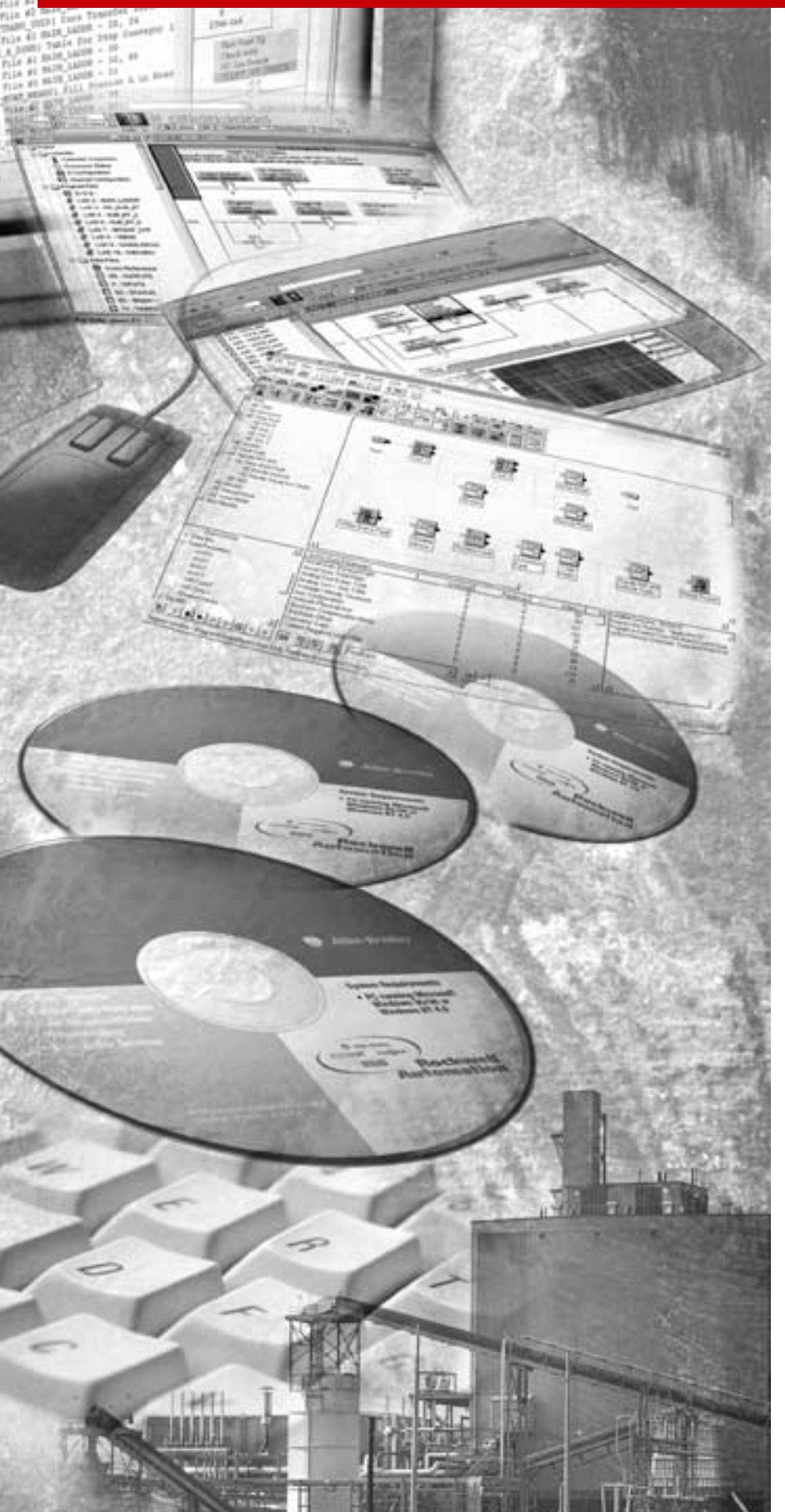


Allen-Bradley

ControlLogix™ Motion Module

(Cat.No. 1756-M02AE, -M08SE)

Setup and Configuration Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Throughout this manual we use notes to make you aware of safety considerations:

WARNING

Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss

Attention statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

IMPORTANT

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

European Communities (EC) Directive Compliance

If this product has the CE mark it is approved for installation within the European Union and EEA regions. It has been designed and tested to meet the following directives.

EMC Directive

This product is tested to meet the Council Directive 89/336/EC Electromagnetic Compatibility (EMC) by applying the following standards, in whole or in part, documented in a technical construction file:

- EN 50081-2 EMC — Generic Emission Standard, Part 2 — Industrial Environment
- EN 50082-2 EMC — Generic Immunity Standard, Part 2 — Industrial Environment

This product is intended for use in an industrial environment.

Low Voltage Directive

This product is tested to meet Council Directive 73/23/EEC Low Voltage, by applying the safety requirements of EN 61131-2 Programmable Controllers, Part 2 - Equipment Requirements and Tests. For specific information required by EN 61131-2, see the appropriate sections in this publication, as well as the Allen-Bradley publication Industrial Automation Wiring and Grounding Guidelines For Noise Immunity, publication 1770-4.1.

This equipment is classified as open equipment and must be mounted in an enclosure during operation to provide safety protection.

Preface **The Manual**

Chapter 1 **The ControlLogix Motion Control System**

Chapter 2 **Getting Started**

Using This Manual

This preface describes how to use this manual.

Who Should Use This Manual

To use this manual, you should be able to program and operate the Allen-Bradley Logix5550™ controller to efficiently use your motion control modules.

If you need more information about programming and operating the Logix5550 controller, refer to the Logix5550 Controller User Manual, publication number 1756-6.5.12.

The Purpose of This Manual

This manual describes how to configure and troubleshoot your ControlLogix motion module.

The following table shows the contents of each section in this manual:

Section	Contains
Chapter 1 The ControlLogix Motion Control System	Information about the ControlLogix motion control system.
Chapter 2 Controller Properties	Explains how to create and edit your controller.
Chapter 3 Adding and Configuring Your 1756-M02AE Motion Module	How to add and configure your 1756-M02AE motion module using the RSLogix™ 5000 programming software.
Chapter 4 Adding and Configuring Your 1756-M08SE Motion Module	How to add and configure your 1756-M08SE motion module using the RSLogix™ 5000 programming software
Chapter 5 The Motion Group	How to create and edit your Motion Group.
Chapter 6 Adding and Configuring Motion Axes	How to name and configure a motion axis using the RSLogix^{tm} 5000 programming software.
Chapter 7 Configuring a 1394C-SJT05/10/22-D Digital Servo Drive	How to add and configure a 1394C Digital Servo Drive.
Chapter 8 Configuring an Ultra 3000 Drive	How to add and configure an Ultra 3000 drive

Chapter 9 The Motion Instructions	Information about the 32 motion instructions provided in the RSLogix 5000 programming software.
Chapter 10 Troubleshooting	Information about troubleshooting your ControlLogix motion control system.
Appendix A Specifications and Performance	Specifications and performance guidelines for the motion module.
Appendix B Loop and Interconnect Diagrams	Loop diagrams and wiring diagrams for your ControlLogix motion control system.
Appendix C The Motion Control Structures	An explanation of the motion control structures.
Appendix D The Motion Attributes	Information about the motion attributes.
Appendix E Instruction Timing	Information about types of timing for motion instructions.
Appendix F Fault Handling	Information about motion control faults.

Related Documentation

The following table lists related ControlLogix documentation:

Publication Number	Publication	Description
1756-IN047	Analog Encoder (AE) Servo Module Installation Instructions	Provides instructions for installing, wiring, and troubleshooting your 1756-M02AE servo module.
1756-UM001	Logix5550 Controller User Manual	Provides information for using your Logix5550 controller and its components.
1756-RM0003	Logix5550 Controller Instruction Set Reference Manual	Provides descriptions of all the instructions supported by the RSLogix 5000 programming software.
1756-RM007	Motion Instructions Reference Set Manual	Provides descriptions of all of the motion instructions used in the RSLogix 5000 software.
1756-IN572	8 Axis SERCOS interface Module Installation Instructions	Provides instructions for installing, wiring, and troubleshooting your 1756-M08SE SERCOS module.

Publication Number	Publication	Description
1394C-5.20	1394 SERCOS Interface Multi Axis Motion Control System	Information on installation, wiring, and setup for the 1394C-SJTxx-D
1394-IN024	1394 SERCOS Integration Manual	Information on integrating the 1394 drive with the 1756-M08SE
2098-IN003	Ultra3000 Hardware Installation Manual	Information on Ultra3000 installation
2098-IN001	Ultra5000 Hardware Installation Manual	Information on Ultra5000 installation
2098-IN005	Ultra3000 SERCOS Integration Manual	Information on integrating the Ultra3000 with the 1756-M08SE

For more information on the documentation, refer to the Allen-Bradley Publication Index, publication number SD499.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 sales/support offices, 512 authorized distributors, and 260 authorized systems integrators located throughout the United States. In addition, Rockwell Automation representatives are located in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, please review the information in this manual. If the problem persists, call your local Rockwell Automation representative.

The Rockwell Automation Technical Support number is:

1-603-443-5419

On the Web

For information about Allen-Bradley, visit the following World Wide Web site:

http://www.ab.com/

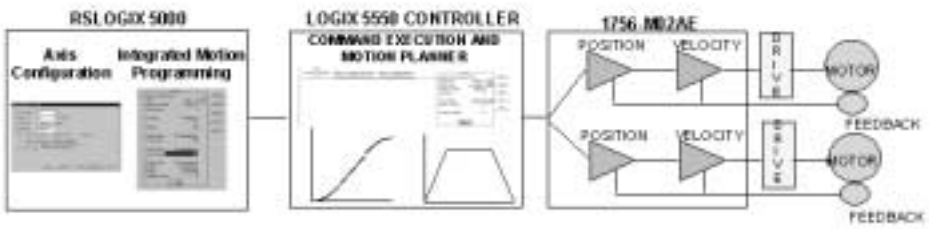
The ControlLogix Motion Control System

This chapter describes the ControlLogix motion control system.

ControlLogix Motion Control

The ControlLogix controller, 1756-M02AE servo module, 1756-M08SE SERCOS interface module, and RSLogix5000 programming software provide integrated motion control support.

- The ControlLogix controller contains a high-speed motion task, which executes ladder motion commands and generates position and velocity profile information. The controller sends this profile information to one or more 1756-M02AE servo modules. You can use several Logix controllers in each chassis. Each controller can control up to 16 1756-M02AE servo modules.
- The 1756-M02AE servo module connects to a servo drive and closes a high-speed position and velocity loop. Each Logix controller can support up to 16 1756-M02AE servo modules. Each 1756-M02AE module can control up to two axes.
- The 1756-M08SE SERCOS interface module serves as the interface between one ControlLogix processor and 1 to 8 axes operating in either position or velocity mode. The module has a programmable ring Cycle Period of 1ms or 2ms depending on the number of axes and a ring Data Rate of 4 Mbaud.
- RSLogix5000 programming software provides complete axis configuration and motion programming support.



Components of the ControlLogix Motion System

The ControlLogix Controller

The ControlLogix controller is the main component in the ControlLogix system. It supports sequential and motion functions, and it performs all of the motion command execution and motion trajectory planner functions. You can use one or more ControlLogix controllers in each chassis, and each controller can control up to 16 motion modules.

The ControlLogix controller provides the following motion support:

- Thirty motion instructions
- A high-speed motion task, which manages motion functions and generates move profiles
- The ability to control up to 16 Analog/Encoder servo modules for a total of 32 axes

The Analog/Encoder Servo Module (1756-MO2AE)

The Analog/Encoder servo module provides an analog/quadrature encoder servo drive interface. The servo module receives configuration and move information from the ControlLogix controller and manages motor position and velocity.

The servo module supports:

- Connection capability for up to two drives
- ±10V analog outputs
- Quadrature encoder inputs
- Home limit switch inputs
- Drive fault inputs
- Drive enable outputs
- 5V or 24V position registration inputs
- 200 µs position and velocity loop updates

The 8 Axis SERCOS interface Module (1756-MO8SE)

The 8 Axis SERCOS interface module (1756-M08SE) serves as a link between the ControlLogix platform and intelligent drives. The communication link between the module and the drive(s) is via IEC 1491 **SE**rial **R**eal-time **CO**mmunication **S**ystem (SERCOS) using fiber optic medium.

The SERCOS interface module supports:

- reliable high speed data transmission
- excellent noise immunity
- elimination of interconnect wiring
- ASA messages converted to SERCOS formatted messages

RSLogix5000 Programming Software

The RSLogix5000 programming software provides complete programming and commissioning support for the ControlLogix system. RSLogix5000 is the only programming software needed to fully configure and program ControlLogix motion control systems.

RSLogix5000 software provides the following motion support:

- Wizards for servo axis configuration including drive hookup diagnostics and auto tuning
- Ladder-based application programming including support for 30 motion commands

Developing a Motion Control Application Program

This section provides an introduction to concepts used in developing application programs for motion control. These concepts include:

- Application program development
- The MOTION_INSTRUCTION tag
- Motion status and configuration parameters
- Modifying motion configuration parameters
- Handling motion faults

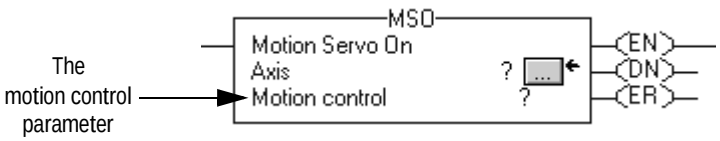
Application Program Development

Developing a motion control application program involves the following:

Task	Description
Select the master coordinated system time	Sets one controller as the master controller. Once you complete this step, you can synchronize all the motion modules and ControlLogix controllers in your chassis
Name and Configure an axis	Adds an axis to your application program
Develop a motion application program	Create a program for your motion control application
Add a motion module	Adds a motion module to your application program
Assign additional servo modules and axes	Adds additional modules and axes to your application program
Run hookup diagnostics and auto tuning	Completes hookup diagnostics and auto tuning for each axis

The MOTION_INSTRUCTION Tag

The controller uses the MOTION_INSTRUCTION tag (structure) to store status information during the execution of motion instructions. Every motion instruction has a motion control parameter that requires a MOTION_INSTRUCTION tag to store status information.



WARNING

Tags used for the motion control parameter of instructions should only be used once. Re-use of the motion control parameter in other instructions can cause unintended operation of the control variables.

For more information about the MOTION_INSTRUCTION tag, refer to Appendix C - *The Motion Control Structures*.

Motion Status and Configuration Parameters

You can read motion status and configuration parameters in your ladder logic program using two methods.

Method	Example	For more information
Directly accessing the AXIS and MOTION_GROUP structures	• Axis faults • Motion status • Servo status	Refer to Appendix C - *The Motion Control Structures*
Using the GSV instruction	• Actual position • Command position • Actual velocity	Refer to the *Input/Output Instructions* chapter of the Logix Controller Instruction Set Reference Manual, publication 1756-RM003B

Modifying Motion Configuration Parameters

In your ladder logic program, you can modify motion configuration parameters using the SSV instruction. For example, you can change position loop gain, velocity loop gain, and current limits within your program.

For more information about the SSV instruction, refer to the Logix Controller Instruction Set Reference Manual, publication 1756-RM003.

Handling Motion Faults

Two types of motion faults exist.

Type	Description	Example
Motion Instruction Errors	• Do not impact controller operation • Should be corrected to optimize execution time and ensure program accuracy	A Motion Axis Move (MAM) instruction with a parameter out of range
Minor/ Major Faults	• Caused by a problem with the servo loop • Can shutdown the controller if you do not correct the fault condition	The application exceeded the PositionErrorTolerance value

For more information about handling faults, see *Handling Controller Faults* in the Logix Controller User Manual, publication 1756-UM001 and *Appendix F Fault Handling* in this manual.

Getting Started

Introduction

Before you can begin programming or configuring your controller, you must create a project file in which to store it.

Create a Project

1. From the Type pull-down menu, choose the controller type that you wish to use for this project.

2. Enter the name you wish to use for the controller.

The same name is used for the project file with the .acd extension.

3. Enter a description of the controller.

4. Choose the appropriate chassis type in which the controller resides.

   Note: This field is disabled if you have chosen a FlexLogix controller type. FlexLogix does not have a physical chassis, and therefore you do not need to select a chassis type.

5. Enter the slot number for the controller.

In ControlLogix, controllers occupy a numbered slot in the chassis and can be placed in any slot. It is also possible to place multiple controllers in the same chassis.

For FlexLogix, since there is no physical chassis, this field is disabled, and slot number 0 is displayed.

6. Verify the appropriate revision information for your controller. This field defaults to the latest revision for the given controller type.

7. Enter the directory in which you want to store the project file.

The directory defaults to the one you configured in the Workstation Options dialog. If you want to use a different directory, type its path or click on the Browse button to find the directory.

The project file is created in this directory with the same name as the controller with a .ACD file extension. For example, if your controller name is Oven1, the project file name becomes Oven1.ACD.

8. Click on OK to create the project.

Once the project file is created, you can see the Controller Organizer, which shows everything in the controller. The default configuration contains a continuous task, called Main Task. The Main Task contains a program called Main Program. The Main Program contains a routine called Main Routine, which is configured as the main routinedef_main_routine@gloss.hlp.

In addition, if you have chosen a FlexLogix controller, 2 FlexBus adapters are created in slots 3 and 4 under the I/O Configuration folder. These 2 folders contain all local I/O for FlexLogix, other than the 2 local slots for communication. The first folder contains all I/O configured on the local Flex rail housing the Flex controller; the second folder contains all I/O configured for the local non-controller rail.

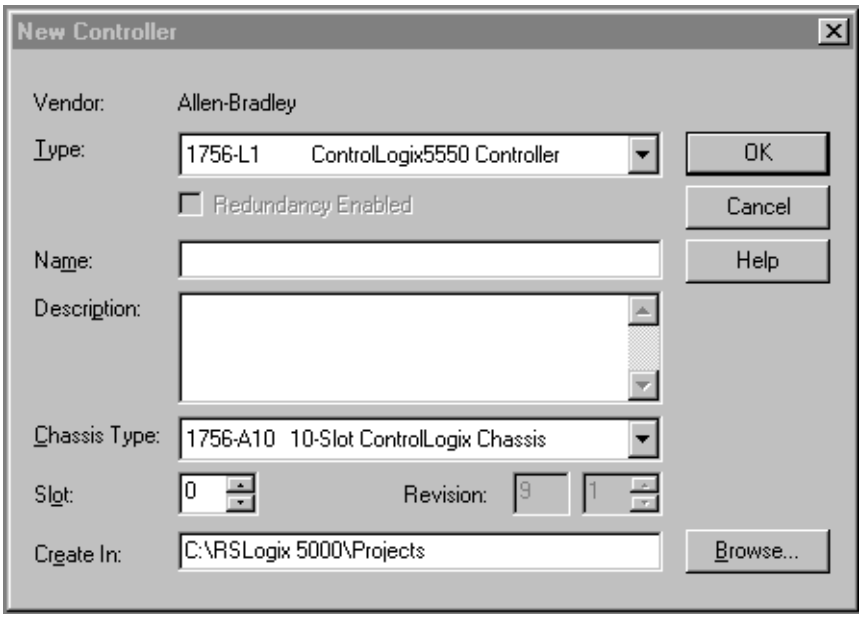
> **Note:** You cannot delete, copy, cut, paste, or drag and drop the FlexBus adapters. Once you create a FlexLogix controller, the adapters appear under the I/O Configuration folder and cannot be altered unless you delete the FlexLogix controller.

Accessing the New Controller dialog

From the File menu, choose New to access the New Controller dialog.

New Controller Dialog

Create a new controller (i.e., project) from this dialog.



Vendor

Displays the name of the controller's manufacturer.

Type

Select the controller type from the pull-down menu, shown here by catalog number, platform, and processor. The default controller is the 1756-L1 ControlLogix 5550 controller.

Choose from:

- 1756-L1 ControlLogix 5550 controller
- 1756-L53/A ControlLogix 5553 controller
- 1756-L55/A ControlLogix 5555 controller
- 1769-L20 CompactLogix 5320 controller
- 1769-L30 CompactLogix 5330 controller
- 1789-L60/A SoftLogix 5860 controller
- 1794-L33/A FlexLogix 5433 controller
- 1794-L34/A FlexLogix 5434 controller
- PowerFlex700S DriveLogix 5720 controller

Redundancy Enabled

Check this box if you wish to enable redundancy for this controller. Note that this option is disabled if you have chosen a controller that does not support redundancy.

> **Note:** Redundancy is not supported in this release of RSLogix 5000.

Name

Enter the name you wish to use for the new controller. This name is also used for the project file, with a .acd extension.

Description

Enter a description of the controller.

Chassis Type

Select the appropriate chassis type from the pull-down menu, shown here by catalog number. The software uses this information to determine the number of slots in the chassis. Depending on the controller type you chose, the available options in this menu vary:

For this platform:	Choose from these chassis types:
ControlLogix	1756-A4, 4-slot ControlLogix chassis 1756-A7, 7-slot ControlLogix chassis 1756-A10, 10-slot ControlLogix chassis 1756-A13, 13-slot ControlLogix chassis 1756-A17, 17-slot ControlLogix chassis
CompactLogix	Not applicable
SoftLogix	1789-17, 17-slot SoftLogix virtual chassis
FlexLogix	Not applicable
DriveLogix	Not applicable

Slot Number

Choose the slot number where the controller resides on the backplane.

For ControlLogix controllers, the default value is 0. If the slot number exceeds the chassis size, an error message appears, prompting you to enter a number within the valid range.

For SoftLogix controllers, the default value is 1.

For CompactLogix, FlexLogix, and DriveLogix controllers, the value in this field is always 0 and cannot be edited.

Revision

Enter the controller revision.

For this release of RSLogix 5000, this field is not editable. The revision defaults to the latest revision for the given controller type.

Create In:

Enter the directory in which you want the project file to be created. The file name is the same as the controller name, with a .acd extension.

Browse

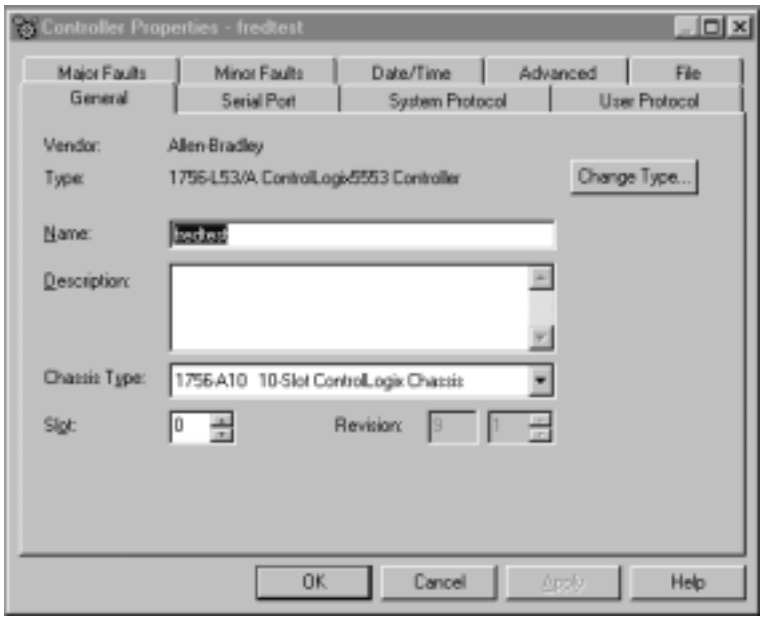
Click on this button to bring up the Choose Directory dialog from which you can browse for the appropriate directory.

Editing Controller Properties

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. This section describes the fields on each of the dialogs for the Controller Properties.

General Tab

The General tab displays the controller name and description, as well as the physical properties of the controller.



Vendor

Displays the name of the controller's manufacturer.

Type

The catalog number and description of the controller. When online, this field includes the catalog number of the memory card (if any).

Name

The name of the controller. When you create a project, this is the same as the name of the project file. When you change the name of the controller, however, the name of the project file does not change. If you want to keep the two the same, then you must rename the file using Windows Explorer or a similar file management tool.

IMPORTANT This name must be IEC_1131 compliant. If you enter an invalid character in this field, or if the name you enter exceeds 40 characters, the software ignores the character.

You cannot change the name when online.

Description

Enter a description for the controller here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software ignores any extra characters.

Chassis Type

Select a supported chassis type from the pull-down list. Each entry in the list consists of the catalog number of the chassis, as well as a brief description.

The chassis type cannot be changed when online.

Slot

Enter the chassis slot number in which the controller resides. The spin button contains values that range from 0 to 1 less than the chassis size (e.g., if you have a 4-slot chassis, the spin button spins from 0 to 3). If you enter a slot number that is out of this range, you receive an error message when you go to apply your changes.

The slot number cannot be changed when online.

Revision

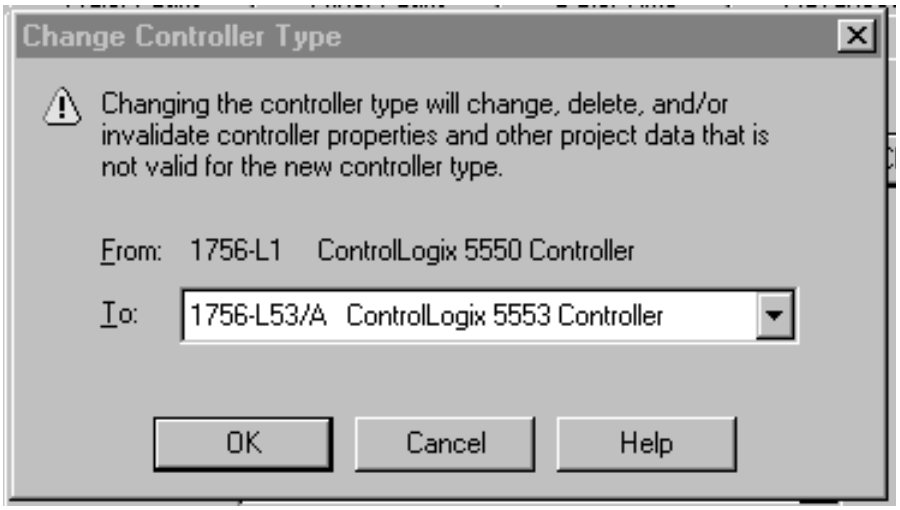
Displays the major and minor revision of the controller. The minor revision is available only when you are online.

Change Type

Click on this button to access the Change Processor Type dialog. This dialog lets you change your controller to another controller within the same platform.

Change Controller Type Dialog Box Overview

Use this dialog to change your controller to another controller within the same platform (e.g. changing from a 1756-L1 ControlLogix 5550 Controller to a 1756-L53/A ControlLogix 5553 controller).



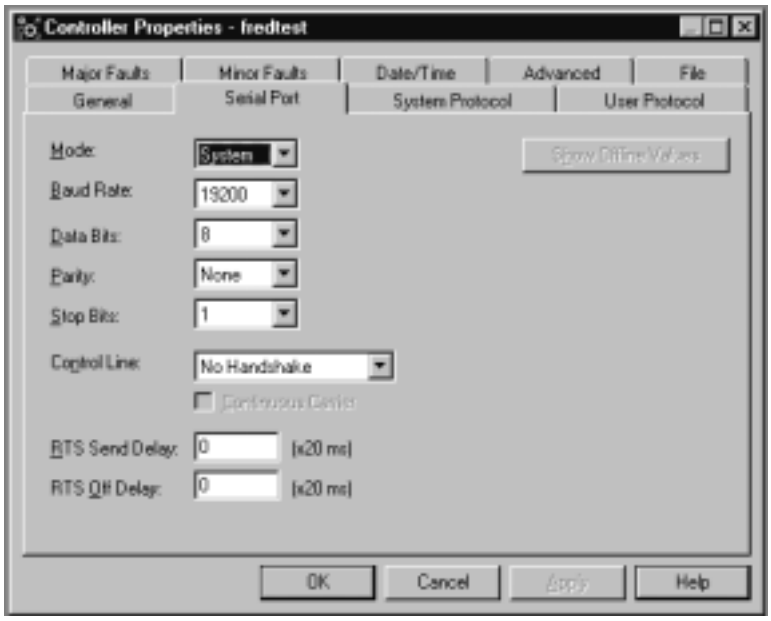
IMPORTANT At this time, RSLogix 5000 does not support changing to a controller from within another platform (e.g., changing from a ControlLogix controller type to a FlexLogix controller type).

Select a processor to change to

Choose the controller you wish to change to from the pull-down menu. The list of available controllers includes all controller types within the same platform as the current processor, with the exception of the current processor itself.

Serial Port Tab

The Serial Port tab allows you to view and configure the controller's serial port.



Mode

The type of protocol you want to use. Choose from System or User (default).

Baud Rate

The baud rate assigned to the serial port on the Logix5550. Choose from 110, 300, 600, 1200, 2400, 4800, 9600, and 19200 (default).

Data Bits

The actual number of bits of data per character. Choose from 7 (ASCII only) or 8 (default).

Parity

The parity for the link. Choose from Even, Odd (ASCII only) or No Parity (default).

Stop Bits

The actual number of stop bits per character. Choose from 2 (ASCII only) or 1 (default).

Control Line

Choose the type of handshaking you wish to use during communications. The choices available to you vary, depending on the protocol you have selected:

For this mode:	And this protocol:	Choose from:
User	ASCII	No Handshake (default) Full Duplex Half Duplex
System	Point-to-Point	No Handshake (default) Full Duplex
	Slave	No Handshake (default) Half Duplex
	Master	No Handshake (default) Full Duplex Half Duplex

Continuous Carrier

Check this box if you wish to use Half Duplex communication with continuous carrier. This checkbox is disabled if you have chosen something other than Half Duplex communication, or if you have chosen Master as your protocol. By default, this option is unchecked when enabled.

RTS Send Delay

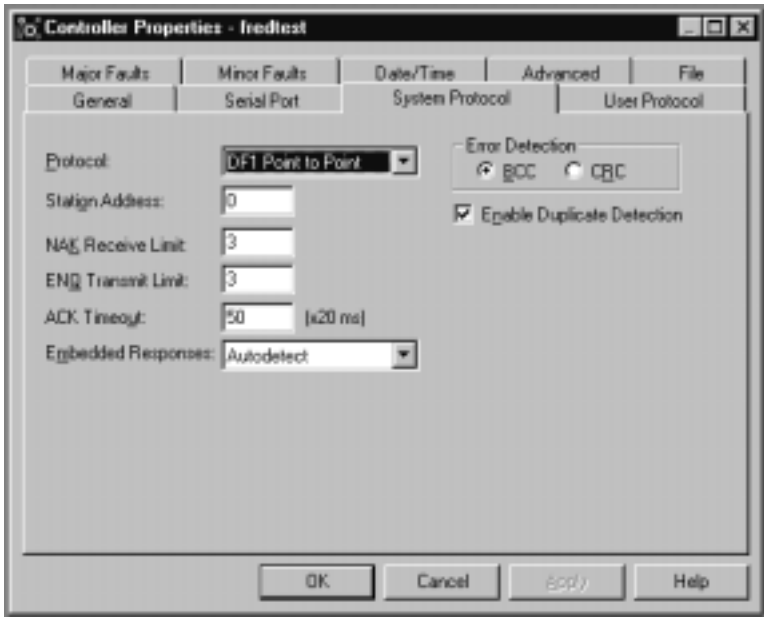
Enter the time (in ms) to delay transmitting the first character of a message after turning on the RTS line. The default value is 0.

RTS Off Delay

Enter the time (in ms) to delay turning off the RTS line after the last character has been transmitted. The default value is 0.

System Protocol Tab

The System Protocol tab allows you to configure the controller's serial port for DF1 Point to Point, DF1 Master, DF1 Slave or DH485. The parameters present on this tab are dependent upon the protocol you select.



IMPORTANT Note: If you wish to configure your system for ASCII, click on the User Protocol tab.

The parameters present on this tab are dependent upon the protocol you select.

Common Parameters

Protocol

Choose the protocol from the pull-down menu. Choose from DF1 Point to Point (default), DF1 Slave, DF1 Master or DH485.

Station Address

Enter the current station link address of the com port to which the DF1 object is now associated. Valid values are from 0 to 254; the default value is 0.

Error Detection

Click on one of the radio buttons to specify the error detection scheme used for all messages.

- BCC - the processor sends and accepts messages that end with a BCC byte.
- CRC - the processor sends and accepts messages with a 2-byte CRC.

Enable Duplicate Detection

Check this box to enable duplicate message detection, which causes the object to ignore all duplicate messages. This option is disabled by default.

ACK Timeout

Enter the time the object waits for an acknowledgment to a message transmission. Valid values are from 0 to 65535, in 20 ms increments; the default value is 50 ms.

DF1 Point to Point Parameters

ENQ Transmit Limit

Enter the number of inquiries you want the processor to send after an ACK Timeout. Valid values are from 0 to 255; the default value is 3.

NAK Receive Limit

Enter the number of NAKs the processor can receive in response to a message before stopping the transmission. Valid values are from 0 to 255; the default value is 3.

Embedded Responses

This parameter sets the flag that enables the embedded response functionality. Your options are:

- Autodetect – embedded responses are initiated only after one is received.
- Enabled – embedded responses are enabled unconditionally.

DF1 Slave Parameters

Transmit Retries

Enter the number of attempted transmits without getting an acknowledgment before a message is deemed undeliverable. Valid values are from 0 to 255; the default value is 3.

Slave Poll Timeout

Enter the amount of time that the master waits for an acknowledgment to a message sent to the slave.

EOT Suppression

Check this box if you want to suppress "End of Text" transmissions at the end of a slave message.

DF1 Master Parameters

Transmit Retries

Enter the number of attempted transmits without getting an acknowledgment before a message is deemed undeliverable. Valid values are from 0 to 255; the default value is 3.

Reply Message Wait

Enter the time (in ms) that the master waits after receiving an acknowledgment to a master-initiated message before polling the slave for a reply. Specify this time in 20 ms increments; the default value is 50 (i.e., 50*20 ms, or 1000 ms, or 1 second).

Polling Mode

Choose a polling mode from the pull-down menu. Choose from:

- Message Based - slave can initiate messages.
- Message Based - slave cannot initiate messages.
- Standard - Multiple message transfers per node scan.
- Standard - Single message transfer per node scan.

The default mode is Message Based, allowing a slave to initiate messages.

Master Transmit

Choose the master message transmit that designates when to send any DF1 master message. Choose from:

- Between Station Polls - The master transmits a message before the next station.
- In Poll Sequence - The master transmits messages only when the station number is encountered in the poll list.

The default is Between Station Polls.

Normal Poll Node Tag

Choose the tag name of the structure that contains the normal poll node list. Use the Tag Browser to select the appropriate tag name.

The default tag is <none>.

Normal Poll Group Size

Enter the total number of active stations polled from the poll node list. Valid values are 0 to 255; the default value is 0.

Priority Poll Node Tag

Choose the tag name of the structure to store the priority poll node list. Use the Tag Browser to select the appropriate tag name.

The default tag is <none>.

Active Station Tag

Choose the tag name of the structure to store the status (active/ non-active) of each node. Use the Tag Browser to select the appropriate tag name.

The default tag is <none>.

DH485 Parameters

Max Station Address

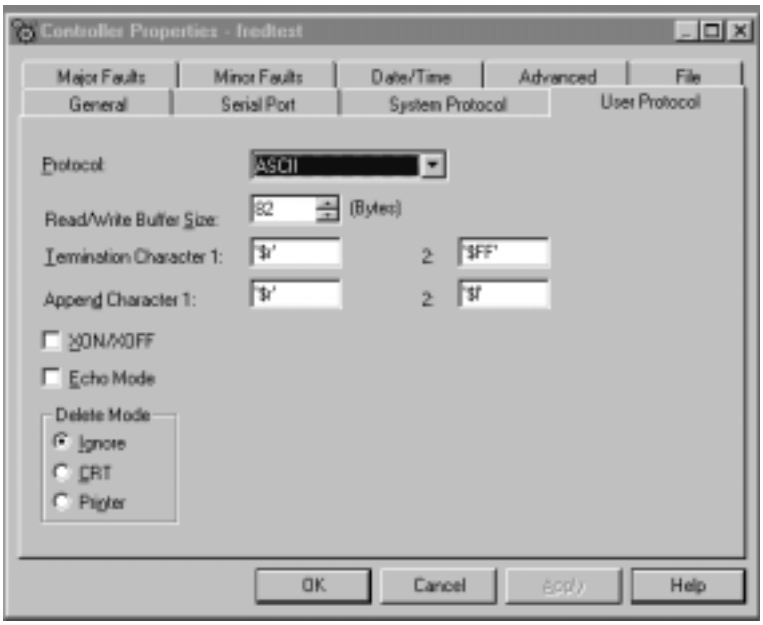
This field is available when you choose DH485 as the protocol. It sets the maximum value allowable for the Station Address. The range is from 0 to 31.

Token Hold Factor

A value between 1 and 4.

User Protocol Tab

The User Protocol tab allows you to configure the controller's serial port for the ASCII protocol.



Protocol

Choose the ASCII protocol.

Buffer Size

Enter the maximum size (in bytes) of the data array that you are planning on sending and receiving. Valid values are from 1 to 65536; the default size is 82.

