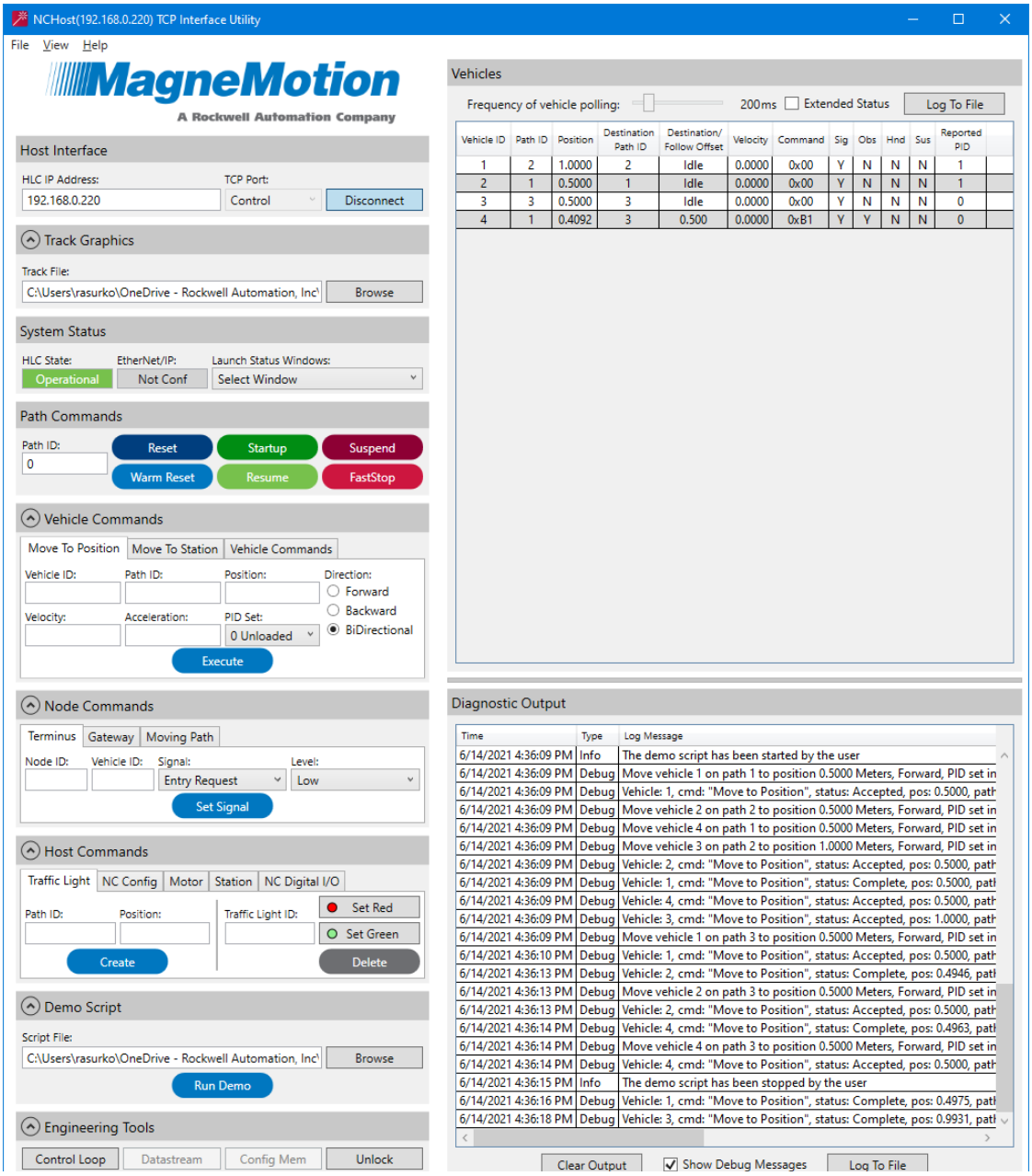
MagneMotion

A Rockwell Automation Company

NCHost TCP Interface Utility
User Manual



Original Instructions

Although every effort is made to keep this manual accurate and up-to-date, MagneMotion® assumes no responsibility for any errors, omissions, or inaccuracies. Information that is provided in this manual is subject to change without notice. Any sample code that is referenced in this manual or included with MagneMotion software is included for illustration only and is, therefore, unsupported.

This product is protected under one or more U.S. and International patents. Additional U.S. and International patents pending.

The information that is included in this manual is proprietary or confidential to MagneMotion, Inc. Any disclosure, reproduction, use, or redistribution of this information by or to an unintended recipient is prohibited. In no event will MagneMotion, Inc. be responsible or liable for indirect or consequential damage that results from the use or application of this equipment.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

MagneMotion, Inc.
A Rockwell Automation Company
139 Barnum Road
Devens, MA 01434
USA
Phone: +1 978-757-9100
Fax: +1 978-757-9200
rok.auto/ict

This technology is subject to United States Export Administration Regulations and authorized to the destination only; diversion contrary to U.S. law is prohibited.

Printed in the U.S.A.

Contents

2 Using the NCHost TCP Interface Utility

Figures

Tables

This page intentionally left blank.

Changes

Overview

This manual is changed as required to keep it accurate and up-to-date to provide the most complete documentation possible for the MagneMotion® NCHost TCP Interface Utility. This section provides a brief description of each change.

NOTE: Distribution of this manual and all addenda and attachments is not controlled. To identify the current revision, see the Literature Library on the Rockwell Automation website.

Rev. A

Initial release to support NCHost.exe Version 4.0.15, which replaces Host.exe.

Rev. B

Revised to support NCHost version 7.1.2.

Added support for the following new functions:

- FastStop (*FastStop Paths*, Host Commands).
- NC Digital I/O (*Node Controller Digital I/O*, *Digital I/O Output Mode Operation*, *Digital I/O Input Mode Operation*).
- Follow ID (*Testing the Transport System – Vehicle Commands*, *Vehicles*).
- Extended Vehicle Status (*Vehicles*).

Updated the following:

- In the *About This Manual* chapter, the list of *Related Documentation*.
- Updated figures to show new functions.
- In the *Appendix*, the *Transport System Limits* table.
- Updated the *Glossary* and *Index*.

Removed the following:

- All references to the standard node controller (replaced by the NC-12 node controller).
- Information about unsupported features (Traffic Light, Lock/Unlock Vehicle).

Rev. C

Revised to support NCHost version 7.3.4.

Added support for the following new functions:

- Moving Paths (monitoring *Moving Path Status*, *Moving Path Node*, *Moving Paths Status Window* status, and *Moving Path Tab* UI reference).

Added the following:

- In the *About This Manual* chapter, safety warnings related to vehicle motion.
- In *Chapter 1, Introduction*, the description of the node controller Console Interface and a description of the Virtual Scope Utility.
- In *Chapter 2, Using the NCHost TCP Interface Utility*, *Temporary Stations*.

Updated the following:

- Changed the logo to “A Rockwell Automation Company” version.
- Trademark and copyright information.
- Figures to show new functions.
- In the *About This Manual* chapter, the list of *Related Documentation*.
- In *Chapter 1, Introduction*, the *Simplified View of Transport System Software Organization*.
- In *Chapter 2, Using the NCHost TCP Interface Utility*, the *Monitoring the Transport System – Status Information* section to include references to appropriate commands. The *Terminus Node* example and *Gateway Node* example. Moved and updated the *Node Controller Digital I/O* usage.
- In *Chapter 4, UI Reference*, the Host Commands, Launch Status Windows, and *Node Commands* references.
- In the *Appendix*, the *Transport System Limits* tables.
- The *Glossary* and *Index*.

Rev. D

Revised to support the following NCHost Versions:

- NCHostML for MM LITE™ – 4.2.2.
- NCHostQS for QS – 7.4.0.

Added the following:

- In *Chapter 2, Using the NCHost TCP Interface Utility*, added *Motor Inverter Control* to the *Testing the Transport System – Path Commands* section.
- In *Chapter 4, UI Reference*, added the *Traffic Light Tab* and the *Motor Tab* to the Host Commands section.
- Added the *Back Cover* to the manual.

Updated the following:

- In the *About This Manual* chapter, the trademark and copyright information. Updated *Prerequisites*, *Manual Conventions*, and Contact Information.
- Moving Paths (monitor *Moving Path Status*, *Moving Paths Status Window* status window) to include Last Exited status.
- Vehicle Status (monitor *Vehicle Status*, *Vehicles* status) to include expanded Extended Status information.
- In *Chapter 4, UI Reference*, updated the *Vehicle Commands* and *Vehicles* sections.

Ver. 5/Rev. E

Revised to support the following NCHost Versions:

- NCHostML for MM LITE – 13.0.0.
- NCHostQS for QS – 7.4.2.

Added the following:

- In *Chapter 2, Using the NCHost TCP Interface Utility*, added the monitoring *HLC Status*, monitoring *Station Status*, and monitoring *Traffic Light Status* sections. Added *Traffic Lights* operation and *NC Configurations* operation.
- In *Chapter 4, UI Reference*, added *NC Config Tab*. Added *HLC Status Window*, *Station Status Window*, *Traffic Lights Status Window*.

Updated the following:

- Changed the revision from alpha (Rev. E) to numeric (Ver. 05).
- Updated the structure of the changes descriptions.
- Trademark and copyright information.
- Figures to show new functions.
- In the *About This Manual* chapter updated the Customer Support contact information.
- In *Chapter 2, Using the NCHost TCP Interface Utility*, the monitor *Node Controller Status* section to include extended status. Expanded the *Temporary Stations* operation section.
- In *Chapter 3, Demo Scripts*, changed the terms Asynchronous and Synchronous to Independent and Dependent respectively. Removed the Relay Node from the *Multiple Path Track Examples* to match the examples in other manuals.
- In *Chapter 4, UI Reference*, changed the name of the Node Controller tab to the *NC Digital I/O Tab*. Added additional, functions to the *Track Graphics*. Added additional fields to the *Node Controller Status Window* and the *Vehicles* pane.
- In the *Appendix*, updated the *MagneMotion Transport System Limits*.
- The *Glossary* and *Index*.

Rev. F

Revised to support the following NCHost Versions:

- NCHostML for MM LITE – 13.1.1.
- NCHostQS for QS – 7.7.1.

Added the following:

- In the *About This Manual* chapter, added safety warnings that are related to vehicle motion.
- In *Chapter 1, Introduction*, added descriptions of new utilities and new file types.
- In *Chapter 2, Using the NCHost TCP Interface Utility*, added support for *Vehicle Platooning* and *Vehicle Commands* including *Clear the Vehicle Suspect Bit*.
- In *Chapter 6, Troubleshooting*, added *Diagnostic Output Troubleshooting*.

Updated the following:

- Changed the revision to alpha (Rev. F) only.
- Updated all trademark and copyright information and moved to the back cover.
- Changed all Customer Support references from MagneMotion to ICT.
- Updated the titles and part numbers for all referenced manuals.
- Updated the appearance of the safety notices to match Rockwell Automation standards.
- Updated all UI images to reflect the current version of software.
- In the *About This Manual* chapter, updated the descriptions of the safety notices and the list of *Related Documentation*.
- In *Chapter 1, Introduction*, updated the *Simplified View of Transport System Software Organization*. Added descriptions of new utilities (NC File Retrieval Tool and Ethernet Motor Commissioning Tool) and file types.
- In *Chapter 2, Using the NCHost TCP Interface Utility*, added disconnect action descriptions to the *Stop the NCHost TCP Interface Utility* section. Restructured the *Testing the Transport System* section into multiple sections that support the different testing functions.
- In *Chapter 4, UI Reference*, updated the *Vehicle Commands* section to show the new tab structure and new functions. Updated the *Vehicles* pane section to include the new status columns that are shown in the Extended Vehicle Status.
- In *Chapter 5, Demo Script Command Reference*, added a description and examples of *Comments* and added the *All_Vehicles* Demo Script vehicle directive.
- In *Chapter 6, Troubleshooting*, updated the *NCHost TCP Interface Utility Troubleshooting*.
- In the *Appendix*, updated the *Transport System Limits* tables.
- Updated the *Glossary* and *Index*.

Removed the following:

- Removed all references to the *Mitsubishi PLC TCP/IP Library User Manual*.

Rev. G

Revised to support the following NCHost Versions:

- NCHost for all motor types – 15.0.2.

Added the following:

- In the *About This Manual* chapter, added safety warnings that are related to vehicle motion.
- In *Chapter 1, Introduction*, added descriptions of new utilities and new file types to the *Transport System Software Overview*.
- In *Chapter 2, Using the NCHost TCP Interface Utility*, added use of the View menu to the *Running the NCHost TCP Interface Utility* section. Added *Power Supply Status* to the *Monitoring the Transport System – Status Information* section. Added *Warm Reset Paths* to the *Testing the Transport System – Path Commands* section.
- In *Chapter 4, UI Reference*, added the View menu to the *Main Menu* section. Added the *Propulsion Power Supply Status Window*. Added Warm Reset to the *Path Commands* menu options in the *System Status* section.

Updated the following:

- Updated the structure of the manual to reflect new functions and changes to the user interface.
- Updated all figures to show new functions and changes to the UI layout.
- Updated all support references to the information on the *Back Cover*.
- In *Chapter 2, Using the NCHost TCP Interface Utility*, updated *Vehicle Platooning* to include new decoupling while in motion function.
- In *Chapter 4, UI Reference*, updated the *Settings Panel* section to show the new UI structure. Updated the *Host Interface* section and moved content to the new structure of *Host Interface*, *Track Graphics*, and *System Status*. Updated the Host Commands section to show the new structure as *Path Commands* and *Host Commands*. Updated the Launch Status Windows section to show its inclusion in the *System Status* section as menu options. Updated the *Vehicle Commands* section to show the new tab structure. Updated the *Host Commands* section to use tabs. Updated the *Vehicles* pane section to include the new status columns that are shown in the Extended Vehicle Status.
- In *Chapter 6, Troubleshooting*, updated the *Diagnostic Output Troubleshooting* tables.
- In the *Appendix*, updated the *MagneMotion Transport System Motion Limits* table.
- Updated the *Glossary* and *Index*.

Removed the following:

- Removed references to PLCs and PCs. These are now referred to as controllers.

This page intentionally left blank.

About This Manual

Overview

This section provides information about the use of this manual, including the manual structure, related documentation, format conventions, and safety conventions.

Purpose

This manual explains how to use the NCHost TCP Interface Utility to run a MagneMotion® transport system for testing and debugging. This manual also explains how to develop Demo Scripts to automate vehicle motion for that testing. This manual is not intended to provide a design guide for the installation, or a reference for the operation of a MagneMotion transport system.

Use this manual in combination with the other manuals and documentation that accompanies the transport system to design, install, configure, test, and operate a MagneMotion transport system. MagneMotion offers instructor-led training classes that provide additional experience.

Audience

This manual is intended for all users of MagneMotion transport systems that are using the NCHost TCP Interface Utility for testing and debugging the transport system.

NOTE: Depending on software version, user access privileges, and application configuration, certain operations that are described in this manual may not be available.

Prerequisites

The information and procedures that are provided in this manual assume the following:

- Basic familiarity with general-purpose computers and with the Windows® operating system.
- The transport system has been installed and configured, and full documentation for the transport system is available.
- All personnel who configure, operate, or service the transport system are properly trained.

MagneMotion Documentation

The documentation that is provided with the MagneMotion transport system includes this manual, which provides complete documentation for the use of the NCHost TCP Interface Utility. Other manuals in the document set, which are listed in the *Related Documentation* section, support installation, configuration, and operation of the transport system.

The examples in this manual are included solely for illustrative purposes. Because of the many variables and requirements that are associated with any LSM system installation, MagneMotion cannot assume responsibility or liability for actual use that is based on these examples.

Manual Conventions

The following conventions are used throughout this manual:

- Bulleted lists provide information in no specific order, they are not procedural steps.
- Numbered lists provide procedural steps or hierarchical information.
- Keyboard keys and key combinations (pressing multiple keys at a time) are shown enclosed in angle brackets. Examples: <F2>, <Enter>, <Ctrl>, <Ctrl-x>.
- Dialog box titles or headers are shown in bold type, capitalized exactly as they appear in the software. Example: the **Open XML Configuration File** dialog box.
- Responses to user actions are shown in italics. Example: *Motion on all specified paths is enabled.*
- Selectable menu choices, option titles, function titles, and area or field titles in dialog boxes are shown in bold type and are capitalized exactly as they appear in the software. Examples: **Add to End...**, **Paths**, **Path Details**, **OK**.
- **Dialog Box** – A window that solicits a user response.
- **Click** or **Left-click** – Press and release the left mouse button[1].
- **Right-click** – Press and release the right mouse button.
- **Double-click** – Press and release the left mouse button twice in quick succession.
- **Control-click** – Hold down <Ctrl> and press and release the left mouse button.
- **Click-and-hold** – Press down the left mouse button and hold it down while moving the mouse.
- **Select** – Highlight a menu item with the mouse or the tab or arrow keys.
- **Code Samples** – Shown in monospaced text. Example: `Paths`.

1. Mouse usage terms assume typical “right-hand” mouse configuration.

- **Data Entry** – There are several conventions for data entry:
  - **Exact** – The text is shown in quotes. Example: Enter the name “Origin” in the text field.
  - **Variable** – The text is shown in italics. Example: Save the file as *file_name*.xml.
- **Numbers** – All numbers are assumed to be decimal unless otherwise noted and use the US number format; that is, one thousand = 1,000.00. Non-decimal numbers (binary or hexadecimal) are explicitly stated.
  - **Binary** – Followed by $_2$, for example, 1100 0001 0101_2, 1111 1111 1111 1111_2.
  - **Hex** – Preceded by 0x, for example, 0xC15, 0xFFFF.
- **Measurements** – All measurements are SI (International System of Units). The format for dual dimensions is *SI_units* [*English_units*]; for example, 250 mm [9.8 in].
- **Part Numbers** – All part numbers that are shown are the catalog numbers for ordering the item being referenced. These numbers are shown in different formats, for example; 700-1483-00, MMI-HT-C2198-D032, 2198-KITCON-D032-L.
- Text in blue is a hyperlink. These links are active when viewing the manual as a PDF. Select a hyperlink to change the manual view to the page of the item referenced. In some cases, the item that is referenced is on the same page, so no change in the view occurs.

Notes, Safety Notices, and Symbols

Notes, Safety Notices, and Symbols that are used in this manual have specific meanings and formats. Examples of notes, the different types of safety notices and their general meanings, and symbols and their meanings are provided in this section. Adhere to all safety notices provided throughout this manual to help achieve safe installation and use.

Notes

Notes are set apart from other text and provide additional or explanatory information. The text for Notes is in standard type as shown in the following example.

NOTE: A note provides additional or explanatory information.

Safety Notices

Safety Notices are set apart from other text. The symbol on the left of the notice identifies the type of hazard. The text in the message panel identifies the hazard, methods to avoid the hazard, and the consequences of not avoiding the hazard.

Examples of the standard safety notices that are used in this manual are provided in this section. Each example includes a description of the hazard indicated. Labels may also be on or inside the equipment to provide specific precautions.

NOTICE	Identifies an informational notice that indicates practices that are not related to personal injury that could result in equipment or property damage.

IMPORTANT	Identifies information that is critical for the successful application and understanding of the product.

	**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help to identify a hazard, avoid a hazard, and recognize the consequence.

	**AUTOMATIC MOTION HAZARD:** Identifies information about practices or circumstances where the possibility of machinery automatically starting or moving exists, which could cause personal injury or death.

	**CRUSH HAZARD:** Identifies information about practices or circumstances where there are exposed parts that move, which could cause personal injury or death.

Manual Structure

This manual contains the following chapters:

- *Introduction*: Provides an overview of the NCHost TCP Interface Utility, which allows a quick start of a MagneMotion transport system for testing.
- *Using the NCHost TCP Interface Utility*: Describes how to use the NCHost TCP Interface Utility to start up and test the transport system.
- *Demo Scripts*: Provides step-by-step procedures and examples for creating Demo Scripts. These scripts, which the NCHost TCP Interface Utility runs, are useful to automate testing of the transport system.
- *UI Reference*: Provides an overview of the NCHost TCP Interface Utility user interface and a description of each option available through the UI.

- *Demo Script Command Reference*: Identifies and provides descriptions of the commands that are used in the Demo Scripts.
- *Troubleshooting*: Provides identification of errors that may be encountered when using the NCHost TCP Interface Utility, their meaning, and how to resolve them.
- *Appendix*: Provides additional information that is related to the NCHost TCP Interface Utility and MagneMotion transport systems.
- *Glossary*: Provides a list of terms and definitions that are used in this manual and for the transport system and its components.
- *Index*: Provides a cross-reference to this manual organized by subject.

NOTE: The version of the NCHost TCP Interface Utility that is supplied by MagneMotion may be newer than the version that is described in this manual (indicated in *Changes on page 15*). All features that are documented in this manual are supported as indicated. Specific builds of the NCHost TCP Interface Utility may not implement all of the features that are described in this manual.

Related Documentation

Before configuring or running the transport system, consult the following documentation:

- *MagneMover LITE Configurator User Manual*, MMI-UM008.
  or
  QuickStick Configurator User Manual, MMI-UM009.
- *Node Controller Interface User Manual*, MMI-UM001.
- *NCHost TCP Interface Utility User Manual*, MMI-UM010 (this manual).
- *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003.
  or
  Host Controller EtherNet/IP Communication Protocol User Manual, MMI-UM004.
- *MagneMover LITE User Manual*, MMI-UM002.
  QuickStick 100 User Manual, MMI-UM006.
  or
  QuickStick HT User Manual, MMI-UM007.
- *Node Controller Hardware User Manual*, MMI-UM013.
- *LSM Synchronization Option User Manual*, MMI-UM005.
- *Virtual Scope Utility User Manual*, MMI-UM011.

NOTE: Distribution of this manual and all addenda and attachments are not controlled. Changes to the document set or the software can be made at any time. To identify the current revisions or to obtain a current version, see *Rockwell Automation Support on page 264*.

This page intentionally left blank.

Introduction 1

Overview

This chapter provides an overview of the NCHost TCP Interface Utility that is used for test and verification trials. This chapter also provides a basic overview of the transport system hardware and software. Additionally, the basic set of tasks that are required to use the NCHost TCP Interface Utility with a MagneMotion® transport system are described.

Use this manual to test and debug the transport system using the NCHost TCP Interface Utility, either directly or by using scripts. Some of the procedures that are provided in this manual as examples vary in actual use depending on the transport system configuration, communication, and other variables.

NOTE: When using NCHost with a MagneMotion transport system, make sure to use the version of NCHost that was supplied with the software file package.

This manual supports:

- MagneMover® LITE transport systems.
- QuickStick® transport systems.

Included in this chapter are overviews of:

- The NCHost TCP Interface Utility.
- The transport system components.
- The transport system software.
- Getting started with the NCHost TCP Interface Utility.

NCHost TCP Interface Utility Overview

The NCHost TCP Interface Utility is a Windows® .NET software application that is provided by MagneMotion to run the transport system for testing and debugging. This utility also permits the use of scripts that are created using a standard text editor to move the vehicles within the transport system.

NOTICE	The NCHost TCP Interface Utility is only for basic vehicle motion testing and demonstrations of operation. A user-supplied host controller must be used for all actual transport system control within the production environment.

The transport system is a configuration of linear synchronous motors that are placed end-to-end to form long chains or paths. These chains are used to move and position vehicles in a controlled manner at various acceleration/deceleration and velocity profiles while carrying a wide range of payloads with high precision. The transport system consists of the following components at a minimum:

- MagneMover LITE or QuickStick motors.
- Vehicles with magnet arrays.
- Node controllers.
- Power supplies.
- Configuration file defining paths and nodes.
- User-supplied host controller.

Each path in the transport system starts at a node and the motor at that node is connected to a node controller. The node controller connection is either direct when using RS-422 communication or through the transport system network when using Ethernet communication. Any path that a vehicle can exit from must also end in a node that is connected to a node controller. Multiple node controllers can be used for systems with many paths. Each of the components of the transport system must be fully defined in the Node Controller Configuration File to achieve proper operation of the transport system. Once the transport system is fully defined, it can be tested, and its behavior fine-tuned, using Demo Scripts run using the NCHost TCP Interface Utility.

NOTE: Additional testing and refinement may be necessary if any aspect of the transport system changes, such as vehicle length, payload weight, or other physical factors.

The NCHost TCP Interface Utility provides a window with sections for the different functions available. These functions include monitoring the status of transport system functions and sending basic commands to the transport system. Additional functions include selecting scripts for transport system test and fine-tuning control loop parameters. Some functions open additional windows, which can be positioned as desired to monitor transport system operation.

Transport System Components Overview

This section identifies the components of a MagneMotion transport system as shown in Figure 1-1 and described after the figure.

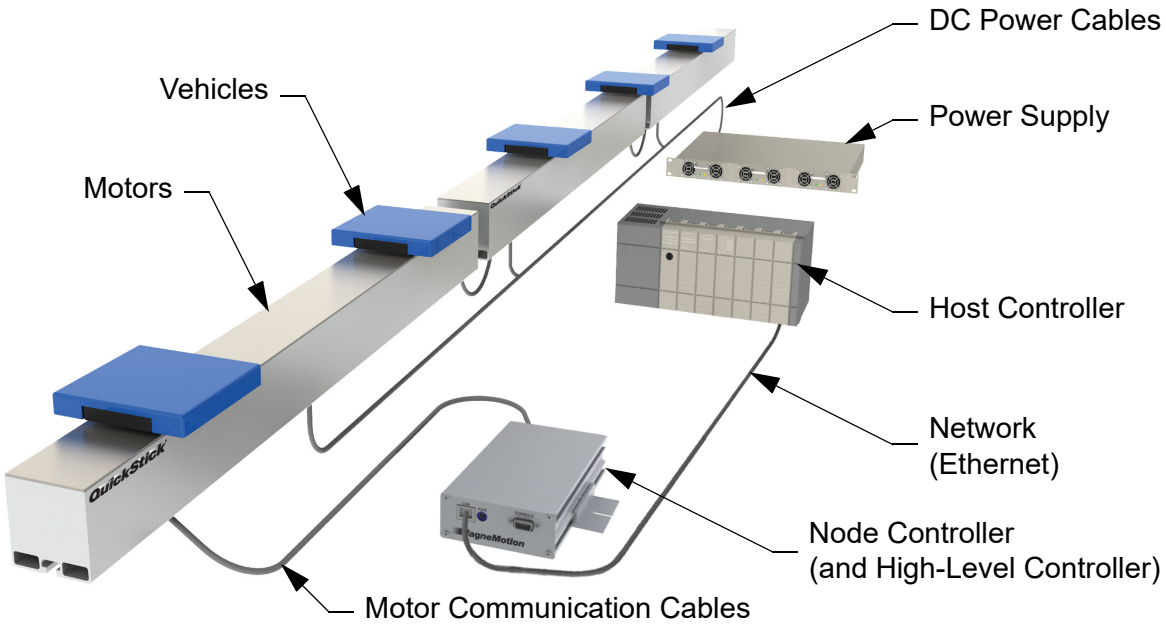


Figure 1-1: Simplified View of the Transport System using RS-422 Communication

- **DC Power Cables and Communication Cables** – Distributes DC power to the motors and carries communications, such as RS-422 or Ethernet, between the components of the transport system.
- **High-Level Controller** (HLC) – Software application that is enabled on one node controller. This application handles all communication with the user-supplied host controller and directs communication as appropriate to individual node controllers.
- **Host Controller** – User-supplied controller for control and monitoring of the MagneMotion transport system using either TCP/IP or EtherNet/IP™ communication.
- **Motor/Stator** – Refers to a MagneMotion linear synchronous motor (LSM).
- **Network** – Ethernet network providing communication (TCP/IP or EtherNet/IP) between the host controller and the HLC (TCP/IP is used between node controllers).
- **Node Controller** (NC) – Coordinates motor operations and communicates with the HLC. Several types of node controllers are available. All node controllers support Ethernet communication with the host controller and the motors, and depending on the model, provide up to 12 RS-422 ports for communication with the motors. Some node controller models also provide Digital I/O and/or Serial I/O for external devices such as switches, E-stops, light stacks, and interlocks.
- **Power Supply** – Provides DC power to the motors.
- **Vehicle with Magnet Array** – Carries a payload through the MagneMotion transport system as directed. The magnet array is mounted to the vehicle facing the motors and interacts with the motors, which move each vehicle independently.

Transport System Software Overview

Several software applications are used to configure, test, and administer a MagneMotion transport system as shown in Figure 1-2 and described after the figure. See *Related Documentation* on page 25 for the reference manuals for these applications.

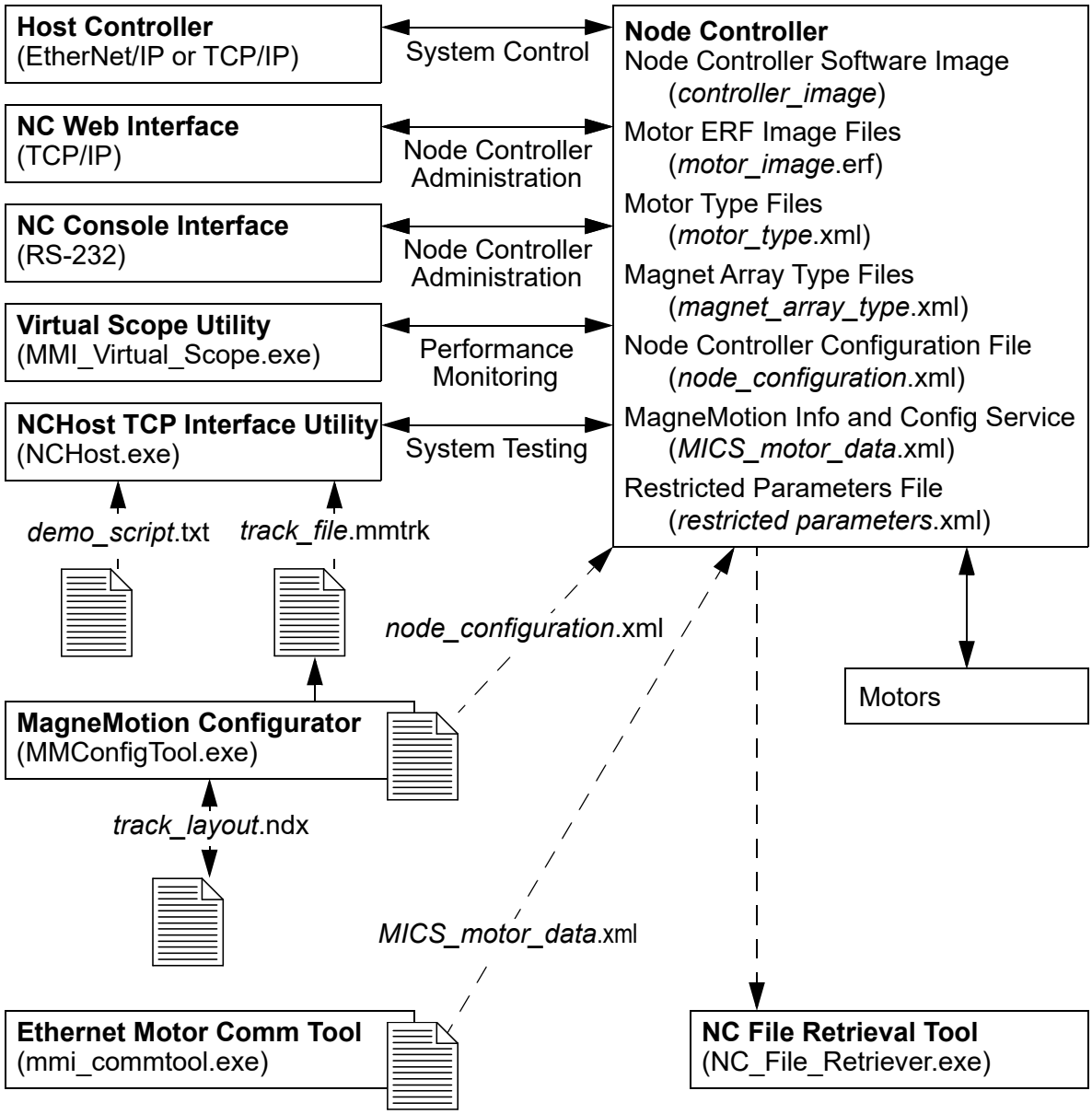


Figure 1-2: Simplified View of Transport System Software Organization

Utilities

- **NC Web Interface** – A web-based software application that is supplied by MagneMotion and resident on the node controllers, for administration of the transport system components.

- **NC Console Interface** – A serial communication software application that is supplied by MagneMotion and resident on the node controllers, for administration of the node controller.
- **Virtual Scope Utility** – A Windows software application that is supplied by MagneMotion to monitor and record the change of motor performance parameters. These parameters are displayed as waveforms to analyze the performance of the motors.
- **NCHost TCP Interface Utility** – A Windows® software application that is supplied by MagneMotion to move vehicles for test or demonstration purposes. This application supports system testing without the host controller to verify that vehicles move correctly before integrating a transport system into a production environment.
- **MagneMotion Configurator Utility** (Configurator) – A Windows software application that is supplied by MagneMotion to create or change the Node Controller Configuration File. This utility is available in both MagneMover LITE and QuickStick versions. The MM LITE version is also used to create or change the Track File and Track Layout File for MagneMover LITE transport systems.
- **Ethernet Motor Commissioning Tool** – A Windows software application that is supplied by MagneMotion to create and edit MagneMotion Information and Configuration Service (MICS) files.
- **NC File Retrieval Tool** – A Windows software application that is supplied by MagneMotion to download configuration and operation files from the specified HLC and all node controllers in the transport system.

File Types

- **Node Controller Software Image File** (IMG file) – The software file for the node controllers (*controller_image*), includes the node controller and HLC applications. The Node Controller Software Image file is uploaded to all node controllers in the transport system.
- **Motor ERF Image Files** (ERF file) – The software files for the MagneMotion motors (*motor_image*.erf). The Motor ERF Image files are uploaded to all node controllers in the transport system and then programmed into all motors.
- **Motor Type Files** – XML files (*motor_type*.xml) that contain basic information about the specific MagneMotion motor types being used. The Motor Type files are uploaded to all node controllers in the transport system.
- **Magnet Array Type File** – An XML file (*magnet_array_type*.xml) that contains basic information about the specific MagneMotion magnet array type that is used on the vehicles in the transport system. The Magnet Array Type file is uploaded to all node controllers in the transport system.
- **Node Controller Configuration Files** (Configuration file) – An XML file (*node_configuration*.xml) that contains all parameters for the components in the transport system. Multiple Node Controller Configuration Files can be uploaded to all node controllers in the transport system, but only one is active.

- **MagneMotion Information and Configuration Service (MICS) File** – An XML file (*MICS_motor_data*.xml) that contains the network topology parameters for the transport system when using Ethernet communication with the motors. The file includes the MAC address of each motor and the location of each motor on a path. The MICS file is uploaded to all node controllers in the transport system.
- **Restricted Parameters File** – An XML file (*restricted_parameters*.xml) that provides access to restricted configuration elements for specific transport systems. The Restricted Parameters file is uploaded to the HLC. For the development of a custom Restricted Parameters file for a specific transport system, see *Rockwell Automation Support* on page 264.
- **Demo Script** – A text file (*demo_script*.txt) uploaded to the NCHost TCP Interface Utility to move vehicles on the transport system for test or demonstration purposes.
- **Track File** – A text file (*track_file*.mmtrk) that contains graphical path and motor information about the transport system. The Track file is used by the NCHost TCP Interface Utility to provide a graphical representation of the transport system to monitor system operation. The Track file is created for MagneMover LITE transport systems using the MagneMover LITE Configurator. See *Rockwell Automation Support* on page 264 for questions regarding the development of Track files for QuickStick transport systems.
- **Track Layout File** – An XML file (*track_layout*.ndx) that contains the parameters for the graphical representation of a MagneMover LITE transport system. The Track Layout file is used by the Configurator to generate the Node Controller Configuration File and the Track file for MagneMover LITE systems.

NOTICE	Modifications to the Image or Type files could cause improper operation of the transport system.

Getting Started with the NCHost TCP Interface Utility

Use this manual as a guide and reference when using the NCHost TCP Interface Utility. Follow the steps in this section to get the entire transport system operational quickly with the aid of the other MagneMotion manuals (see *Related Documentation* on page 25).

NOTE: Make sure that all components and complete design specifications, including the physical layout of the transport system, are available before starting to install or test the transport system.

To get started quickly with the transport system:

1. Download the software for the appropriate MagneMotion transport system and the MagneMotion Utilities from rok.auto/pcdc.

   NOTE: The minimum requirements for running MagneMotion software applications are a general-purpose computer running Microsoft® Windows® 7 with .NET 4.0. An Ethernet port (web interface) and an optional RS-232 port (console interface) are required to connect to the node controllers.

2. Install the components of the MagneMotion transport system as described in the *Node Controller Hardware User Manual*, MMI-UM013, and either the *MagneMover LITE User Manual*, MMI-UM002, the *QuickStick 100 User Manual*, MMI-UM006, or the *QuickStick HT User Manual*, MMI-UM007.

3. Install the MagneMotion Configurator on a computer for user access (see either the *MagneMover LITE Configurator User Manual*, MMI-UM008, or the *QuickStick Configurator User Manual*, MMI-UM009*).*

   A. For MM LITE™ systems, define the motors and paths and their relationships in the transport system graphically to create the Track Layout File (*track_layout*.ndx) and the Track File (*track_file*.mmtrk).

   B. For all transport systems, create the Node Controller Configuration File (*node_configuration*.xml) to define the components and operating parameters of the transport system.

4. Set the IP address for each node controller and specify the node controller to be used as the high-level controller (see the *Node Controller Interface User Manual*, MMI-UM001).

5. Upload the configuration, image, and type files to each node controller using the node controller web interface (see the *Node Controller Interface User Manual*, MMI-UM001). Once configured, the node controllers can be used to simulate the transport system.

6. When using motors with Ethernet communication, create the MICS file and provision the motors (see either the *MagneMover LITE User Manual*, MMI-UM002, or the

QuickStick HT User Manual, MMI-UM007, and the *Node Controller Interface User Manual*, MMI-UM001).

7. Use the web interface to program the motors with the Motor ERF Image Files (see the *Node Controller Interface User Manual*, MMI-UM001, and *Reset Paths* on page 64).

8. Test and debug the transport system by using the NCHost TCP Interface Utility and Demo Scripts as described in the following sections. These scripts provide an easy method to verify proper operation and make adjustments such as refining the control loop tuning.

   NOTE: The NCHost TCP Interface Utility is for test and verification trials only. The host controller must be used to control the transport system after verification of functionality.

   A. *Running the NCHost TCP Interface Utility* on page 39.

   B. *Connecting to the Transport System* on page 42.

   C. *Testing the Transport System – Path Commands* on page 63.

   D. *Testing the Transport System – Vehicle Commands* on page 72.

   E. *Testing the Transport System – Demo Scripts* on page 110.

9. Configure the host controller for transport system control as required to meet the material movement needs of the facility where the system is installed. When using TCP/IP communication, see the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003. When using EtherNet/IP communication, see the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004.

Using the NCHost TCP Interface Utility 2

Overview

This chapter describes how to use the NCHost TCP Interface Utility to startup and test the transport system. After starting the transport system, vehicles can be moved individually or by using a script file. To create the scripts to automate simple test procedures, see *Demo Scripts* on page 113.

NOTE: Use the NCHost TCP Interface Utility only for basic vehicle motion testing. A user-supplied host controller must be used for all actual transport system control within the production environment.

Included in this chapter are instructions for:

- Installing the NCHost TCP Interface Utility.
- Running the NCHost TCP Interface Utility.
- Connecting to the transport system.
- Monitoring the transport system status.
- Transport system operation, including startup and reset, moving vehicles to positions and stations, deleting vehicles, and entering or exiting through a Terminus Node.
- Refining the control loop.
- Monitoring vehicle motion.
- Running Demo Scripts.
- Exercising node controller digital I/O.

Installing the NCHost TCP Interface Utility

This section describes how to install the MagneMotion® NCHost TCP Interface Utility that was supplied with the transport system software. During the installation process, the computer is checked to make sure that all required software components are installed on the computer. The minimum software requirements for installing and running the NCHost TCP Interface Utility are:

- Microsoft® Windows® 7, SP1.
- Microsoft .NET 4.0.

1. Access the folder where the software was copied and run the NCHost.msi file.

   NOTE: The permissions and security settings on the computer may need to be adjusted to allow this installation.

   *The **MagneMotion NCHost TCP Interface Utility** setup wizard is opened as shown in Figure 2-1. If Microsoft .NET 4.0 is not installed on the computer where the NCHost TCP Interface Utility is being installed, an alert is displayed instead.*

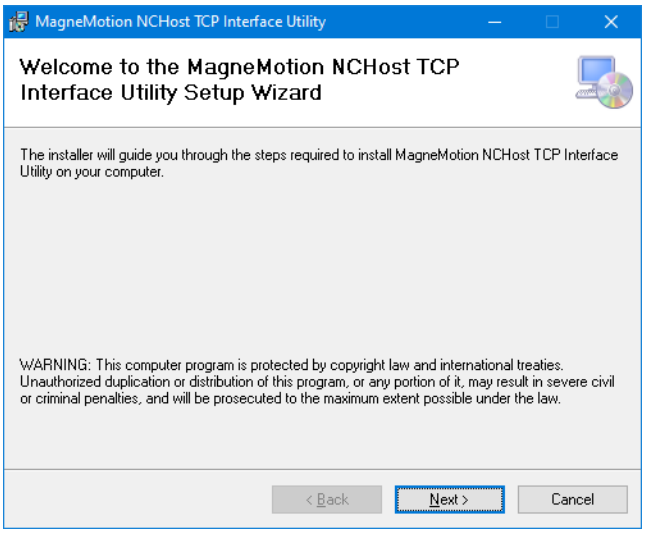


Figure 2-1: MagneMotion NCHost TCP Interface Utility Installer

   A. If the .NET Framework alert is displayed, follow the directions to install Microsoft .NET 4.0.

   A. Once .NET 4.0 is installed, restart the installation process from Step 1.

2. Select **Next**.

   *The **MagneMotion NCHost TCP Interface Utility** setup wizard prompts for an installation path and usage permissions as shown in Figure 2-2.*

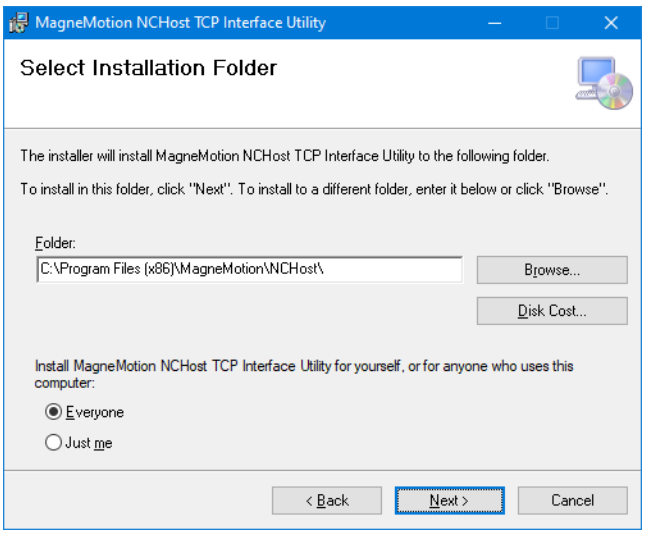


Figure 2-2: MagneMotion NCHost TCP Interface Utility Installation Setup

3. Accept the defaults and select **Next**.

   *The **MagneMotion NCHost TCP Interface Utility** setup wizard prompts for confirmation of the installation as shown in Figure 2-3.*

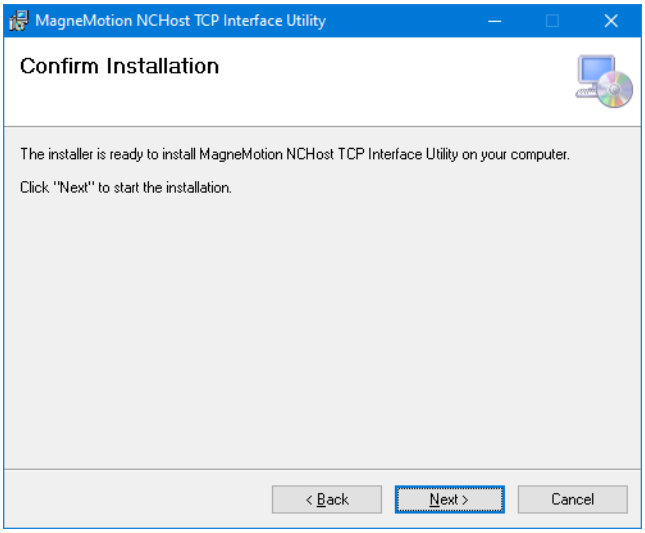


Figure 2-3: MagneMotion NCHost TCP Interface Utility Installation Confirmation

4. Select **Next**.

5. Adjust the security settings for this computer to allow this installation as necessary.

   *The **MagneMotion NCHost TCP Interface Utility** setup wizard installs the software as shown in Figure 2-4 and places an Icon on the desktop for access as shown in Figure 2-6.*

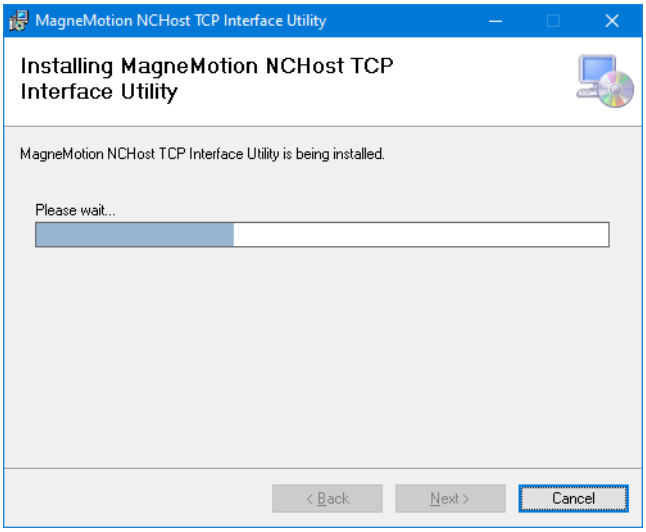


Figure 2-4: MagneMotion NCHost TCP Interface Utility Installation Progress

Once the installation is complete, as shown in Figure 2-5, the MagneMotion NCHost TCP Interface Utility can be used as described in Running the NCHost TCP Interface Utility on page 39.

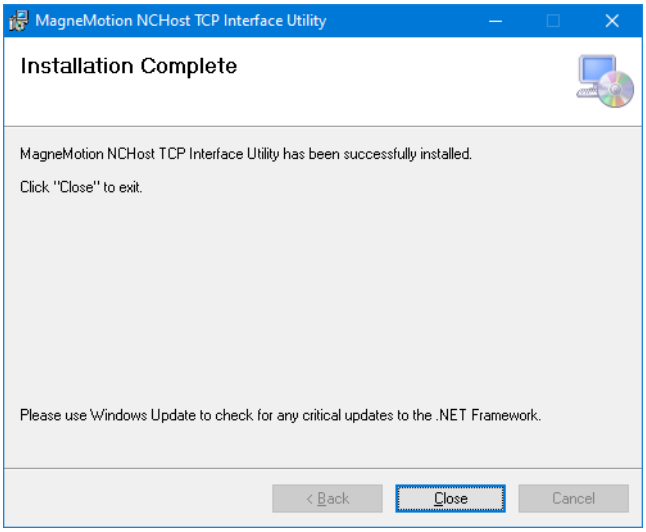


Figure 2-5: MagneMotion NCHost TCP Interface Utility Installation Complete

6. Select **Close**.

Running the NCHost TCP Interface Utility

This section describes how to start and stop the NCHost TCP Interface Utility to test and verify proper operation of a MagneMotion transport system.

1. Select the **MagneMotion NCHost TCP Interface Utility** icon, which is shown in Figure 2-6, on the desktop.

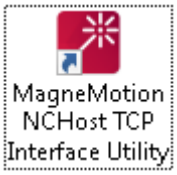


Figure 2-6: MagneMotion NCHost TCP Interface Utility Icon

NOTE: An alternative method is to start the program from the Windows **Start** icon; select **All Programs**, select **MagneMotion** from the list, then select **MagneMotion NCHost TCP Interface Utility.**

*The **MagneMotion NCHost TCP Interface Utility** window opens and displays the **Settings** panel, **Vehicles** pane, and **Diagnostic Output** pane as shown in Figure 2-8. All functions except the **Host Interface** section in the **Settings** panel are disabled.*

*The first time the NCHost TCP Interface Utility is started, the **Select System Type** dialog, which is shown in Figure 2-7, is displayed. This configures the user interface to display only those functions appropriate to the selected motor type. This selection can be changed at any time through the **Select System Type** option on the **View** menu.*

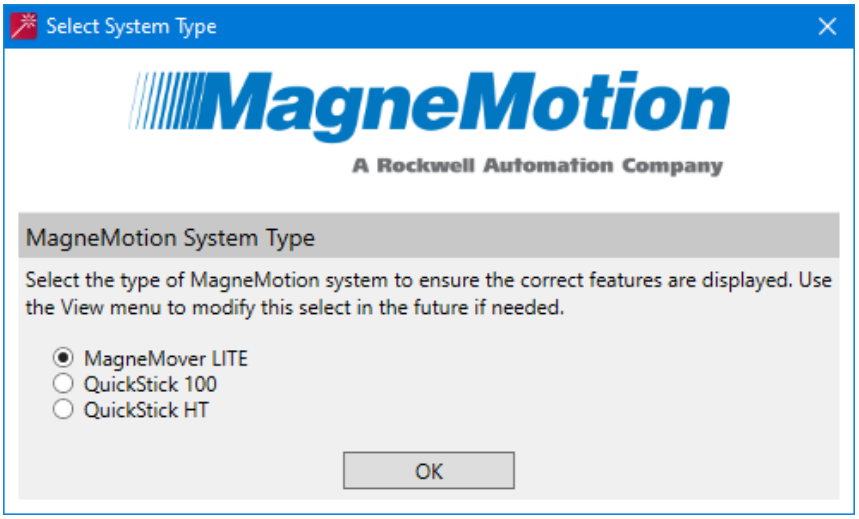


Figure 2-7: Select System Type

2. Select the type of MagneMotion system this installation of NCHost is being used with (this can always be changed from the **View** menu when NCHost is running) and select **OK**.

The dialog closes and the user interface is configured for the selected motor type.

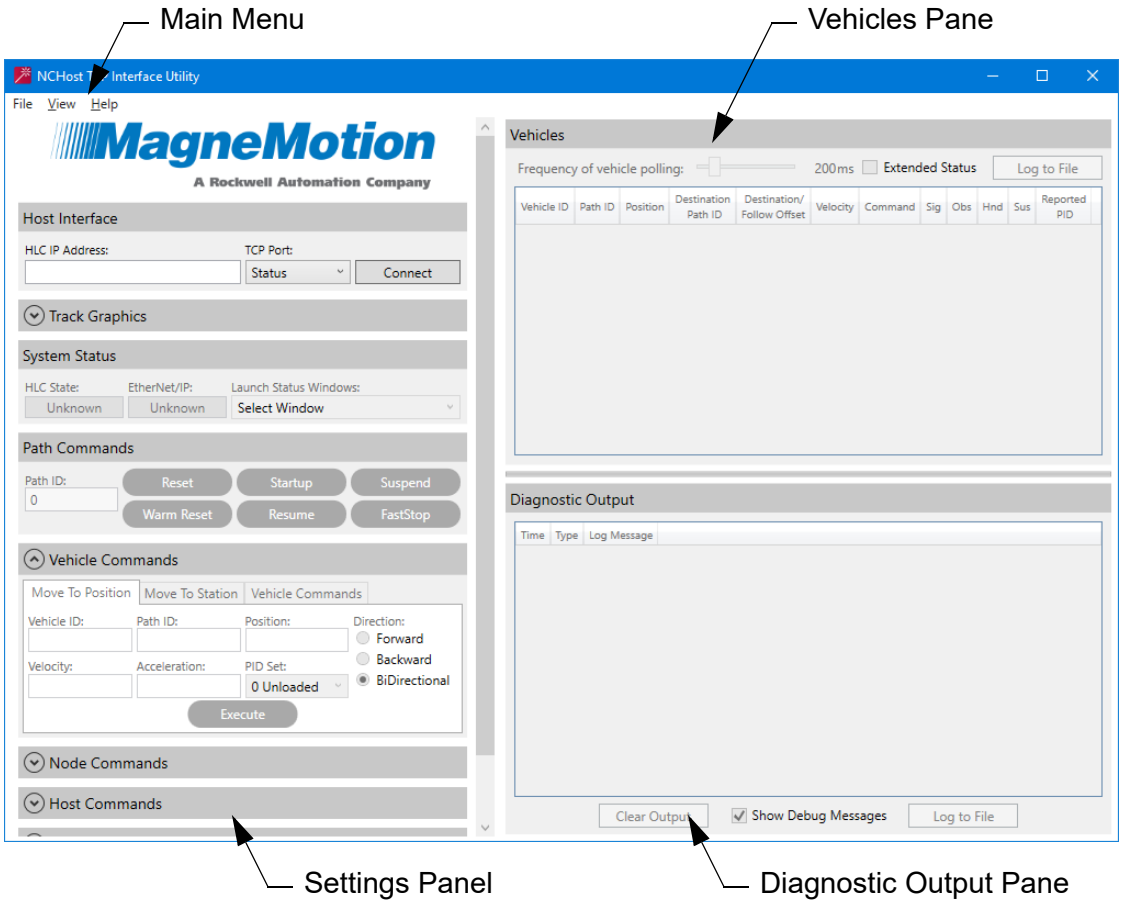


Figure 2-8: The NCHost TCP Interface Utility Window

3. Adjust or resize the window as necessary.

   - Move the bar that separates the **Vehicles** pane and the **Diagnostic Output** pane in the window.

     NOTE: Moving the cursor over the separator bar between the panes causes the cursor to change to the Vertical Resize cursor (⇳). Click-and-hold the left mouse button once the cursor changes to move the bar.

   - Expand or collapse those sections within the **Settings** panel that have a section control (⌃ or ⌄) to display or hide those functions.
   - Use the scroll bars to see additional text or other sections in the **Settings** panel and panes.
   - Hovering over most active controls displays a tool tip with information about that control.

4. Select the viewing options from the **View** menu, see Figure 2-9. See *Main Window* on page 142 for detailed descriptions of all items. These selections determine the options that are displayed in the user interface.

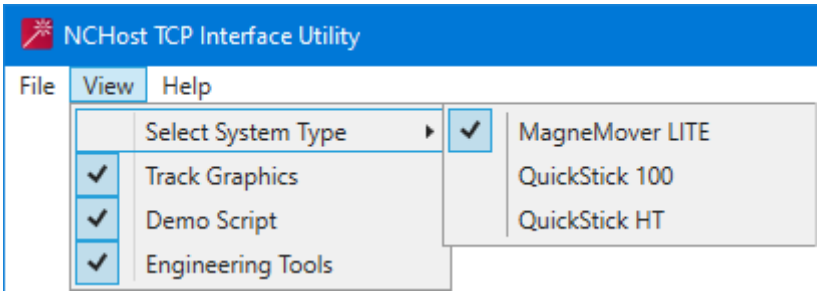


Figure 2-9: View Menu

5. To connect to the transport system and enable all functions to begin testing see *Connecting to the Transport System* on page 42.

