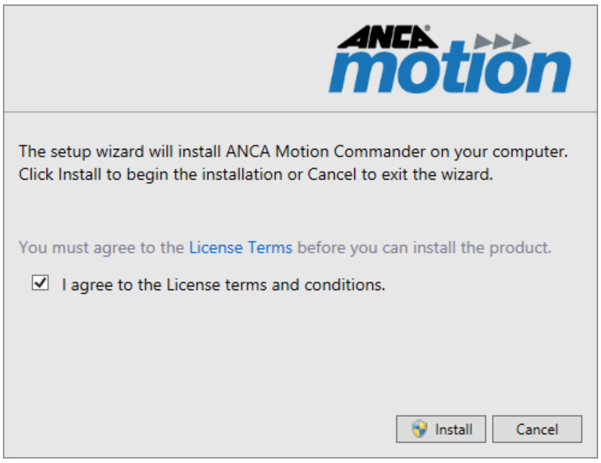
1. Visit the Commander download link provided by ANCA Motion in any browser.



2. Click on the green download button to download the Commander installer. Depending on your connection speed, the download may take up to a few minutes.

3. If prompted, click 'Run' or 'Save'.

4. If you have saved the installer, double-click the installer file to start the installation process.



The setup wizard will install ANCA Motion Commander on your computer. Click Install to begin the installation or Cancel to exit the wizard.

You must agree to the License Terms before you can install the product.

☑ I agree to the License terms and conditions.

5. After agreeing to the license terms (available by clicking the 'License Terms' link) click 'Install'

2.2.2 Install via command line

Installing Commander via the command line allows you to install, repair, or uninstall silently. A return code (console error level) of 0 indicates that the action was successful.

2.2.2.1 Installation parameters

Use the parameters in the following table to develop command line scripts for installation.

Parameter	Description
`-install, -repair, -uninstall` **Optional**	Optionally specify the installation action to perform (one of: install, repair, or uninstall). By default the install action is performed.
`-quiet` **Optional**	Specifies that Setup runs in a quiet mode without any user interface. This is used for unattended installations.
`-log` **Optional**	Output installation log to a specified file (e.g. `/log log.txt`). By default a log file is created in `%TEMP%`

Table 2-2 Installation parameters

Sample syntax:

To silently repair Commander without showing the user interface.

```
Commander-1.0.33006.0.exe -repair -quiet
```

Note

If you specify the `-quiet` switch you should check the returned error level for success (0). For example, if installation is attempted on an unsupported OS version, the error level will be set to (HRESULT) 0x8007047E.

2.3 Create your first interface with Commander

If you follow the steps in this section, you'll create an interactive Commander application which visually simulates a car dashboard without learning a programming language. Before beginning you will need to download and extract *CarDashboard.zip*, which contains a partial sample, and remember where you extracted it.

This section makes use of a virtual controller which is predefined to simulate aspects of a car by updating outputs (e.g. SPEED, REVS, etc.) based on various inputs (e.g. ACCELERATE, BRAKE, etc.). These inputs and outputs are referred to as **Tags**. In a typical application an interface would be connected to tags defined on an external controller, such as ANCA Motion’s AMCore.

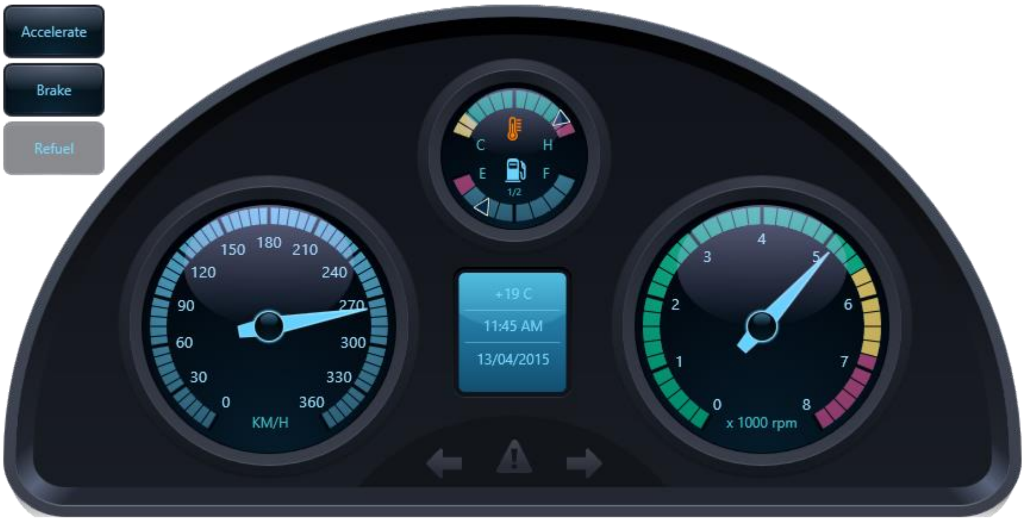
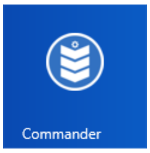