When the controller sends out the data, if it detects an array that is larger than this buffer size, a minor fault occurs and the extra data is truncated. When the controller receives data, if it detects data that is larger than the size of the buffer, the extra characters are dropped.

Termination Character 1 and 2

Enter the characters that be used to define the end of a line. Valid hex range values are from 0 to 255. The default value for Termination

Character 1 is $0D, and the default value for Termination Character 2 is $FF.

The ARL and ABL instructions use these characters to signal the end of a line. If you do not wish to use these characters, you can either avoid the use of these instructions, or you can define Termination Character 1 as $FF, where $FF tells the controller not to use any definable termination characters when using the ARL or ABL instructions.

If you want to use only one character to signal the end of a line, use Termination Character 1, and define Termination Character 2 as $FF.

Append Character 1 and 2

Enter the characters that are appended to the end of a line. Valid hex range values are from 0 to 255. The default value for Append Character 1 is $0D, and the default value for Append Character 2 is $0A.

The AWA instruction appends the specified characters to the end of the messages it sends out. If you do not wish to use these characters, you can either avoid the use of the AWA instruction, or you can define Append Character 1 as $FF, where $FF tells the controller not to append characters when using the AWA instruction.

If you want to append only one character, define Termination Character 1 as the desired character, and define Termination Character 2 as 0xFF.

XON/OFF

Check this option to regulate the flow of incoming data.

For example, when this option is checked and the receive buffer gets to be 80% full, an XOFF (0x13) character is transmitted to tell the sending device to stop sending. When the buffer has been processed so that it is less than 80% full, the XON (0x11) character is sent to tell the device to resume sending.

This option is disabled when the Control Line option is configured for Half Duplex.

Echo Mode

Check this option to cause any data received in the ASCII port to be sent right back out to the device that sent it. For example, you could use this option with a dumb terminal that is unable to display what it sends, but can display what is echoed to it.

This option is disabled when the Control Line option is configured for Half Duplex.

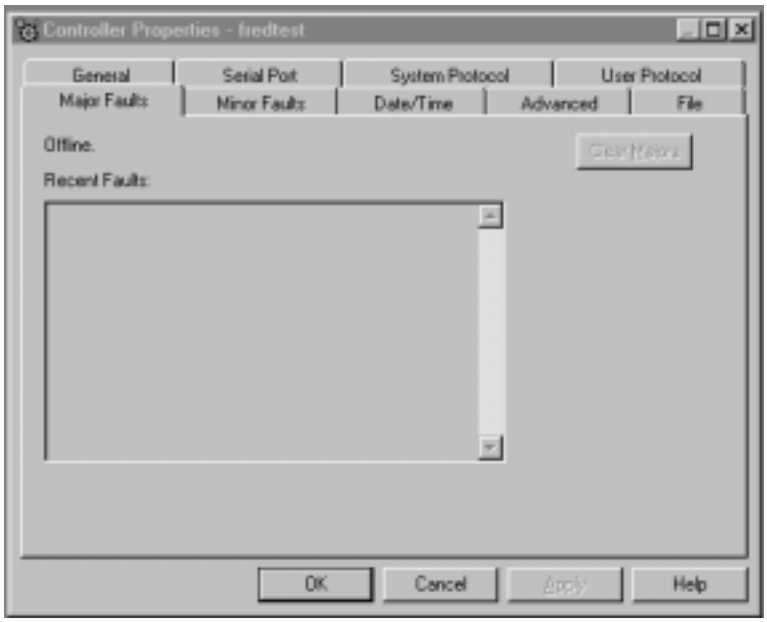
Delete Mode

The character received just before the delete character sequence (0x7F) is removed by the serial port driver before it is given to the ladder logic. Choose from:

- Ignore - The delete character sequence is treated the same as any other character that is read in.
- CRT or Printer - The preceding character in the string buffer is removed before being given to the ladder logic. The only difference between CRT and Printer modes is the type of device sending the string to the controller. If Echo mode is disabled, CRT and Printer do exactly the same thing.

Major Faults Tab

The Major Faults tab displays information on the major faults that have occurred in the controller.



Number of Major Faults Since Last Cleared

Displays the number of major fault events that have been reported since the log was last cleared.

Recent Faults

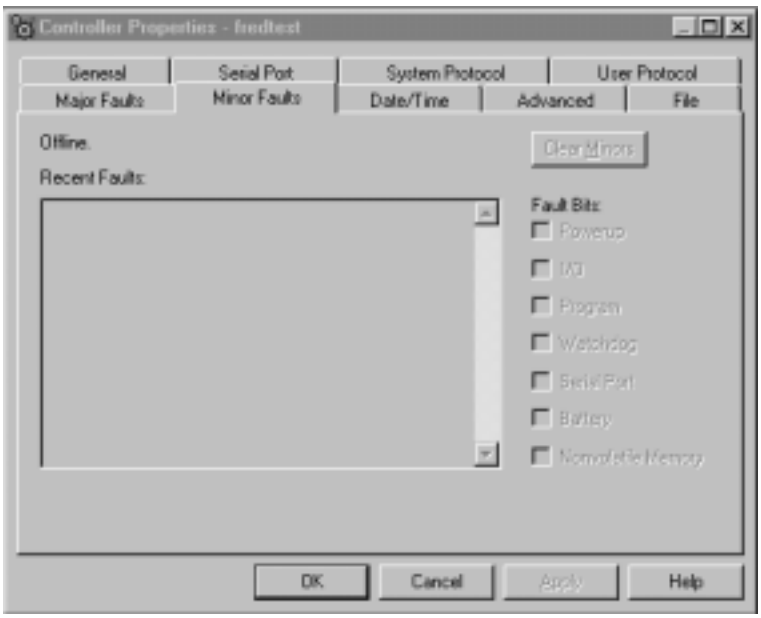
Displays a description of the last three major faults that have occurred. These faults are stored in reverse chronological order. When offline, this field contains the stored contents of the last online session.

Clear Majors

Click on this button to clear the Major Fault log.

Minor Faults Tab

The Minor Faults tab displays information on the minor faults that have occurred in the controller.



Number of Minor Faults Since Last Cleared

Displays the number of minor fault events that have been reported since the log was last cleared.

Recent Faults

Displays a description of the last eight minor faults that have occurred. These faults are stored in reverse chronological order. When offline, this field contains the stored contents of the last online session.

Clear Minors

Click on this button to clear the Minor Fault log.

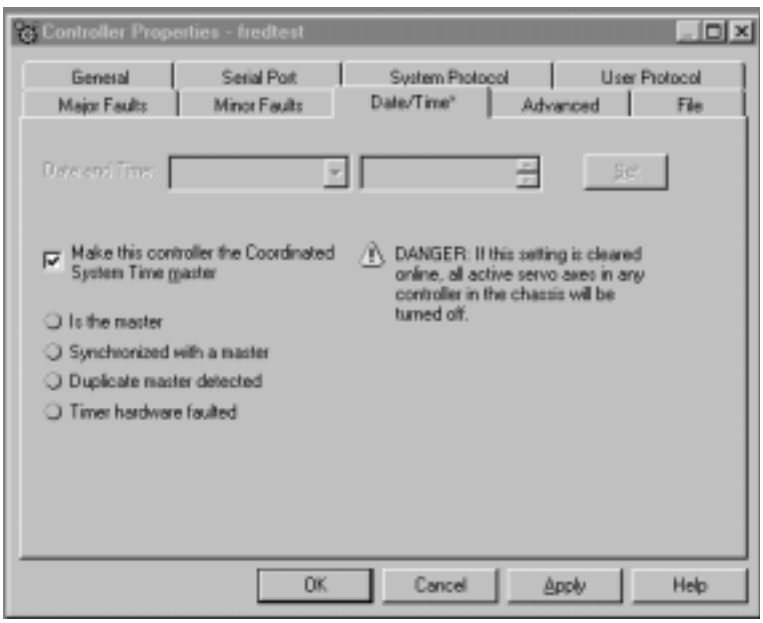
Fault Bits

Lists the minor fault bits that have a specific fault type assigned to them. If the bit is set, the checkbox is set.

When offline, these checkboxes are disabled, but display the contents of the last online session.

Date/Time Tab

The Date/Time tab allows you to view and edit the controller's wall clock time and the coordinated system time status.



Date

The wall clock date, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.

This parameter is read-only. When offline, this parameter is empty.

Time

The wall clock time, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.

This parameter is read-only. When offline, this parameter is empty.

Set

Click on this button to bring up the Set Date/Time dialog, from which you can set the date and/or time.

This button is disabled when offline.

Make this controller the Coordinated System Time master

Click on this checkbox to select this controller as the CST master. This does not mean that this controller IS the master, it means that you intend for this controller to be the master. If another controller is already the CST master, “duplicate master detected” appears in the status field.

If you are using a Servo card on this controller, set the CST Master.

Status

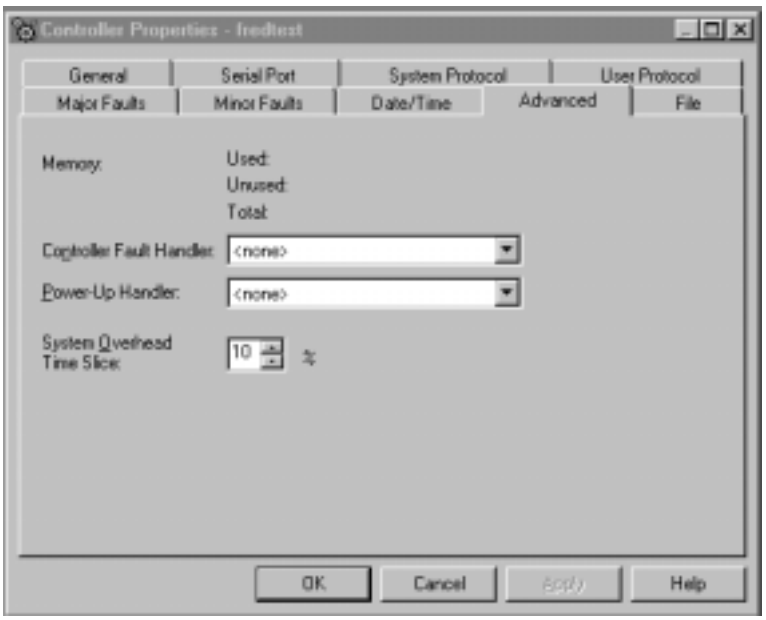
Indicates the state of the coordinated system time. There are four status fields, with a circular indicator to the left of each. This indicator is blue if the corresponding status condition is true; otherwise, it is clear. The status conditions are:

- Is the master - you checked the “Make this controller the master” box, and this controller is the CST master.
- Synchronized with a master - this controller is not the master; its time is being synchronized by a master.
- Duplicate master detected - you checked the “Make this controller the master” box, but there is already a CST master.
- Timer hardware faulted - there is a hardware fault.

All of the circular indicators are clear when you are offline.

Advanced Tab

The Advanced tab allows you to view and edit advanced controller properties.



Memory Used

The amount of memory used in the controller. When offline, this parameter is empty.

Memory Unused

The amount of memory available in the controller. When offline, this parameter is empty.

Memory Total

The total amount of memory in the controller (used plus unused). If a memory daughter card is present, this total includes that memory. When offline, this parameter is empty.

Controller Fault Handler

Choose the program that runs as the result of a system fault from the pull-down menu. The list contains all of the unscheduled programs.

Power-Up Handler

Choose the program the processor executes when it powers up in Run mode after a power-down in Run mode. The list contains all of the unscheduled programs.

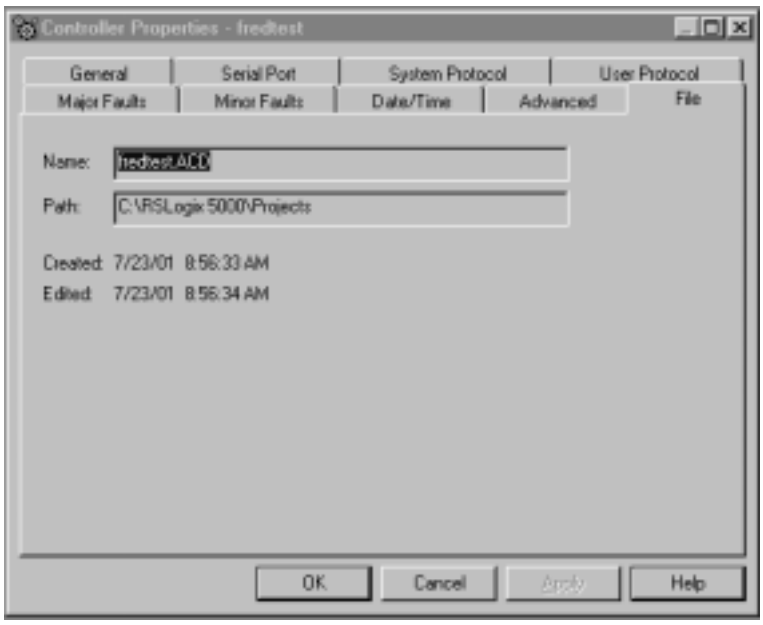
System Overhead Time Slice

Enter or select the percentage of time the controller spends running its system task, relative to running user tasks.

File Tab

The File tab displays information about the project file.

The fields on this tab cannot be edited. To change the file name or path, you must use the Save As command.



Name

The name of the project file

Path

The drive and directory of the project file.

Created

The creation date and time of the project file, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.

Edited

The date and time that the project file was last edited, in the format currently selected in the Regional Settings application in your Windows NT Control Panel.

Adding and Configuring Your 1756-M02AE Motion Module

This chapter describes how to add, configure, and edit your 1756-M02AE motion module for use in your motion control application.

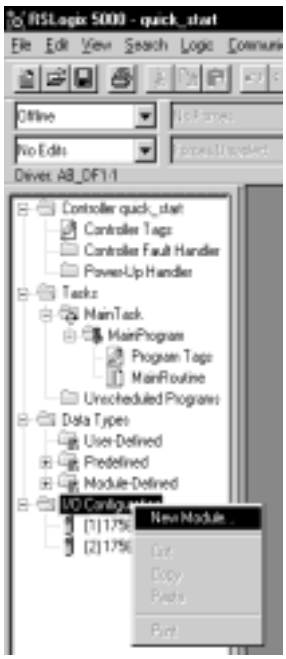
This chapter describes each of the tasks for adding and configuring a motion module.

Adding the 1756-M02AE Module

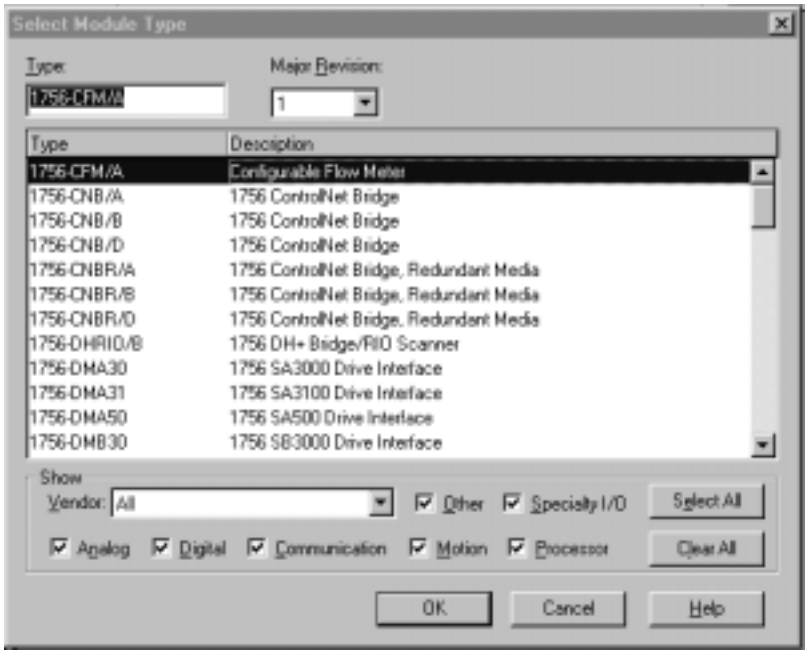
To use your motion module in a control system, you must add your motion module to the application program.

To add a motion module:

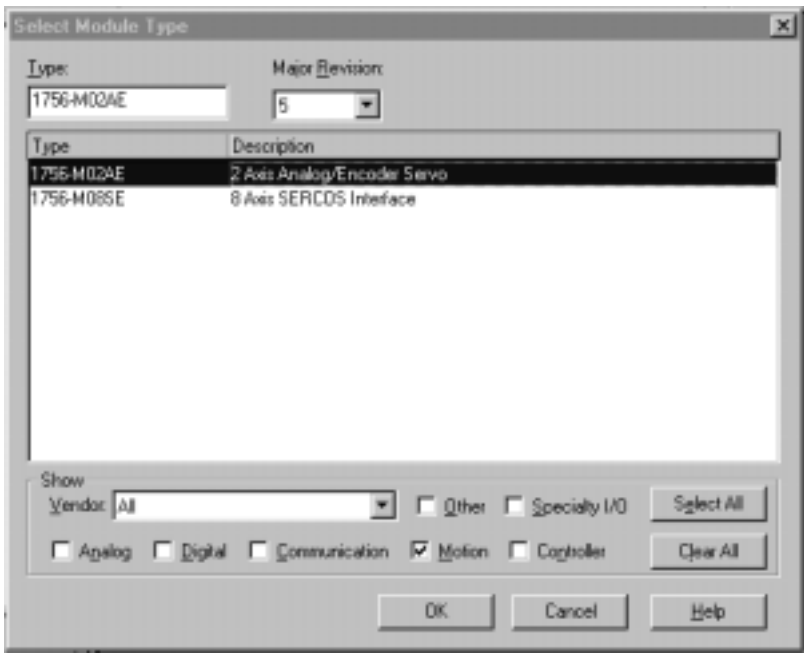
1. Right-click the I/O Configuration folder.



2. Select **New Module**. The Select Module Type window appears.



3. Click on the Clear All button to clear the dialog window then click on Motion to list the available Motion Controllers.



New Module

Use this dialog to select and create a new module. Highlight the 1756-M02AE The context sensitive menu appears, from which you can select a New Module.

Type

The Type field displays the catalog number of the module highlighted in the Type list box. You can either type in a module catalog number in this field to quickly select/find the module you want to create or you can scroll through the list of modules in the Type list box.

Major Revision

Select the major revision number of the physical module that you think want to reside in the chassis.

The major revision is used to indicate the revision of the interface to the module.

Type (list box)

This box lists the installed module catalog numbers based on the selected check boxes.

Description (list box)

This portion of the list box contains descriptions of the modules.

Show:

Displays check boxes, which support filtering on particular types of modules.

Check this box:	If you want to:
Digital	display digital modules supported by the software
Analog	display analog modules supported by the software
Communication	display communication modules supported by the software
Motion	display motion modules supported by the software
Controller	display controller modules supported by the software
Vendor	display a particular vendor's module profiles that are installed on the system.
Other	display modules that do not fit under the rest of the check box categories.

Select All

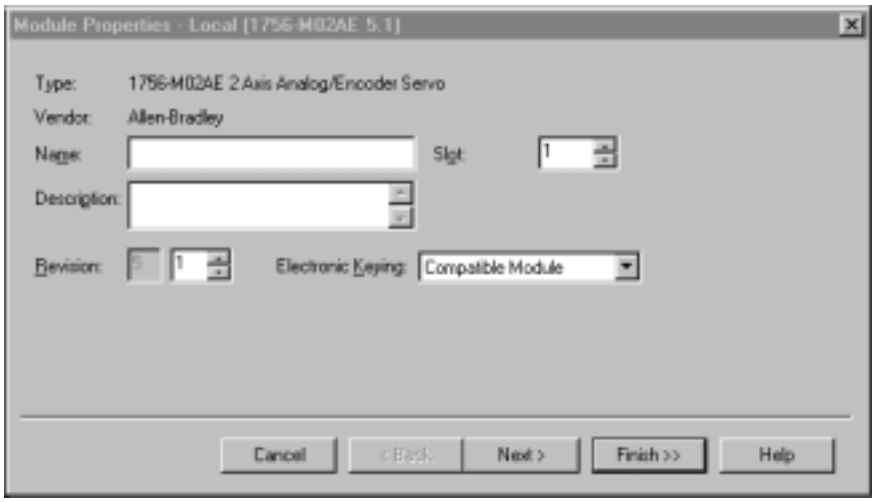
Click on this button to display all modules in the list box; all the check boxes in the Show field are checked.

Clear All

Click on this button to clear all check boxes in the Show field.

4. In the *Type* field, select **1756-M02AE 2 Axis Analog/Encoder Servo**.

5. Select **OK**. The Module Crate Wizard displays.

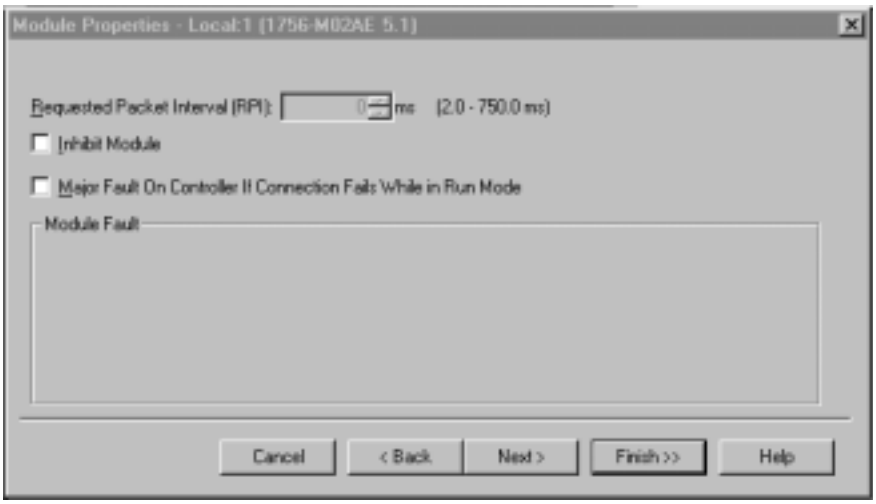


6. Make entries in the following fields.

Field	Entry
Name	Type a name for the servo module. The name can: • have a maximum of 40 characters • contain letters, numbers and underscores (_).
Slot	Enter the number of the chassis slot that contains your module.
Description	Type a description for your motion module. This field is optional.

Field	Entry	
Electronic keying	Select the electronic keying level.	
	To	**Select**
	Match the vendor, catalog number, and major revision attributes of the physical module and the software configured module	Compatible module
	Disable the electronic keying protection mode	Disable keying
	Match the vendor, catalog number, major revision, and minor revision attributes of the physical module and the software configured module	Exact match

7. Press the Next button to proceed to the next Create Wizard screen.



8. This screen is where you determine how faults are to be handled. The choices are to inhibit module or to configure the module so that a loss of connection to this module causes a major fault. Make your entries and press the Next button to

proceed to the next wizard screen.

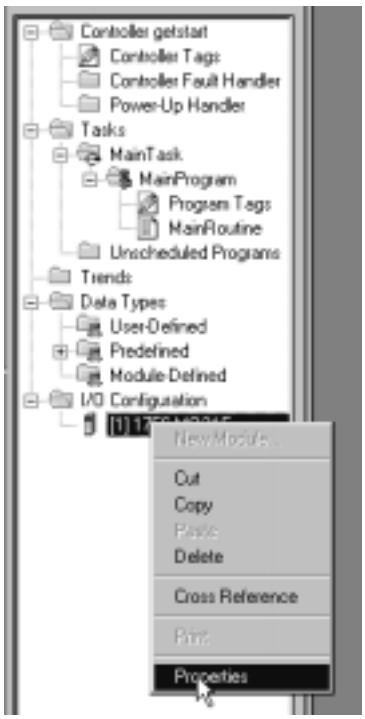
9. This screen lets you associate an axis with the module. Make the appropriate choices for you situation. At this point, the rest of the screens are informational only and it would be best to press the Finish button to create the module.

All of the above screens can be accessed and edited by going to the tabbed Module Property screens. Further explanations of the fields in this dialog are detailed below.

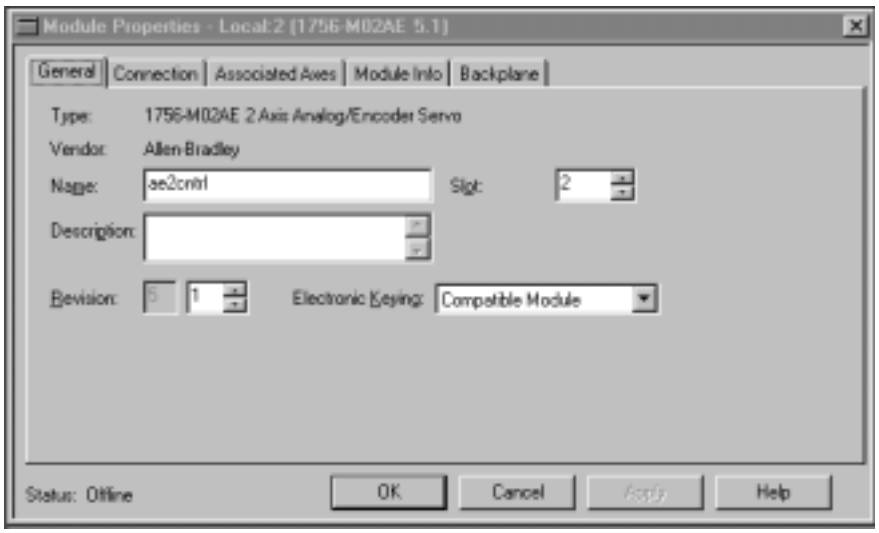
Editing Your Motion Module Settings

The following section provides explanations of the Motion Module Properties screens. Use these screens to edit the properties of the module when changes need to be made. You can access the Module Properties screen by highlighting the motion module and right

clicking the mouse. Select **Properties** from the displayed pop-up menu screen as shown in the following figure.



This accesses the Module Properties screen. This screen is tabbed to take you to the particular dialog you require.



General Tab

Use this tab to create/view module properties for 1756-M02AE motion module. This dialog provides you with the means to view the type, description, vendor, and the name of the parent module. You can also enter the name and a description for the module. Other fields and buttons on this dialog let you set the slot location of the module, review information for both channels, go to the New Tag dialog to create an axis to associate with one of the channels, select the minor revision number and select an electronic keying option. You can also view the status the controller has about the module but, only when online.

Type

Displays the type and description of the module being created (read only).

Vendor

Displays the vendor of the module being created (read only).

Name

Enter the name of the module.

The name must be IEC 1131-3 compliant. If you attempt to enter an invalid character or exceed the maximum length, the software beeps and ignores the character.

Description

Enter a description for the module here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software beeps to warn you, and ignores any extra characters.

Slot

Enter the slot number where the module resides. The spin button contains values that range from 0 to 1 less than the chassis size (e.g., if you have a 4-slot chassis, the spin button will spin from 0 to 3). If you enter a slot number that is out of this range, you will receive an error message when you go to apply your changes.

The slot number cannot be changed when online.

Revision

Select the minor revision number of your module.

The revision is divided into the major revision and minor revision. The major revision displayed statically is chosen on the Select Module Type dialog.

Electronic Keying

Select one of these keying options for your module during initial module configuration:

- Exact Match - all of the parameters described below must match or RSLogix will reject the inserted module.
- Compatible Module
  - the Module Types, Catalog Number, and Major Revision must match
  - the Minor Revision of the physical module must be equal to or greater than the one specified in the software or RSLogix 5000 will reject the inserted module.
- Disable Keying - RSLogix 5000 will not employ keying at all.

When you insert a module into a slot in a ControlLogix chassis, RSLogix 5000 compares the following information for the inserted module to that of the configured slot:

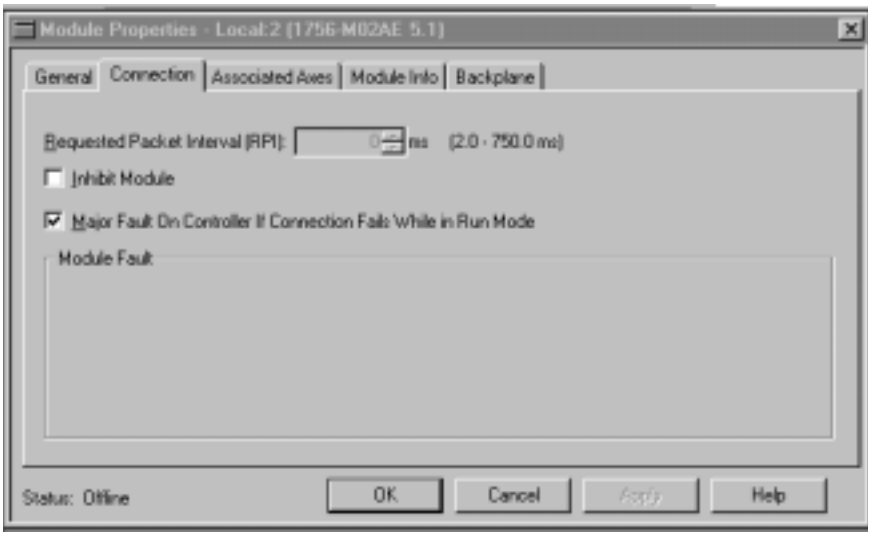
- Vendor
- Product Type
- Catalog Number
- Major Revision
- Minor Revision

This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

Connection Tab

The Connection Tab is used to define controller to module behavior. This is where you select a requested packet interval, choose to inhibit

the module, configure the controller so loss of the connection to this module causes a major fault, and view module faults.



The data on this tab comes directly from the controller. This tab displays information about the condition of the connection between the module and the controller.

Requested Packet Interval

This does not apply to motion module.

Inhibit Module checkbox

Check/Uncheck this box to inhibit/uninhibit your connection to the module. Inhibiting the module causes the connection to the module to be broken.

TIP

Inhibiting/uninhibiting connections applies mainly to direct connections, and not to the CNB module

ATTENTION

Inhibiting the module causes the connection to the module to be broken and may result in loss of data

When you check this box and go online, the icon representing this module in the controller organizer displays the Attention Icon.

If you are:	Check this checkbox to:
offline	put a place holder for a module you are configuring
online	stop communication to a module If you inhibit the module while you are online and connected to the module, the connection to the module is nicely closed. The module's outputs go to the last configured Program mode state. If you inhibit the module while online but a connection to the module has not been established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is 'Inhibited' and not 'Faulted'. If you uninhibit a module (clear the checkbox) while online, and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured (if you are the owner controller) with the configuration you have created for that module. If you are a listener (have chosen a "Listen Only" Communications Format), you can not re-configure the module. If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.

Major Fault on Controller if Connection Fails checkbox

Check this box to configure the controller so that failure of the connection to this module causes a major fault on the controller if the connection for the module fails.

Module Fault

Displays the fault code returned from the controller (related to the module you are configuring) and the text detailing the Module Fault that has occurred.

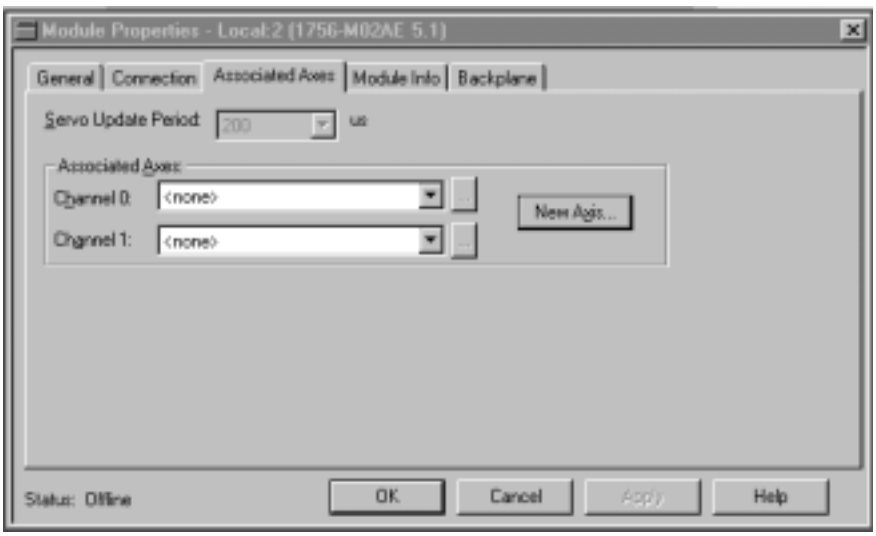
The following are common categories for errors:

- Connection Request Error - The controller is attempting to make a connection to the module and has received an error . The connection was not made.
- Service Request Error - The controller is attempting to request a service from the module and has received an error. The service was not performed successfully.
- Module Configuration Invalid - The configuration in the module is invalid. (This error is commonly caused by the Electronic Key Passed fault).
- Electronic Keying Mismatch - Electronic Keying is enabled and some part of the keying information differs between the software and the module.

Associated Axes Tab

This tab lets you assign axis tags to specific channels of the servo module. Use this tab to configure the selected 1756-M02AE motion modules by:

- setting the selected 1756-M02AE motion module's Servo Update Period
- associating axis tags, of the type AXIS_SERVO, with channels 0 and 1



Servo Update Period

Selects the periodic rate at which the 1756-M02AE module closes the servo loop for the axis, in microseconds (µs).

Channel 0

Represents Channel 0 on the servo module. This field allows you to associate an AXIS_SERVO tag with channel 0. This field transitions to a read-only state while online. Click on the button to the right of this field to open the Axis Properties dialog for the associated axis.

Channel 1

Represents Channel 1 on the servo module. This field allows you to associate an AXIS_SERVO tag with channel 1. This field transitions to a read-only state while online. Click on the button to the right of this field to open the Axis Properties dialog for the associated axis.

New Axis button

Click on this button to navigate to the New Tag dialog to create an AXIS_SERVO tag to associate with one of the channels.

Module Info Tab

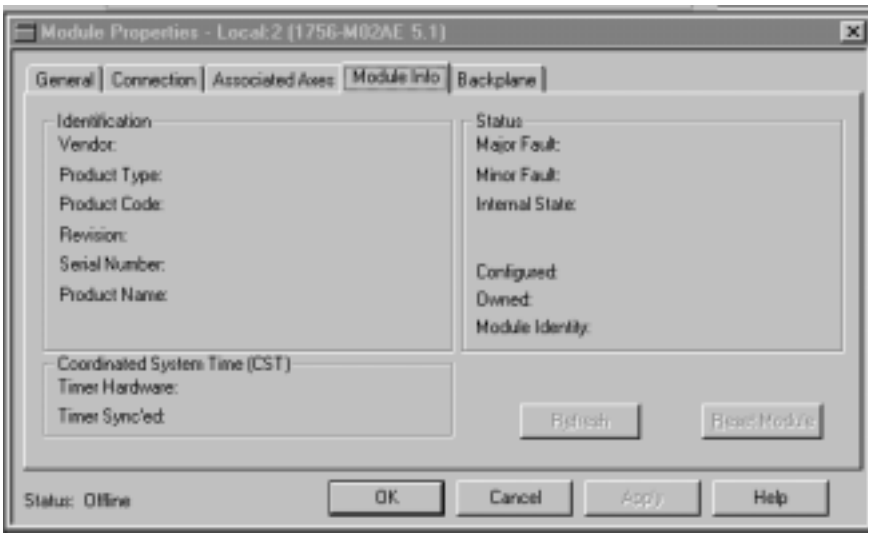
The Module Info tab contains information about the selected module, however, you can click on:

- Refresh – to display new data from the module.
- Reset Module – to return the module to its power-up state by emulating the cycling of power. By doing this, you also clear all faults.

The Module Info Tab displays module and status information about the module. It also allows you to reset a module to its power-up state. The information on this tab is not displayed if you are offline or currently creating a module.

Use this tab to determine the identity of the module.

The data on this tab comes directly from the module. If you selected a Listen-Only communication format when you created the module, this tab is not available.



Identification

Displays the module's:

- Vendor
- Product Type
- Product Code
- Revision Number
- Serial Number
- Product Name

The name displayed in the Product Name field is read from the module. This name displays the series of the module. If the module is a 1756-L1 module, this field displays the catalog number of the memory expansion board (this selection applies to any controller catalog number even if additional memory cards are added: 1756-L1M1, 1756-L1M2).

Major/Minor Fault Status

If you are configuring a:	This field displays one of the following:
digital module	EEPROM fault Backplane fault None
analog module	Comm. Lost with owner Channel fault None
any other module	None Unrecoverable Recoverable

Internal State Status

This field displays the module's current operational state.

- Self-test
- Flash update
- Communication fault
- Unconnected
- Flash configuration bad
- Major Fault
- Run mode
- Program mode
- (16#xxxx) unknown

If you selected the wrong module from the module selection tab, this field displays a hexadecimal value. A textual description of this state is only given when the module identity you provide is a match with the actual module.

Configured

This field displays a yes or no value indicating whether the module has been configured by an owner controller connected to it. Once a module has been configured, it stays configured until the module is

reset or power is cycled, even if the owner drops connection to the module.

Owned

This field displays a yes or no value indicating whether an owner controller is currently connected to the module.

Module Identity

Displays:	If the module in the physical slot:
Match	agrees with what is specified on the General Tab. In order for the Match condition to exist, all of the following must agree: • Vendor • Module Type (the combination of Product Type and Product Code for a particular Vendor) • Major Revision
Mismatch	does not agree with what is specified on the General Tab

This field does not take into account the Electronic Keying or Minor Revision selections for the module that were specified on the General Tab.