Connecting to the Transport System

This section describes how to connect the NCHost TCP Interface Utility to the transport system to test operation and/or monitor system status. Both Control and Status sessions are configured through the **Host Interface** section (see Figure 2-10) by selecting the appropriate TCP Port. See *Host Interface* on page 146 for detailed descriptions of all items. Both the HLC and the computer running the NCHost TCP Interface Utility must be on the same network.

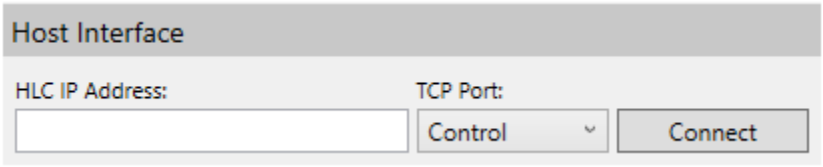


Figure 2-10: Host Interface Section on Startup

Host Control Session

The Host Control Session provides a connection to the HLC that allows issuing commands to monitor and control transport system operation for testing.

1. To control transport system operation for testing, select **Control** from the **TCP Port** menu in the **Host Interface** section (see Figure 2-10).

   NOTE: Only one Host Control session can be connected to an HLC at a time. If a second Host Control session is connected, the first Host Control session is automatically disconnected. The disconnect action is described in *Stop the NCHost TCP Interface Utility* on page 45.

2. Enter the IP address of the HLC in the **HLC IP address** text field and select **Connect**.

   If a Control connection is made to the HLC specified, all control and status functions of the NCHost TCP Interface Utility are enabled. The low-order byte of the address for the HLC is displayed in the title bar and the status of the connections are displayed in the ***System Status*** *section as described.*

   If the connection to the host controller is TCP/IP, the ***EtherNet/IP*** *field displays Not Conf (see Figure 2-11). If the connection to the host controller is EtherNet/IP™, the* ***EtherNet/IP*** *field displays the status of the EtherNet/IP connection between the host controller and the HLC (see Figure 2-12).*

   If a connection is made to the HLC specified, the ***Diagnostic Output*** *section shows information about the connection. If no connection is made to the HLC specified, a diagnostic message that describes the error is displayed in the* ***Diagnostic Output*** *section as shown in Figure 2-13.*

3. Use the other functions to initialize the transport system, move vehicles, and monitor transport system status, as described in this chapter.

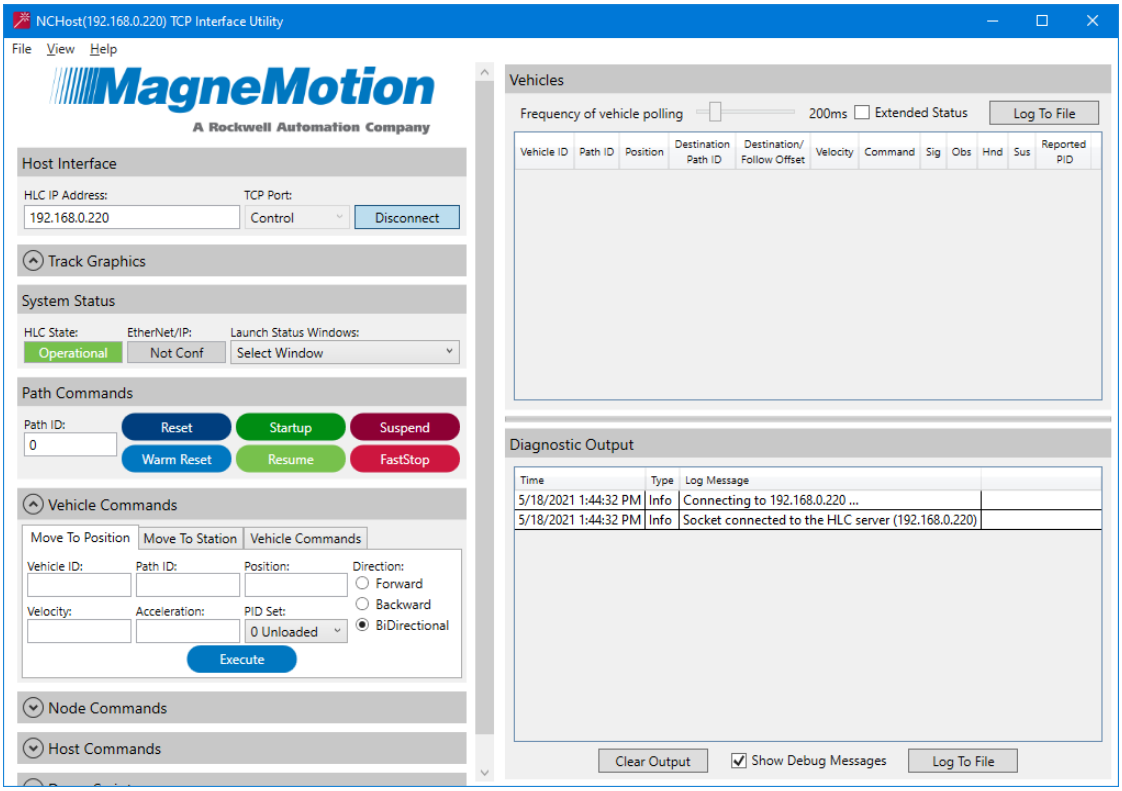


Figure 2-11: System Status Section Showing HLC Connected

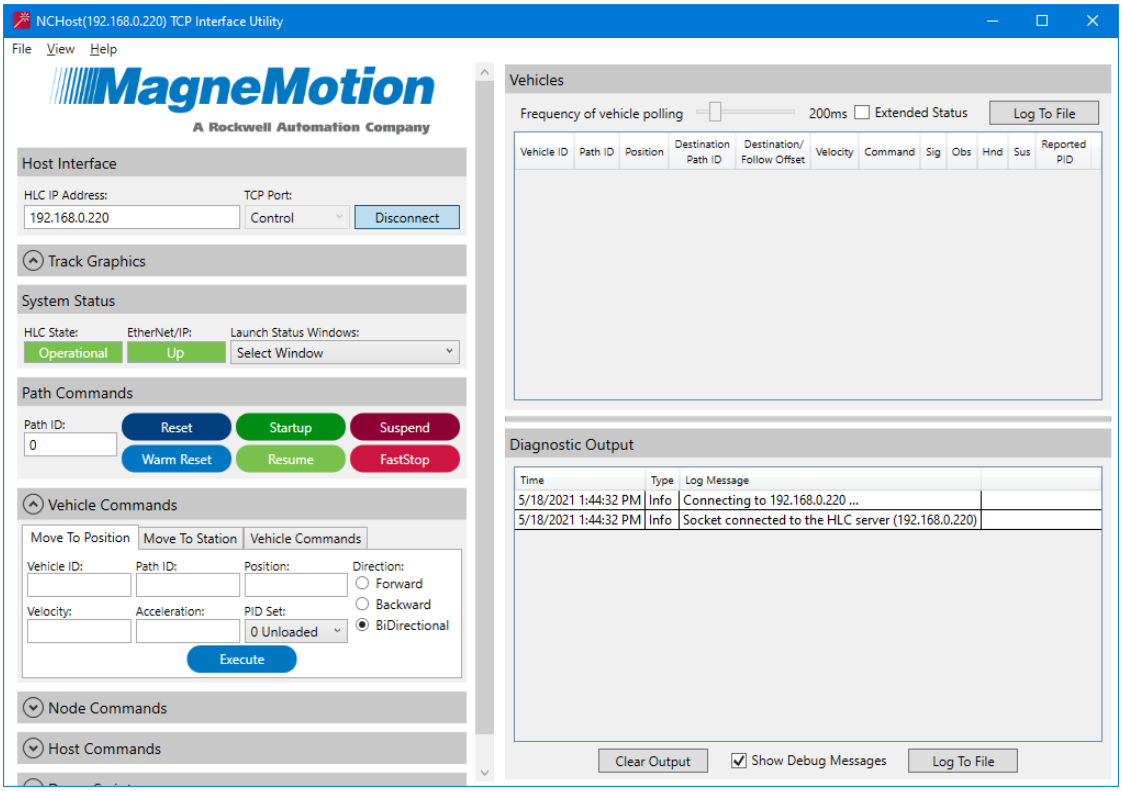


Figure 2-12: System Status Section Showing HLC with Active EtherNet/IP Connection

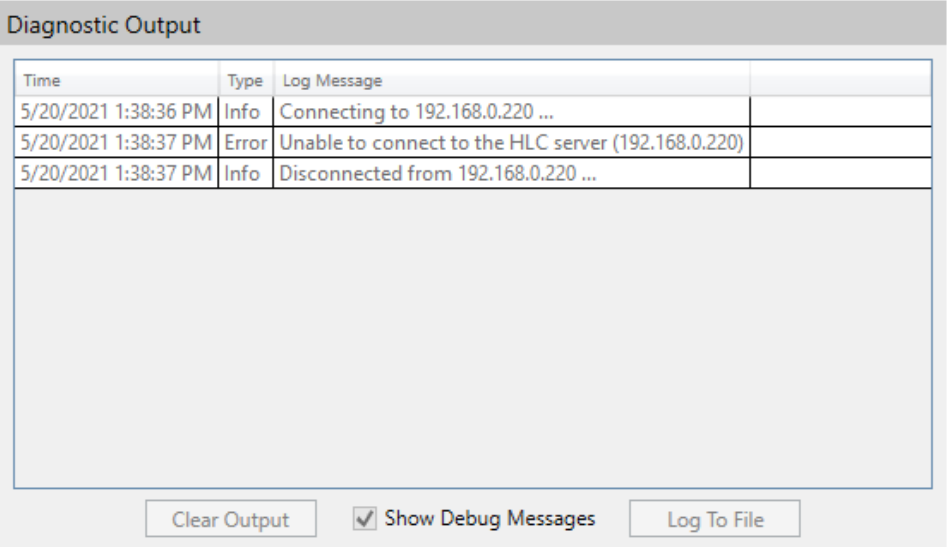


Figure 2-13: Connection Error

Host Status Session

The Host Status Session provides a connection to the HLC that allows real-time status monitoring of the transport system.

1. To monitor transport system operation, select **Status** from the **TCP Port** menu in the **Host Interface** section (see Figure 2-10).

   NOTE: Up to four Host Status sessions can be connected to an HLC while the Host Control session is connected. If additional Host Status sessions are connected to the HLC, the first Host Status sessions are disconnected in the order they were connected.

2. Enter the IP address of the HLC in the **HLC IP address** text field and select **Connect**.

   *If a Status connection is made to the HLC specified, all status functions of the NCHost TCP Interface Utility are enabled and available through the **Launch Status Windows** menu. The low-order byte of the address for the HLC is displayed in the title bar and the status of the connections are displayed in the **System Status** section as described.*

   *If the connection to the host controller is TCP/IP, the **EtherNet/IP** field displays Not Conf (see Figure 2-11). If the connection to the host controller is EtherNet/IP™, the **EtherNet/IP** field displays the status of the EtherNet/IP connection between the host controller and the HLC (see Figure 2-12).*

   *If a connection is made to the HLC specified, the **Diagnostic Output** section shows information about the connection. If no connection is made to the HLC specified, a diagnostic message that describes the error is displayed in the **Diagnostic Output** section as shown in Figure 2-13.*

3. Use the other functions to monitor transport system status, as described in this chapter.

Stop the NCHost TCP Interface Utility

1. Verify that any tasks that have been started have completed or have been stopped and all vehicles are at known positions.

2. Suspend motion on all paths. Use the **Path Commands** section to suspend all motion. Enter zero in the **Path ID** field then select **Suspend**.

3. Once all motion has stopped, reset all paths to clear the vehicle records for all motors. Use the **Path Commands** section to reset all the motors. Enter zero in the **Path ID** field then select **Reset**.

4. Disconnect from the high-level controller (HLC) for the transport system by selecting **Disconnect** in the **Host Interface** section of the **Settings** panel.

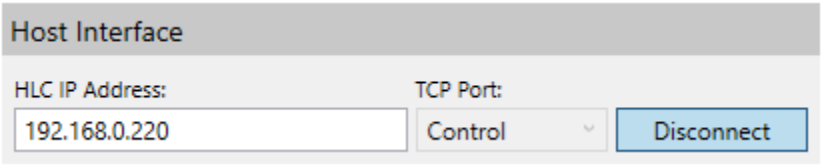


Figure 2-14: Host Interface Section When Running

*All functions except the **Host Interface** section in the **Settings** panel are disabled.*

- *If the **PC Host Disconnect Action** in the Global Settings of the Node Controller Configuration File is set to **None**, any incomplete motion will complete.*
- *If the **PC Host Disconnect Action** in the Global Settings of the Node Controller Configuration File is set to **Suspend**, motion on all paths is suspended and any incomplete motion will stop.*
- *If the **PC Host Disconnect Action** in the Global Settings of the Node Controller Configuration File is set to **FastStop**, motion on all paths is fast stopped and any incomplete motion will come to an abrupt stop.*

5. Stop the NCHost TCP Interface Utility by selecting **Exit** from the **File** menu or by selecting the **X** in the upper-right corner to close the window.

Monitoring the Transport System – Status Information

This section describes how to monitor the basic components of the transport system. Transport system status is displayed through the windows that are accessed through the **Launch Status Windows** menu in the **System Status** section (see Figure 2-15). See *System Status* on page 150 for detailed descriptions of all items. Vehicle status is always displayed in the **Vehicles** pane (see *Vehicles* on page 190 for detailed descriptions of all items). Event and error messages are displayed in the **Diagnostic Output** section (see *Diagnostic Output* on page 195 for detailed descriptions of all items).

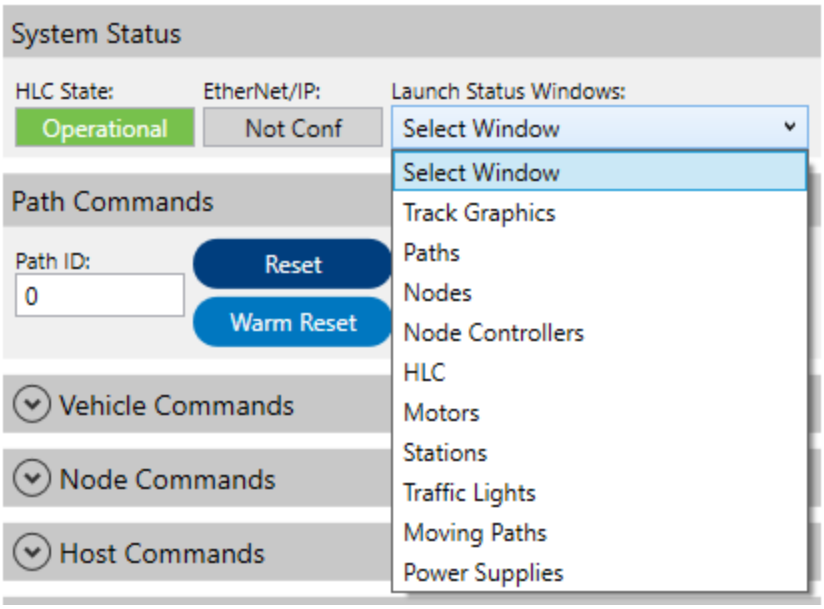


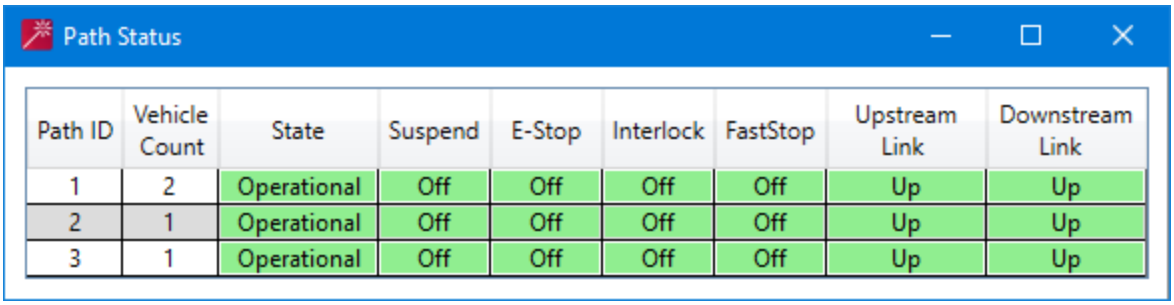
Figure 2-15: Launch Status Windows Menu

Path Status

To monitor path status:

1. In the **System Status** section (see Figure 2-15), select **Paths** from the **Launch Status Windows** menu.

   *The **Path Status** window is displayed as shown in Figure 2-16 (see Path Status Window on page 152 for detailed descriptions of all fields).*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	2	Operational	Off	Off	Off	Off	Up	Up
2	1	Operational	Off	Off	Off	Off	Up	Up
3	1	Operational	Off	Off	Off	Off	Up	Up

Figure 2-16: Path Status Window

2. Move the status window to a convenient location.

3. To issue path-related commands, see *Testing the Transport System – Path Commands on page 63*.

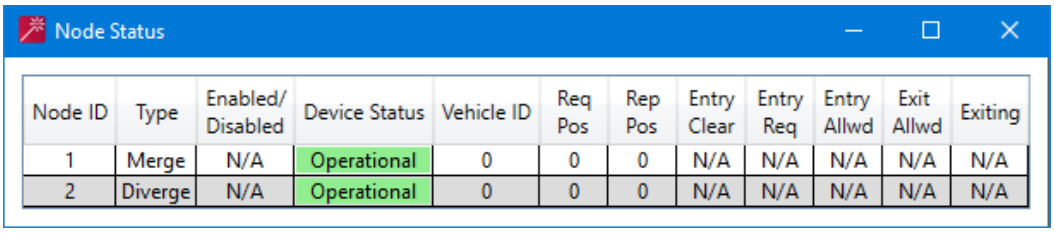
4. When finished using the **Path Status** window, select the **X** in the upper-right corner to close the window.

Node Status

To monitor node status:

1. In the **System Status** section (see Figure 2-15), select **Nodes** from the **Launch Status Windows** menu.

   *The **Node Status** window is displayed as shown in Figure 2-17 (see Node Status Window on page 154 for detailed descriptions of all fields).*



Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-17: Node Status Window

2. Move the status window to a convenient location.

3. To issue node-related commands, see *Testing the Transport System – Node Commands on page 85*.

4. When finished using the **Node Status** window, select the **X** in the upper-right corner to close the window.

Node Controller Status

To monitor node controller status:

1. In the **System Status** section (see Figure 2-15), select **Node Controllers** from the **Launch Status Windows** menu.

   *The **Node Controller Status** window is displayed as shown in Figure 2-18 (see Node Controller Status Window on page 156 for detailed descriptions of all fields).*

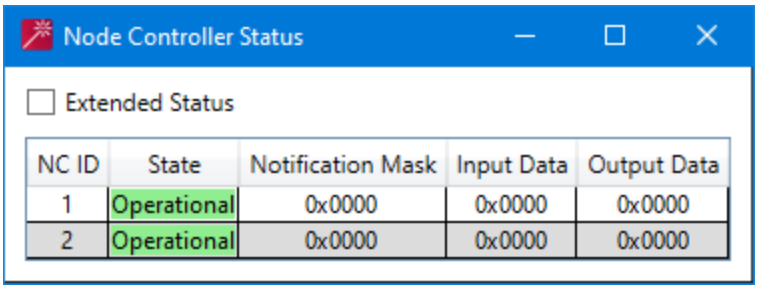


Figure 2-18: Node Controller Status Window

2. Move the status window to a convenient location.

3. Select the **Extended Status** checkbox at the top of the window to view the software version and configuration information, see Figure 2-19.

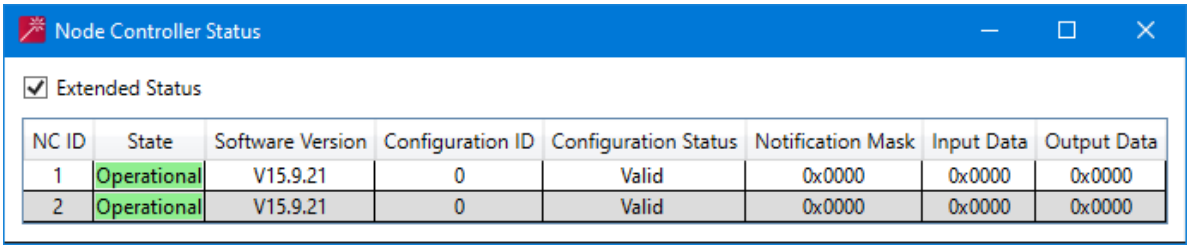


Figure 2-19: Node Controller Status Window Showing Extended Status

4. To issue node controller-related commands, see *Testing the Transport System – Host Commands* on page 101.

5. When finished using the **Node Controller Status** window, select the **X** in the upper-right corner to close the window.

HLC Status

To monitor high-level controller (HLC) status.

1. In the **System Status** section (see Figure 2-15), select **HLC** from the **Launch Status Windows** menu.

   *The **HLC Status** window is displayed as shown in Figure 2-20 (see HLC Status Window on page 158 for detailed descriptions of all fields).*

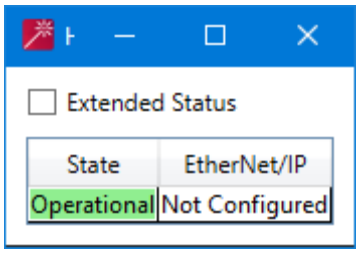


Figure 2-20: HLC Status Window

2. Move the status window to a convenient location.

3. Select the **Extended Status** checkbox at the top of the window to view the software version and configuration information, see Figure 2-21.

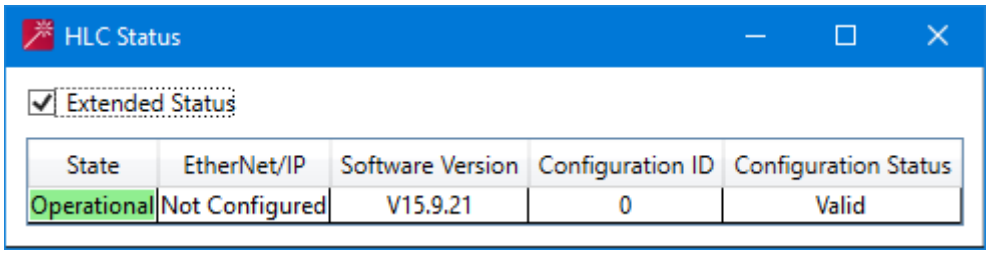


Figure 2-21: HLC Status Window Showing Extended Status

4. When finished using the **HLC Status** window, select the **X** in the upper-right corner to close the window.

Motor Status

To monitor motor status:

1. In the **System Status** section (see Figure 2-15), select **Motors** from the **Launch Status Windows** menu.

   *The **Motor Status** window is displayed as shown in Figure 2-22 (see Motor Status Window on page 160 for detailed descriptions of all fields).*

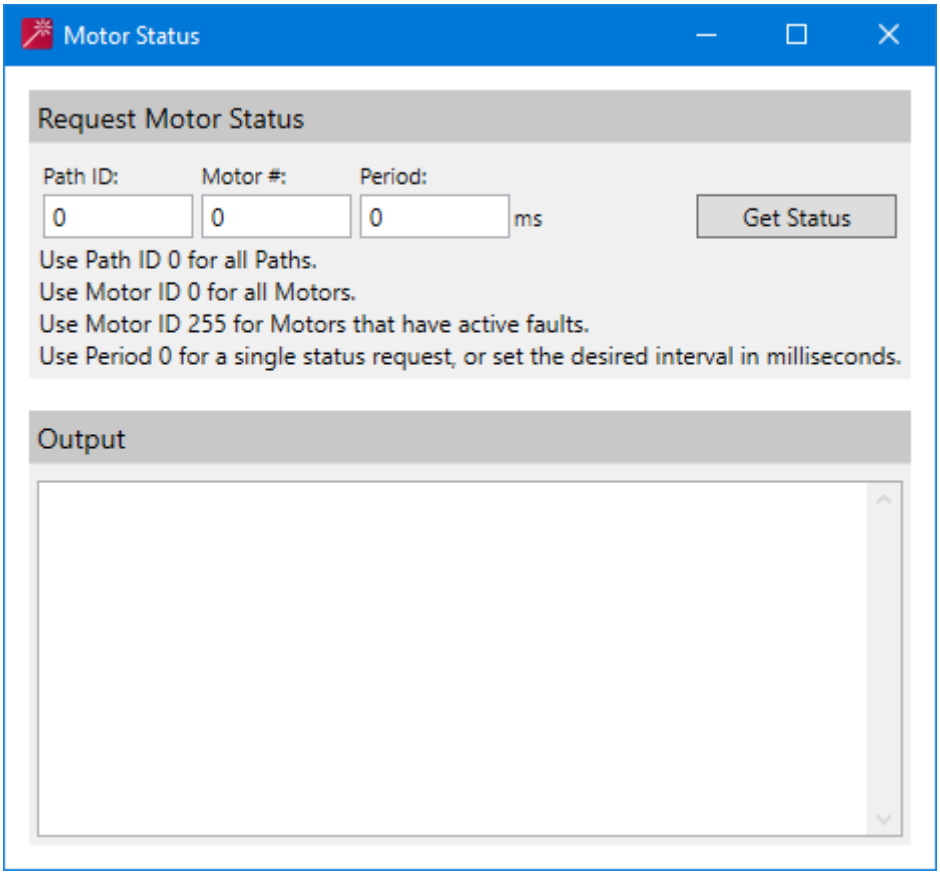


Figure 2-22: Motor Status Window

2. Move the status window to a convenient location.

3. In the **Motor Status** window, specify the **Path ID** of the path where the motors are located.

   NOTE: A **Path ID** of 0 requests the status of motors on all paths. To request the status of motors on a specific path, enter its ID in the **Path ID** field.

4. Specify the **Motors** to be monitored.

   NOTE: A **Motor #** of 0 requests the motor status for all motors on the specified paths. To request the motor status for a specific motor, enter its ID in the **Motor #** field. To request the motor status of only those motors that have faults, enter 255 in the **Motor #** field.

5. Specify the status update **Period**.

   NOTE: A **Period** of 0 issues one status request. To request continuously updated motor status, enter the update rate in milliseconds in the **Period** field.

6. Select **Get Status**.

   *The requested motor status is displayed in the **Output** field as shown in Figure 2-23. If the **Period** was set to a value other than 0, the **Output** field continuously updates at that interval.*

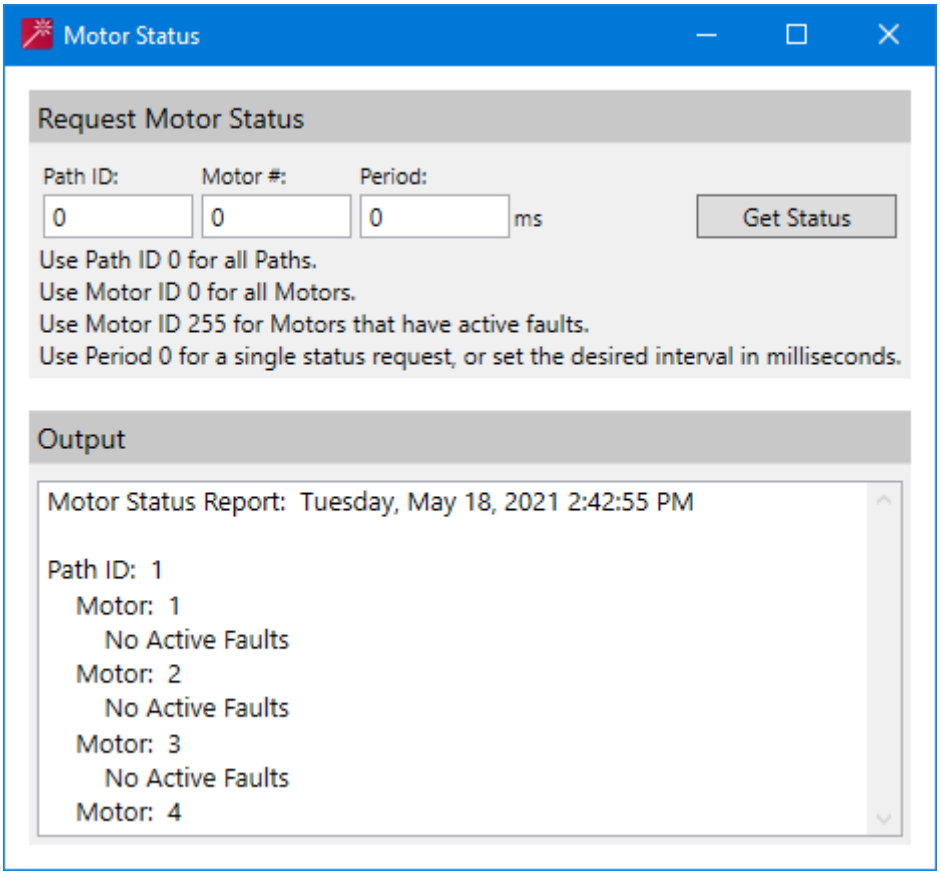


Figure 2-23: Motor Status Window Showing Status

7. To stop automatic updates of the motor status, set the **Period** to 0 and select **Get Status**.

8. When finished using the **Motor Status** window, select the **X** in the upper-right corner to close the window.

Station Status

To monitor station status:

1. In the **System Status** section (see Figure 2-15), select **Stations** from the **Launch Status Windows** menu.

   *The **Station Status** window is displayed as shown in Figure 2-24 (see Station Status Window on page 161 for detailed descriptions of all fields).*

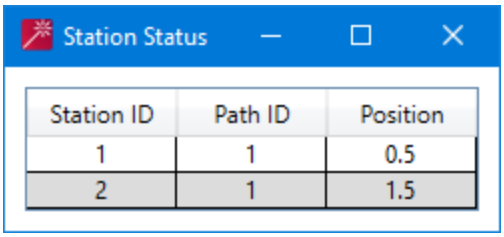


Figure 2-24: Station Status Window

2. Move the status window to a convenient location.

3. To issue station-related commands, see *Move to Station* on page 75 and *Temporary Stations* on page 105.

4. When finished using the **Station Status** window, select the **X** in the upper-right corner to close the window.

Traffic Light Status

To monitor traffic light status:

1. In the **System Status** section (see Figure 2-15), select **Traffic Lights** from the **Launch Status Windows** menu.

   *The **Traffic Light Status** window is displayed as shown in Figure 2-25 (see Traffic Lights Status Window on page 162 for detailed descriptions of all fields).*

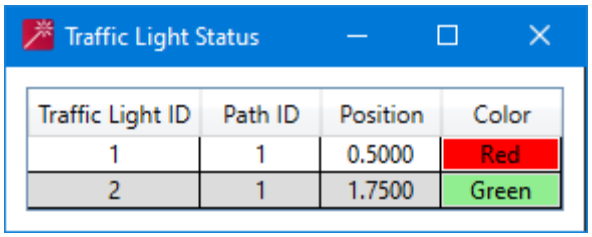


Figure 2-25: Traffic Light Status Window

2. Move the status window to a convenient location.

3. To issue traffic light-related commands, see *Traffic Lights* on page 101.

4. When finished using the **Traffic Light Status** window, select the **X** in the upper-right corner to close the window.

Moving Path Status

To monitor Moving Path Node status:

NOTE: This feature is only available on QuickStick and QSHT configurations.

1. In the **System Status** section (see Figure 2-15), select **Moving Paths** from the **Launch Status Windows** menu.

   *The **Moving Paths Status** window is displayed as shown in Figure 2-26 (see Moving Paths Status Window on page 163 for detailed descriptions of all fields).*

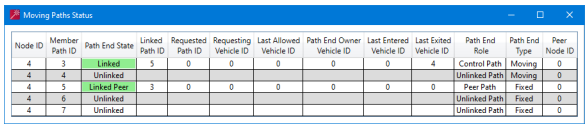


Figure 2-26: Moving Paths Status Window

2. Move the status window to a convenient location.

3. To issue Moving Path related commands, see *Moving Path Node* on page 97.

4. When finished using the **Moving Paths Status** window, select the **X** in the upper-right corner to close the window.

Power Supply Status

To monitor the status of the Kinetix 2198-Pxxx DC-bus power supplies for the QSHT 5700 Inverters:

NOTE: This feature is only available on QSHT systems using QuickStick HT 5700 Inverters with Kinetix 2198-Pxxx DC-bus power supplies.

1. In the **System Status** section (see Figure 2-15), select **Power Supplies** from the **Launch Status Windows** menu.

   *The **Propulsion Power Supply Status** window is displayed as shown in Figure 2-27 (see Propulsion Power Supply Status Window on page 166 for detailed descriptions of all fields).*

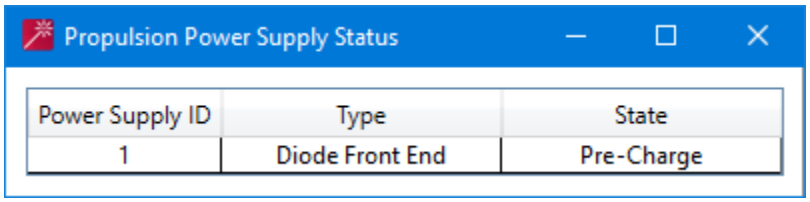


Figure 2-27: Propulsion Power Supply Status Window

2. Move the status window to a convenient location.

3. When finished using the **Power Supplies Status** window, select the **X** in the upper-right corner to close the window.

Vehicle Motion

Vehicle motion on the transport system can be monitored using either the *Track Graphics* window or the *Vehicle Status* pane.

Track Graphics

This feature is typically used when running simulations and during setup to compare the graphic representation of vehicle positions with their actual positions. It can also be used while the NCHost is in Status mode to monitor movement on the track.

NOTE: The **Track Graphics** window can only be used if a Track file for the specific configuration has been created. If a Track file is not available, vehicle motion can be monitored using the **Vehicles** pane, see *Vehicle Status* on page 59.

1. In the **Track Graphics** section (see Figure 2-28), select **Browse**.

   *The **Open** dialog box is displayed.*

Figure 2-28: Track Graphics Section

2. Use the dialog box controls to select the Track File (*track_file*.mmtrk) and select **Open**.

   *The selected file is loaded with the path and name of the file displayed in the Track File field. The **Track Graphics** window is displayed with the title of the Track File in the title bar and the loaded graphic centered in the graphics canvas (see Figure 2-29).*

   Once a Track File is loaded, the **Track Graphics** window can always be reopened by selecting Track Graphics from the Launch Status Windows menu in the System Status section.

3. Move the **Track Graphics** window to a convenient location and adjust its size as necessary (see *Track Graphics* on page 147 for detailed descriptions of all functions).

4. Adjust the display of the track as desired using the **Home**, **Zoom In**, **Zoom Out**, **Center**, and **Rotate** buttons.

5. Use the **Grid**, **Motors**, and **Guideway** checkboxes to specify the elements of the display.

6. Use the **Visible**, **Show IDs**, and **Solid** checkboxes and the **Color Picker** button to adjust the display of the vehicles.

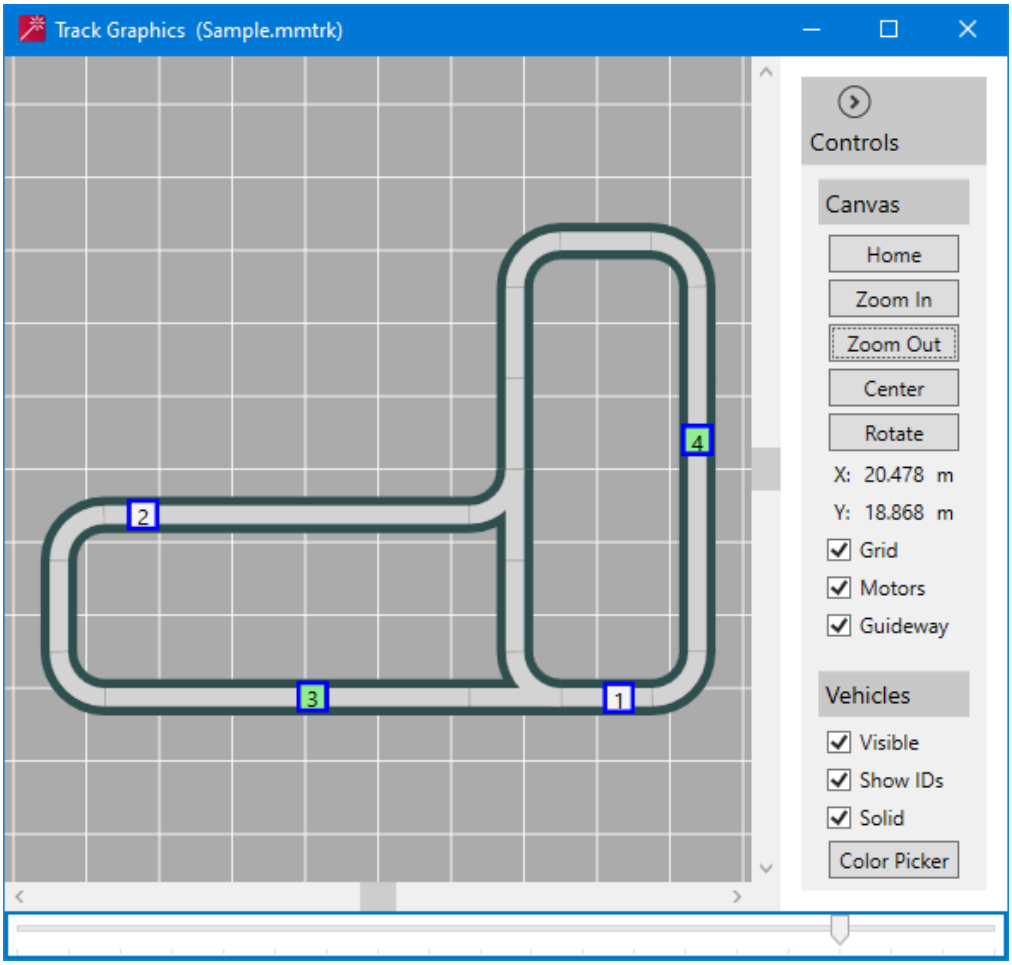


Figure 2-29: Track Graphics Window

7. See *Testing the Transport System – Vehicle Commands* on page 72 to issue vehicle-related commands.

   NOTE: If the graphic is not displayed in the viewing area, it can be repositioned by using the scroll bars or by selecting **Home** or **Center**.

8. When finished using the **Track Graphics** window, select the **X** in the upper-right corner to close the window.

Vehicle Status

The **Vehicles** pane, shown in Figure 2-30, provides a real-time display of the status of all vehicles on the transport system and a method to record that status for review. See *Vehicles* on page 190 for detailed descriptions of all fields.

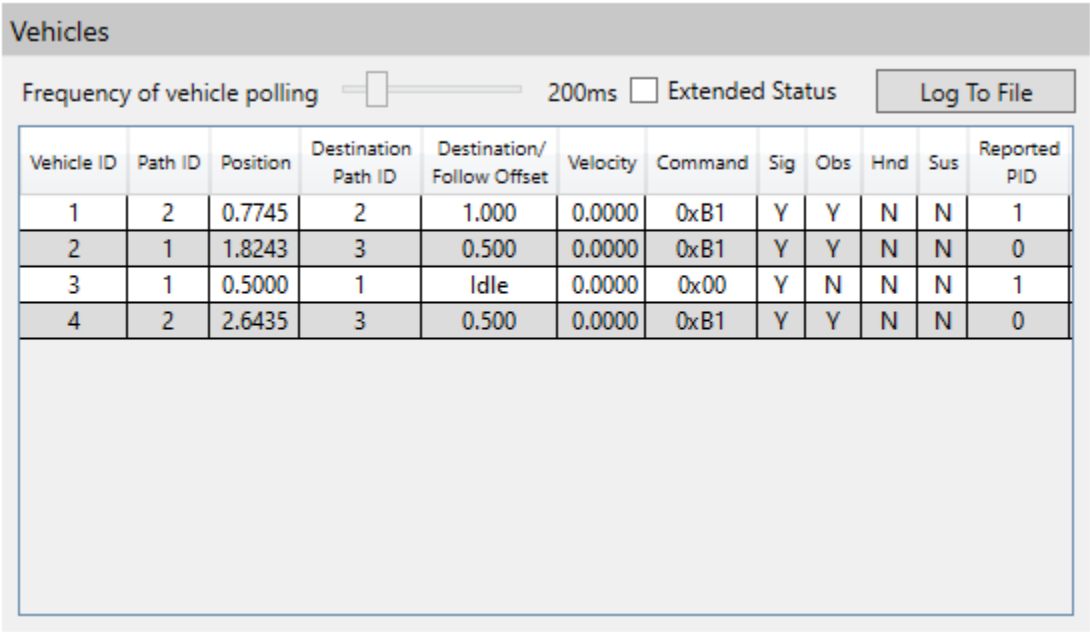


Figure 2-30: Vehicles Pane

1. If necessary, adjust the refresh rate for the data that is shown within the **Vehicles** pane, by sliding the **Frequency of vehicle polling** at the top of the panel. To increase the refresh rate, slide the bar left. To decrease the refresh rate, slide the bar to the right. The selectable refresh rate range is between 20…1000 milliseconds.

2. Select the **Extended Status** checkbox at the top of the pane to view all status columns for all vehicles (see Figure 2-31).

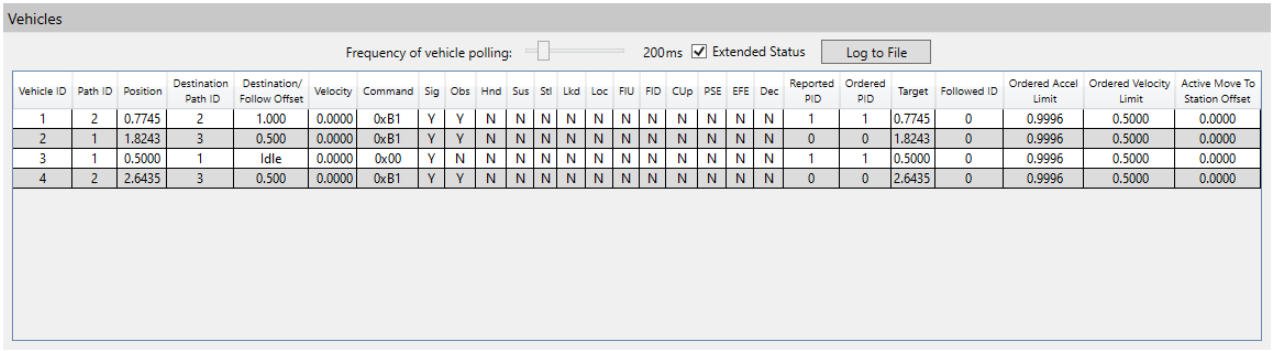


Figure 2-31: Vehicles Pane Showing Extended Status

3. Save the vehicle status information to a log file by selecting **Log To File**. The vehicle status information is added to the log file at the same rate the display is updated.

   *The **Save As** dialog box is displayed.*

   A. Use the dialog box controls to select the location to save the file, enter the name for the log file with the extension ".csv" (for example, vehicle_log_1.csv), and select **Save**.

      *The **Log To File** button changes to **Close Log** and the vehicle status is continuously recorded in the log file.*

      NOTE: Using the name of an existing log file causes the NCHost TCP Interface Utility to prompt for permission to replace it.

   B. When finished working, or when enough data has been captured, select **Close Log** to stop adding vehicle status information to the log file.

Diagnostic Messages

The **Diagnostic Output** pane, shown in Figure 2-32, provides a real-time display of all transport system events and errors and a method to record those messages for review. See *Diagnostic Output* on page 195 for detailed descriptions of all fields.

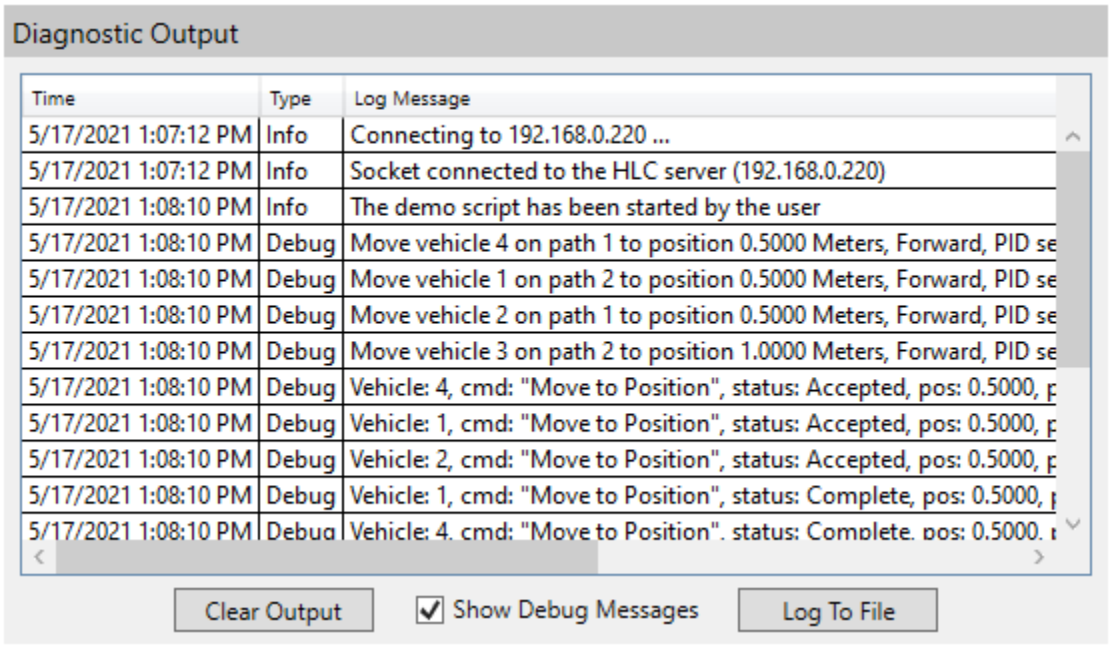


Figure 2-32: Diagnostic Output Pane

1. Define the type of messages to be displayed by selecting or clearing the **Show Debug Messages** checkbox. When selected, the pane displays Debug message along with Error and Info messages (see *Diagnostic Output* on page 195 and *Diagnostic Output Troubleshooting* on page 235).

2. To remove all messages that are currently displayed, select **Clear Output**.

3. Save the messages to a log file if desired by selecting **Log to File**.

   NOTE: Only those messages that are displayed after the log file is opened, and before it is closed, are saved to the log file.

   *The **Save As** dialog box is displayed.*

   A. Use the **Save As** dialog box to select an existing log file or create and name a new log file. When specifying the log file name, specify the file extension as TXT (for example, diagnostic_log_1.txt). Then navigate to the location where the log file is to be stored and select **Save**.

      *The **Log to File** button changes to **Close Log**. Opening the log file is displayed in the Log Viewer and recorded as the first entry in the log file.*

      NOTE: Using the name of an existing log file causes the NCHost TCP Interface Utility to prompt for permission to replace it.

B. When finished working, or when enough data has been captured, select **Close Log** to stop adding diagnostic messages to the log file.

4. To view the contents of the log file, open the TXT log file with a text editor.

Testing the Transport System – Path Commands

This section describes how to select and perform host-related tasks in the transport system. These tasks include; system startup, creating test stations and traffic lights, testing digital I/O operation, selecting managed node controller configurations, and controlling the motor inverters. The examples in this section reference the transport system layout shown in Figure 2-29.

Path Commands for Transport System Operation

Use the **Path Commands** section, which is shown in Figure 2-33, to start or restart the basic components of the transport system during initial startup or when returning it to service. Initialization also identifies and assigns a Vehicle ID to all vehicles on the transport system. The **Path Commands** section is also used to send commands to stop or suspend motion on the transport system during testing.

Figure 2-33: Host Commands Section – Path Commands

NOTE: Motion cannot be started if the E-stop or Interlock is On.

Reset Paths

Use the Reset control to reset motors on one or more specified paths, which clears all vehicle records from the motors and releases control of all vehicles. Paths must be reset when starting a system, after replacing a motor on the path, or after programming the motors on the path.

NOTE: A Reset command must be issued on all paths whenever a new Node Controller Configuration File has been uploaded to the node controllers.

1. Open the **Path Status** window to view the path status during the reset if desired (see *Path Status* on page 47).

2. Open the **Motor Status** window to view the motor status during the reset if desired (see *Motor Status* on page 51).

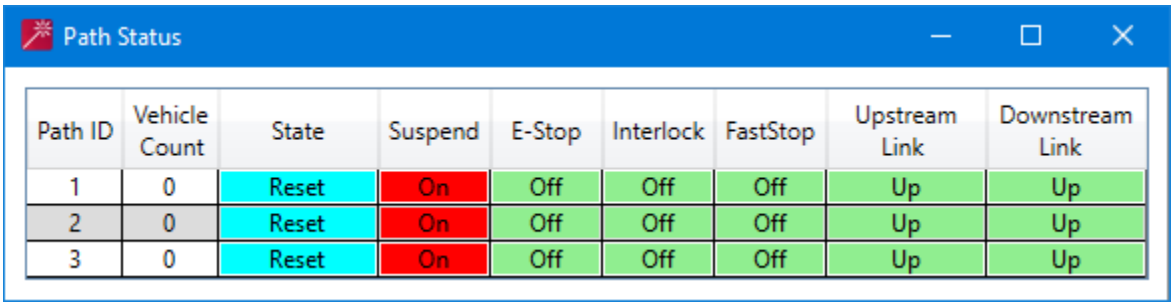
3. Open the **Track Graphics** window to view the graphic of the motors and paths during the reset if desired (see *Track Graphics* on page 57).

4. Use the **Path Commands** section to reset the motors on a specific path. Enter the ID of the path in the **Path ID** field then select **Reset**.

   NOTE: To reset the motors on all paths, enter 0 in the **Path ID** field.

   *All motors on the specified paths are reset, which clears all vehicle records. If the **Path Status** window is open, the **Suspend** indicator changes to **On** and the **State** indicator changes to **Reset** for the specified paths. See Figure 2-34. The downstream link for each path resets and the **Downstream Link** indicator displays **Down** until the status of the link is verified.*

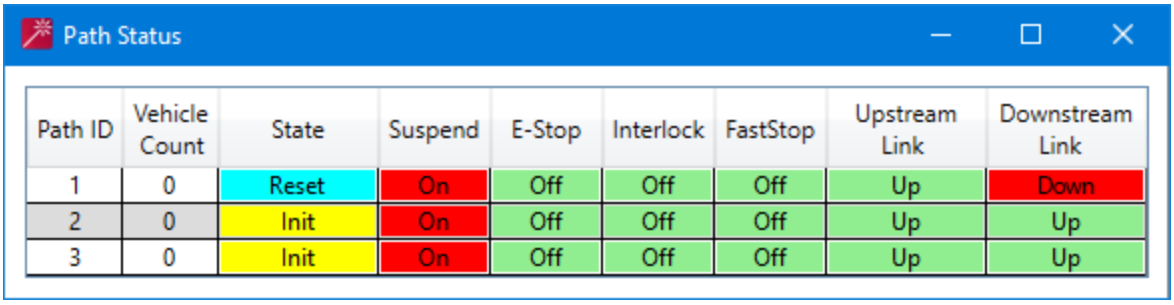
   *If the **Track Graphics** window is open, all vehicles are cleared from the display. If the **Motor Status** widow is open, all motors on the specified paths display "State Error: Movement Suspended by Node Controller".*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	0	Reset	On	Off	Off	Off	Up	Up
2	0	Reset	On	Off	Off	Off	Up	Up
3	0	Reset	On	Off	Off	Off	Up	Up

Figure 2-34: Path Status – Reset Started

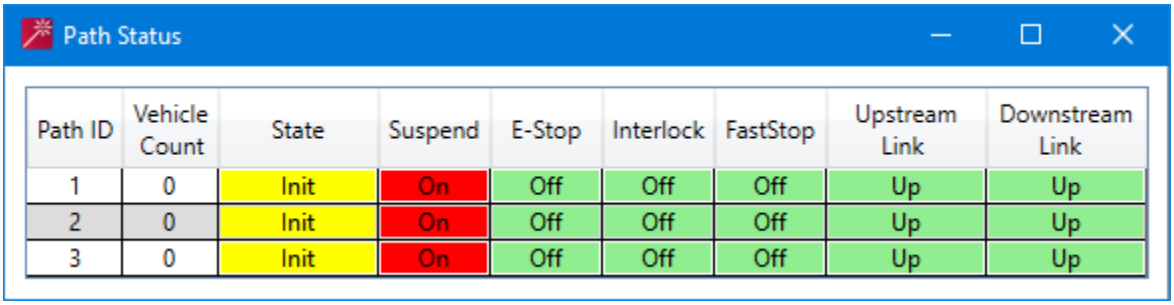
*The **Downstream Link** indicator changes to **Up** as each path is initialized. The **State** indicator for each path changes to **Init** once the reset is complete as shown in Figure 2-35.*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	0	Reset	On	Off	Off	Off	Up	Down
2	0	Init	On	Off	Off	Off	Up	Up
3	0	Init	On	Off	Off	Off	Up	Up

Figure 2-35: Path Status – Reset In Progress

*Once the reset is complete, the **State** indicator of all specified paths displays **Init** as shown in Figure 2-36.*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	0	Init	On	Off	Off	Off	Up	Up
2	0	Init	On	Off	Off	Off	Up	Up
3	0	Init	On	Off	Off	Off	Up	Up

Figure 2-36: Path Status – Reset Complete

Startup Paths

Use the Startup control following a Reset or Warm Reset command, to enable one or more paths on the transport system. Startup forces a rescan of the specified paths to locate all vehicles on those paths.

NOTE: Paths cannot be started if the E-stop, Interlock, or FastStop is On.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Vehicles on QuickStick® and QuickStick® HT transport systems move during startup.

Make sure that barriers are in place before starting the transport system or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

1. Open the **Path Status** window to view the path status during startup if desired (see *Path Status* on page 47).

2. Open the **Motor Status** window to view the motor status during startup if desired (see *Motor Status* on page 51).

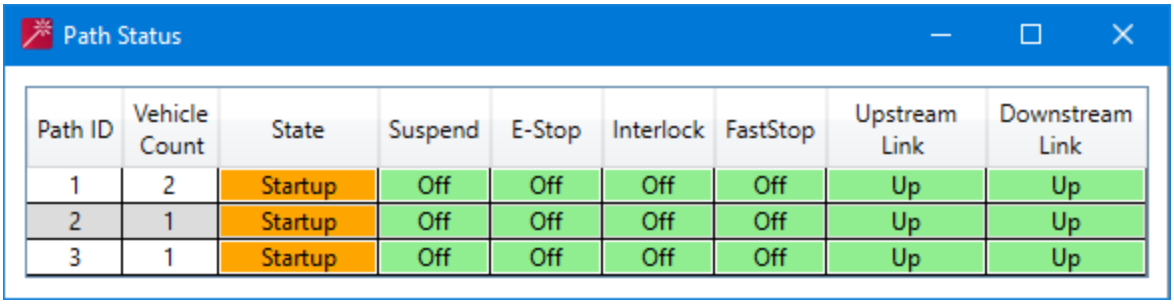
3. Open the **Track Graphics** window to view the graphic of the motors and paths during startup if desired (see *Track Graphics* on page 57).