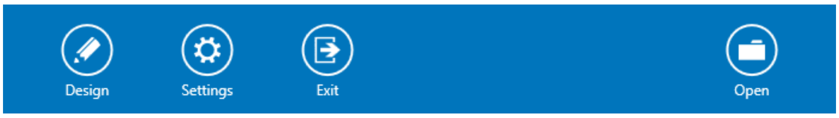
Figure 2-1 Create your first interface

2.3.1 Install and open Commander

1. Install Commander from the provided installation package by following the steps in section *2.2 Installation*, and then start Commander by clicking or tapping its tile on the Start screen.



2. The Commander app bar will open across the bottom of your screen by default.



The app bar will automatically hide itself if it loses focus. To show the app bar either: move your mouse cursor to the bottom edge of the screen, or click / tap the translucent Commander icon in the bottom left corner of your screen.

3. On the right side of the app bar, click **Open** and browse to the folder containing the files extracted from CarDashboard.zip. Locate and open the tag table for the application - CarDashboardTags.ctt.

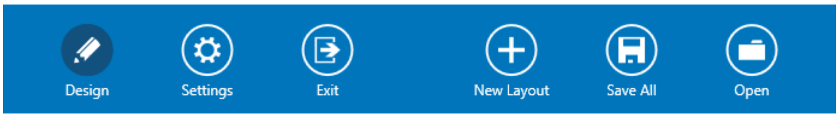
4. Repeat step 3 to open CarDashboard.cla - the (partially complete) dashboard layout.

2.3.2 Add and configure a widget

When you develop a Commander application, you’ll add UI elements, called widgets or visuals, to show text, graphics, and other information. In this section, you’ll add a button which will simulate the accelerator of the car, and then you’ll configure the button by changing several of its properties. For example, you’ll specify the text that appears in the button, the tags that are bound to the button, and the size and location of the button itself. Other types of widgets, such as labels, have similar properties, and you can change them in similar ways.

2.3.2.1 Add a widget, and change a property

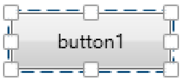
1. On the left side of the app bar, click **Design** to enter Commander's design-mode.



> **Tip**
>
> Click or tap **Design** at any time to toggle between design-mode and run-mode. The button changes colour to indicate that design-mode is active. Whilst in design-mode the design *palette* is available. The *palette* appears on the right side of the screen by default and can be moved by dragging its title bar.

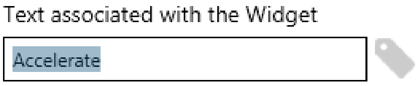
2. In the *palette toolbox*, drag and drop a **Button** onto the car dashboard layout.

3. Click or tap the newly created button to select it (surrounded by a dashed line).



   Note that when the widget is selected the *palette* changes to display the *properties* of the widget. Whilst a UI element is selected you can toggle back and forth between the *toolbox* and *properties* using the button in the title bar of the *palette*.

4. In the property labelled *'Text associated with the Widget'*, change the text that appears on the button by typing **Accelerate**.



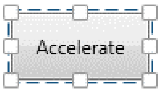
Text associated with the Widget

Accelerate

5. Update the text on the button by clicking anywhere outside of the text box.

2.3.2.2 Resize and move a widget

1. With the accelerate button selected, resize it by dragging the small white square in the lower-right corner of the selection box until the button is roughly the same size as the brake and refuel buttons.



2. Drag the button to the top-left of the layout (above the brake button – refer to *Figure 2-1 Create your first interface*). Note that you will be prevented from dragging the widget outside of the layout bounds.

> **Tip**
>
> If you want to delete a widget, select it, and then click or tap the **Delete** command in the lower section of the *palette*.

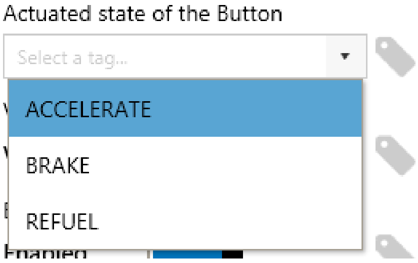
2.3.2.3 Connect properties to data

1. With the accelerate button selected, scroll or flick through the list of *properties* in the *palette* to find *'Actuated state of the Button'* (under the *Behavior* category).