Refresh

Click on this button to refresh the tab with new data from the module.

Reset Module

Click on this button to return a module to its power-up state by emulating the cycling of power.

Resetting a module causes all connections to or through the module to be closed, and this may result in loss of control.

IMPORTANT The following modules return an error if a reset is attempted:1756-L1 ControlLogix5550 Programmable Controller; 1336T AC Vector Drive; 1395 Digital DC Drive.

A controller cannot be reset.

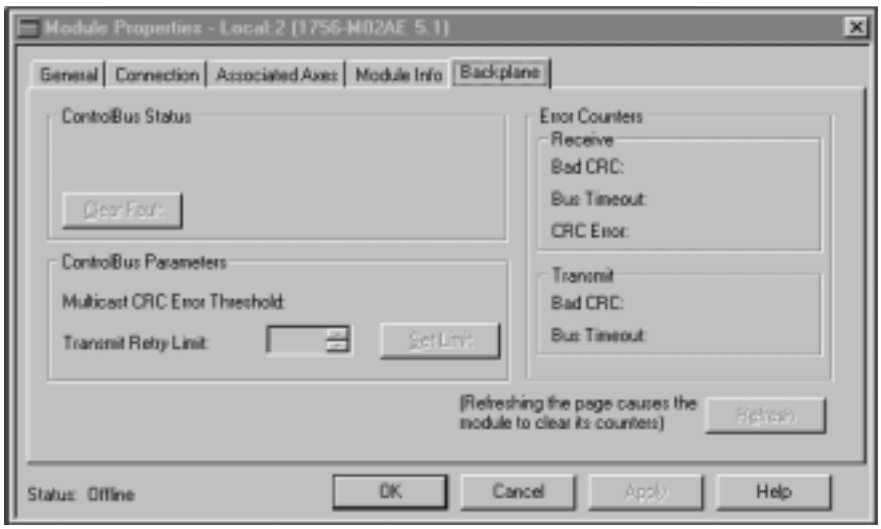
Backplane Tab

The Backplane tab on the Module Properties window is displayed for informational purposes. You can use this tab to review diagnostic information about the module's communications over the backplane and the chassis in which it is located, clear a fault, and set the transmit retry limit.

Information on this tab is displayed only if you are online.

If you selected a Listen-Only communication format when you created the module , this tab is not available.

The data on this tab comes directly from the module.



ControlBus Status

This box either displays OK or one of the following errors:

- Receiver disabled
- Multicast addresses disabled
- RA/GA miscompare

To clear the module's backplane fault, click the Clear Fault button .

ControlBus Parameters

This box contains the following fields and button.

Multicast CRC Error Threshold

This value is the point where it enters a fault state because of Cyclic Redundancy Check (CRC) errors.

Transmit Retry Limit

Not applicable to motion module.

Set Limit Button

You must click on the Reset Limit button to make the new Transmit Retry Limit effective. If you do not and then click either the OK or the Apply button, this limit is not set.

Receive Error Counters

This box displays the number of receiving errors that occurred in the following categories:

- Bad CRC – errors that occurred on received frames (messages)
- Bus time-out – when the receiver timed out
- CRC error – multicast receive errors

Transmit Error Counters

This box displays the number of transmitting errors that occurred in the following categories:

- Bad CRC – errors that occurred on transmitted frames
- Bus Time-out – when the transmitter bus timed out

Refresh

Click on the Refresh button to refresh the tab. When you refresh the tab:

if you're using:	**then:**
digital, analog, or motion modules	counters are cleared
another module	the tab is refreshed but the counters are not cleared

Assigning Additional Motion Modules

You can assign additional modules by repeating the preceding sections. You can assign up to 16 1756-M02AE modules to each Logix5550 controller. Each module uses a maximum of two axes.

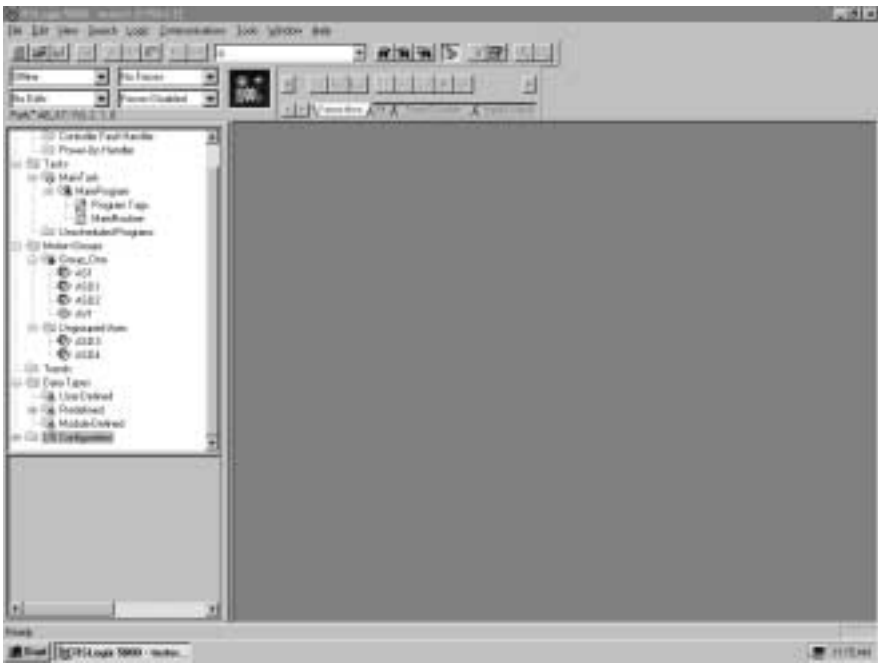
Configuring the 1756-M08SE Module

Adding the 1756-M08SE

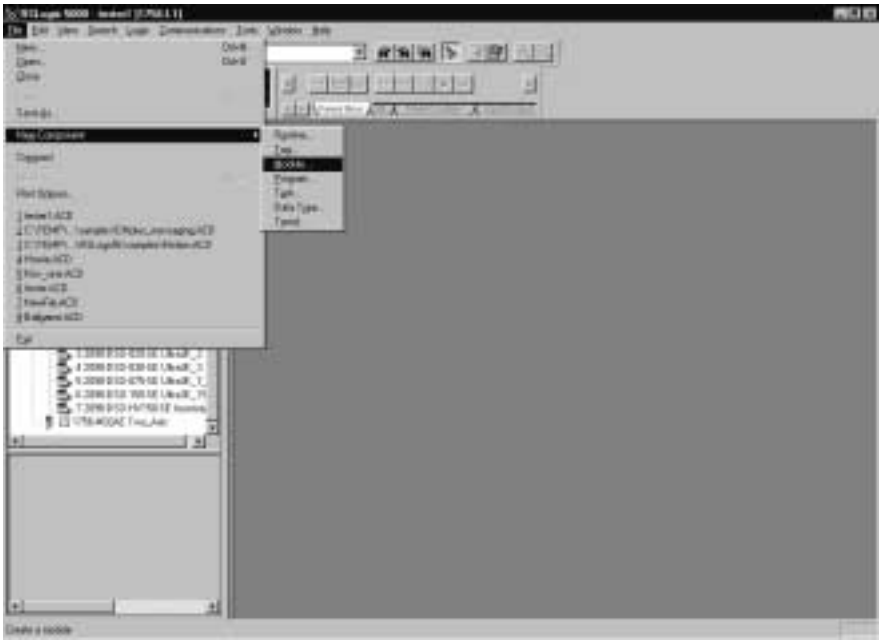
This chapter reviews the necessary steps for configuring the 1756-M08SE motion module. Much of this information is the same as for adding and configuring the 1756-M02AE as discussed in the previous chapter.

To configure a 1756-M08SE motion module:

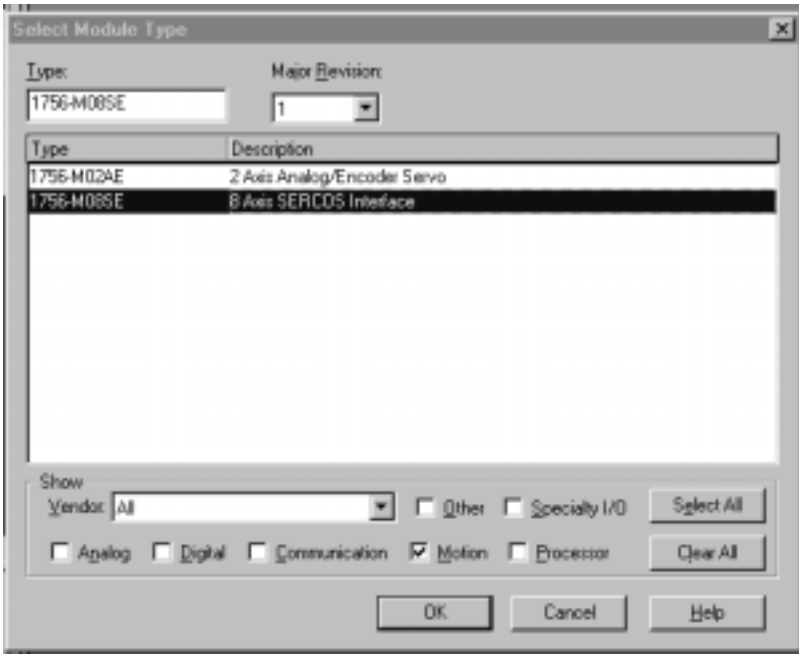
1. In the Controller Organizer, select the I/O Configuration branch.



2. In the File menu, select New Component then Module…

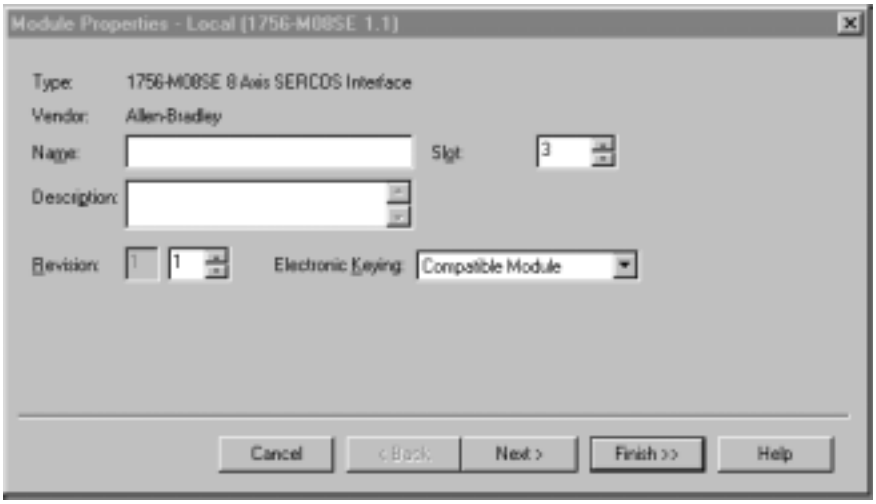


3. The Select Module Type screen displays. Select Clear All. Select Motion. The list displays only available motion modules.



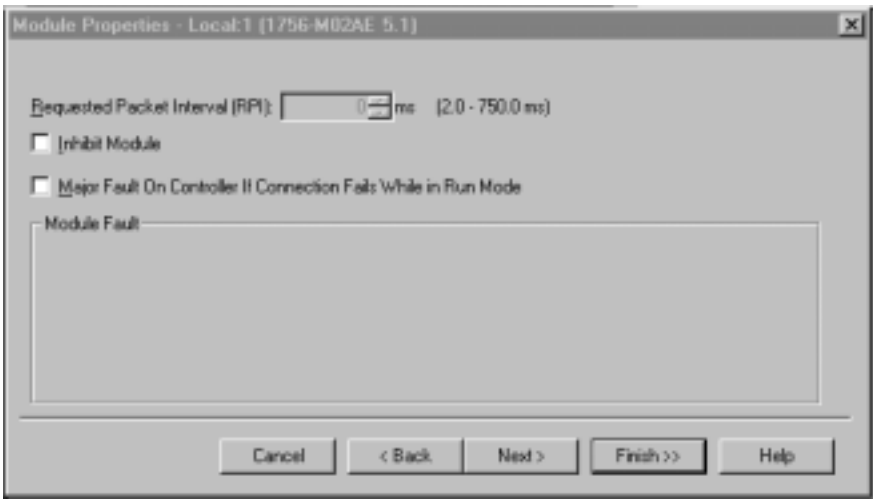
4. Select 1756-M08SE.

5. Press the OK button to close the Select Module Type dialog. The Create Module Wizard opens.



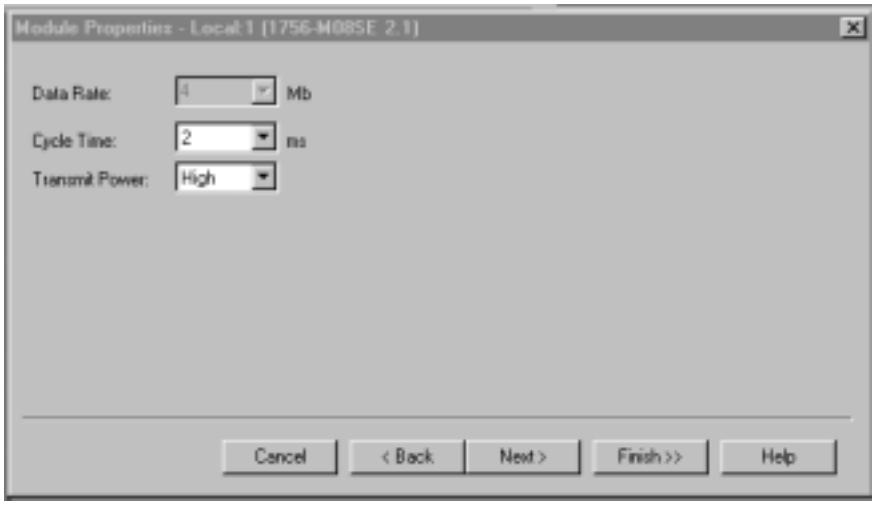
6. Name is the only required field that must be entered to create the M08SE module. It must conform to the IEC 1131-3 standard. You can also enter a description for the module, select the minor revision number of your module, and select the method for Electronic Keying. Fill in the at least the required Name filed and click the Next> button to advance to the next wizard screen to enter Connection information. (See the section titled *1756-M08SE 8 Axis Motion Module Overview* in this chapter for more information on the fields in these screens.)

7. The Connection Screen displays.



8. On this screen there are no required fields but you can enter how you want to handle connection faults. You can either choose to inhibit the module if the connection fails or you can have a major fault. After checking the appropriate box, click on Next> to advance the SERCOS interface Create Wizard screen.

9. The SERCOS interface screen displays.

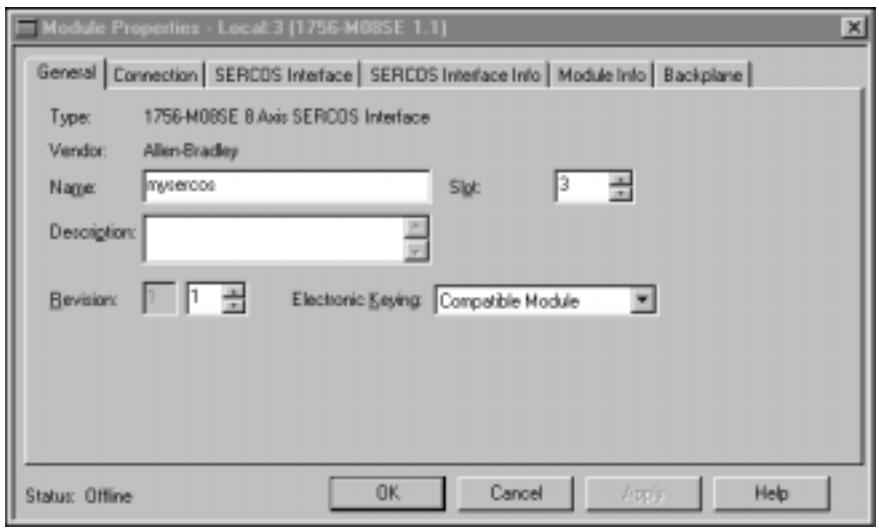


10. On this scree you can enter the SERCOS ring Cycle time and the transmit power for the SERCOS ring. The rest of the Create Wizard screens are only informational and do not let you enter any information. It saves time if you click on the Finish>> button at this time.

11. The 1756-M08SE motion module appears in the I/O Configuration branch of the Controller Organizer. It can now be put into use or edited as you require.

1756-M08SE 8 Axis Motion Module Overview

The 1756-M08SE 8 Axis SERCOS motion module has been added. To edit the 1756-M08SE Module Properties, go to the I/O Configuration organizer and right click on the 1756-M08SE module and select Properties from the drop down menu. The tabbed Module Properties screen displays.



The Module Properties screen has the following tabs:

- The General tab references the 1756-M08SE 8 Axis motion module.
- The Connection tab references the connection of the module to the controller.
- The SERCOS Interface tab is for configuring SERCOS communication settings for the 1756-M08SE 8 Axis motion module.
- The SERCOS Interface Info tab is used to monitor the status of the SERCOS communication ring.
- The Module Info tab, when Online, displays the current condition of the module.
- The Backplane tab, when Online, displays diagnostic information about the module's communication over the backplane and the chassis in which it is located.

Editing 1756-M08SE Module Properties

General Tab

Use this tab to create/view module properties for the 1756-M08SE 8 Axis SERCOS motion module.

On this tab, you can:

- view the type and description of the module being created
- view the vendor of the module being created
- enter the name of the module
- enter a description for the module
- select the slot number of the module on the network
- select the minor revision number of your module
- select Exact Match, Compatible Module, or Disable Keying

Type

Displays the type and description of the module being created (read only).

Vendor

Displays the vendor of the module being created (read only).

Name

Enter the name of the module. The name must be IEC 1131-3 compliant. If you attempt to enter an invalid character or exceed the maximum length, the software beeps and ignores the character.

Description

Enter a description for the module here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software beeps to warn you, and ignores any extra characters.

Slot

Enter the slot number where the module resides. The spin button contains values that range from 0 to 1 less than the chassis size (e.g., if you have a 4-slot chassis, the spin button spins from 0 to 3). If you enter a slot number that is out of this range, you receive an error message when you apply your changes.

The slot number cannot be changed when online.

Revision

The revision is divided into the major revision and minor revision. The major revision displayed statically is chosen on the Select Module Type dialog.

The major revision is used to indicate the revision of the interface to the module. The minor revision is used to indicate the firmware revision.

Select the minor revision number of your module.

Electronic Keying

Select one of these keying options for your module during initial module configuration:

Exact Match - all of the parameters must match or the inserted module rejects the connection.

Compatible Module - the Module Types, Catalog Number, and Major Revision must match. The Minor Revision of the physical module must be equal to or greater than the one specified in the software or the inserted module rejects the connection.

Disable Keying – Logix5550 does not employ keying at all.

WARNING

Changing the RPI and Electronic Keying selections may cause the connection to the module to be broken and may result in a loss of data.

Be extremely cautious when using this option; if used incorrectly, this option can lead to personal injury or death, property damage or economic loss.

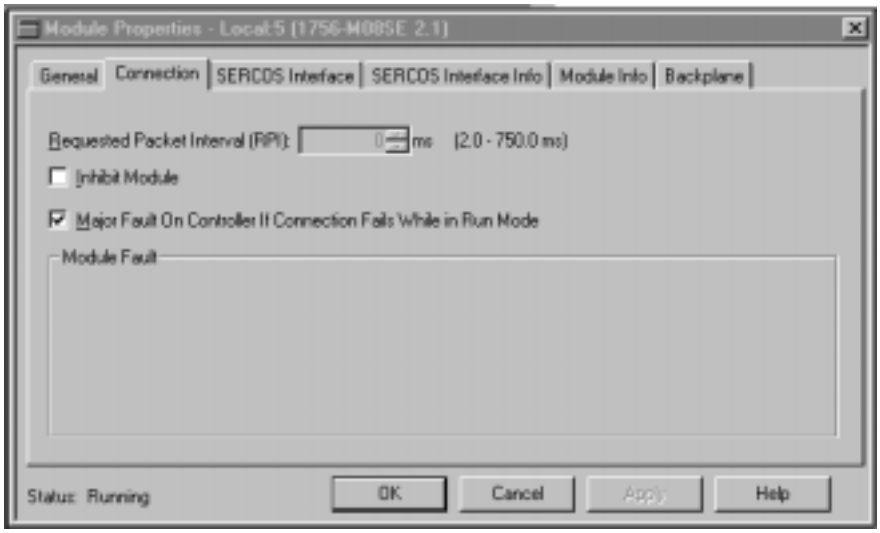
When you insert a module into a slot in a ControlLogix chassis, RSLogix5000 compares the following information for the inserted module to that of the configured slot:

- Vendor
- Product Type
- Catalog Number
- Major Revision
- Minor Revision

This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

Connection Tab

The Connection Tab is used to define controller to module behavior. This is where you select a requested packet interval, choose to inhibit the module, configure the controller so loss of the connection to this module causes a major fault, and view module faults.



The data on this tab comes directly from the controller. This tab displays information about the condition of the connection between the module and the controller.

Requested Packet Interval

This does not apply to motion module.

Inhibit Module checkbox

Check/Uncheck this box to inhibit/uninhibit your connection to the module. Inhibiting the module causes the connection to the module to be broken.

TIP

Inhibiting/uninhibiting connections applies mainly to direct connections, and not to the CNB module

WARNING

Inhibiting the module causes the connection to the module to be broken and may result in loss of data

When you check this box and go online, the icon representing this module in the controller organizer displays the Attention Icon.

If you are:	Check this checkbox to:
offline	put a place holder for a module you are configuring
online	stop communication to a module If you inhibit the module while you are online and connected to the module, the connection to the module is nicely closed. The module's outputs go to the last configured Program mode state. If you inhibit the module while online but a connection to the module has not been established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is 'Inhibited' and not 'Faulted'. If you uninhibit a module (clear the checkbox) while online, and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured (if you are the owner controller) with the configuration you have created for that module. If you are a listener (have chosen a "Listen Only" Communications Format), you can not re-configure the module. If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.

Major Fault on Controller if Connection Fails checkbox

Check this box to configure the controller so that failure of the connection to this module causes a major fault on the controller if the connection for the module fails.

Module Fault

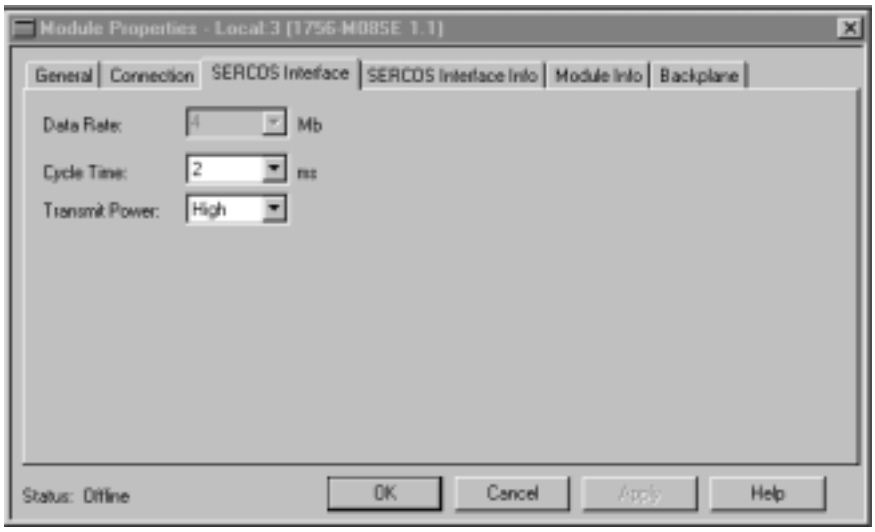
Displays the fault code returned from the controller (related to the module you are configuring) and the text detailing the Module Fault that has occurred.

The following are common categories for errors:

- Connection Request Error - The controller is attempting to make a connection to the module and has received an error. The connection was not made.
- Service Request Error - The controller is attempting to request a service from the module and has received an error. The service was not performed successfully.
- Module Configuration Invalid - The configuration in the module is invalid. (This error is commonly caused by the Electronic Key Passed fault).
- Electronic Keying Mismatch - Electronic Keying is enabled and some part of the keying information differs between the software and the module.

SERCOS Interface Tab

The SERCOS interface Tab is where you set the specific Data Rate, Cycle Time, and Transmit Power for the named 1756-M08SE SERCOS interface module.



Use the SERCOS Interface Tab to set and display the:

- SERCOS baud rate
- update rate for the SERCOS ring
- fiber optic transmit power range for the SERCOS ring

The SERCOS ring consists of the drives and axes connected to the 1756-M08SE motion controller.

TIP The settings on this tab are specific to the 1756-M08SE motion controller.

Data Rate

Select the baud rate for the SERCOS ring. (For this release, this value is set to 4 MB and is Read Only.)

Cycle Time

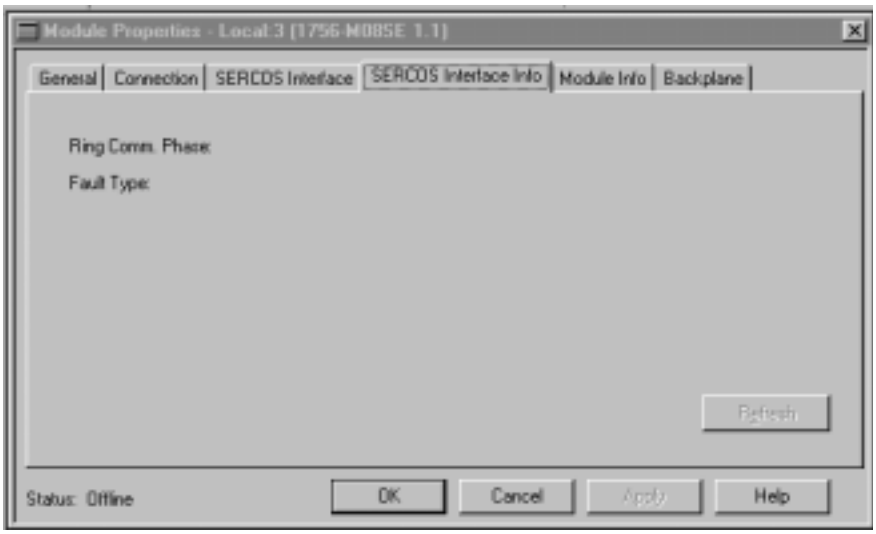
Select the update rate for the SERCOS ring: 1 ms or 2 ms.

Transmit Power

Select the optic transmit power range for the SERCOS ring: High or Low.

SERCOS Interface Info Tab

The SERCOS interface Tab is for monitoring the SERCOS ring of the selected 1756-M08SE while it is on-line. A REFRESH button is available to access the current values.



Use this tab to monitor the following:

Ring Comm. Phase

Displays the communications phase of the SERCOS ring:

0: Ring Integrity

1: Polling

2: Identity

3: Configuration

4: Cyclic communication

Fault Type

Displays the current fault type, if any, on the SERCOS ring. Values include:

No Fault

Open Ring

Not communicating

Not responding

Timing error

Duplicate node

Excess nodes on Ring

Invalid data rate

Invalid cycle time

Refresh

Click this button to update this page.

Note: this information does not refresh automatically.

Module Info Tab

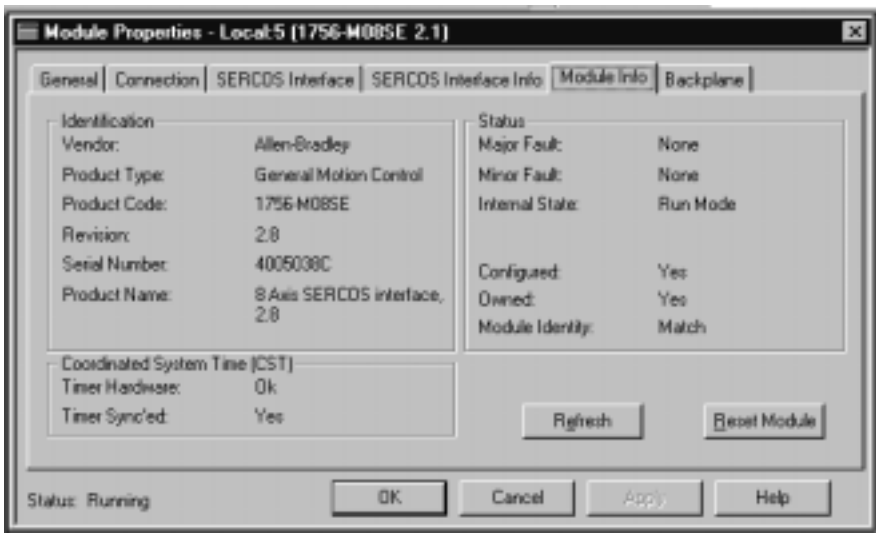
The Module Info tab contains information about the selected module, however, you can click on:

- Refresh – to display new data from the module.
- Reset Module – to return the module to its power-up state by emulating the cycling of power. By doing this, you also clear all faults.

The Module Info Tab displays module and status information about the module. It also allows you to reset a module to its power-up state. The information on this tab is not displayed if you are offline or currently creating a module.

Use this tab to determine the identity of the module.

The data on this tab comes directly from the module. If you selected a Listen-Only communication format when you created the module, this tab is not available.



Identification

Displays the module's:

- Vendor
- Product Type
- Product Code
- Revision Number
- Serial Number
- Product Name

The name displayed in the Product Name field is read from the module. This name displays the series of the module. If the module is a 1756-L1 module, this field displays the catalog number of the memory expansion board (this selection applies to any controller catalog number even if additional memory cards are added: 1756-L1M1, 1756-L1M2).

Major/Minor Fault Status

If you are configuring a:	This field displays one of the following:
digital module	EEPROM fault Backplane fault None
analog module	Comm. Lost with owner Channel fault None
any other module	None Unrecoverable Recoverable

Internal State Status

This field displays the module's current operational state.

- Self-test
- Flash update
- Communication fault
- Unconnected
- Flash configuration bad
- Major Fault
- Run mode
- Program mode
- (16#xxxx) unknown

If you selected the wrong module from the module selection tab, this field displays a hexadecimal value. A textual description of this state is only given when the module identity you provide is a match with the actual module.

Configured

This field displays a yes or no value indicating whether the module has been configured by an owner controller connected to it. Once a module has been configured, it stays configured until the module is

reset or power is cycled, even if the owner drops connection to the module.

Owned

This field displays a yes or no value indicating whether an owner controller is currently connected to the module.

Module Identity

Displays:	If the module in the physical slot:
Match	agrees with what is specified on the General Tab. In order for the Match condition to exist, all of the following must agree: • Vendor • Module Type (the combination of Product Type and Product Code for a particular Vendor) • Major Revision
Mismatch	does not agree with what is specified on the General Tab

This field does not take into account the Electronic Keying or Minor Revision selections for the module that were specified on the General Tab.

Refresh

Click on this button to refresh the tab with new data from the module.

Reset Module

Click on this button to return a module to its power-up state by emulating the cycling of power.

Resetting a module causes all connections to or through the module to be closed, and this may result in loss of control.

IMPORTANT The following modules return an error if a reset is attempted:1756-L1 ControlLogix5550 Programmable Controller; 1336T AC Vector Drive; 1395 Digital DC Drive.

A controller cannot be reset.

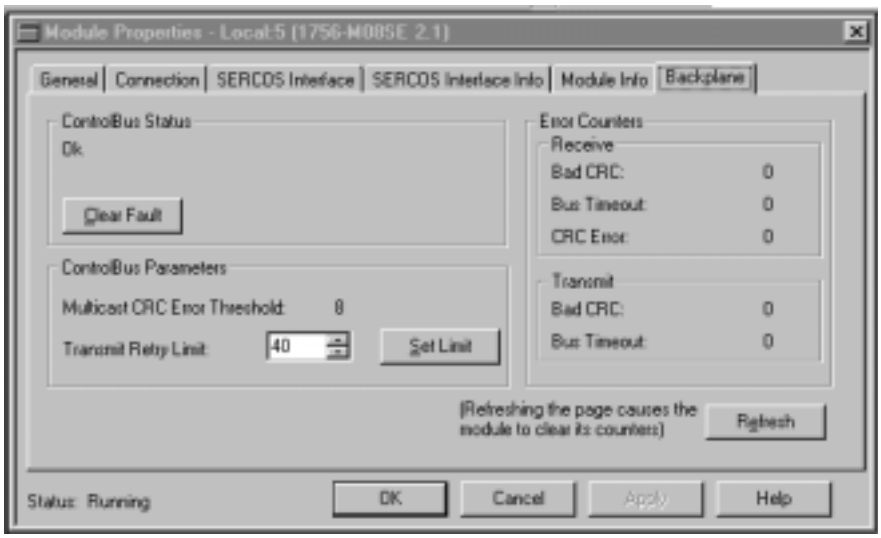
Backplane Tab

The Backplane tab on the Module Properties window is displayed for informational purposes. You can use this tab to review diagnostic information about the module's communications over the backplane and the chassis in which it is located, clear a fault, and set the transmit retry limit.

Information on this tab is displayed only if you are online.

If you selected a Listen-Only communication format when you created the module, this tab is not available.

The data on this tab comes directly from the module.



ControlBus Status

This box either displays OK or one of the following errors:

- Receiver disabled
- Multicast addresses disabled
- RA/GA miscompare

To clear the module's backplane fault, click the Clear Fault button.

ControlBus Parameters

This box contains the following fields and button.

Multicast CRC Error Threshold

This value is the point where it enters a fault state because of Cyclic Redundancy Check (CRC) errors.

Transmit Retry Limit

Not applicable to motion module.

Set Limit Button

You must click on the Reset Limit button to make the new Transmit Retry Limit effective. If you do not and then click either the OK or the Apply button, this limit is not set.

Receive Error Counters

This box displays the number of receiving errors that occurred in the following categories:

- Bad CRC – errors that occurred on received frames (messages)
- Bus time-out – when the receiver timed out
- CRC error – multicast receive errors

Transmit Error Counters

This box displays the number of transmitting errors that occurred in the following categories:

- Bad CRC – errors that occurred on transmitted frames
- Bus Time-out – when the transmitter bus timed out

Refresh

Click on the Refresh button to refresh the tab. When you refresh the tab:

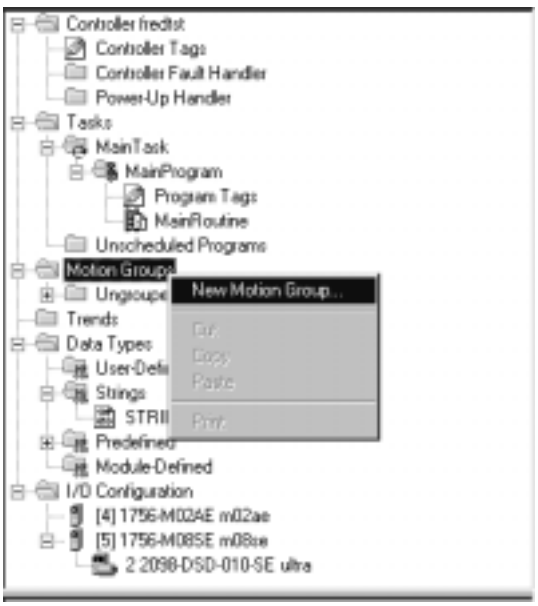
if you're using:	**then:**
digital, analog, or motion modules	counters are cleared
another module	the tab is refreshed but the counters are not cleared

The Motion Group

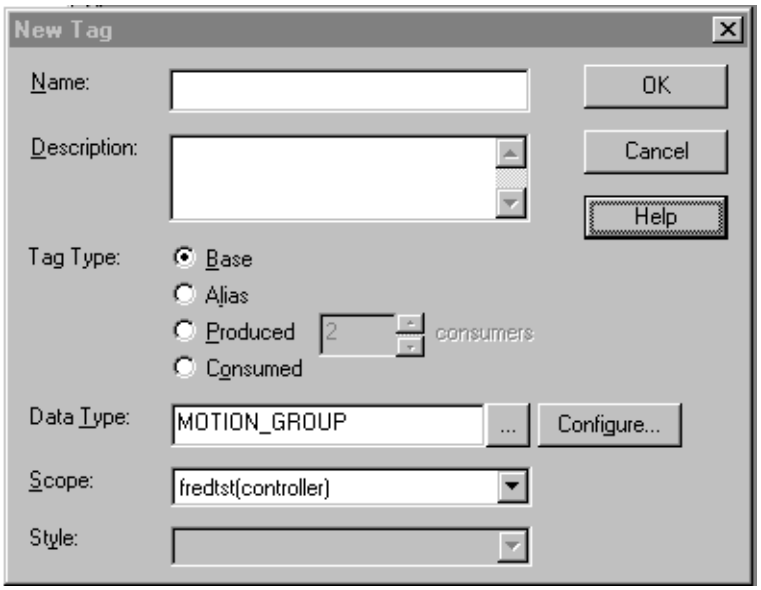
Creating A Motion Group

Each .acd program must have one motion group. (There can be only one.) You must create it before an axis can be assigned to the group and have it function within the .acd program.