4. Use the **Path Commands** section to startup the motors on a specific path. Enter the ID of the path in the **Path ID** field then select **Startup**.

   NOTE: To startup the motors on all paths, enter 0 in the **Path ID** field.

   *All motors on the specified paths are started, which identifies all vehicles on the paths. If the **Path Status** window is open, the **Suspend** indicator changes to **Off** and the **State** indicator changes to **Startup** for the specified paths. See Figure 2-37.*

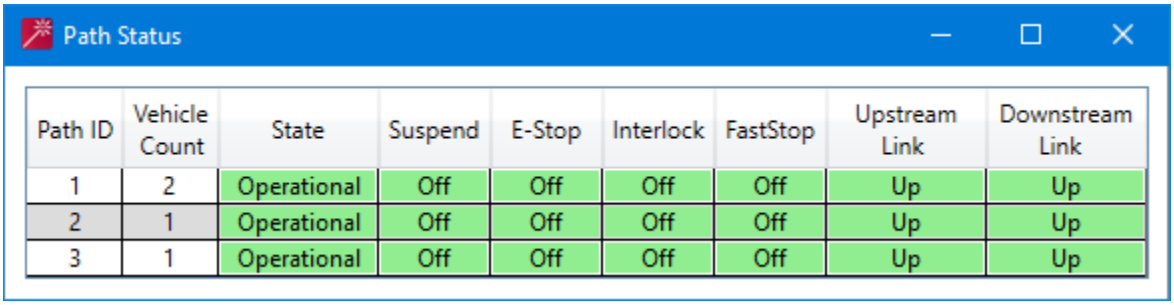
*If the **Track Graphics** window is open, all vehicles are shown on the display at their current location. If the **Motor Status** widow is open, all motors on the specified paths display "No Active Faults"*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	2	Startup	Off	Off	Off	Off	Up	Up
2	1	Startup	Off	Off	Off	Off	Up	Up
3	1	Startup	Off	Off	Off	Off	Up	Up

Figure 2-37: Path Status – Startup Started

*Once all vehicles are located, the **State** indicator changes from **Startup** to **Operational** as shown in Figure 2-38.*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	2	Operational	Off	Off	Off	Off	Up	Up
2	1	Operational	Off	Off	Off	Off	Up	Up
3	1	Operational	Off	Off	Off	Off	Up	Up

Figure 2-38: Path Status – Startup Complete

Suspend Paths

Use the Suspend control to stop all vehicle motion on the specified paths in the transport system and to keep any additional movement from occurring.

1. Open the **Path Status** window to view the path status during the suspend if desired (see *Path Status* on page 47).

2. Open the **Motor Status** window to view the motor status during the suspend if desired (see *Motor Status* on page 51).

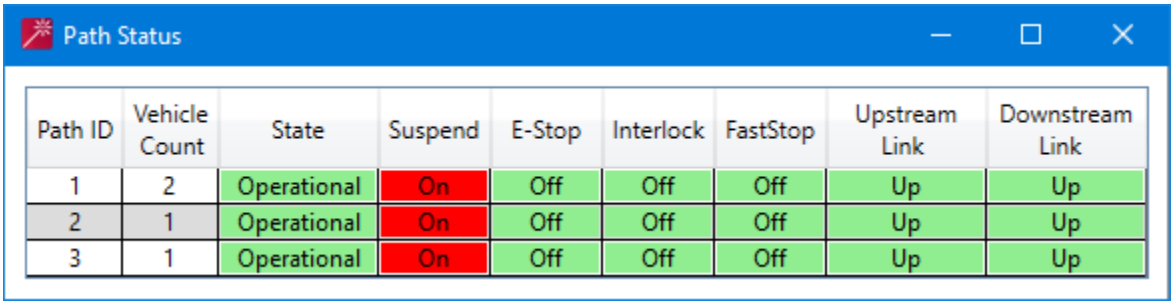
3. Open the **Track Graphics** window to view the graphic of the motors and paths during the suspend if desired (see *Track Graphics* on page 57).

4. Use the **Path Commands** section to suspend the motors on a specific path. Enter the ID of the path in the **Path ID** field then select **Suspend**.

   NOTE: To suspend the motors on all paths, enter 0 in the **Path ID** field.

   All motors on the specified paths are suspended. All vehicles on the specified paths decelerate and stop at a rate that is determined by their current velocity, payload, and active motion profile.

   *If the **Path Status** window is open, the **Suspend** indicator for all specified paths changes to **On** as shown in Figure 2-39. If the **Motor Status** widow is open, all motors on the suspended path display "State Error: Movement Suspended by Node Controller". If the **Track Graphics** window is open, all vehicle motion stops.*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	2	Operational	On	Off	Off	Off	Up	Up
2	1	Operational	On	Off	Off	Off	Up	Up
3	1	Operational	On	Off	Off	Off	Up	Up

Figure 2-39: Path Status – Suspend

Use the **Resume** control to restart vehicle motion on the specified paths after issuing a Suspend command.

Warm Reset Paths

Use the Warm Reset control to reset the motors on the specified paths, which clears all vehicle records from the motors and releases control of all vehicles. The Warm Reset does not interrupt Ethernet communication between the motors on those paths.

NOTE: This feature is only available on QSHT 5700 configurations.

1. Open the **Path Status** window to view the path status during the warm reset if desired (see *Path Status* on page 47).

2. Open the **Motor Status** window to view the motor status during the warm reset if desired (see *Motor Status* on page 51).

3. Open the **Track Graphics** window to view the graphic of the motors and paths during the warm reset if desired (see *Track Graphics* on page 57).

4. Use the **Path Commands** section to issue a warm reset to the motors on a specific path. Enter the ID of the path in the **Path ID** field then select **Warm Reset**.

   NOTE: To perform a warm reset of the motors on all paths, enter 0 in the **Path ID** field.

   *All motors on the specified paths are reset, which clears all vehicle records. If the **Path Status** window is open, the **Suspend** indicator changes to **On** and the **State** indicator changes to **Reset** for the specified paths. The downstream link for each path resets and the **Downstream Link** indicator displays **Down** until the status of the link is verified.*

   *If the **Track Graphics** window is open, all vehicles are cleared from the display. If the **Motor Status** widow is open, all motors on the specified paths display "State Error: Movement Suspended by Node Controller"*

Resume Paths

Use the Resume control to restart vehicle motion on the specified paths after a Suspend Motion or FastStop command was issued. Resume is also used after an Emergency Stop was issued and the E-stop button has been manually reset.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

1. Open the **Path Status** window to view the path status during the resume if desired (see *Path Status* on page 47).

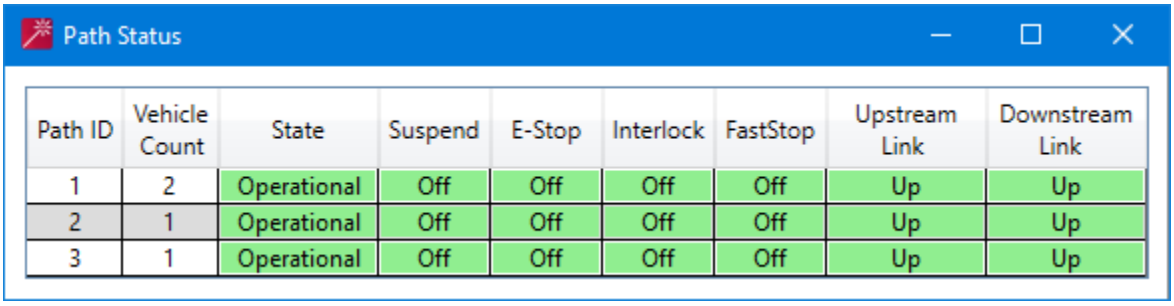
2. Open the **Motor Status** window to view the motor status during the resume if desired (see *Motor Status* on page 51).

3. Open the **Track Graphics** window to view the graphic of the motors and paths during the resume if desired (see *Track Graphics* on page 57).

4. To resume motion on the motors on a specific path, enter the ID of the path in the **Path ID** field then select **Resume**.

   NOTE: To resume motion on all paths, enter 0 in the **Path ID** field.

   *Motion on all specified path resumes. If the **Path Status** window is open, depending upon which function caused the motion to halt, the **Suspend**, **FastStop, or E-Stop** indicator of all specified paths changes from **On** to **Off** as shown in Figure 2-40.*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	2	Operational	Off	Off	Off	Off	Up	Up
2	1	Operational	Off	Off	Off	Off	Up	Up
3	1	Operational	Off	Off	Off	Off	Up	Up

Figure 2-40: Path Status – Operational

FastStop Paths

Use the FastStop control to stop all vehicle motion on the specified paths, abruptly. FastStop causes the motors to apply reverse thrust to all vehicles on the specified paths, which causes the vehicles to stop as fast as possible.

NOTE: This feature is only available on QuickStick and QSHT configurations.

1. Open the **Path Status** window to view the path status during the FastStop if desired (see *Path Status* on page 47).

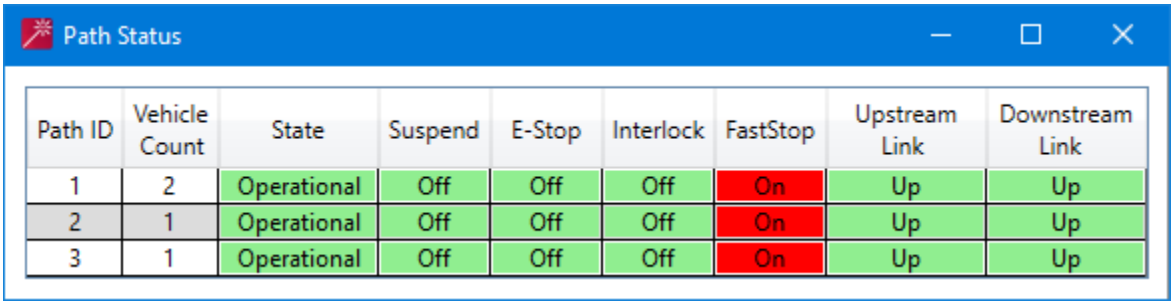
2. Open the **Motor Status** window to view the motor status during the FastStop if desired (see *Motor Status* on page 51).

3. Open the **Track Graphics** window to view the graphic of the motors and paths during the FastStop if desired (see *Track Graphics* on page 57).

4. To stop all vehicles abruptly over the motors on a specific path, enter the ID of the path in the **Path ID** field then select **Reset**.

   NOTE: To stop all vehicles over the motors on all paths, enter 0 in the **Path ID** field.

   *Motion on all specified paths stops abruptly and all vehicles are held in their stopped location. If the **Path Status** window is open, the **FastStop** indicator for all specified paths changes to **On** as shown in Figure 2-41. If the **Motor Status** widow is open, all motors on the fast stopped path display "State Error: Movement Suspended by Node Controller".*



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	2	Operational	Off	Off	Off	On	Up	Up
2	1	Operational	Off	Off	Off	On	Up	Up
3	1	Operational	Off	Off	Off	On	Up	Up

Figure 2-41: Path Status – FastStop

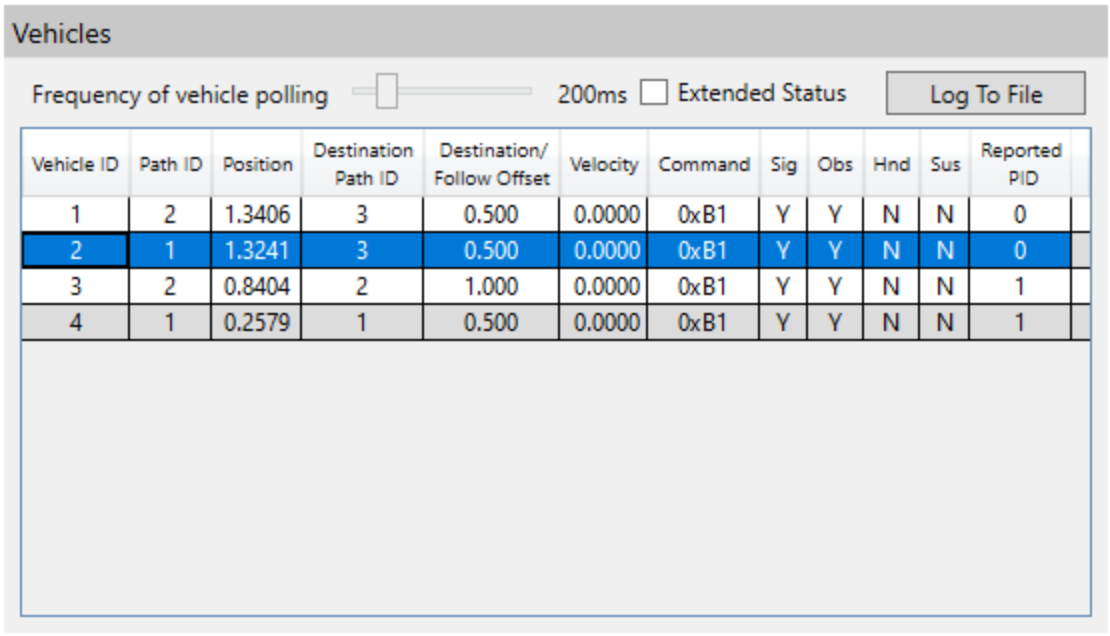
Use the **Resume** control to restart vehicle motion on the specified paths after issuing a Fast-Stop command.

Testing the Transport System – Vehicle Commands

This section describes how to perform vehicle-related tasks. These tasks include; move vehicles to either positions or predefined stations, couple, move, and uncouple platoons, and use vehicle-specific commands.

Setup Vehicle Commands

Instead of entering current vehicle information in the tabs in the Vehicle Commands section, select the row for the appropriate vehicle in the **Vehicles** pane as shown in Figure 2-42. If there are more vehicles on the system than being displayed, use the scroll bar to locate the vehicle to be selected. Once selected, the tabs in the Vehicle Commands section are updated with the data for that vehicle as appropriate.



Vehicle ID	Path ID	Position	Destination Path ID	Destination/ Follow Offset	Velocity	Command	Sig	Obs	Hnd	Sus	Reported PID
1	2	1.3406	3	0.500	0.0000	0xB1	Y	Y	N	N	0
2	1	1.3241	3	0.500	0.0000	0xB1	Y	Y	N	N	0
3	2	0.8404	2	1.000	0.0000	0xB1	Y	Y	N	N	1
4	1	0.2579	1	0.500	0.0000	0xB1	Y	Y	N	N	1

Figure 2-42: Vehicles Pane – Select Vehicle

Move to Position

Moves the specified vehicle to a specified position, relative to the start of a path using the specified parameters.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

1. Open the **Track Graphics** window to view the transport system graphic and the vehicle motion if desired (see *Track Graphics* on page 57).

2. Open the **Node Status** window to view node status during vehicle motion if desired (see *Node Status* on page 48).

3. In the **Vehicles Commands** section, select the **Move To Position** tab.

   *The **Move To Position** tab is displayed as shown in Figure 2-43.*

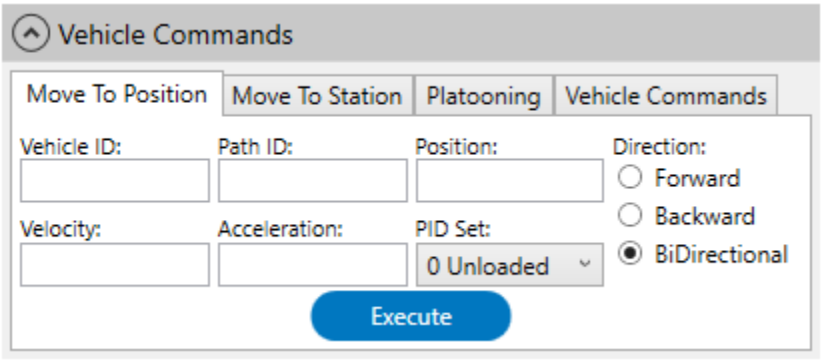


Figure 2-43: Vehicle Commands – Move To Position

4. In the **Vehicle ID** field, enter the ID of the vehicle.

5. In the **Path ID** field, enter the destination path.

6. In the **Position** field, enter the destination for the vehicle as a distance from the start of the specified path in meters.

7. Adjust the **Acceleration** and **Velocity** as needed.

8. From the **PID Set** menu, select the PID set to apply for the load condition of the vehicle.

9. From the **Direction** option list, select either **Forward**, **Backward**, or **Bi-Directional** as the direction for vehicle motion.

10. Select **Execute**.

    *The vehicle moves on the transport system to the designated position. The status for the vehicle in the **Vehicles** pane is updated and the command results are displayed in the **Diagnostic Output** section.*

Move to Station

Moves the specified vehicle to a specified station using the specified parameters.

NOTE: At least one station must be defined to use **Move to Station**.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

1. Open the **Track Graphics** window to view the transport system graphic and the vehicle motion if desired (see *Track Graphics* on page 57).

2. Open the **Node Status** window to view node status during vehicle motion if desired (see *Node Status* on page 48).

3. In the **Vehicles Commands** section, select the **Move To Station** tab.

   *The **Move To Station** tab is displayed as shown in Figure 2-44.*

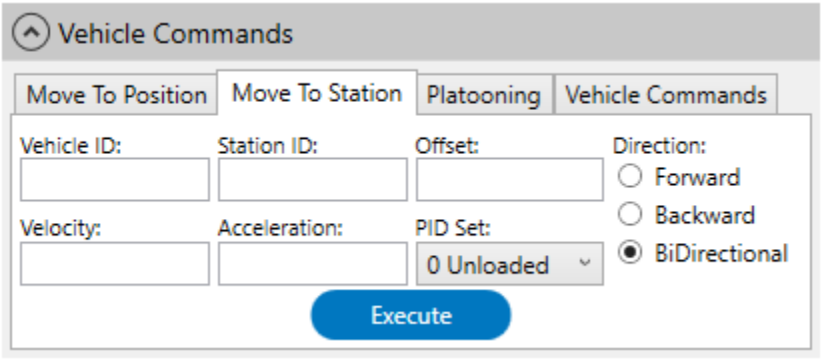


Figure 2-44: Vehicle Commands – Move To Station

4. In the **Vehicle ID** field, enter the ID of the vehicle.

5. In the **Station ID** field, enter the destination station.

6. In the **Offset** field, enter the offset for the vehicle as a distance from the station in meters. Positive values are downstream of the station, negative values are upstream of the station.

7. Adjust the **Acceleration** and **Velocity** as needed.

8. From the **PID Set** menu, select the PID set to apply for the load condition of the vehicle.

9. From the **Direction** option list, select either **Forward**, **Backward**, or **Bi-Directional** as the direction for vehicle motion.

10. Select **Execute**.

    *The vehicle moves on the transport system to the designated station. The status for the vehicle in the **Vehicles** pane is updated and the command results are displayed in the **Diagnostic Output** section.*

Vehicle Platooning

Platooning allows multiple vehicles on a path to move simultaneously without brick-wall headway between the vehicles in the platoon. Brick-wall headway is still maintained before and after the platoon. Platooning is useful for stations that have multiple processes. It allows a set of vehicles to move into, and out of, position simultaneously, which minimizes change over time.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

When coupling a vehicle to a platoon using the vehicle follow order, the vehicle and the platoon must be stopped (movement must be below the Arrival Velocity Tolerance defined on the Global Settings page in the Configurator when creating the Node Controller Configuration File).

When uncoupling a vehicle from a platoon using a vehicle position order, the platoon must be stopped. When uncoupling a vehicle from a platoon using a vehicle follow order with a decouple destination, the vehicle decouples from the platoon when the platoon approaches the decouple destination. The vehicle then stops at the decouple destination when the platoon passes that position.

NOTE: This feature is only available on MagneMover LITE and QuickStick configurations.

Couple Vehicles into a Platoon

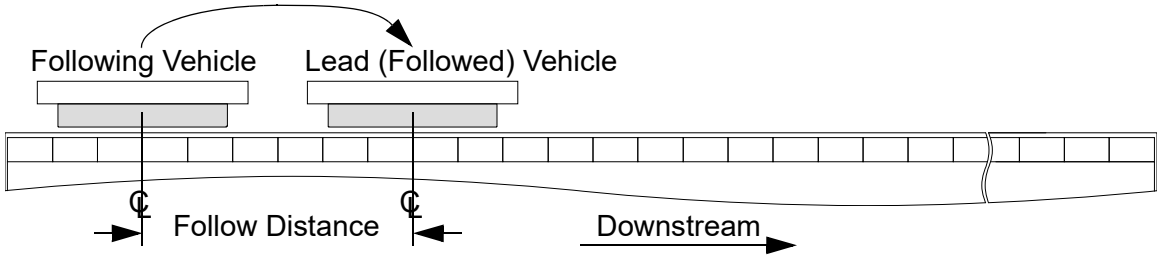


Figure 2-45: Create a Platoon

1. Open the **Track Graphics** window to view the transport system graphic and the vehicle motion if desired (see *Track Graphics* on page 57).

2. Open the **Node Status** window to view node status during vehicle motion if desired (see *Node Status* on page 48).

3. In the **Vehicles Commands** section, use the **Move To Position** tab to stop the vehicle that is, or will be, the lead (followed) vehicle (see Figure 2-45).

4. In the **Vehicles Commands** section, use the **Move To Position** tab to position the following vehicle behind the vehicle to follow in the platoon (see Figure 2-45). The platooning command must specify this distance between the vehicles as the Follow Distance.

5. In the **Vehicles Commands** section, select the **Platooning** tab.

   *The **Platooning** tab is displayed as shown in Figure 2-46.*

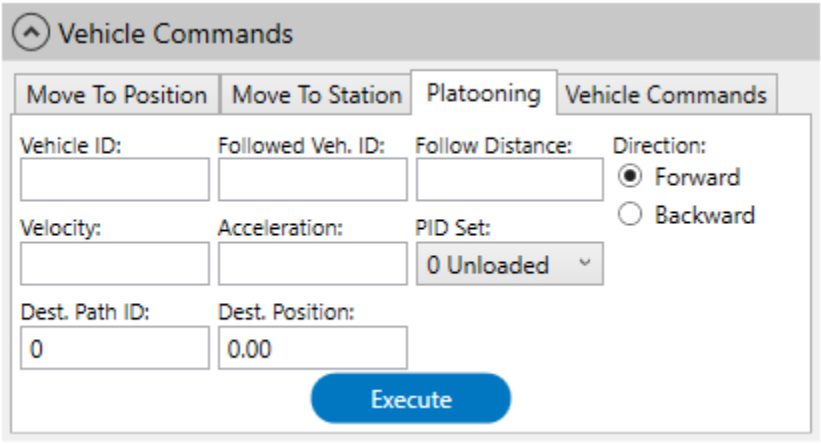


Figure 2-46: Vehicle Commands – Platooning

6. In the **Vehicle ID** field, enter the ID of the vehicle to command. This vehicle becomes the following vehicle.

7. In the **Followed Veh. ID** field, enter the ID of the vehicle to be followed.

8. In the **Follow Distance** field, enter the distance for the vehicle to follow its leader (center of vehicle to center of vehicle).

9. In the **Acceleration** and **Velocity** fields, enter the acceleration and velocity to be used when the vehicle is catching up to a moving vehicle to follow or when decoupling from a moving vehicle.

10. From the **PID Set** menu, select the PID set to apply for the load condition of the vehicle.

11. In the **Dest. Path ID** field, enter the ID of the path where the automatic decoupling position is located (where the vehicle stops after automatic decoupling while in motion). Or, enter zero to disable the automatic decoupling feature.

12. In the **Dest. Position** field, enter the decoupling destination for the vehicle as a distance from the start of the specified decoupling path in meters (where the vehicle stops after automatic decoupling while in motion).

13. From the **Direction** option list, select either **Forward** or **Backward** as the direction for the vehicle to move to join the platoon. This is also the direction of motion for the platoon.

14. Select **Execute**.

    The current vehicle is coupled to the vehicle to follow and becomes the following vehicle.

15. Use the **Move To Position** tab to command the lead vehicle of the platoon as required. All vehicles in the platoon follow it at the specified distances.

Uncouple a Platoon

When uncoupling a vehicle from a platoon using a vehicle position order, the platoon must be stopped.

1. In the **Vehicles Commands** section, use the **Move To Position** tab to stop the platoon leader.

2. In the **Vehicles Commands** section, use the **Move To Position** tab to command each following vehicle to move as required. If desired, vehicles can be moved to their current position.

   The vehicle moves as directed, which uncouples it from the vehicle it is following. If the vehicle was commanded to its current position, no actual motion occurs. When there is a vehicle that is coupled to this vehicle, it remains coupled and this vehicle becomes the leader of a new platoon.

3. Once the vehicles are uncoupled, command them to their new positions as required.

To uncouple a vehicle from a moving platoon, use a vehicle follow order with a decouple destination. Configure the decouple destination as shown in Step 11 and Step 12 of *Couple Vehicles into a Platoon*.

Vehicle Commands

The Vehicle Commands tab is used to send commands for specific vehicles to the motors.

Lock/Unlock a Vehicle

While a vehicle is locked, any motion commands are rejected.

1. In the **Vehicles Commands** section, select the **Vehicle Commands** tab.

   *The **Vehicle Commands** tab is displayed as shown in Figure 2-47. The Command Menu defaults to Lock or the last command selected.*

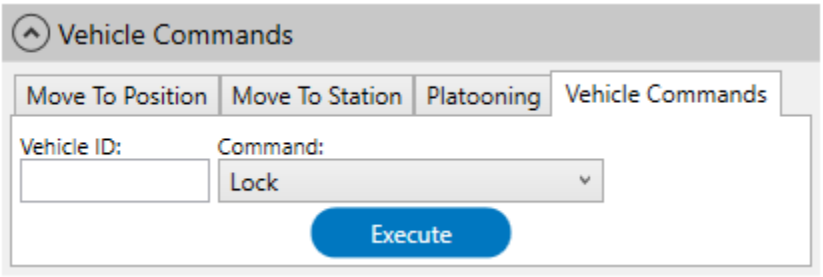


Figure 2-47: Vehicle Commands – Vehicle Commands, Lock

2. In the **Vehicle ID** field, enter the ID of the vehicle.
3. To lock a vehicle, select **Lock** from the Command Menu.
4. Select **Execute**.

   The vehicle is locked and cannot be moved.

5. To unlock a vehicle, select **Unlock** from the Command Menu.
6. Select **Execute**.

   The vehicle is unlocked and can be moved.

Delete a Vehicle

Delete vehicles in the transport system that the system cannot detect (Vehicle Signal = N) or that have been moved out of position (Vehicle Suspect = Y). Typically, vehicles are no longer detected if they have been moved manually. While not recommended, the **Delete Vehicle** command can be used in place of Terminus Nodes to remove a vehicle from the system for maintenance. Once a vehicle has been deleted, it can be recovered at its new location using the **Startup** option on the **Path** tab in the **Host Commands** section (see *Startup Paths* on page 66).

IMPORTANT	Using the Delete Vehicle command to delete a vehicle that is physically present on the transport system can cause collisions as the system no longer accounts for that vehicle as it moves other vehicles.

1. Open the **Track Graphics** window to view the transport system graphic and the vehicle status if desired (see *Track Graphics* on page 57).

2. In the **Vehicles Commands** section, select the **Vehicle Commands** tab.

   *The **Vehicle Commands** tab is displayed as shown in Figure 2-48. The Command Menu defaults to Lock or the last command selected.*

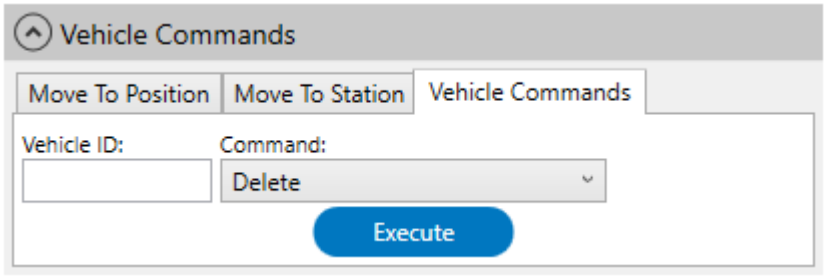


Figure 2-48: Vehicle Commands – Vehicle Commands, Delete

3. In the **Vehicle ID** field, enter the ID of the vehicle.

4. Select **Delete** from the Command Menu.

5. Select **Execute**.

   *The vehicle is deleted from the Vehicle List and removed from the **Vehicles** pane.*

Recover a Deleted Vehicle

1. In the **Path** tab of the **Host Commands** section of the **Settings** panel, enter the ID of the path where the vehicle is now located, then select **Startup**.

   All vehicles on the specified path are identified and placed back under motor control.

IMPORTANT	The IDs of vehicles that were deleted may be assigned to the rediscovered vehicles. However, the Vehicle ID assigned to a vehicle cannot be assumed to be the same ID that the vehicle had before it was deleted.

	**CRUSH HAZARD:** Moving mechanisms have no obstruction sensors. Vehicles on QuickStick and QuickStick HT transport systems move during startup. Make sure that barriers are in place before starting the transport system or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

2. Move the recovered vehicle as required.

Clear the Vehicle Suspect Bit

The Suspect Vehicle Bit identifies vehicles that are manually moved out of the control region for that vehicle. Typically, control of the vehicle cannot be regained in this situation. Even if control is regained, the vehicle continues to be suspect. This command provides user control of the Suspect bit status, which allows the Suspect bit to be cleared once control of the vehicle is regained.

1. In the **Vehicles Commands** section, select the **Vehicle Commands** tab.

   *The **Vehicle Commands** tab is displayed as shown in Figure 2-49. The Command Menu defaults to Lock or the last command selected.*

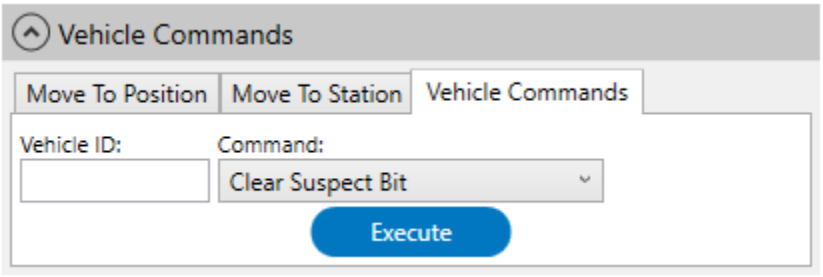


Figure 2-49: Vehicle Commands – Vehicle Commands, Clear Suspect Bit

2. In the **Vehicle ID** field, enter the ID of the vehicle.

3. Select **Clear Suspect Bit** from the Command Menu.

4. Select **Execute**.

   *The vehicle suspect status (Sus) changes to N. The **Vehicles** pane updates to show the new status.*

Testing the Transport System – Node Commands

This section describes how to perform node-related tasks. These tasks include; perform hand-shaking for Terminus Nodes, enable/disable Gateway Nodes, and link/unlink Moving Path Nodes.

Terminus Node

Terminus Nodes are used to move a vehicle onto or off the transport system. Any vehicle entering or exiting the transport system must move through a Terminus Node. These nodes allow vehicles to move on or off a path from equipment separate from the MagneMotion transport system. See the *MagneMover LITE Configurator User Manual*, MMI-UM008, or the *QuickStick Configurator User Manual*, MMI-UM009, for a description of the Terminus Node.

Vehicle Exit from the Transport System

Movement of a vehicle off the transport system through a Terminus Node requires that the Terminus Node be clear as shown in the examples in Figure 2-50 and Figure 2-51.

Node Status

Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Terminus	N/A	N/A	0	-	-	On	Off	Off	Off	Off
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-50: Node Status Window Showing Terminus Node Ready for Vehicle

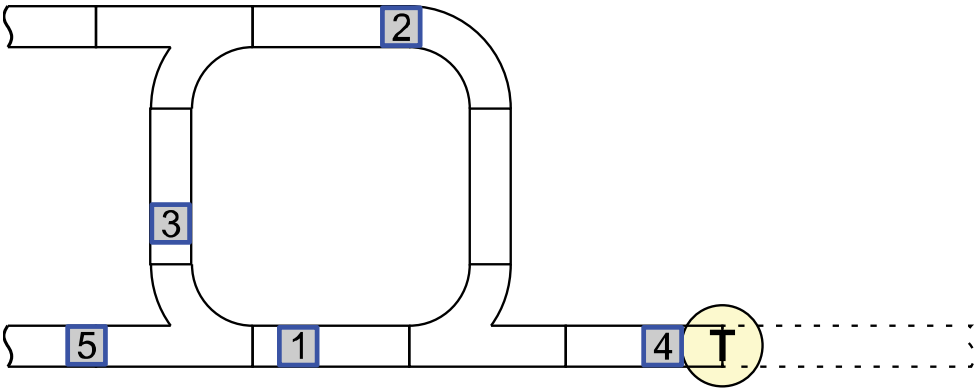


Figure 2-51: Track Layout Showing Terminus Node Ready for Vehicle

1. Configure the Terminus Node for vehicle exit as shown in Figure 2-52. See *Terminus Tab on page 177* for detailed descriptions of all items.

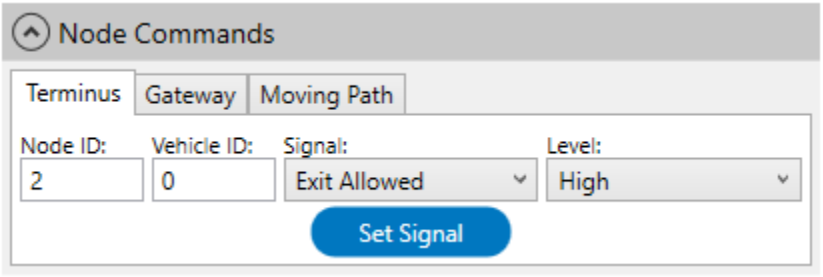


Figure 2-52: Move a Vehicle Off a Terminus Node

A. In the **Node Commands** section of the **Settings** panel, select the **Terminus** tab.

B. Enter the **Node ID** of the Terminus Node.

C. Enter a **Vehicle ID**.

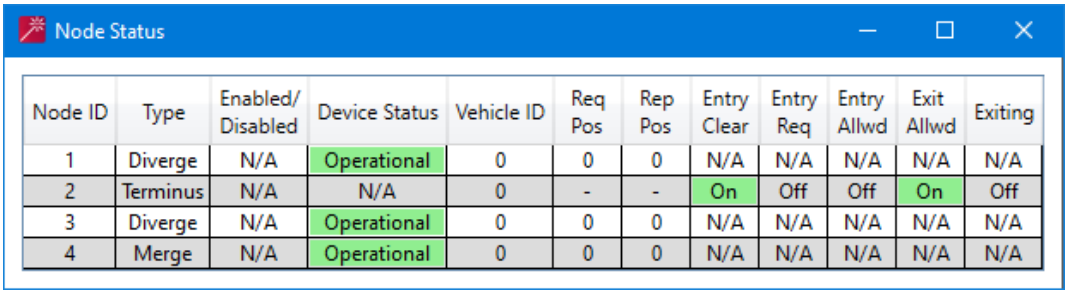
NOTE: A Vehicle ID of 0 can be used since the Vehicle ID does not need to be explicitly specified when exiting. Also, the vehicle exiting is specified in the **Vehicle Commands** section.

D. Set the **Signal** type to Exit Allowed.

E. Set the signal **Level** to High to indicate that the remote equipment is ready to receive a vehicle.

F. Select **Set Signal**.

*The **Exit Allowed** signal goes high as shown in Figure 2-53.*



Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Terminus	N/A	N/A	0	-	-	On	Off	Off	On	Off
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-53: Node Status Window Showing Terminus Node Exit Allowed

2. Use the **Vehicle Commands** section in the **Settings** panel to move the vehicle through the Terminus Node.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

A. Enter the **Vehicle ID** of the vehicle exiting from the transport system.

B. From the **PID Set** menu, select the PID set to apply as the load condition for the vehicle.

C. From the **Direction** menu, select either Forward, Backward, or Bi-Directional as the direction of vehicle motion.

D. Enter the **Path ID** where the Terminus Node is located.

E. Enter the destination **Position** for the vehicle exiting from the transport system.

For motion downstream when the Terminus Node is at the end of the path, enter a destination **Position** for the vehicle as a distance from the start of the path that locates the vehicle past the Terminus Node and off the guideway (typically *PathLength* + *VehicleLength*) as shown in Figure 2-54.

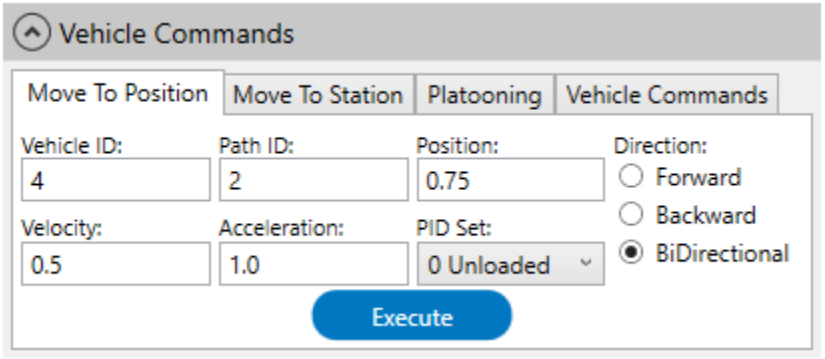


Figure 2-54: Exit Terminus Node at Path End

For motion upstream when the Terminus Node is at the beginning of the path, enter a destination **Position** for the vehicle as a distance from the start of the path that locates the vehicle past the Terminus Node and off the guideway (typically - *VehicleLength*) as shown in Figure 2-55.

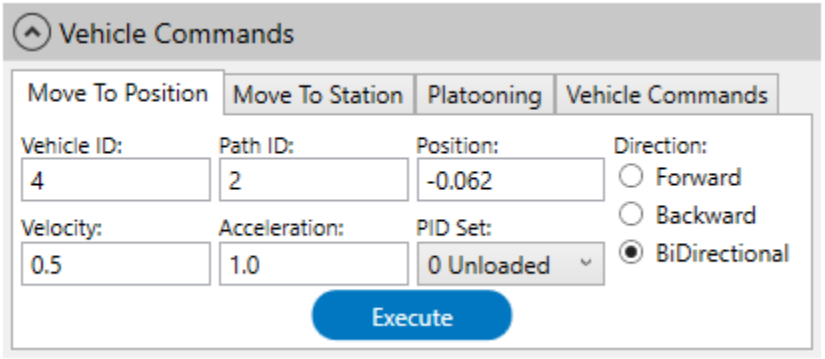


Figure 2-55: Exit Terminus Node at Path Beginning

F. Adjust the **Acceleration** and **Velocity** as needed.

G. Select **Execute**.

*The **Exiting** signal goes high as shown in Figure 2-56 and the vehicle moves through the Terminus Node and onto the remote equipment, see Figure 2-57 and Figure 2-58.*

Node Status

Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Terminus	N/A	N/A	4	-	-	Off	On	On	Off	Off
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-56: Node Status Window Showing Terminus Node Exiting Signal High

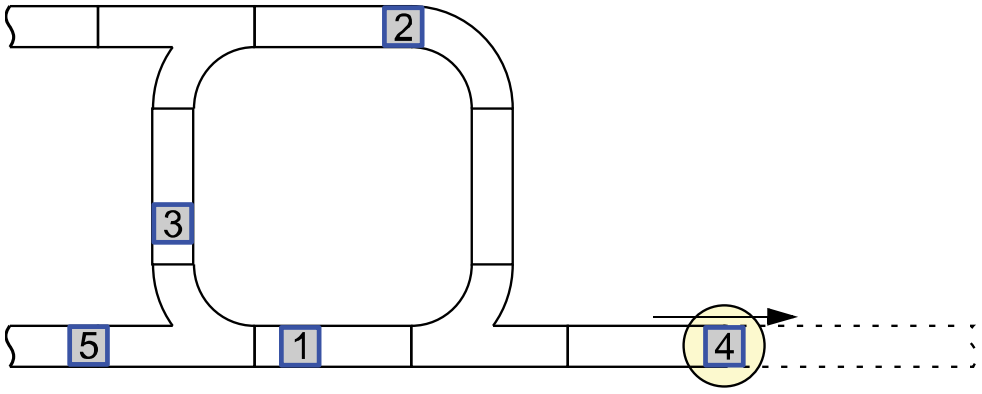


Figure 2-57: Track Layout Showing Vehicle Exiting Terminus Node

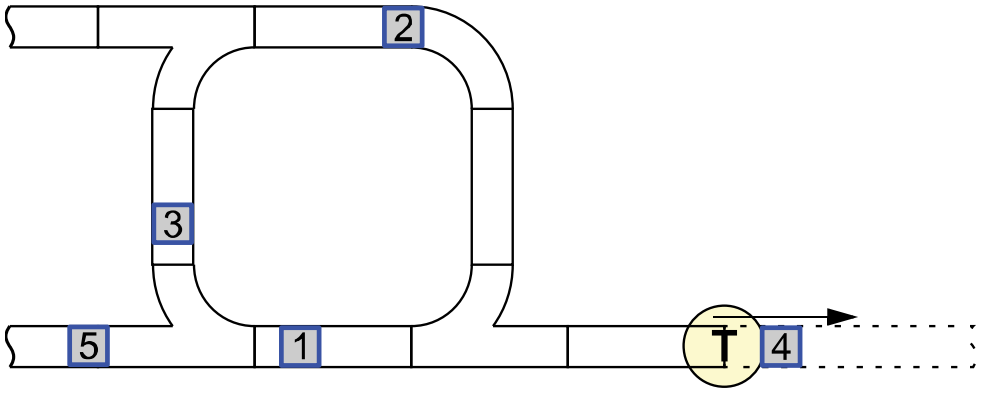


Figure 2-58: Track Layout Showing Vehicle Moved to Remote Equipment

NOTE: The remote equipment must take control of the vehicle and move it the rest of the way off the MagneMotion transport system.

3. Configure the Terminus Node to complete the vehicle exit as shown in Figure 2-59.

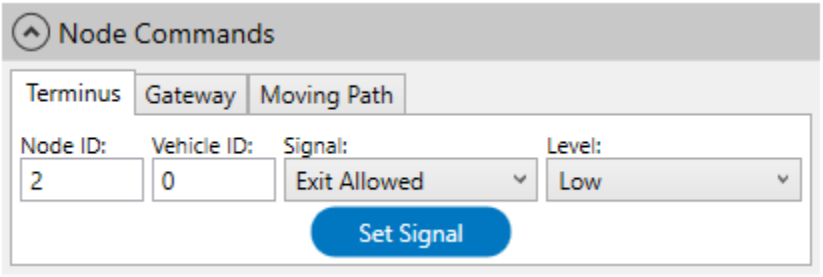


Figure 2-59: Complete Movement of Vehicle Off a Terminus Node

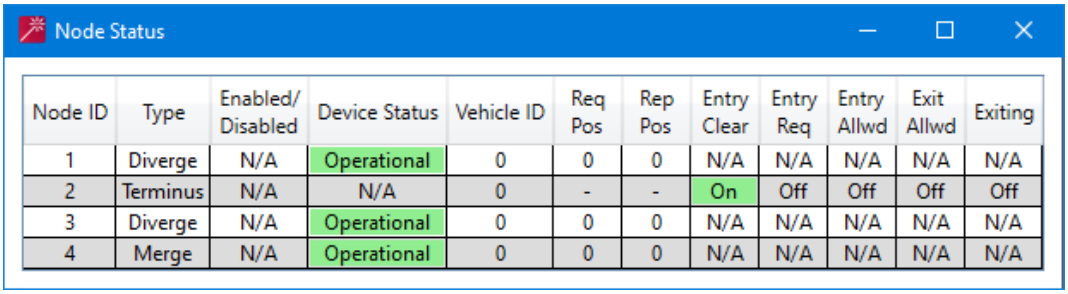
A. Set the **Signal** type to Exit Allowed.

B. Set the signal **Level** to Low to acknowledge completion of the transfer and to command the HLC to delete the vehicle from the Vehicle List.

C. Select **Set Signal**.

*The **Exit Allowed** signal goes low and the **Entry Clear** signal goes high as shown in Figure 2-60 and the vehicle is deleted from the Vehicle List.*

NOTE: The **Exiting** signal goes low. The vehicle is deleted from the Vehicle List once the motor in the Terminus Node no longer detects the magnet array on the vehicle and **Exit Allowed** is low. Once **Entry Clear** goes high, the node is available for another transfer.



Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Terminus	N/A	N/A	0	-	-	On	Off	Off	Off	Off
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-60: Node Status Window Showing Terminus Node Ready for Next Vehicle

Vehicle Entry onto the Transport System

Movement of a vehicle onto the transport system through a Terminus Node requires that the Terminus Node be clear as shown in the examples in Figure 2-61 and Figure 2-62.

Node Status

Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Terminus	N/A	N/A	0	-	-	On	Off	Off	Off	Off
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-61: Node Status Window Showing Terminus Node Ready for Vehicle

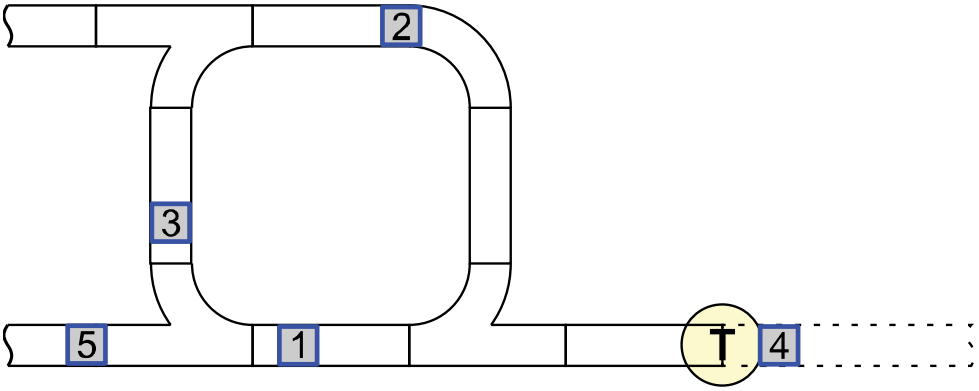


Figure 2-62: Track Layout Showing Vehicle on Remote Equipment

1. Configure the Terminus Node for vehicle entry as shown in Figure 2-63. See *Terminus Tab on page 177* for detailed descriptions of all items.

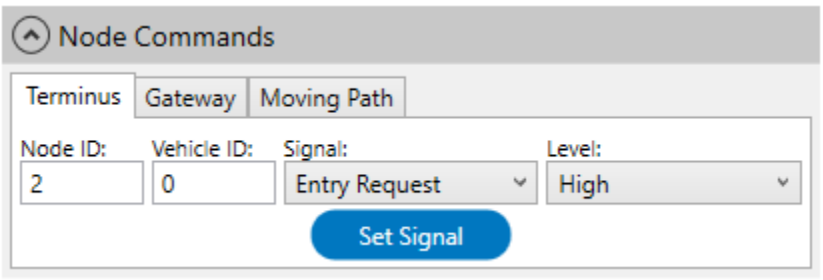


Figure 2-63: Move a Vehicle Onto a Terminus Node

A. In the **Node Commands** section of the **Settings** panel, select the **Terminus** tab.

B. Enter the **Node ID** of the Terminus Node.

C. Enter a **Vehicle ID**.

NOTE: A Vehicle ID of 0 commands the HLC to use the next available Vehicle ID from the Vehicle List.

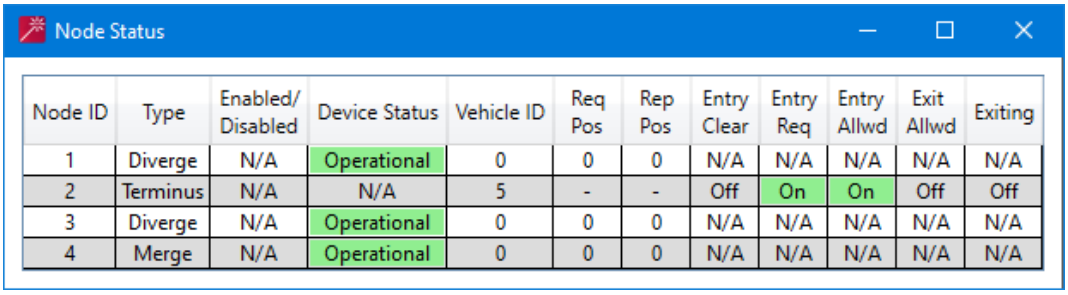
D. Set the **Signal** type to Entry Request.

E. Set the signal **Level** to High to indicate that the remote equipment is ready to insert a vehicle.

F. Select **Set Signal**.

*The **Entry Request** signal goes high as shown in Figure 2-64, the transport system responds by setting the **Entry Allowed** signal high.*

*When the **Entry Allowed** signal goes high, the HLC assigns the Vehicle ID for the vehicle entering the transport system, which allows commands for the vehicle to be entered.*



Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Terminus	N/A	N/A	5	-	-	Off	On	On	Off	Off
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-64: Node Status Window Showing Terminus Node Entry Allowed

2. Command the remote equipment to position the vehicle on the Terminus Node.

When the motor detects the magnet array on the vehicle, the vehicle is moved from the remote equipment and onto the MagneMotion transport system as shown in Figure 2-65. If there is a pending command for the vehicle, it is moved as directed by the command.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

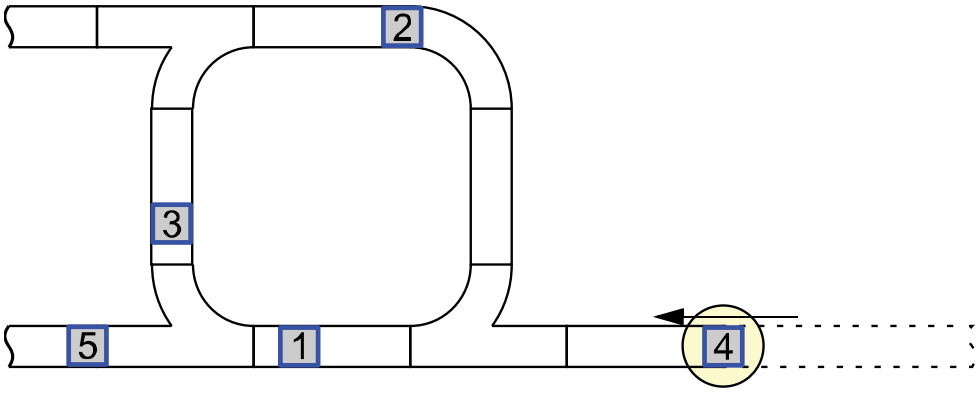


Figure 2-65: Track Layout Showing Vehicle Entering Terminus Node

3. Configure the Terminus Node to complete the vehicle entry as shown in Figure 2-66.

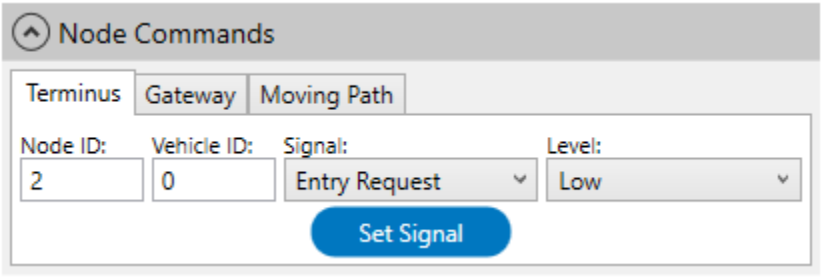


Figure 2-66: Complete Movement of Vehicle Onto a Terminus Node

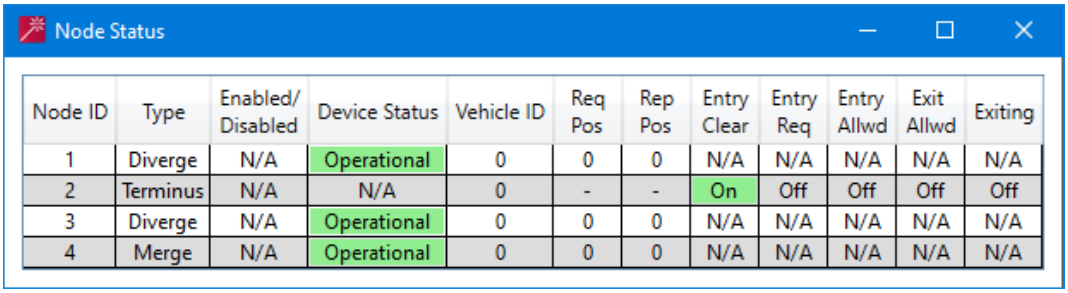
A. Set the **Signal** type to Entry Request.

B. Set the signal **Level** to Low to acknowledge completion of the transfer.

C. Select **Set Signal**.

*The **Entry Request** signal goes low and **Entry Clear** goes high as shown in Figure 2-67.*

NOTE: The **Entry Allowed** signal goes low when the motor detects the magnet array on the vehicle in the Terminus Node and **Entry Request** is low. The **Entry Clear** signal goes high once the vehicle moves away from the end of the motor where the node is located as shown in Figure 2-68. Once the **Entry Clear** signal goes high, the node is available for another transfer.



Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Terminus	N/A	N/A	0	-	-	On	Off	Off	Off	Off
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 2-67: Node Status Window Showing Terminus Node Ready for Next Vehicle

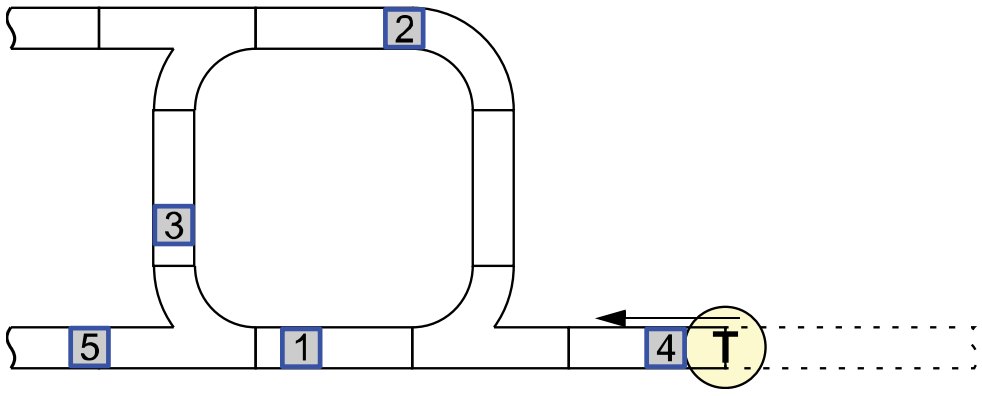


Figure 2-68: Track Layout Showing Vehicle on Transport System

Gateway Node

Gateway Nodes are used to move a vehicle from one HLC Control Group to another HLC Control Group within a transport system. While Terminus Nodes can be used for these transfers, Gateway Nodes provide additional functionality to simplify the transfer. Vehicles should use a Gateway Node when moving between HLC Control Groups, which allows the vehicles to maintain their Vehicle ID. See the *MagneMover LITE Configurator User Manual*, MMI-UM008, or the *QuickStick Configurator User Manual*, MMI-UM009, for a description of HLC Control Groups and the Gateway Node.