   This property is a read-only Boolean (true/false) value corresponding to whether the button is actuated (in this case, pressed). For example, if the button is clicked and held, this property will remain true until it is released.

2. The tag icon on the right side of the property editor indicates that this value can be connected (bound) to a *tag*. Click or tap the tag icon to access tags which can be bound to this property (based on data type and accessibility).

3. From the list, select the **ACCELERATE** tag. The property is now bound to this tag.

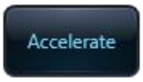


4. Repeat steps 2 and 3 to bind the **CAN_ACCELERATE** tag to the property labelled *'Enabled state of the Widget'*.

5. Exit design-mode by clicking or tapping **Design** on the app bar. Test your interface by clicking and holding Accelerate, you should see the gauges come alive.

2.3.3 Refine your interface

- Change the **Visual style** and **Foreground colour** of the accelerate button to match the brake and refuel buttons.



- Personalize your work by adding a **Label** with your name in a script font.

- Play an engine sound effect (e.g. wav audio file) when the accelerate button is actuated (pressed). To do this add and configure a **Play Sound** action to the trigger property *'Action(s) to trigger when the Button is actuated'.*

Tip

Multiple actions are triggered in the order they are added. Click or tap an action to expand available properties for that action. To delete an action, right-click or tap and hold it, and then click or tap the red delete icon that is shown.

3 Publish and deploy

3.1 Launch your interface

A Commander interface typically consists of a combination of one or more *Layout* files (.cla) and one or more *Tag Table* files (.ctt). Although it is possible to open each file one by one from the Commander app bar, as described in *2.3 Create your first interface with Commander*, this is not suited to end-user deployment scenarios.

In general, it is recommended that you launch your interface using command line parameters. A single command in a shortcut or script, for example, can launch multiple Layouts (and Tag Table(s)). By default, if an interface is launched in this way, the Commander app bar is hidden. This is desirable when end-users should not have the ability to alter the interface.

3.1.1 Command line parameters

Use the parameters in the following table to develop command line scripts for launching Commander. The command should follow the format:

```
Commander.exe [OPTION]... [FILE]...
```

If no files are specified, Commander will launch in its default state.

Parameter	Description
`-m, --menu` **Optional**	Force the Commander app bar (main menu) to be shown. By default the app bar is hidden if one or more files is specified, and shown if no files are specified.
`-s, --source=PATH` **Optional**	A source repository PATH to search for plugins. Use this parameter in combination with the `--install` parameter to install and/or update plugins.
`-i, --install=NAME` **Optional**	The NAME of a plugin package to install. Use this parameter in combination with the `--source` parameter to install and/or update plugins.
`-q, --quiet` **Optional**	Output launch errors to the command prompt only.
`-h, -?, --help` **Optional**	Output this usage information to the command prompt.

Table 3-1 Command line parameters

Sample syntax:

To install or update the plugin with filename `pluginName.1.0.0.nupkg` located in the directory `C:\My Plugins`

```
Commander.exe --source=”C:\My Plugins” --install=pluginName
```

To launch a Layout `C:\My Interfaces\layout1.cla` using the Tag Table `C:\My Interfaces\tags1.ctt` with the Commander app bar hidden (default)

```
Commander.exe “C:\My Interfaces\tags1.ctt” “C:\My Interfaces\layout1.cla”
```

3.2 Deploy a Commander interface

There are a number of considerations to be made when developing an end-user deployment solution for your Commander interface.

- Location of Layout files (.cla)
- Location of Tag Table files (.ctt)
- Referenced files and paths
- Plugins used
- How the end-user will launch your interface

The simplest approach is to deploy any referenced / used files (cla, ctt, .nupkg, wav) to a single directory on the target machine, ensuring that any absolute paths are maintained. A simple script deployed to this directory can act as a single entry point for launching your interface via a shortcut or otherwise.