To create the motion group, right click on Motion Group and select New Motion Group from the drop down menu.



This calls the New Tag window.



1. Enter a name for the Motion Group in the Name field.
2. In the Description field, enter a description of the tag.
3. Click on the respective radio button to select one of the following tag types:
   - Base - refers to a normal tag (selected by default)
   - Alias - refers to a tag, which references another tag with the same definition. Special parameters appear on the New Tag dialog that allows you to identify to which base tag the alias refers.
4. Select MOTION_GROUP for the Data Type.
5. From the Scope pull-down menu, select the scope for the tag.
6. If you want to produce this tag for other controllers to consume, check the Produce box and enter the maximum number of consumers.

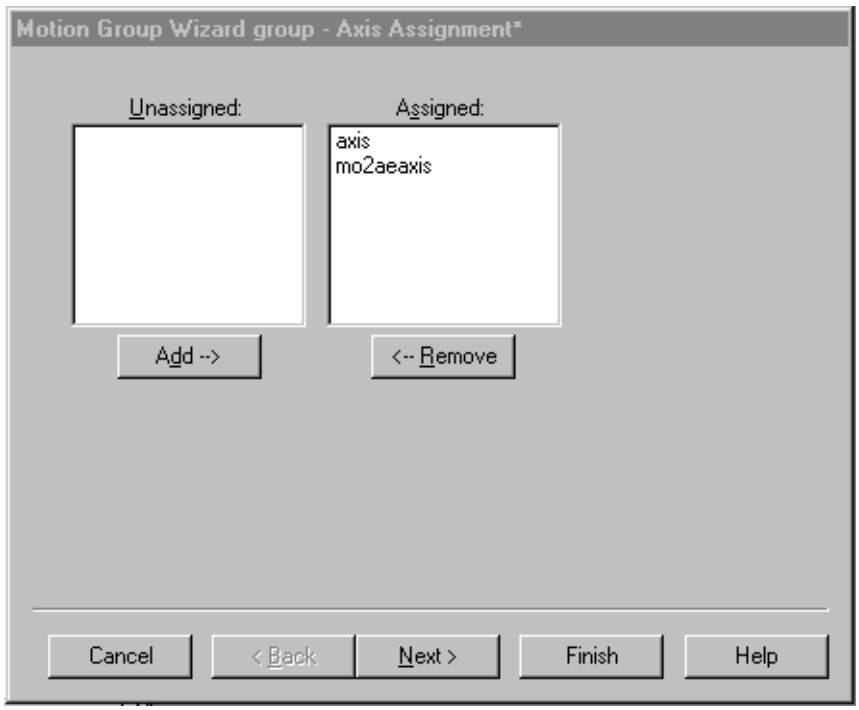
IMPORTANT Producing a tag requires a connection for each consumer. Connections are a limited resource in the controller, so only produce tags that you know you are needed in other controllers.

7. Click on OK to create the tag and close the New Tag dialog.

If there are errors, you will be prompted.

8. Click on the Configure button.

The Motion Group Wizard group - Axis Assignment screen displays.

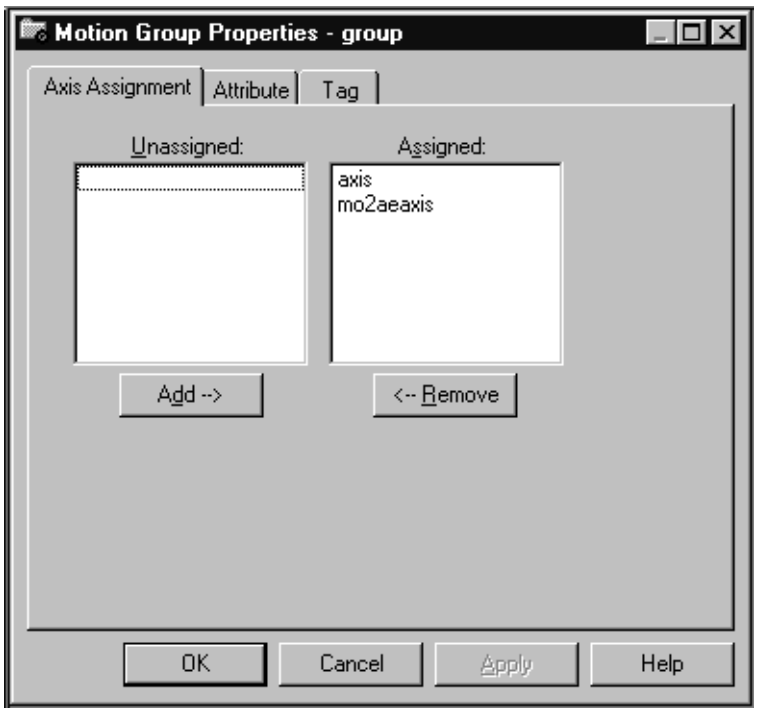


Add any existing axes to the group.

9. Continue on through the Motion Group Wizard to configure your Motion Group tag as necessary. Click on Finish>> to close the wizard.

Editing the Motion Group Properties

The Motion Group properties can be edited by right clicking on the group name and selecting Motion Group Properties from the drop down menu. The Motion Group Properties tabbed screen displays.



Axis Assignment Tab

The Axis Assignment screen is where axes are either assigned or unassigned to the Motion Group. When RSLogix 5000 software is online, all attributes on this dialog transition to a read-only state. When an attribute transitions to a read-only state, all pending attribute changes revert back to their offline status.

Unassigned

Lists the axes that are not assigned to any group in the controller.

Assigned

Lists the axes that are assigned to this motion group.

Add

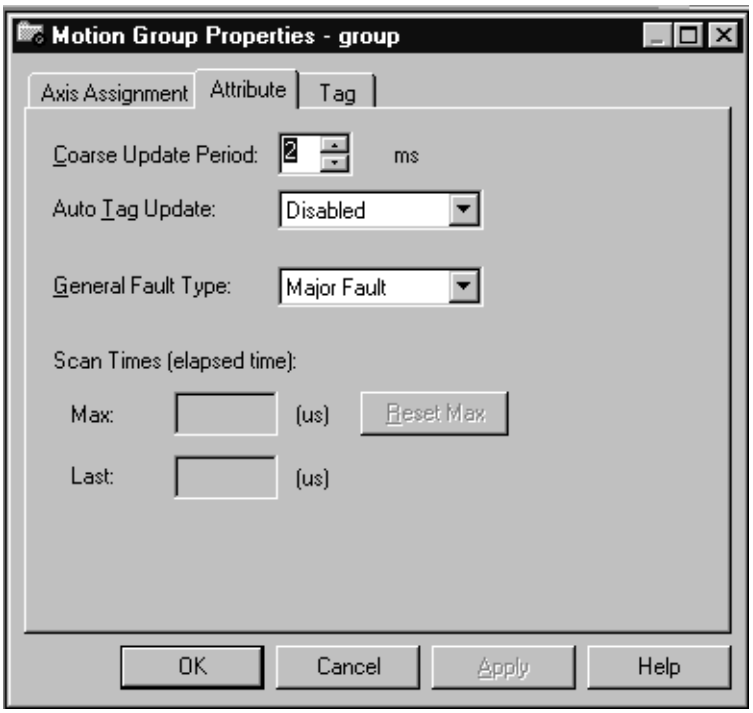
Click on this button to add axes to the motion group.

Remove

Click on this button to remove axes from the motion group.

Attribute Tab

The Attribute tab is used to modify the group attributes.



When RSLogix 5000 software is online, all of the attributes on this tab transition to a read-only state. When an attribute transitions to a read-only state, all pending attribute changes are reverted.

Coarse Update Period

Selects the periodic rate at which the motion task executes to compute the servo commanded position, velocity, and accelerations to be sent

to the 1756-M02AE or 1756-MO8SE modules when executing motion instructions.

TIP

If the coarse update period is too small, the controller may not have time to execute non-motion related ladder logic. As a general rule, one millisecond per axis is needed by the motion task for the controller to have reasonable execution time. As a result, the configuration sets the lower limit on the coarse update period based on the number of axes in the group.

Auto Tag Update

Determines whether axis parameter values will be automatically updated during operation. Choose from:

- Enabled – turns On automatic tag updating
- Disabled – turns Off automatic tag updating

General Fault Type

Selects the general fault type mechanism for the motion group. The available selections are:

- Non Major Fault – Any faults detected by the motion group will not cause the processor to fault. The application programmer needs to handle the fault in the program.
- Major Fault – Any faults detected by the motion group will cause the processor OK light to go blinking red and the fault routine to be invoked. If the fault routine handles the fault and clears it, then the OK light turns green. If the fault routine does not clear the fault, then the OK light becomes solid red and the processor stops executing the program.

Scan Times (elapsed time)

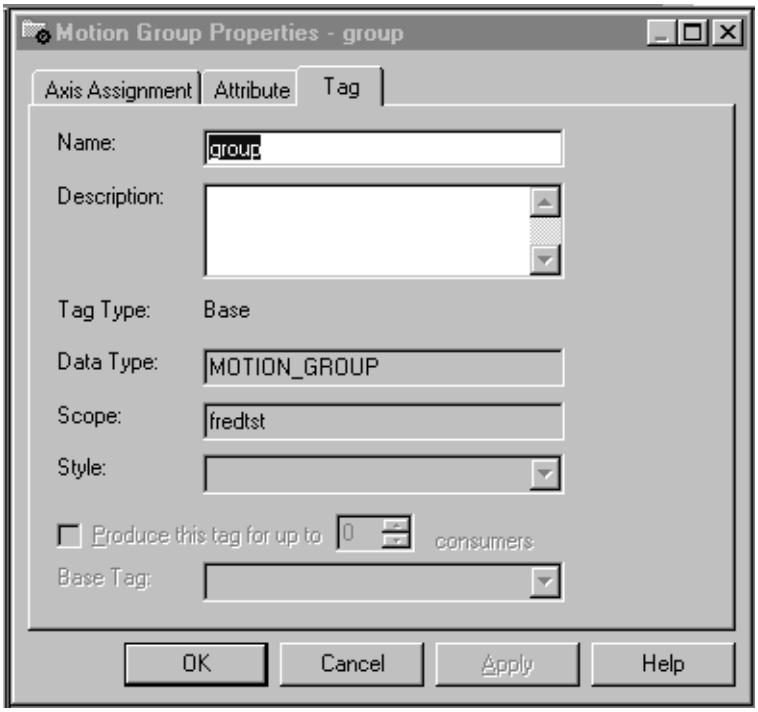
- Max – displays the value from the previous scan; clear this value, if necessary
- Disabled – displays the value from the previous scan

Reset Max

Click on this button to clear the Scan Times Max value.

Tag Tab

Use this tab to modify the name and description of the group.



When you are online, all of the parameters on this tab transition to a read-only state, and cannot be modified. If you go online before you save your changes, all pending changes revert to their previously-saved state.

Name

Enter the name of the motion group. This name must not exceed 40 characters. If you enter more than 40 characters, the system notifies you and it ignores the extra characters.

Description

Enter a description of the motion group. This description must not exceed 128 characters. If you enter more than 128 characters, the system notifies you and it ignores the extra characters.

Tag Type (read-only)

Displays the type of tag.

- Base - a normal tag
- Alias - a tag that references another tag with the same definition

Data Type (read-only)

The axis data type: MOTION_GROUP

Scope

Displays the scope of the current tag. The scope is either controller scope, or program scope, based on one of the existing programs in the controller.

Style

Not applicable to motion group tags.

Produce this tag for up to

A checked box indicates that this tag is available to remote controllers through controller-to-controller messaging. If this box is checked, the system displays the maximum number of consumers (i.e., connections) allowed for this tag.

The default number of consumers is 2.

Base Tag

If this tag is an alias, this field displays the name of the motion group tag on which this alias was based. The base tag actually defines the memory where the data element is stored.

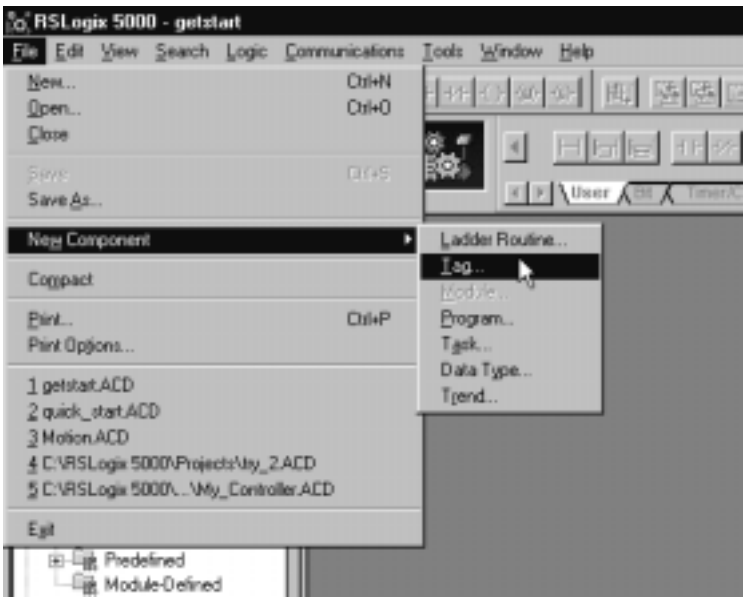
Naming & Configuring Your Motion Axis

This chapter describes how to name, configure, and edit your axis properties.

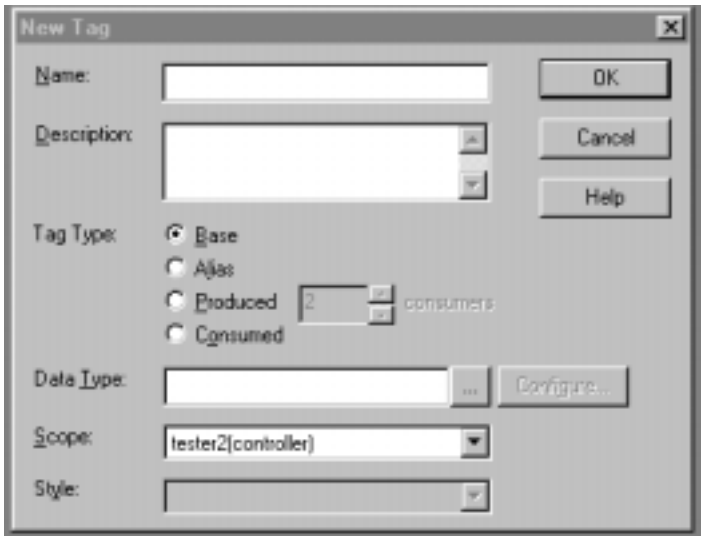
Naming an Axis

Naming an axis adds it to your application. To name an axis:

Go to the **File** pull-down menu, select **New Component**, and then select **Tag**.



The New Tag window appears.



Entering Tag Information

A tag allows you to allocate and reference data stored in the controller. A tag can be a simple, single element, or an array, or a structure. There are four types of tags that you can create:

- A base tag allows you to create your own internal data storage.
- An alias tag allows you to assign your own name to an existing tag, structure tag member, or bit.
- A produced tag lets you make the tag available to remote controllers through controller-to-controller messaging.
- A consumed tag allows you to retrieve data from a tag in another controller.

You must set up only one consumed tag to get data from the same producing tag in another controller.

WARNING

Setting up more than one consumed tag results in unpredictable controller to controller behavior.

Use this dialog to create new tags. The parameters that appear on this dialog depend upon the type of tag you are creating.

You can create base tags and alias tags while the controller is online or offline, as long as the new tag is verified. You can only create consumed tags while the controller is offline.

Common Parameters

The following parameters appear on the New Tag dialog whether you are creating a base tag, alias tag, or consumed tag.

Name

Enter the name of the tag you want to create.

Description

Enter a description of the tag.

Tag Type

Check the type of tag you are creating:

- Base – refers to a normal tag (selected by default)
- Alias – refers to a tag, which references another tag with the same definition. Special parameters appear on the New Tag dialog that allow you to identify to which base tag the alias refers.
- Produced – refers to a tag that has been made available to other controllers. If this type is chosen, then you can set the maximum number of consumers allowed for this tag.
- Consumed (only available when the controller is offline) – refers to a tag that is produced by another controller whose data you want to use in this controller. Special parameters appear on the New Tag dialog that allow you to identify from where the consumed tag is to come.

Data Type

In the Data Type field you can either enter the type of tag you want to create directly or click on the ellipsis button to go to the Select Data Type dialog. From this dialog you can select the appropriate axis data type: AXIS_CONSUMED, AXIS_SERVO, AXIS_SERVO_DRIVE, or AXIS VIRTUAL.

Make entries in the following fields.

Field	Entry	
Name	Type a name for the servo axis. The name can: have a maximum of 40 characters contain letters, numbers and underscores (_).	
Description	Type a description for your motion axis. This field is optional.	
Data type	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE, AXIS_VIRTUAL	
Scope	Select the scope of the axis variable.	
	To use the axis	**Select**
	Within the entire program	Controller

Editing Motion Axis Properties

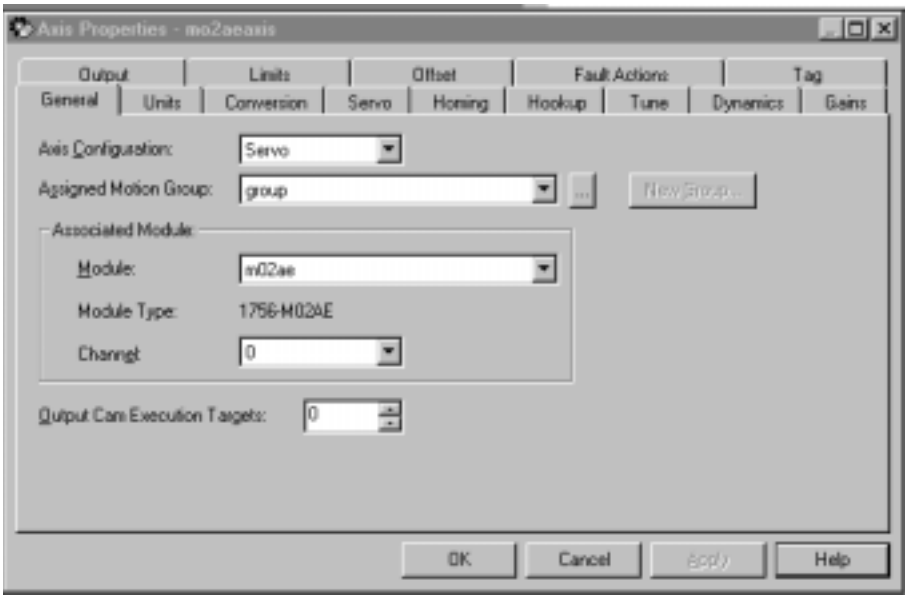
Once you have named your axis in the New Tag window, you must then configure it. You can make your configuring options in the Axis Properties screen. These have a series of Tabs that access a specific dialog for configuring the axis. Make the appropriate entries for each of the fields. An asterisk appears on the Tab to indicate changes have been made but not implemented. Press the Apply button at the bottom of each dialog to implement your selections.

TIP

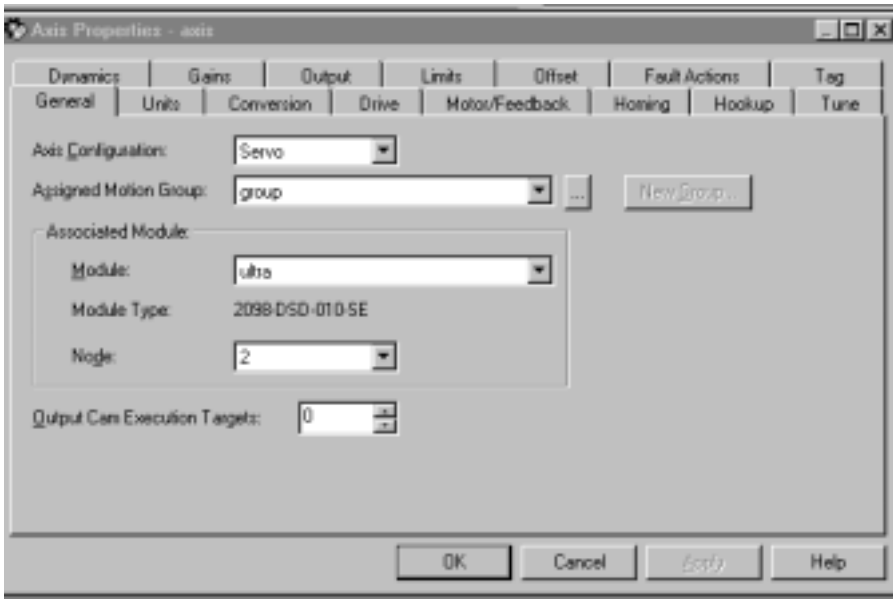
When you configure your axis, some fields may be unavailable (greyed-out) because of choices you made in the New Tag window.

In the Controller Organizer, right click on the axis to edit and select Axis Properties from the drop down menu. The Axis Properties

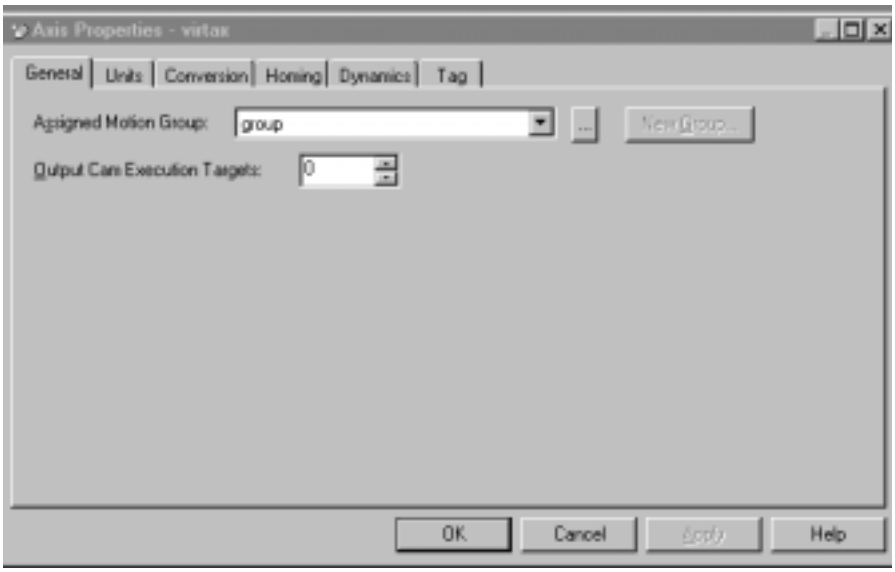
General window appears. The General screen depicted below is for an AXIS_SERVO data type.



The General screen shown below is for an AXIS_SERVO DRIVE Data Type.



The AXIS_VIRTUAL General Tab is shown below.



General Tab – SERVO_AXIS

Use this tab to do the following for an axis, of the data type AXIS_SERVO:

- Configure the axis for Servo operation, or for position Feedback Only.
- Assign the axis, or terminate the assignment of an axis, to a Motion Group.
- Associate the axis with a 1756-M02AE motion module.
- Select the channel, 0 or 1, on the 1756-M02AE motion module to which the axis is connected.

  Note: RSLogix 5000 supports only one Motion Group tag per controller.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Axis Configuration

Selects and displays the intended use of the axis:

- Feedback Only: If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters.
- Servo: If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.

Assigned Motion Group

Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motions Groups branch.

Ellipsis (…) button

Opens the Motion Group Properties dialog box for the Assigned Motion Group, where you can edit the properties of the Assigned Motion Group. If no Motion Group is assigned to this axis, this button is disabled.

New Group button

Opens the New Tag dialog box, where you can create a new Motion Group tag. This button is enabled only if no Motion Group tag has been created.

Module

Selects and displays the name of the motion module to which the axis is associated. Displays <none> if the axis is not associated with any motion module.

Module Type

This read-only field displays the type of motion module, if any, with which the axis is associated. An axis of the AXIS_SERVO data type can be associated only with 1756-M02AE motion modules. Displays <none> if the axis is not associated with any motion module.

Channel

Selects and displays the 1756-M02AE motion module channel - either 0 or 1 - to which the axis is assigned. Disabled when the axis is not associated with any motion module.

Output Cam Execution Targets

Determines how many Output Cam execution nodes (instances) are created for a specific axis. Note that the Execution Target parameter for the MAOC/MDOC instructions specify which of the configured execution nodes the instruction is affecting. In addition, the number specified in the Axis Properties dialog specifies the number of instances of Output Cam in which the value of zero means “none”, and the value specified for Execution Target in the MAOC instruction references a specific instance in which a value of zero selects the first instance.

General Tab - AXIS_SERVO_DRIVE

Use this tab to do the following for an axis, of the data type AXIS_SERVO_DRIVE:

- Configure the axis for Servo operation, or for position Feedback Only.
- Assign the axis, or terminate the assignment of an axis, to a Motion Group.
- Associate the axis with a 1756-M08SE motion module.
- View the base node of the associated 1756- M08SE motion module.

  Note: RSLogix 5000 supports only one Motion Group tag per controller.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Axis Configuration

Selects and displays the intended use of the axis:

- Feedback Only: If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters.
- Servo: If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.

Assigned Motion Group

Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motions Groups branch.

Ellipsis (…) button

Opens the Motion Group Properties dialog box for the Assigned Motion Group, where you can edit the properties of the Assigned Motion Group. If no Motion Group is assigned to this axis, this button is disabled.

New Group button

Opens the New Tag dialog box, where you can create a new Motion Group tag. This button is enabled only if no Motion Group tag has been created.

Module

Selects and displays the name of the motion module to which the axis is associated. Displays <none> if the axis is not associated with any motion module.

Module Type

This read-only field displays the type of motion module, if any, with which the axis is associated. An axis of the AXIS_SERVO_DRIVE data type can be associated only with 1756- M08SE motion modules. Displays <none> if the axis is not associated with any motion module.

Node

Displays the base node of the associated 1756- M08SE motion module. Disabled when the axis is not associated with any motion module.

Output Cam Execution Targets

Determines how many Output Cam execution nodes (instances) are created for a specific axis. Note that the Execution Target parameter for the MAOC/MDOC instructions specify which of the configured execution nodes the instruction is affecting. In addition, the number specified in the Axis Properties dialog specifies the number of instances of Output Cam in which the value of zero means "none", and the value specified for Execution Target in the MAOC instruction references a specific instance in which a value of zero selects the first instance.

General Tab - AXIS_VIRTUAL

Use this tab to associate the axis, of the data type AXIS_VIRTUAL, to a Motion Group.

Note: RSLogix 5000 supports only one Motion Group tag per controller.

When RSLogix 5000 software is online, the parameters on this tab transition to a read-only state. When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Assigned Motion Group

Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motions Groups branch.

Ellipsis (…) button

Opens the Motion Group Properties dialog box for the Assigned Motion Group, where you can edit the properties of the Assigned Motion Group. If no Motion Group is assigned to this axis, this button is disabled.

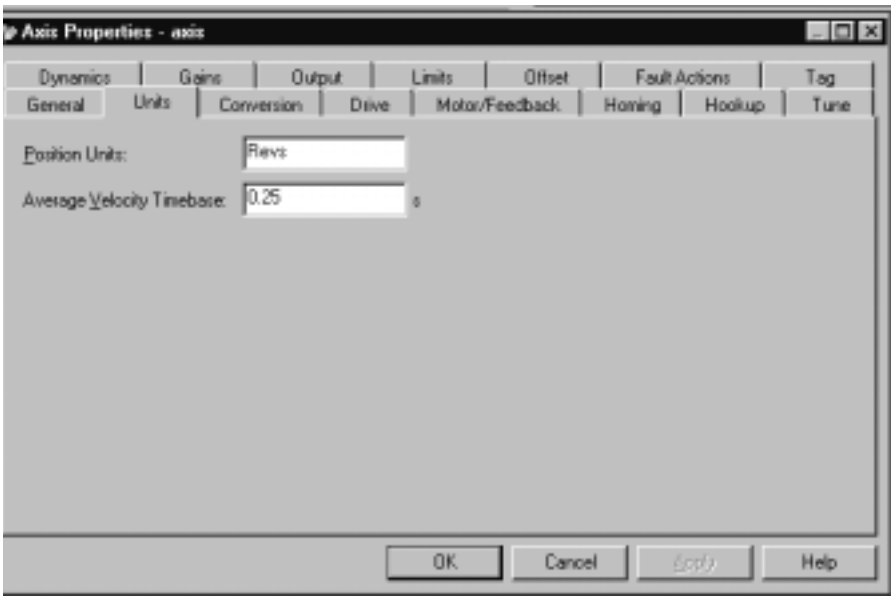
New Group button

Opens the New Tag dialog box, where you can create a new Motion Group tag. This button is enabled only if no Motion Group tag has been created.

Output Cam Execution Targets

Determines how many Output Cam execution nodes (instances) are created for a specific axis. Note that the Execution Target parameter for the MAOC/MDOC instructions specify which of the configured execution nodes the instruction is affecting. In addition, the number specified in the Axis Properties dialog specifies the number of instances of Output Cam in which the value of zero means "none", and the value specified for Execution Target in the MAOC instruction references a specific instance in which a value of zero selects the first instance.

Press **Apply** then select the Units tab to access the Axis Properties Units dialog.



Units Tab

The Units Tab is the same for all axis data types. Use this tab to determine the units to define your motion axis.

When RSLogix 5000 software is online and the controller transitions to hard run, or the servo loop is on (i.e., active), then all the attributes on this tab transition to a read only state. When any attribute transitions to a read only state, then any pending attribute changes are reverted.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Position Units

User-defined engineering units (rather than feedback counts) used for labeling all motion-related values (e.g., position, velocity, etc.) These position units can be different for each axis.

> **Note:** Position Units should be chosen for maximum ease of use in your application. For example, linear axes might use position units of Inches, Meters, or mm whereas rotary axes might use units of Revs or Degrees.

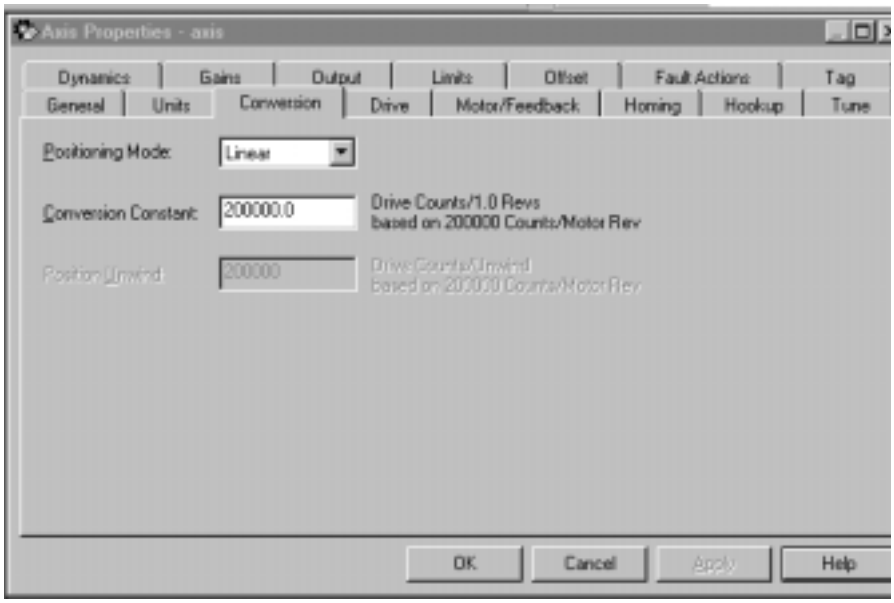
Average Velocity Timebase

Specifies the time (in seconds) to be used for calculating the average velocity of the axis. This value is computed by taking the total distance the axis travels in the amount of time specified, and dividing this value by the timebase.

The average velocity timebase value should be large enough to filter out the small changes in velocity that would result in a "noisy" velocity value, but small enough to track significant changes in axis velocity. A value of 0.25 to 0.50 seconds should work well for most applications.

Click on the Apply button to accept your changes.

Click on the Conversion Tab to access the Axis Properties Conversion dialog.



Conversion Tab

Use this tab to view the Positioning Mode and configure the Feedback Resolution for an axis, of the tag types AXIS_SERVO, AXIS_SERVO_DRIVE and AXIS_VIRTUAL.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Positioning Mode

This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is set in and taken from a producing axis in a networked Logix processor. The values that can be passed are:

- Linear - provides a maximum total linear travel of 1 billion feedback counts. With this mode, the unwind feature is disabled and you can limit the linear travel distance traveled by the axis by specifying the positive and negative travel limits for the axis.

- Rotary - enables the rotary unwind capability of the axis. This feature provides infinite position range by unwinding the axis position whenever the axis moves through a complete unwind distance. The number of encoder counts per unwind of the axis is specified by the Position Unwind parameter.

Conversion Constant

Type the number of feedback counts per position unit. This conversion – or "K" – constant allows axis position to be displayed, and motion to be programmed, in the position units set in the Units tab. The conversion constant is used to convert axis position units into feedback counts and vice versa.

Note: For axes of the type AXIS_SERVO_DRIVE, the label will indicate the number of counts per motor revolution, as set in the Drive Resolution field of the Drive tab.

Position Unwind

This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is set in and taken from a producing axis in a networked Logix processor. For a Rotary axis, this value represents the distance (in feedback counts) used to perform automatic electronic unwind. Electronic unwind allows infinite position range for rotary axes by subtracting the unwind distance from both the actual and command position, every time the axis travels the unwind distance.

Note: For axes of the type AXIS_SERVO_DRIVE:

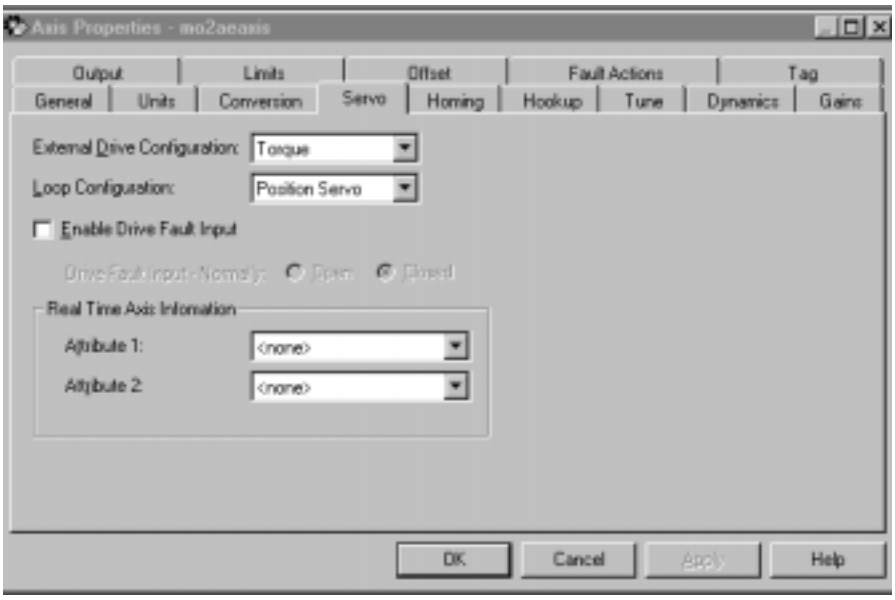
- when you save an edited Conversion Constant or a Drive Resolution value, a message box appears, asking you if you want the controller to automatically recalculate certain attribute settings. (Refer to Conversion Constant and Drive Resolution Attributes.)

- the label will indicate the number of counts per motor revolution, as set in the Drive Resolution field of the Drive tab.

Click on Apply to accept your changes.

Servo Tab - AXIS_SERVO

Click on the Servo Tab from the Axis Properties for SERVO_AXIS to access the Servo dialog.



For an axis of the data type AXIS_SERVO, configured for Servo operation in the General tab of this dialog box, use the SERVO tab to:

- configure an external drive
- configure the drive fault input
- select up to two axis attributes whose status can be monitored

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

External Drive Configuration

Select the drive type for the servo loop:

- Velocity - disables the servo module's internal digital velocity loop.
- Torque - the servo module's internal digital velocity loop is active, which is the required configuration for interfacing the servo axis to a torque loop servo drive.

Loop Configuration

Select the configuration of the servo loop. For this release, only Position Servo is available.

Enable Drive Fault Input

Check this box if you wish to enable the Drive Fault detection. When the drive fault is detected, appropriate action is taken based on the Drive Fault Action specified in the Fault Actions tab of this dialog box.

Drive Fault Input

Specifies the usual state of the drive fault input – Normally:

- Open
- Closed

Real Time Axis Information

Attribute 1/Attribute 2

Select up to two axis attributes whose status will be transmitted – along with the actual position data – to the Logix processor. The values of the selected attributes can be accessed via the standard GSV or Get Attribute List service.