NOTE: Gateway Nodes are not supported when running in simulation mode.

Vehicle Movement Between HLC Control Groups

The Gateway Nodes must be enabled in both HLC Control Groups as shown in Figure 2-69 to enable vehicle motion from one HLC Control Group to another through the Gateway Node. See *Gateway Tab* on page 178 for detailed descriptions of all items.

Node Status

Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Gateway	Enabled	N/A	0	-	-	N/A	N/A	N/A	N/A	N/A
3	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

HLC Control Group A

Node Status

Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
3	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
4	Gateway	Enabled	N/A	0	-	-	N/A	N/A	N/A	N/A	N/A

HLC Control Group B

Figure 2-69: Node Status Window Showing Gateway Node Ready for Vehicle

1. If the Gateway Nodes are not enabled, setup each Gateway Node for vehicle transit using the *Gateway Tab* in the Node Commands section as shown in Figure 2-70.

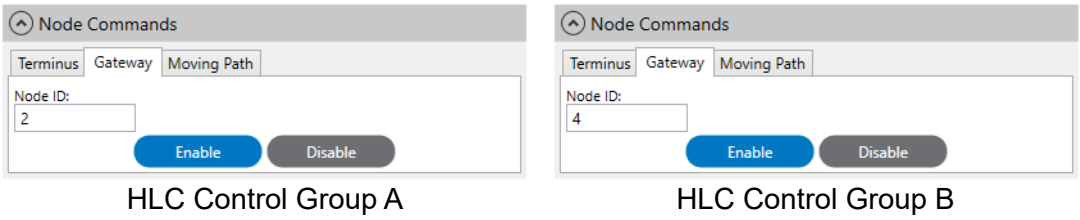


Figure 2-70: Enable a Gateway Node

   A. Using the NCHost application for HLC Control Group A, in the **Node Commands** section of the **Settings** panel, select the **Gateway** tab.

   B. Enter the **Node ID** of the Gateway Node.

   C. Select **Enable**.

   D. Using the NCHost application for HLC Control Group B, in the **Node Commands** section of the **Settings** panel, select the **Gateway** tab.

   E. Enter the **Node ID** of the Gateway Node.

   F. Select **Enable**.

2. Use the **Vehicle Commands** section of the NCHost application for HLC Control Group A to move the vehicle through the Gateway Node and into the other HLC Control Group.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

   A. Enter the **Vehicle ID** of the vehicle to exit the HLC Control Group.

   B. From the **PID Set** menu, select the PID set to apply for the load condition of the vehicle.

   C. From the **Direction** menu, select either Forward, Backward, or Bi-Directional as the direction of vehicle motion.

   D. Enter the **Path ID** where the Gateway Node is located.

   E. Enter the destination **Position** for the vehicle exiting from the HLC Control Group.

      For motion downstream when the Gateway Node is at the end of a path, enter a destination **Position** for the vehicle as a distance from the start of the path that

locates the vehicle past the Gateway Node and into the other HLC Control Group (typically *PathLength* + *VehicleLength*) as shown in Figure 2-71.

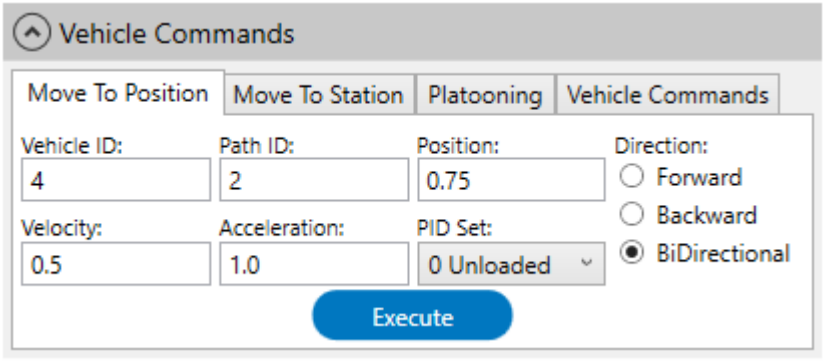


Figure 2-71: Transit Gateway Node at Path End

For motion upstream when the Gateway Node is at the beginning of the path, enter a destination **Position** for the vehicle as a distance from the start of the path that locates the vehicle past the Gateway Node and onto the other HLC Control Group (typically - *VehicleLength*) as shown in Figure 2-72.

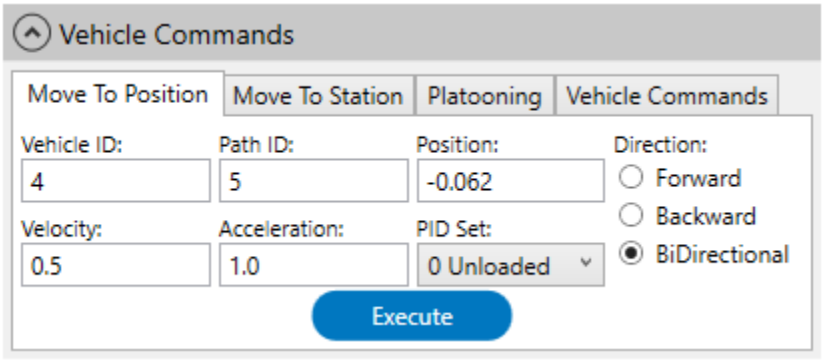


Figure 2-72: Transit Gateway Node at Path Beginning

F. Adjust the **Acceleration** and **Velocity** as needed.

G. Select **Move to Position**.

The vehicle moves through the Gateway Node as shown in Figure 2-73, Figure 2-74, and Figure 2-75 to the defined destination position in HLC Control Group B. Once the vehicle is in HLC Control Group B, that HLC Control Group owns it and can move it as required.

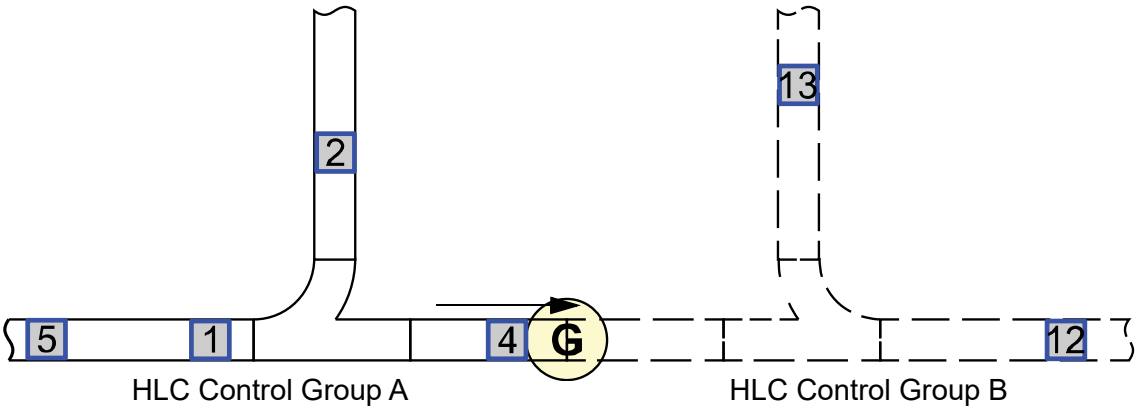


Figure 2-73: Vehicle in HLC Control Group A Approaching Gateway Node

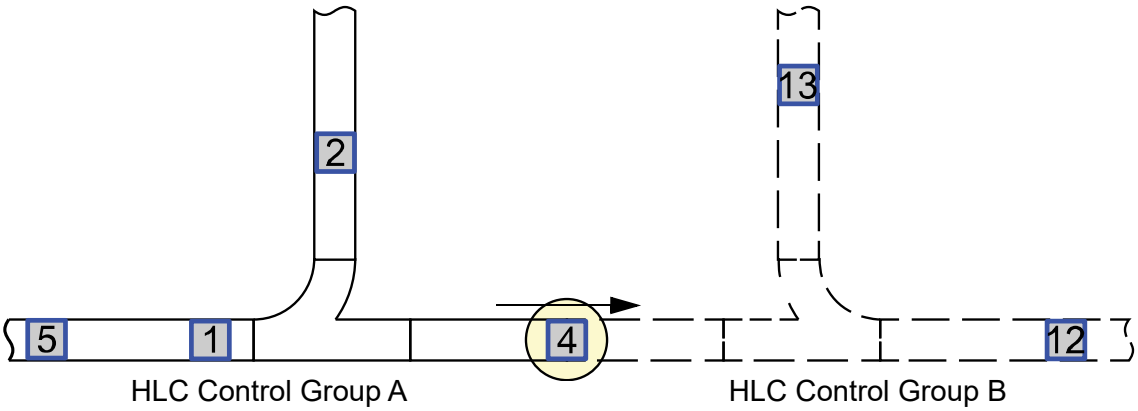


Figure 2-74: Vehicle Transiting Gateway Node

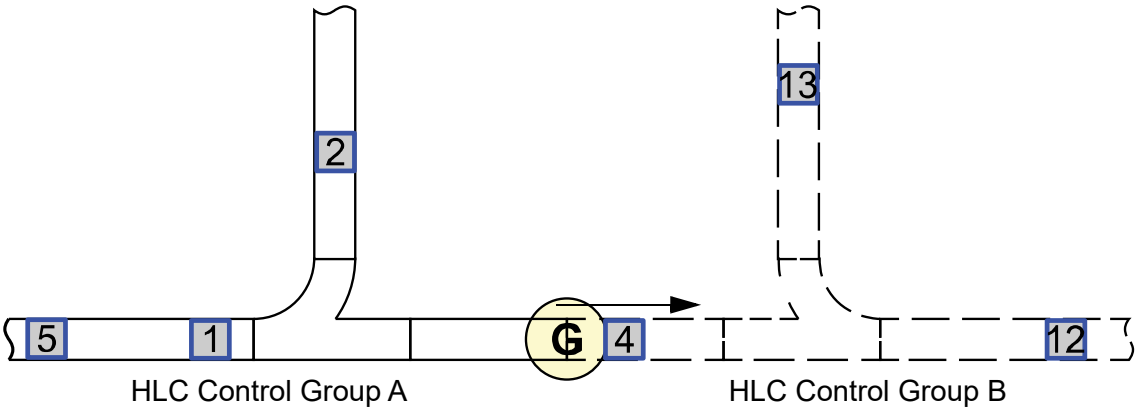


Figure 2-75: Vehicle in HLC Control Group B Exiting Gateway Node

Moving Path Node

A Moving Path Node is used to connect the ends of multiple paths using a path that is moved by a host-controlled mechanism. These nodes use a shuttle, or moving path, mounted on a drive path to move a vehicle between multiple parallel paths. For proper vehicle transfer, the Moving Path must be properly positioned and configured as linked to the Fixed Path. Once the transfer is complete, the Moving Path must be configured as unlinked.

See the *QuickStick Configurator User Manual*, MMI-UM009, for a description of the Moving Path Node.

NOTE: This feature is only available on QuickStick and QSHT configurations.

Moving Path Nodes are not supported when running in simulation mode.

Position the Moving Path

The Moving Path must be positioned so that it mechanically aligns with one of the Fixed Paths in the node using a host-controlled drive mechanism. This mechanism may be any user-supplied mechanism, including QuickStick motors. For this example, the Moving Paths are mounted on shuttles on a separate QuickStick drive path as shown in Figure 2-76.

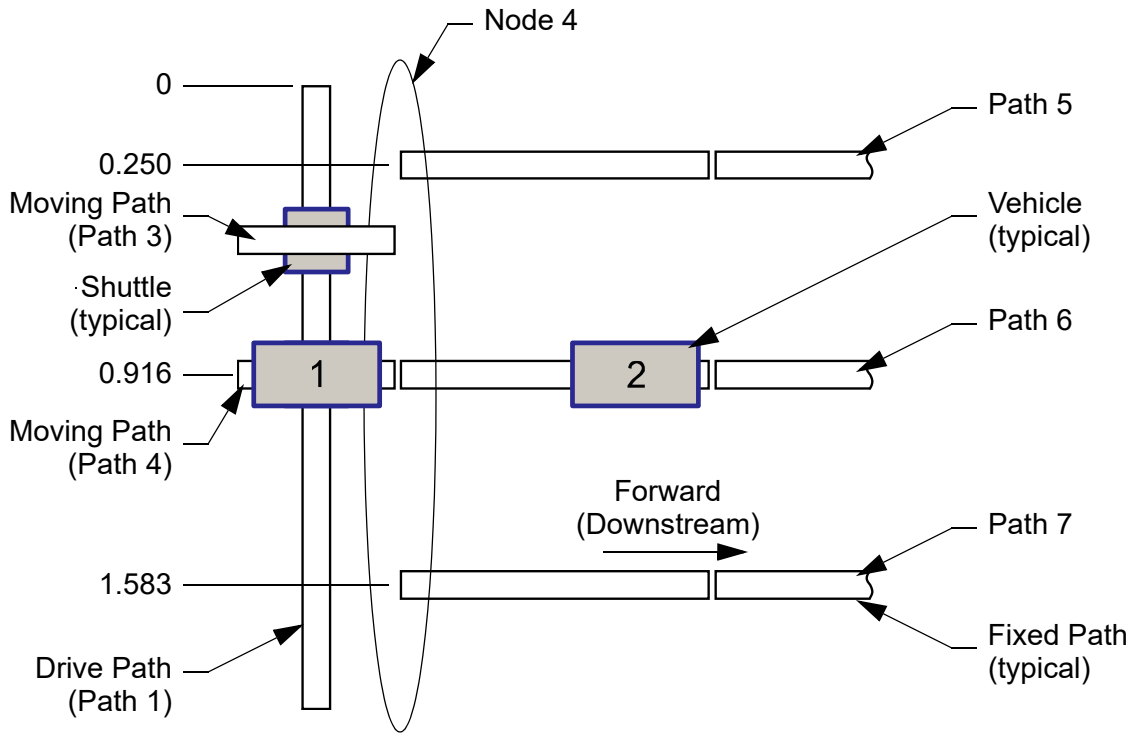


Figure 2-76: Moving Path Track Layout Showing Drive Mechanism

Use the **Vehicle Commands** section of the **Settings** panel to move the shuttle on the drive path as shown in Figure 2-77.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

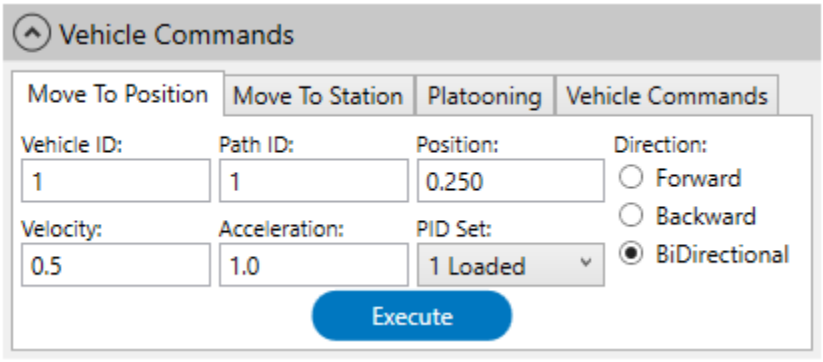


Figure 2-77: Move Shuttle on Drive Path

1. Enter the **Vehicle ID** of the shuttle being moved.
2. From the **PID Set** menu, select the PID set to apply as the load condition for the vehicle.
3. From the **Direction** menu, select either Forward, Backward, or Bi-Directional as the direction of vehicle motion.
4. Enter the **Path ID** of the drive path.
5. Enter the destination **Position** for the shuttle as a distance from the start of the drive path.
6. Adjust the **Acceleration** and **Velocity** as needed.
7. Select **Move to Position**.

   The shuttle moves on the drive path as specified, and aligns the Moving Path with the specified Fixed Path.

Link a Moving Path to a Fixed Path

Once the Moving Path is positioned so that it mechanically aligns with one of the Fixed Paths, it must be linked to that path to create a junction for vehicle motion. In the **Host Commands** section, select the **Moving Path** tab (see Figure 2-78) to link the Moving Path to a Fixed Path. See *Moving Path Tab* on page 179 for detailed descriptions of all items.

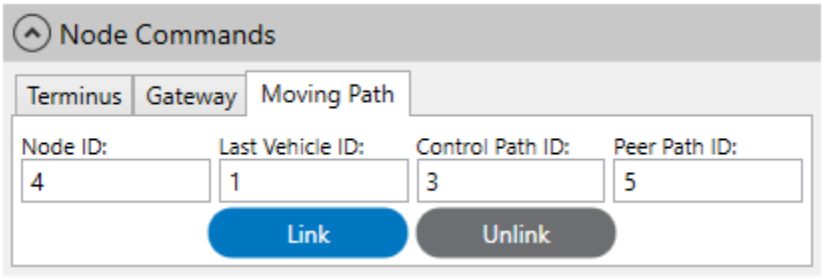


Figure 2-78: Moving Path Link Command

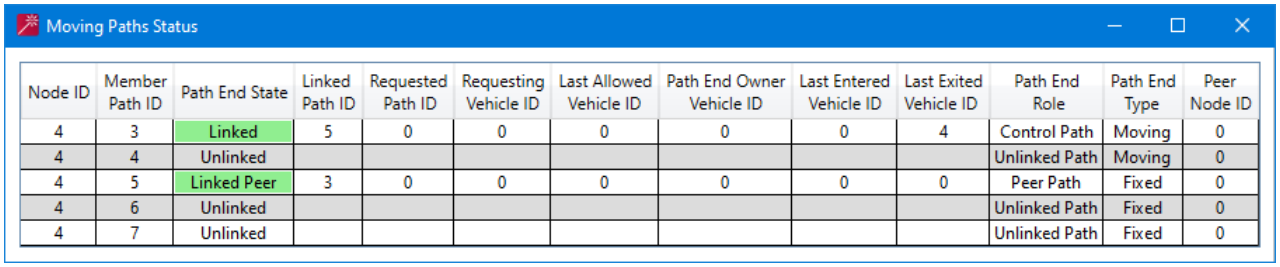
IMPORTANT The ends of the moving path and the fixed path must be aligned for vehicle motion before linking.

1. Enter the **Node ID** of the Moving Path Node.
2. Enter the **Last Vehicle ID** for this link to enable automatic unlinking of the junction.

   If only one vehicle is going to move through the junction, specify the Vehicle ID of that vehicle. When multiple vehicles are going to move through the junction, specify the Vehicle ID of the last vehicle.

   To keep automatic unlinking of the junction from occurring, enter 0. This setting requires explicit unlinking.
3. Enter the **Control Path ID** for this link, this path is the Moving Path.
4. Enter the **Peer Path ID** for this link, this path is the Fixed Path.
5. Select **Link**.

   The two path ends are linked, which forms a junction for vehicles to cross as shown in Figure 2-79.



Node ID	Member Path ID	Path End State	Linked Path ID	Requested Path ID	Requesting Vehicle ID	Last Allowed Vehicle ID	Path End Owner Vehicle ID	Last Entered Vehicle ID	Last Exited Vehicle ID	Path End Role	Path End Type	Peer Node ID
4	3	Linked	5	0	0	0	0	0	4	Control Path	Moving	0
4	4	Unlinked								Unlinked Path	Moving	0
4	5	Linked Peer	3	0	0	0	0	0	0	Peer Path	Fixed	0
4	6	Unlinked								Unlinked Path	Fixed	0
4	7	Unlinked								Unlinked Path	Fixed	0

Figure 2-79: Moving Paths Status Window Showing Linked Paths

6. Use *Move to Position* on page 73 to move the vehicle across the junction.

Unlink a Moving Path from a Fixed Path

After all vehicles that are commanded to move through the node have fully crossed the junction, the paths must be unlinked. Unlinking allows the Moving Path to be moved to another position so it can be linked to another Fixed Path. In the **Host Commands** section, select the **Moving Path** tab (see Figure 2-80) to unlink the Moving Path from the Fixed Path to allow movement of the Moving Path. See *Moving Path Tab* on page 179 for detailed descriptions of all items.

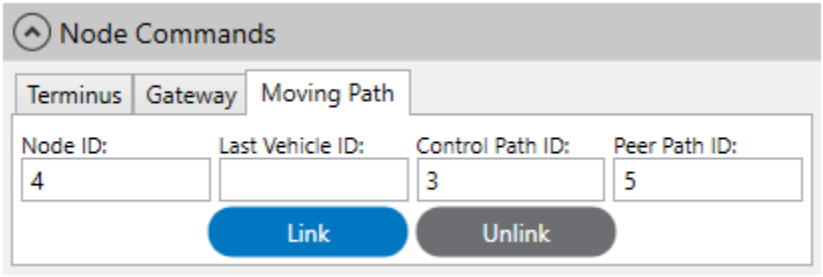
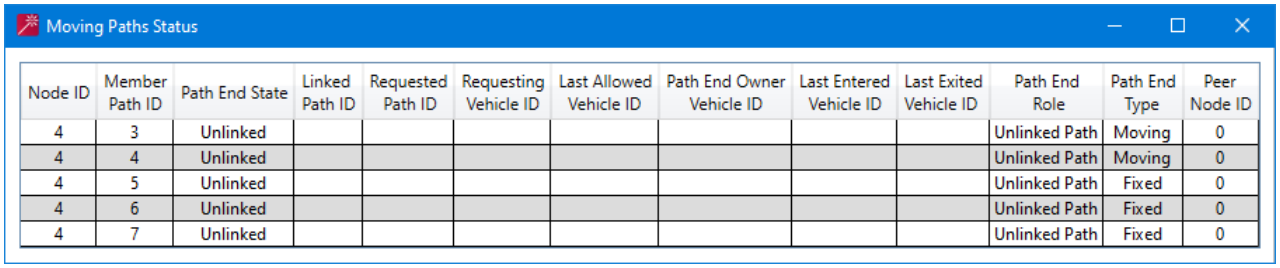


Figure 2-80: Moving Path Unlink Command

1. Enter the **Node ID** of the Moving Path Node.
2. Enter the **Control Path ID** to be unlinked.
3. Select **Unlink**.

   The Control Path end is unlinked from its associated Peer Path, breaking the junction as shown in Figure 2-81. Vehicles are not able to cross the junction unless it is relinked. Once unlinked, the Moving Path can be positioned at a new location.



Node ID	Member Path ID	Path End State	Linked Path ID	Requested Path ID	Requesting Vehicle ID	Last Allowed Vehicle ID	Path End Owner Vehicle ID	Last Entered Vehicle ID	Last Exited Vehicle ID	Path End Role	Path End Type	Peer Node ID
4	3	Unlinked								Unlinked Path	Moving	0
4	4	Unlinked								Unlinked Path	Moving	0
4	5	Unlinked								Unlinked Path	Fixed	0
4	6	Unlinked								Unlinked Path	Fixed	0
4	7	Unlinked								Unlinked Path	Fixed	0

Figure 2-81: Moving Paths Status Window Showing Unlinked Paths

Testing the Transport System – Host Commands

This section describes how to select and perform host-related tasks in the transport system. These tasks include; creating traffic lights, selecting managed node controller configurations, controlling the QSHT motor inverters, creating test stations, and testing digital I/O operation. The examples in this section reference the transport system layout shown in Figure 2-29.

Traffic Lights

Use the **Traffic Light** tab in the **Host Commands** section of the **Settings** panel, which is shown in Figure 2-82, to create, set, or remove traffic lights that are used by NCHost during testing. Once traffic light locations have been verified, edit the Node Controller Configuration File (*node_configuration*.xml) using the MagneMotion Configurator to update or add the traffic lights.

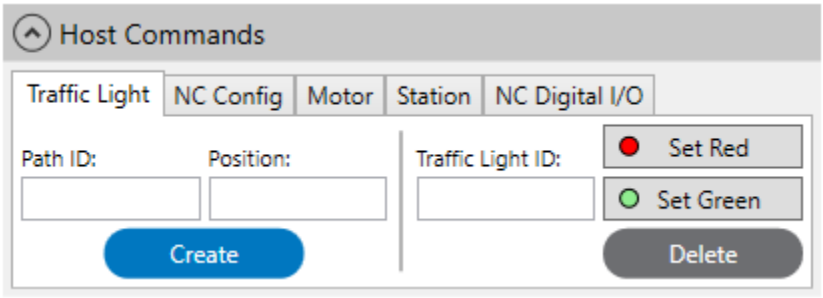


Figure 2-82: Host Commands Section – Traffic Light Tab

NOTE: Traffic lights that are created using the NCHost TCP Interface Utility are temporary and are not restored if the node controller is reset or rebooted.

Create Traffic Lights

1. Open the **Traffic Light Status** window to view the traffic light status if desired (see *Traffic Light Status* on page 54).

2. In the **Host Commands** section, select the **Traffic Light** tab (see Figure 2-82). See *Traffic Light Tab* on page 181 for detailed descriptions of all items.

3. In the **Path ID** field, enter the ID number of the path for the traffic light.

4. In the **Position** field, enter the location from the start of the path, in meters, for the traffic light.

5. Select **Create** to create a temporary traffic light at the specified position on the specified path.

*The traffic light is created with an initial Color of green and assigned the next available Traffic Light ID. If the **Traffic Light Status** window is open, the new traffic light is listed as shown in Figure 2-83.*

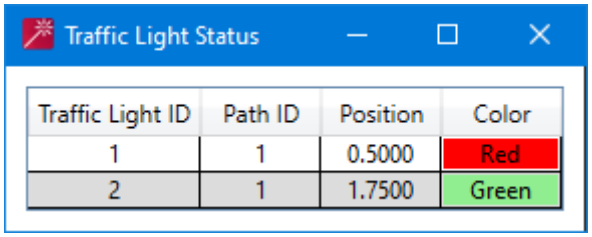


Figure 2-83: Traffic Light Status Window

Set Traffic Lights

1. Open the **Traffic Light Status** window to view the traffic light status if desired (see *Traffic Light Status* on page 54).
2. In the **Host Commands** section, select the **Traffic Light** tab (see Figure 2-82).
3. In the **Traffic Light ID** field, enter the ID of the traffic light to set.
4. Select either the Green or Red button.

   The color of the traffic light changes to the selected color.

Delete Traffic Lights

1. Open the **Traffic Light Status** window to view the traffic light status if desired (see *Traffic Light Status* on page 54).
2. In the **Host Commands** section, select the **Traffic Light** tab (see Figure 2-82). See *Traffic Light Tab* on page 181 for detailed descriptions of all items.
3. In the **TL ID** field, enter the ID of the traffic light to delete.
4. Select **Delete** to remove the specified temporary traffic light.

   *The traffic light is deleted. If the **Traffic Light Status** window is open, the traffic light is removed as shown in Figure 2-84.*

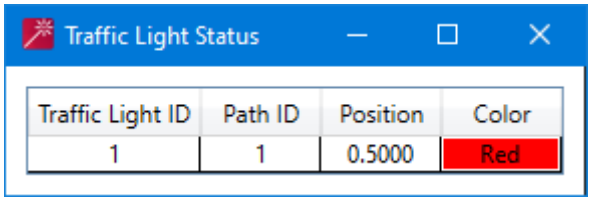


Figure 2-84: Traffic Light Status Window - Updated

NC Configurations

Use the **NC Config** tab in the **Host Commands** section of the **Settings** panel, which is shown in Figure 2-85, to select and activate a Managed Node Controller Configuration File for use during testing. Managed Configuration files must already be uploaded to all node controllers in the system.

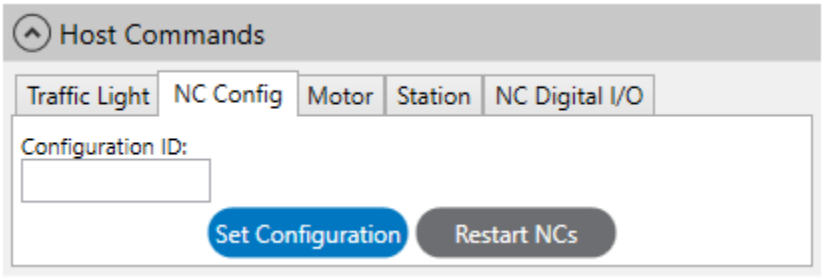


Figure 2-85: Host Commands Section – NC Config Tab

1. Open the **Node Controller Status** window and select **Extended Status** to view the configuration file status if desired (see *Node Controller Status* on page 49).

   If the Configuration ID is shown as zero, Managed Configurations is not being used.

2. In the **Host Commands** section, select the **NC Config** tab (see Figure 2-85). See *NC Config Tab* on page 182 for detailed descriptions of all items.

3. In the **Configuration ID** field, enter the ID number of a valid Managed Node Controller Configuration File that is stored on the HLC. Reference the **Managed Configurations** tab in the NC Web Interface (see the *Node Controller Interface User Manual*, MMI-UM001).

4. Select **Set Configuration** to distribute the selected Managed Node Controller Configuration File to all node controllers.

5. Select **Restart NCs** to force a restart of all node controllers, which causes them to start using the new configuration file.

   *The selected Managed Node Controller Configuration File is activated. If the **Node Controller Status** window is open and **Extended Status** is selected, the Configuration ID and Status is listed as shown in Figure 2-86 once NCHost is reconnected to the HLC.*

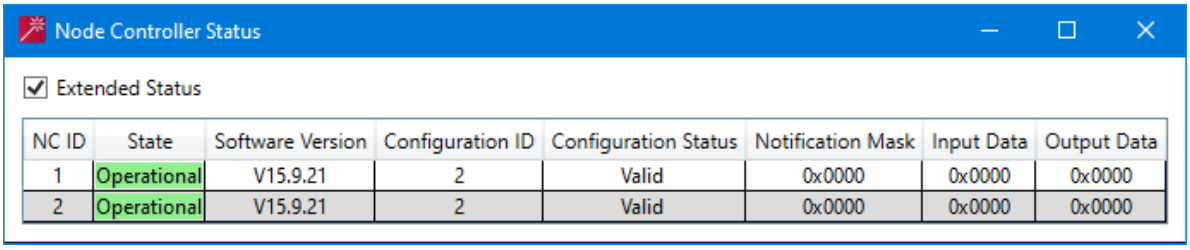


Figure 2-86: Node Controller Status Window Showing Managed Configurations

Motor Inverter Control

Use the **Motor** tab in the **Host Commands** section of the **Settings** panel, which is shown in Figure 2-87, to enable or disable inverters on specified motors.

NOTE: This feature is only available on QSHT configurations.

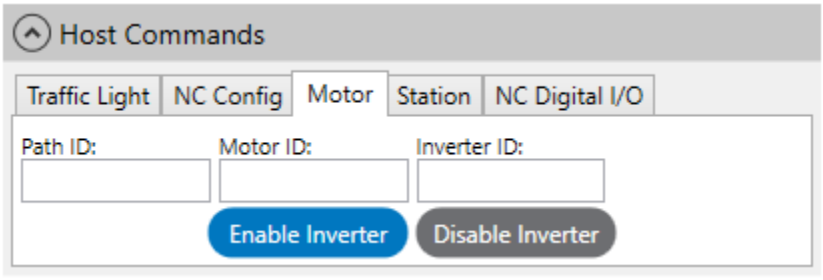


Figure 2-87: Host Commands Section – Motor Tab

1. Open the **Motor Status** window to view the motor status if desired (see *Motor Status* on page 51).

2. In the **Host Commands** section, select the **Motor** tab (see Figure 2-87). See *Motor Tab* on page 183 for detailed descriptions of all items.

3. In the **Path ID** field, enter the ID number of the path where the motor to be commanded is located.

4. In the **Motor ID** field, enter the ID number of the motor to be commanded.

5. In the **Inverter ID** field, enter the ID number of the inverter controlling the blocks to be commanded.

6. Select **Enable Inverter** or **Disable Inverter** to command the specified inverter to that state.

Temporary Stations

Use the **Station** tab in the **Host Commands** section of the **Settings** panel, which is shown in Figure 2-88, to create or remove temporary stations that are used by NCHost during testing. Once station locations have been verified, edit the Node Controller Configuration File (*node_configuration*.xml) using the MagneMotion Configurator to update or add the stations.

NOTE: Stations that are created using the NCHost TCP Interface Utility are temporary and are not restored if the node controller is reset or rebooted.

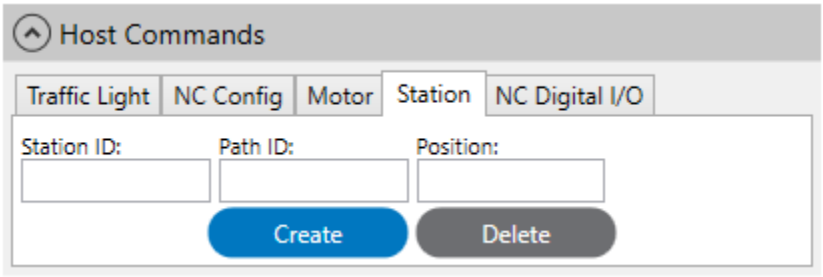


Figure 2-88: Host Commands Section – Station Tab

Create Stations

1. Open the **Stations Status** window to view the station status if desired (see *Station Status* on page 53).

2. In the **Host Commands** section, select the **Stations** tab (see Figure 2-88). See *Station Tab* on page 184 for detailed descriptions of all items.

3. In the **Station ID** field, enter the ID number for the station to be created.

   NOTE: Enter 0 to have the NCHost Utility assign a Station ID.

   Entering an existing Station ID reassigns that station to the new location.

4. In the **Path ID** field, enter the ID number of the path for the station.

5. In the **Position** field, enter the location from the start of the path, in meters, for the station.

6. Select **Create** to create a temporary station at the specified position on the specified path.

   *The specified station is created. If the **Stations Status** window is open, the new station is listed as shown in Figure 2-89.*

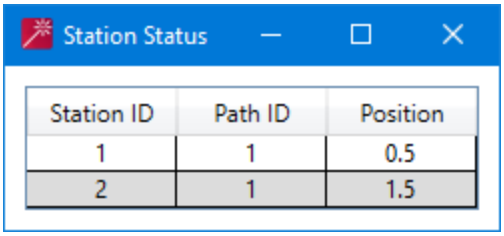


Figure 2-89: Station Status Window, All Stations

7. Use the **Vehicle Commands** to move vehicles to the temporary station.

   NOTE: The **Vehicles** pane shows the destination for the vehicle as the position of the station, not the station ID.

Delete Stations

1. Open the **Stations Status** window to view the station status if desired (see *Station Status* on page 53).

2. In the **Host Commands** section, select the **Stations** tab (see Figure 2-88). See *Station Tab* on page 184 for detailed descriptions of all items.

3. In the **Station ID** field, enter the ID number for the station to be deleted.

   NOTE: Using a Station ID of 0 deletes all stations.

4. Select **Delete** to remove the specified station.

   *The specified station is deleted. If the **Stations Status** window is open, the deleted station is removed from the stations list as shown in Figure 2-90.*

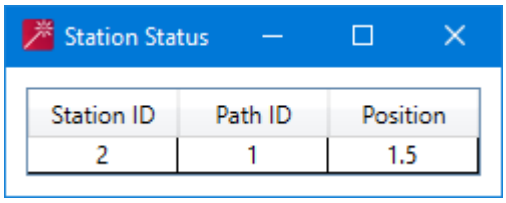


Figure 2-90: Station Status Window, Station Deleted

Node Controller Digital I/O

The digital I/O bits on node controllers with digital I/O can be accessed using the **Node Controller** tab in the **Host Commands** section. The digital I/O is referenced as a set of hexadecimal digits, where each hexadecimal digit maps to a block of 4 bits on the node controller. See the *Node Controller Hardware User Manual*, MMI-UM013, for information on wiring.

NC-12 Digital I/O Mapping

Figure 2-91, shows how the Digital I/O on an NC-12 maps to the four-digit hexadecimal values (0x0000-0xFFFF) used in the **Node Controller** tab. Each hex digit represents 4 bits, with each bit corresponding to a Digital I/O bit (0…15) on the NC-12.

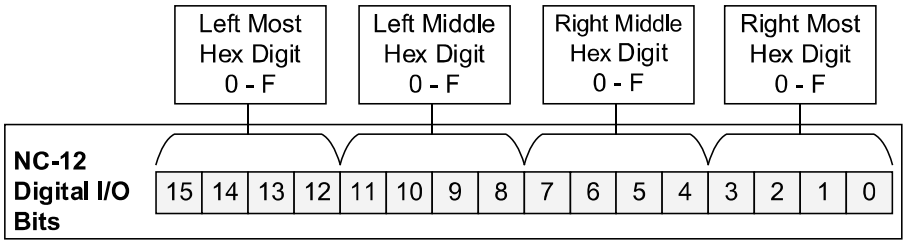


Figure 2-91: NC-12 Hexadecimal to Digital I/O Mapping

Using the Digital I/O Output Bits

1. Open the **Node Controller Status** window to view the node controller data status during Digital I/O operations if desired (see *Node Controller Status* on page 49).

2. In the **Host Commands** section**,** select the **NC Digital I/O** tab (see Figure 2-92). See *NC Digital I/O Tab* on page 185 for detailed descriptions of all items.

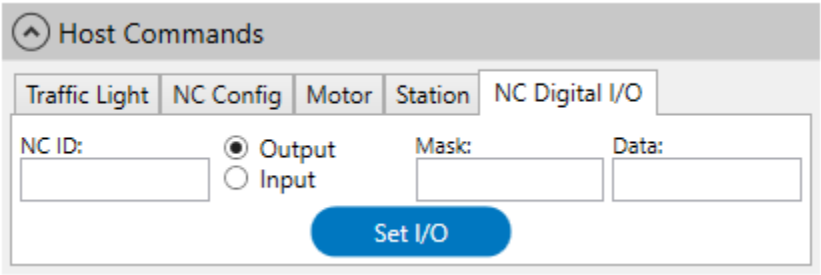


Figure 2-92: Host Commands Section – NC Digital I/O Tab, Output Mode

3. In the **NC ID** field, enter the ID number of the node controller for the Digital I/O operation.

4. Select the **Output** option button.

   *The **Mask** and **Data** fields are displayed.*

5. In the **Mask** field, enter the four-digit hexadecimal value for the output data mask. This mask specifies the data bits to be passed to the output (mask bit set high).

   NOTE: A one enables the specified bit, a zero disables the bit preserving the current output state of the bit.

6. In the **Data** field, enter the four-digit hexadecimal data to be output.

7. Select **Set I/O** to output the data.

   *The data is written to the output bits of the node controller as defined by the **Mask** field. If the **Node Controller Status** window is open, the **Output Data** field is updated showing the data being output through the Digital I/O as shown in Figure 2-93.*

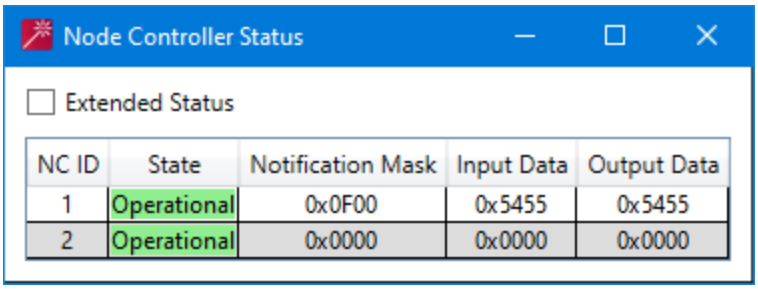


Figure 2-93: Node Controller Status Window, Data Output

The status of the Output command is written to the log file for the node controller as shown in Figure 2-94.

```
Oct 17 08:44:17.876694 INFO: Command: 'Set NC Digital I/O Output': Sta-
tus Accepted, NC ID 1, Output Mask: 0xF0F0, Output Data: 0x9A9A
```

Figure 2-94: NC-12 Digital I/O Output Log Message

NOTE: When running in simulation mode, the Digital I/O Output operations write the contents of the **Output Data** field (with the **Mask** applied) to the **Input Data** field.

Using the Digital I/O Input Bits

1. In the **Host Commands** section**,** select the **Node Controller** tab (see Figure 2-95). See *NC Digital I/O Tab* on page 185 for detailed descriptions of all items.

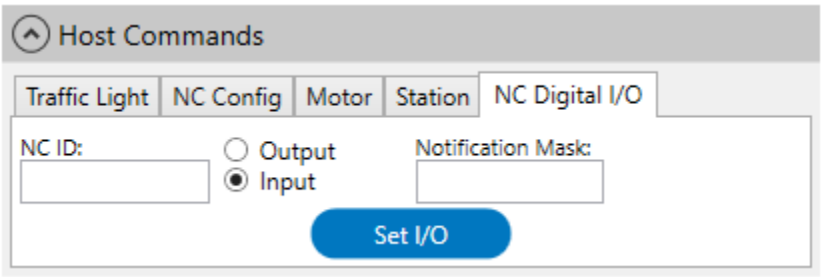


Figure 2-95: Host Commands Section – NC Digital I/O Tab, Input Mode

2. In the **NC ID** field, enter the ID number of the node controller for the Digital I/O operation.

3. Select the **Input** option button.

   *The **Notification Mask** field is displayed.*

4. In the **Notification Mask** field, enter the four-digit hexadecimal value for the input notification mask. This mask specifies the data bits to be monitored for sending a notification message to the HLC when a state change is detected (mask bit set high).

   NOTE: A one enables the bit notification, a zero disables the bit notification.

5. Select **Set I/O** to change the input data **Notification Mask**.

   *The input data **Notification Mask** is updated. If the **Node Controller Status** window is open, the **Notification Mask** field is updated showing the data bits being monitored for sending notification messages to the HLC as shown in Figure 2-96.*

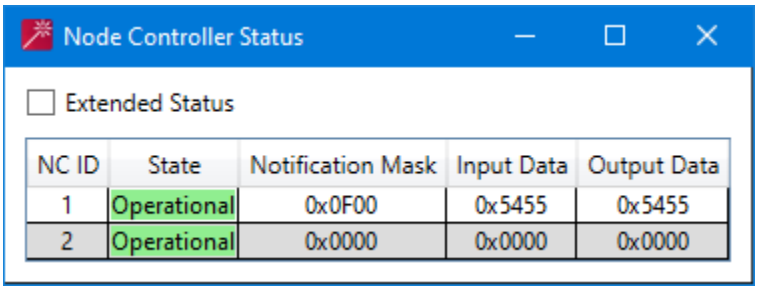


Figure 2-96: Node Controller Status Window, Data Input

The status of the Input command is written to the log file for the node controller as shown in Figure 2-97.

```
Oct 17 08:44:17.876694 INFO: Command: 'Set NC Digital I/O Input': Sta-
tus Accepted, NC ID 1, Notification Mask: 0x0F00
```

Figure 2-97: NC-12 Digital I/O Input Log Message

NOTE: When running in simulation mode, the Digital I/O Output operations write the contents of the **Output Data** field to the **Input Data** field and the **Notification Mask** is not used.

Testing the Transport System – Demo Scripts

This section describes how to run a Demo Script. These text files can be created to automate testing of a transport system, a section of the system, specific paths, or vehicles (see *Demo Scripts* on page 113). Demo Scripts are only used for basic vehicle motion testing and demonstrations of operation.

Run a Script

1. In the **Demo Script** section, select **Browse** (see Figure 2-98). See *Demo Scripts* on page 186 for detailed descriptions of all items.

   *The **Open** dialog box is displayed.*

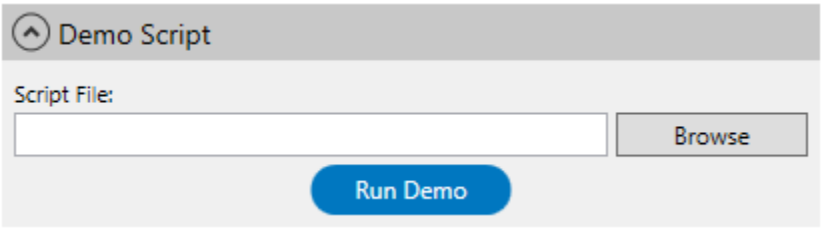


Figure 2-98: Demo Script Section

2. Use the dialog box controls to select the appropriate Demo Script file and select **Open**.

   *The path and name of the Demo Script file is displayed in the **Script File** text field.*

3. Select **Run Demo**.

   *The **Run Demo** button changes to **Stop Demo**. The Demo Script runs until it reaches a stopping point in the script, or until it is stopped manually.*

   NOTE: Stopping the NCHost TCP Interface Utility causes the Demo Script to stop.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

Stop A Script

1. In the **Demo Script** section, select **Stop Demo**.

   *The **Stop Demo** button changes to **Run Demo**, and all vehicle motion stops once the current command completes.*

Testing the Transport System – Refining the Control Loop

This section describes how to perform basic tuning of the PID Control Loop used for vehicle motion. The default Node Controller Configuration File supplies nominal values for the Control Loop that governs vehicle motion. To optimize vehicle motion, these values can be adjusted to fit the specific configuration of the transport system.

Once the correct PID values have been determined, they can be entered into the Node Controller Configuration File. The PID values are typically set in the Motor Defaults section, which applies the settings to all motors on the path, or for specific motors.

NOTE: Any change to the PID values using the NCHost TCP Interface Utility are only applied to the specified path and are only effective until the path is reset. These changes do not change the values in the Node Controller Configuration File. Resetting the paths reloads the values from the Configuration File.

1. Create and run a Demo Script that has a vehicle move repeatedly along the same path.

   NOTE: For comprehensive information about Demo Scripts, see *Demo Scripts* on page 113. See *Run a Script* on page 110 for instructions on how to load, start, and stop Demo Scripts. See *Single Vehicle Examples* on page 116 for single vehicle, single path, Demo Script examples.

2. In the **Engineering Tools** section, select **Control Loop** (see Figure 2-99). See *Engineering Tools* on page 187 for detailed descriptions of all items.

   *The **Tune Control Loop Parameters** dialog box is displayed as shown in Figure 2-100.*

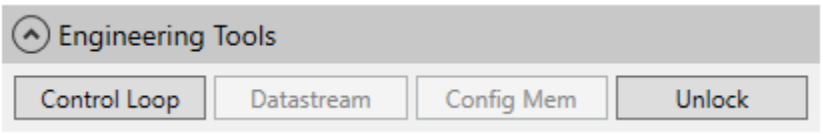


Figure 2-99: Engineering Tools Selections

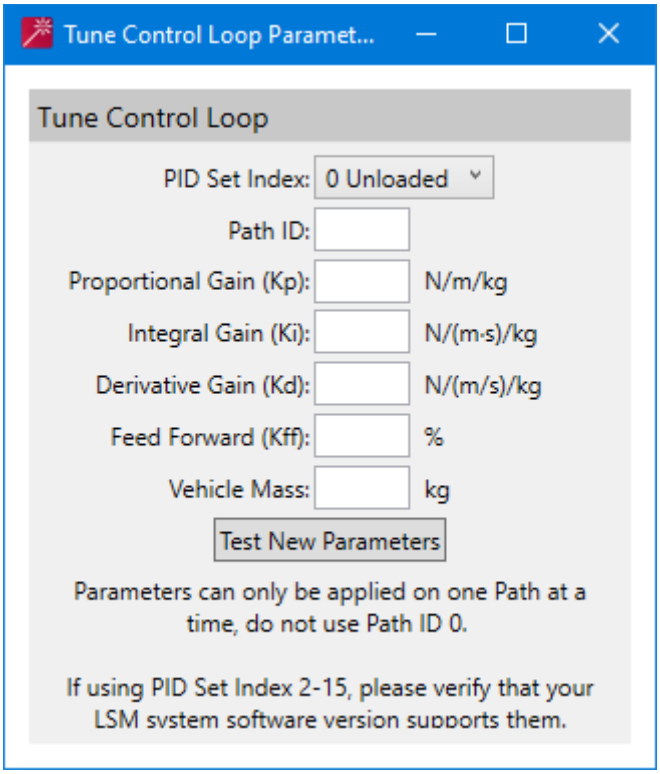


Figure 2-100: Tune Control Loop Parameters Dialog Box

3. Enter the ID of the path to be used for testing vehicle motion.

4. Enter new values for the PID settings.

   NOTE: See the Node Controller Configuration File for the current values.

   Vehicle Mass must be set to either the mass of the loaded vehicle (V*ehicle* + *MagnetArray* + *Payload*) or the mass of the unloaded vehicle (*Vehicle* + *MagnetArray*).

5. Select **Test New Parameters** and observe how the vehicle moves and stops.

6. Repeat Step 4 and Step 5 as necessary to achieve the vehicle motion that is desired for the application.

7. Update the transport system with the new PID values.

   A. Edit the Node Controller Configuration File (*node_configuration*.xml) using the MagneMotion Configurator with the new PID values.

   B. Upload the updated Node Controller Configuration File to all node controllers.

   C. Restart all node controllers to load the updated Node Controller Configuration File.

   D. Reset all paths to update all motors with the updated PID values.

Demo Scripts 3

Overview

This chapter shows how to create and use Demo, Scripts. Demo Scripts are text files that are used by the NCHost TCP Interface Utility to automate testing of a transport system, a section of the system, specific paths, or vehicles.

NOTE: Demo Scripts are only for basic vehicle motion testing and demonstrations of operation. A user-supplied host controller must be used for all actual transport system control within the production environment.

Included in this chapter are:

- Demo Script Overview.
- Script Setup.
- Single and multiple-path transport system examples.
- Creating complex scripts.

Demo Script Overview

This section describes the basic structure and components of a Demo Script. The other sections in this chapter show examples of Demo Scripts, including basic vehicle motions. The *Demo Script Command Reference*, which starts on page 197 describes all Demo Script commands. See *Testing the Transport System – Demo Scripts* on page 110 for information on script execution.

CRUSH HAZARD: Moving mechanisms have no obstruction sensors.

Do not run Demo Scripts to operate the transport system without barriers in place or personal injury could result from the squeezing or compression of fingers, hands, or other body parts between moving mechanisms.

AUTOMATIC MOTION HAZARD: Demo scripts can cause unexpected motion.

Do not run Demo Scripts to operate the transport system without barriers in place or personal injury could result.

Demo Script Example

The following basic Demo Script shows the general format of a simple script for two vehicles on a straight track at least 5 meters long. The examples that are shown in this manual are for one or two vehicles, however the scripts can be configured for any number of vehicles.

```
# Global Directives
Paths 1
Acceleration 5.0
Velocity 2.5

# Vehicle Directives
# 1
Vehicle
    DemoStep 1.0
    DemoStep 2.0
# 2
Vehicle
    DemoStep 4.0
    DemoStep 5.0
```

Figure 3-1: Demo Script Example

The examples in this chapter provide the details of controlling vehicle motion. The following table explains the previous Demo Script, line by line. All terms are defined in the *Demo Script Command Reference* on page 197.

Table 3-1: Demo Script Example – Line Descriptions

Demo Script Line	Description
`# Global Directives`	This line is a comment. It identifies this section of the script file as the *Global Directives*. These commands define the general behavior characteristics for all vehicles.
`Paths 1`	This line defines the script as only applying to path 1. This script uses only one path, however scripts can include all paths in the transport system.
`Acceleration 5.0`	This line specifies that all vehicles in the transport system accelerate/decelerate at 5.0 meters/second^2.
`Velocity 2.5`	This line specifies that all vehicles have a maximum speed of 2.5 meters/second.
`# Vehicle Directives`	This line is a comment. It identifies this section of the script file as the *Vehicle Directives*. These commands define the individual behavior characteristics for each vehicle.
`# 1`	This line is a comment. It identifies this section as the first vehicle.
`Vehicle`	This line specifies that all commands between it and the next Vehicle section apply to the first vehicle* found on the path.
`DemoStep 1.0`	This line commands the vehicle to move from its current position to a position 1.0 meters from the start of path 1.
`DemoStep 2.0`	This line commands the vehicle to move from its current position to a position 2.0 meters from the start of path 1.
`# 2`	This line is a comment. It identifies this section as the second vehicle.
`Vehicle`	This line specifies that all commands following it apply to the second vehicle* found on the path.
`DemoStep 4.0`	This line commands the vehicle to move from its current position to a position 4.0 meters from the start of path 1.
`DemoStep 5.0`	This line commands the vehicle to move from its current position to a position 5.0 meters from the start of path 1.

* The NCHost Interface Utility identifies all vehicles on a path starting from the zero position at the upstream end of the path. The utility assigns the first vehicle section in the script to the first vehicle detected on the first path listed. The utility continues to sections in the script to the vehicles on the paths according to the order the vehicles are detected during startup. Vehicle 1 in the Demo Script may not correspond to the transport system vehicle with ID 1.

See *Vehicle Directives* on page 209 for the specifications for the DemoStep command, such as velocity, acceleration, delay periods, and direction options.

Single Vehicle Examples

Examples are provided in *Single Path Track Examples* on page 118 and *Multiple Path Track Examples* on page 124 that show the basic movement of one vehicle. These examples use a straight track with one path and a looped track with multiple paths. Since there is only one vehicle, there is no need to synchronize its movements with other vehicles.

Multiple Vehicle Examples

Examples are provided in *Single Path Track Examples* on page 118 and *Multiple Path Track Examples* on page 124 that show the basic movement of two vehicles. These examples use a straight track with one path and a looped track with multiple paths. Since there are multiple vehicles, their movement can be either dependent on, or independent of the other vehicles, which are based on their requirement to be synchronized with the other vehicles.

NOTE: When multiple vehicles are used, all vehicles must have the same number of Demo-Step commands.

Script Setup

Script Header

All scripts require a header that defines the environment for execution of the script. The header consists of the Global Directives that are applied to all vehicles. See *Global Directives* on page 200 for detailed descriptions of these commands. The NCHost TCP Interface Utility uses the default values for any Global Directives that are not included in the header.

The script header must contain at a minimum the following command:

```
Paths 1
```

Script Body

All scripts require a body that defines all vehicles on the paths that are included in the script and their movement. Demo Scripts cannot be run if the number of Vehicle sections in the script do not match the number of vehicles on the paths that are included in the script. See *Vehicle Directives* on page 209 for detailed descriptions of the vehicle commands.

The script body must contain at a minimum the following commands:

```
Vehicle
    DemoStep 0.5
```

Comments

Comments can be used within a script to identify the script and the various sections of the script as shown in the following example and in Figure 3-1. Comments can also be used to improve readability of the script, to add and remove functions, to identify vehicles, and to describe vehicle motion. Any line that starts with a “#” is considered a comment and is ignored.

NOTE: Insert blank lines, spaces, and tabs into the script as desired to improve readability.

```
# Demo Script

Paths 1

Vehicle
    DemoStep 0.5
```

Single Path Track Examples

The following single path script examples feature a straight track 6 meters long, which shown in Figure 3-2. This track has one path and a Simple Node, which is the starting point of the path.