Sample script:

For example, we have an interface which uses the following files deployed to `C:\My Interface\`

Filename	**Description**
main.cla	Our main interface layout (utilises some XAML styles from myPlugin)
menu.cla	A menu layout which is opened from main.cla via an **Open Layout** *Action* referencing `C:\My Interface\menu.cla`
myTags.ctt	Tag definitions used by main.cla
click.wav	A sound effect played on menu.cla button presses via **Play Sound** *Actions* referencing `C:\My Interface\click.wav`
myPlugin.1.0.0.nupkg	A custom plugin containing some XAML visual style used by main.cla

Table 3-2 Example resources

We can create a very simple Windows batch script named myInterface.bat in the `C:\My Interface\` directory to conveniently launch our interface.

myInterface.bat

```
START “Commander” /b “C:\Program Files (x86)\ANCA
Motion\Commander\commander.exe” --source=”C:\My Interface” --install=myPlugin
myTags.ctt main.cla
```

This script will:

- Install myPlugin if it is not already installed
- Open myTags.ctt
- Open main.cla (Commander app bar will be hidden)

Note

Commander will ignore the `--source` and `--install` parameters if the latest version of the plugin it finds is already installed.

4 Extending Commander – plugins

Commander is designed to be extremely modular and extensible. All of the components in Commander are based on a few key extension points made available through Commander's plugin system. Plugins can provide various user interface components including Widgets and Visual Styles, Actions that can be triggered, property type editors and so on.

The Commander plugin system is built around NuGet packages (.nupkg). Commander extension points are made available via NuGet packages located in the Commander application directory following installation (e.g. `C:\Program Files (x86)\ANCA Motion\Commander\Packages\`), and Commander plugins created using these extension points are themselves packaged and installed as NuGet packages, see *4.1 Install a Commander plugin*.

4.1 Install a Commander plugin

Before beginning, ensure you have installed Commander on the target machine, see *2.2 Installation*.

1. Save the plugin you wish to install (.nupkg file) to your target machine (e.g. `C:\My Plugins\pluginName.1.0.0.nupkg`). You may delete the .nupkg file after the plugin has been installed.

2. Open a console window
   a. Open the charms bar by swiping in from the right (or moving the mouse cursor to the top / bottom right corner) of the screen
   b. Tap or click the **Search** charm
   c. Enter `cmd` in the search box
   d. In the search results, tap or click **Command Prompt**

3. Change the working directory to the Commander application directory, e.g.

```
cd "C:\Program Files (x86)\ANCA Motion\Commander
```

4. Launch Commander specifying the parameters to install your plugin (see *3.1.1 Command line parameters* for more information), e.g.

```
Commander.exe --source="C:\My Plugins" --install=pluginName
```

⚠ **Caution**

Only the NuGet package name (in this example 'pluginName') should be specified in the `--install` switch. Do not include the version part of the file name.

✓ **Tip**

Follow the same steps to update an existing plugin with a newer version. Commander will automatically uninstall the older version if it is installed.

5 Web services

5.1 Interact with Commander using HTTP

Commander provides an API for interacting with layouts. Layouts can be opened, closed, or queried using the HTTP verbs POST, DELETE and GET. GET to query open layouts, DELETE to close a layout or POST to open a layout.

5.1.1 General

5.1.1.1 Accepted types

It is possible to negotiate which media type will be returned as a representation of the resource being retrieved. Use the Accept header in the request to specify the desired media type. The application/xml and application/json media types are understood as response formats for layout related resources.

5.1.1.2 Built-in layouts

Generally there is a direct mapping from a layout to a file, however there are some 'built-in' layouts which do not follow this rule. Built-in layouts can be referenced using the Commander URI scheme **`cmdr://`**. Under this scheme built-in layouts are identified by name (just as normal layouts) i.e. `cmdr://layouts/{name}`.

The following built-in layouts are available.

Name	URI	Description
keyboard1	`cmdr://layouts/keyboard1`	The Windows default virtual keyboard
keyboard2	`cmdr://layouts/keyboard2`	The Windows thumb layout virtual keyboard (central number pad)
keyboard3	`cmdr://layouts/keyboard3`	The Windows full virtual keyboard (top row with numbers)

Table 5-1 Built-in layouts

5.1.2 REST API

5.1.2.1 Layouts resource

Lists all currently open layouts (GET) and accepts new layouts to open (POST).

Verbs	URI
`GET, POST`	`http://localhost/commander/layouts`

Table 5-2 Layouts resource

Layouts resource example (GET)

```
URI: http://localhost/commander/layouts
Method: GET
Accept: application/json

Status Code: 200
Body: [“layout1”, “layout2”]
```

Layouts resource example (POST - Success)

```
URI: http://localhost/commander/layouts
Method: POST
Accept: application/json
Content-Type: application/json
Body: { “uri”: “file:///C:/layout2.cla” }

Status Code: 200
```

Layouts resource example (POST - Success)

```
URI: http://localhost/commander/layouts
Method: POST
Accept: application/json
Content-Type: application/json
Body: { “uri”: “cmdr://layouts/keyboard2” }

Status Code: 200
```

Layouts resource example (POST - Error)

```
URI: http://localhost/commander/layouts
Method: POST
Accept: application/json
Content-Type: application/json
Body: { “uri”: “file:///C:/foo.txt” }

Status Code: 500
Body: “The file ‘C:\\foo.txt’ has an unsupported file extension.”
```

5.1.2.2 Layout resource

Resource for interacting with individual layouts.