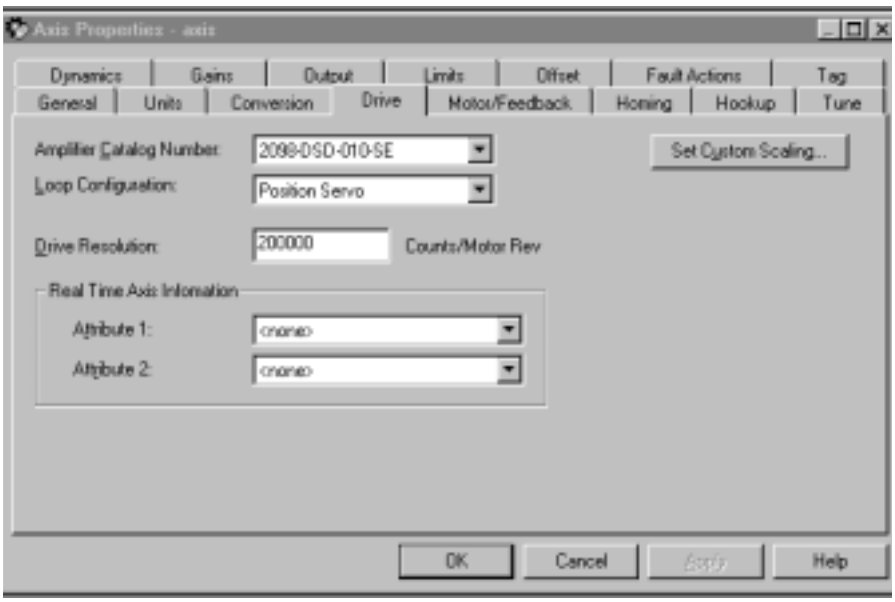
> **Note:** The servo status data update time is precisely the coarse update period.

If a GSV is done to one of these servo status attributes without having selected this attribute via the Drive Info Select attribute, the attribute value will be static and will not reflect the true value in the servo module.

Click on the Apply button to accept your changes.

Drive Tab - (AXIS_SERVO_DRIVE)

Use this tab to configure the servo loop for an AXIS_SERVO_DRIVE axis, and open the Custom Drive Scaling Attributes dialog box.



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

Amplifier Catalog Number

Select the catalog number of the amplifier to which this axis is connected.

Loop Configuration

Select the configuration of the servo loop:

- Motor Feedback Only
- Aux Feedback Only
- Position Servo
- Auxiliary Position Servo (not applicable to Ultra3000 drives)
- Dual Position Servo (not applicable to Ultra3000 drives)
- Motor Dual Command Servo
- Auxiliary Dual Command Servo
- Velocity Servo
- Torque Servo

Drive Resolution

Type in the number of counts per motor revolution. This value will apply to all position data. Valid values range from 1 to 2^32 - 1. One Least Significant Bit (LSB) for position data equals 360° / Rotational Position Resolution.

> **Note:** Drive Resolution is also referred to as Rotational Position Resolution.

When you save an edited Conversion Constant or a Drive Resolution value, a message box appears, asking you if you want the controller to automatically recalculate certain attribute settings. (Refer to Conversion Constant and Drive Resolution Attributes for a list of these attributes.)

Drive Resolution is especially helpful for either fractional unwind applications or multi-turn applications requiring cyclic compensation. You can modify the Drive Resolution value so that dividing it by the Unwind Value yields a whole integer value. The higher the Drive Resolution setting, the finer the resolution.

Real Time Axis Information

Attribute 1/Atrribute 2

Select up to two axis attributes whose status will be transmitted – along with the actual position data – to the Logix processor. The values of the selected attributes can be accessed via the standard GSV or Get Attribute List service.

> **Note:** The servo status data update time is precisely the coarse update period.

If a GSV is done to one of these servo status attributes without the having selected this attribute via the Drive Info Select attribute, the attribute value will be static and will not reflect the true value in the servo module.

Set Custom Scaling…button

Opens the Custom Drive Scaling Attributes dialog, where you can read scaling-related attributes for the axis.

Motor/Feedback Tab - AXIS_SERVO_DRIVE

Use this tab to configure motor and auxiliary feedback device (if any) parameters, for an axis of the type AXIS_SERVO_DRIVE.

> **Note:** The Axis Configuration selection made on the General tab, and the Loop Configuration selection made on the Drive tab determine which sections of this dialog box – Motor and Auxiliary Feedback – are enabled.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

(Motor) Catalog Number

Select the catalog number of the motor associated with this axis. When you change a Motor Catalog Number, the controller recalculates the values of the following values using (among other values) the default Damping Factor of 0.8.

On this tab or dialog:	These attributes are recalculated:
Motor/Feedback tab	Motor Feedback Type Motor Feedback Resolution
Gains tab	Position Proportional Gains Velocity Proportional Gains
Dynamics tab	Maximum Velocity Maximum Acceleration Maximum Deceleration
Limits tab	Position Error Tolerance
Custom Stop Action Attributes dialog	Stopping Torque
Custom Limit Attributes dialog	Velocity Limit Bipolar Velocity Limit Positive Velocity Limit Negative Acceleration Limit Bipolar Acceleration Limit Positive Acceleration Limit Negative Torque Limit Bipolar Torque Limit Positive Torque Limit
Tune Bandwidth dialog	Position Loop Bandwidth Velocity Loop Bandwidth

> **Note:** The Associated Module selection (selected on the General tab), determines available catalog numbers.

(Motor) Feedback Type

This read-only field displays the type of feedback associated with the selected motor (in the Catalog Number field, above). If you selected <None> as the Catalog Number, you must select a Feedback Type.

(Motor) Interpolation Factor

This field displays a fixed, read-only value for each feedback type. This value is used to compute the resolution of the feedback device.

(Motor) Cycles

The number of cycles per revolution of the associated feedback device. This helps the Drive Compute Conversion constant used to convert drive units to feedback counts. Depending on the feedback type you select, this value may be either read-only or editable.

(Auxiliary Feedback)Type

For applications that use auxiliary feedback devices, select the type of auxiliary feedback device type.

> **Note:** The Auxiliary Feedback section of this tab is enabled only if the Drive tab's Loop Configuration field is set to Aux Feedback Only, Aux Position Servo, or Aux Dual Command Servo.

Feedback Type	Description
SRS60	Stegmann Single Turn, Absolute 1024 S-C/Rev (Rotary)
SRM60	Stegmann Multi-Turn, Absolute 1024 S-C/Rev (Rotary)
SCS60	Stegmann Single Turn, Absolute 512 S-C/Rev (Rotary)
SCM60	Stegmann Multi Turn, Absolute 512 S-C/Rev (Rotary)
SNS60	Stegmann High Res Incremental, 1024 S-C/Rev (Rotary)
MHG90	Stegmann Encoder
Resolver	0.25 TR Resolver
Analog Reference	Differential Analog
Sin/Cos	Generic S-C Device w/Z marker
TTL	Generic AQB Device w/Z marker
UVW	Differential Hall Effect
Unknown Stegmann	Unrecognized Stegmann
Endat	Heidenhain Endat SSI Encoder
RCM21S-4	Renco Smart Encoder 4 Pole

Feedback Type	Description
RCM21S-6	Renco Smart Encoder 6 Pole
RCM21S-8	Renco Smart Encoder 8 Pole
LINCODER	5mm Cycle Linear Hiperface Magnetic Encoder

(Aux) Interp Factor

This field displays a fixed constant value for the selected feedback type. This value is used to compute the resolution of the feedback device.

(Aux) Cycles

The number of cycles per revolution of the auxiliary feedback device. This helps the Drive Compute Conversion constant used to convert drive units to feedback counts. Depending on the feedback type selected, this value may either be read-only or editable.

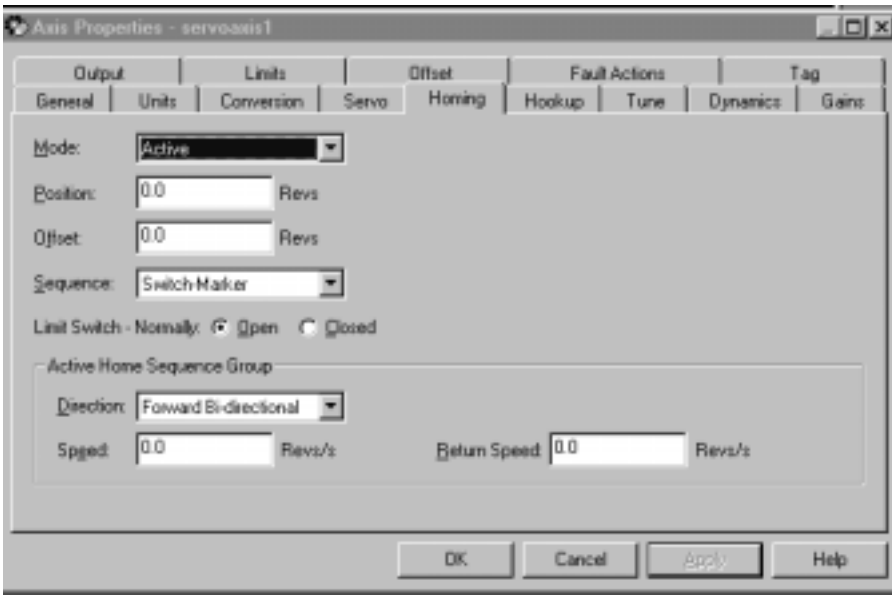
(Aux) Ratio

Represents the quantitative relationship between the auxiliary feedback device and the motor.

Homing Tab - SERVO_AXIS and SERVO_AXIS_DRIVE

Use this tab to configure the attributes related to homing an axis of the type AXIS_SERVO or AXIS_SERVO_DRIVE.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.



Mode

Select the homing mode:

- Active: In this mode, the desired homing sequence is selected by specifying whether a home limit switch and/or the encoder marker is used for this axis. Active homing sequences always use the trapezoidal velocity profile.

- Passive: In this mode, homing redefines the absolute position of the axis on the occurrence of a home switch or encoder marker event. Passive homing is most commonly used to calibrate uncontrolled axes, although it can also be used with controlled axes to create a custom homing sequence. Passive homing, for a given home sequence, works similar to the corresponding active homing sequence, except that no motion is commanded; the controller just waits for the switch and marker events to occur.

- Absolute: (AXIS_SERVO_DRIVE only) In this mode, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. The only valid Home Sequence for an absolute Homing Mode is immediate.

Position

Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position will be set to zero, although any value within the

software travel limits can be used. After the homing sequence is complete, the axis is left in this position.

If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.

Offset

Type the desired offset (if any) in position units the axis is to move, upon completion of the homing sequence, to reach the home position. In most cases, this value will be zero.

Sequence

Select the event that will cause the Home Position to be set:

Sequence Type:	**Description:**
Immediate	Sets the Home Position to the present actual position, without motion.
Switch	Sets the Home Position when axis motion encounters a home limit switch.
Marker	Sets the Home Position when axis encounters an encoder marker.
Switch-Marker	Sets the Home Position when axis first encounters a home limit switch, then encounters an encoder marker.

> **Note:** See the section “Homing Configurations,” below, for a detailed description of each combination of homing mode, sequence and direction.

Limit Switch

If a limit switch is used, indicate the normal state of that switch (i.e., before being engaged by the axis during the homing sequence):

- Normally Open
- Normally Closed

Direction

For active homing sequences, except for the Immediate Sequence type, select the desired homing direction:

Direction	Description
Forward Uni-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
Forward Bi-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).
Reverse Uni-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
Reverse Bi-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).

Speed

Type the speed of the jog profile used in the first leg of an active homing sequence. The homing speed specified should be less than the maximum speed and greater than zero.

Return Speed

The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.

Homing Configurations

The following examples of Active and Passive homing assume that the initial motion, if any, is in a positive axial direction. Click on an individual homing configuration for more information.

- Active Homing Configurations
- Active Immediate Home
- Active Bi-directional Home with Switch
- Active Bi-directional Home with Marker
- Active Bi-directional Home with Switch then Marker
- Active Uni-directional Home with Switch
- Active Uni-directional Home with Marker
- Active Uni-directional Home with Switch then Marker
- Passive Homing Configurations
- Passive Immediate Home
- Passive Home with Switch
- Passive Home with Marker
- Passive Home with Switch then Marker

Homing Tab - AXIS_VIRTUAL

Use this tab to configure the attributes related to homing an axis of the type AXIS_VIRTUAL.

Only an Active Immediate Homing sequence can be performed for an axis of the type AXIS_VIRTUAL. When this sequence is performed, the controller immediately enables the servo drive and assigns the Home Position to the current axis actual position and command position. This homing sequence produces no axis motion.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

Mode

This read-only parameter is always set to Active.

Position

Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position will be set to zero, although any value within the

software travel limits can be used. After the homing sequence is complete, the axis is left at this position.

If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.

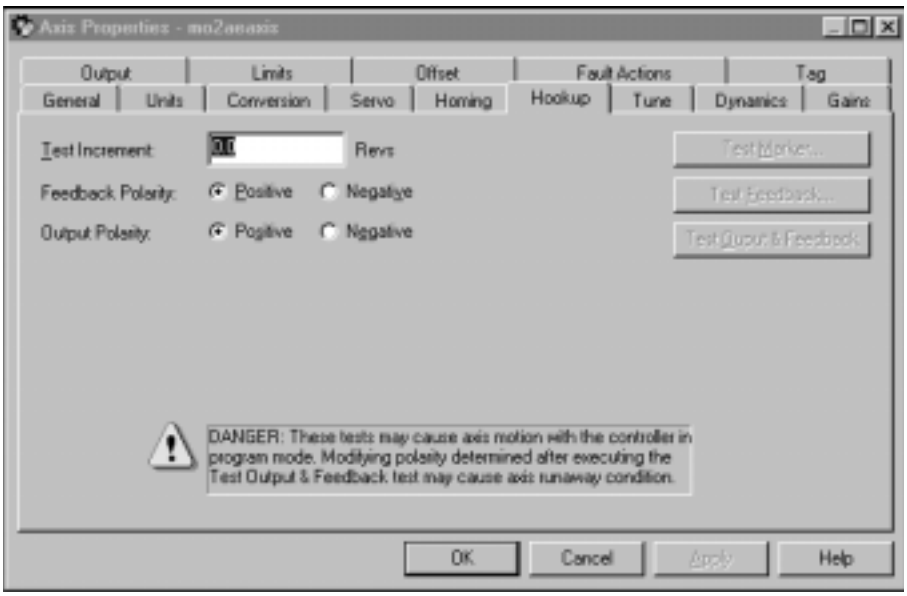
Sequence

This read-only parameter is always set to Immediate

Hookup Tab - AXIS_SERVO

Use this tab to configure and initiate axis hookup and marker test sequences for an axis of the type AXIS_SERVO.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.



Test Increment

Specifies the amount of distance traversed by the axis when executing the Output & Feedback test. The default value is set to approximately a quarter of a revolution of the motor in position units.

Feedback Polarity

The polarity of the encoder feedback, this field is automatically set by executing either the Feedback Test or the Output & Feedback Test:

- Positive
- Negative

Note: When properly configured, this setting insures that axis Actual Position value increases when the axis is moved in the user defined positive direction. This bit can be configured automatically using the MRHD and MAHD motion instructions.

WARNING

Modifying automatically input polarity values by running the Feedback or Output & Feedback Tests can cause a runaway condition.

Output Polarity

The polarity of the servo output to the drive, this field is automatically set by executing the Output & Feedback Test:

- Positive
- Negative

Note: When properly configured, this setting and the Feedback Polarity setting insure that, when the axis servo loop is closed, it is closed as a negative feedback system and not an unstable positive feedback system. This bit can be configured automatically using the MRHD and MAHD motion instructions.

Test Marker

Runs the Marker test, which ensures that the encoder A, B, and Z channels are connected correctly and phased properly for marker detection. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Feedback

Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Output & Feedback

Runs the Output & Feedback Test, which checks and, if necessary, reconfigures both the polarity of encoder feedback (the Feedback Polarity setting) and the polarity of the servo output to the drive (the Output Polarity setting), for an axis configured for Servo operation in the General tab of this dialog box.

> **Note:** Executing any test operation automatically saves all changes to axis properties.

Hookup Tab Overview - AXIS_SERVO_DRIVE

Use this tab to configure and initiate axis hookup and marker test sequences for an axis of the type AXIS_SERVO_DRIVE.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

Test Increment

Specifies the amount of distance traversed by the axis when executing the Command & Feedback test. The default value is set to approximately a quarter of a revolution of the motor in position units.

Drive Polarity

The polarity of the servo loop of the drive, set by executing the Command & Feedback Test:

- Positive
- Negative

> **Note:** Proper wiring guarantees that the servo loop is closed with negative feedback. However there is no guarantee that the servo drive has the same sense of forward direction as the user for a given application. Negative Polarity inverts the polarity of

both the command position and actual position data of the servo drive. Thus, selecting either Positive or Negative Drive Polarity makes it possible to configure the positive direction sense of the drive to agree with that of the user. This attribute can be configured automatically using the MRHD and MAHD motion instructions.

WARNING

Modifying polarity values, automatically input by running the Command & Feedback Test, can cause a runaway condition.

Test Marker

Runs the Marker test, which ensures that the encoder A, B, and Z channels are connected correctly and phased properly for marker detection. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Feedback

Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

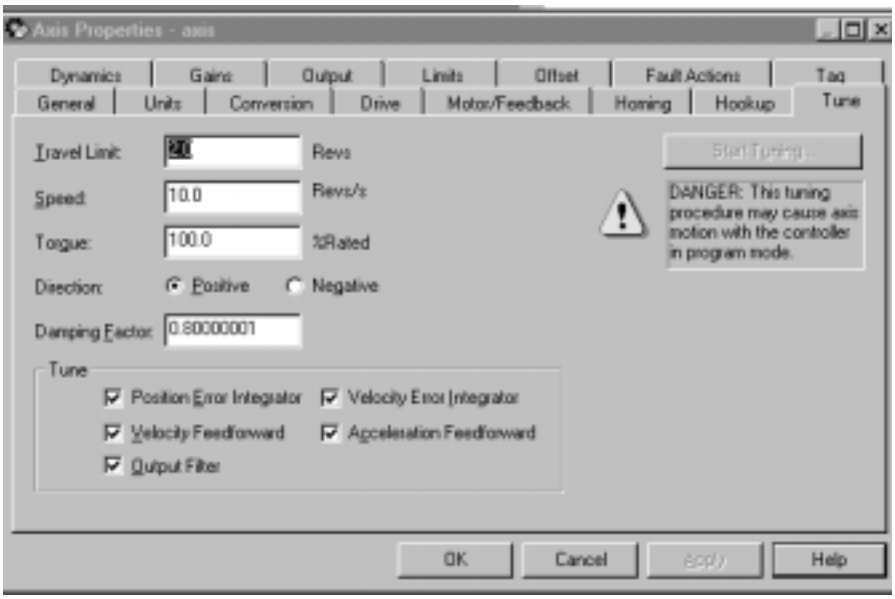
Test Output & Feedback

Runs the Output & Feedback Test, which checks and, if necessary, reconfigures both the polarity of encoder feedback (the Feedback Polarity setting) and the polarity of the servo output to the drive (the Output Polarity setting), for an axis configured for Servo operation in the General tab of this dialog box.

Note: Executing any test operation automatically saves all changes to axis properties.

Tune Tab - AXIS_SERVO, AXIS_SERVO_DRIVE

Use this tab to configure and initiate the axis tuning sequence for an axis of the types AXIS_SERVO or AXIS_SERVO_DRIVE.



Travel Limit

Specifies a limit to the excursion of the axis during the tune test. If the servo module determines that the axis will not be able to complete the tuning process before exceeding the tuning travel limit, it will terminate the tuning profile and report that this limit was exceeded.

Speed

Determines the maximum speed for the tune process. This value should be set to the desired maximum operating speed of the motor (in engineering units) prior to running the tune test.

Torque

The maximum torque of the tune test. This attribute should be set to the desired maximum safe torque level prior to running the tune test. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system.

> **Note:** In some cases a lower tuning torque limit value may be desirable to limit the stress on the mechanics during the tuning procedure. In this case the acceleration and deceleration capabilities of the system are extrapolated based on the ratio of

the tuning torque to the maximum torque output of the system. Extrapolation error increases as the Tuning Torque value decreases.

Direction

The direction of the tuning motion profile. Negative indicates the reverse direction; positive indicates the forward direction.

Damping Factor

Specifies the dynamic response of the servo axis. The default is set to 0.8. When gains are tuned using a small damping factor, a step response test performed on the axis may generate uncontrolled oscillation. The gains generated using a larger damping factor would produce a system step response that has no overshoot and is stable, but may be sluggish in response to changes.

> **Note:** The tuning procedure uses the Damping Factor that is set in this field. However, when the controller recalculates certain attributes in response to a Motor Catalog Number change (on the Motor/Feedback tab), the controller uses the default Damping Factor value of 0.8, and not a different value set in this field.

Tune

Select the gains to be determined by the tuning test:

- Position Error Integrator
- Velocity Feedforward
- Output Filter
- Velocity Error Integrator
- Acceleration Feedforward

Start Tuning

Click on this button to begin the tuning test. If the tuning process completes successfully the following attributes will be set.

On this tab:	These attributes are set:
Gains tab	Velocity Feedforward Gain (if checked under Tune, above) Acceleration Feedforward Gain (if checked under Tune, above) Position Proportional Gain Position Integral Gain (if checked under Tune, above) Velocity Proportional Gain Velocity Integral Gain (if checked under Tune, above)
Dynamics tab	Maximum Velocity Maximum Acceleration Maximum Deceleration
Output tab	Torque Scaling Velocity Scaling (AXIS_SERVO only) Low Pass Output Filter (see Note, below)
Limits	Position Error Tolerance

The Tune Bandwidth dialog opens for Servo drives, where you can "tweak" bandwidth values.

> **Note:** During tuning, if the controller detects a high degree of tuning inertia, it enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

Executing a Tune operation automatically saves all changes to axis properties.

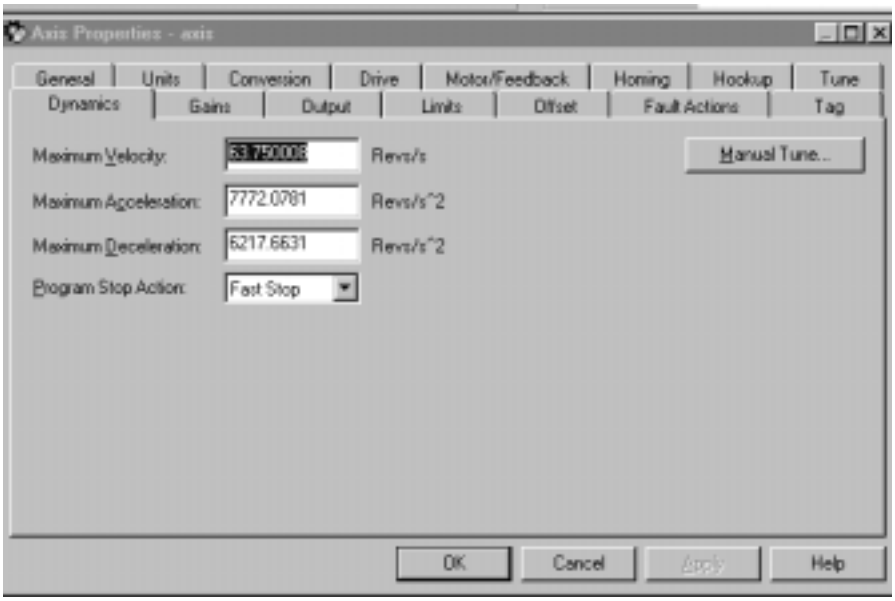
WARNING

This tuning procedure may cause axis motion with the controller in program mode.

Dynamics Tab

Use this tab to view or edit the dynamics related parameters for an axis of the type AXIS_SERVO or AXIS_SERVO_DRIVE configured for

Servo operations in the General tab of this dialog box, or AXIS_VIRTUAL.



IMPORTANT The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Maximum Velocity

The steady-state speed of the axis, it is initially set to Tuning Speed by the tuning process. This value is typically set to about 90% of the

maximum speed rating of the motor. This provides sufficient "head-room" for the axis to operate at all times within the speed limitations of the motor. Any change in value, caused by manually changing the spin control, is instantaneously sent to the controller.

Maximum Acceleration

The maximum acceleration rate of the axis, in Position Units/second, it is initially set to about 85% of the measured tuning acceleration rate by the tuning process. If set manually, this value should typically be set to about 85% of the maximum acceleration rate of the axis. This provides sufficient "

head-room" for the axis to operate at all times within the acceleration limits of the drive and motor. Any change in value, caused by manually changing the spin control, is instantaneously sent to the controller.

Maximum Deceleration

The maximum deceleration rate of the axis, in Position Units/second, it is initially set to about 85% of the measured tuning deceleration rate by the tuning process. If set manually, this value should typically be set to about 85% of the maximum deceleration rate of the axis. This provides sufficient "head-room" for the axis to operate at all times within the deceleration limits of the drive and motor. Any change in value, caused by manually changing the spin control, is instantaneously sent to the controller.

Program Stop Action

Select how a specific axis will stop when the processor undergoes a mode change, or when an explicit Motion Group Programmed Stop (MGPS) instruction is executed:

- **Fast Disable:** The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained until the axis motion has stopped at which time the axis is disabled (i.e., Drive Enable is disabled, and Servo Action is disabled).
- **Fast Shutdown:** The axis is decelerated to a stop using the current configured value for maximum deceleration. Once the axis motion is stopped, the axis is placed in the shutdown state (i.e., Drive Enable is disabled, Servo Action is disabled, and the OK contact is opened). To recover from this state, a reset instruction must be executed.
- **Fast Stop:** The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained after the axis motion has stopped. This mode is useful for gravity or loaded systems, where servo control is needed at all times.
- **Hard Disable:** The axis is immediately disabled (i.e. Drive Enable is disabled, Servo Action is disabled, but the OK contact is left closed). Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop.
- **Hard Shutdown:** The axis is immediately placed in the shutdown state. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. To recover from this state, a reset instruction must be executed.

Manual Tune

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

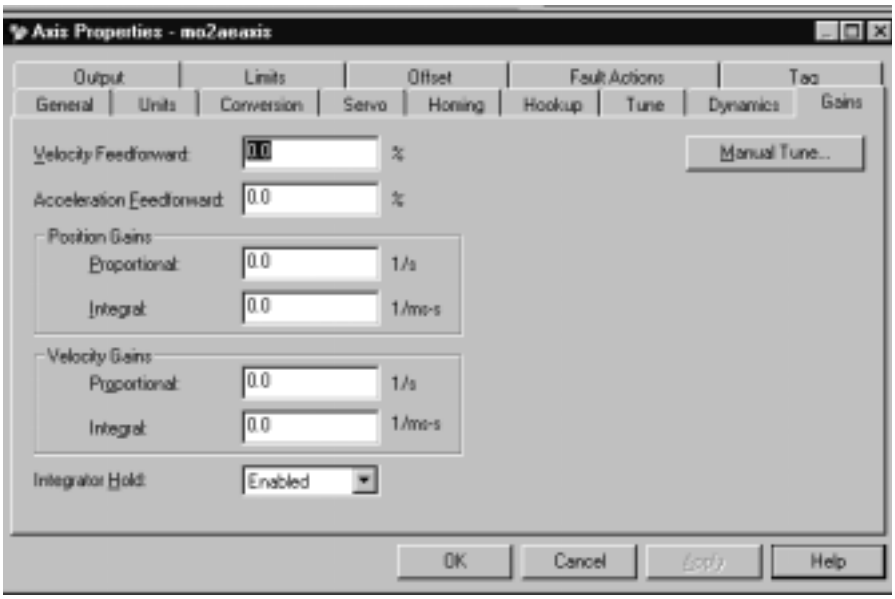
> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Gains Tab - AXIS_SERVO

Use this tab to perform the following offline functions:

- adjust, or “tweak” gain values that have been automatically set by the tuning process (in the Tune tab of this dialog)
- manually configure gains for the velocity and position loops

for an axis of the type AXIS_SERVO, which has been configured for Servo operations (set in the General tab of this dialog box), with Position Loop Configuration.



The drive module uses a nested digital servo control loop consisting of a position loop with proportional, integral and feed-forward gains around an optional digitally synthesized inner velocity loop. The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Velocity Feedforward

Velocity Feedforward Gain scales the current Command Velocity by the Velocity Feedforward Gain and adds it as an offset to the Velocity Command. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward

Acceleration Feedforward Gain scales the current Command Acceleration by the Acceleration Feedforward Gain and adds it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much of a contribution to the Servo Output, hence the Position and/or Velocity Error values are significantly reduced. Hence, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

> **Note:** Acceleration Feedforward Gain is not applicable for applications employing velocity loop servo drives. Such systems would require the acceleration feedforward functionality to be located in the drive itself.

Proportional (Position) Gain

Position Error is multiplied by the Position Loop Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain, on the other hand, can result in axis oscillation due to classical servo instability.

To set the gain manually, you must first set the appropriate output scaling factor (either the Velocity Scaling factor or Torque Scaling factor) in the Output tab of this dialog. Your selection of External Drive Configuration type – either Torque or Velocity – in the Servo tab of this dialog will determine which scaling factor you must configure before manually setting gains.

If you know the desired loop gain in inches per minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain:

Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil)

If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain:

Pos P Gain = Bandwidth (Hertz) / 6.28

The typical value for the Position Proportional Gain is ~100 Sec-1.

Integral (Position) Gain

The Integral (i.e., summation) of Position Error is multiplied by the Position Loop Integral Gain, or Pos I Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Pos I Gain improves the steady-state positioning performance of the system. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure (in the Tuning tab of this dialog), the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Output Scaling factor for the axis must be established for the drive system. Once this is done, the Pos I Gain can be computed based on the current or computed value for the Pos P Gain using the following formula:

Pos I Gain = .025 * 0.001 Sec/mSec * (Pos P Gain)2

Assuming a Pos P Gain value of 100 Sec-1 this results in a Pos I Gain value of 2.5 ~0.1 mSec-1 - Sec-1.

Proportional (Velocity) Gain

> **Note:** This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab of this dialog box.

Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Servo Output or Torque Command that ultimately attempts to correct for the velocity error, creating a damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. However, too much Velocity Proportional Gain leads to high frequency instability and resonance effects.

The typical value for the Velocity Proportional Gain is ~250 mSec-1.

Integral (Velocity) Gain

> **Note:** This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab of this dialog box.

At every servo update the current Velocity Error is accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Servo Output or Torque Command that attempts to correct for the velocity error. The higher the Vel I Gain value, the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability.

In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other, but not both. In general, where static positioning accuracy is required, Position Integral Gain is the better choice.

The typical value for the Velocity Proportional Gain is ~15 mSec-2.

Integrator Hold

If the Integrator Hold parameter is set to:

- Enabled, the servo loop temporarily disables any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.

- Disabled, all active position or velocity integrators are always enabled.

Manual Tune

Click on this button to access the Gains tab of the Manual Tune dialog for online editing.

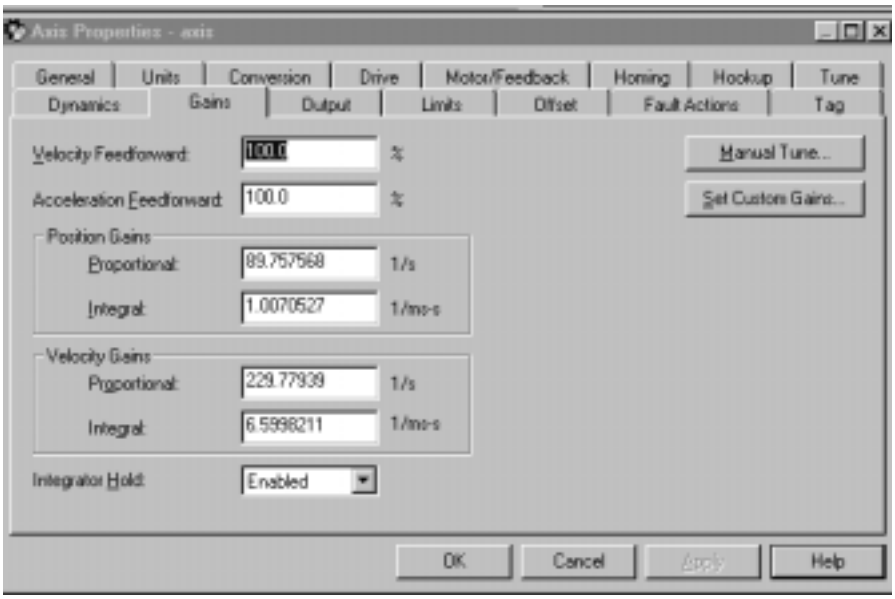
> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.

Gains Tab - AXIS_SERVO_DRIVE

Use this tab to perform the following offline functions:

- Adjust, or "tweak" gain values that have been automatically set by the tuning process (in the Tune tab of this dialog)
- Manually configure gains for the velocity and position loops

for an axis of the type AXIS_SERVO_DRIVE.



The drive module uses a nested digital servo control loop consisting of a position loop with proportional, integral and feed-forward gains around an optional digitally synthesized inner velocity loop. The specific design of this nested loop depends upon the Loop Configuration selected in the Drive tab of this dialog box. For a discussion, including a diagram, of a loop configuration, click on the following loop configuration types:

- Motor Position Servo Loop
- Auxiliary Position Servo Loop
- Dual Position Servo Loop
- Motor Dual Command Servo Loop
- Auxiliary Dual Command Servo Loop
- Velocity Servo Loop
- Torque Servo Loop

The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
-
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Velocity Feedforward

Velocity Feedforward Gain scales the current command velocity (derivative of command position) by the Velocity Feedforward Gain and adds it as an offset to the Velocity Command. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100%, theoretically. In reality, however, the value may need to be tweaked

to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward

Acceleration Feedforward Gain scales the current Command Acceleration by the Acceleration Feedforward Gain and adds it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much of a contribution to the Servo Output, hence the Position and/or Velocity Error values are significantly reduced. Hence, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

> **Note:** Acceleration Feedforward Gain is not applicable for applications employing velocity loop servo drives. Such systems would require the acceleration feedforward functionality to be located in the drive itself.

This value is also not applicable for Ultra3000 drives.

Proportional (Position) Gain

Position Error is multiplied by the Position Loop Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain, on the other hand, can result in axis oscillation due to classical servo instability.

> **Note:** To set the gain manually, you must first set the Torque scaling in the Output tab of this dialog.

If you know the desired loop gain in inches per minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain:

Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil)

If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain:

Pos P Gain = Bandwidth (Hertz) / 6.28

The typical value for the Position Proportional Gain is ~100 Sec-1.

Integral (Position) Gain

The Integral (i.e., summation) of Position Error is multiplied by the Position Loop Integral Gain, or Pos I Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Pos I Gain improves the steady-state positioning performance of the system. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure (in the Tuning tab of this dialog), the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Torque Scaling factor for the axis must be established for the drive system (in the Output tab of this dialog box). Once this is done, the Pos I Gain can be computed based on the current or computed value for the Pos P Gain using the following formula:

Pos I Gain = .025 * 0.001 Sec/mSec * (Pos P Gain)2

Assuming a Pos P Gain value of 100 Sec-1 this results in a Pos I Gain value of 2.5 ~0.1 mSec-1 - Sec-1.

Proportional (Velocity) Gain

> **Note:** This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab of this dialog box.

Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Torque Command that ultimately attempts to correct for the velocity error, creating a damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater

system stability. However, too much Velocity Proportional Gain leads to high frequency instability and resonance effects.

If you know the desired unity gain bandwidth of the velocity servo in Hertz, you can use the following formula to calculate the corresponding P gain.

Vel P Gain = Bandwidth (Hertz) / 6.28

The typical value for the Velocity Proportional Gain is ~250 mSec-1.

Integral (Velocity) Gain

Note: This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab of this dialog box.

At every servo update the current Velocity Error is accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Torque Command that attempts to correct for the velocity error. The higher the Vel I Gain value, the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability.

In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other, but not both. In general, where static positioning accuracy is required, Position Integral Gain is the better choice.

While the Vel I Gain, if employed, is typically established by the automatic servo tuning procedure (in the Tune tab of this dialog box), the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Torque Scaling factor for the axis must be established for the drive system, in the Output tab of this dialog box. Once this is done the Vel I Gain can be computed based on the current or computed value for the Vel P Gain using the following formula:

Vel I Gain = 0.25 * 0.001 Sec/mSec * (Vel P Gain)2

The typical value for the Velocity Proportional Gain is ~15 mSec-2.

Integrator Hold

If the Integrator Hold parameter is set to:

- Enabled, the servo loop temporarily disables any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.
- Disabled, all active position or velocity integrators are always enabled.

Set Custom Gains

Click on this button to open the Custom Gain Attributes dialog.

Manual Tune

Click on this button to access the Gains tab of the Manual Tune dialog for online editing.

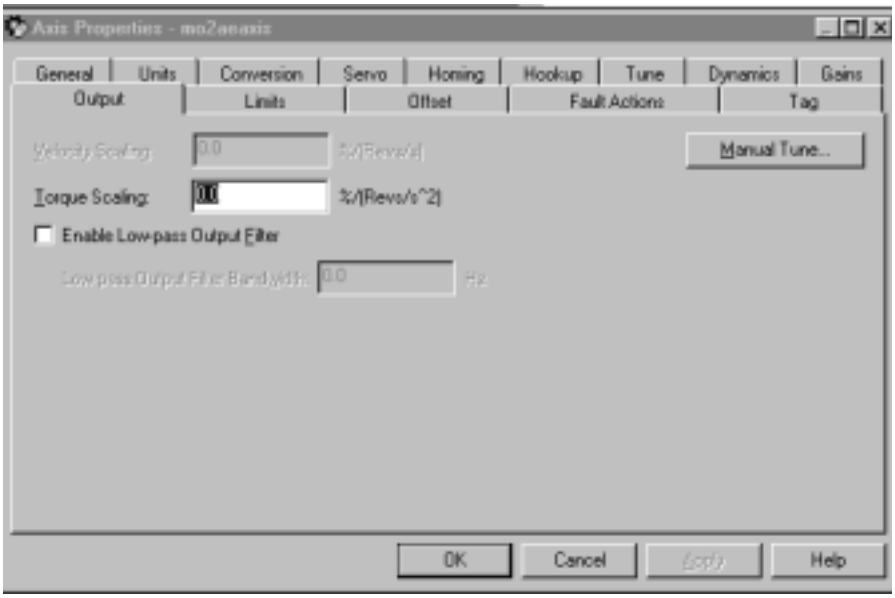
> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.