See *Global Directives* on page 200 for detailed descriptions of the individual global vehicle commands that are used in these examples, such as Paths, Velocity, Acceleration, and Direction. See *Vehicle Directives* on page 209 for detailed descriptions of the DemoStep command options.

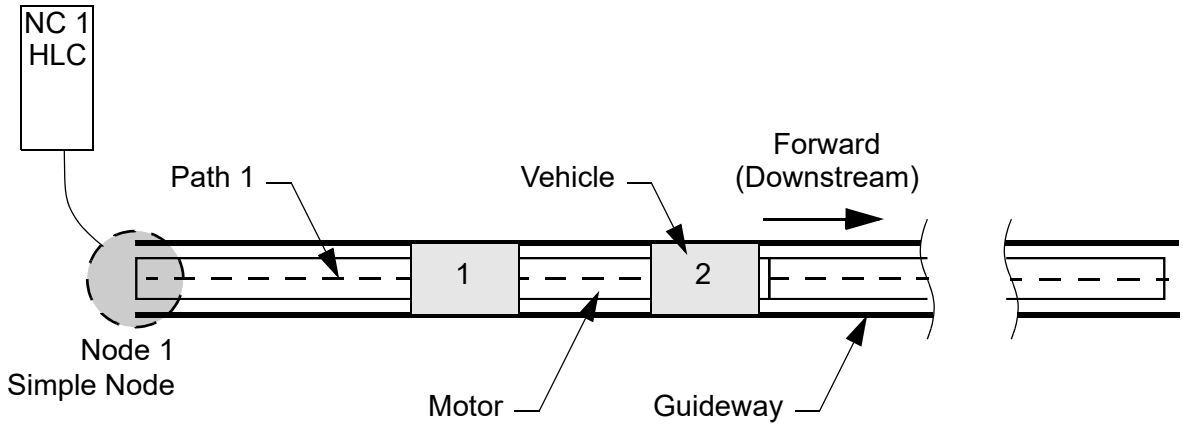


Figure 3-2: Straight Track for Demo Script Examples

One Vehicle

This example demonstrates movement of one vehicle on one path (see Figure 3-2 for the track configuration). The script moves the vehicle on the guideways with an acceleration of 5.0 m/s^2, and a velocity of 2.5 m/s.

When the script is run, the vehicle moves to the 1.0 meter position on path 1 and then to the 4.0 meter position on path 1. Once the vehicle arrives at each position, it moves to the other position. This movement repeats as shown in Figure 3-3 until the script is stopped.

```
# Global Directives
Paths 1
Velocity 1.5
Acceleration 1.0

# Vehicle Directives
Vehicle
    DemoStep 1.0
    DemoStep 4.0
```

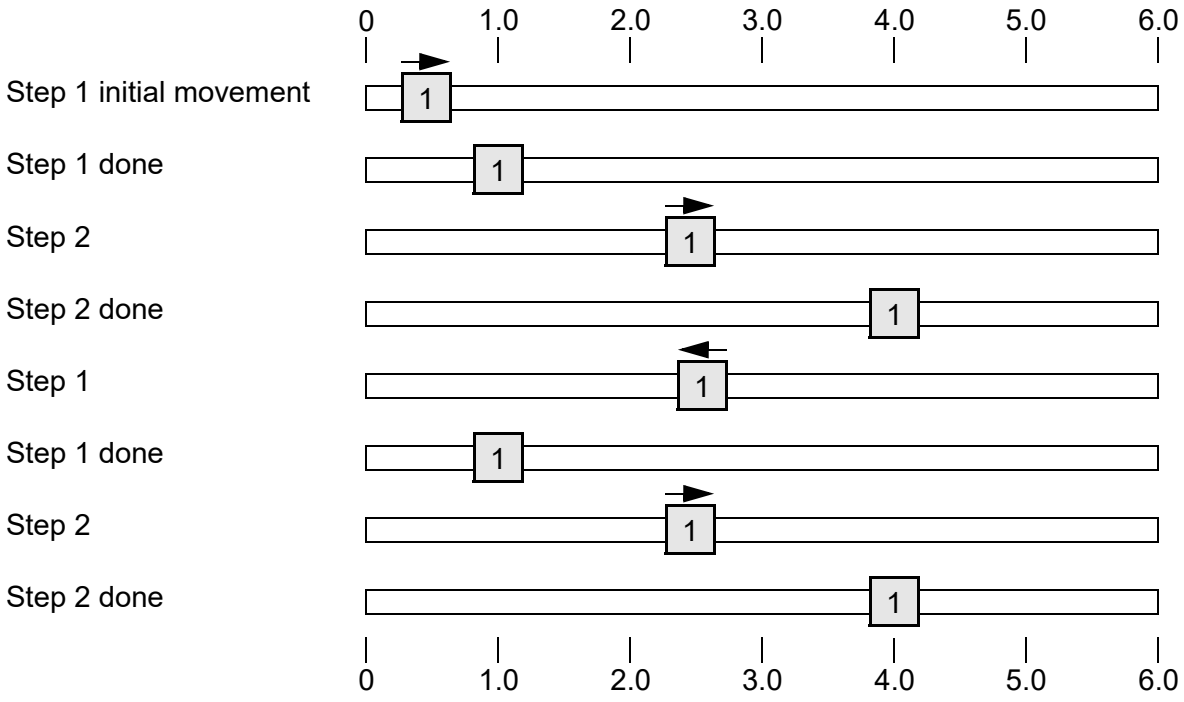


Figure 3-3: One Vehicle, Single Path

Two Vehicles, Dependent Movement

This example demonstrates dependent movement of two vehicles on one path (see Figure 3-2 on page 118 for the track configuration). The script moves both vehicles with the defined acceleration and velocity. Since the movement of each vehicle is dependent on the other vehicles, all vehicles are required to complete a step before the next step begins executing.

NOTE: The term synchronous as used within this section is unrelated to the behavior and operation of the MagneMotion® LSM Sync product (SYNC IT™). For a comprehensive description of the LSM Sync product, see the Theory of Operation section, within the *LSM Synchronization Option User Manual*, MMI-UM005.

Dependent movement using `Demo_Arrival_Method Together` is the default, and is only shown in the example for reference.

When the script is run, the vehicle closest to the beginning of path 1 is designated as the first vehicle in the script. The next vehicle is then designated as the second vehicle in the script. Both vehicles move simultaneously, the first vehicle moves to the 1.0 meter position and the second vehicle moves to the 2.0 meter position. Once both vehicles arrive at their first position, both vehicles move again simultaneously. The first vehicle moves to the 4.0 meter position and the second vehicle moves to the 5.0 meter position. Once both vehicles arrive at their second position these movements repeat, as shown in Figure 3-4 until the script is stopped.

```
# Global Directives
Paths 1
Acceleration 1.0
Velocity 1.5
Direction Bi-Directional
Demo_Arrival_Method Together

# Vehicle Directives
Vehicle
    DemoStep 1.0
    DemoStep 4.0

Vehicle
    DemoStep 2.0
    DemoStep 5.0
```

NOTE: The Vehicle Directive designations do not have any correspondence with the actual Vehicle IDs as assigned by the HLC for the transport system and do not affect those IDs.

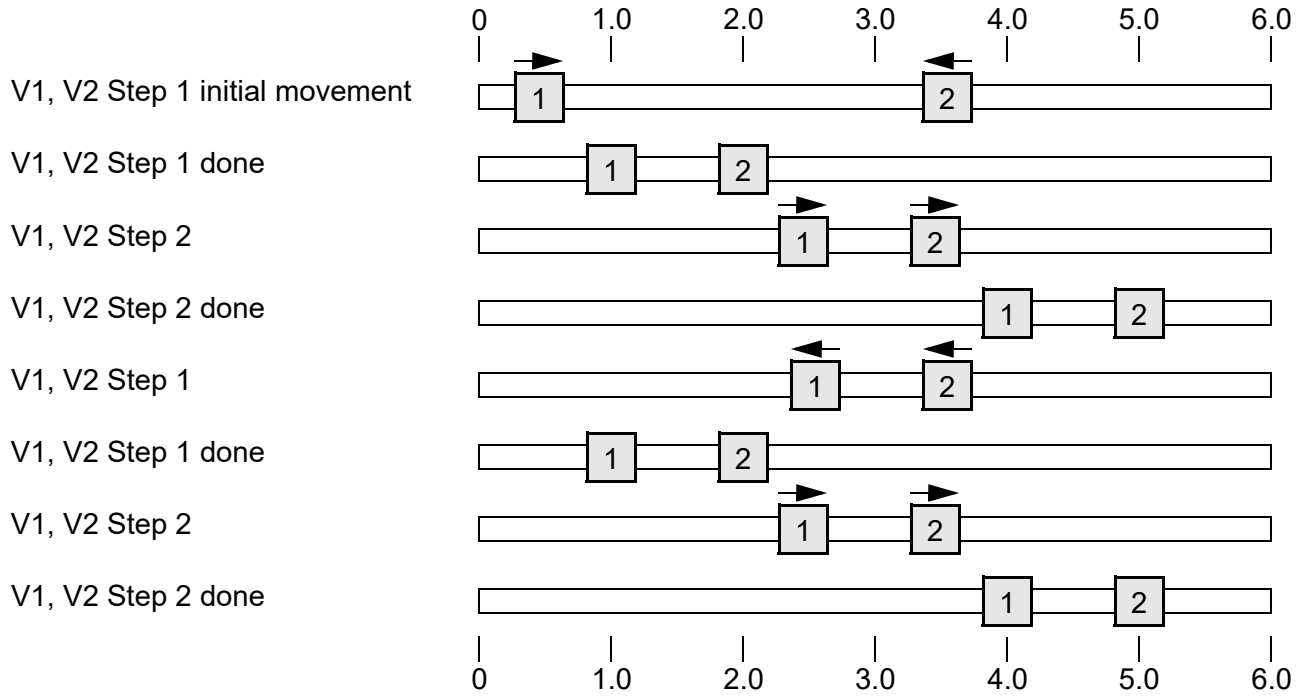


Figure 3-4: Two Vehicles, Single Path Dependent Movement

Two Vehicles, Independent Movement

This example demonstrates independent movement of two vehicles on one path (see Figure 3-2 on page 118 for the track configuration). The script moves both vehicles with the defined acceleration and velocity. Since all movement is independent of other vehicles, each vehicle completes its steps independently of the other vehicle with no requirement that the vehicles stay synchronized.

When the script is run, the vehicle closest to the beginning of path 1 is designated as the first vehicle in the script. The next vehicle is then designated as the second vehicle in the script. Both vehicles move simultaneously, the first vehicle moves to the 1.0 meter position and the second vehicle moves to the 3.0 meter position. Once either vehicle arrives at its first position, it moves to its second position. Once either vehicle arrives at its second position, it moves to its first position. These movements repeat as shown in Figure 3-5 until the script is stopped.

NOTE: If the vehicles travel over the same track section on a straight track while using Arrival Method Single, they may become immovable because each vehicle blocks the movement of the other.

```
# Global Directives
Paths 1
Acceleration 1.0
Velocity 1.5
Direction Bi-Directional
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 1.0
    DemoStep 2.0

Vehicle
    DemoStep 3.0
    DemoStep 5.0
```

NOTE: The Vehicle Directive designations do not have any correspondence with the actual Vehicle IDs as assigned by the HLC for the transport system and do not affect those IDs.

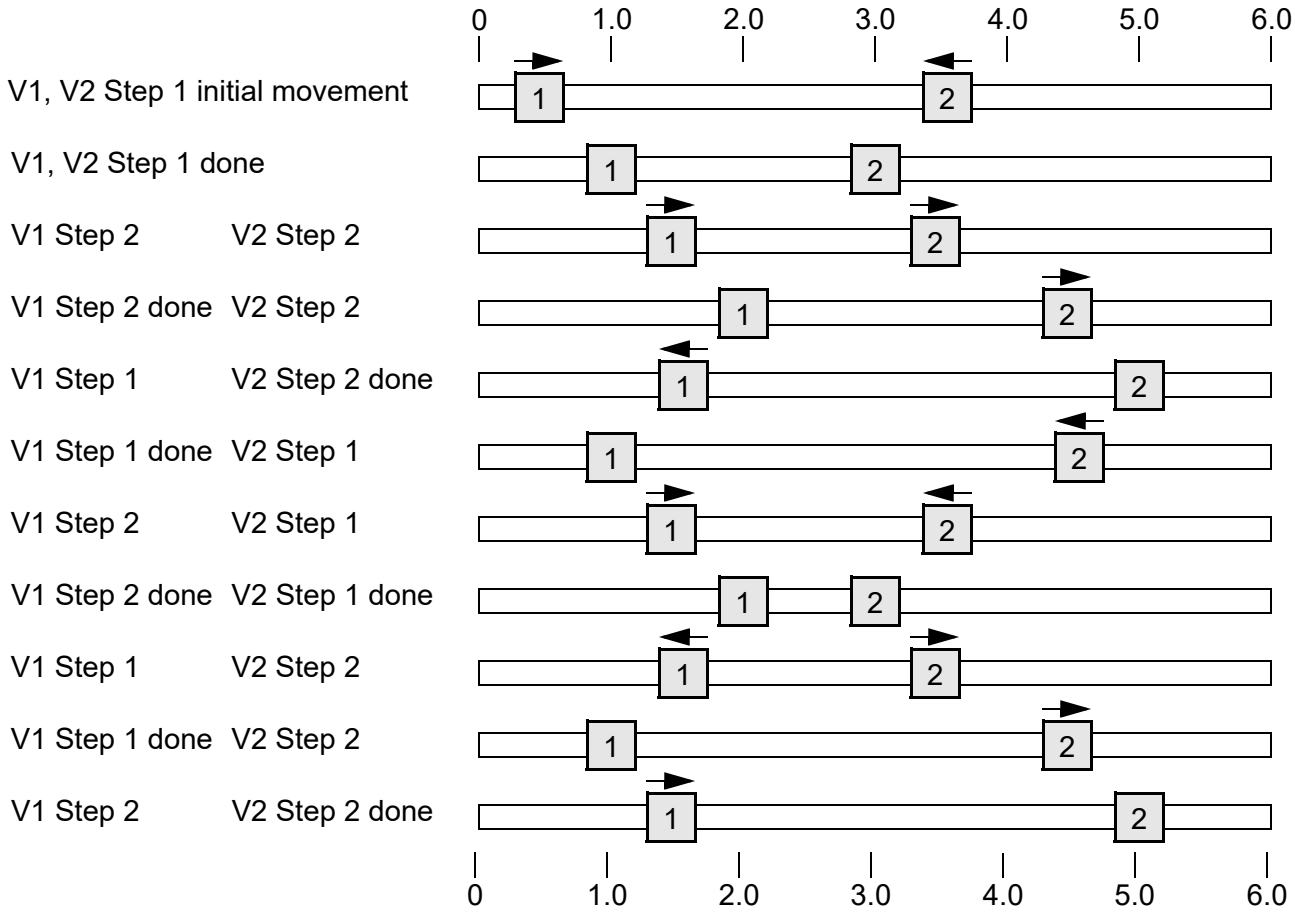


Figure 3-5: Two Vehicles, Single Path Independent Movement

Multiple Path Track Examples

The following multiple path script examples feature a two loop track with four paths, a Merge Node, and a Diverge Node as shown in Figure 3-6.

Since this track consists of two closed loops, the vehicle can move to any position by moving Forward (downstream) or Backward (upstream). When a specific direction of travel is required, define the **Direction** in the Global Directives as either Forward, Backward, or Bi-Directional. Bi-Directional is the default if the **Direction** command is not used in the Global Directives to specify another direction option.

See *Global Directives* on page 200 for detailed descriptions of the individual global vehicle commands that are used in these examples, such as Paths, Velocity, Acceleration, and Direction. See *Vehicle Directives* on page 209 for detailed descriptions of the DemoStep command options.

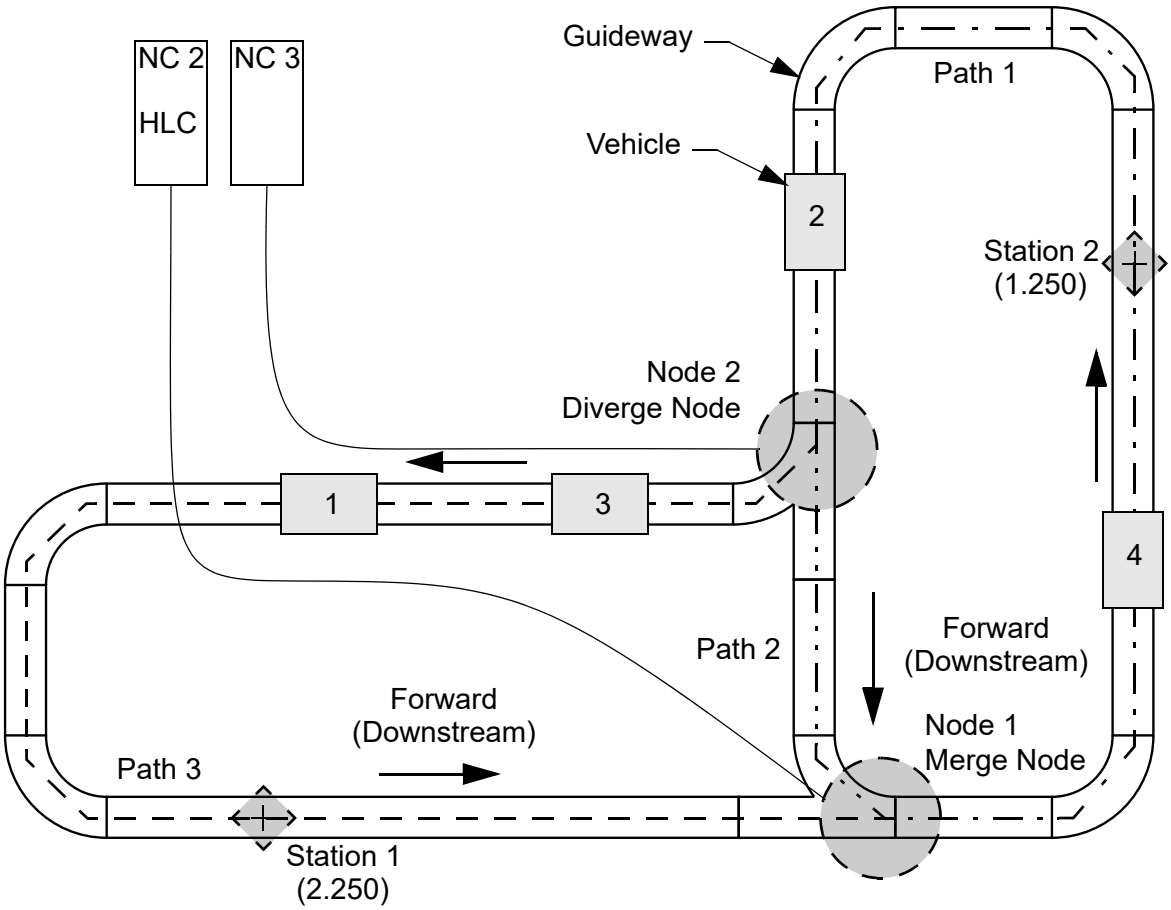


Figure 3-6: Merge/Diverge Transport System for Demo Script Examples

One Vehicle

This example demonstrates movement of one vehicle on all paths (see Figure 3-6 on page 124 for the track configuration). The script moves the vehicle forward with the defined acceleration and velocity.

When the script is run, the vehicle starts by moving to the 2.5 meter position on path 1. At the next step, it moves straight through the Diverge Node to the 0.35 meter position on path 2. At the next step it moves through the Merge Node to path 1, around path 1, right through the Diverge Node to path 3, around path 3 to the 2.5 meter position. In the last step, the vehicle moves from the position on path 3 through the merge to path 1, around path 1, and right through the Diverge Node to the 0.75 meter position on path 3. These movements repeat as shown in Figure 3-7 until the script is stopped.

Once the vehicle arrives at each position, it moves to its next position. All motion is always in the forward direction. Using forward motion causes the vehicle to keep going around the track. This movement repeats until the script is stopped.

NOTE: If vehicles moving across multiple paths can reach their destination moving forward, they always use the Forward direction when Direction is set to Bi-Directional. This direction selection can cause the vehicle to travel farther than in the Backward direction to arrive at a destination.

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Forward

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2
    DemoStep 2.5 Path 3
    DemoStep 0.75 Path 3
```

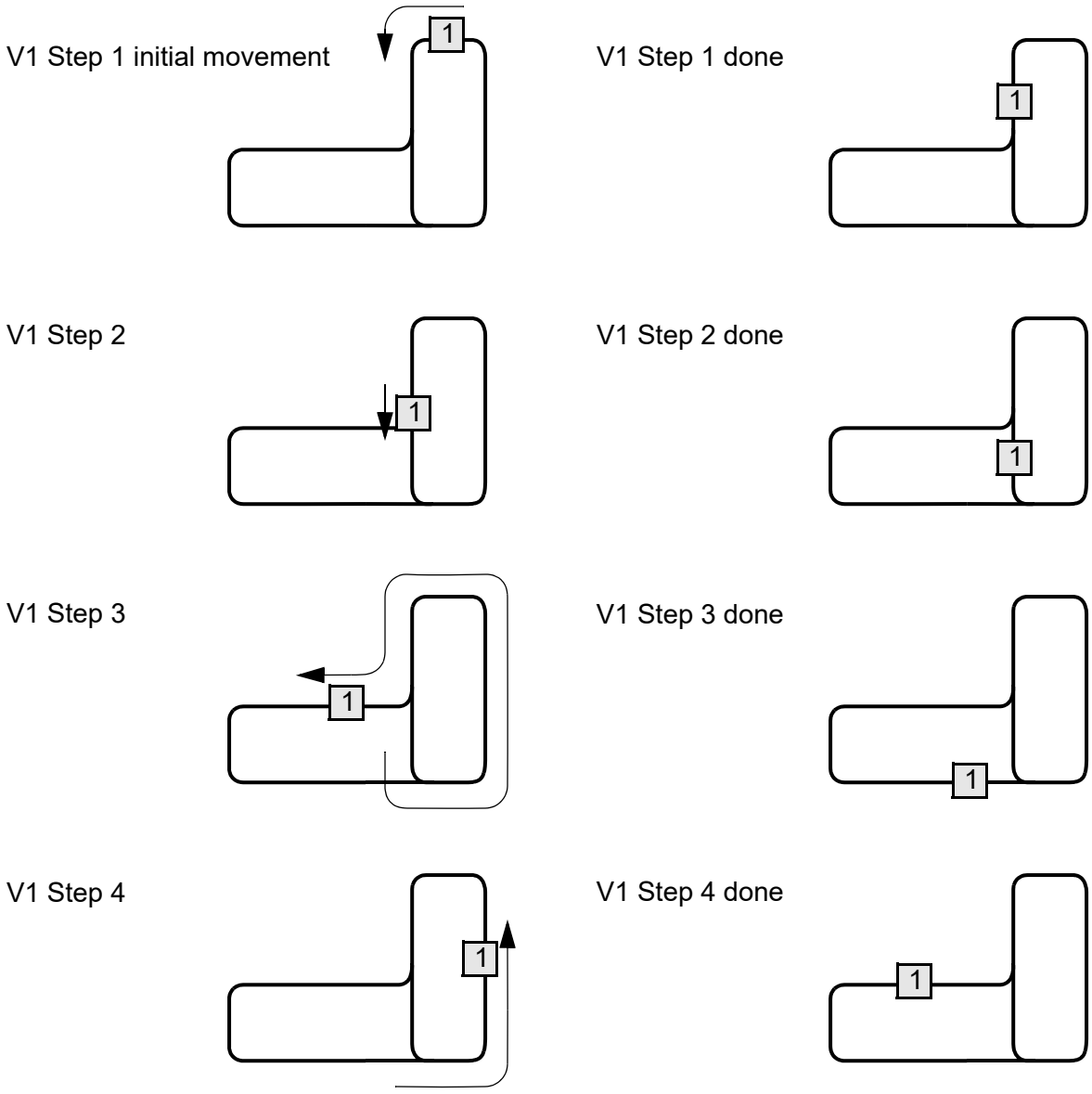


Figure 3-7: One Vehicle, Multiple Path Movement

Two Vehicles, Dependent Movement

This example demonstrates dependent movement of two vehicles on all paths (see Figure 3-6 on page 124 for the track configuration). The script moves both vehicles forward with the defined acceleration and velocity. Since the movement of each vehicle is dependent on the other vehicles, all vehicles are required to complete a step before the next step begins executing.

NOTE: Dependent movement using `Demo_Arrival_Method Together` is the default, and is only shown in the example for reference.

When the script is run, the vehicle closest to the beginning of the first path in the Paths command is designated as the first vehicle in the script. The next vehicle on the system is designated as the second vehicle in the script. Both vehicles move simultaneously. The first vehicle moves forward to the 2.5 meter position on path 1 and the second vehicle moves forward to the 0.75 meter position on path 3. Once both vehicles arrive at the first position in the script, both vehicles move again simultaneously. The first vehicle moves to the 0.35 meter position on path 2 and the second vehicle moves to the 2.75 meter position on path 3. These movements repeat as shown in Figure 3-8 until the script is stopped.

The script seems to stop running if the first vehicle goes through node 1 (Merge) before the second vehicle. In this case, the first vehicle arrives at its destination, which keeps the second vehicle from arriving at its destination. Since the second vehicle never arrives at its destination, the step cannot complete and the script cannot continue to the next command.

NOTE: If vehicles moving across multiple paths can reach their destination moving forward, they always use the Forward direction when Direction is set to Bi-Directional. This direction selection can cause the vehicle to travel farther than in the Backward direction to arrive at a destination.

```
# Global Directives
Paths 1 2 3
Acceleration 5.0
Velocity 2.5
Direction Forward
Demo_Arrival_Method Together

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2

Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 3
```

NOTE: The Vehicle Directive designations do not have any correspondence with the actual Vehicle IDs as assigned by the HLC for the transport system and do not affect those IDs.

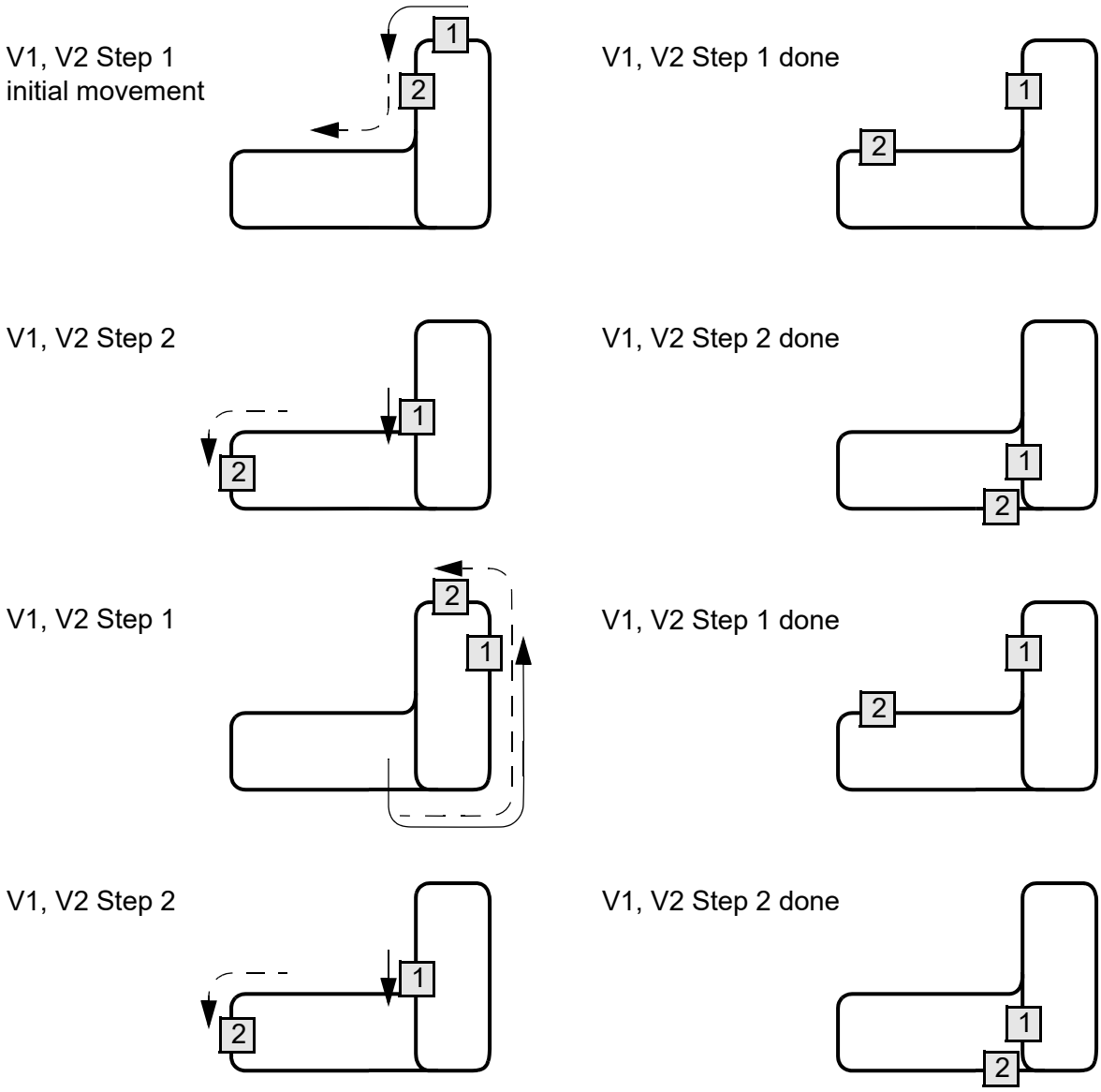


Figure 3-8: Two Vehicles, Multiple Path Dependent Movement

Two Vehicles, Independent Movement

This example demonstrates independent movement of two vehicles on all paths (see Figure 3-6 on page 124 for the track configuration). The script moves both vehicles forward with the defined acceleration and velocity. Since movement of each vehicle is independent of other vehicles, each vehicle completes its steps independently of the other vehicle with no requirement that the vehicles stay synchronized.

When the script is run, the vehicle closest to the beginning of the first path in the Paths command is designated as the first vehicle in the script and the next vehicle on the path is designated as the second vehicle in the script. Both vehicles move simultaneously, with the first vehicle moving to the 2.5 meter position on path 1. At the next step, it moves straight through the Diverge Node to the 0.35 meter position on path 2. The second vehicle starts by moving to the 0.75 meter position on path 3. At the next step, it moves to the 2.75 meter position on path 3.

Once either vehicle arrives at its first position, it moves around the track to its second position. Once either vehicle arrives at its second position, it moves around the track to its first position. The two vehicles move such that one vehicle may obstruct the movement of the other vehicle for a short amount of time. These movements repeat as shown in Figure 3-9 until the script is stopped.

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Forward
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2

Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 3
```

NOTE: The Vehicle Directive designations do not have any correspondence with the actual Vehicle IDs as assigned by the HLC for the transport system and do not affect those IDs.

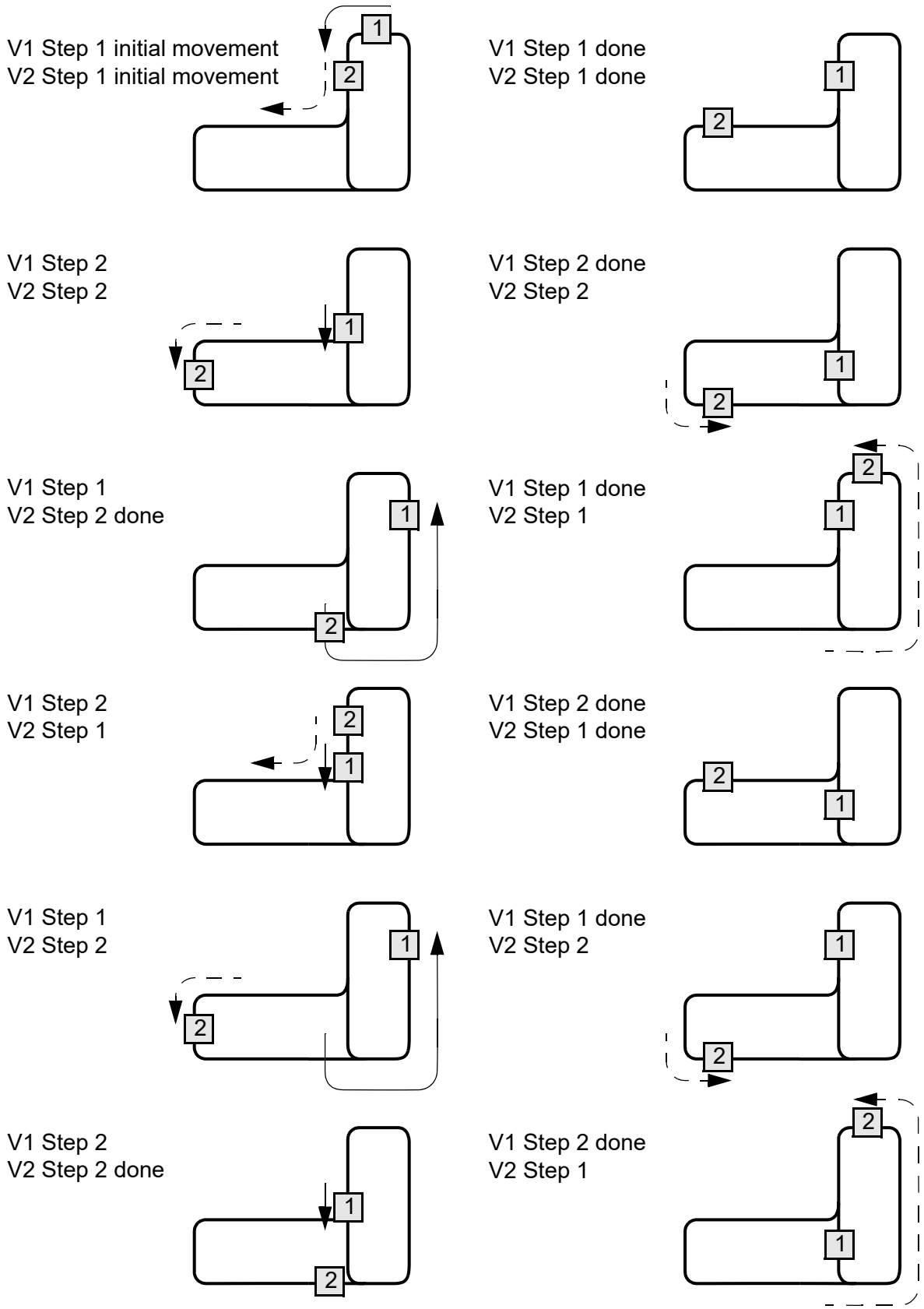


Figure 3-9: Two Vehicles, Multiple Path Independent Movement

Additional Command Examples

The following scripts reference the track that is shown in Figure 3-6. See *Global Directives* on page 200 for detailed descriptions of the individual header commands that are used in these examples, such as Paths, Velocity, Acceleration, and Direction. See *Vehicle Directives* on page 209 for detailed descriptions of the DemoStep commands and Command Extensions.

Stop Script After One Cycle

This example demonstrates movement of two vehicles on all paths (see Figure 3-6 on page 124 for the track configuration) for one pass through the script. The script moves both vehicles forward with the defined acceleration and velocity. All movement is independent, which allows the steps of each vehicle to execute independently of the other vehicle.

When the script is run, both vehicles start simultaneously and move to their first position. Once each vehicle arrives at its first position, it moves to its second position. The third step for the first vehicle causes the script to stop once the vehicle completes its second step. Since the vehicles in the script are running independently, the actual position of the second vehicle is not considered before stopping the script. The **Null** step for the second vehicle is required to make sure that both vehicles have the same number of steps.

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Forward
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2
    DemoStep Stop

Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 3
    DemoStep Null
```

Stop Script After Specified Cycle Count

This example demonstrates movement of two vehicles on all paths (see Figure 3-6 on page 124 for the track configuration) for a specified number of passes through the script. The script moves both vehicles forward with the defined acceleration and velocity. All movement is independent, which allows the steps of each vehicle to execute independently of the other vehicle.

When the script is run, both vehicles start simultaneously and move to their first position. Once each vehicle arrives at its first position, it moves to its second position. The third step for the first vehicle acts as a counter that causes the script to stop once the first vehicle has executed its steps the specified number of times. Since the vehicles in the script are running independently the actual position of the second vehicle, and the number of times its steps have executed is not considered before stopping the script. The **Null** step for the second vehicle is required to make sure that both vehicles have the same number of steps.

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Forward
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2
    DemoStep Stop Iterations 30

Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 3
    DemoStep Null
```

Two Vehicles, Pause at a Rendezvous Step

This example demonstrates independent movement of two vehicles on all paths (see Figure 3-6 on page 124 for the track configuration) with a synchronization step. The script moves both vehicles forward with the defined acceleration and velocity. All movement is independent, which allows the steps of each vehicle to execute independently of any other vehicles.

When the script is run, both vehicles start simultaneously. Once each vehicle arrives at its first position, it moves to the second position. Once either vehicle arrives at its second position, it waits for the other vehicle to arrive at its second position. Once both vehicles are at their second position, both vehicles move simultaneously to their third position. These movements repeat until the script is stopped.

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Forward
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2
    DemoStep Rendezvous Point 1
    DemoStep 1.0 Path 1

Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 4
    DemoStep Rendezvous Point 1
    DemoStep 1.0 Path 3
```

Chain Multiple Scripts

This example demonstrates movement of two vehicles on all paths (see Figure 3-6 on page 124 for the track configuration) using multiple scripts, with the scripts chained to each other. The first script moves both vehicles forward with the defined acceleration and velocity. The second script moves both vehicles backward with the defined acceleration and velocity. All movement is independent, which allows the steps of each vehicle to execute independently of any other vehicles.

When the first script (s1.txt) is run, both vehicles start simultaneously. Once each vehicle arrives at its first position, it moves to its second position. Once the first vehicle arrives at its second position, it stops and the referenced script (s2.txt) starts running. When the second script is run, both vehicles start simultaneously. Once each vehicle arrives at its first position, they move to their second position. Once the first vehicle arrives at its second position, it stops and the referenced script (s1.txt) starts running. These movements repeat until either script is stopped.

NOTE: Both scripts must be in the same folder.

s1.txt

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Forward
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2
    DemoStep Stop Chain
s2.txt

Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 3
    DemoStep Null
```

s2.txt

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Backward
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 3
    DemoStep Stop Chain
s1.txt

Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2
    DemoStep Null
```

Set Velocity, Acceleration, and Direction by Vehicle

This example demonstrates movement of two vehicles on all paths (see Figure 3-6 on page 124 for the track configuration) where the velocity, acceleration, and direction of the vehicles is controlled at the vehicle level. The script moves both vehicles forward with the defined acceleration and velocity. All movement is independent, which allows the steps of each vehicle to execute independently of any other vehicles.

When the script is run, both vehicles start simultaneously using the acceleration and velocity settings that are defined in the header of the script. Once each vehicle arrives at its first position, it moves to the second position using the acceleration and velocity settings for that step. Once the first vehicle arrives at its second position, it moves backwards to its third position using the acceleration and velocity settings of the script. The **Null** step for the second vehicle is required to make sure that both vehicles have the same number of steps. Since the second vehicle does not have any movement commanded in its third step, it starts moving to its first position. These movements repeat until the script is stopped.

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Forward
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1
    DemoStep 0.35 Path 2 Acceleration 2.5
    DemoStep 1.0 Path 1 Direction Backwards

Vehicle
    DemoStep 0.75 Path 3
    DemoStep 2.75 Path 3 Velocity 1.25
    DemoStep Null
```

Combine Command Extensions

This example shows how Command Extensions can be combined with the DemoStep commands to create more complex vehicle motion steps (see Figure 3-6 on page 124 for the track configuration). See Table 5-2 on page 209 for the command structure, which identifies how the Command Extensions can be combined with the DemoStep commands.

The script moves both vehicles with the defined acceleration and velocity. All movement is independent, which allows the steps of each vehicle to execute independently of any other vehicles. The first vehicle constantly moves forward. The second vehicle moves forward to its first stop when the script is started then it moves back and forth between its two stops. If the first vehicle is in the way of the second vehicle moving to its second stop, the second vehicle moves forward around the track to get to the second stop.

For the first vehicle, the path, acceleration, velocity, and direction are specified for the first step. The path and a delay are specified for the second step. The path, a delay, and the direction are specified for the third step.

For the second vehicle, the path, direction, and payload are specified for the first step. The path is specified for the second step and the third step is Null to keep the number of steps the same for each vehicle.

```
# Global Directives
Paths 1 2 3
Acceleration 1.0
Velocity 1.5
Direction Bi-Directional
Demo_Arrival_Method Single

# Vehicle Directives
Vehicle
    DemoStep 2.5 Path 1 Acceleration 0.8 Velocity 1.2
Direction Forward
    DemoStep 0.35 Path 2 Delay 500
    DemoStep 1.0 Path 3 Delay 500 Direction Forward

Vehicle
    DemoStep 1.0 Path 3 Direction Forward Payload Loaded
    DemoStep 2.0 Path 3 Payload Unloaded
    DemoStep Null
```

UI Reference 4

Overview

This chapter provides an overview of the User Interface (UI) for the NCHost TCP Interface Utility. Examples of each window, pane, and dialog in the UI and descriptions of their features are included. The following parts of the User Interface are covered:

- User Interface Window Layout.
- User Interface Features.
- Window and Dialog Box Reference.

NOTE: This User Interface Reference reflects the version of the software that is indicated in *Changes* on page 15. Specific builds of the NCHost TCP Interface Utility may not implement all features that are described in this manual. See the Release Notes that are supplied with this application for more information.

All versions of the NCHost TCP Interface Utility support all MagneMotion® product lines except as stated within this reference.

Window Layout

All User Interface elements that are presented through the NCHost TCP Interface Utility window follow the guidelines that are described in this chapter. The UI elements consist of three main areas that are always visible as shown in Figure 4-1 and temporary dialog boxes that are displayed over the main UI window. Additional windows that are opened through the UI can be positioned as desired.

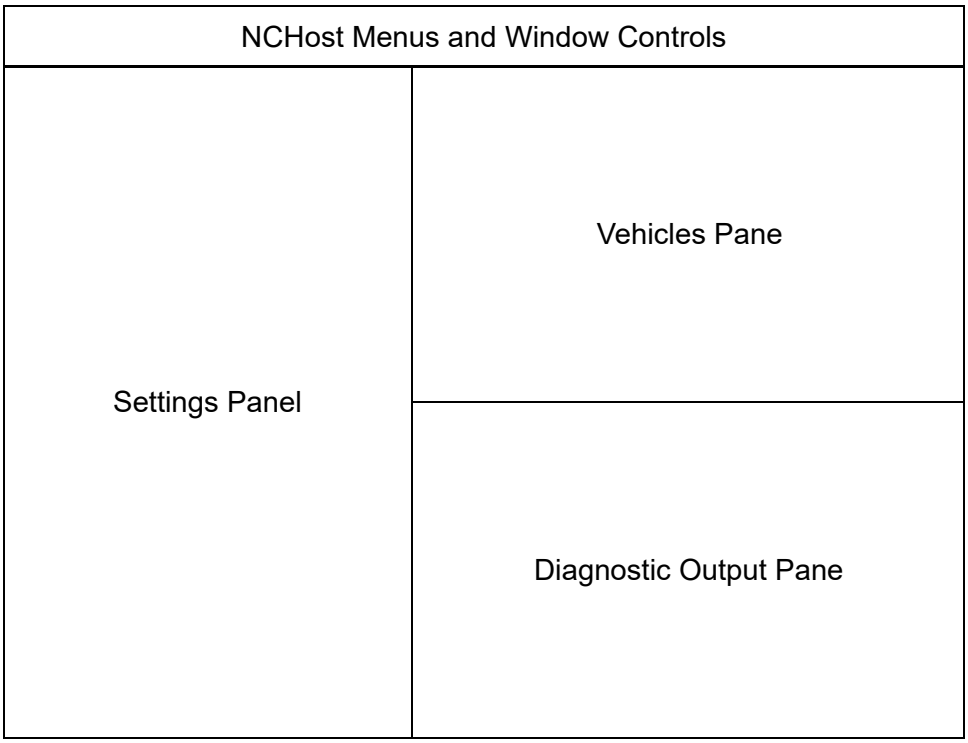


Figure 4-1: Window Layout

Window Behavior

- **Window Size** – The window can be sized as required, it opens at its default size each time the NCHost TCP Interface Utility is run.
- **Panel Size** – The width of the **Settings** panel is fixed, but it can be collapsed or expanded by selecting the Settings icon in the upper-left corner of the window. It is expanded each time the NCHost TCP Interface Utility is run.
- **Pane Size** – The bar between the **Vehicles** pane and the **Diagnostic Output** pane can be moved vertically as required. It is placed at its default location each time the NCHost TCP Interface Utility is run.

User Interface Features

Dialog Boxes

Dialog boxes, which are shown in Figure 4-2, are used to select options, input information, and start operations. Dialog boxes are displayed when certain functions (for example, buttons that allow user input) are selected in the current window or dialog box. The dialog box that is displayed depends on the type of input required.

To close a dialog box and save the selection, select the action button for the dialog box (**Open**, **OK**, and so on). To close a dialog box without saving, select **Cancel**.

NOTE: When certain dialog boxes are open, no actions are permitted until the dialog box is closed.

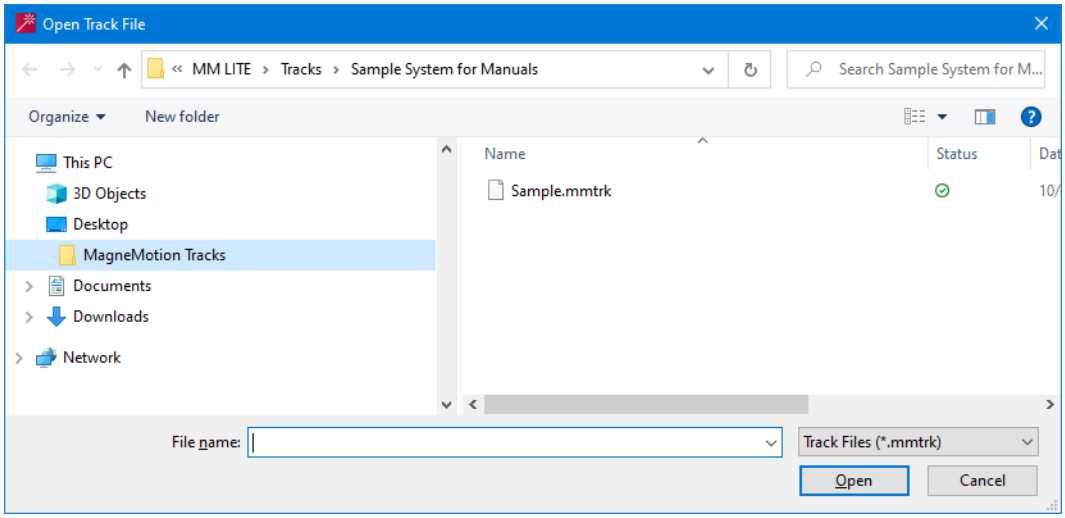


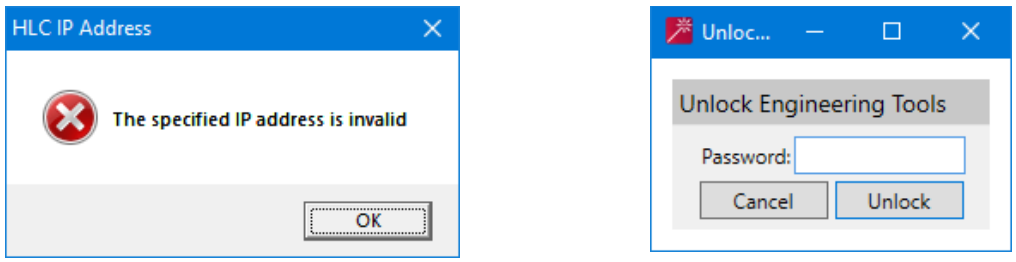
Figure 4-2: Dialog Box Example

Messages

Messages, which are shown in Figure 4-3, are used to display information, to confirm user input, or to warn of user input that is disallowed. The NCHost TCP Interface Utility automatically displays these messages when a restricted operation is attempted, or when some important operational information must be conveyed.

- Single-button Messages provide a descriptive message that is related to the previously selected action. Close the Message after reading by selecting the **OK** button.
- Multi-button Messages provide a descriptive message that is related to the previously selected action and options for response to the message. Close the Message after reading by selecting one of the options presented.

NOTE: When certain Messages are open, no actions are permitted until the Message is closed.



Single Button Message

Multi-Button Message

Figure 4-3: Message Examples

Dialog Box and Window Elements

Checkboxes, which are shown in Figure 4-4, are used to enable or disable an option by selecting (checking) or clearing (unchecking) the box. In those areas where multiple checkboxes are presented, more than one checkbox can be selected.

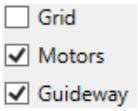


Figure 4-4: Checkbox Example

Option buttons, which are shown in Figure 4-5, are used for selecting one option from a group of mutually exclusive options.

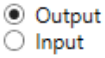


Figure 4-5: Option Button Example

Drop-down menus or list boxes, which are shown in Figure 4-6, are used to select one option from a list of available options. A drop-down menu or list box is used when no confirmation of the selection is required. Once an option is selected, it is immediately activated.

Drop-down Menu

Drop-down List Box

Figure 4-6: Drop-Down Menu Examples

Text fields, which are shown in Figure 4-7, are used to enter variable information, such as names, setpoints, and position values. Display fields, which are shown in Figure 4-7, are used to display fixed and variable information, such as property names and values returned from the controlled components.

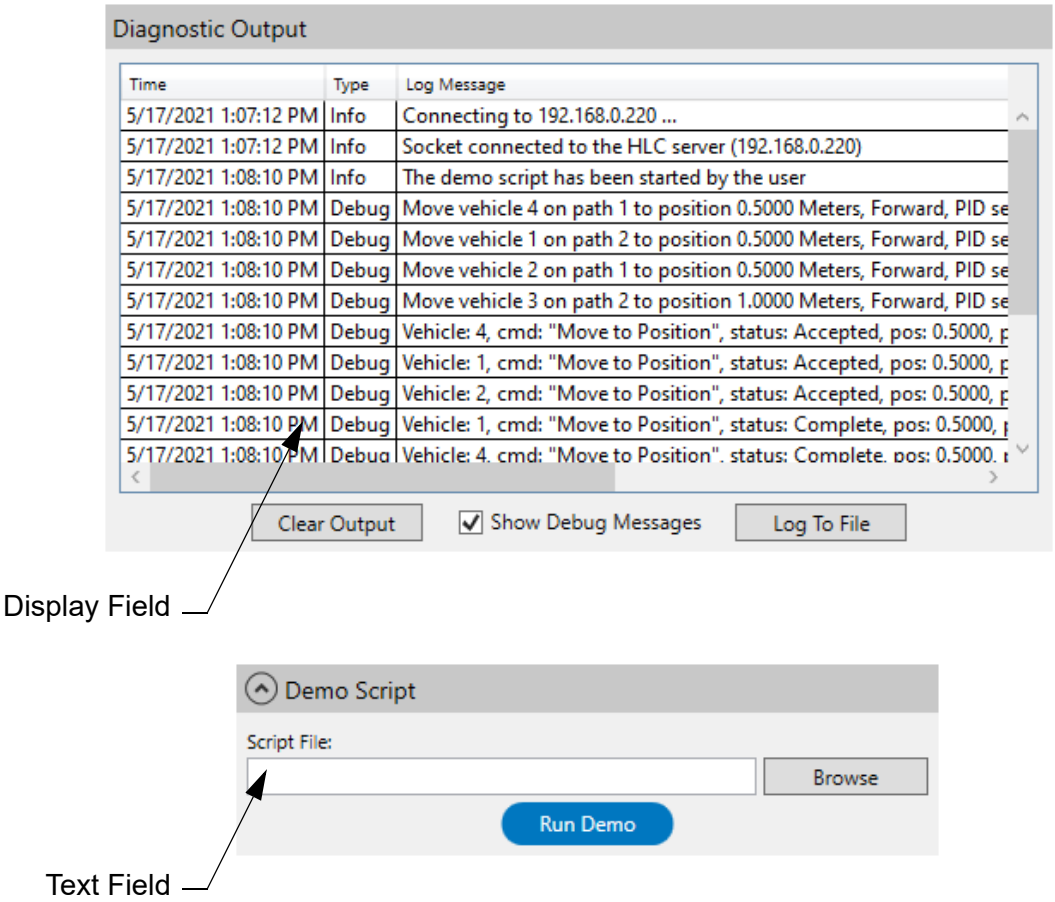


Figure 4-7: Text and Display Field Examples

Each window, dialog box, and message contain graphical buttons (shown in Figure 4-8), which are used to perform various actions, including:

- Initiating an action.
- Changing the display.
- Opening dialogs, menus, or windows.

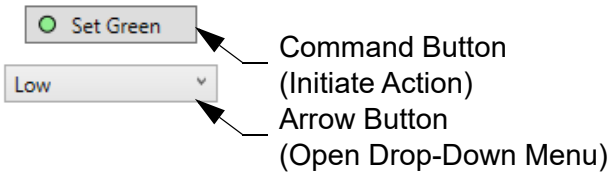


Figure 4-8: Button Examples

Window and Dialog Box Reference

This reference provides a detailed description of each window, page, and dialog box displayed by the NCHost TCP Interface Utility. All windows, menus, and dialog boxes within this reference are listed in functional order.

Main Window

The window for the NCHost TCP Interface Utility is shown in Figure 4-9. All window functions are described after the figure. This window provides access to all functions of the utility. The function of each item in the window and the properties for those items are described in this chapter.