Verbs	URI
`GET, DELETE`	`http://localhost/commander/layouts/{name}`

Table 5-3 Layout resource

⚠ **Caution**

If a layout is open but has not been saved to disk: a GET will return a status code of 200, but the URI returned will be empty; a DELETE will close the layout losing any unsaved changes.

Layout resource example (GET - Success)

```
URI: http://localhost/commander/layouts/layout1
Method: GET
Accept: application/json

Status Code: 200
Body: { “uri”: “file:///C:/layout1.cla” }
```

Layout resource example (GET - Success)

```
URI: http://localhost/commander/layouts/keyboard1
Method: GET
Accept: application/json

Status Code: 200
Body: { “uri”: “cmdr://layouts/keyboard1” }
```

Layout resource example (GET - Error)

```
URI: http://localhost/commander/layouts/foo
Method: GET
Accept: application/json

Status Code: 404
```

Layout resource example (DELETE - Success)

```
URI: http://localhost/commander/layouts/layout1
Method: DELETE

Status Code: 200
```

Layout resource example (DELETE - Error)

```
URI: http://localhost/commander/layouts/foo
Method: DELETE

Status Code: 404
```

6 Troubleshooting

Use the following troubleshooting table to find solutions to problems that have identifiable symptoms.

Symptom	Action
When I try to run the Commander installer / setup wizard I get the error "The specified program requires a newer version of Windows"	1. Ensure you are installing Commander on a supported OS, see *2.1 System requirements*.
I don't know which version of Commander I have installed.	Please find the version using one of the following techniques: Using Commander 1. If Commander is running and the app bar is not hidden, click or tap **Settings.** 2. The version is specified under **About** Using Windows Control Panel 1. Open Control Panel 2. Navigate to **Programs and Features** 3. In the list of programs, find *ANCA Motion Commander* 4. The version is specified in the **Version** column Using Windows File Explorer 1. Navigate to Commander's application directory (e.g. `C:\Program Files (x86)\ANCA Motion\Commander`) 2. Right-click on Commander.exe and select **Properties** from the context menu. 3. Select the **Details** tab 4. The version is specified as the **Product version**
When launching Commander I get the warning "Failed to start web service. The required network port is already in use."	By default Commander listens for traffic on port 80. If another application is using (and has locked out) port 80 (e.g. Skype) then Commander may be unable to start its web service. If you are not utilizing Commander's web service, you may safely ignore this warning.
When I click / tap the Tag icon next to a property there are no Tags listed to select from.	1. Ensure you have a *Tag Table* (.ctt) file open 2. Check that there is one of more Tags which match the data type and accessibility of this property. For example, if the property is a string, Commander will only list Tags of data type string; or if the property is read-only, Commander will only list Tags that are writeable, etc.
When I open a Tag Table (.ctt) file nothing happens.	This is a known limitation at the time of writing. If no errors were raised you can assume that the Tag Table opened successfully.
My interface doesn't seem to be reading / writing the I/O of my controller.	1. Ensure you have a *Tag Table* (.ctt) file open 2. Check that expected properties are bound to expected Tags. 3. Confirm that your controller is running
After deploying my interface to another computer the visual styles have changed.	1. Confirm that any necessary plugins are installed. If unsure try installing the plugin(s) again, see *4.1 Install a Commander plugin*. 2. Restart Commander

Table 6-1 Troubleshooting

7 Product, Sales and Service Enquiries

If after reading the User Manual you still require assistance for installation, training or other customer support issues, please contact the closest ANCA Motion Customer Service Office in your area for details.

ANCA Motion Pty. Ltd.
1 Bessemer Road
Bayswater North
VIC 3153
AUSTRALIA
Telephone: +613 9751 8900
Fax: +613 9751 8901
www.ancamotion.com/Contact-Us
Email: sales.au@ancamotion.com

ANCA Motion Taiwan
1F, No.57, 37 Road
Taichung Industrial Park
Taichung 407
TAIWAN
Telephone: +886 4 2359 0082
Fax: +886 4 2359 0067
www.ancamotion.com/Contact-Us
Email: sales.tw@ancamotion.com