Output Tab - SERVO_AXIS

Use this dialog for offline configuration of:

- scaling values, which are used to generate gains, and
- the servo's low-pass digital output filter

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Velocity Scaling

The Velocity Scaling attribute is used to convert the output of the servo loop into equivalent voltage to an external velocity servo drive. This has the effect of “normalizing” the units of the servo loop gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, or mechanical gear ratios. The Velocity Scaling value is typically established by servo’s automatic

tuning procedure but these values can be calculated, if necessary, using the following guidelines.

If the axis is configured for a velocity external servo drive (in the Servo tab of this dialog), the software velocity loop in the servo module is disabled. In this case the Velocity Scaling value can be calculated by the following formula:

Velocity Scaling = 100% / (Speed @ 100%)

For example, if this axis is using position units of motor revolutions (revs), and the servo drive is scaled such that with an input of 100% (e.g. 10 Volts) the motor goes 5,000 RPM (or 83.3 RPS), the Velocity Scaling attribute value would be calculated as:

Velocity Scaling = 100% / (83.3 RPS) = 1.2% / Revs Per Second

Torque Scaling

The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of "normalizing" the units of the servo loops gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. The Torque Scaling value is typically established by the controller's automatic tuning procedure but the value can be manually calculated, if necessary, using the following guidelines:

Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque)

For example, if this axis is using position units of motor revolutions (revs), with 100% rated torque applied to the motor, if the motor accelerates at a rate of 3000 Revs/Sec2, the Torque Scaling attribute value would be calculated as shown below:

Torque Scaling = 100% Rated / (3000 RPS2) = 0.0333% Rated/ Revs Per Second2

Note: If the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains also does not reflect the true performance of the system.

Enable Low-pass Output Filter

Select this to enable the servo's low-pass digital output filter. De-select this to disable this filter.

Note: During tuning, if the controller detects a high degree of tuning inertia, it enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

Low-pass Output Filter Bandwidth

With Enable Low-pass Output Filter selected, this value sets the bandwidth, in Hertz, of the servo's low-pass digital output filter. Use this output filter to filter out high frequency variation of the servo module output to the drive. All output from the servo module greater than the Filter Bandwidth setting will be filtered-out, and not sent to the drive.

If the Low-pass Output Filter Bandwidth value is set to zero, the low-pass output filter is disabled. The lower the Filter Bandwidth value, the greater the attenuation of these high frequency components of the output signal. Because the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Filter Bandwidth value usually requires lowering the Position or Velocity Proportional Gain settings to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

Manual Tune

Click on this button to access the Gains tab of the Manual Tune dialog for online editing.

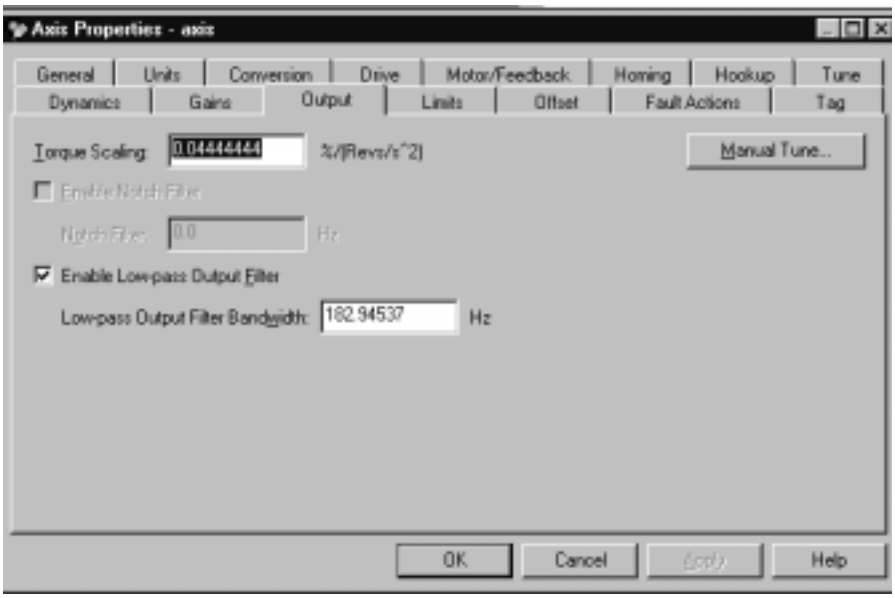
Note: The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.

Output Tab Overview - AXIS_SERVO_DRIVE

Use this dialog box to make the following offline configurations:

- set the torque scaling value, which is used to generate gains
- enable and configure the Notch Filter
- enable and configure servo's low-pass digital output filter

for an axis of the type AXIS_SERVO_DRIVE, configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Torque Scaling

The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of "normalizing" the units of the servo loops gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. The Torque Scaling value is typically established by the controller's automatic tuning procedure but the value can be manually calculated, if necessary, using the following guidelines:

Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque)

For example, if this axis is using position units of motor revolutions (revs), with 100% rated torque applied to the motor, if the motor accelerates at a rate of 3000 Revs/Sec2, the Torque Scaling attribute value would be calculated as shown below:

Torque Scaling = 100% Rated / (3000 RPS2) = 0.0333% Rated/ Revs Per Second2

Note: If the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains will also not reflect the true performance of the system.

Enable Notch Filter

Select this to enable the drive's notch filter. De-select this to disable this filter.

Notch Filter

With Enable Notch Filter selected, this value sets the center frequency of the drive's digital notch filter. If the Notch Filter value is set to zero, the notch filter is disabled.

Currently implemented as a 2nd order digital filter with a fixed Q, the Notch Filter provides approximately 40DB of output attenuation at the Notch Filter frequency. This output notch filter is particularly useful in attenuating mechanical resonance phenomena. The output filter is particularly useful in high inertia applications where mechanical resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

Note: This value is not applicable for Ultra3000 drives.

Enable Low-pass Output Filter

Select this to enable the servo's low-pass digital output filter. De-select this to disable this filter.

Note: During tuning, if the controller detects a high degree of tuning inertia, the controller enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

Low-pass Output Filter Bandwidth

With Enable Low-pass Output Filter selected, this value sets the bandwidth, in Hertz, of the servo's low-pass digital output filter. Use this output filter to filter out high frequency variation of the servo module output to the drive. All output from the servo module greater than the Filter Bandwidth setting will be filtered-out, and not sent to the drive.

If the Low-pass Output Filter Bandwidth value is set to zero, the low-pass output filter is disabled. The lower the Filter Bandwidth value, the greater the attenuation of these high frequency components of the output signal. Because the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Filter Bandwidth value usually requires lowering the Position or Velocity Proportional Gain settings to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

Manual Tune

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

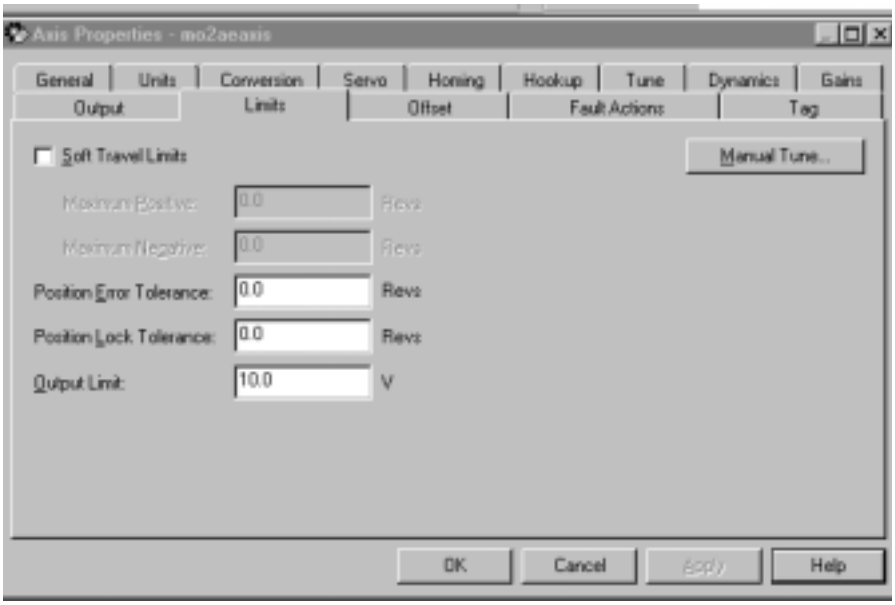
> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Limits Tab - AXIS_SERVO

Use this tab to make the following offline configurations:

- enable and set maximum positive and negative software travel limits, and
- configure both Position Error Tolerance and Position Lock Tolerance, and
- set the servo drive's Output Limit

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits

- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Soft Travel Limits

Enables software overtravel checking for an axis when Positioning Mode is set to Linear (in the Conversion tab of this dialog). If an axis is configured for software overtravel limits and if that axis passes beyond these maximum travel limits (positive or negative), a software overtravel fault is issued. The response to this fault is specified by the Soft Overtravel setting (in the Fault Actions tab of this dialog). Software overtravel limits are disabled during the tuning process.

Maximum Positive

Type the maximum positive position to be used for software overtravel checking, in position units.

> **Note:** The Maximum Positive limit must always be greater than the Maximum Negative limit.

Maximum Negative

Type the maximum negative position to be used for software overtravel checking, in position units.

> **Note:** The Maximum Negative limit must always be less than the Maximum Positive limit.

Position ErrorTolerance

Specifies how much position error the servo will tolerate before issuing a position error fault. This value is interpreted as a +/- quantity.

For example, setting Position Error Tolerance to 0.75 position units means that a position error fault will be generated whenever the position error of the axis is greater than 0.75 or less than -0.75 position units, as shown here:

> **Note:** This value is set to twice the following error at maximum speed based on the measured response of the axis, during the autotuning process. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you need to change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

Position Lock Tolerance

Specifies the maximum position error the servo module will accept in order to indicate the Position Lock status bit is set. This is useful in determining when the desired end position is reached for position moves. This value is interpreted as a +/- quantity.

For example, specifying a lock tolerance of 0.01 provides a minimum positioning accuracy of +/- 0.01 position units, as shown here:

Output Limit

Provides a method of limiting the maximum servo output voltage of a physical axis to a specified level. The servo output for the axis as a function of position servo error, both with and without servo output limiting, is shown below.

The servo output limit may be used as a software current or torque limit if you are using a servo drive in torque loop mode. The percentage of the drive's maximum current that the servo controller will ever command is equal to the specified servo output limit. For example, if the drive is capable of 30 Amps of current for a 10 Volt input, setting the servo output limit to 5V limits the maximum drive current to 15 Amps.

The servo output limit may also be used if the drive cannot accept the full ±10 Volt range of the servo output. In this case, the servo output limit value effectively limits the maximum command sent to the amplifier. For example, if the drive can only accept command signals up to ±7.5 Volts, set the servo output limit value to 7.5 volts.

Manual Tune

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Limits Tab - AXIS_SERVO_DRIVE

Use this tab to make the following offline configurations:

- enable and set maximum positive and negative software travel limits, and
- configure both Position Error Tolerance and Position Lock Tolerance,

for an axis of the type AXIS_SERVO_DRIVE configured as a Servo drive in the General tab of this dialog.

The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Hard Travel Limits

Enables a periodic test that monitors the current state of the positive and negative overtravel limit switch inputs, when Positioning Mode is set to Linear (in the Conversion tab of this dialog). If an axis is configured for hardware overtravel checking and if that axis passes beyond a positive or negative overtravel limit switch, a Positive Hard Overtravel Fault or Negative Hard Overtravel Fault is issued. The response to this fault is specified by the Hard Overtravel setting (in the Fault Actions tab of this dialog).

Soft Travel Limits

Enables software overtravel checking for an axis when Positioning Mode is set to Linear (in the Conversion tab of this dialog). If an axis is configured for software overtravel limits and if that axis passes beyond these maximum travel limits (positive or negative), a software overtravel fault is issued. The response to this fault is specified by the Soft Overtravel setting (in the Fault Actions tab of this dialog). Software overtravel limits are disabled during the tuning process.

Maximum Positive

Type the maximum positive position to be used for software overtravel checking, in position units.

Note: The Maximum Positive limit must always be greater than the Maximum Negative limit.

Maximum Negative

Type the maximum negative position to be used for software overtravel checking, in position units.

> **Note:** The Maximum Negative limit must always be less than the Maximum Positive limit.

Position Error Tolerance

Specifies how much position error the servo will tolerate before issuing a position error fault. This value is interpreted as a +/- quantity.

For example, setting Position Error Tolerance to 0.75 position units means that a position error fault will be generated whenever the position error of the axis is greater than 0.75 or less than -0.75 position units, as shown here:

> **Note:** This value is set to twice the following error at maximum speed based on the measured response of the axis, during the autotuning process. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you need to change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

Position Lock Tolerance

Specifies the maximum position error the servo module will accept in order to indicate the Position Lock status bit is set. This is useful in determining when the desired end position is reached for position moves. This value is interpreted as a +/- quantity.

For example, specifying a lock tolerance of 0.01 provides a minimum positioning accuracy of +/- 0.01 position units, as shown here:

Set Custom Limits

Click this button to open the Custom Limit Attributes dialog.

Manual Tune

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

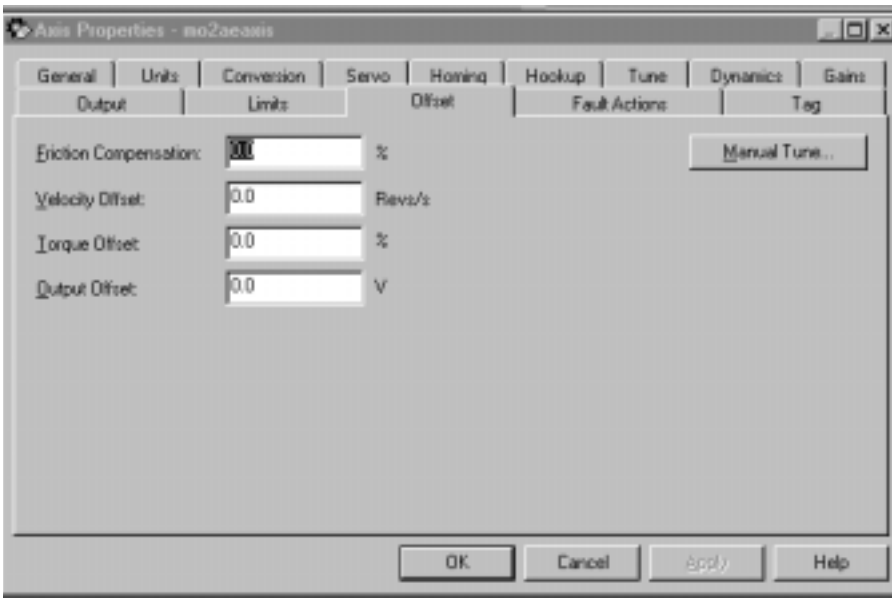
Note: The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Offset Tab - AXIS_SERVO

Use this tab to make offline adjustments to the following Servo Output values:

- Friction Compensation
- Velocity Offset
- Torque Offset
- Output Offset

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits

- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Friction Compensation

The percentage of output level added to a positive current Servo Output value, or subtracted from a negative current Servo Output value, for the purpose of moving an axis that is stuck in place due to static friction.

It is not unusual for an axis to have enough static friction – called “sticktion” – that, even with a significant position error, the axis refuses to budge. Friction Compensation is used to break “

sticktion” in the presence of a non-zero position error. This is done by adding, or subtracting, a percentage output level), called Friction Compensation to the Servo Output value.

The Friction Compensation value should be just less than the value that would break the “sticktion”

A larger value will cause the axis to “dither”, i.e. move rapidly back and forth about the commanded position.

Velocity Offset

Provides a dynamic velocity correction to the output of the position servo loop, in position units per second. Because the position servo loop output value is updated synchronously every Coarse Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block programming.

Torque Offset

Provides a dynamic torque command correction to the output of the velocity servo loop, as a percentage of velocity servo loop output. Because velocity servo loop output is updated synchronously every

Coarse Update Period, the Torque Offset can be tied into custom outer control loop algorithms using Function Block programming.

Output Offset

Corrects the problem of axis “drift”, by adding a fixed voltage value (not to exceed ±10 Volts) to the Servo Output value. Input a value to achieve near zero drive velocity when the uncompensated Servo Output value is zero.

When interfacing an external Servo Drive – especially for velocity servo drives, it is necessary to compensate for the effect of drive offset. Cumulative offsets of the servo module’s DAC output and the Servo Drive Input result in a situation where a zero commanded Servo Output value causes the axis to “drift”. If the drift is excessive, it can play havoc on the Hookup Diagnostic and Tuning procedures, as well as result in a steady-state non-zero position error when the servo loop is closed.

Manual Tune

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Offset Tab - AXIS_SERVO_DRIVE

Use this tab to make offline adjustments to the following Servo Output values:

- Friction Compensation,
- Velocity Offset, and
- Torque Offset

for an axis of the type AXIS_SERVO_DRIVE configured as a Servo drive in the General tab of this dialog.

The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Tune dialog: click on the Manual Tune button to open the Manual Tune dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

  Note: The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Friction Compensation

The percentage of output level added to a positive current Servo Output value, or subtracted from a negative current Servo Output value, for the purpose of moving an axis that is stuck in place due to static friction.

It is not unusual for an axis to have enough static friction – called "sticktion" – that, even with a significant position error, the axis refuses to budge. Friction Compensation is used to break "sticktion" in the presence of a non-zero position error. This is done by adding, or subtracting, a percentage output level), called Friction Compensation to the Servo Output value.

The Friction Compensation value should be just less than the value that would break the "sticktion". A larger value will cause the axis to "dither", i.e. move rapidly back and forth about the commanded position.

Velocity Offset

Provides a dynamic velocity correction to the output of the position servo loop, in position units per second. Because the position servo loop output value is updated synchronously every Coarse Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block programming.

Torque Offset

Provides a dynamic torque command correction to the output of the velocity servo loop, as a percentage of velocity servo loop output.

Because velocity servo loop output is updated synchronously every Coarse Update Period, the Torque Offset can be tied into custom outer control loop algorithms using Function Block programming.

Manual Tune

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

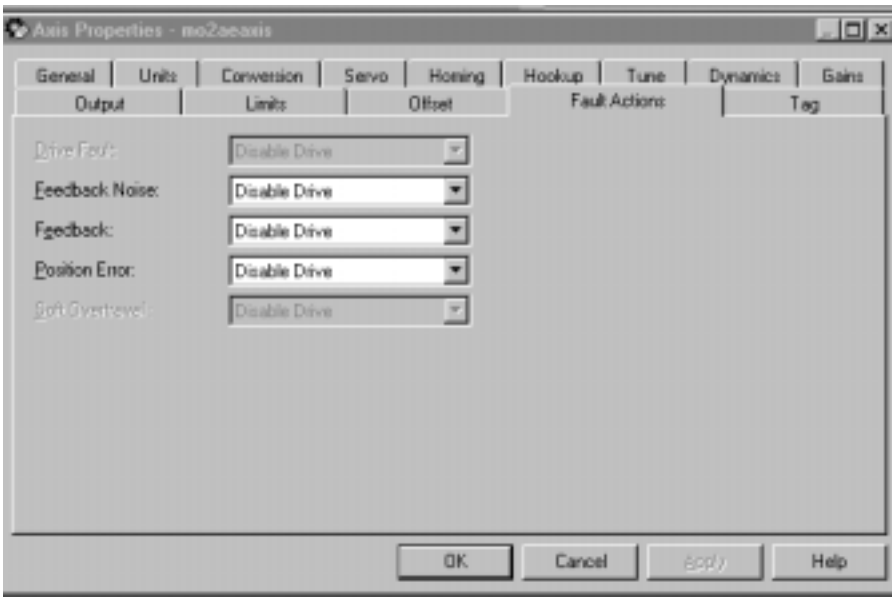
> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Fault Actions Tab - AXIS_SERVO

Use this tab to specify the actions that will be taken in response to the following faults:

- Drive Fault
- Feedback Noise Fault
- Feedback Loss Fault
- Position Error Fault
- Soft Overtravel Fault

for an axis of the type AXIS_SERVO.



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Select one of the following fault actions for each fault type:

- Shutdown - If a fault action is set to Shutdown, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated. Furthermore, this fault action opens the OK contact associated with the servo module which can be used to open the E-stop string to the drive power supply. Shutdown is the most severe action to a fault and it is usually reserved for faults that could endanger the machine or the operator if power is not removed as quickly and completely as possible.

- Disable Drive - If a fault action is set to Disable Drive, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated.

- Stop Motion - If a fault action is set to Stop Motion, then when the associated fault occurs, the axis immediately starts decelerating the axis command position to a stop at the configured Maximum Deceleration Rate without disabling servo action or the servo modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults. After the stop command fault action has stopped the axis, no further motion can be generated until the fault is first cleared.

- Status Only - If a fault action is set to Status Only, then when the associated fault occurs, no action is taken. The application program must handle any motion faults. In general, this setting should only be used in applications where the standard fault actions are not appropriate.

WARNING

Selecting the wrong fault action for your application can cause a dangerous condition. Keep clear of moving machinery.

Drive Fault

Specifies the fault action to be taken when a drive fault condition is detected, for an axis with the Drive Fault Input enabled (in the Servo tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown and Disable Drive.

Feedback Noise

Specifies the fault action to be taken when excessive feedback noise is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Feedback Loss

Specifies the fault action to be taken when feedback loss condition is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Position Error

Specifies the fault action to be taken when position error exceeds the position tolerance set for the axis, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Soft Overtravel

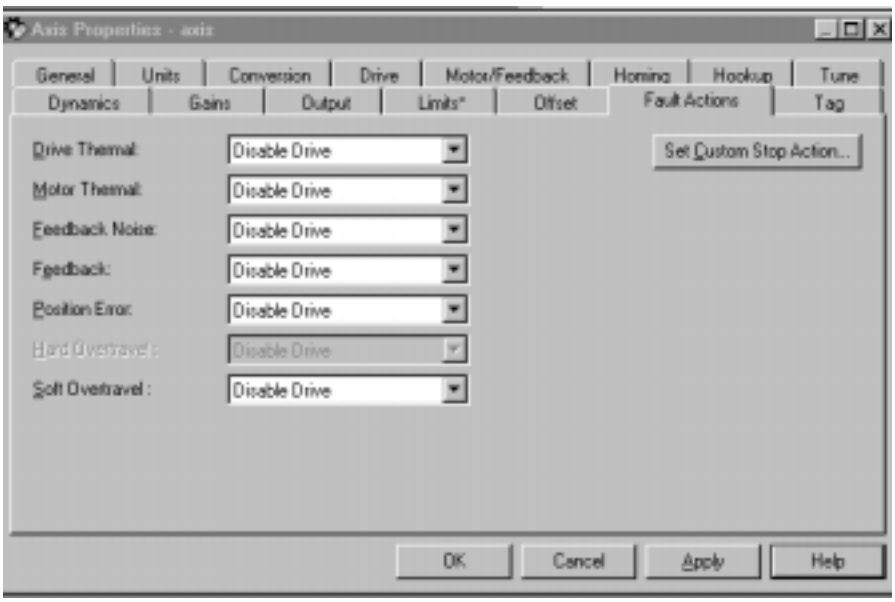
Specifies the fault action to be taken when a software overtravel error occurs, for an axis with Soft Travel Limits enabled and configured (in the Limits tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Fault Actions Tab - AXIS_SERVO_DRIVE

Use this tab to specify the actions that will be taken in response to the following faults:

- Drive Thermal Fault
- Motor Thermal Fault
- Feedback Noise Fault
- Feedback Fault
- Position Error Fault
- Hard Overtravel Fault
- Soft Overtravel Fault

for an axis of the type AXIS_SERVO_DRIVE.



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Select one of the following fault actions for each fault type:

- Shutdown - If a fault action is set to Shutdown, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated. Furthermore, this fault action opens the OK contact associated with the servo module which can be used to open the E-stop string to the drive power supply. Shutdown is the most severe action to a fault and it is usually reserved for faults that could endanger the machine or the operator if power is not removed as quickly and completely as possible.
- Disable Drive - If a fault action is set to Disable Drive, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated.
- Stop Motion - If a fault action is set to Stop Motion, then when the associated fault occurs, the axis immediately starts decelerating the axis command position to a stop at the configured Maximum Deceleration Rate without disabling servo action or the servo modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults. After the stop command fault action has stopped the axis, no further motion can be generated until the fault is first cleared.
- Status Only - If a fault action is set to Status Only, then when the associated fault occurs, no action is taken. The application program must handle any motion faults. In general, this setting should only be used in applications where the standard fault actions are not appropriate.

WARNING

Selecting the wrong fault action for your application can cause a dangerous condition. Keep clear of moving machinery.

Drive Thermal

Specifies the fault action to be taken when a Drive Thermal Fault is detected, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Motor Thermal

Specifies the fault action to be taken when a Motor Thermal Fault is detected, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Feedback Noise

Specifies the fault action to be taken when excessive feedback noise is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Feedback

Specifies the fault action to be taken when Feedback Fault is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Position Error

Specifies the fault action to be taken when position error exceeds the position tolerance set for the axis, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Hard Overtravel

Specifies the fault action to be taken when an axis encounters a travel limit switch, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Soft Overtravel

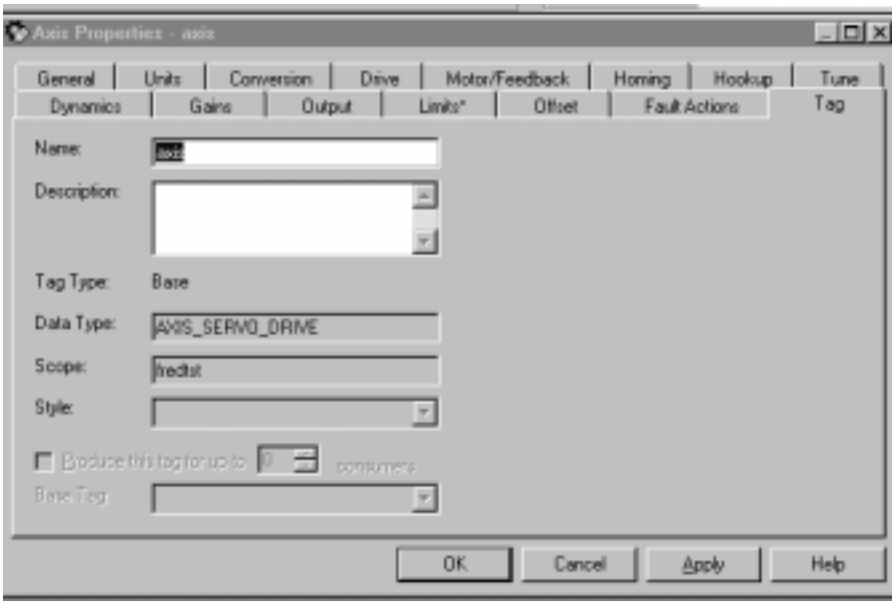
Specifies the fault action to be taken when a software overtravel error occurs, for an axis with Soft Travel Limits enabled and configured (in the Limits tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only.

Set Custom Stop Action

Opens the Custom Stop Action Attributes dialog

Tag Tab

Use this tab to modify the name and description of the axis. When you are online, all of the parameters on this tab transition to a read-only state, and cannot be modified. If you go online before you save your changes, all pending changes revert to their previously-saved state.



Name

Displays the name of the current tag. You can rename this tag, if you wish.

Description

Displays the description of the current tag, if any is available. You can edit this description, if you wish.

Tag Type

Indicates the type of the current tag. This type may be:

- Base
- Alias
- Consumed

Displays the data type associated with the current tag.

Scope

Displays the scope of the current tag. The scope is either controller scope, or program scope, based on one of the existing programs in the controller.

Style

Displays the default style in which to display the value of the tag. Note that style is only applicable to an atomic tag; a structure tag does not have a display style.

Produce this tag for up to

A checked box indicates that this tag is available to remote controllers through controller-to-controller messaging. If this box is checked, the system displays the maximum number of consumers (i.e., connections) allowed for this tag.

The default number of consumers is 2.

Base Tag

If this tag is an alias, this field displays the name of the underlying tag on which this alias was based. The base tag actually defines the memory where the data element is stored.

Manual Tune

Click on this button to open the Dynamics tab of the Manual Tune dialog for online editing of the Maximum Velocity, Maximum Acceleration, and Maximum Deceleration parameters.

> **Note:** The Manual Tune button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Assigning Additional Motion Axes

You can assign additional axes by repeating the preceding sections. To name and assign another axis, refer to the *Naming an Axis* section.

You can assign up to 16 1756-M02AE modules to each Logix5550 controller. Each module uses a maximum of two axes.

Developing a Motion Application Program

To write a motion application program, you can insert motion instructions directly into the ladder diagram application program. The motion instruction set consists of five groups of motion instructions:

- Motion state instructions
- Motion move instructions
- Motion group instructions
- Motion event instructions
- Motion configuration instructions

For more information about these instructions, refer to the Logix5550 Controller Motion Instruction Set Reference Manual – publication 1756-6.4.3.

Understanding a Programming Example

The following figure shows several rungs of a motion control application program.

Rung 0:

Enables the Feed and Cut axes when you press the servo_on button.

Rung 1:

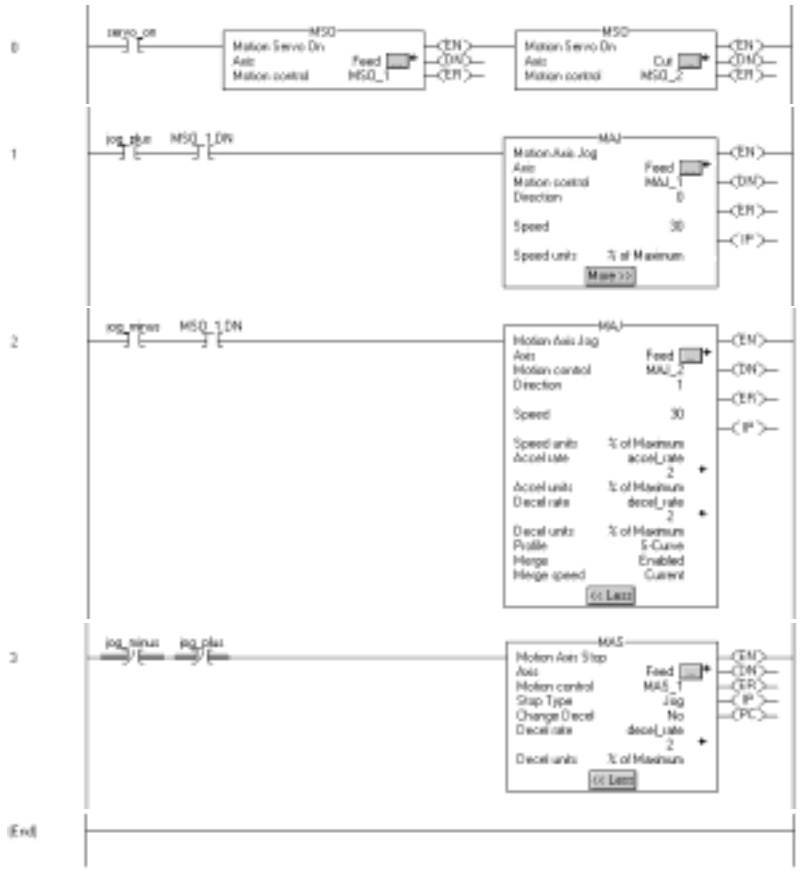
Jogs the Feed axis in the positive direction when you press the jog_plus button.

Rung 2:

Jogs the Feed axis in the reverse direction when you press the jog_minus button.

Rung 3:

Stops the Feed axis when you release with the jog_plus button or the jog_minus button.



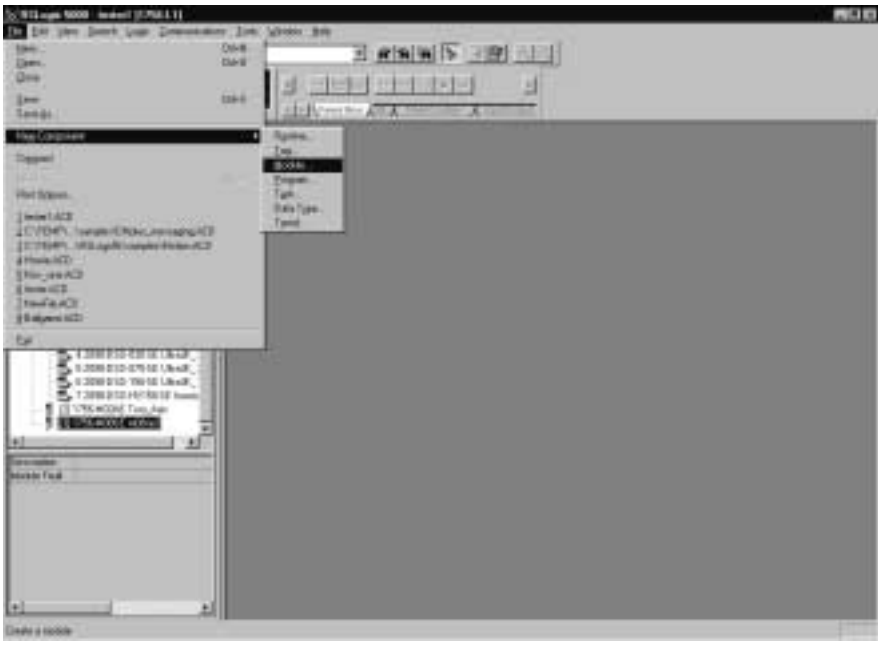
For more information about instructions and creating application programs, refer to the *Logix5550 Controller Instruction Set Reference Manual*, publication 1756-RM003 and the *Logix5550 Motion Instruction Reference Set Manual*, publication 1756-RM007.

Configuring a 1394C-SJT05/10/22-D Digital Servo Drive

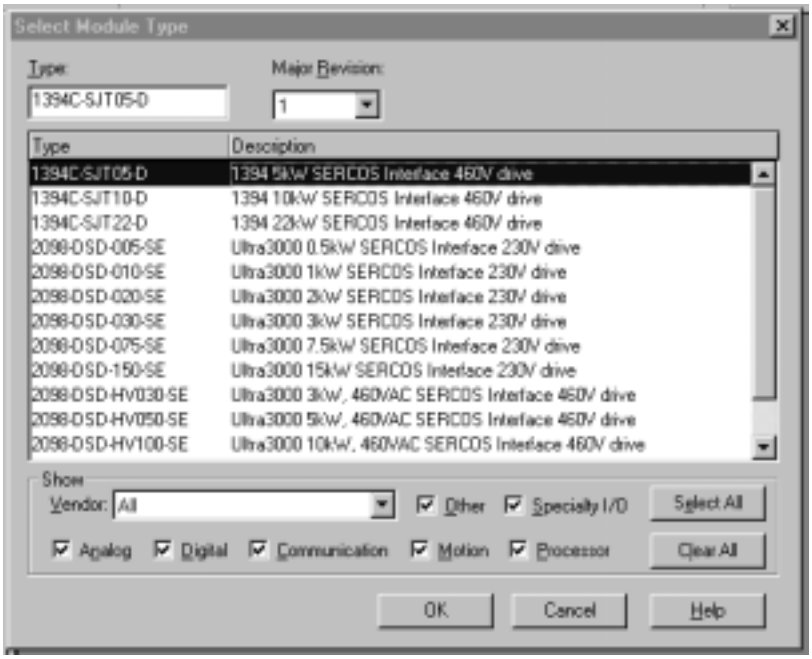
To configure a 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module:

1. In the Controller Organizer, in the I/O Configuration branch, select a 1756-M08SE motion module.

2. In the File menu, select New Component then Module.

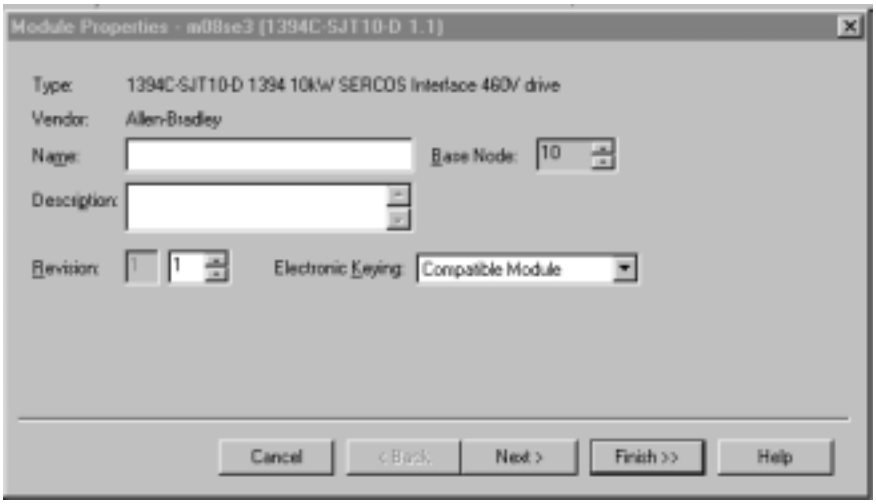


3. In the Select Module Type dialog, select the desired drive module: 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D



4. Press the OK button to close the Select Module Type dialog. The Module Properties wizard opens.

5. Fill in the required parameters for each page, then click the Next> button.



6. When you complete the last page, click the Finish> button. A new drive module displays beneath the selected 1756-M08SE motion module.