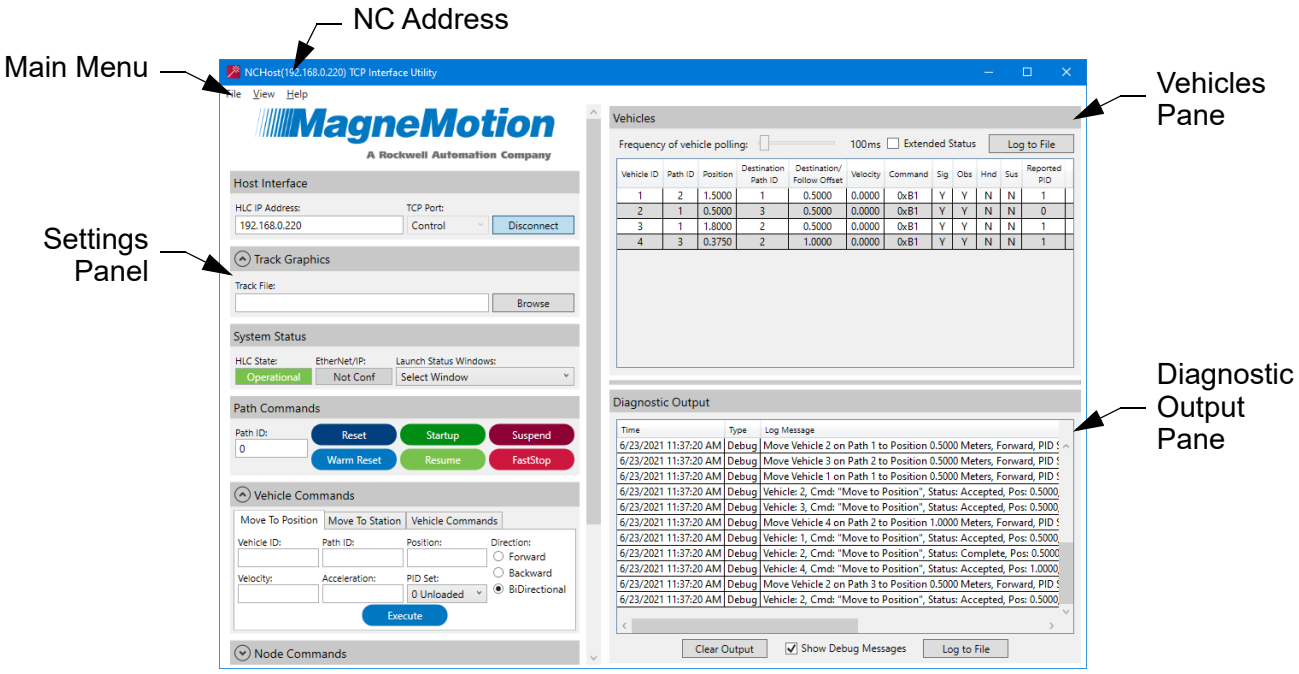


Figure 4-9: NCHost TCP Interface Utility Main Window Overview

- **Main Menu** – Provides menus for various NCHost TCP Interface Utility options.
- **NC Address** – Displays the low-order byte of the address for the node controller when the NCHost TCP Interface Utility is connected to a node controller.
- **Settings Panel** – Provides tools for monitoring and controlling the operations of a MagneMotion transport system.
- **Vehicles Pane** – Displays the status of all vehicles on the transport system.
- **Diagnostic Output Pane** – Displays a log of the actions that are performed through the NCHost TCP Interface Utility and the responses from the transport system.

Main Menu

The Main Menu (see Figure 4-10), at the top of the NCHost TCP Interface Utility window as shown in Figure 4-9, provides menus for access to various functions.

Figure 4-10: Main Menu

- **File** – Includes options that are related to the operation of the NCHost TCP Interface Utility.
  - **Exit** – Select to shut down the utility.
- **View** – Includes options that are related to the configuration of the user interface for the NCHost TCP Interface Utility.
  - **Select System Type** – Configures the utility for operation with a specific MagneMotion motor family by enabling only those functions that are supported for those motors.
    - **MagneMover LITE** – Select to configure the utility for MagneMover LITE motors.
    - **QuickStick 100** – Select to configure the utility for QuickStick 100 and QuickStick Plus motors.
    - **QuickStick HT** – Select to configure the utility for QSHT motors.
  - **Track Graphics** – Select to display the Track Graphics section of the Settings Panel.
  - **Demo Script** – Select to display the Demo Script section of the Settings Panel.
  - **Engineering Tools** – Select to display the Engineering Tools section of the Settings Panel.
- **Help** – Includes options that are related to using the NCHost TCP Interface Utility.
  - **About NCHost TCP Interface Utility** – Select to view version information as shown in Figure 4-11.

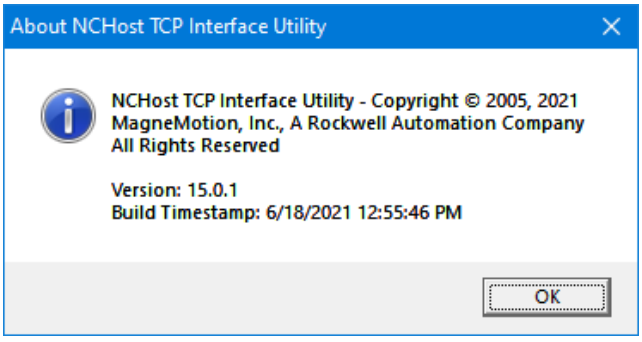


Figure 4-11: About NCHost TCP Interface Utility Message

Settings Panel

The **Settings** panel (see Figure 4-12), on the left side of the NCHost TCP Interface Utility window as shown in Figure 4-9, provides access to all monitoring and control tools.

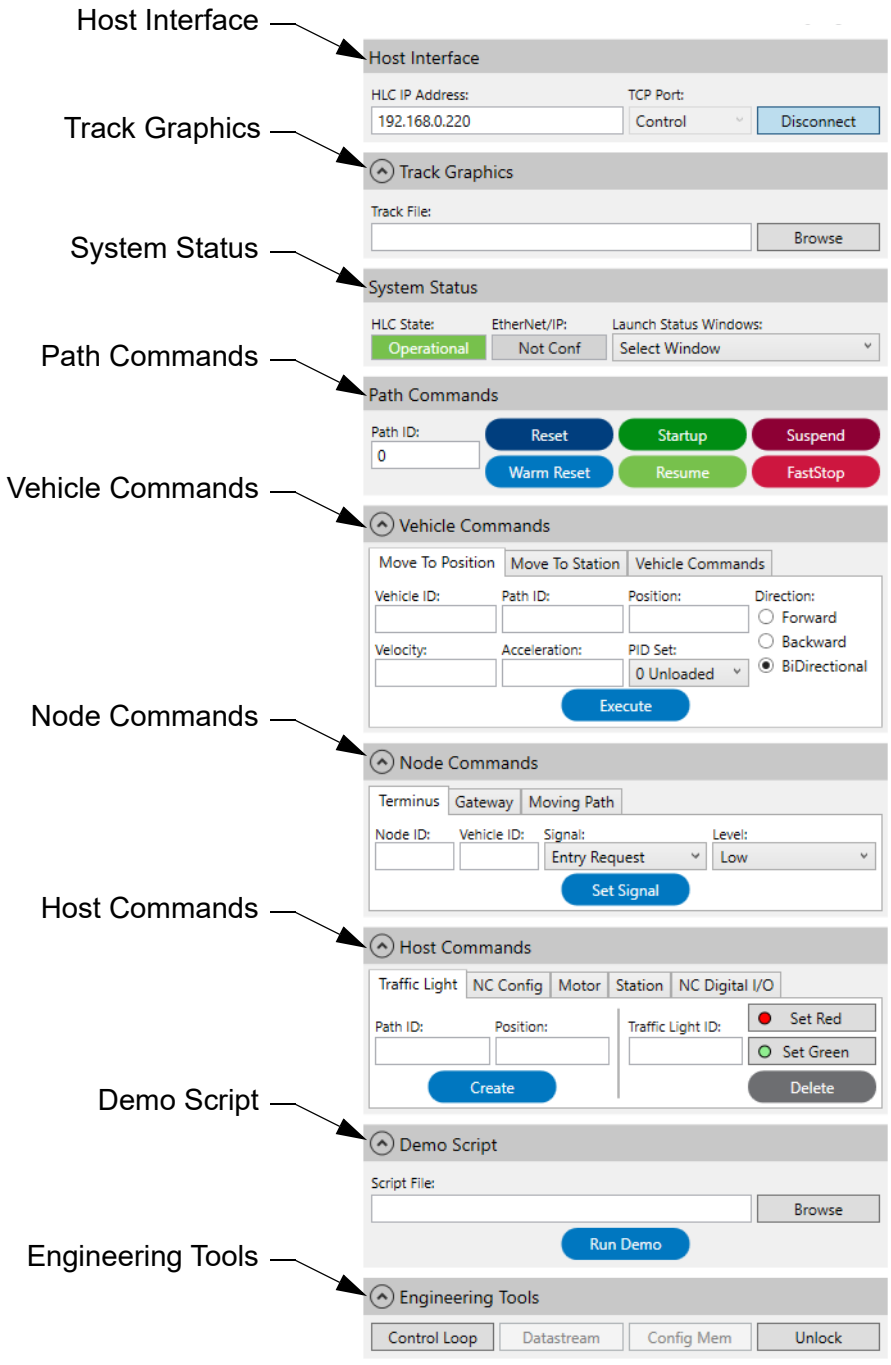


Figure 4-12: Settings Panel

- **Host Interface** – Provides a tool for connecting to the node controller designated as the high-level controller (HLC). See *Host Interface* on page 146.

- **Track Graphics** – Provides a tool for selecting and displaying a graphic of the transport system. Only displayed when Track Graphics is selected on the View menu. See *Track Graphics* on page 147.
- **System Status** – Provides a tool for displaying the status of the connection to the high-level controller and for displaying various transport system status windows. See *System Status* on page 150.
- **Path Commands** – Provides tools for sending basic path-related commands to the HLC. See *Path Commands* on page 167.
- **Vehicle Commands** – Provides a tool for commanding motion and other functions for specific vehicles on the transport system. See *Vehicle Commands* on page 168.
- **Node Commands** – Provides tools for moving specific vehicles on or off the transport system through Terminus or Gateway Nodes or controlling the linking of Moving Path Nodes. See *Node Commands* on page 176.
- **Host Commands** – Provides tools for sending basic host-lever transport system commands to the HLC. See *Host Commands* on page 180.
- **Demo Script** – Provides a tool for loading and running a Demo Script for transport system testing. Only displayed when Demo Script is selected on the View menu. See *Demo Scripts* on page 186.
- **Engineering Tools** – Provides a tool for fine-tuning the PID control loop. Only displayed when Engineering Tools is selected on the View menu. See *Engineering Tools* on page 187.

Host Interface

The **Host Interface** section (see Figure 4-13) is used to connect to the node controller designated as the high-level controller. See *Connecting to the Transport System* on page 42 for usage.

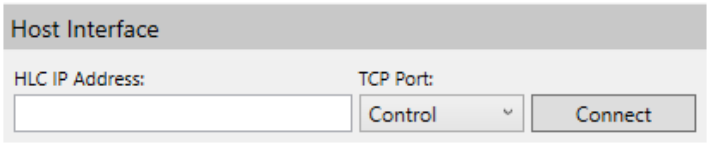


Figure 4-13: Settings Panel, Host Interface Section

- **HLC IP Address** – Provides a text field for the entry of the IP address of the high-level controller.
- **TCP Port** – Provides a menu to select the operating mode for the NCHost TCP Interface Utility.
  - **Control** – The utility operates in Host Control mode and provides both status and control functions. Only one Host Control session can be connected to a high-level controller at a time.

    NOTE: Connecting a second Host Control session causes the first Host Control session to be disconnected.
  - **Status** – The utility operates in Host Status mode and provides status information only, control is not allowed. Up to four Host Status sessions and a Host Control session can be connected to a high-level controller simultaneously.

    NOTE: Connecting a fifth Host Status session causes the first Host Status session to be disconnected.
- **Connect/Disconnect** – Toggles between Connect and Disconnect based on the current connection status.
  - **Connect** – Displayed when the NCHost TCP Interface Utility is not connected, select to connect to the high-level controller specified in **HLC IP Address**.
  - **Disconnect** – Displayed when the NCHost TCP Interface Utility is connected, select to disconnect from the HLC.

Track Graphics

The **Track Graphics** section of the settings panel (see Figure 4-14) is used to select the transport system track layout graphic and display it in the track graphics window. See *Track Graphics* on page 57 for usage.

NOTE: This section is only displayed when Track Graphics is selected on the **View** menu.

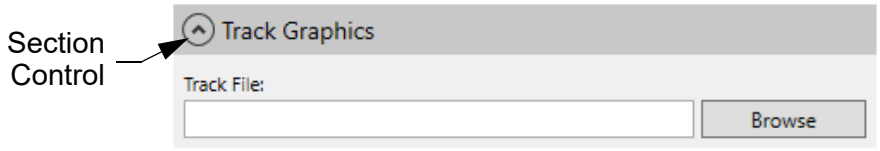


Figure 4-14: Settings Panel, Track Graphics Section

- **Section Control** – Select to collapse or expand this section.
- **Track File** – Displays the location and name of the current Track file. The Track File is created for MagneMover® LITE systems using the MagneMotion Configurator. See *Rockwell Automation Support* on page 264 for questions regarding the development of Track files for QuickStick® transport systems.
- **Browse** – Select to open a dialog box for the selection of a track file (*track_file*.mmtrk). Once selected, the **Track Graphics** window opens (see Figure 4-15) and displays the loaded track configuration.

The Track Graphics window displays a graphic of the transport system layout as defined in the currently loaded Track File. The graphic displays the current state of all vehicles and vehicle motion on the transport system. The default settings for the Track Graphics window are used when the window is opened after starting the NCHost TCP Interface Utility. See *Vehicle Motion* on page 57 for usage.

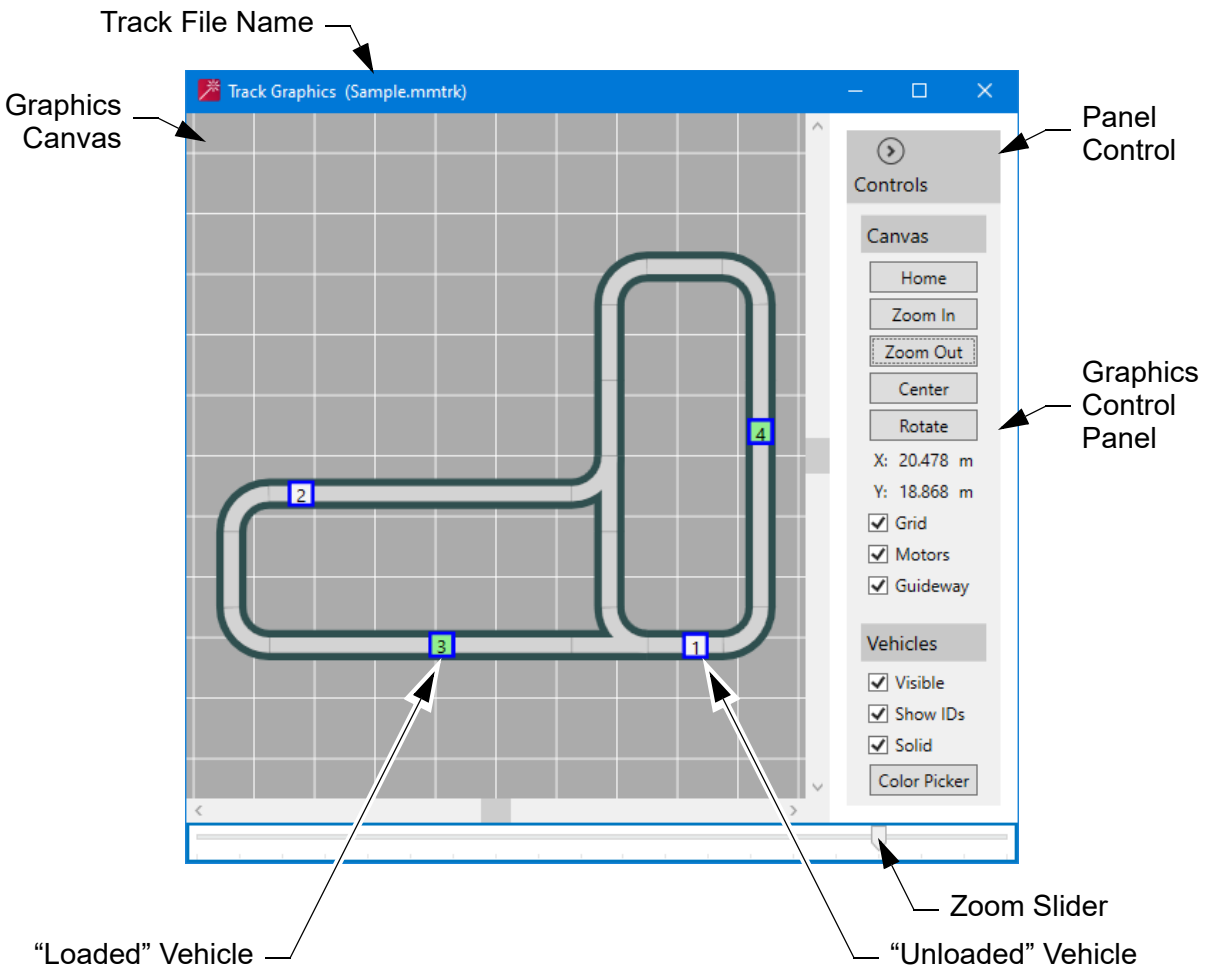


Figure 4-15: Track Graphics Window

- **Track File Name** – Displays the name of the current Track file.
- **Graphics Canvas** – Displays the transport system layout as defined in the currently loaded Track file and all vehicles and vehicle motion on the transport system.

  Zoom in and out on the track graphics display using the mouse wheel. Moved the graphic by clicking the canvas and dragging the graphic to a new location. Or, the graphic controls to the right of the canvas can be used.
- **Graphics Control Panel** – Displays the **Canvas** and **Vehicles** controls, which are used to adjust the display of the track and vehicles.
  - **Panel Control** – Select to collapse or expand the **Graphics Control Panel** panel.
  - **Canvas**
    - **Home** – Select to reset the display to its original zoom and X/Y settings.
    - **Zoom In** – Select to make the track graphic larger in 10% increments.

- **Zoom Out** – Select to make the track graphic smaller in 10% increments.
- **Center** – Select to center the track graphic in the canvas.
- **Rotate** – Select to rotate the track graphic 90° clockwise.
- **X/Y** – Shows the X and Y position of the cursor.
- **Grid** – Select to display the grid. The default is selected (on).
- **Motors** – Select to display the motors. The default is selected (on).
- **Guideway** – Select to display the guideway. The default is selected (on).

- **Vehicles**
  - **Visible** – Select to show all vehicles on the track graphic. The default is selected (on). Vehicles are displayed using the default color for the PID set they are using.
  - **Show IDs** – Select to show the Vehicle IDs on the vehicles when **Visible** is selected. The default is selected (on).
  - **Solid** – Clear to hide the vehicles on the track graphic until the PID set they are using changes. Once vehicles are displayed, they use the default color for their current PID set.
  - **Color Picker** – Select to open the **Color** dialog box for selection of the fill color for the vehicles currently using PID Set 0 (unloaded). Once the PID set for a vehicle changes from Set 0, the vehicle is displayed in the default color for the new PID set. Once the PID set for a vehicle returns to Set 0, the vehicle is displayed in the default color for PID Set 0 (white).

- **Zoom Slider** – Click and drag the slider to make the track graphic smaller or larger (10…200%). Click the slider to display the zoom level. The default is 20%.

System Status

The **System Status** section (see Figure 4-16) is used to provide the status of the connection to the HLC and for displaying various transport system status windows. See *Connecting to the Transport System* on page 42 for usage.

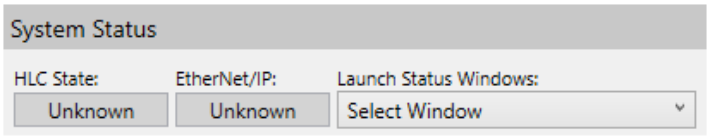


Figure 4-16: Settings Panel, System Status Section

- **HLC State** – Displays the state of the high-level controller that is specified in **HLC IP Address**.
  - **Degraded** – The HLC is unable to communicate with one or more node controllers in the transport system (background is orange).
  - **Initialization** – The HLC is loading the Node Controller Configuration File or an error is keeping the HLC from exiting this state (background is yellow).
  - **Operational** – The HLC is running properly (background is green).
  - **Unknown** – The HLC is not connected, or an IP address has not been specified (background is gray).
- **EtherNet/IP** – Displays the state of the EtherNet/IP™ connection from the host controller to the high-level controller. This connection is only used when EtherNet/IP communication has been defined in the Configuration File for communication with a host controller.
  - **Down** – The EtherNet/IP connection is down, the HLC is unable to contact the host controller at the configured IP address, or connectivity has been lost (background is red).
  - **Not Conf** (Not Configured) – The specified IP address does not use EtherNet/IP (background is gray) or the Node Controller Configuration File is not specifying EtherNet/IP communication.
  - **Unknown** – The HLC is not connected, or an IP address has not been specified (background is gray).
  - **Up** – The EtherNet/IP connection is running properly (background is green).
- **Launch Status Windows** – Provides a menu to select and open individual windows that provide information on the paths, nodes, and other components in the transport system. See *Monitoring the Transport System – Status Information* on page 46 for usage.
  - **Track Graphics** – Opens the Track Graphics window (see Figure 4-15, *Track Graphics Window*) to display a graphic of the transport system layout as defined in the currently loaded Track File.

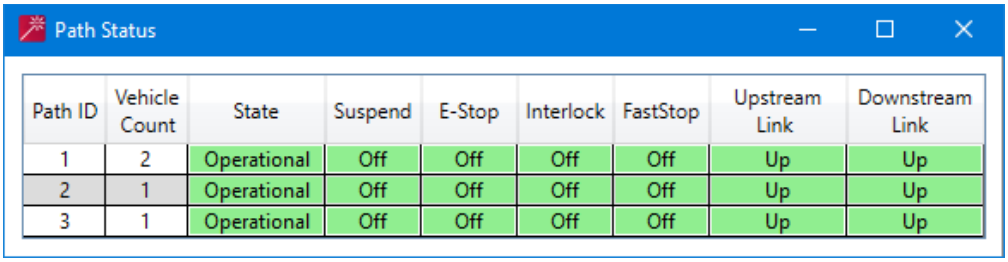
- **Paths** – Opens a window (see *Path Status Window*) that displays the status of all paths in the transport system.
- **Nodes** – Opens a window (see *Node Status Window*) that displays the status of all nodes in the transport system.
- **Node Controllers** – Opens a window (see *Node Controller Status Window*) that displays the status of all node controllers in the transport system.
- **HLC** – Opens a window (see *HLC Status Window*) that displays the status of the high-level controller.
- **Motors** – Opens a window (see *Motor Status Window*) that displays the status of all Motors in the transport system.
- **Stations** – Opens a window (see *Station Status Window*) that displays the status of all stations in the transport system.
- **Traffic Lights** – Opens a window (see *Traffic Lights Status Window*) that displays the status of all traffic lights in the transport system.
- **Moving Paths** – Opens a window (see *Moving Paths Status Window*) that displays the status of all Moving Path Nodes in the transport system and the paths that are connected to them.

  NOTE: This feature is only available on QuickStick and QSHT configurations.
- **Power Supplies** – Opens a window (see *Propulsion Power Supply Status Window*) that displays the status of all Kinetix 2198-Pxxx DC-bus propulsion power supplies in the transport system.

  NOTE: This feature is only available on QSHT configurations.

Path Status Window

The **Path Status** window (see Figure 4-17) is used to display the status of all paths in the transport system. See *Path Status* on page 47 for usage.



Path ID	Vehicle Count	State	Suspend	E-Stop	Interlock	FastStop	Upstream Link	Downstream Link
1	2	Operational	Off	Off	Off	Off	Up	Up
2	1	Operational	Off	Off	Off	Off	Up	Up
3	1	Operational	Off	Off	Off	Off	Up	Up

Figure 4-17: Path Status

Select any of the column headers to sort the contents of the Path Status window by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Path ID** – Displays the ID number of the path.
- **Vehicle Count** – Displays the number of vehicles currently on the path.
- **State** – Displays the state of the path.
  - **Init** – All motors on the path are initialized (background is yellow).
  - **Operational** – All motors on the path are running (background is green).
  - **Programming** – All motors on the path are being programmed (background is pink).
  - **Reset** – All motors on the path are being reset (background is blue).
  - **Startup** – All motors on the path are being started (background is orange).
- **Suspend** – Displays the state of the Suspend command for the path (indicates if vehicle motion has been suspended).
  - **Off** – All motion is permitted (background is green).
  - **On** – All motion has been Suspended (background is red). Any vehicle in motion on the path decelerates and stops based on its current velocity, payload, and commanded acceleration. Once stopped, vehicles are held in place where they stopped. Motion is not permitted while **Suspend** is on.
- **E-Stop** – Displays the state of the user-supplied E-stop circuit for the path.

  NOTE: The **E-stop** function is not supported in simulation mode.
  - **Off** – The E-stop is inactive, motion is permitted (background is green).
  - **On** – The E-stop is active, motion is not permitted (background is red). Any vehicle in motion on the path decelerates and stops based on its current velocity, payload, and commanded acceleration. Once stopped, vehicles are held in place where they stopped. Motion is not permitted while **E-stop** is on.

- **Interlock** – Displays the state of the user-supplied interlock circuit for the path.

  NOTE: The **Interlock** function is not supported in simulation mode.

  - **Off** – The Interlock is inactive, motion is permitted (background is green).
  - **On** – The Interlock is active, all motion has been Suspended (background is red). Any vehicle in motion on the path decelerates and stops based on its current velocity, payload, and commanded acceleration. Once stopped, vehicles are held in place where they stopped. Motion is not permitted while **Interlock** is on.

- **FastStop –** Displays the state of the FastStop command on the path (indicates if vehicle motion has been FastStopped).

  NOTE: This function is only available on QuickStick and QS HT configurations.

  - **Off** – All motion is permitted (background is green).
  - **On** – All motion has been FastStopped (background is red). Any vehicle in motion on the path immediately decelerates with maximum thrust opposite to the current direction of motion and stops. Once stopped, vehicles are held in place where they stopped. Motion on the path is not permitted while **FastStop** is on.

- **Upstream Link** – Displays the state of the RS-422 connection to the upstream end of the first motor in the path.

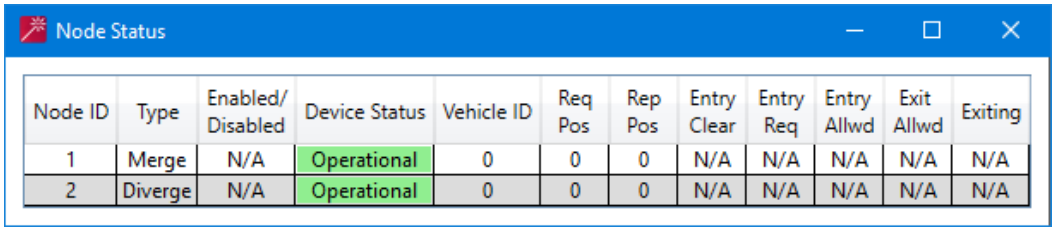
  - **Up** – The connection is working (background is green).
  - **Down** – The connection has a fault (background is red).
  - **Not Conf** (Not Configured) – There is no connection (background is light gray).

- **Downstream Link** – Displays the state of the RS-422 connection to the downstream end of the last motor in the path.

  - **Up** – The connection is working (background is green).
  - **Down** – The connection has a fault (background is red).
  - **Not Conf** (Not Configured) – There is no connection (background is light gray).

Node Status Window

The **Node Status** window (see Figure 4-18) is used to display the status of all nodes in the transport system. See *Node Status* on page 48 for usage.



Node ID	Type	Enabled/ Disabled	Device Status	Vehicle ID	Req Pos	Rep Pos	Entry Clear	Entry Req	Entry Allwd	Exit Allwd	Exiting
1	Merge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A
2	Diverge	N/A	Operational	0	0	0	N/A	N/A	N/A	N/A	N/A

Figure 4-18: Node Status

Select any of the column headers to sort the contents of the Node Status window by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Node ID** – Displays the ID number of the node.
- **Type** – Displays the type of the node (defined in the Node Controller Configuration File).
- **Enabled/Disabled** – Displays the state of a Gateway Node.
  - **N/A** – Does not apply to the node.
  - **Enabled** – The Gateway Node can be used for vehicle transfer between HLC Control Groups (background is green).
  - **Disabled** – The Gateway Node cannot be used for vehicle transfer between HLC Control Groups (background is red).
- **Device Status** – Displays the state of the device that is associated with the node.
  - **Init** – The node is initializing (background is yellow).
  - **Faulted** – The node has an error (background is red).
  - **N/A** – Does not apply to the node.
  - **Operational** – The node is running properly (background is green).
  - **Junction Fault** – There is a vehicle located in an unlinked Moving Path Node junction (background is orange).
- **Vehicle ID** – Displays the ID number of the vehicle currently in the node; 0 if there is no vehicle in the node.
- **Req Pos** (Requested Position) – The last requested position for the device that is associated with the node (only applies to nodes with devices, for example, switch, turntable).
- **Rep Pos** (Reported Position) – The last reported position for the device that is associated with the node (only applies to nodes with devices, for example, switch, turntable).

- **Entry Clear** – Signal from the node controller that indicates there is sufficient room to insert a vehicle at a Terminus Node. See the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003, or the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004, for the handshake sequence.
  - **N/A** – Does not apply to the node.
  - **Off** – The node is occupied.
  - **On** – The node is clear (background is green).
- **Entry Req** (Entry Request) – Signal from host controller that indicates that a vehicle is ready for entry into a Terminus Node. See the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003, or the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004, for the handshake sequence.
  - **N/A** – Does not apply to the node.
  - **Off** – No request.
  - **On** – Request to insert vehicle into the node (background is green).
- **Entry Allwd** (Entry Allowed) – Signal from node controller that indicates that motion permission is acquired for the vehicle, and the vehicle is allowed to enter the Terminus Node. See the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003, or the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004, for the handshake sequence.
  - **N/A** – Does not apply to the node.
  - **Off** – Vehicle cannot be inserted.
  - **On** – Vehicle can be inserted into the node and commands can be issued for the vehicle (background is green).
- **Exit Allwd** (Exit Allowed) – Signal from host controller that indicates that it is ready to receive a vehicle from a Terminus Node. See the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003, or the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004, for the handshake sequence. handshake sequence.
  - **N/A** – Does not apply to the node.
  - **Off** – Vehicle cannot exit.
  - **On** – Vehicle can exit (background is green).
- **Exiting** – Signal from node controller that indicates a vehicle is exiting a Terminus Node. See the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003, or the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004, for the handshake sequence.
  - **N/A** – Does not apply to the node.
  - **Off** – Vehicle not exiting from node.
  - **On** – Vehicle exiting from node (background is green).

Node Controller Status Window

The **Node Controller Status** window (see Figure 4-19 and Figure 4-20) is used to display the status of all node controllers in the transport system. See *Node Controller Status* on page 49 for usage.

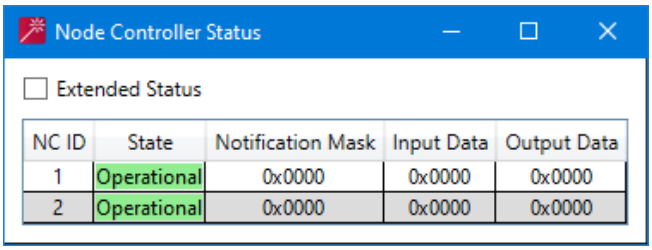


Figure 4-19: Node Controller Status

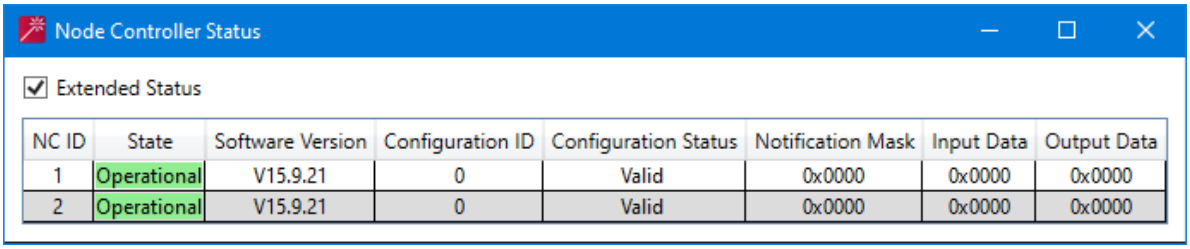


Figure 4-20: Node Controller Extended Status

Select any of the column headers to sort the contents of the Node Controller Status window by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Extended Status** – Select to display the extended status information.
- **NC ID** – Displays the ID number of the node controller.
- **State** – Displays the state of the node controller.
  - **Disconnected** – The TCP connection from the high-level controller to the node controller is down (background is red).
  - **Initializing** – The high-level controller is loading the Node Controller Configuration File or an error is keeping the node controller from exiting this state (background is yellow).
  - **Operational** – The node controller is running (background is green).
- **Software Version** – The version of the node controller software, provided as major.minor.patch (only displayed when Extended Status is selected).
- **Configuration ID** – The ID of the active Node Controller Configuration File for the node controller (only displayed when Extended Status is selected).
  - **0** – Static Node Controller Configuration File.
  - **1…20** – Managed Node Controller Configuration File.

- **Configuration Status** – Displays the configuration status of the node controller (only displayed when Extended Status is selected).
  - **Valid** – The active node controller configuration is valid.
  - **Invalid** – The active node controller configuration is invalid.
- **Notification Mask –** Displays the hexadecimal value that was specified during the last node controller Digital I/O input operation. Not used when in simulation mode.
- **Input Data** – Displays the status (in hexadecimal) of all Digital I/O input bits on the specified node controller. The status of the Digital I/O bits automatically updates when the state of one or more Digital I/O input bits changes.
- **Output Data** – Displays a hexadecimal value corresponding to the last node controller Digital I/O output bit operation performed.

HLC Status Window

The **HLC Status** window (see Figure 4-21 and Figure 4-22) is used to display the status of the high-level controller for the transport system. See *HLC Status* on page 50 for usage.

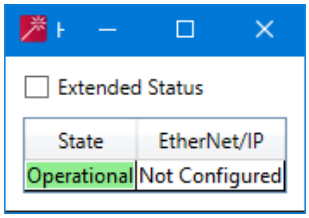


Figure 4-21: HLC Status

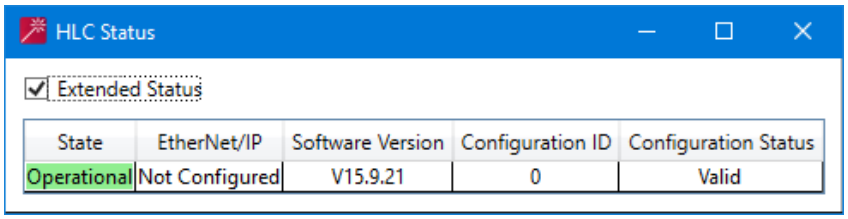


Figure 4-22: HLC Extended Status

Select any of the dividers between columns to change the width of the column to the left.

- **Extended Status** – Select to display the extended status information.
- **State** – Displays the state of the HLC.
  - **Disconnected** – The TCP connection from the host controller to the high-level controller is down (background is red).
  - **Initializing** – The high-level controller is loading the Node Controller Configuration File or an error is keeping the HLC from exiting this state (background is yellow).
  - **Operational** – The high-level controller is running (background is green).
- **EtherNet/IP** – Displays the state of the EtherNet/IP connection from the host controller to the HLC. This connection is only used when EtherNet/IP communication has been defined in the Node Controller Configuration File for communication with a host controller.
  - **Down** – The EtherNet/IP connection is down, the HLC is unable to contact the host controller at the configured IP address, or connectivity has been lost (background is red).
  - **Not Conf** (Not Configured) – The specified IP address does not use EtherNet/IP (background is gray).
  - **Unknown** – The HLC is not connected, or an IP address has not been specified (background is gray).
  - **Up** – The EtherNet/IP connection is running properly (background is green).

- **Software Version** – The version of the HLC software, provided as major.minor.patch (only displayed when Extended Status is selected).
- **Configuration ID** – The ID of the active Node Controller Configuration File for the HLC (only displayed when Extended Status is selected).
  - **0** – Static Node Controller Configuration File.
  - **1…20** – Managed Node Controller Configuration File.
- **Configuration Status** – Displays the configuration status of the HLC (only displayed when Extended Status is selected).
  - **Valid** – The active node controller configuration is valid.
  - **Invalid** – The active node controller configuration is invalid.

Motor Status Window

The **Motor Status** window (see Figure 4-23) is used to display the status of all Motors in the transport system. See *Motor Status* on page 51 for usage.

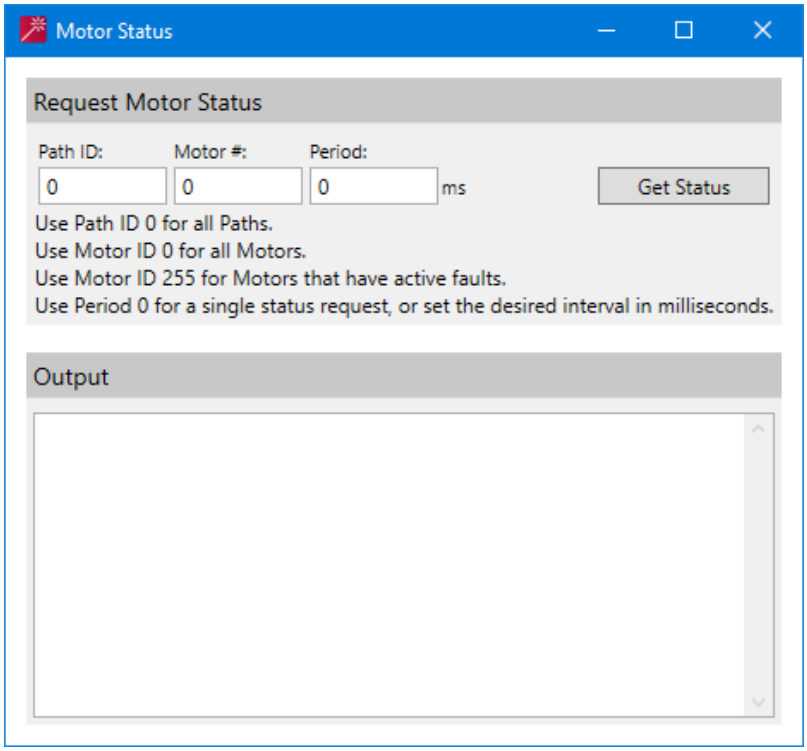


Figure 4-23: Motor Status

- **Path ID** – The ID number of the path for the status request. Set to 0 for all paths.
- **Motor #** – The ID number of the motor for the status request on the specified paths.
  - **0** – Status of all motors.
  - **Motor ID** – Status of only the specified motor.
  - **255** – Status of only motors with active faults.
- **Period** – The amount of time between motor status updates, in milliseconds. Set to 0 for one status update.
- **Get Status** – Select to get the motor status as defined through the **Path ID**, **Motor**, and **Period** settings and display it in the **Output** field.
- **Output** – Displays a log of motor status as defined through the **Path ID**, **Motor**, and **Period** settings. If the **Period** is not set to 0, each update replaces the status currently being displayed. See the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003, or the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004, for descriptions of the motor status. The text that is displayed in the log can be copied and pasted into another application (for example, a text editor).

Station Status Window

The **Station Status** window (see Figure 4-24) is used to display the status of all stations in the transport system. See *Station Status* on page 53 for usage.

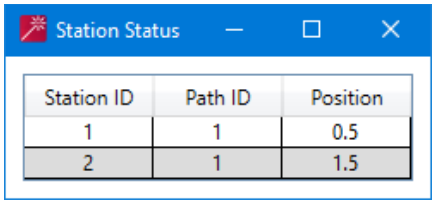


Figure 4-24: Station Status

Select any of the column headers to sort the contents of the Station Status window by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Station ID** – Displays the ID of the station.
- **Path ID** – Displays the ID of the path where the station is located.
- **Position** – Displays the position from the start of the path, in meters, where the station is located.

Traffic Lights Status Window

The **Traffic Light Status** window (see Figure 4-25) is used to display the status of all traffic lights in the transport system. See *Traffic Light Status* on page 54 for usage.

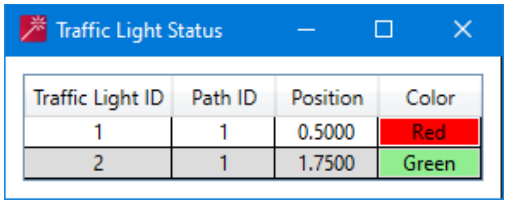


Figure 4-25: Traffic Light Status

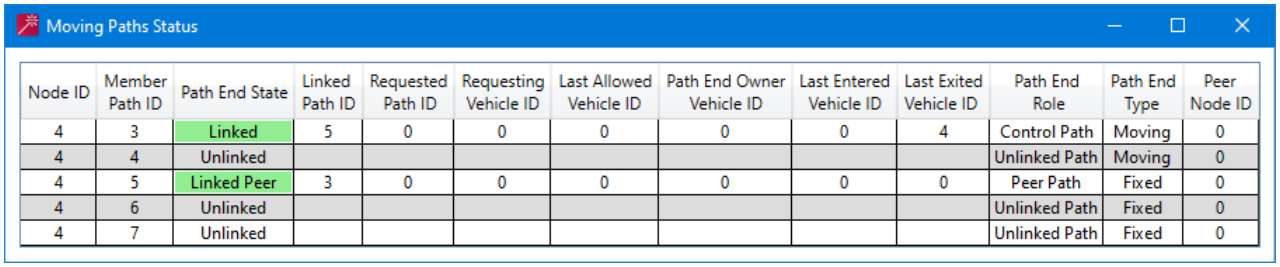
Select any of the column headers to sort the contents of the Traffic Light Status window by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Traffic Light ID** – Displays the ID of the traffic light.
- **Path ID** – Displays the ID of the path where the traffic light is located.
- **Position** – Displays the position from the start of the path, in meters, where the traffic light is located.
- **Color** – Displays the color of the traffic light.
  - **Red** – Vehicles are not granted permission to move into or beyond the motor block at the position of the traffic light.
  - **Green** – Vehicles are granted permission to move beyond the position of the traffic light.

Moving Paths Status Window

The **Moving Paths Status** window (see Figure 4-26) is used to display the status of all Moving Path Nodes in the transport system. See *Moving Path Status* on page 55 for usage.

NOTE: This feature is only available on QuickStick and QSHT configurations.



Node ID	Member Path ID	Path End State	Linked Path ID	Requested Path ID	Requesting Vehicle ID	Last Allowed Vehicle ID	Path End Owner Vehicle ID	Last Entered Vehicle ID	Last Exited Vehicle ID	Path End Role	Path End Type	Peer Node ID
4	3	Linked	5	0	0	0	0	0	4	Control Path	Moving	0
4	4	Unlinked								Unlinked Path	Moving	0
4	5	Linked Peer	3	0	0	0	0	0	0	Peer Path	Fixed	0
4	6	Unlinked								Unlinked Path	Fixed	0
4	7	Unlinked								Unlinked Path	Fixed	0

Figure 4-26: Moving Paths Status

Select any of the column headers to sort the contents of the Moving Paths Status window by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Node ID** – Displays the ID number of the Moving Path Node.
- **Member Path ID** – Displays the ID number of the path that is connected to the node.
- **Path End State** – Displays the state of the path end.
  - **Unlinked** – The path end is not linked.
  - **Linked Unlink Pending** – The junction is linked, which allows vehicles already navigating the junction to proceed (background is yellow). The junction is unlinked once the vehicle specified by **Last Allowed** transitions through the node.
  - **Unlinked Alignment Requested** – The path end is not linked (background is orange).
  - **Linked** – The Control Path is linked to a Peer Path to form a path junction (background is green).
  - **Linked Comm Loss** – Communication between the host controller and the HLC or between the HLC and the NC that owns the node is lost while a junction is linked (background is yellow).
  - **Linked Peer** – The Peer Path is linked to the Control Path specified by **Linked Path** to form a path junction (background is green).
  - **Unlinked Vehicle Present** – A vehicle was located during startup that extends past the entry gate on a path end in the Moving Path Node (background is yellow).
- **Linked Path ID** – Displays the ID number of the path that is linked with the Member Path to create a junction. Blank if the Member Path is unlinked.

- **Requested Path ID** – Displays the signal to the host controller when a path end transitions to the **Unlinked Alignment Requested** state that a vehicle is requesting permission to navigate the Moving Path Node as follows:
  - **0** – The path end is not linked and the host controller may align any equivalent moving path to provide a route for the vehicle to its destination.
  - **1…65535** – The requested path must be aligned to provide a route for the vehicle to its destination.

  The Requested Path ID persists on a control path from the time an alignment request is issued until the control path enters the **Unlinked Alignment Requested** state. The Requested Path ID field is cleared on the peer path when linked with a control path to form a junction.
- **Requesting Vehicle ID** – Displays the Vehicle ID of the vehicle that is requesting permission to navigate the specified Moving Path node.

  When in the **Unlinked Alignment Requested** state, the Requesting Vehicle ID field identifies the vehicle that is requesting permission to navigate the Moving Path node. The Requesting Vehicle ID field is cleared once the path end is linked and the vehicle is granted permission to navigate the junction.
- **Last Allowed Vehicle ID** – Displays the Vehicle ID of the last vehicle that is allowed through the node. The HLC updates the Last Allowed Vehicle ID field only on the Control Path of a linked junction.
  - **0** – The HLC keeps the Control and Peer Paths linked until the host controller sends an Unlink command. Vehicles approaching the junction are granted permission to navigate the junction if the Control Path offers an equivalent route to the vehicle's destination.
  - **1…65535** – Once the last allowed vehicle begins to navigate the path junction, the **Path End State** of the Control Path transitions to the **Linked Unlink Pending** state and no other vehicles are granted permission to navigate the junction. The HLC unlinks the path junction as soon as all navigating vehicles are clear of the path junction.
- **Path End Owner Vehicle ID** – Displays the vehicle ID of the vehicle that currently owns the path junction. Updated by the HLC only on the Control Path of a linked junction.
  - **0** – There are no vehicles navigating the path junction.
  - **1…65535** – The ID of the vehicle that currently owns the path junction.
- **Last Entered Vehicle ID** – Displays the vehicle ID of the last vehicle granted permission to enter the junction. This field is updated by the HLC only on the control path of a linked junction.
  - Displays 0 before any vehicle is granted permission and is cleared when the last entered vehicle clears the junction.

- **Last Exited Vehicle ID** – Displays the Vehicle ID of the most recent vehicle to clear the path junction. Updated by the HLC only on the Control Path of a linked junction.
  - The Last Exited Vehicle ID field is zero when no vehicles have exited the path junction since it was linked.
  - The Last Exited Vehicle ID field is zeroed when the path junction is unlinked.
- **Path End Role** – Displays the role for this path end.
  - **Unlinked Path** – The path is not currently linked to another path in the node.
  - **Control Path** – The path that is linked to the path specified by **Linked Path**.
  - **Peer Path** – The **Linked Path** that is linked to the **Control Path**.
- **Path End Type** – Displays the type of path end.
  - **Fixed** – The position of the path is fixed in the track layout (a path end that is configured as a specific-route path).
  - **Moving** – The position of the path is moved to align with the end of another path (a path end that is configured as an equivalent-route path).
- **Peer Node ID** – Displays the ID number of the node at the far end of this Moving Path Node member path.
  - **0** – There is no Moving Path Node at the far end of a moving path or if the member path is a fixed path (that is, configured as a specific-route path).
  - **1…65535** – The ID of the node at the far end of a moving path if the member path is a moving path (that is, configured as an equivalent-route path).

Propulsion Power Supply Status Window

The **Power Supply Status** window (see Figure 4-27) is used to display the status of all Kinetix 2198-Pxxx DC-bus power supplies in the transport system. See *Power Supply Status on page 56* for usage.

NOTE: This feature is only available on QSHT systems using QuickStick HT 5700 Inverters with Kinetix 2198-Pxxx DC-bus power supplies.

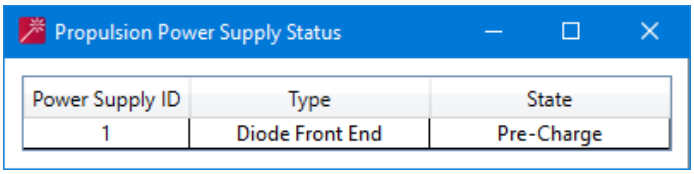


Figure 4-27: Propulsion Power Supply Status

Select any of the column headers to sort the contents of the Power Supply Status window by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Power Supply ID** – Displays the ID of the power supply.
- **Type** – Displays the type of power supply.
  - **Diode Front End** – Kinetix 2198-Pxxx DC-bus Power Supply.
- **State** – Displays the status of the power supply. See the *Integrated Motion on the EtherNet/IP Network Manual*, MOTION-RM003.

Path Commands

The **Path Commands** section (see Figure 4-28) is used to send a basic set of path management commands to the high-level controller. See *Path Commands for Transport System Operation* on page 63 for usage.

Figure 4-28: Settings Panel, Path Commands Section

- **Path ID** – The ID number of the path to be commanded.
  - **0** – The command is applied to all paths.
  - **1…65535** – The command is applied to the path specified.
- **Reset** – Sends a Reset command for the specified paths to the HLC. The Reset clears all vehicle records from the motors and releases control of all vehicles.
- **Startup** – Sends a Startup command for the specified paths to the HLC. The Startup forces a rescan of the paths to detect all vehicles and assign unique Vehicle ID numbers in the order they are detected.
- **Suspend** – Sends a Suspend Motion command for the specified paths to the HLC. The Suspend command causes all vehicles to decelerate immediately at the commanded acceleration rate to a controlled stop at the location that they were last given permission to move to. Once stopped, all vehicles are held in place. Motion is not permitted while **Suspend** is on.
- **Warm Reset** – Sends a Warm Reset command for the specified paths to the HLC. Warm Reset is the same as a full Reset, but it preserves Ethernet connectivity with the motors during the reset.

  NOTE: This feature is only available on QSHT configurations.
- **Resume** – Sends a Resume Motion command for the specified paths to the HLC.
- **FastStop** – Sends a FastStop command for the specified paths to the HLC. The FastStop command causes the motor to apply maximum thrust, opposite to the current direction of motion for all vehicles on those paths. All vehicles halt abruptly and are held in place where they stopped. Motion is not permitted while **FastStop** is on.

  NOTE: This feature is only available on QuickStick and QSHT configurations.

Vehicle Commands

The **Vehicle Commands** section (see Figure 4-29) is used to command specific vehicles to various positions on the transport system for testing. Other vehicle-related commands can also be sent to specific vehicles. See *Testing the Transport System – Vehicle Commands* on page 72 for usage.

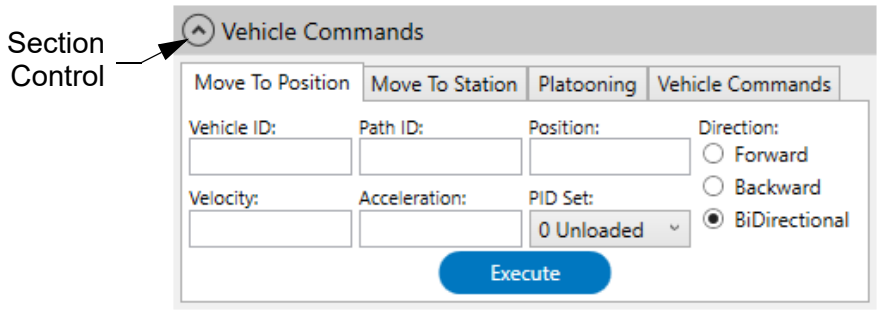


Figure 4-29: Vehicle Commands Section

- **Section Control** – Select to collapse or expand this section.
- **Move To Position** – Provides controls to move the specified vehicle to the position on the specified path using the acceleration and velocity values provided (see the *Move To Position Tab*).
- **Move To Station** – Provides controls to move the specified vehicle to the specified station using the acceleration and velocity values provided (see the *Move To Station Tab*).
- **Platooning** – Provides controls to manage vehicle platoons (see the *Platooning Tab*).

  NOTE: This feature is only available on MagneMover LITE and QuickStick configurations.
- **Vehicle Commands** – Provides vehicle-level commands (see the *Vehicle Commands Tab*).

Move To Position Tab

The **Move To Position** tab (see Figure 4-30) is used to move the specified vehicle to a specified position, relative to the start of a path using the specified parameters. See *Move to Position* on page 73 for usage.

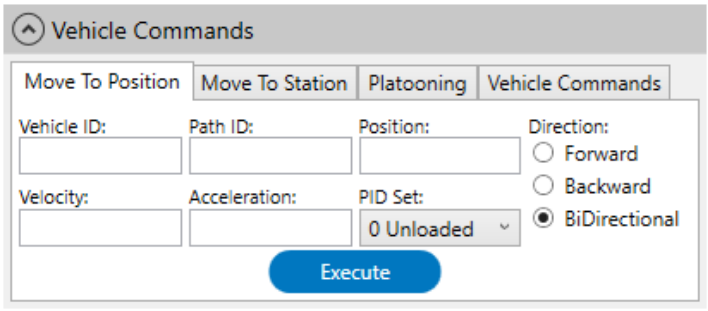


Figure 4-30: Vehicle Commands Section – Move To Position Tab

- **Vehicle ID** – The ID number of the vehicle.
- **Path ID** – The ID number of the path where the vehicle destination is located.
- **Position** – The vehicle destination from the start of the path that is specified in **Path ID**, in meters.
- **Direction** – Provides an option list for selecting the direction of motion for the vehicle.
  - **Forward** – The vehicle must move forward (downstream) to go to the designated position.
  - **Backward** – The vehicle must move backward (upstream) to go to the designated position.
  - **Bi-Directional** – The vehicle can move forward (downstream) or backward (upstream) to go to the designated position. When the destination is on the same path the vehicle is on, the vehicle moves in either direction to get to the destination in the shortest distance. When the destination is on a path other than the path the vehicle is on, the forward direction takes preference for a transport system that is a closed-loop.
- **Velocity** – The maximum velocity for the vehicle (in m/s).
- **Acceleration** – The maximum acceleration/deceleration for the vehicle to use (in m/s^2).
- **PID Set** – Provides a menu to select one of the PID sets defined in the Node Controller Configuration File to use when moving the vehicle.

  NOTE: If the transport system is not configured for loaded/unloaded PID values, the unloaded values are always used.

  - **0 Unloaded** – The vehicle is unloaded. If the transport system is configured for loaded/unloaded PID values, the unloaded values are used.

- **1 Loaded** – The vehicle is loaded. If the transport system is configured for loaded/unloaded PID values, the loaded values are used.
- **2…15** – The vehicle uses the specified custom PID set.

- **Execute** – Select to move the specified vehicle to the position on the specified path using the acceleration and velocity values provided.

Move To Station Tab

The **Move To Station** tab (see Figure 4-31) is used to move the specified vehicle to a specified station using the specified parameters. See *Move to Station* on page 75 for usage.

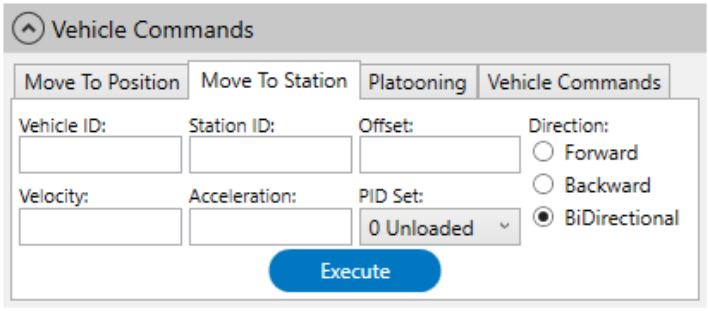


Figure 4-31: Vehicle Commands Section – Move To Station Tab

- **Vehicle ID** – The ID number of the vehicle.
- **Station ID** – The ID number of the station for vehicle destination on the path.