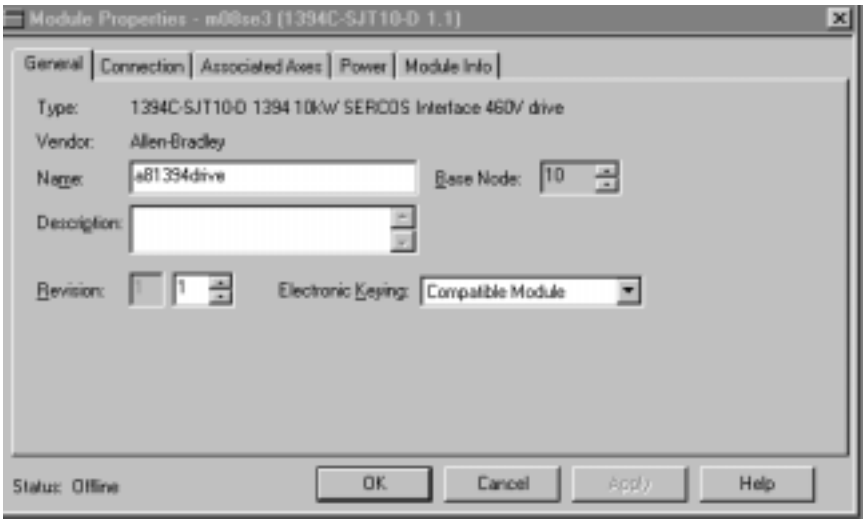
1394C-SJT05/10/22-D Digital Servo Drive Overview

The 1756-M08SE 8 Axis SERCOS interface motion module can be connected to any of three drives:

- 1394C-SJT05-D 5 KW digital servo drive
- 1394C-SJT10-D 10 KW digital servo drive
- 1394C-SJT22-D 22 KW digital servo drive.

Each drive can be associated with up to 4 axes of the AXIS_SERVO_DRIVE tag type. The 1756-M08SE 8 Axis SERCOS interface module can support up to 8 axes (using varying combinations of from 2 to 8 drives).

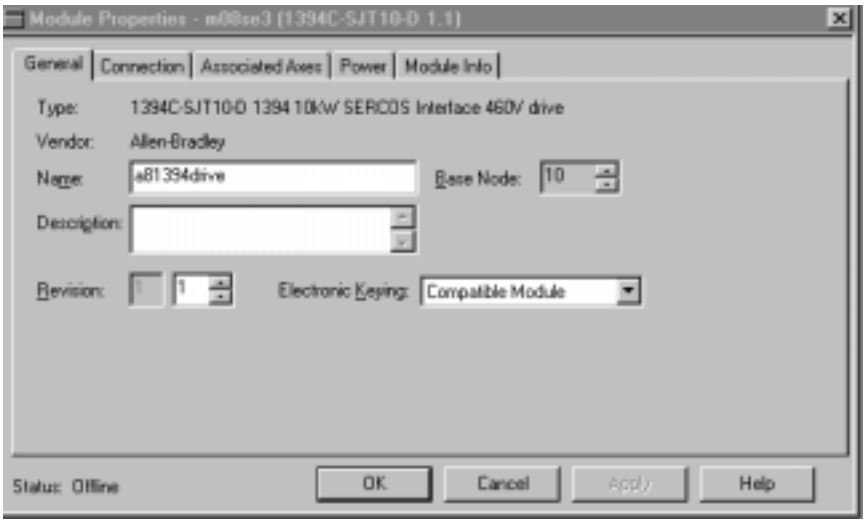
The module for a 1394C-SJT05/10/22-D drive has 5 tabs:



- General tab
- Connection tab
- Axes Association
- Power tab
- Module Info tab.

General Tab

Use this tab to enter the module properties for a 1394C-SJT05-D 5 KW, 1394C-SJT10-D 10 KW or 1394C-SJT22-D 22 KW digital servo drive module.



IMPORTANT To create any one of the 1394C-SJT modules, the parent module must be a 1756-M08SE 8 Axis SERCOS interface module.

On this tab, you can:

- view the type and description of the module being created
- view the vendor of the module being created
- enter the name of the module
- enter a description for the module
- set the Base Node for the module
- select the minor revision number of your module
- select Electronic Keying (Exact Match, Compatible Module, or Disable Keying)
- view the status the controller has about the module (you can only view the status while online)

Type

Displays the module type of the module being created: 1394C-SJT05-D 5 KW, 1394C-SJT10-D 10 KW, or 1394C-SJT22-D 22 KW digital servo drive module (read only).

Vendor

Displays the vendor of the module being created (read only).

Name

Enter the name of the module. The name must be IEC 1131-3 compliant. If you attempt to enter an invalid character or exceed the maximum length, the software beeps and ignores the character.

Description

Enter a description for the module here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software beeps to warn you, and ignores any extra characters.

Base Node

Type or select the Base Node number of the drive module. This node number is determined by multiplying the node number from the module's rotary switch (1 to 9) by a factor of ten. Thus, valid Base Node values are 10, 20, 30, 40, 50, 60, 70, 80 or 90.

Revision

Select the minor revision number of your module.

The revision is divided into the major revision and minor revision. The major revision displayed statically is chosen on the Select Module Type dialog.

The major revision is used to indicate the revision of the interface to the module. The minor revision is used to indicate the firmware revision.

Electronic Keying

Select one of these keying options for your module during initial module configuration:

- Exact Match - all of the parameters described below must match or the inserted module will reject the connection.
- Compatible Module
  - the Module Types, Catalog Number, and Major Revision must match
  - the Minor Revision of the physical module must be equal to or greater than the one specified in the software

  or the inserted module will reject the connection
- Disable Keying – Logix5550 will not employ keying at all.

WARNING

Changing the RPI and Electronic Keying selections may cause the connection to the module to be broken and may result in a loss of data.

Be extremely cautious when using this option; if used incorrectly, this option can lead to personal injury or death, property damage or economic loss.

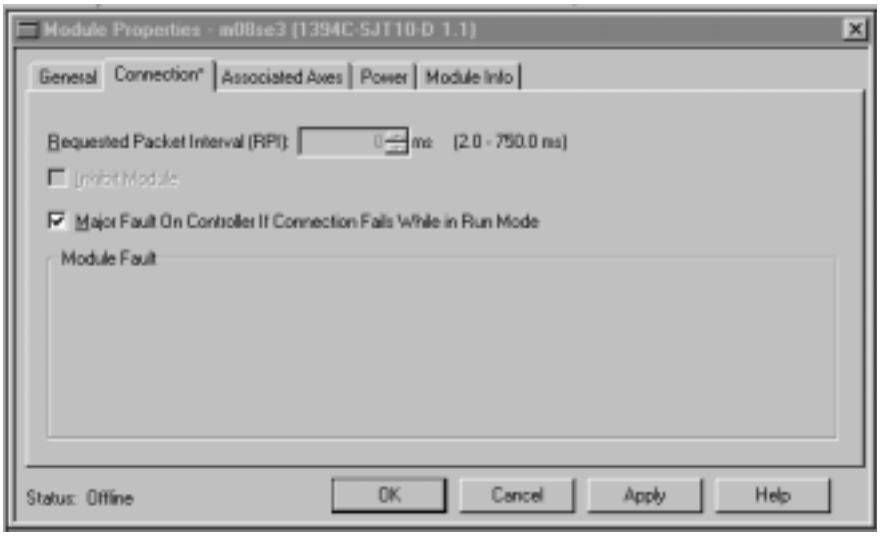
When you insert a module into a slot in a ControlLogix chassis, RSLogix 5000 compares the following information for the inserted module to that of the configured slot:

- Vendor
- Product Type
- Catalog Number
- Major Revision
- Minor Revision

This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

Connection Tab

Use this tab to define controller to drive module behavior.



On this tab, you can:

- choose to inhibit the module
- configure the controller so loss of the connection to this module causes a major fault
- view module faults

TIP

The data on this tab comes directly from the controller. This tab displays information about the condition of the connection between the module and the controller.

Requested packet Interval

The amount of time (in milliseconds) between updates of data from the remote controller. This value is scaled as a real value from the microseconds stored in the controller. The local controller will receive data at least this fast. (Not enabled for this release.)

Inhibit Module checkbox

Check/Uncheck this box to inhibit/uninhibit your connection to the module. Inhibiting the module causes the connection to the module to be broken.

IMPORTANT Inhibiting/uninhibiting connections applies mainly to direct connections, and not to the CNB module.

WARNING Inhibiting the module causes the connection to the module to be broken and may result in loss of data.

When you check this box and go online, the icon representing this module in the controller organizer displays the Attention Icon.

If you are:	Check this checkbox to:
offline	put a place holder for a module you are configuring
online	stop communication to a module. If you inhibit the module while you are online and connected to the module, the connection to the module is nicely closed. The module's outputs will go to the last configured Program mode state.If you inhibit the module while online but a connection to the module has not been established (perhaps due to an error condition or fault), the module is inhibited. The module status information will change to indicate that the module is 'Inhibited' and not 'Faulted'.If you uninhibit a module (clear the checkbox) while online, and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured (if you are the owner controller) with the configuration you have created for that module. If you are a listener (have chosen a “Listen Only” Communications Format), you can not re-configure the module. If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.

Major Fault on Controller if Connection Fails checkbox

Check this box to configure the controller so that failure of the connection to this module causes a major fault on the controller if the connection for the module fails.

Module Fault

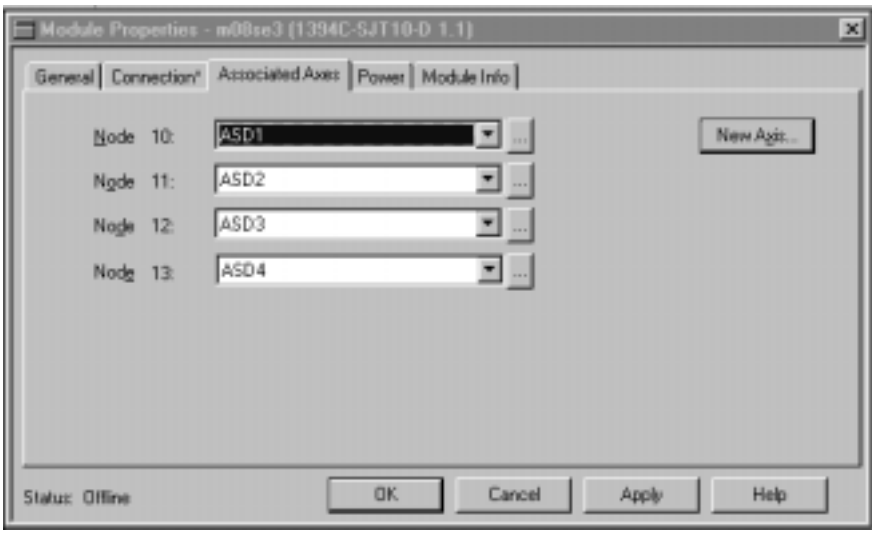
Displays the fault code returned from the controller (related to the module you are configuring) and the text detailing the Module Fault that has occurred.

The following are common categories for errors:

- Connection Request Error - The controller is attempting to make a connection to the module and has received an error. The connection was not made.
- Service Request Error - The controller is attempting to request a service from the module and has received an error. The service was not performed successfully.
- Module Configuration Invalid - The configuration in the module is invalid. (This error is commonly caused by the Electronic Key Passed fault).
- Electronic Keying Mismatch - Electronic Keying is enabled and some part of the keying information differs between the.

Associated Axes Tab

Use this tab to configure the selected 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module by associating up to four AXIS_SERVO_DRIVE axis tags with configured axis modules.



Node X0

Represents Axis 0 on the 1756-M08SE SERCOS module. The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (1). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 0. This field transitions to a read only state while online. Click on the Ellipses (…) button to the right of this field to open the Axis properties dialog box for the associated axis.

Node X1

Represents Axis 1 on the 1756-M08SE SERCOS module. The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (1). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 1. This field transitions to a read only state while online. Click on the Ellipses (…) button to the right of this field to open the Axis properties dialog box for the associated axis.

Node X2

Represents Axis 2 on the 1756-M08SE SERCOS module The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (2). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 2. This field transitions

to a read only state while online. Click on the Ellipses (…) button to the right of this field to open the Axis properties dialog box for the associated axis.

Node X3

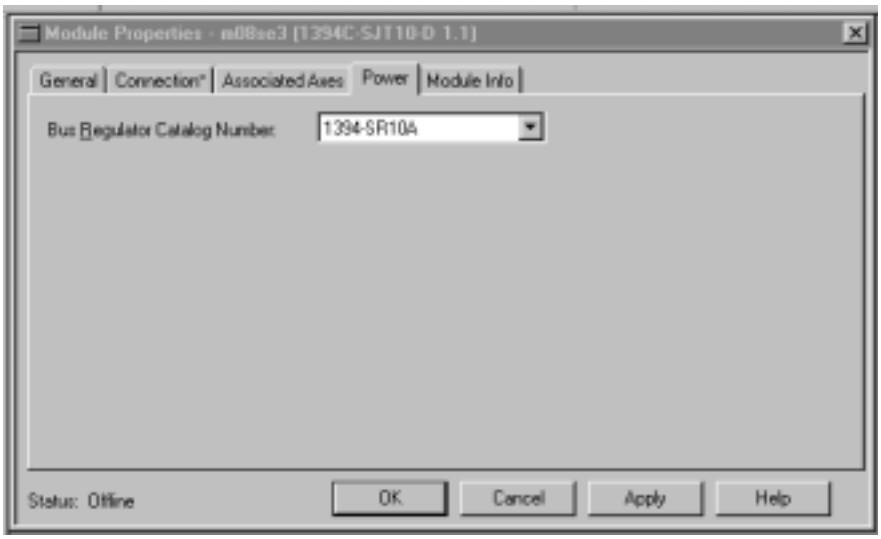
Represents Axis 3 on the 1756-M08SE SERCOS module The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (3). This field allows you to associate an AXIS_SERVO_DRIVE tag with Axis 3. This field transitions to a read only state while online. Click on the Ellipses (…) button to the right of this field to open the Axis properties dialog box for the associated axis.

New Axis button

Click this button to navigate to the New Tag dialog to create an AXIS_SERVO_DRIVE tag to associate with one of the channels.

Power Tab

Use this tab to select a bus regulator for your 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module.



Bus Regulator ID

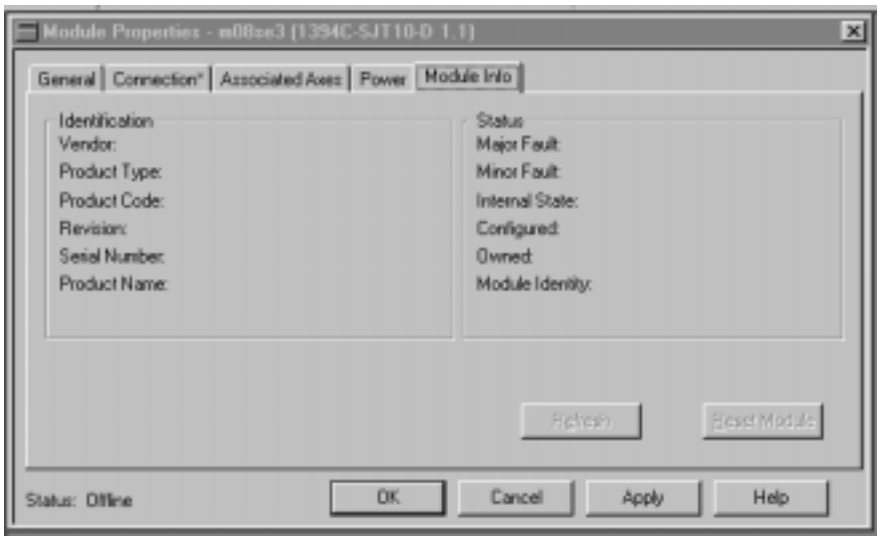
Select the catalog number that describes bus regulator device used by the 1394C-SJT05-D, 1394C-SJT10-D or 1394C-SJT22-D drive module.

Depending upon the Drive Module you have selected, one or more of the following are available:

Bus Regulator ID	Description
1394-SR10A	1400 Watt Resistor, for 5 and 10 kW modules
1394-SR9A	300 Watt External Shunt, No Fan, for 22 kW modules
1394-SR9AF	900 Watt External Shunt, No Fan, for 22 kW modules
1394-SR36A	1800 Watt External Shunt, No Fan, for 22 kW modules
1394-SR36AF	3600 Watt External Shunt, No Fan, for 22 kW modules
<none>	No bus regulator
Internal	The bus regulator is internal to the drive and need not be specified
Custom	A bus regulator not listed above

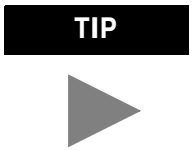
Module Info tab

Use this tab to display identifying and status information about the 1394C-SJT05/10/22-D drive module. It also allows you to refresh a module and reset a module to its power-up state.



The information on this tab is not displayed if you are:

- offline, or
- currently creating a module



The data on this tab comes directly from the module. If you selected a Listen-Only communication format when you created the module, this tab is not available.

Identification

Displays the module's:

- Vendor
- Product Type
- Product Code
- Revision
- Serial Number

Product Name

The name displayed in the Product Name field is read from the module. This name displays the series of the module.

Major/Minor Fault Status

Statuses are: EEPROM fault, Backplane fault, None

Internal State Status

Displays the module's current operational state.

- Self-test
- Flash update
- Communication fault
- Unconnected
- Flash configuration bad
- Major Fault (please refer to "Major/Minor Fault Status" above)
- Run mode
- Program mode

(16#xxxx) unknown

If you selected the wrong module from the module selection tab, this field displays a hexadecimal value. A textual description of this state is only given when the module identity you provide is a match with the actual module.

Configured

Displays a yes or no value indicating whether the module has been configured by an owner controller connected to it. Once a module has been configured, it stays configured until the module is reset or power is cycled, even if the owner drops connection to the module. This information does not apply to adapters.

Owned

Displays a yes or no value indicating whether an owner controller is currently connected to the module. This information does not apply to adapters.

Module Identity

Displays:	If the module in the physical slot:
Match	agrees with what is specified on the General Tab. In order for the Match condition to exist, all of the following must agree: Vendor Module Type (the combination of Product Type and Product Code for a particular Vendor) Major Revision
Mismatch	does not agree with what is specified on the General Tab

This field does not take into account the Electronic Keying or Minor Revision selections for the module that were specified on the General Tab.

Refresh

Click on this button to refresh the tab with the new data from the module.

Reset Module

Click on this button to return a module to its power-up state by emulating the cycling of power.

WARNING

Resetting a module causes all connections to or through the module to be closed; this may result in loss of control.

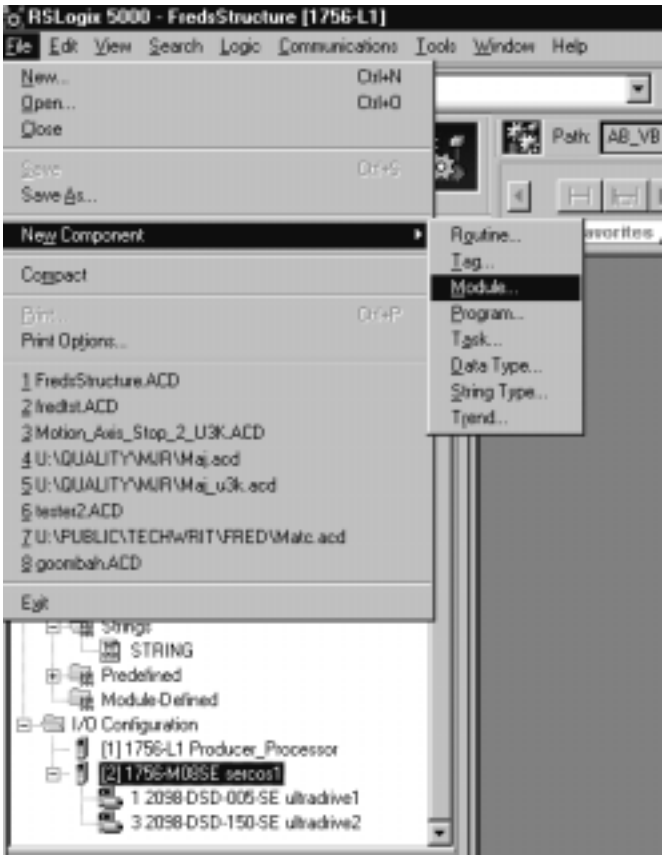
Configuring an Ultra 3000 Drive

The Ultra3000 Digital Servo Drive with fiber optic SERCOS interface simplifies the integration of the Ultra3000 with the ControlLogix architecture by providing single point drive commissioning through RSLogix5000 software and reducing the control wiring to a single fiber optic cable.

You can initiate the configuration of an Ultra3000 drive module by either of two methods:

1. In the Controller Organizer, in the I/O Configuration branch, select a 1756-M08SE motion module.

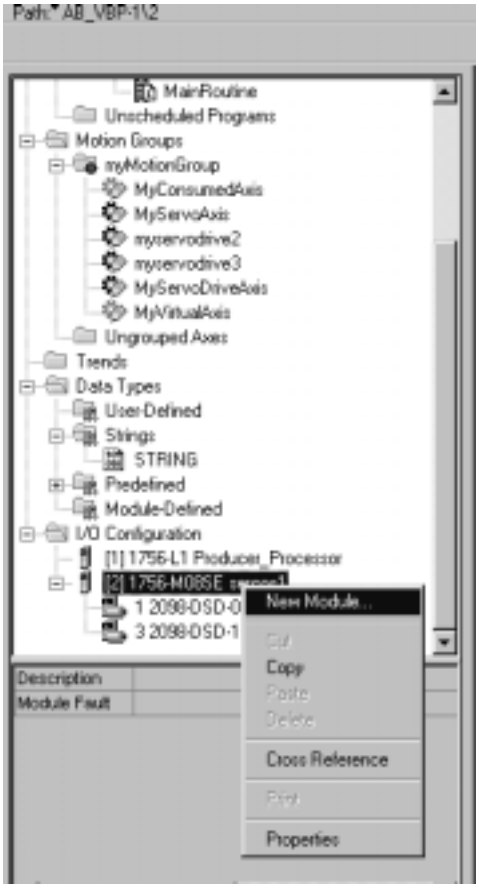
2. In the File menu, select New Component then Module.



OR

3. Right click on the selected 1756-M08SE in the I/O Configuration branch of the Controller Organizer.

4. Select New Module from the pop up menu.



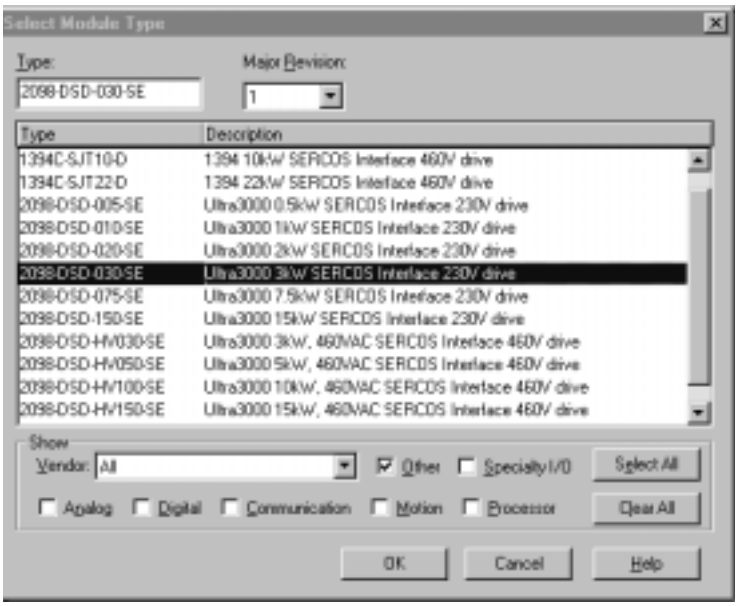
The following fields are displayed only if you are viewing this tab through the Create wizard.

Next> – Click this button to view the next Create wizard page.

<Back – Click this button to view the previous Create wizard page.

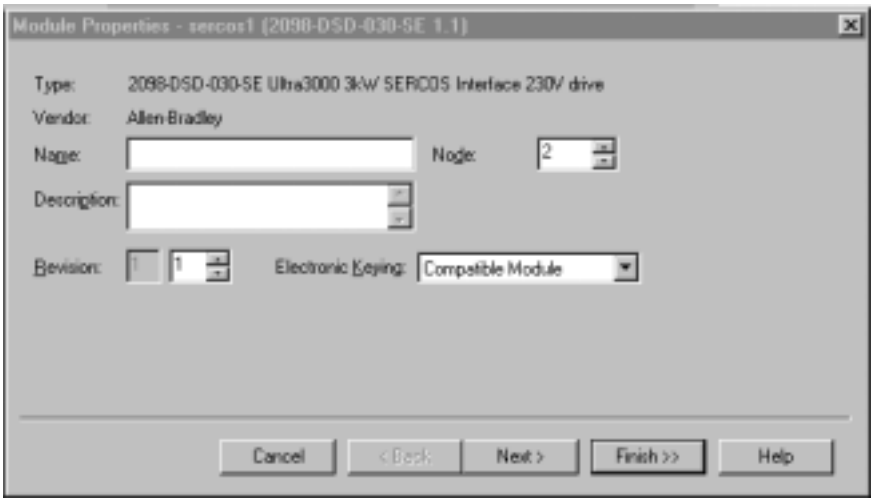
Finish>> – Click this button to close the Create wizard.

The Select Module Type dialog displays.



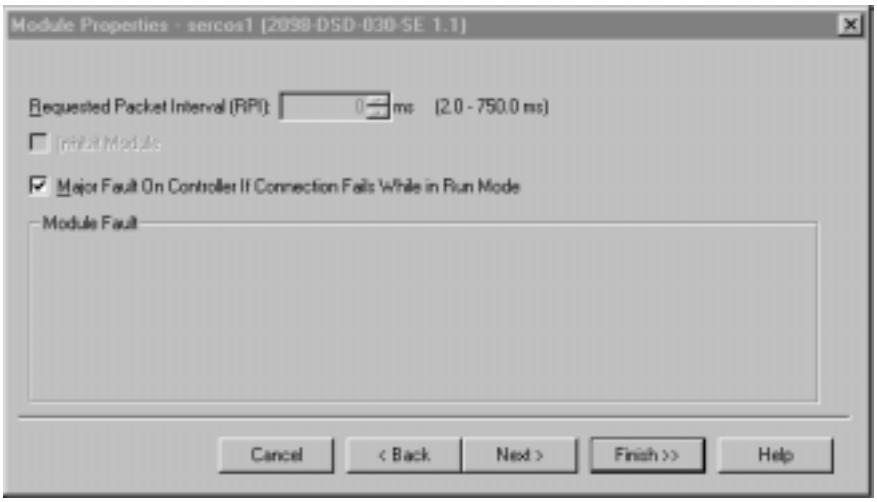
5. In the Select Module Type dialog, select the desired drive module. The Ultra drives begin with the 2098 prefix.

6. Press the OK button to close the Select Module Type dialog. The Ultra Drive Create Wizard Module Properties dialog opens.

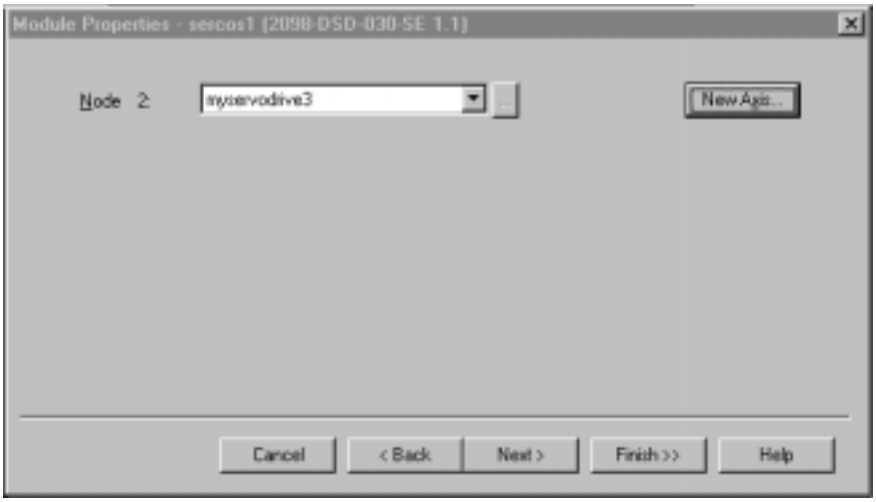


7. You must fill in a name for the drive; this is a required field. Fill in the responses for the other parameters as needed, then click

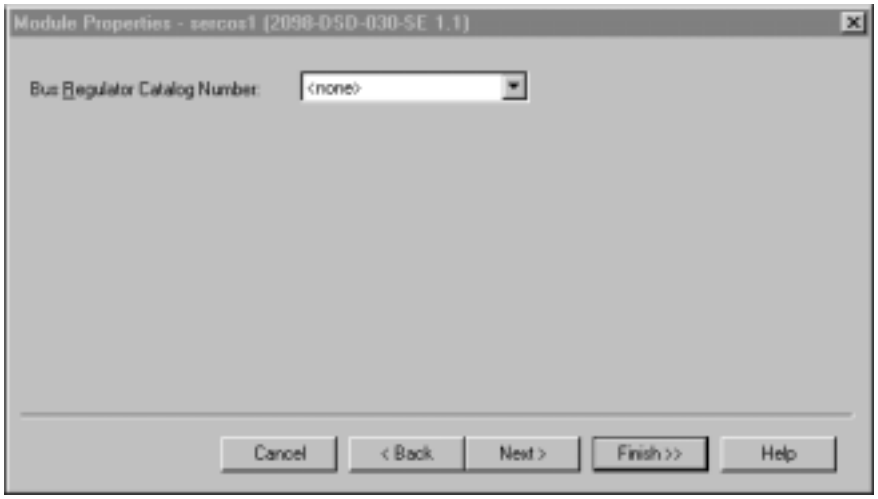
the Next> button to advance to the next wizard screen.



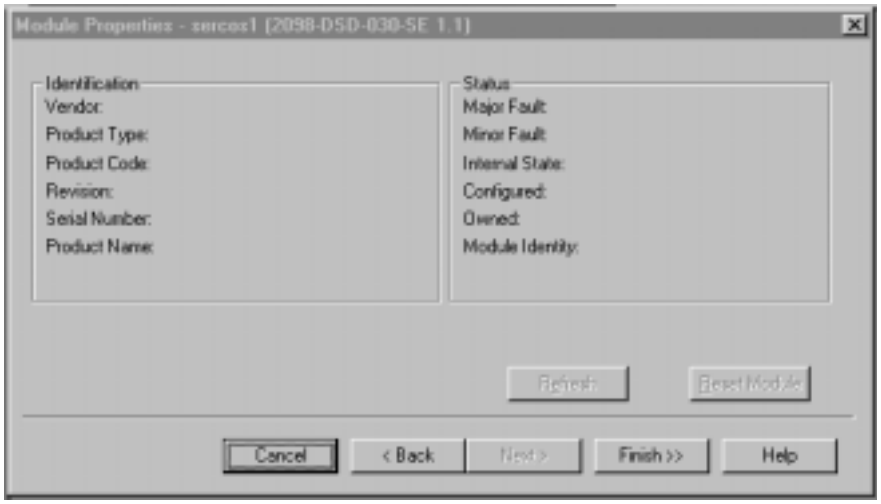
8. Fill in the required information and click on the Next button to advance to the next screen.



9. Fill in the node information for the drive and press Next.

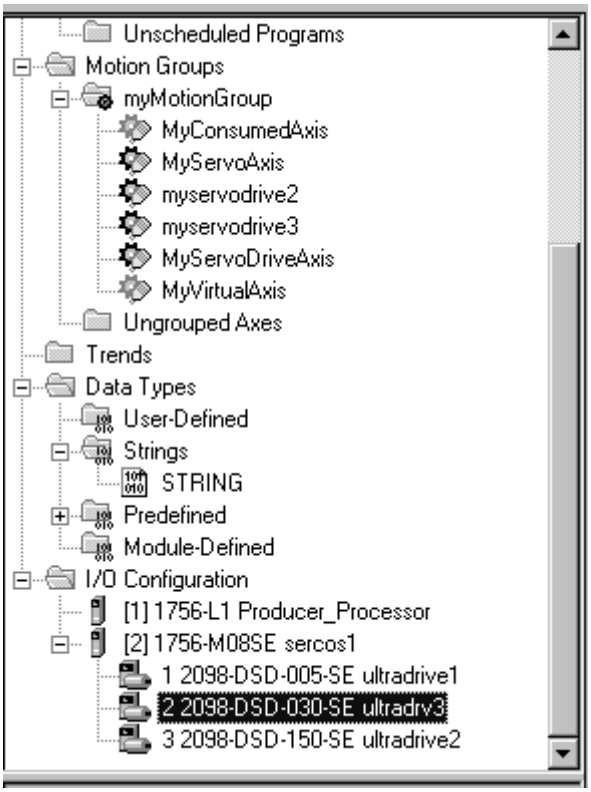


10. Enter the Bus regulator information and press Next to continue.



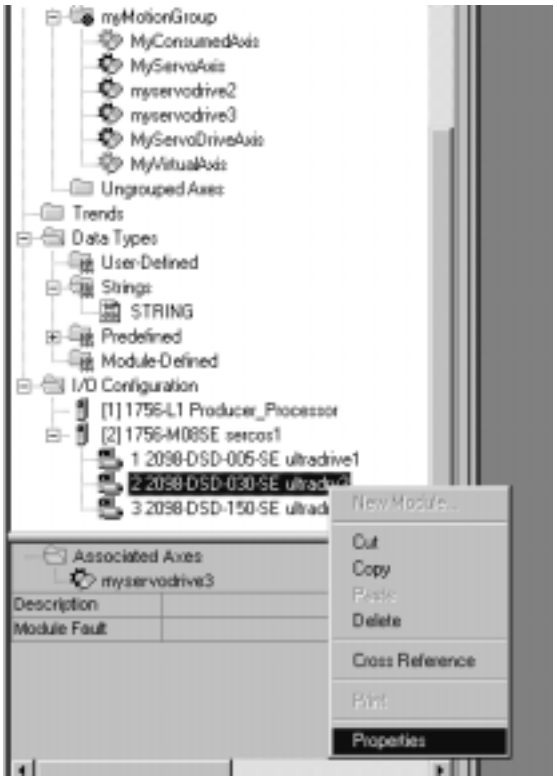
11. When you complete the last page, click the Finish> button. A new drive module displays beneath the selected 1756-M08SE

motion module.

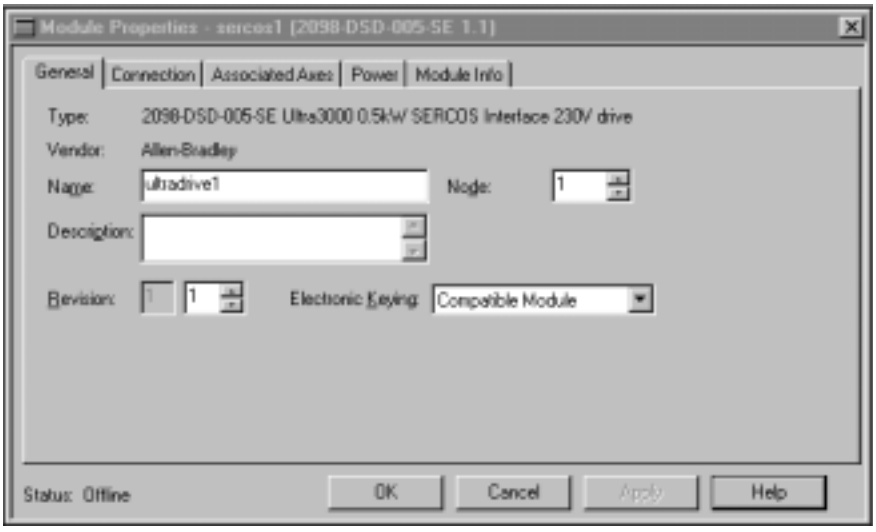


Editing the Ultra Drive Properties

The Module Properties for any of the Ultra3000 drives can be edited by highlighting the drive to be edited, right click with the mouse and selecting Properties.



The Module Properties screen displays.



General Tab

The General Tab is where you edit the basic values for the Ultra drive.