  NOTE: At least one station must be defined.
- **Offset** – Station offset, which is used when moving a vehicle to a station to shift the final position from the actual station position. Positive values are downstream of the station, negative values are upstream of the station.
- **Direction** – Provides an option list for selecting the direction of motion for the vehicle.
  - **Forward** – The vehicle must move forward (downstream) to go to the designated position.
  - **Backward** – The vehicle must move backward (upstream) to go to the designated position.
  - **Bi-Directional** – The vehicle can move forward (downstream) or backward (upstream) to go to a destination. When the destination is on the same path the vehicle is on, the vehicle moves in either direction to get to the destination in the shortest distance. When the destination is on a path other than the path the vehicle is on, the forward direction takes preference for a transport system that is a closed-loop.
- **Velocity** – The maximum velocity for the vehicle (in m/s).
- **Acceleration** – The maximum acceleration/deceleration for the vehicle to use (in m/s^2).
- **PID Set** – Provides a menu to select one of the PID sets defined in the Node Controller Configuration File to use when moving the vehicle.

  NOTE: If the transport system is not configured for loaded/unloaded PID values, the unloaded values are always used.

- **0 Unloaded** – The vehicle is unloaded. If the transport system is configured for loaded/unloaded PID values, the unloaded values are used.
- **1 Loaded** – The vehicle is loaded. If the transport system is configured for loaded/unloaded PID values, the loaded values are used.
- **2…15** – The vehicle uses the specified custom PID set.

- **Execute** – Select to move the specified vehicle to the specified station using the offset, acceleration, and velocity values provided.

Platooning Tab

The **Platooning** tab (see Figure 4-32) is used to couple and uncouple platoons. See *Vehicle Platooning* on page 77 for usage.

NOTE: This feature is only available on MagneMover LITE and QuickStick configurations.

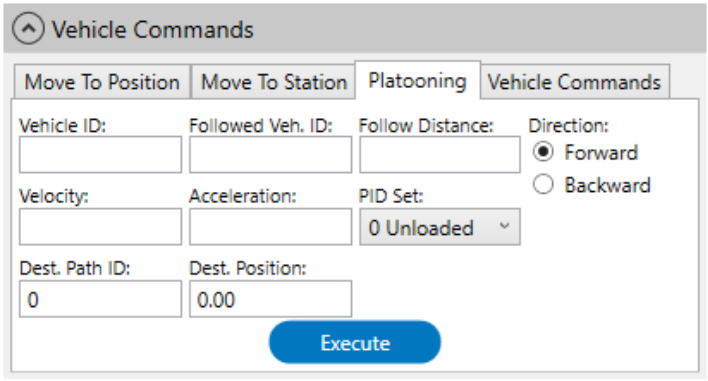


Figure 4-32: Vehicle Commands Section – Platooning Tab

- **Vehicle ID** – The ID number of the vehicle being configured.
- **Followed Veh. ID** (Followed Vehicle ID) – The ID number of the vehicle to be followed (that is, leading this vehicle in the platoon).
- **Follow Distance** – The distance center-to-center that this vehicle maintains behind the vehicle being followed (specified by **Followed Vehicle ID**) in the direction that is specified, once the pair of vehicles are coupled into a platoon.

  NOTE: When coupling to a stopped platoon, the vehicle must be within the Arrival Position Tolerance setting (defined on the Global Settings page in the Configurator when creating the Node Controller Configuration File) of this distance before sending the vehicle follow order.
- **Direction** – Provides an option list for selecting the direction of motion for the vehicle.
  - **Forward** – The vehicle to follow is downstream of this vehicle (motion is forward).
  - **Backward** – The vehicle to follow is upstream of this vehicle (motion is backward).
- **Velocity** – The maximum velocity for the vehicle to use (in m/s) when the vehicle is catching up to a moving vehicle to follow or when decoupling from a moving vehicle.

  NOTE: Currently not used for coupling as the vehicles must not be moving when creating a platoon, but it must be defined.

- **Acceleration** – The maximum acceleration/deceleration for the vehicle to use (in m/s^2) when the vehicle is catching up to a moving vehicle to follow or when decoupling from a moving vehicle.

  NOTE: Currently not used for coupling as the vehicles must not be moving when creating a platoon, but it must be defined.

- **PID Set** – Provides a menu to select one of the PID sets defined in the Node Controller Configuration File to use when moving the vehicle.

  NOTE: If the transport system is not configured for loaded/unloaded PID values, the unloaded values are always used.

  - **0 Unloaded** – The vehicle is unloaded. If the transport system is configured for loaded/unloaded PID values, the unloaded values are used.
  - **1 Loaded** – The vehicle is loaded. If the transport system is configured for loaded/unloaded PID values, the loaded values are used.
  - **2…15** – The vehicle uses the specified custom PID set.

- **Dest. Path ID** (Destination Path ID) – The path ID number for the automatic decoupling position.

  NOTE: This feature is only available on MagneMover LITE configurations.

  - **1…65535** – The ID number of the path where the automatic decoupling position is located (where the vehicle stops after auto-decoupling while in motion).
  - **0** – The automatic decoupling feature is disabled.

- **Dest. Position** (Destination Position) – The decoupling destination for the vehicle from the start of the path specified in **Dest. Path ID** in meters (where the vehicle stops after automatic decoupling while in motion).

  NOTE: This feature is only available on MagneMover LITE configurations.

- **Execute** – Select to implement the specified platooning command using the values provided.

Vehicle Commands Tab

The **Vehicle Commands** tab (see Figure 4-33) is used to send vehicle-specific commands to the motors. See *Vehicle Commands* on page 81 for usage.

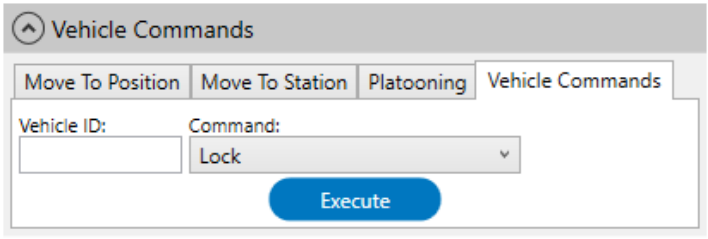


Figure 4-33: Vehicle Commands Section – Vehicle Commands Tab

- **Vehicle ID** – The ID number of the vehicle the command is applied to.
- **Command** – Provides a menu to select vehicle-specific commands.
  - **Lock** – Select to keep the specified vehicle from accepting new motion commands.

    NOTE: The vehicle lock status set using the NCHost TCP Interface Utility is temporary and is not restored if the node controller is reset or rebooted.
  - **Unlock** – Select to unlock a previously set lock, which allows the vehicle to be moved.

    NOTE: The vehicle lock status set using the NCHost TCP Interface Utility is temporary and is not restored if the node controller is reset or rebooted.
  - **Delete** – Deletes the specified vehicle record from the node controller and all motors. This function is typically used when a vehicle is physically removed from the guideway because it is inoperable. It can also be used when a vehicle entry is canceled or a vehicle has been manually moved and the motor can no longer track it. If a vehicle entry or exit is active for the vehicle, the entry/exit state machine is also cleared.

    IMPORTANT Using the Delete Vehicle command to delete a vehicle that is physically present on the transport system can cause collisions as the system no longer accounts for that vehicle as it moves other vehicles.
  - **Clear Suspect Bit** – Allows the Suspect bit to be cleared once control of the vehicle is regained.
- **Execute** – Select to implement the specified vehicle command using the values provided.

Node Commands

The **Node Commands** section (see Figure 4-34) is used to test motion of a vehicle through a Terminus Node, a Gateway Node, or a Moving Path Node. See *Testing the Transport System – Path Commands* on page 63 for usage.

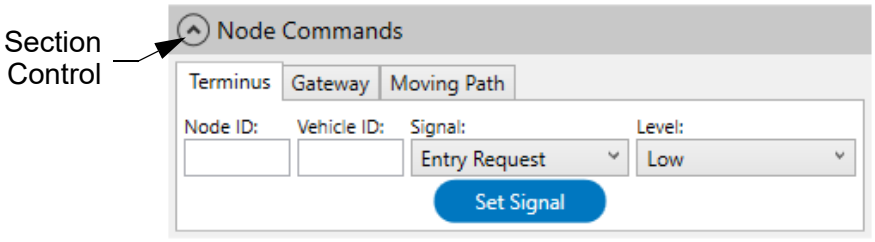


Figure 4-34: Settings Panel, Node Commands Section

- **Section Control** – Select to collapse or expand this section.
- **Terminus** – Provides controls to test vehicle motion off or onto the transport system through a Terminus Node (see the *Terminus Tab*).
- **Gateway** – Provides controls to enable/disable the Gateway Node (see the *Gateway Tab*).
- **Moving Path** – Provides controls to test motion of a vehicle through a Moving Path Node (see the *Moving Path Tab*).

  NOTE: This feature is only available on QuickStick and QSHT configurations.

Terminus Tab

The **Terminus** tab (see Figure 4-35) is used to set the Terminus Node handshake signals to test vehicle motion off or onto the transport system through a Terminus Node. See *Terminus Node* on page 85 for usage.

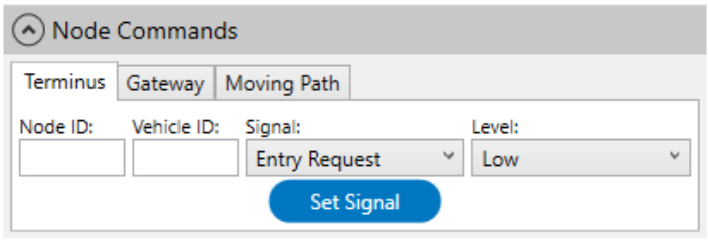


Figure 4-35: Node Commands Section – Terminus Tab

- **Node ID** – The ID number of the Terminus Node, as defined in the Node Controller Configuration File.
- **Vehicle ID** – The ID number of the vehicle to be moved through the node.
- **Signal** – Provides a menu to specify direction of motion through the node.
  - **Entry Request** – Indicates that a vehicle is ready to enter the MagneMotion transport system at the Terminus Node.
  - **Exit Allowed** – Indicates that the external equipment is ready to receive a vehicle from the MagneMotion transport system through the Terminus Node.
- **Level** – Provides a menu to set the value of the Terminus Node handshaking **Signal**.
  - **Low** (Entry Request) – No request active (Signal = 0).
  - **High** (Entry Request) – Request for vehicle to enter the transport system (Signal = 1).
  - **Low** (Exit Allowed) – No permission granted (Signal = 0).
  - **High** (Exit Allowed) – Permission for vehicle to leave the transport system (Signal = 1).
- **Set Signal** – Select to send the configured Terminus command to the node controller.

Gateway Tab

The **Gateway** tab (see Figure 4-36) is used to enable the Gateway Node. Once enabled, vehicle motion from one HLC Control Group to another HLC Control Group through the Gateway Node can be tested. See *Gateway Node* on page 93 for usage.

NOTE: Gateway Nodes are not supported when running in simulation mode.

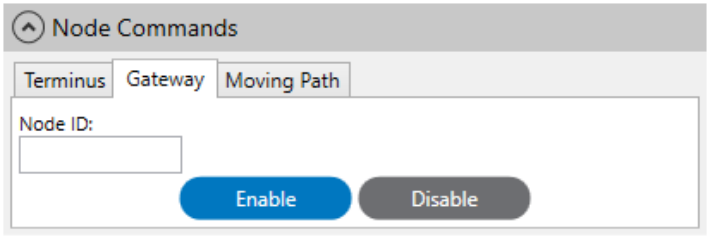


Figure 4-36: Node Commands Section – Gateway Tab

- **Node ID** – The ID number of the Gateway Node, as defined in the Node Controller Configuration File.
- **Enable** – Select to specify the Gateway Node can be used for vehicle transfer between HLC Control Groups.
- **Disable** – Select to specify the Gateway Node cannot be used for vehicle transfer between HLC Control Groups.

Moving Path Tab

The **Moving Path** tab (see Figure 4-37) is used to set the Moving Path Node handshake signals to test motion of a vehicle through a Moving Path Node. See *Moving Path Node* on page 97 for usage.

NOTE: This feature is only available on QuickStick and QSHT configurations.

Moving Path Nodes are not supported when running in simulation mode.

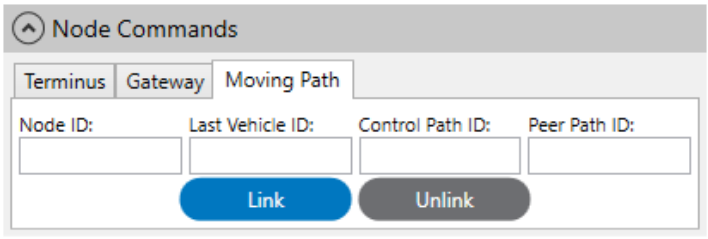


Figure 4-37: Node Commands Section – Moving Path Tab

- **Node ID** – The ID number of the Moving Path Node, as defined in the Node Controller Configuration File.
- **Last Vehicle ID** – The path junction is automatically unlinked when this vehicle clears the junction. If set to 0, the path junction remains linked until explicitly unlinked.
- **Control Path ID** – The ID number of the Control Path end to link to the specified Peer Path end to form a path junction.
- **Peer Path ID** – The ID number of the Peer Path end to link to the specified Control Path end to form a path junction.
- **Link** – Select to link the Peer Path to the Control Path.
- **Unlink** – Select to unlink the Peer Path from the Control Path.

Host Commands

The **Host Commands** section (see Figure 4-38) is used to send basic host-level commands to the transport system HLC.

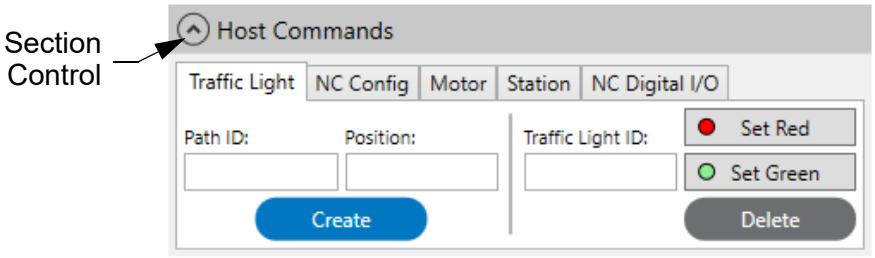


Figure 4-38: Settings Panel, Host Commands Section

- **Section Control** – Select to collapse or expand this section.
- **Traffic Light** – Provides controls to create temporary Traffic Lights and set the color of any traffic light (see the *Traffic Light Tab*).
- **NC Config** – Provides controls to select and activate a managed Node Controller Configuration File (see the *NC Config Tab*).
- **Motor** – Provides controls to enable or disable the inverters on specified motors (see the *Motor Tab*).

  NOTE: This feature is only available on QSHT configurations.
- **Station** – Provides controls to create temporary stations (see the *Station Tab*).
- **NC Digital I/O** – Provides controls to exercise digital I/O bits (see the *NC Digital I/O Tab*).

  NOTE: This feature is only available on node controllers that provide digital I/O.

Traffic Light Tab

The **Traffic Light** tab (see Figure 4-39) is used to create temporary traffic lights for use during system testing. Any traffic light in the system can be set to either red or green. When the traffic light is set to green, vehicles are granted permission to move beyond the position of the traffic light. When the traffic light is set to red, vehicles are not granted permission to move beyond the position of the traffic light. See *Traffic Lights* on page 101 for usage.

NOTE: Traffic lights created using the NCHost TCP Interface Utility are temporary and are not restored if the node controller is reset or rebooted.

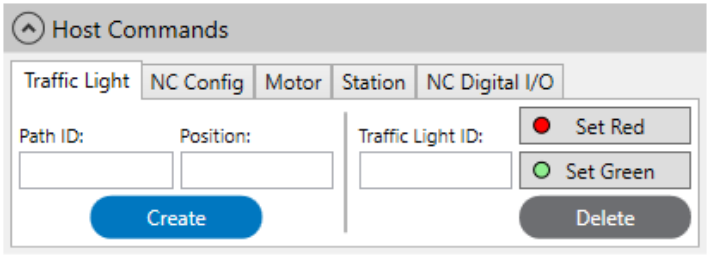


Figure 4-39: Host Commands Section – Traffic Light Tab

- **Path ID** – The ID number of the path where the traffic light is to be located.
- **Position** – The traffic light location from the start of the path in meters.
- **Create** – Select to create a traffic light at the specified position on the specified path for testing. The new traffic light is assigned the next available Traffic Light ID.
- **Traffic Light ID** – The ID number of the traffic light to be controlled or deleted.
- **Set Red** – Set the specified traffic light to red.
- **Set Green** – Set the specified traffic light to green.
- **Delete** – Select to delete the specified traffic light.

NC Config Tab

The **NC Config** tab (see Figure 4-40) is used to select and activate a managed Node Controller Configuration File for use during testing. See *NC Configurations* on page 103 for usage.

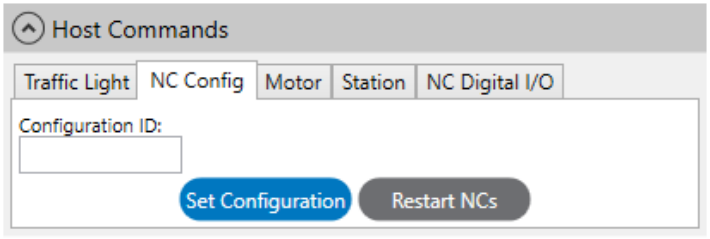


Figure 4-40: Host Commands Section – NC Config Tab

- **Configuration ID** – The ID number of one of the Managed Node Controller Configuration File options that are stored on the high-level controller.
- **Set Configuration** – Select to distribute the selected Managed Node Controller Configuration File to all node controllers.
- **Restart NCs** – Select to force a restart of all node controllers, which causes them to start using the new configuration file.

  NOTE: This will cause the NCHost TCP Interface Utility to be disconnected from the HLC.

Motor Tab

The **Motor** tab (see Figure 4-41) is used for enabling or disabling the inverters on specified motors. See *Motor Inverter Control* on page 104 for usage.

NOTE: This feature is only available on QSHT configurations.

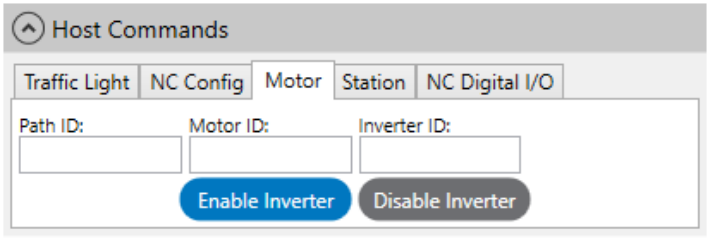


Figure 4-41: Host Commands Section – Motor Tab

- **Path ID** – The ID number of the path where the motor is located.
- **Motor ID** – The ID number of the motor to be commanded.
- **Inverter ID** – The ID number of the inverter controlling the blocks to be enabled/disabled.
- **Enable Inverter** – Select to turn on use of the specified inverter. Vehicle thrust is enabled on the blocks that inverter controls.
- **Disable Inverter** – Select to turn off use of the specified inverter. Vehicle thrust is disabled on the blocks that inverter controls.

Station Tab

The **Station** tab (see Figure 4-42) is used to create temporary stations for use during system testing. See *Temporary Stations* on page 105 for usage.

NOTE: Stations that are created using the NCHost TCP Interface Utility are temporary and are not restored if the node controller is reset or rebooted.

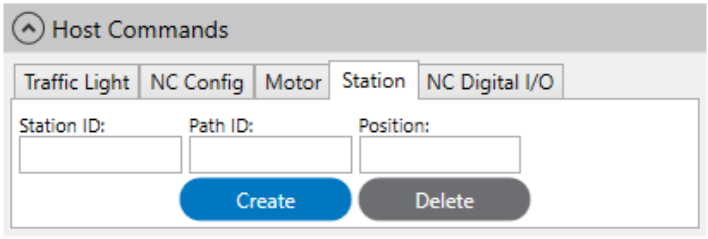


Figure 4-42: Host Commands Section – Station Tab

- **Station ID** – The ID number for the station to be created or deleted.
  - If **Station ID** is set to a specific station when a Create or Delete command is issued, the command is applied to that station only. Entering an existing Station ID reassigns that station to the new location.
  - If **Station ID** is set to zero when a Create command is issued, the next available ID is assigned.
  - If **Station ID** is set to zero when a Delete command is issued, all stations are deleted.
- **Path ID** – The ID number of the path for the station.
- **Position** – The location for the station from the start of the path, in meters.
- **Create** – Select to create a station at the specified position on the specified path for testing.
- **Delete** – Select to delete the specified station.

NC Digital I/O Tab

The **NC Digital I/O** tab (see Figure 4-43) is used for exercising the digital I/O bits on specified node controllers during system testing. See *Node Controller Digital I/O* on page 107 for usage.

NOTE: This feature is only available on node controllers that provide digital I/O.

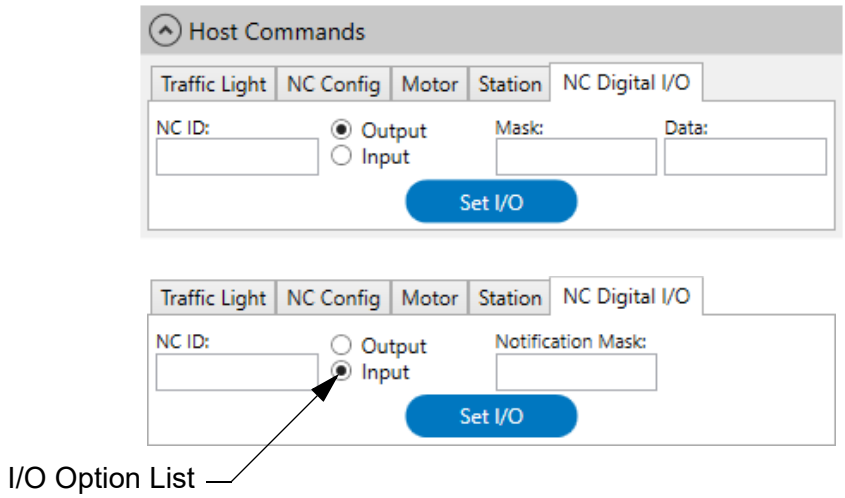


Figure 4-43: Host Commands Section – NC Digital I/O Tab

- **NC ID** – The ID number of the node controller to be commanded.
- **I/O Option List** – Specifies the type of digital I/O operation to execute.
  - **Output** – Select to specify a digital output operation. The output **Mask** and **Data** fields are displayed.
  - **Input** – Select to specify a digital input operation. The **Notification Mask** field is displayed.
- **Mask** – Displayed when **Output** is selected. Text field for the entry of the output data mask as a four-digit hex value, which specifies the data bits to be passed to the output (mask bit set high).
- **Data** – Displayed when **Output** is selected. Text field for the entry of the output data as a four-digit hex value.
- **Notification Mask** – Displayed when **Input** is selected. Text field for the entry of the input data notification mask as a four-digit hex value. This mask identifies the digital I/O input bits that are enabled (bit set high) to send unsolicited notification messages to the HLC whenever a state change is detected.
- **Set I/O** – Select to execute the digital I/O input/output operation.

  NOTE: The results of the digital I/O host commands are displayed in the Node Controller Status display (see *System Status* on page 150).

  Additionally, the digital I/O commands and responses can be viewed in the NCHost Diagnostic Output window (see *Diagnostic Messages* on page 61).

Demo Scripts

The **Demo Script** section (see Figure 4-44) is used to load and run Demo Script files. See *Testing the Transport System – Demo Scripts* on page 110 for usage.

NOTE: This section is only displayed when Demo Script is selected on the **View** menu.

Changes made to the currently running script file do not become active until the changes are saved, the demo is stopped, the updated script file is loaded, and started.

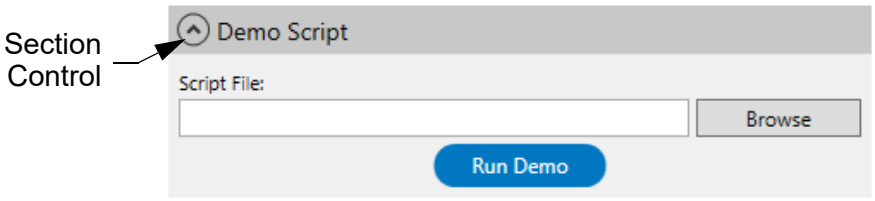


Figure 4-44: Settings Panel, Demo Script Section

- **Section Control** – Select to collapse or expand this section.
- **Script File** – Displays the path and name of the selected demo script file.
- **Browse** – Select to open a dialog box to locate the demo script file.
- **Run Demo/Stop Demo** – Toggles between Run Demo and Stop Demo based on the current status.
  - **Run Demo** – Displayed when there is no script file currently running. Select to start execution of the file that is shown in the **Script File** text box.
  - **Stop Demo** – Displayed when there is a script file currently running. Select to stop execution of the file.

Engineering Tools

The **Engineering Tools** section (see Figure 4-45) is used to access tools that are used for fine-tuning the transport system.

NOTE: This section is only displayed when Engineering Tools is selected on the **View** menu.

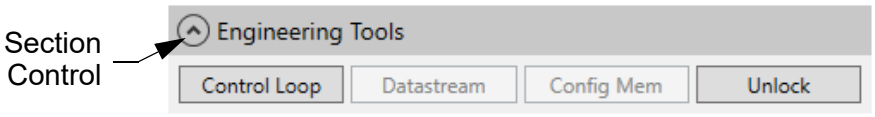


Figure 4-45: Settings Panel, Engineering Tools Section

- **Section Control** – Select to collapse or expand this section.
- **Control Loop** – Select to open a dialog box for tuning the motion control loop (see *Tune Control Loop Parameters* and Figure 4-46).
- **Datastream** – For MagneMotion engineering use only.
- **Config Mem** – For MagneMotion engineering use only.
- **Propulsion Power Supplies** – Available in the Host Commands section for MagneMotion engineering use only.
- **Unlock** – Select to open a dialog box for entry of the password to unlock the MagneMotion engineering functions.

Tune Control Loop Parameters

Different control loop PID values can be tested on specific paths within the transport system to optimize vehicle motion. See *Testing the Transport System – Refining the Control Loop* on page 111 for usage.

NOTE: PID parameter changes made using the NCHost TCP Interface Utility are temporary and are not restored when the node controller is reset or rebooted.

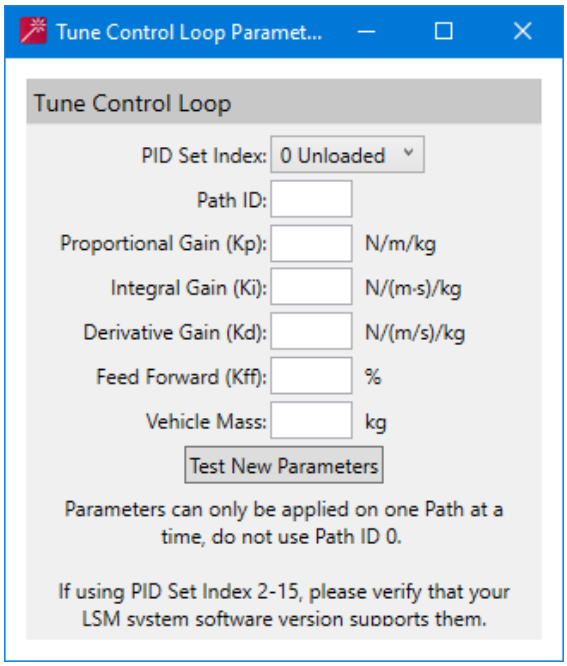


Figure 4-46: Tune Control Loop Parameters

- **PID Set Index** – Provides a menu to select the PID control loop parameter set to use to apply the new values (see either the *MagneMover LITE Configurator User Manual*, MMI-UM008, or the *QuickStick Configurator User Manual*, MMI-UM009*)*.
  - **0 Unloaded** – The PID values are assigned to Set 0. This PID set is predefined as the vehicle is unloaded. If the transport system is configured to use loaded/unload PID values, the unloaded (Set 0) values are used whenever a vehicle is defined as being unloaded.
  - **1 Loaded** – The PID values are assigned to Set 1. This PID set is predefined as the vehicle is loaded. If the transport system is configured to use loaded/unloaded PID values, the loaded (Set 1) values are used whenever a vehicle is defined as being loaded.
  - **2…14** – The PID values are assigned to Set 2…14.
  - **15** – The PID values are assigned to Set 15. This PID Set is used during startup. If it is not defined, PID Set 0 is scaled by 25% and used for startup.
- **Path ID** – The ID number of the path for testing the new control loop parameters.

- **Proportional Gain (Kp)** – The control loop proportional gain (controls the amount of force applied proportional to the position error).
- **Integral Gain (Ki)** – The control loop integral gain (controls the amount of force applied proportional to the integral of the position error, correcting errors in position over time).
- **Derivative Gain (Kd)** – The control loop derivative gain (controls the amount of force applied proportional to the velocity error, providing damping in the control loop).
- **Feed Forward (Kff)** – The control loop feed forward scale (increases or decreases the feed-forward force without affecting other control loop gains). Not implemented on 0.9.x version of the Node Controller Software Image File that is used with QuickStick systems. The default is 100%.
- **Vehicle Mass** – The mass of the vehicle (V*ehicle* + *MagnetArray* + *Payload*). The mass is used as a scaling factor for the rest of the values in the PID Set.
- **Test New Parameters** – Select to apply the new PID values to the specified path for testing.

Vehicles

The **Vehicles** pane (see Figure 4-47 and Figure 4-48), in the upper right of the NCHost TCP Interface Utility window as shown in Figure 4-9, displays the status of all vehicles currently on the transport system. See *Vehicle Status* on page 59 for usage.

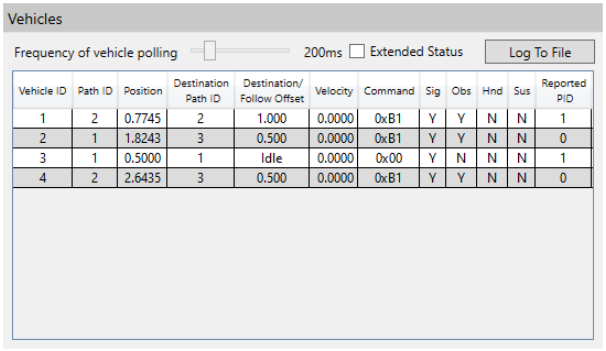


Figure 4-47: Vehicles Pane

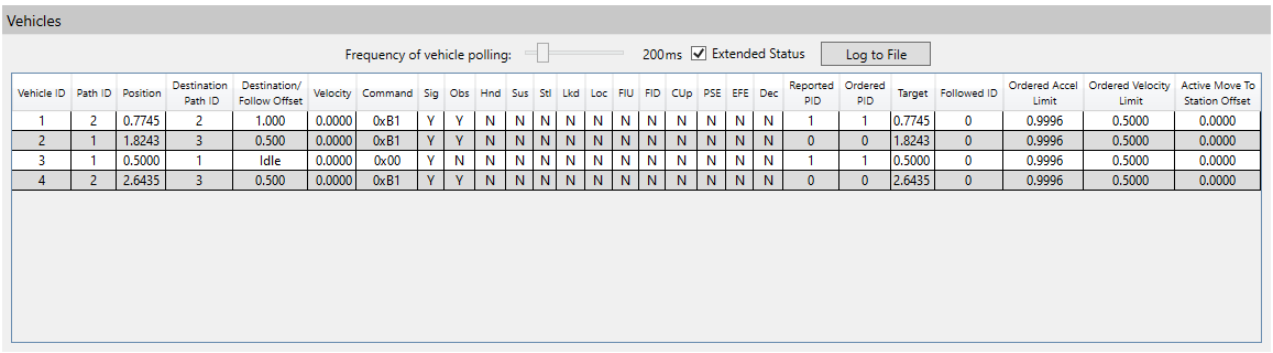


Figure 4-48: Vehicles Pane Showing Extended Status

Select any of the column headers to sort the contents of the Vehicles pane by that column. An arrow is displayed that indicates the sort order. Select any of the dividers between columns to change the width of the column to the left.

- **Frequency of vehicle polling** – Defines the rate at which the data within the **Vehicles** pane is updated. The slider range is 20…1000 milliseconds (1 second).
- **Extended Status** – Select to display extended vehicle status.
- **Log To File/Close Log** – Toggles between **Log to File** and **Close Log** based on the current log status.
  - **Log To File** – Displayed when there is no log file currently open. Select to open a dialog box to either create a file or select an existing file and start logging vehicle data to that file. Data is logged at the rate set by the **Frequency of vehicle polling** slider.
  - **Close Log** – Displayed when there is a log file in use. Select to stop file logging. The log file is locked while in use, once the log file is closed, the file can be opened to view the contents using a text editor.

- **Vehicle ID** – Displays the ID number of each vehicle in the transport system as assigned by the high-level controller.
- **Path ID** – Displays the ID number of the path, as defined in the Node Controller Configuration File, where the vehicle is located.
- **Position** – Displays the last reported position of the vehicle in meters from the beginning of the path where the vehicle is located.
- **Destination Path ID** – Displays the ID of the path where the vehicle is headed. Equal to the ID of the path where the vehicle is located if motion has completed. Equal to 0 if the vehicle has never moved.
    - ***value*** – For one vehicle or the lead vehicle in a platoon, when the vehicle is under a motion command, this value is the ID of the path where the vehicle is heading. When vehicle motion has completed, this value is the ID of the path where the vehicle is located.
    - **0** – Indicates that the vehicle has never moved.
    - **Following** – The vehicle is following another vehicle in a platoon. The destination of the lead vehicle determines the destination for all vehicles in the platoon.
- **Destination/Follow Offset** – Displays the commanded position on the destination path or the vehicle movement status for the vehicle.
    - ***value*** – For one vehicle or the lead vehicle in a platoon, when the vehicle is under a motion command, this value is the commanded position on the destination path. When the vehicle is coupled into a platoon, this value is the distance to the vehicle it is following even if it is not moving.
    - **Idle** – The vehicle is at the commanded position. Or the vehicle has never moved.
- **Velocity** – Displays the current velocity of the vehicle (in m/s).
- **Command** – Displays the current command from the host controller for the vehicle.
- **Sig** (Signal Detected) – Displays the detection status for the vehicle as determined by the motor currently in charge of the vehicle.
    - **N** – The motor does not detect the vehicle (detection and location of the vehicle on the path is lost).
    - **Y** – Vehicle has been detected and its location identified.
- **Obs** (Obstructed) – Displays the vehicle obstruction status.
    - **N** – The vehicle is not obstructed and is free to move.
    - **Y** – The vehicle is obstructed and waiting for permission to move further. This happens when a vehicle is in the way (entering a switch) or a hardware fault.
- **Hnd** (Hindered) – Displays the vehicle hindered (jam) status.
    - **N** – The vehicle is not hindered and moving as expected.

- **Y** – The vehicle is not moving despite having permission to move. This happens when a vehicle is held back by an external force including a foreign object jamming a vehicle on the guideway or someone physically holding the vehicle back. The motor continues to apply force on the vehicle to try to move it indefinitely.

- **Sus** (Suspect) – Displays the vehicle suspect status.

  NOTE: This feature is only available on MM LITE and QuickStick configurations.

  - **N** – The vehicle is not suspect and is free to move.
  - **Y** – The vehicle is suspect. The motor has detected that the vehicle has been manually moved out of the control region and the motor cannot guarantee vehicle control, even after moving the vehicle back into its control region.

    NOTE: Suspect vehicles should be deleted, then restart each path reporting an unlocated vehicle fault.

- **Stl** (Stall) – Displays the vehicle stall status (only displayed when **Extended Status** is selected).
  - **N** – The vehicle is not stalled and is free to move.
  - **Y** – The vehicle has stalled. The motor has detected that the vehicle is not moving and the motor has had to reduce power to prevent overheating.
- **Lkd** (Locked) – Displays the vehicle lock status (only displayed when **Extended Status** is selected). See *Lock Vehicle*.
  - **N** – The vehicle is not locked and is free to move.
  - **Y** – The vehicle is locked and cannot be moved.
- **Loc** (Located) – Displays the vehicle locate status (only displayed when **Extended Status** is selected).
  - **N** – Vehicle locate is not completed (startup is in progress).
  - **Y** – Vehicle locate has completed.
- **FlU** (Following Upstream) – Displays the vehicle following upstream status (only displayed when **Extended Status** is selected).
  - **N** – This vehicle is not following another vehicle or not following a vehicle upstream.
  - **Y** – This vehicle is in a platoon and following the vehicle that is specified in the **Followed ID** field and that vehicle is upstream from this vehicle.
- **FlD** (Following Downstream) – Displays the vehicle following downstream status (only displayed when **Extended Status** is selected).
  - **N** – This vehicle is not following another vehicle or not following a vehicle downstream.
  - **Y** – This vehicle is in a platoon and following the vehicle that is specified in the **Followed ID** field and that vehicle is downstream from this vehicle.

- **CUp** (Caught Up) – Displays the vehicle caught up status (only displayed when **Extended Status** is selected).
  - **N** – This vehicle is not following another vehicle.
  - **Y** – This vehicle is in a platoon and is caught up with the vehicle it is following and is at the requested following distance.
- **PSE** (Profile Stale Error) – Displays the vehicle profile stale error status (only displayed when **Extended Status** is selected).
  - **N** – This vehicle is not following another vehicle.
  - **Y** – This vehicle is in a platoon and has not received profile data (acceleration and velocity) from the vehicle it is following in the expected time (~10 ms). The vehicle slows to a stop. If the profile data is restored, the vehicle catches up to its leader using the catchup acceleration and velocity.
- **EFE** (Excessive Following Error) – Displays the vehicle excessive following error status (only displayed when **Extended Status** is selected).
  - **N** – This vehicle is not following another vehicle.
  - **Y** – This vehicle is in a platoon and is too far from the vehicle it is following (possibly hindered or stopped receiving following profile data).
- **Dec** (Decoupled) – Displays the vehicle auto-decouple status (only displayed when **Extended Status** is selected).
  - **N** – This vehicle is a platoon follower that is not in the decoupling state, or not a platooned vehicle.
  - **Y** – This vehicle is in a platoon and is decoupling from the vehicle it is following. This bit is set when the motor reports decoupling and stays set until the decoupled vehicle reaches its new destination under the 0xB1 position order.
- **Reported PID** – Displays the PID Set being used by the vehicle for motion.
  - **0** – The vehicle is not loaded (using PID Set index 0).
  - **1** – The vehicle is loaded (using PID Set index 1).
  - **2…14** – The vehicle is using the specified PID Set.
  - **15** – Startup PID values. This PID Set is used during startup. If it is not defined, PID Set 0 is scaled by 25% and used for startup.
- **Ordered PID** – Displays the currently commanded PID Set for use for vehicle motion during the current move command (only displayed when **Extended Status** is selected).
  - **0** – The vehicle is not loaded (PID Set index 0).
  - **1** – The vehicle is loaded (PID Set index 1).
  - **2…14** – The vehicle uses the requested PID Set.
  - **15** – The vehicle uses startup PID values.

- **Target** – Displays the next interim target location (in meters) that the vehicle has permission to move to from the beginning of the path where the vehicle is located. It is equal to the position of the vehicle when the vehicle is not moving (only displayed when Extended Status is selected).
- **Followed ID** – Displays the ID number of the vehicle that this vehicle is following when this vehicle is under a vehicle follow order and is part of a platoon. Equal to 0 if this vehicle is not under a vehicle follow order (only displayed when **Extended Status** is selected).
- **Ordered Accel Limit** (Ordered Acceleration Limit) – Displays the commanded acceleration and deceleration for vehicle motion during the current move command (only displayed when Extended Status is selected).
- **Ordered Velocity Limit** – Displays the commanded maximum velocity for vehicle motion during the current move command (only displayed when Extended Status is selected).
- **Active Move to Station Offset** – Displays the commanded offset for the vehicle from the station that is the current destination (only displayed when Extended Status is selected and the TCP/IP interface is being used to move vehicles to stations). See *Offset*.
- **Horizontal Scroll Bar** – Scrolls the status columns to display all columns. The scroll bar appears below the display when the vehicle display contains more vehicle status columns than can be displayed in the screen space provided (typically when Extended Status is selected).
- **Vehicle Scroll Bar** – Scrolls the list of vehicles to display all vehicles on the transport system. The scroll bar appears when the vehicle display is populated with more vehicles than can be displayed in the screen space provided.

Diagnostic Output

The **Diagnostic Output** pane (see Figure 4-49), in the lower right of the NCHost TCP Interface Utility window. This pane displays a running log of actions that are performed through the NCHost TCP Interface Utility and the responses from the transport system. See *Diagnostic Messages* on page 61 for usage.

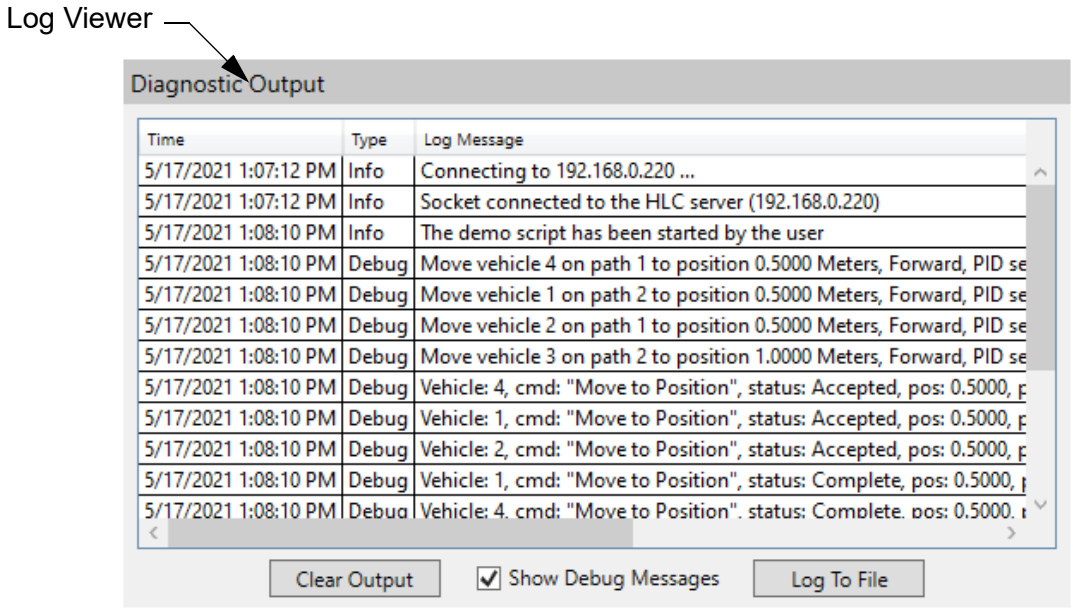


Figure 4-49: Diagnostic Output Pane

Select any of the column headers to sort the contents of the Diagnostic Output pane by that column. An arrow is displayed that the column being referenced and the sort order.

- **Log Viewer** – Displays the log of all NCHost actions and responses.
  - **Time** – Displays the date and time the message was sent.
  - **Type** – Displays the type of message.
    - **Debug** – Provides information about key events that are related to an executing Demo Script and application debug information for MagneMotion engineers when an error occurs.
    - **Error** – Indicates an error that is related to the use of one of the application features or an error that is related to a command sent to the HLC.
    - **Info** – Provides informational feedback that is related to the use of one of the application features or a command sent to the HLC.
  - **Log Message** – Text of message.
- **Clear Output** – Select to delete the contents of the Log Viewer.
- **Show Debug Messages:** – Select to include Debug messages in the log. The default is selected (on).

- **Log To File/Close Log** – Toggles between **Log to File** and **Close Log** based on the current log status.
  - **Log To File** – Displayed when there is no log file currently open. Select to open a dialog box to either create a file or select an existing file and start logging diagnostic messages to that file. Data is logged as it is received.
  - **Close Log** – Displayed when there is a log file in use. Select to stop file logging. The log file is locked while in use, once the log file is closed, the file can be opened to view the contents using a text editor.

Demo Script Command Reference 5

Overview

This chapter describes the sections of the Demo Script and the commands that can be used in each section. See *Demo Scripts* on page 113 for instructions about creating and using Demo Scripts.

Included in this chapter are:

- An overview of the Demo Script structure, including the use of comments and formatting in the Demo Script.
- Global Directives, which define the general behavior characteristics for all vehicles on the paths that are specified in the Demo Script.
- Vehicle Directives, which define the individual movement characteristics for specific vehicles in the Demo Script.

NOTE: The commands that are used in the Demo Scripts are not the same as the commands used by the host controller for running the transport system during production. For information on the command protocol, see the *Host Controller TCP/IP Communication Protocol User Manual*, MMI-UM003, or the *Host Controller EtherNet/IP Communication Protocol User Manual*, MMI-UM004.

This Command Reference reflects the version of the software that is indicated in *Changes* at the front of the manual. Specific builds of the NCHost TCP Interface Utility may not implement all features that are described in this manual. See the Release Notes that are supplied with this application for more information.

Demo Script Structure

The following example of a Demo Script shows the general format of a simple script for two vehicles on a straight track. This track is at least 6 meters long with one path. When this script is executed, the first vehicle moves to a position 1.0 meter from the start of the track and the second vehicle moves to a position 4.0 meters from the start of the track.

Once both vehicles have arrived at their initial position the first vehicle moves to a position 2.0 meters from the start of the track and the second vehicle moves to a position 5.0 meters from the start of the track. Once both vehicles have arrived at their second position the script loops with both vehicles moving from their first position to their second position.

```
# Global Directives
Paths 1
Acceleration 5.0
Velocity 2.5

# Vehicle Directives
Vehicle
    DemoStep 1.0
    DemoStep 2.0

Vehicle
    DemoStep 4.0
    DemoStep 5.0
```

NOTE: Make sure that each step is one line in the script file. Some of the steps in the examples that are shown in this manual are on multiple lines due to “text wrap” line-space restrictions.

See *Demo Scripts* on page 113 for information on Demo Script creation and use.

Global Directives are the first group of commands. See *Global Directives* on page 200 for the definitions of the individual commands.

Vehicle Directives are the next group of commands. See *Vehicle Directives* on page 209 for the definitions of the individual commands.

Comments are lines that are preceded by a # that allows the insertion of annotations that are not executed in the script to make it easier to read. See *Comments* on page 199 for the description and examples.

Blank Lines can be inserted into the script as desired to improve readability.

NOTE: All directives, commands, and command extensions that are described in this chapter are not case-sensitive.

Comments

Comments allow the insertion of notes or other information helpful to read and troubleshoot the Demo Script, but are not executed. Precede each comment line with # (pound sign), to indicate it is a comment.

NOTE: In-line comments are not supported, each comment must be on its own line.

Format

```
# This line is a comment
```

Examples

Both examples that are shown in Figure 5-1 define two vehicles on a loop track and are executed the same. The second example adds comments and blank lines to make the script more readable.

```
Paths 1
Velocity 0.5
Acceleration 1.0
Direction Forward
Demo_Arrival_Method Single
Vehicle
    DemoStep 1.0
    DemoStep 3.0
Vehicle
    DemoStep 1.0
    DemoStep 3.0
```

Example 1

```
# Global Directives
Paths 1
Velocity 0.5
Acceleration 1.0
Direction Forward
Demo_Arrival_Method Single

# Vehicle Directives
# 1
Vehicle
    DemoStep 1.0
    DemoStep 3.0

# 2
Vehicle
    DemoStep 1.0
    DemoStep 3.0
```

Example 2

Figure 5-1: Demo Script Comments Example

Global Directives

Global Directives are commands that define the general behavior characteristics for all vehicles on the paths that are specified in the script. The Global Directive commands are defined in this section in the general order that they should appear in the script.

NOTE: All Global Directives and arguments are not case-sensitive.

The Vehicle IDs assigned by the HLC at startup are not used by the Demo Script Global Directives. Vehicle Directives that are used in Demo Scripts do not correspond to the Vehicle ID numbers that the HLC assigns during system startup.

Table 5-1: Global Directives

Directive	Type	Argument	Page
Paths	Required	*Path_ID*	201
Velocity	Optional	*Vel_Value*	202
Acceleration	Optional	*Acc_Value*	203
Direction	Optional	*Dir_Value*	204
Payload	Optional	*PID_Value*	205
PID_Set_Index	Optional	*Index_Value*	206
Demo_Arrival_Method	Optional	*Arv_Value*	207
Log	Optional	–	208

Paths

Purpose

Specifies the paths, as defined in the Node Controller Configuration File, to which the script applies. All scripts must start with this command and specify at least one path. When multiple paths are specified, the order they are listed in the command is the order the NCHost TCP Interface Utility uses to identify the vehicles on the paths.

When running a Demo Script, the NCHost TCP Interface Utility identifies all vehicles on a path starting from the zero position at the upstream end of the path. As such, the utility assigns the first vehicle command section in the script to the first vehicle detected on the first path listed. Vehicle command sections continue to be assigned to vehicles working downstream on each path that is listed until all command sections have been assigned to vehicles.

NOTE: This command is required in the Global Directives for all Demo Scripts.

Format

```
Paths Path_ID1 Path_ID2 ... Path_IDn
```

Arguments

Path_ID*n* – The Path ID for the paths to be included in the script. Separate multiple Path IDs by spaces.

Examples

The following example shows one path being specified.

```
Paths 1
```

The following example shows multiple paths being specified. Vehicles are identified first from the beginning of path 1, then from the beginning of path 3, and finally from the beginning of path 2.

```
Paths 1 3 2
```

Velocity

Purpose

Defines the maximum velocity of all vehicles on the paths that are specified in the script. If this command is not used, the script defaults to a velocity of 1.0 m/s.

NOTE: This command is optional for all Demo Scripts.

Format

```
Velocity Vel_Value
```

Arguments

Vel_Value – The maximum velocity (in meters/second) for all vehicles that the script is controlling. This value must be less than or equal to the value specified as the Velocity Limit in the Node Controller Configuration File.

See Also

Velocity on page 218

Example

The following example sets the velocity of all vehicles to 2.5 m/s.

```
Velocity 2.5
```

Acceleration

Purpose

Defines the maximum acceleration of all vehicles on the paths that are specified in the script. If this command is not used, the script defaults to an acceleration of 1.0 m/s^2.

NOTE: This Directive is optional.

Format

```
Acceleration Acc_Value
```

Arguments

Acc_Value – The maximum acceleration/deceleration (in meters/second^2) for all vehicles the script is controlling. This value must be less than or equal to the value specified as the Acceleration Limit in the Node Controller Configuration File.

See Also

Acceleration on page 217

Example

The following example sets the acceleration of all vehicles to 5.0 m/s^2.

```
Acceleration 5.0
```

Direction

Purpose

Defines the default direction of movement of all vehicles on the paths that are specified in the script. If this command is not used, the script defaults to Bi-Directional motion.

NOTE: This Directive is optional.

Format

```
Direction Dir_Value
```

Arguments

Dir_Value – The direction of movement for all vehicles.

Bi-Directional – Vehicles can move Downstream (Forward) or Upstream (Backward) to go to a destination. When the destination is on the same path the vehicle is on, the vehicle moves in either direction to get to the destination in the shortest distance. When the destination is on a path other than the path the vehicle is on, the forward direction takes preference for a transport system that is a closed-loop.

Forward – Vehicles can move in the Downstream direction only (towards the end of the path they are on).

Backward – Vehicles can move in the Upstream direction only (towards the beginning of the path they are on).

See Also

Direction on page 219

Example

The following example sets the direction of all vehicles to Forward (motion is downstream).

```
Direction Forward
```

Payload

Purpose

Defines the default load state of all vehicles on the paths that are specified in the script. The load state determines which PID values (specified in the Node Controller Configuration File) are used for motion control if loaded and unloaded PID values have been configured. If this command is not used, the script defaults to Unloaded (PID Set 0).

NOTE: This Directive is optional.

This Directive is deprecated. It only supports the Unloaded and Loaded PID sets. For full support of all PID sets, use *PID_Set_Index* on page 206.

Format

```
Payload PID_Value
```

Arguments

PID_Value – The load condition for movement of the vehicle.

Unloaded – Vehicle moves using the unloaded Control Loop parameters (PID Set 0).

Loaded – Vehicle moves using the loaded Control Loop parameters (PID Set 1).

See Also

PID_Set_Index on page 206
Payload on page 220
PID_Set_Index on page 221

Example

The following example sets the payload of all vehicles to Unloaded (all vehicles move using the unloaded PID values unless overridden by a specific DemoStep).

```
Payload Unloaded
```

PID_Set_Index

Purpose

Defines the default load state of all vehicles on the paths that are specified in the script. The load state determines which PID values (specified in the Node Controller Configuration File) are used for motion control if loaded and unloaded PID values have been configured. If this command is not used, the script defaults to Unloaded (PID Set 0).

NOTE: This Directive is optional.

Format

```
PID_Set_Index Index_Value
```

Arguments

Index_Value – The PID Set (0…15) to be used for the vehicle motion (Set 0 is predefined as unloaded, Set 1 is predefined as loaded, and Set 15 is predefined for startup).

If PID Set 15 is defined, it is used during startup. If it is not defined, PID Set 0 is scaled by 25% and used for startup.

See Also

Payload on page 205
Payload on page 220
PID_Set_Index on page 221

Example

The following example sets the payload of all vehicles to Unloaded (all vehicles move using the unloaded PID values unless overridden by a specific DemoStep).

```
PID_Set_Index 0
```

Demo_Arrival_Method

Purpose

Defines dependent or independent movement for all vehicles to the next DemoStep Position. See *Two Vehicles, Independent Movement* on page 122 for details. If this command is not used, the script defaults to dependent movement.

NOTE: This Directive is optional.

Format

```
Demo_Arrival_Method Arv_Value
```

Arguments

Arv_Value – Identifies the movement method that is used for vehicle motion.

Single – Allows independent execution of all steps in the script. The steps for each vehicle are completed independently of any other vehicles. This method allows individual vehicles to proceed to their next step position without waiting for other vehicles to finish executing the same step.

Together – Forces dependent execution of all steps in the script. Each step must be completed for every vehicle before the next step is started. This method causes vehicles to wait until all vehicles arrive at each step position before they all move to the next step position.

See Also

Rendezvous Point on page 222

Examples

The following example sets the script to run in an independent mode.

```
Demo_Arrival_Method Single
```

The following example sets the script to run in a dependent mode.

```
Demo_Arrival_Method Together
```

Log

Purpose

Sets the script to display additional information (DEMO log messages) in the NCHost Utility Diagnostic Output window. In addition, if Log To File is selected, a log file of the information that is displayed on the screen is created.

DEMO log messages include information about the script processing and execution and can be used for troubleshooting when creating a Demo Script file. If the information is captured in a log file, the contents of the file can then be viewed and searched for errors in the script.

NOTE: This Directive is optional and is not recommended for normal use as it can double the size of the log file. Normal log information is displayed in the Diagnostic Output window if the Log command is not included in the script file.

Format

```
Log
```

Example

The following example sets the script to log additional information as shown in Figure 5-2. Figure 5-3 shows the same log file without the additional information.

```
Log
```

```
Jul 16 11:29:53: DEMO: Logging demo script processing per Log directive...
Jul 16 11:29:53: DEMO: Found Vehicle directive # 1 on line 8
Jul 16 11:29:53: DEMO: Processing DemoStep at line 9 with step value: 0.25
Jul 16 11:29:53: DEMO: Processing DemoStep at line 10 with step value: 0.75
Jul 16 11:29:53: DEMO: Processing DemoStep at line 11 with step value: null
Jul 16 11:29:53: DEMO: This is a NULL step
Jul 16 11:29:53: DEMO: Found Vehicle directive # 2 on line 14
Jul 16 11:29:53: DEMO: Processing DemoStep at line 15 with step value: 0.75
Jul 16 11:29:53: DEMO: Processing DemoStep at line 16 with step value: 0.25
Jul 16 11:29:53: DEMO: Processing DemoStep at line 17 with step value: 0.50
Jul 16 11:29:53: DEMO: demo_execute_step index: 0 step: 0
Jul 16 11:29:53: Move vehicle 1 on path 1 to position 0.25 Meters, Bi-Directional,
loaded: 0, order# 1
Jul 16 11:29:53: DEMO: demo_execute_step index: 1 step: 0
Jul 16 11:29:53: Move vehicle 2 on path 2 to position 0.75 Meters, Bi-Directional,
loaded: 0, order# 2
```

Figure 5-2: Extended Log File

```
Jul 16 11:29:53: Move vehicle 1 on path 1 to position 0.25 Meters, Bi-Directional,
loaded: 0, order# 1
Jul 16 11:29:53: Move vehicle 2 on path 2 to position 0.75 Meters, Bi-Directional,
loaded: 0, order# 2
```

Figure 5-3: Typical Log File

Vehicle Directives

The Vehicle Directives are commands that define the individual movement characteristics for specific vehicles for one movement cycle. The command definition is shown in this section in multiple configurations for ease of documentation. Most of the command arguments that are shown can be included in one command as shown in some of the examples and in *Combine Command Extensions* on page 136 if desired.

NOTE: Vehicle Directives and arguments are not case-sensitive.

The Demo Script Vehicle Directives do not affect the Vehicle IDs assigned by the HLC at startup.

Table 5-2: Vehicle Command Structure

Command	Step Command	Command Extension	Argument	Page
Vehicle				210
All_Vehicles				212
DemoStep	*value*			213
		Path	*value*	214
		Delay	*value*	215
		DelayOnce	*value*	216
		Acceleration	*value*	217
		Velocity	*value*	218
		Direction	*value*	219
		Payload	*value*	220
		PID_Set_Index	*value*	221
	Rendezvous	Point	*value*	222
	Freeze			223
		Chain	*value*	229
	Null			224
		Delay	*value*	215
		DelayOnce	*value*	216
	Stop			226
		Iterations	*value*	228
		Chain	*value*	229
		SyncPath *value* Count *value* Chain *value*		230

Vehicle

Purpose

Identifies a vehicle in the Demo Script. This command must precede the DemoSteps for each vehicle in the script to identify them as being for a specific vehicle. There is no need to identify the individual vehicles, however it can be convenient to place a comment before each vehicle for identification as shown in the example.

The NCHost TCP Interface Utility identifies all vehicles on a path starting from the zero position at the upstream end of the path where they are located when the software is started. For Demo Scripts only, the order of the paths for vehicle identification is specified in the **Paths** Global Directive (see *Paths* on page 201). The Demo Script **Paths** Global Directive has no effect on the Vehicle IDs assigned by the HLC at startup.

NOTE: When the NCHost TCP Interface Utility assigns vehicles to the vehicle directives, the vehicle with an ID of 1 (as assigned by the HLC) may not be assigned to the first vehicle directive in the Demo Script. There is no correlation to Vehicle IDs in Demo Scripts and Vehicle IDs assigned to vehicles by the HLC during system startup operations. Vehicle IDs entered or specified in Demo Scripts are arbitrary and only apply to vehicles involved in the execution of the Demo Script directives. Indicating the Vehicle ID in the script is not important or necessary. The NCHost TCP Interface Utility assigns the first vehicle command section in the script to the first vehicle detected at the upstream end of the first path listed. The NCHost Utility continues assigning vehicle command sections in the order they appear in the script, to the next vehicle detected in the downstream path.

Demo Scripts cannot be run if the number of Vehicle sections in the script do not match the number of vehicles on the paths that are included in the script.

When multiple vehicles are used, the Vehicle Directives for all vehicles must have the same number of DemoStep commands.

Format

```
Vehicle
```

Example

The following example moves each vehicle as directed until stopped. The number of `Vehicle` sections in the script must match the number of vehicles on the system.

```
Paths 1 2 3
Velocity 0.5
Acceleration 1.0
Direction Forward
Demo_Arrival_Method Single
```

```
# 1
Vehicle
    DemoStep 0.5 Path 1
    DemoStep 0.5 Path 3

# 2
Vehicle
    DemoStep 0.5 Path 2
    DemoStep 0.5 Path 3

# 3
Vehicle
    DemoStep 0.5 Path 1
    DemoStep 0.5 Path 3

# 4
Vehicle
    DemoStep 1.0 Path 2
    DemoStep 0.5 Path 3
```

All_Vehicles

Purpose

Identifies this Demo Script as being for all vehicles on all paths in the transport system using one set of DemoSteps. Using this directive, there is no need to provide movement steps for the individual vehicles. This command must precede the set of DemoSteps steps in the script.

To make sure that the script executes properly, observe the following:

- The `Paths` global directive is not used.
- The `Demo_Arrival_Method` global directive must be Single.

NOTE: When the NCHost TCP Interface Utility assigns vehicles to the vehicle directives, the vehicle with an ID of 1 (as assigned by the HLC) may not be assigned to the first vehicle commanded by the Demo Script.

Format

```
All_Vehicles
```

Example

The following example moves all vehicles until stopped.

```
Velocity 0.5
Acceleration 1.0
Direction Forward
Demo_Arrival_Method Single

All_Vehicles
DemoStep 2.0000 Path 1
DemoStep 1.5000 Path 2
```

DemoStep

Purpose

This basic DemoStep command defines a position on the path the vehicle is on that the vehicle moves to, measured from the start of the path in meters. This basic command is used when vehicle motion only occurs on one path.

This command can be extended by adding any of the command extensions to it as shown in *Combine Command Extensions* on page 136. See the extensions that are listed in the See Also section of this command.

Format

```
DemoStep Pos_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

See Also

Path on page 214
Delay on page 215
DelayOnce on page 216
Acceleration on page 217
Velocity on page 218
Direction on page 219
Payload on page 220

Example

The following example moves the vehicle to a position 3.0 meters from the start of the path.

```
Vehicle
    DemoStep 3.0
```

Path

Purpose

This Command Extension for the DemoStep command defines the specific path where the DemoStep command moves the vehicle. This extension is used when vehicle motion occurs on multiple paths.

NOTE: If the script uses only one path, this command is not required.

Format

```
DemoStep Pos_Value Path Path_ID
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

Path_ID – The ID of the path where the vehicle destination is located.

Example

The following example moves the vehicle to a position 3.0 meters from the start of path 2. If the vehicle does not start on path 2, it moves there first.

```
Vehicle
    DemoStep 3.0 Path 2
```

Delay

Purpose

This Command Extension for the DemoStep command defines a delay to the start of the execution of the DemoStep command. This extension can be used with either the **Destination** step command or the **Null** step command.

Format

```
DemoStep Pos_Value Delay D_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

D_Value – The delay to the start of movement, in milliseconds.

See Also

Null on page 224

Examples

The following example moves the vehicle to a position 3.0 meters from the start of the path. Actual movement is delayed for 500 milliseconds (0.5 seconds) after the DemoStep command has been processed.

```
Vehicle
    DemoStep 3.0 Delay 500
```

The following example does not move the vehicle, but creates a delay of 8000 milliseconds (8 seconds) after the DemoStep command has been processed.

```
Vehicle
    DemoStep Null Delay 8000
```

DelayOnce

Purpose

This Command Extension for the DemoStep command defines a delay to the start of the execution of the DemoStep command only on the first time the command is executed. This extension can be used with either the **Destination** step command or the **Null** step command.

Format

```
DemoStep Pos_Value DelayOnce D_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

D_Value – The delay to the start of movement, in milliseconds.

See Also

Null on page 224

Examples

The following example moves the vehicle to a position 3.0 meters from the start of the path, but delays the actual movement 500 milliseconds (0.5 seconds) after the DemoStep command has been processed. The delay only occurs the first time the DemoStep is executed. Each time the DemoStep is executed afterwards, the vehicle moves without any delay.

```
Vehicle
    DemoStep 3.0 DelayOnce 500
```

The following example does not move the vehicle, but delays execution of the DemoStep by 8000 milliseconds (8 seconds) after the DemoStep has been processed. The delay only occurs the first time the DemoStep is executed. Each time the DemoStep is executed afterwards, the vehicle moves without any delay.

```
Vehicle
    DemoStep Null DelayOnce 8000
```

Acceleration

Purpose

This Command Extension for the DemoStep command defines the acceleration that the DemoStep command applies to the vehicle as it moves to the specified position. This extension overrides the Acceleration Global Directive for the vehicle for this step only.

Format

```
DemoStep Pos_Value Acceleration Acc_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

Acc_Value – The maximum acceleration for the vehicle, in meters/second^2. This value can be greater than or equal to the value specified by the **Acceleration** Global Directive. But, it must be less than or equal to the value specified as the Acceleration Limit in the Node Controller Configuration File.

See Also

Acceleration on page 203

Example

The following example moves the vehicle to a position 3.0 meters from the start of the path with an acceleration of 5.0 m/s^2.

```
Vehicle
    DemoStep 3.0 Acceleration 5.0
```

Velocity

Purpose

This Command Extension for the DemoStep command defines the velocity that the DemoStep command applies to the vehicle as it moves to the specified position. This extension overrides the Velocity Global Directive for the vehicle for this step only.

Format

```
DemoStep Pos_Value Velocity Vel_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

Vel_Value – The maximum velocity for the vehicle, in meters/second. This value can be greater than or equal to the value specified by the **Velocity** Global Directive. But, it must be less than or equal to the value specified as the Velocity Limit in the Node Controller Configuration File.

See Also

Velocity on page 202

Example

The following example moves the vehicle to a position 3.0 meters from the start of the path with a velocity of 2.5 m/s.

```
Vehicle
    DemoStep 3.0 Velocity 2.5
```

Direction

Purpose

This Command Extension for the DemoStep command defines the direction that the DemoStep command uses as it moves the vehicle to the specified position. This extension overrides the Direction Global Directive for the vehicle for this step only.

Format

```
DemoStep Pos_Value Direction Dir_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

Dir_Value – The direction of movement for the vehicle.

Bi-Directional – Vehicles move Downstream (Forward) or Upstream (Backward) to go to a destination. When the destination is on the same path the vehicle is on, the vehicle moves in either direction to get to the destination in the shortest distance. When the destination is on a path other than the path the vehicle is on, the forward direction takes preference for a transport system that is a closed-loop.

Forward – Vehicles move in the Downstream direction only (towards the end of the path they are on).

Backward – Vehicles move in the Upstream direction only (towards the beginning of the path they are on).

See Also

Direction on page 204

Example

The following example moves the vehicle to a position 3.0 meters from the start of the path. The vehicle moves in the direction that provides the shortest route to that position.

```
Vehicle
    DemoStep 3.0 Direction Bi-Directional
```

Payload

Purpose

This Command Extension for the DemoStep command defines the payload setting that the DemoStep command uses as it moves the vehicle to the specified position. This extension overrides the Payload or PID_Set_Index Global Directive for the vehicle for this step only.

NOTE: This command is deprecated. It only supports the Unloaded and Loaded PID sets. For full support of all PID sets, use the *PID_Set_Index* command.

Format

```
DemoStep Pos_Value Payload PID_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

PID_Value – The load condition for movement of the vehicle.

Unloaded – Vehicle moves using the unloaded Control Loop parameters (PID Set 0).

Loaded – Vehicle moves using the loaded Control Loop parameters (PID Set 1).

See Also

Payload on page 205
PID_Set_Index on page 206
PID_Set_Index on page 221

Example

The following example moves the vehicle to a position 3.0 meters from the start of the path. The vehicle moves using the loaded PID values (PID Set 1).

```
Vehicle
    DemoStep 3.0 Payload Loaded
```

PID_Set_Index

Purpose

This Command Extension for the DemoStep command defines the PID Index setting that the DemoStep command uses as it moves the vehicle to the specified position. The PID sets must be defined in the Node Controller Configuration File. This extension overrides the Payload or PID_Set_Index Global Directive for the vehicle for this step only.

Format

```
DemoStep Pos_Value PID_Set_Index Index_Value
```

Arguments

Pos_Value – The vehicle destination from the start of the path, in meters.

Index_Value – The PID Set (0…15) to be used for vehicle motion (Set 0 is predefined as unloaded, Set 1 is predefined as loaded, and Set 15 is predefined for startup).

If PID Set 15 is defined, it is used during startup. If it is not defined, PID Set 0 is scaled by 25% and used for startup.

See Also

Payload on page 205
PID_Set_Index on page 206
Payload on page 220

Example

The following example moves the vehicle to a position 3.0 meters from the start of the path. The vehicle moves using the loaded PID values (PID Set 1).

```
Vehicle
    DemoStep 3.0 PID_Set_Index 1
```

Rendezvous Point

Purpose

This Command Extension for the DemoStep command defines a step where specific vehicles arrive at specified positions independently and then leave those positions synchronously. This extension is only used when the Global Demo_Arrival _Method command is specified as **Single**, which allows independent movement, so that specific vehicles can be synchronized at a particular step.

Format

```
DemoStep Rendezvous Point RP_ID
```

Arguments

RP_ID – A unique numeric ID for the rendezvous. If multiple Rendezvous Points are used, each must have a unique ID.

See Also

Demo_Arrival_Method on page 207

Example

The following example moves the first vehicle to the 1.0 meter position and the second vehicle to the 2.0 meter position on the loop. Both vehicles move to their second position as soon as they have stopped at their first position. Once either vehicle arrives at its second position it waits until the other vehicle arrives at its second position, then both vehicles move to their third position.

```
Demo_Arrival_Method Single
Vehicle
    DemoStep 1.0
    DemoStep 4.0
    DemoStep Rendezvous Point 1
    DemoStep 3.0

Vehicle
    DemoStep 2.0
    DemoStep 1.0
    DemoStep Rendezvous Point 1
    DemoStep 2.5
```

Freeze

Purpose

This Command Extension for the DemoStep command defines a step where movement of a specific vehicle is stopped. This extension is typically used to make sure that all vehicles are stopped at the same step when independent movement is being allowed. It can also be used to make sure that all vehicles are stopped at the same step before control is passed to another script when vehicle motion is independent.

Format

```
DemoStep Freeze
```

See Also

Chain on page 229

Example

The following example moves the first vehicle to a position 1.0 meter from the start of the path and the second vehicle to a position 2.0 meters from the start. Both vehicles then move to their second position. The first vehicle stops at its second position and the script stops running once the second vehicle stops at its second position.

```
Demo_Arrival_Method Single
Vehicle
    DemoStep 1.0
    DemoStep 3.0
    DemoStep Freeze

Vehicle
    DemoStep 2.0
    DemoStep 6.0
    DemoStep Stop
```

Null

Purpose

This Command Extension for the DemoStep command defines a step where no action occurs. When there are multiple vehicles with different numbers of steps, this extension is used as a place holder to keep the total number of steps for each vehicle the same. This extension can be extended by adding the Delay command extension to it as shown in the Examples.

NOTE: When defining or executing independent scripts (Arrival Method Single), the command set for a vehicle cannot be set to all Null values (DemoStep Null) to keep the vehicle from moving while other vehicles execute their commands independently. Setting the command set for a vehicle to all Null values causes the NCHost Utility to stop. If there is a vehicle on the system that is not to be moved, use the Freeze extension to keep that vehicle from moving; not the Null command.

Format

```
DemoStep Null
```

See Also

Delay on page 215
DelayOnce on page 216

Examples

In the first example, if the script is being executed dependently, the first vehicle moves to 1 meter from the start of the path, as the second vehicle moves to 4 meters from the start of the same path. Once both vehicles have completed the step, the first vehicle moves to the 3 meter position as the second vehicle moves to the 5 meter position. Once both vehicles have completed the step, the first vehicle does not move and the second vehicle moves to the 6 meter position. Once the second vehicle completes its movement, both vehicles move to their first position and the movements repeat until the script is stopped.

If the script is being executed independently, the first vehicle moves to 1 meter from the start of the path, as the second vehicle moves to 4 meters from the start of the same path. Once each vehicle completes the first step, it moves to its second position. Once each vehicle completes the second step, the first vehicle skips the third step and starts its first step again while the second vehicle moves to the 6 meter position. Once the second vehicle completes the third step, it starts its first step again. These movements repeat until the script is stopped.

```
Vehicle
    DemoStep 1.0
    DemoStep 3.0
    DemoStep Null

Vehicle
    DemoStep 4.0
```

```
    DemoStep 5.0
    DemoStep 6.0
```

In the second example, if the script is being executed dependently, the motion is the same as for the first example being executed dependently. Except in this example, if the second vehicle finishes its third step in less than 5 seconds it must wait until the 5 second timer for the first vehicle ends before both vehicles move to their first position with the movements repeating until the script is stopped.

If the script is being executed independently, the motion is the same as for the first example being executed independently except that the first vehicle must now wait 5 seconds before executing the first step again. While the first vehicle is waiting, the second vehicle continues to execute its steps. These movements repeat until the script is stopped.

```
Vehicle
    DemoStep 1.0
    DemoStep 3.0
    DemoStep Null Delay 5000

Vehicle
    DemoStep 4.0
    DemoStep 5.0
    DemoStep 6.0
```

Stop

Purpose

This Command Extension for the DemoStep command defines a step where execution of the script is stopped.

In a loop track with multiple vehicles, using independent movement, all vehicles on the same path as the stopped vehicle queue next to the vehicle that executes the DemoStep Stop. In any track, using dependent movement, all vehicles stop at the completion of the step where the DemoStep Stop was executed.

NOTE: When using DemoStep Stop, use either DemoStep Null or DemoStep Freeze for the equivalent step for all other vehicles.

This extension can be extended by adding any of the Stop command extensions to it as shown in *Combine Command Extensions* on page 136. See the extensions that are listed in the See Also section of this command.

Format

```
DemoStep Stop
```

See Also

Iterations on page 228
Chain on page 229
SyncPath on page 230

Examples

In the first example, if the script is being executed independently, the first vehicle moves to a position 1.0 meter from the start of the path and the second vehicle to a position 2.0 meters from the start. Both vehicles then move to their second position. These movements repeat for the first vehicle until the script stops running once the second vehicle stops at its second position.

```
Vehicle
    DemoStep 1.0
    DemoStep 3.0
    DemoStep Null

Vehicle
    DemoStep 2.0
    DemoStep 6.0
    DemoStep Stop
```

In the second example, if the script is being executed independently, the first vehicle moves to a position 1.0 meter from the start of the path and the second vehicle to a position 2.0 meters from the start. Both vehicles then move to their second position. The first vehicle stops at its second position and the script stops running once the second vehicle stops at its second position.

```
Vehicle
    DemoStep 1.0
    DemoStep 3.0
    DemoStep Freeze

Vehicle
    DemoStep 2.0
    DemoStep 6.0
    DemoStep Stop
```

Iterations

Purpose

This Command Extension for the DemoStep command defines the number of cycles the specified vehicle runs before the script is stopped, which stops all vehicle motion. If there are additional Command Extensions after the **Iterations** extension, they are executed once the script stops.

Format

```
DemoStep Stop Iterations I_Value
```

Arguments

I_Value – The number of times the script is run before it stops.

See Also

Stop on page 226

Examples

The following example moves the first vehicle to a position 7.0 meters, then 2.0 meters on the loop. The second vehicle moves to 2.0 meters, then 7.0 meters on the loop. This sequence repeats 30 times then the script stops.

```
Vehicle
    DemoStep 7.0
    DemoStep 2.0
    DemoStep Stop Iterations 30

Vehicle
    DemoStep 2.0
    DemoStep 7.0
    DemoStep Null
```

The following example provides the same motion as the first example. The sequence repeats 10 times then the script *single_arrival_script.txt* is loaded and started (see *Chain* on page 229).

```
Vehicle
    DemoStep 7.0
    DemoStep 2.0
    DemoStep Stop Iterations 10 Chain single_arrival_script.txt

Vehicle
    DemoStep 2.0
    DemoStep 7.0
    DemoStep Null
```

Chain

Purpose

This Command Extension for the DemoStep command defines the step in the script where all vehicle motion is stopped and another Demo Script is started. This extension can be used to create a complex script that requires items such as different arrival methods at different times. The number of vehicles in each Demo Script must be the same.

NOTE: All Demo Scripts being chained must be in the same folder.

Format

```
DemoStep Stop Chain Script
```

Arguments

Script – The name of the Demo Script file to run. The Demo Script must be in the same folder as the Demo Script that calls it.

See Also

Stop on page 226

Example

The following example moves the first vehicle to a position 7.0 meters, then 2.0 meters on the loop. The second vehicle moves to 2.0 meters, then 7.0 meters on the loop. Once the first vehicle arrives at the 2.0 meter position the Demo Script *single_arrival_script.txt* is loaded and run, which replaces the existing script. A **Null** command is used for the third step of the second vehicle to maintain the same number of steps for both vehicles.

```
Vehicle
    DemoStep 7.0
    DemoStep 2.0
    DemoStep Stop Chain single_arrival_script.txt

Vehicle
    DemoStep 2.0
    DemoStep 7.0
    DemoStep Null
```

SyncPath

Purpose

This Command Extension for the DemoStep command defines the step in the script where a defined number of vehicles must accumulate on a specified path before all vehicle motion is stopped and another script is started. This extension is used to make sure that vehicles are in specific positions before control is passed to another script and can be used to create a complex script that requires items such as different arrival methods at different times.

Format

```
DemoStep Stop SyncPath Path_ID Count V_Value Chain Script
```

Arguments

Path_ID – The ID of the path for vehicle accumulation.

V_Value – The total number of vehicles that must be on the path before the specified script begins. This must be the total number of vehicles on the transport system.

Script – The name of the script file to run.

See Also

Chain on page 229

Example

The following example moves the first vehicle to the 2 meter position on path 1 and the second vehicle to the 7 meter position on path 2. The first vehicle then moves to the 7 meter position on path 1 while the second vehicle moves to the 2 meter position on path 1. Once both vehicles have completed the second step, all motion is stopped and the script *single_arrival_script.txt* starts.

```
Vehicle
    DemoStep 2.0 Path 1
    DemoStep 7.0 Path 1
    DemoStep Stop SyncPath 1 Count 2 Chain single_arrival_script.txt

Vehicle
    DemoStep 7.0 Path 2
    DemoStep 2.0 Path 1
    DemoStep Null
```

Troubleshooting 6

Overview

This chapter describes the common difficulties that may be encountered when using the NCHost TCP Interface Utility. Included in this chapter are:

- NCHost TCP Interface Utility Troubleshooting.
- Demo Script Troubleshooting.

For assistance, contact Rockwell Automation TechConnect℠ (rockwellautomation.custhelp.com).

NCHost TCP Interface Utility Troubleshooting

Table 6-1 briefly identifies common NCHost TCP Interface Utility faults, and general solutions.

Table 6-1: NCHost TCP Interface Utility Troubleshooting

Symptom	Problem	Solution
Vehicle does not move.	The motor is unable to sense the magnet array on the vehicle.	Send a Reset command to the path, manually move the vehicle, then send a Startup command. If the vehicle still does not move, then manually move the vehicle to another path, and restart the path where the vehicle is located. If the vehicle moves, troubleshoot the motor where the vehicle was originally.
	There is a physical obstruction.	Make sure that there are no obstructions to vehicle movement.
	The vehicle is binding on the guideway.	Make sure that the vehicle can move freely.
	The vehicle movement command is rejected.	Review the HLC status message for the command to determine the reason for rejection. Correct and resend the command.
The NCHost TCP Interface Utility does not connect to the HLC even though it connected previously.	There is another device that is connected to the HLC.	When the HLC is available again, reconnect using the IP address previously used. See *Running the NCHost TCP Interface Utility* on page 39.
	The IP address of the HLC has been changed.	Connect to the HLC to verify the address. See the *Node Controller Interface User Manual*, MMI-UM001.
	The network is not connected.	Verify all network connections.

Table 6-1: NCHost TCP Interface Utility Troubleshooting (Continued)

Symptom	Problem	Solution
The vehicles remain in the **Init** state (Initializing), and the transport system does not work.	There could be a number of problems with the node controller, such as incorrectly structured XML in the Node Controller Configuration File.	Review the Diagnostic Messages and look for an error that identifies the problem. See *Diagnostic Messages* on page 61.
		Review the View Log page of the Node Controller Web Interface and look for an error that identifies the problem. See the *Node Controller Interface User Manual*, MMI-UM001.

Demo Script Troubleshooting

Table 6-2 briefly identifies common Demo Script faults, and general solutions.

Table 6-2: Demo Script Troubleshooting

Symptom	Problem	Solution
Vehicles do not move when a script is started.	There are an inconsistent number of DemoSteps for each Vehicle section in the script.	Verify that all Vehicle sections contain the same number of DemoSteps.
	The number of vehicles on the specified paths and the number of Vehicle sections in the script are not the same.	Add, or remove, Vehicle sections in the script to match the number of vehicles on the specified paths.
Vehicles do not move to the expected positions.	The DemoSteps are configured incorrectly.	Verify that the correct path is specified.
		Verify that the distance to the position is measured from the start of the path.

Diagnostic Output Troubleshooting

The messages that are displayed in the Diagnostic Output pane are related to actions that are performed through the NCHost TCP Interface Utility and the responses from the transport system. There are three types of diagnostic messages: Debug (Table 6-3), Error (Table 6-4), and Info (Table 6-5).

Table 6-3: Diagnostic Output Messages, Debug

Message	User Response
Command: "*command*" status "*cmd_status*" node ID: *n*	None - Displays the status of the vehicle command sent to the HLC for node *n*.
Command: "*command*" status "*cmd_status*" path ID: *n*	None - Displays the status of the path command sent to the HLC for path *n*.
Command: "*command*" status "*cmd_status*" vehicle ID: *n*	None - Displays the status of the vehicle command sent to the HLC for vehicle *n*.
DELAY *t* ms for vehicle: *n*	None, Demo script is executing a Delay step for vehicle *n*.
Move vehicle *n* on path *n* to position *command_data*	None - Displays the move vehicle command sent to the HLC for vehicle *n*.
Remote server closed connection	Determine the reason for the connection to the HLC going down.
Vehicle *n* Follow Order follows vehicle *m*, *command_data*	None - Displays the status of the platooning command for vehicle *n*.
Vehicle *n*: DELAY is complete	None, Demo script Delay step for vehicle *n* is complete.
Vehicle: *n*, cmd: "*command*" status *cmd_status*, *cmd_data*	None - Displays the status of the move vehicle command for vehicle *n*.

Table 6-4: Diagnostic Output Messages, Error

Message	User Response
Command: '*command*' status: *cmd_status*	Determine the reason for the command failure and correct the command.
Command: "*command*" status "*cmd_status*" for Node ID: *n* and Path ID: *p*	None - Displays the status of the path command sent to the HLC for node *n* and path *p*.
Engineering Tools password incorrect	Enter a valid password.
Invalid DemoStep, bad Delay value at line *n*	Enter a valid delay value.

Table 6-4: Diagnostic Output Messages, Error (Continued)

Message	User Response
Kd must be a real number equal to 0.0 or larger	Enter a valid Kd when testing control loop parameters.
Kff must be a real number in the range 0.0 to 100.0	Enter a valid Kff when testing control loop parameters.
Ki must be a real number equal to 0.0 or larger	Enter a valid Ki when testing control loop parameters.
Kp must be a real number equal to 0.0 or larger	Enter a valid Kp when testing control loop parameters.
Node ID must be all numeric and between 1 and 65535	Enter a valid node ID.
Notification mask for NC ID *n* must be a hex value between 0000 and FFFF	Enter a valid mask in hex.
Output mask for NC ID *n* must be a hex value between 0000 and FFFF	Enter valid data in hex.
Path ID for Control Loop Parameters Request must be between 0 and 65535	Enter a valid path ID when testing control loop parameters.
Path ID must be numeric and between 0 and 65535	Enter a valid path ID.
Please specify an IP address	Enter the HLC IP address in the Host Interface section, then select **Connect**.
Please use the Browse button to specify a track file	Specify a Track file before launching the Track Graphics window.
Script error: Found no vehicles on Paths directive paths at line *n*	Make sure that the system is running and all paths are started.
Script error: Invalid directive at line *n*	Make sure that a valid demo script file is loaded.
Script error: Too few vehicles in script	Make sure that the number of vehicles in the Demo script is the same as the number of vehicles on the system.
Script error: The number of vehicle directives exceeds the number of vehicles on the specified paths	Make sure that the number of vehicles in the Demo script is the same as the number of vehicles on the system.
The command at line *n* is invalid	Correct the error in the track file.

Table 6-4: Diagnostic Output Messages, Error (Continued)

Message	User Response
Unable to connect to the HLC server (*IP_addr*)	• Verify that the HLC IP address is correct. • Make sure the HLC and the computer running NCHost are connected to the network. • Ping the HLC from the computer to verify communication.
Vehicle ID must be all numeric and between 0 and 65535	Enter a valid vehicle ID.
Vehicle Mass must be a real number equal to 0.0 or larger	Enter a valid mass when testing control loop parameters.

Table 6-5: Diagnostic Output Messages, Info

Message	User Response
Command: '*command*' status: *cmd_status*	None - Displays the status of the command sent to the HLC.
Connecting to *IP_addr ...*	None, NCHost is connecting to the HLC at *IP_addr*.
Control Node *node_ID* action *node_action*	None, indicates that the specified node has been either enabled or disabled.
Diagnostic log file closed	None, indicates that the Diagnostic log file is closed and data collection has stopped.
Diagnostic log file opened	None, indicates that the Diagnostic log file is opened to start data collection.
Disconnected from *IP_addr ...*	None, NCHost is disconnected from the HLC at *IP_addr*.
Engineering Tools Locked	Enter a valid password.
Load Track: *path\file*	None, identifies the track file that is selected to load into the Track Graphics window.
Socket connected to the HLC server (*IP_addr*)	None, NCHost is connected to the HLC at *IP_addr*.
The demo script has been started by the user	None, identifies the Demo Script has been started.
The demo script has been stopped by the user	None, identifies the Demo Script has been stopped.
Track Load of *path\file status*	None, Displays the status of the track file loading into the Track Graphics window.

Contact Rockwell Automation Technical Support

Before contacting Technical Support, have the following information ready.

1. The name, email address, and telephone number of the person to contact.
2. The facility address where the system is located and the project name.
3. The date, time, and a detailed description of the anomaly, including:
   - The command the equipment was executing when the anomaly occurred.
   - The effect on system performance (for example, stalled vehicles, overheating).
   - How the system was operating before the anomaly occurred and for how long.
   - Any recent changes to the system (physical reconfiguration, speed/acceleration changes, configuration file changes).
   - Any corrective actions performed (for example, system reset, replaced parts, loaded software, power cycle). Describe the results of those actions.
   - Any special system environmental conditions (for example, vacuum, high heat, high humidity).
   - Any potential non-MagneMotion causes of the issue (for example, power outage, mechanical interference, host failure).
   - Reproducibility of the anomaly.
4. The equipment type, part number, serial number, and location in the system (path, motor id).
5. Include the following files from the time of the anomaly.
   - Node controller and high-level controller log files.
   - Node Controller Configuration Files.
   - Host controller command logs (if available).
   - Any product-related faults and error messages observed through the system host, NCHost, the web interface, and so on.
6. Include any other information that can assist our specialist.
7. Contact Rockwell Automation TechConnect℠ (rockwellautomation.custhelp.com).

Appendix

Overview

The following appendices are included to provide the user with additional information that is related to the NCHost TCP Interface Utility and MagneMotion® transport systems.

Included in this appendix are:

- Communications protocol.
- File maintenance.
- Additional documentation.
- Transport system configuration limits.

Communications Protocol

Communications Format

The high-level controller (HLC) supports communication with the NCHost TCP utility application that runs on a general-purpose computer via TCP/IP over Ethernet as defined in the OSI model. Each layer of network communication within the standard OSI (Open Systems Interconnect, ISO/IEC 7498-1) model is described here. Ethernet TCP/IP runs directly over the TCP and UDP layer 4 transport layers in the OSI model. The MagneMotion® implementation of TCP/IP only uses the TCP transport layer.

Ethernet TCP/IP

The standard OSI model for layers 1…4 are outlined here. The HLC application runs on top of the TCP transport layer. The standard 10/100/1000 Base-TX, half-duplex, or full-duplex twisted-pair Ethernet is used for all network communications.

Physical Layer

The electrical interface that is supported for communication between the HLC software that runs on the node controller hardware and the host controller is twisted-pair Ethernet. This interface is based on the IEEE 802.3 and IEEE 802.3u Ethernet and Fast Ethernet communication standards.

The interface uses IEEE 802.3 and IEEE 802.3u standard signal levels for 10/100/1000 Base-TX. A standard Category 5 or better cable is used to connect the HLC directly to the host controller when a hub, switch, or router is not used for the connection.

Data Link Layer

The data link layer is standard 802.3/802.3u Ethernet packet framing. This framing uses 6 byte destination and source MAC addresses, 2 byte protocol type, 1500 byte MTU (maximum transmission unit) of payload data, and 2 byte FCS (frame check sequence). 10 Mbps and 100 Mbps signaling rates in full-duplex and half-duplex modes are supported. It is expected that any standard Ethernet device that uses a twisted-pair physical layer will have no problems interoperating with the HLC computer at the physical and data link layers.

Network Layer

The network layer communications for the HLC is Internet Protocol Version 4 (IPv4). For diagnostic purposes during initial setup and optional network layer health status checks, the HLC supports accepting Internet Control Message Protocol (ICMP) echo requests and supports responding to echo requests with ICMP echo responses. The HLC supports the additional ICMP messages that are required to support IPv4 fragmentation and redirection of network layer packets to the next hop IP router if customer provided networking equipment

requires the HLC to honor such IP fragmentation or packet routing redirection. It is expected that any standard modern IPv4 implementation will have no problems interoperating with the HLC computer.

The HLC does not support dynamic routing protocols and does not act as a router in the customer network. The HLC drops any packets that it receives via some other peer on the network that uses the HLC computer as its next hop IPv4 router.

The HLC supports configuration with a static IPv4 address, network mask, and gateway address of the next hop IP router.

The HLC supports IPv4 Address Resolution Protocol (ARP) to support the discovery of peer Ethernet data link layer MAC addresses.

Transport Layer

The HLC uses a Transmission Control Protocol (TCP) stream over the IPv4 network layer to support an end to end connection.

The HLC is the server and the host controller is the client for TCP endpoint communication roles. The HLC listens for incoming TCP connections from the host controller (general-purpose computer) on the following ports:

- 799 for the Host Control connection.
- 800 for the Host Status connections.
- 801 for the System Monitoring connection.
- 8000 for the MagneMotion Virtual Scope connection.

When the HLC receives an invalid message on an established TCP connection (due to a framing error), the HLC software drops the connection.

The HLC throttles incoming TCP connections to ports 799 and 800 if a problem occurs in the host controller software that causes an excessive number of connections to be established. In extreme cases, the HLC is forced to drop the connection. In the event a connection is dropped, the HLC waits one second before allowing a new connection.

Host Control Connection

The HLC supports one Host Control TCP connection from the host controller at a time. If a new TCP connection is made to TCP port 799 when the HLC already has an established connection, the established connection is dropped and the new connection is used. Even though TCP/IP supports a more or less arbitrary number of peers at the transport layer, for safety reasons, the HLC supports only one Host Control TCP connection at a time.

Host Status Connection

The HLC supports up to four Host Status TCP connections at a time. If a new TCP connection is made to port 800 when the HLC already has four established connections, the oldest connection is dropped and the new connection is used.

NOTE: The Host Status connections can be used to monitor the status of the transport system using the various status request commands. Any other commands are dropped.

System Monitoring Connection

The HLC supports one System Monitoring connection at a time. If a new TCP connection is made to port 801 when the HLC already has an established connection, the established connection is dropped and the new connection is used.

NOTE: The System Monitoring connection can be used to monitor the status of transport system components using the various monitoring request commands. Any other commands are dropped.

Virtual Scope Connection

The HLC supports one Virtual Scope TCP connection at a time. If a new TCP connection is made to port 8000 when the HLC already has an established connection, the established connection is dropped and the new connection is used.

File Maintenance

Backup Files

Making regular backups of all files that have been changed is recommended. Keep copies of all original and backup files at a remote location for safety.

Creating Backup Files

Backup files are not created automatically. It is the responsibility of the user to create backups of all files by copying them to a secure location.

Restoring from Backup Files

Damaged files can be restored by copying the backup files into the appropriate locations.

Additional Documentation

Release Notes

The release notes that are supplied with MagneMotion software include special instructions, identification of software versions, identification of new features and enhancements, and a list of known issues. Reading this file is recommended before using the software.

Upgrade Procedure

The upgrade procedures that are supplied with MagneMotion software provide instructions for upgrading from one version of MagneMotion software to another. They also include the procedures for file and driver upgrades that are associated with the software.

Transport System Limits

Table A-1: MagneMotion Transport System Limits

	Path	Node Controller	System (HLC)
Motors	20/RS-422 30/Ethernet	–†	3,840
Node Controllers	–	–	96
Nodes	2	–†	320
Paths	–	–†	160
Stations	–	–	2048
Vehicles	50/RS-422* 300/Ethernet	384	5,120‡

* When using RS-422 communication with the motors, 50 vehicles maximum per path when all vehicles on the path are commanded forward (downstream).
45 vehicles maximum per path when all vehicles on the path are commanded backwards (upstream).

† When using RS-422 communication with the motors, limited by the number of RS-422 connections on the node controller (NC LITE up to 4 connections, NC-12 up to 12 connections, NC-S up to 8 connections).
When using Ethernet communication with the motors, limited by the node controller configuration and processor loading (NC LITE up to 5 nodes, NC-12 up to 16 nodes, NC-E up to 36 nodes, NC-S up to 16 nodes), see the *Node Controller Hardware User Manual*, MMI-UM013.

‡ 6,000 vehicles maximum when using HLC Control Groups.

*Table A-2: MagneMotion Transport System Motion Limits**

	Acceleration	Velocity	Thrust
MagneMover® LITE†	2.0 m/s^2 [0.2 *g*]	2.0 m/s [4.5 mph]	10.0 N/cycle‡
QuickStick® 100	9.8 m/s^2 [1.0 *g*]	2.5 m/s [5.6 mph]	16.3 N/cycle§
QuickStick® HT with QSMC	60.0 m/s^2 [6.1 *g*]	3.5 m/s [7.8 mph]	182.0 N/cycle**
QuickStick® HT with QSHT 5700 Inverter	60.0 m/s^2 [6.1 *g*]	5.0 m/s [11.2 mph]	182.0 N/cycle**

* The limits that are shown are at the typical payloads (contact your Motion Solution Consultant or TechConnect℠ (rockwellautomation.custhelp.com) for payload guidance). Use of a smaller payload may permit higher limits. Use of a larger payload may lower the limits.

† The limits that are shown are for standard MagneMover LITE glide pucks. Using other pucks, or custom vehicles, may permit different limits.

‡ Thrust at 25% duty cycle, nominal Vehicle Gap is 1 mm for G3 and 1.5 mm for G4.2 magnet arrays.

§ Thrust at 4.0 A stator current with a nominal Vehicle Gap of 3 mm with a standard magnet array.

** Thrust at 10.9 A stator current with a nominal Vehicle Gap of 12 mm with a high flux magnet array.

This page intentionally left blank.

Glossary

Block: See *Motor Block*.

Bogie: A structure underneath a vehicle to which a magnet array is attached. The structure is then attached to the vehicle. For vehicles that travel over curves, the attachment is through a bearing that allows independent rotation.

Brick-wall Headway: The space that is maintained between vehicles to make sure that a trailing vehicle is able to stop safely if the lead vehicle stops suddenly ("hits a brick wall").

Byte: An octet of data (8 bits).

Clearance Distance: The distance from a node where the trailing edge of a vehicle is considered cleared from a node.

Component: The main parts that form a MagneMotion® *Transport System*. Also called system components, these include the *High-Level Controller*, *Motors, Nodes, Node Controllers, Paths*, and *Vehicles*.

Configuration File: See *Node Controller Configuration File*.

Configurator: The application that is used to define and edit the basic operating parameters of the transport system that are stored in the *Node Controller Configuration File*.

Controller: A device that monitors and controls the operating conditions of the equipment being monitored. In a MagneMotion transport system, the types of controllers include the *High-Level Controller*, *Node Controller*, and *Host Controller*.

Couple: The joining of a vehicle into a platoon, where one vehicle follows another vehicle at a defined distance.

Cycle Length: The distance between the centerlines of two like poles on the magnet array.

Demo Script: A text file that is used with the NCHost TCP Interface Utility for test or demonstration purposes to move vehicles on the transport system.

Design Specifications: The unique parameters for a specific MagneMotion transport system.

Downstream: The end of a motor or path as defined by the logical forward direction. Vehicles typically enter the motor or path on the *Upstream* end.

Downstream Gap: The physical distance from the end of the stator in one motor to the beginning of the stator in the next motor downstream on the same path. This distance includes the *Motor Gap*.

Drive: The remote QSMC motor controller or the QSHT 5700 inverter for a QSHT motor.

Emergency Off: A user-supplied device that disconnects AC power to the transport system.

Emergency Stop: A user-supplied circuit with a locking button that anyone can press to stop motion in the transport system. It can be wired through the digital I/O on some *Node Controllers*.

EMO: See *Emergency Off*.

Entry Gate: The position on a path associated with a node where the leading edge of a vehicle is considered cleared from the node. See *Node Clearance Distance*.

Entry Path: A path whose downstream end is a member of a node. A vehicle that is moving downstream enters a node on an entry path.

E-stop: See *Emergency Stop*.

Ethernet Chain: Ethernet chains allow devices to be connected in series with standard Ethernet cable, without the need for additional network switches. A daisy chain device has two embedded Ethernet ports that function as an Ethernet switch and an interface to the local device. This embedded switch allows information to flow to the device, or flow through the ports to other devices in the chain.

Exit Path: A path whose upstream end is a member of a node. A vehicle that is moving downstream exits a node on an exit path.

Following Vehicle: A vehicle following another vehicle in a platoon. This vehicle can be following either the lead vehicle or another following vehicle.

Forward Direction: The default direction of motion, from *Upstream* to *Downstream*, on a MagneMotion transport system.

Glide Puck: A preconfigured vehicle for use on MagneMover® LITE transport systems that uses low friction skids to slide on the integral rails.

Global Directives: The *Demo Script* commands that define the general operating characteristics for all vehicles specified. See also *Vehicle Directives*.

Ground: The reference point in an electrical circuit from which voltages are measured. This point is typically a common return path for electric current. See also *PE*.

Guideway: A component of the *Track System* that consists of rails or other devices in contact with the *Vehicle*, either through wheels or low friction runners on the vehicle. The guideway maintains the proper relationship between the vehicles and the motors. In the MagneMover LITE transport system, the guideway is the integral rails that are mounted on the motors.

Hall Effect Sensor: A transducer that varies its output in response to changes in a magnetic field. Hall Effect Sensors (HES) are used by MagneMotion LSMs for vehicle positioning and speed detection.

Headway: The space that is maintained before a vehicle to make sure that the vehicle is able to stop safely. See *Brick-wall Headway*.

High-Level Controller: The application in a node controller that communicates with the host controller. Only one node controller per HLC Control Group runs the high-level controller application. In a transport system with only one node controller, it runs both the node controller and high-level controller applications.

HLC: See *High-Level Controller*.

HLC Control Group: The portion of a multi-HLC LSM transport system under control of a specific HLC.

Host Application: The software on the host controller that provides monitoring and control of the transport system.

Host Control Session: A session between a host controller application (such as the NCHost TCP Interface Utility) and an HLC that allows control of all aspects of transport system operation. The Host Control Session also allows active monitoring of transport system status.

Host Controller: The user-supplied controller that provides control and sequencing for the operation of the transport system.

Host Status Session: A session between a host controller application (such as the NCHost TCP Interface Utility) and an HLC that only provides active monitoring of transport system status.

ICT: See *Independent Cart Technology*.

ID: The software labels used to identify various components of the transport system to make sure proper execution of commands involving vehicle position, vehicle destination, and transport system configuration. ID types include vehicle and path.

Independent Cart Technology: A programmable intelligent conveyor system that uses linear synchronous motors for moving multiple independently controlled vehicles.

Interlock: A user-supplied circuit that is used to stop motion in the transport system. It is wired through the digital I/O on the *Node Controller*.

Inverter: The hardware that converts DC from the propulsion power bus to pulse-width modulated AC to energize the coils in a *Motor Block* and contains the *Master* and the *Inverter* for QSHT motors. See *Drive*.

Keep-out Area: A unidirectional area of a *Path*. A vehicle that is moving in the specified direction of the area is not allowed to enter the area unless it has permission from the motors to either move past or stop within the area. Once a vehicle enters the keep-out area in the specified direction, all other vehicles that are moving in the same direction must wait to enter the area until that vehicle exits.

Lead Vehicle: The vehicle at the front of a platoon. This vehicle determines the destination, acceleration, and velocity of the platoon.

Logic Power: The power that is used for the controllers and signals. See also, *Propulsion Power*.

LSB: Least Significant Byte.

LSM: Linear Synchronous Motor. See *MagneMover LITE* and *QuickStick*.

MagneMover LITE: A MagneMotion linear synchronous motor with integrated guideways and vehicles that enable quick, efficient conveyance of small loads.

MagneMover LITE System: A group of specific components that contribute to a *Transport System*. These components include *MagneMover LITE* motors, *Node Controllers*, *Pucks*, and other parts available from MagneMotion.

Magnet Array: The magnets that are attached to the *Vehicle*. It is the motor secondary, moved by the primary in the motor.

Master (also Master Controller): The supervisory controller for each motor, it communicates with the *Slaves* to direct *Motor Block* operation and read motor sensors. It communicates vehicle positions and other information to the *Node Controller* and upstream and downstream motors. It is internal to the motor assembly on MagneMover LITE and QuickStick® motors. For QuickStick HT motors the master is in the *Drive*.

MM LITE™: See *MagneMover LITE*.

MML™: See *MagneMover LITE*.

Motion Controller: The user-supplied controller for direct control of vehicles through the LSM Synchronization option. It can reside on the host controller.

Motor: See *LSM*.

Motor Block: A discrete motor primary section (coil or set of coils) in a motor that can be energized independently. This section can contain only one vehicle during transport system operation.

Motor Controller: The hardware that converts DC from the propulsion power bus to AC to energize the coils in a *Motor Block* and contains the *Master* and the *Inverter* for QuickStick HT motors. See *Drive*.

Motor Gap: The physical distance between two motors that are mounted end to end. This gap excludes the distance from the end of the stator to the end of the motor housing.

MSB: Most Significant Byte.

NC: See *Node Controller*.

Node: A junction that is defined as the beginning, end, or intersection of *Paths*. The different node types define their use: Simple, Relay, Terminus, Merge, Diverge, and so on.

Node Clearance Distance: The position on a path associated with a node where the trailing edge of a vehicle is considered cleared from the node. See *Entry Gate*.

Node Controller: The hardware and the application running on that hardware that coordinates vehicle motions along a path or paths of motors. The node controller is responsible for the motors on all paths that begin at nodes that the node controller is responsible for.

There can be multiple node controllers in a transport system, each responsible for a subset of the nodes within the transport system.

Node Controller Configuration File: The XML file unique to the transport system that defines the basic operating parameters of the transport system. A copy of the Node Controller Configuration File is uploaded to each node controller in the transport system.

NRTL/ATL: Nationally Recognized Test Lab/Accredited Test Lab.

OSHA recognizes NRTL organizations in accordance with 29 CFR 1910.7 to test and certify equipment or materials (products).

Accreditation bodies evaluate ATL organizations to ISO/IEC 17025 for testing and calibration laboratories.

OSSD: Output Signal Switching Device. The interface of a sensor (such as a light curtain) designed for reliably signaling a safety-related event. OSSD signals are outputs from the protective device to a safety relay.

Path: A designation for one or more motors placed end to end, which defines a linear route for vehicle travel. A path begins at the *Upstream* end of the first motor in the series and ends at the *Downstream* end of the last motor in the series. All paths must begin at a *Node* and the beginning of a path is always the zero position for determining positions along that path.

PE: Protective Earth. A conductor that is provided for safety purposes (for example, against the risk of electric shock) and which also provides a conductive path to earth. See also, *Ground*.

Pitch: The distance between a point on one coil and the corresponding point on an adjacent coil in the same motor or an adjacent motor. Or, the distance between a point on one magnet and the corresponding point on an adjacent equivalent magnet in the same array.

Platooning: A group of vehicles that are moving together and following a lead vehicle. This group of vehicles is allowed to maintain a distance between each other while in motion that is less than the *Brick-wall Headway*.

Position: A specific location on a *Path*, which is measured from the beginning of that path, which is used as a vehicle destination. Position zero on any path is defined as the leading edge of the first LSM in the path.

A vehicle at a specific position has its midpoint over that location on the path.

Power Supply: The equipment that is used to convert facility AC power to the correct voltages for the transport system.

Propulsion Power: The power that is used for vehicle motion. See also, *Logic Power*.

Protected Area: The area around a node that is defined by the entry gates and clearance distances. This area is used to make sure that vehicles do not collide with other vehicles in the node or with the mechanism that is related to the node.

Puck: A preconfigured vehicle for use on MagneMover LITE transport systems. The magnet array is mounted to the puck and interacts with the motors, which move each vehicle independently. See *Glide Puck* and *Wheeled Puck*. See also, *Vehicle*.

QS: See *QuickStick*.

QuickStick: A MagneMotion linear synchronous motor that enables quick, efficient conveyance of large loads on user-designed guideways and vehicles. QuickStick (QS) motors move loads up to 100 kg [220 lb] per vehicle. QuickStick High Thrust (QSHT) motors move loads up to 4,500 kg [9,900 lb] per vehicle.

QuickStick System: A group of specific components that contribute to a *Transport System*. These components include *QuickStick* motors, *Node Controllers*, *Motor Controllers* (QSHT only), *Magnet Arrays*, and other parts available from MagneMotion.

Sensor Map: A snapshot of the signal state of vehicle magnet array sensors that are collected from all blocks of a motor.

Signal: Each motor contains sensors that detect the magnetic field from the magnet array. When the signal from the sensors is higher than a threshold, the signal bit for the associated sensor is set high, otherwise it is set low.

Single Vehicle Area: A unidirectional area of a *Path*. Only one vehicle that is moving in the specified direction of the area is allowed to enter the area at a time. Other vehicles on the path that are moving in the same direction as the initial vehicle in the SVA must wait to enter this area until the previous vehicle exits. This queueing allows one vehicle to move backward and forward along a portion of a path without interfering with any other vehicles.

Slave (also Slave Controller): The subordinate controllers for the motor, they communicate with the *Master* and operate the *Inverters* and position-sense hardware. They are internal to the motor assembly on MagneMover LITE and QuickStick motors. For QuickStick HT motors the slaves are in the *Drive*.

Station: A specific location on a *Path*, which is measured from the beginning of that path, and identified with a unique ID, used as a vehicle destination.

Stator: The stationary part of the motor over which the magnet array moves.

Switch: The mechanical guide for positioning a vehicle through guideway sections that merge or diverge.

SYNC IT™: Provides direct control by a motion controller of up to three sync-zones (requires sync-enabled motors) where the host controller generates the vehicle motion profile.

Sync Zone: An area where vehicle motion can be synchronized with other systems through direct control of the motor by the host controller.

System Component: See *Component*.

Tandem Vehicle: A vehicle that uses dual *Bogies* to provide enough thrust to carry larger loads.

Track System: The components that physically support and move vehicles. For a QuickStick transport system, the track includes a *Guideway*, one or more *QuickStick* motors, mounting hardware, and a stand system. For a MagneMover LITE transport system, the track includes the *MagneMover LITE* motors and stands.

Transport System: The components that collectively move user material. These components include the *Motors*, external *Motor Controllers* (QSHT only), *Track System*, *Node Controllers*, *Vehicles,* cables, and hardware.

Uncouple: Remove a vehicle from a platoon.

Upstream: The beginning of a motor or path as defined by the logical forward direction. The upstream ends of all paths are connected to node controllers. Vehicles typically exit the motor or path on the *Downstream* end.

V-Brace: The mechanical fixture that is used to align and secure MagneMover LITE guide rail and motor sections.

Vehicle: The independently controlled moving element in a MagneMotion transport system. The vehicle consists of a platform that carries the payload and a passive magnet array to provide the necessary propulsion and position sensing. All vehicles on paths in the transport system that are connected through nodes must be the same length.

The transport system constantly monitors and controls vehicle position and velocity for the entire time the vehicle is on the transport system. All vehicles are assigned a unique ID at startup and retain that ID until the transport system is restarted or the vehicle is removed or deleted.

Vehicle Directives: The *Demo Script* commands that define the individual motion characteristics for a specific vehicle. See also *Global Directives*.

Vehicle Gap: The distance between the bottom of the magnet array that is attached to a vehicle and the top surface of a motor.

Vehicle ID Master Database: The HLC database for the assignment and tracking of Vehicle IDs in the transport system. When using *HLC Control Groups*, the Master HLC maintains this database.

Vehicle ID Slave Database: The Slave HLC database for tracking of Vehicle IDs in the HLC Control Group managed by that Slave HLC and assigned by the Master HLC. This database is only used when using *HLC Control Groups* to subdivide a transport system.

Vehicle Master: The motor controlling the vehicle.

Vehicle Signal: A motor software flag for each vehicle that is used to indicate if the vehicle is detected on the transport system.

Vehicle Spacing: The distance between two vehicles on the same path.

Wheeled Puck: A preconfigured vehicle for use on MagneMover LITE transport systems that uses low friction wheels to ride on the integral rails.

Zero Point: The position on the *Upstream* end of a *Path* that denotes the first part on which a *Vehicle* travels.

Index

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

This page intentionally left blank.

Notes:

Rockwell Automation Support

Use the following resources to access support information.

Technical Support Center	Knowledgebase Articles, How-to Videos, FAQs, Chat, User Forums, Product Notification Updates, and the Direct Dial Code for your product.	rok.auto/support
Literature Library	Installation Instructions, Manuals, Brochures, and Technical Data.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	rok.auto/pcdc
Product Catalog and Configurator	Additional product information including CAD drawing files, 3D models, photos, and more.	rok.auto/productdrawings

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the How Are We Doing? form at https://rok.auto/docfeedback.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

139 Barnum Road, Devens, MA 01434, USA
+1 978-757-9100
Publication MMI-UM010G-EN-P - August 2021 DIR 990000562
Supersedes MMI-UM010F-EN-P