Type

Displays the type and description of the module being created (read only).

Vendor

Displays the vendor of the module being created (read only).

Name

Enter the name of the module.

The name must be IEC 1131-3 compliant. This is a required field and must be completed, otherwise you receive an error message when you exit this tab. An error message is also displayed if a duplicate name is detected, or you enter an invalid character. If you exceed the maximum name length allowed by the software, the extra character(s) are ignored.

Description

Enter a description for the module here, up to 128 characters. You can use any printable character in this field. If you exceed the maximum length, the software ignores any extra character(s).

Node

Select the network node number of the module on the network. Valid values include those network nodes not in use between 1 to 99.

Revision

Select the minor revision number of your module.

The revision is divided into the major revision and minor revision. The major revision displayed statically is chosen on the Select Module Type dialog.

The major revision is used to indicate the revision of the interface to the module. The minor revision is used to indicate the firmware revision.

Slot

Enter the slot number in which the module resides.

Electronic Keying

Select one of these keying options for your module during initial module configuration:

- Exact Match - all of the parameters described below must match or the inserted module will reject the connection.
- Compatible Modules – The following criteria must be met, or else the inserted module will reject the connection:
  - The Module Types, Catalog Number, and Major Revision must match.
  - The Minor Revision of the physical module must be equal to or greater than the one specified in the software.
- Disable Keying – Logix5550 does not employ keying at all.

WARNING

Changing the RPI and Electronic Keying selections may cause the connection to the module to be broken and may result in a loss of data.

Be extremely cautious when using this option; if used incorrectly, this option can lead to personal injury or death, property damage or economic loss.

When you insert a module into a slot in a ControlLogix chassis, RSLogix 5000 compares the following information for the inserted module to that of the configured slot:

- Vendor
- Product Type
- Catalog Number
- Major Revision
- Minor Revision

This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

Status

Displays the status the controller has about the module:

This status:	Indicates:
Standby	A transient state that occurs when shutting down.
Faulted	The controller is unable to communicate with the module. When the status is Faulted, the Connection tab displays the fault.
Validating	A transient state that occurs before connecting to the module.
Connecting	A state that occurs while the connection(s) are being established to the module.
Running	The module is communicating and everything is working as expected.
Shutting Down	The connections are closing.
Inhibited	The connection to the module is inhibited.
Waiting	The connection to this module has not yet been made due to one of the following: • its parent has not yet made a connection to it • its parent is inhibited§ • its parent is faulted
Offline	You are not online.

Connection Tab

Use this tab to define controller to module behavior. On this tab, you can:

- Select a requested packet interval.
- Choose to inhibit the module.
- Configure the controller so loss of the connection to this module causes a major fault.
- View module faults.

TIP

The data on this tab comes directly from the controller. This tab displays information about the condition of the connection between the module and the controller.

Requested Packet Interval

Enter the requested rate of packet arrival (connection update rate). The connection will be scheduled to move data to or from the module at least this often or the connection will fail with the Connection Not Scheduled fault. The minimum and maximum RPI values are shown parenthetically to the right of the box/spin control.

> **Note:** These minimum and maximum values are module-dependent and will differ depending on the limits of the module. For instance, for the 1769-MODULE, this value is limited to 2.0 ms and cannot be changed. For the CompactBus 1769 Virtual Backplane adapter, this value must be 0. And for the 1756-DM modules, valid values range from 1.0 to 750.0 ms.

The RPI is determined by the Owner Controller(s) of a module. If a Listen-Only connection is established, the RPI for that connection cannot be faster than the fastest RPI configured for all owner controllers (for input modules), or faster than the RPI configured for the one owner controller (for output modules).

WARNING

Changing the RPI while online temporarily disables the connection and can result in a loss of data.

For a remote module – if the RPI is changed online, the connection to the module is broken until RSNetWorx is re-run to schedule the network. Breaking the connection may result in a loss of data.

For modules in remote racks, when communicating on a ControlNet network, the fastest the RPI should be set to is 2ms.

Note: The RPI for the CNB module affects only the CNB Rack Object; no other direct connections are affected.

Note: This field is disabled for all motion modules (e.g., 1756-MO2AE, 1756-MO8SE, and all 1394- and Ultra3000 modules).

Inhibit Module

Check/Uncheck this box to inhibit/uninhibit your connection to the module. Inhibiting the module causes the connection to the module to be broken.

Note: Inhibiting/uninhibiting connections applies mainly to direct connections, and not to the CNB module.

Note: A FLEX I/O module using rack communication cannot be inhibited; the Inhibit checkbox on the Connection tab is disabled in this case.

WARNING

Inhibiting the module causes the connection to the module to be broken and may result in loss of data.

When you check this box and go online, the icon representing this module in the controller organizer displays the Warning Icon.

If you are:	Check this checkbox to:
offline	put a place holder for a module you are configuring
online	stop communication to a module • If you inhibit the module while you are online and connected to the module, the connection to the module is nicely closed. The module's outputs go to the last configured Program mode state. • If you inhibit the module while online but a connection to the module has not been established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is 'Inhibited' and not 'Faulted'. • If you uninhibit a module (clear the checkbox) while online, and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured (if you are the owner controller) with the configuration you have created for that module. If you are a listener (have chosen a "Listen Only" Communications Format), you can not re-configure the module. • If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.

Major Fault on Controller if Connection Fails checkbox

Check this box to configure the controller so that failure of the connection to this module causes a major fault on the controller if the connection for the module fails.

Module Fault

Displays the fault code returned from the controller (related to the module you are configuring) and the text detailing the Module Fault that has occurred.

The following are common categories for errors:

- Connection Request Error - The controller is attempting to make a connection to the module and has received an error. The connection was not made.

- Service Request Error - The controller is attempting to request a service from the module and has received an error. The service was not performed successfully.

- Module Configuration Invalid - The configuration in the module is invalid. (This error is commonly caused by the Electronic Key Passed fault).

- Electronic Keying Mismatch - Electronic Keying is enabled and some part of the keying information differs between the software and the module.

Associated Axes Tab (Ultra3000 Drives)

Use this tab to configure the selected 1756-MO8SE motion module by associating axis tags (of the type AXIS_SERVO_DRIVE) with nodes available on the module.

Node

Displays the selected node of the Ultra3000 drive, as selected on the General tab. This field allows you to associate an AXIS_SERVO_DRIVE tag with the driver's node.

> **Note:** This field is read-only while you are online.

Ellipsis (...)

Click on this button to access the Axis Properties dialog for the associated axis.

New Axis

Click on this button to access the New Tag dialog, with the scope, data type, and produced settings appropriate for a produced axis tag.

Power Tab - Ultra Drive

Use this tab to select a bus regulator for your Ultra 3000 drive module.

> **Note:** This parameter does not apply to the Ultra3000 SERCOS drives. The only available selection in the Bus Regulator ID pull-down menu is <none>.

Bus Regulator ID

Select the catalog number that describes bus regulator device used by the Ultra 3000 drive module. Depending upon the Drive Module you have selected, one or more of the following are available:

> **Note:** This parameter does not apply to the Ultra3000 SERCOS drives. The only available selection in the pull-down menu is <none>.

Module Info

The Module Info Tab displays module and status information about the module. It also allows you to reset a module to its power-up state.

The information on this tab is not displayed if you are either offline or currently creating a module

TIP You can use this tab to determine the identity of the module.

The data on this tab comes directly from the module. If you selected a Listen-Only communication format when you created the module, this tab is not available.

- Refresh to display new data from the module.
- Reset Module to return the module to its power-up state by emulating the cycling of power. By doing this, you also clear all faults.

Identification

Displays the module's:

- Vendor
- Product Type
- Product Code
- Revision
- Serial Number
- Product Name

The name displayed in the Product Name field is read from the module. This name displays the series of the module. If the module is a 1756-L1 module, this field displays the catalog number of the memory expansion board (this selection applies to any controller catalog number even if additional memory cards are added.

Major/Minor Fault Status

If you are configuring a:	This field displays one of the following:
digital module	EEPROM fault Backplane fault None
analog module	Comm. Lost with owner Channel fault None
Any other module	None Unrecoverable Recoverable

Internal State Status

Displays the module's current operational state.

- Self-test
- Flash update
- Communication fault
- Unconnected
- Flash configuration bad
- Major Fault (please refer to "Major/Minor Fault Status" above)
- Run mode
- Program mode
- (16#xxxx) unknown

If you selected the wrong module from the module selection tab, this field displays a hexadecimal value. A textual description of this state is only given when the module identity you provide is a match with the actual module.

Configured

Displays a yes or no value indicating whether the module has been configured by an owner controller connected to it. Once a module has been configured, it stays configured until the module is reset or power is cycled, even if the owner drops connection to the module. This information applies to I/O modules only and does not apply to adapters, scanners, bridges, or other communications modules.

Owned

Displays a yes or no value indicating whether an owner controller is currently connected to the module. This information applies to I/O

modules only and does not apply to adapters, scanners, bridges, or other communications modules.

Module Identity

Displays:	If the physical module:
Match	agrees with what is specified on the General Tab order for the Match condition to exist, all of the following must agree: • Vendor • Module Type (the combination of Product Type and Product Code for a particular Vendor) • Major Revision
Mismatch	does not agree with what is specified on the General Tab

This field does not take into account the Electronic Keying or Minor Revision selections for the module that were specified on the General Tab.

> **Note:** The Generic modules, such as the 1756-MODULE, always show a Mismatch because the configured Generic Key does not match any target device.

Reset Module

Click on this button to return a module to its power-up state by emulating the cycling of power.

Resetting a module causes all connections to or through the module to be closed, and this may result in loss of control.

Note: The following modules return an error if a reset is attempted:

- 1756-L1 ControlLogix5550 Programmable Controller
- 1336T AC Vector Drive
- 1395 Digital DC Drive

> **Note:** A controller cannot be reset.

Refresh

Click on this button to refresh the tab with new data from the module.

If you are online in Program, Remote Program or Remote Run mode, and this controller is the owner controller, and you have changed the module's configuration in the software, then when you click the Apply or the OK button, the information is automatically sent to the

controller. The controller tries to send the information to the module (if the module's connection is not inhibited). If you don't click Apply, your changes are not sent to the controller.

Motion Instructions

This chapter describes the 32 motion instructions for RSLogix 5000 programming software.

The motion instructions for the RSLogix 5000 programming software consist of five main categories:

- Motion state instructions – to control or change the operating state of an axis.
- Motion move instructions – to control all aspects of axis position.
- Motion group instructions – to control a group of axes.
- Motion event instructions – control the arming and disarming of special event checking functions.
- Motion configuration instructions – to tune an axis and to run diagnostic tests for the system.

For more information about	Refer to
Motion instructions	The Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007
Types of motion instruction timing	Appendix E - *Instruction Timing*

Motion State Instructions

Motion state instructions directly control or change the operating state of an axis.

The motion state instructions are:

Instruction	Abbreviation	Description	Type of Timing
Motion Servo On	MSO	Enables the servo drive and activates the axis servo loop	Message
Motion Servo Off	MSF	Disables the servo drive and deactivates the axis servo loop	Message

Motion Axis Shutdown	MASD	Forces an axis into the shutdown operating state Once the axis is in the shutdown state, the controller will block any instructions that initiate axis motion.	Message
Motion Axis Shutdown Reset	MASR	Changes an axis from an existing shutdown operating state to an axis ready operating state If all of the axes of a servo module are removed from the shutdown state as a result of this instruction, the OK relay contacts for the module will close.	Message
Motion Direct Drive On	MDO	Enables the servo drive and sets the servo output voltage of an axis	Message
Motion Direct Drive Off	MDF	Disables the servo drive and sets the servo output voltage to the output offset voltage	Message
Motion Axis Fault Reset	MAFR	Clears all motion faults	Message

For more information about motion state instructions, refer to the *Motion State Instructions* chapter of the Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - *Instruction Timing*.

Motion Move Instructions

Motion move instructions control all aspects of axis position.

The motion move instructions are:

Instruction	Abbreviation	Description	Type of Timing
Motion Axis Stop	MAS	Initiates a controlled stop of any motion process on an axis	Immediate Process
Motion Axis Home	MAH	Homes an axis	Message Process
Motion Axis Jog	MAJ	Initiates a jog motion profile for an axis	Immediate Process
Motion Axis Move	MAM	Initiates a move profile for an axis	Immediate Process
Motion Axis Gear	MAG	Enables electronic gearing between two axes	Immediate Process
Motion Change Dynamics	MCD	Changes the speed, acceleration rate, or deceleration rate of a move profile or jog profile in progress	Immediate
Motion Redefine Position	MRP	Changes the command or actual position of an axis	Message

Motion Calculate Cam Profile	MCCP	Calculates a Cam Profile based on an array of cam points.	Immediate
Motion Axis Position Cam	MAPC	Performs electronic camming between any two axes designated in the specified Cam Profile.	Immediate Process
Motion Axis Time Cam	MATC	Performs electronic camming as a function of time designated in the specified Cam Profile.	Immediate Process

For more information about motion state instructions, refer to the *Motion Move Instructions* chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - *Instruction Timing*.

Motion Group Instructions

Motion group instructions initiate action on all axes in a group.

The motion group instructions are:

Instruction	Abbreviation	Description	Type of Timing
Motion Group Stop	MGS	Initiates a stop of motion on a group of axes	Process
Motion Group Programmed Stop	MGPS	Initiates a stop of all motion on all the axes in a group using the method that you set for each axis.	Message Process
Motion Group Shutdown	MGSD	Forces all the axes in a group into the shutdown operating state	Message
Motion Group Shutdown Reset	MGSR	Transitions a group of axes from the shutdown operating state to the axis ready operating state	Message
Motion Group Strobe Position	MGSP	Latches the current command and actual positions of all the axes in a group	Immediate

For more information about motion state instructions, refer to the *Motion Group Instructions* chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - *Instruction Timing*.

Motion Event Instructions

Motion event instructions control the arming and disarming of special event checking functions, such as registration and watch position.

The motion event instructions are:

Instruction	Abbreviation	Description	Type of Timing
Motion Arm Watch Position	MAW	Arms watch-position event checking for an axis	Message Process
Motion Disarm Watch Position	MDW	Disarms watch-position event checking for an axis	Message
Motion Arm Registration	MAR	Arms servo module registration event checking for an axis	Message Process
Motion Disarm Registration	MDR	Disarms servo module registration event checking for an axis	Message
Motion Arm Output Cam	MAOC	Arms an Output Cam for a particular Axis and Output as determined by the operands for the instruction.	Immediate Process
Motion Disarm Output Cam	MDOC	Disarms either one or all Output Cams connected to a specified axis depending on the selection in the Disarm Type operand.	Immediate Process

For more information about motion state instructions, refer to the *Motion Event Instructions* chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - *Instruction Timing*.

Understanding Motion Configuration Instructions

Motion configuration instructions allow you to tune an axis and to run diagnostic tests for your control system. These tests include:

- A motor/encoder hookup test
- An encoder hookup test
- A marker test

The motion configuration instructions are:

Instruction	Abbreviation	Description	Type of Timing
Motion Apply Axis Tuning	MAAT	Computes a complete set of servo gains and dynamic limits based on a previously executed MRAT instruction The MAAT instruction also updates the servo module with the new gain parameters.	Message
Motion Run Axis Tuning	MRAT	Commands the servo module to run a tuning motion profile for an axis	Message Process

Motion Apply Hookup Diagnostic	MAHD	Applies the results of a previously executed MRHD instruction The MAHD instruction generates a new set of encoder and servo polarities based on the observed direction of motion during the MRHD instruction.	Message
Motion Run Hookup Diagnostic	MRHD	Commands the servo module to run one of three diagnostic tests on an axis	Message Process

For more information about motion state instructions, refer to the *Motion Configuration Instructions* chapter of Logix5550 Controller Motion Instruction Set Reference Manual, publication 1756-RM007.

For more information about instruction timing, refer to Appendix E - *Instruction Timing.*

Troubleshooting

This chapter describes how to troubleshoot your ControlLogix motion control system.

1756-M02AE Module Status Using the OK Indicator

If the OK LED displays	Then the module status is	Take this action
Off	The module is not operating.	• Apply chassis power. • Verify the module is completely inserted into the chassis and backplane.
Flashing green light	The module has passed internal diagnostics, but it is not communicating axis data over the backplane.	• None, if you have not configured the module. • If you have configured the module, check the slot number in the 1756-M02AE Properties dialog box.
Steady green light	• Axis data is being exchanged with the module. • The module is in the normal operating state.	None. The module is ready for action.
Flashing red light	• A major recoverable failure has occurred. • A communication fault, timer fault, or NVS update is in progress. • The OK contact has opened.	• Check the servo fault word for the source of the error. • Clear the fault condition using the motion instructions. • Resume normal operation. • If the flashing persists, reconfigure the module.
Solid red light	• A potential non-recoverable fault has occurred. • The OK contact has opened.	• Reboot the module. • If the solid red persists, replace the module.

1756-M02AE Module Status Using the FDBK Indicator

If the FDBK LED displays	Then the module status is	Take this action
Off	The axis is not used.	• None, if you are not using this axis. • If you are using this axis, make sure you configured the module and associated an axis tag with the module.
Flashing green light	The axis is in the normal servo loop inactive state.	None. You can change the servo axis state by executing motion instructions.
Steady green light	The axis is in the normal servo loop active state.	None. You can change the servo axis state by executing motion instructions.
Flashing red light	The axis servo loop error tolerance has been exceeded.	• Correct the source of the problem. • Clear the servo fault using a fault reset instruction. • Resume normal operation.
Solid red light	An axis encoder feedback fault has occurred.	• Correct the source of the problem by checking the encoder and power connections. • Clear the servo fault using the MAFR instruction. • Resume normal operation.

1756-M02AE Module Status Using the DRIVE Indicator

If the DRIVE LED displays	Then the module status is	Take this action
Off	• The axis is not used. • The axis is a position-only axis type.	• None, if you are not using the axis or have configured it as a position-only axis. • Otherwise, make sure you have configured the module, associated an axis tag with the module, and configured the axis as a servo axis.

If the DRIVE LED displays	Then the module status is	Take this action
Flashing green light	The axis drive is in the normal disabled state.	None. You can change the servo axis state by executing a motion instruction.
Steady green light	The axis drive is in the normal enabled state.	None. You can change the servo axis state by executing a motion instruction.
Flashing red light	The axis drive output is in the Shutdown state.	• Check for faults that may have generated this state. • Execute the shutdown reset motion instruction. • Resume normal operation.

Solid red light	The axis drive is faulted.	• Check the drive status.
		• Clear the drive fault condition at the drive.
		• Execute a fault reset motion instruction.
		• Resume normal operation.
		• Check the configuration for the Drive Fault.
		  • If configured to be normally open and there is no voltage, this is the normal condition.
		  • If configured to be normally closed and there is 24V applied, this is the normal condition.

1756-M08SE SERCOS Communication Phase Status Using the CP Indicator

If the CP LED displays:	Then the module status is:
Off	The module is not operating.
Solid red light	• In Phase 0: looking for a closed ring.
Flashing red light	• In Phase 1: looking for active nodes.
Alternating Red/Green light	• In Phase 2: configuring nodes for communication.
Flashing green light	• In Phase 3: configuring device specific parameters
Solid green light	• In Phase 4: configured and active.

1756-M08SE Module Status Using the OK Indicator

If the OK LED displays:	Then the module status is:	Take this action:
Off	The module is not operating.	• Apply chassis power. • Verify the module is completely inserted into the chassis and backplane.
Flashing green light	The module has passed internal diagnostics, but has not established active communications.	• None, if you have not configured the module.
Solid green light	• Data is being exchanged. • The module is in the normal operating state.	None. The module is ready for action.
Flashing red light	• A major recoverable failure has occurred. • An NVS update is in progress.	If an NVS update is in progress, complete the NVS update. If an NVS update is not in progress: • Reboot
Solid red light	• A potential nonrecoverable fault has occurred.	• Reboot the module. • If the solid red persists, replace the module.

1756-M08SE SERCOS Ring Status

If the SERCOS Ring LED displays:	Then the ring status is:	Take this action:
Solid green light	The ring, drive, and axes are configured and are actively communicating through to the nodes on the ring.	None.
Flashing red light	The module has detected a setup or configuration fault with the ring.	Check your system setup and configuration as follows:. • Ensure drive and axes addresses are correct. • Eliminate duplicate drive or axes addresses on ring. • Remove excess axes from ring. • Make sure application program has selected the proper Ring Cycle Period and Baud Rate.

If the SERCOS Ring LED displays:	Then the ring status is:	Take this action:
Solid red light	The module has detected a hardware or installation fault with the ring.	Check your system hardware and installation as follows: • Make sure all cables are properly installed. • Make sure cable is of the correct type and length. • Make sure application program has configured the module's ring transmit level to High when using specified cables. • Make sure the drive's transmit levels are set appropriately. • Inspect cables for degradation. • Inspect drives for any faults and correct them.
Off	The module has detected no ring data on its receiver.	Check your system and installation as follows: • Make sure all cables are properly installed • Inspect cable for degradation and breakage. • Inspect drives for faults.
Flashing green light	The ring, drive, or axes are not configured.	Not a problem if the system has not been configured. If you are having trouble configuring the ring, drive, and axes: • Make sure that the application program is setup properly for the equipment in use.

Specifications and Performance

This appendix shows specifications and performance guidelines for the motion modules.

1756-M02AE Motion Module Specifications

Number of axes per chassis	Configurable
Motion commands	32
Number of axes per module	2 axes maximum
Servo loop Type Gain resolution Absolute position range Rate	 Nested PI digital position and velocity servo 32-bit floating point ±1,000,000,000 encoder counts 5 kHz
Module location	1756 ControlLogix chassis
Module keying	Electronic
Power dissipation	5.5W maximum
Backplane current	5V dc @ 700 mA 24V dc @ 2.5 mA
Encoder input Type Mode Rate Electrical interface Voltage range On state Off state Input impedance	 Incremental AB quadrature with marker 4X quadrature 4 MHz counts per second maximum Optically isolated 5V differential 3.4V to 5.0V 0V to 1.8V 531 Ohms differential
Registration inputs Type 24V input voltage Maximum Minimum on Maximum off 5V input voltage Maximum Minimum on Maximum off Input impedance 24V input 5V input Response time (position latched)	 Optically isolated, current sourcing input +24V dc nominal 26.4V 18.5V 3.5V +5V dc nominal 5.5V 3.7V 1.5V 1.2 kOhms 9.5 kOhms 1µs

All other inputs	
Type	Optically isolated, current sinking input
Input voltage	+24V dc nominal
Maximum	26.4V
Minimum on	17.0V
Maximum off	8.5V
Input impedance	7.5 kOhms
Servo output	
Type	Analog voltage
Isolation	200 kOhms
Voltage range	±10V
Voltage resolution	16 bits
Load	5.6 kOhms resistive minimum
Maximum offset	25 mV
Gain error	±4%
All other outputs	
Type	Solid-state isolated relay contacts
Operating voltage	+24V dc nominal
Maximum	26.4V
Operating current	75 mA
RTB keying	User-defined
Field wiring arm	36-position RTB (1756-TBCH or -TBS6H)[1]
RTB screw torque (cage clamp)	5lb-in. (0.5 Nm) maximum
Conductors Wire size Category	 22 gauge (3.1 mm^2) minimum to copper[1] 3/64 inch (1.2 mm) insulation maximum 1[2,3]
Screwdriver blade width for RTB	1/8 inch (3.2 mm) maximum
Environmental conditions Operating temperature Storage temperature Relative humidity	 0 to 60ºC (32 to 140ºF) -40 to 85ºC (-40 to 185ºF) 5 to 95% noncondensing
Agency certification (when product or packaging is marked)	  Class 1, Division 2, hazardous location marked for all applicable directives

[1] Maximum wire size will require the extended depth RTB housing (1756-TBE).

[2] Use this conductor category information for planning conductor routing as described in the system level installation manual.

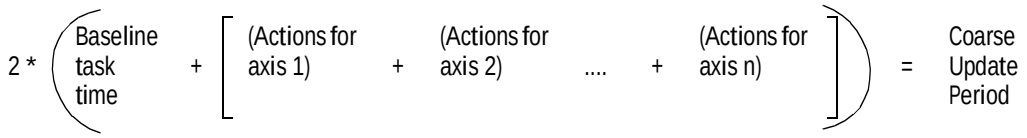
[3] Refer to *Industrial Automation Wiring and Grounding Guidelines*, publication number 1770-4.1.

1756-M08SE Motion Module Specifications

Number of Nodes	8 axes maximum
Module location	1756 ControlLogix chassis
Module keying	Electronic
Power dissipation	3.2W maximum
Backplane current	5.1V dc @ 600 mA 24V dc @ 2.5 mA
Environmental Specifications Operational Temperature Storage temperature Relative Humidity	 0 to 60° C (32-140° F) -40 to 85ºC (-40 to 185ºF) 5 to 95% non condensing IEC 68-2-30
Shock Operating Non-operating	 30g peak acceleration, 11(±2)ms pulse width 50g peak acceleration, 11(±2)ms pulse width
Vibration	2g @ 10-500Hz per IEC 68-2-6
Plastic Fiber Optic Specifications Transmission Range Core Diameter Cladding Diameter Cable Attenuation Operating Temperature Connector Bend Radius	 1-32 meters 980µm ± 60µm 1000µm ± 60µm 140 dB/km @ 650nm -55 to 85° C F-SMA standard screw-type connector 2.5 cm
Glass Fiber Optic Specifications Transmission Range Core Diameter Cladding Diameter Cable Attenuation Operating Temperature Connector Bend Radius	 1-200 meters 200µm ± 4µm 230µm +0/−10µm 6.0 dB/km @ 820nm -20 to 85° C F-SMA standard screw-type connector 2.5cm
SERCOS SERCOS Class Data Rate Operating cycle	 Class B (Position or Velocity) 4 Mbits per second 1 ms for 1-4 axes 2 ms for 5-8 axes
Agency certification (when product or packaging is marked)	 c UL us UL508 – Industrial Control Equipment UL1604 – Class I, Division 2, Groups A, B, C, D Hazardous location CE marked for all applicable directives

Coarse Update Period Calculations

To calculate the coarse update period for the number axes in your application, you can use the following formula:



The result of the above calculation must be divided by 1000 rounded up to the nearest milliseconds.

You can use the sample calculation worksheet in this section to determine your coarse update period. To determine the values for your equation, refer to the following table.

The coarse update period can have a significant bearing on the quality of motion control for a given application. If an application requires commanded acceleration or deceleration times that are comparable to the coarse update period, significant velocity and position overshoot can occur as the axis attempts to follow the command profile. The amount of velocity overshoot can be calculated as the product of the acceleration or deceleration rate and the coarse update period. As a general rule of thumb the acceleration and deceleration times for a motion application should be at least 10 times the coarse update period to avoid significant velocity or position overshoot.

For example, an application that requires an axis to accelerate and decelerate to full speed in 100 milliseconds is best handled by choosing a coarse update period of 10 milliseconds or shorter. Be sure to check what the minimum coarse update period is for the associated controller before selecting the coarse update period. The minimum coarse update period can be determined based on the number of axes and the worst case motion activity according to the execution time tables provided.

Understanding Action Timing

Every action performed by an axis requires an amount of motion task time. For example to perform a trapezoidal move requires 310 µs.

The following table shows execution times for common motion actions.

State/Action (Δ = per axis)	Typical Execution Time (in µs)
Motion Task Overhead	190
Servo Axis Δ	205
Virtual Axis Δ	175
Consumed Axis Δ*	900*
Servo On Δ	40
Trap Move Δ	310
S–Curve Move Δ	435
Trap Jog Δ	210
S–Curve Jog Δ	340
Gearing (Actual) Δ	290
Clutch Δ	105
Gearing (Command) Δ	125
Clutch Δ	110
Position Camming (Actual,linear) Δ	495
Position Camming (Actual,cubic) Δ	550
Position Camming (Command,linear) Δ	295
Position Camming (Command,cubic) Δ	380
Time Camming (linear) Δ	260
Time Camming (cubic) Δ	320

* The task time of a consumed axis can be lowered to approximately 650 µs by using a consumer coarse update period that is an integer multiple of the producer coarse update period. The value of 900µs reported above is the worst case (producer/consumer update ratio = 2/3). Using 2/4 would yield ~ 650µs.

Using the Sample Calculations Worksheet

You can use this sample calculation worksheet to determine the coarse update period for the group in your application.

1. Complete the following table.

System Describe the type of system you are using.			
	1	Enter baseline task time = Motion Task Overhead + (# of Servo Axes * Servo Axis Δ) + (# of Virtual Axes * Virtual Axis Δ) + (# of Consumed Axes * Consumed Axis Δ)	µs

2. For each axis in your application, use the following table to determine the action value for each axis.

Actions If you are using an action, enter its execution time from the Action Timing table. If you are not using an action, enter zero (0).			
	2	Servo on	µs
	3	Trapezoidal move	µs
	4	S-curve move	µs
	5	Trapezoidal jog	µs
	6	S-curve jog	µs
	7	Actual gear	µs
	8	Command gear	µs
	9	Actual PCAM	µs
	10	Command PCAM	µs
	11	TCAM	µs
	12	Total Axis Action Value (Add lines 2 through 11). Place total here.	µs

3. Calculate the Total Axes Action Value by adding all of the Total Axis Action Values (line 12) for all axes in the group.

4. The Recommended Coarse Update Period is calculated by: (Baseline Task Time(line 1) + Total of Axes Action Values(line12)) * 2 / 1000 then round up to the milliseconds to get your Recommended Coarse Update Period.

Sample Calculation

You have the following situation:

- You have a system consisting of 2 modules, 3 Servo axes, and 1 virtual axis.
- You are turning the servo on and performing a trapezoidal move for axis1.
- You are turning the servo on and doing an S-Curve jog for axis2.
- Axis3 is a virtual axis and you are doing a command gear for on it.
- You are turning the servo on and doing a cubic command position cam for axis4.

1. Complete the following table.

System Describe the type of system you are using.			
	1	Enter baseline task time = Motion Task Overhead + (3 * Servo Axis) + (1 * Virtual Axis).	980 µs

2. For each axis in your application, use the following tables to determine the action value for each axis.

Axis 1:

Actions If you are using an action, enter its execution time from the Action Timing table. If you are not using an action, enter zero (0).			
	2	Servo on	40 µs
	3	Trapezoidal move	310 µs
	4	S-curve move	0 µs
	5	Trapezoidal jog	0 µs
	6	S-curve jog	0 µs
	7	Actual gear	0 µs
	8	Command gear	0 µs
	9	Actual PCAM	0 µs
	10	Command PCAM	0 µs
	11	TCAM	0 µs
	12	Total Axis Action Value (Add lines 2 through 11).	350 µs

Axis 2:

Actions If you are using an action, enter its execution time from the Action Timing table. If you are not using an action, enter zero (0).		
2	Servo on	40 µs
3	Trapezoidal move	0 µs
4	S-curve move	0 µs
5	Trapezoidal jog	0 µs
6	S-curve jog	340 µs
7	Actual gear	0 µs
8	Command gear	0 µs
9	Actual PCAM	0 µs
10	Command PCAM	0 µs
11	TCAM	0 µs
12	Total Axis Action Value (Add lines 2 through 11).	380 µs

Axis 3:

Actions If you are using an action, enter its execution time from the Action Timing table. If you are not using an action, enter zero (0).		
2	Servo on	0 µs
3	Trapezoidal move	0 µs
4	S-curve move	0 µs
5	Trapezoidal jog	0 µs
6	S-curve jog	0 µs
7	Actual gear	0 µs
8	Command gear	125 µs
9	Actual PCAM	0 µs
10	Command PCAM	0 µs
11	TCAM	0 µs
12	Total Axis Action Value (Add lines 2 through 11).	125 µs

Axes 4:

Actions If you are using an action, enter its execution time from the Action Timing table. If you are not using an action, enter zero (0).		
2	Servo on	40 µs
3	Trapezoidal move	0 µs
4	S-curve move	0 µs
5	Trapezoidal jog	0 µs
6	S-curve jog	0 µs
7	Actual gear	0 µs
8	Command gear	0 µs
9	Actual PCAM	0 µs
10	Command PCAM	380 µs
11	TCAM	0 µs
12	Total Axis Action Value (Add lines 2 through 11).	420 µs

3. The calculated coarse rate for this application is

Baseline task time (line 2)	980 µs
Total Axis Action Value for axis 1 (line 12)	350 µs
Total Axis Action Value for axis 2 (line 12)	380 µs
Total Axis Action Value for axis 3 (line 12)	125 µs
Total Axis Action Value for axis 4 (line 12)	420 µs
Total Axes Action Value (add all of the above)	2255 µs
TOTAL (Total Axes Action Value * 2)	4510 µs
Recommended Coarse Update Period = (TOTAL / 1000) rounded up to nearest ms	5 ms

Output Cam Timing For 1756 Controller

The impact on the coarse update period is primarily dependent on three factors:

- total number of output cam array entries
- total number of output compensation array entries
- number of outputs compensation array entries with non-zero latch/unlatch time delay values

The following formulas ballpark additional coarse update time required for each unique Output Cam execution target being used.

In the following formulas:

- A = number of Output Cam array elements
- B = number of Output Compensation array elements
- C = number of Output Compensation array elements with Latch/Unlatch non-zero delay values

All values are expressed in micro-seconds.

For the 1756-L50 Controller

1200 + (A * 130) + (B * 140) + (C * 90)

For the 1756-L53 Controller

1000 + (A * 100) + (B * 110) + (C * 60)

Loop and Interconnect Diagrams

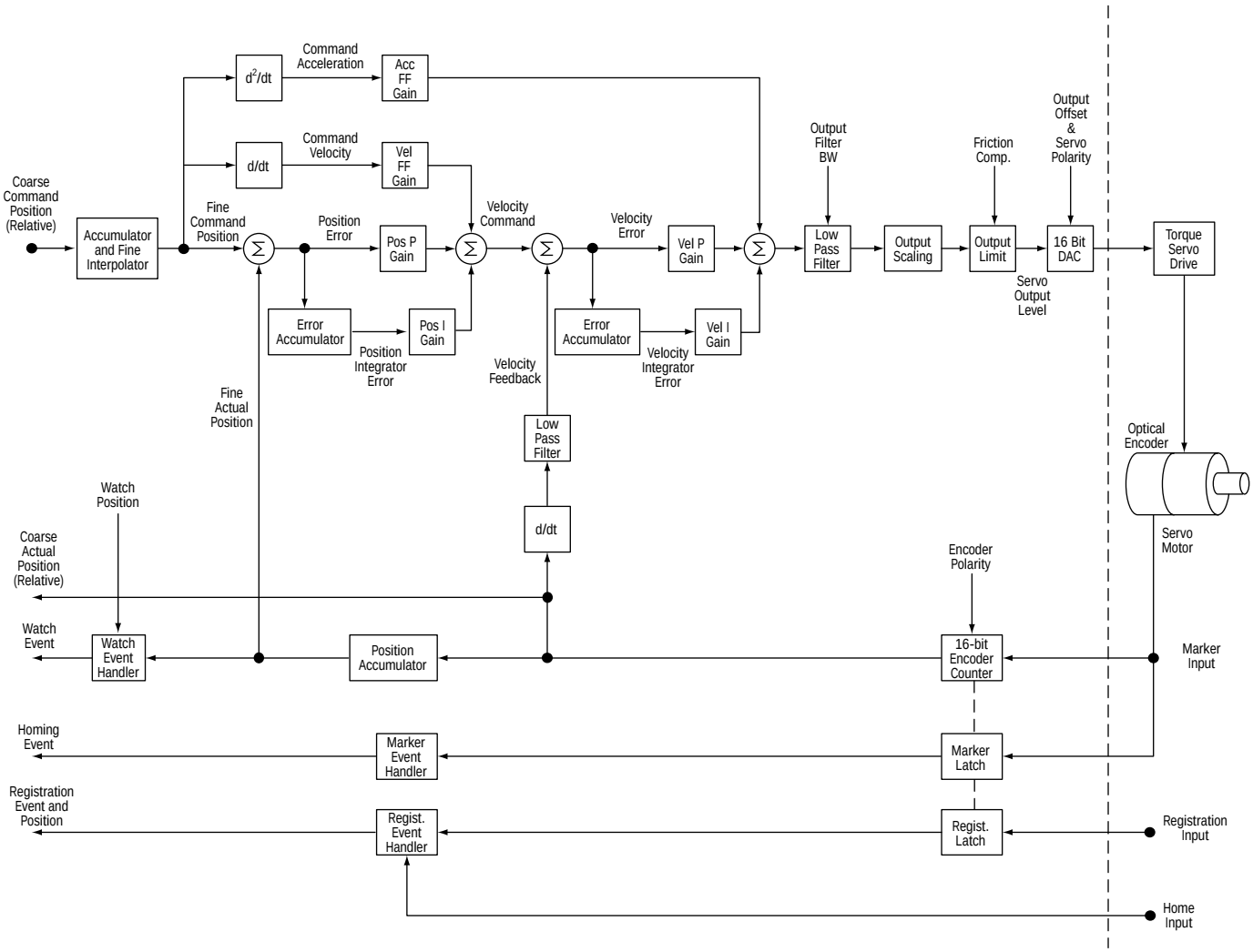
This appendix shows the loop interconnect diagrams for common motion configurations.

Understanding Block Diagrams

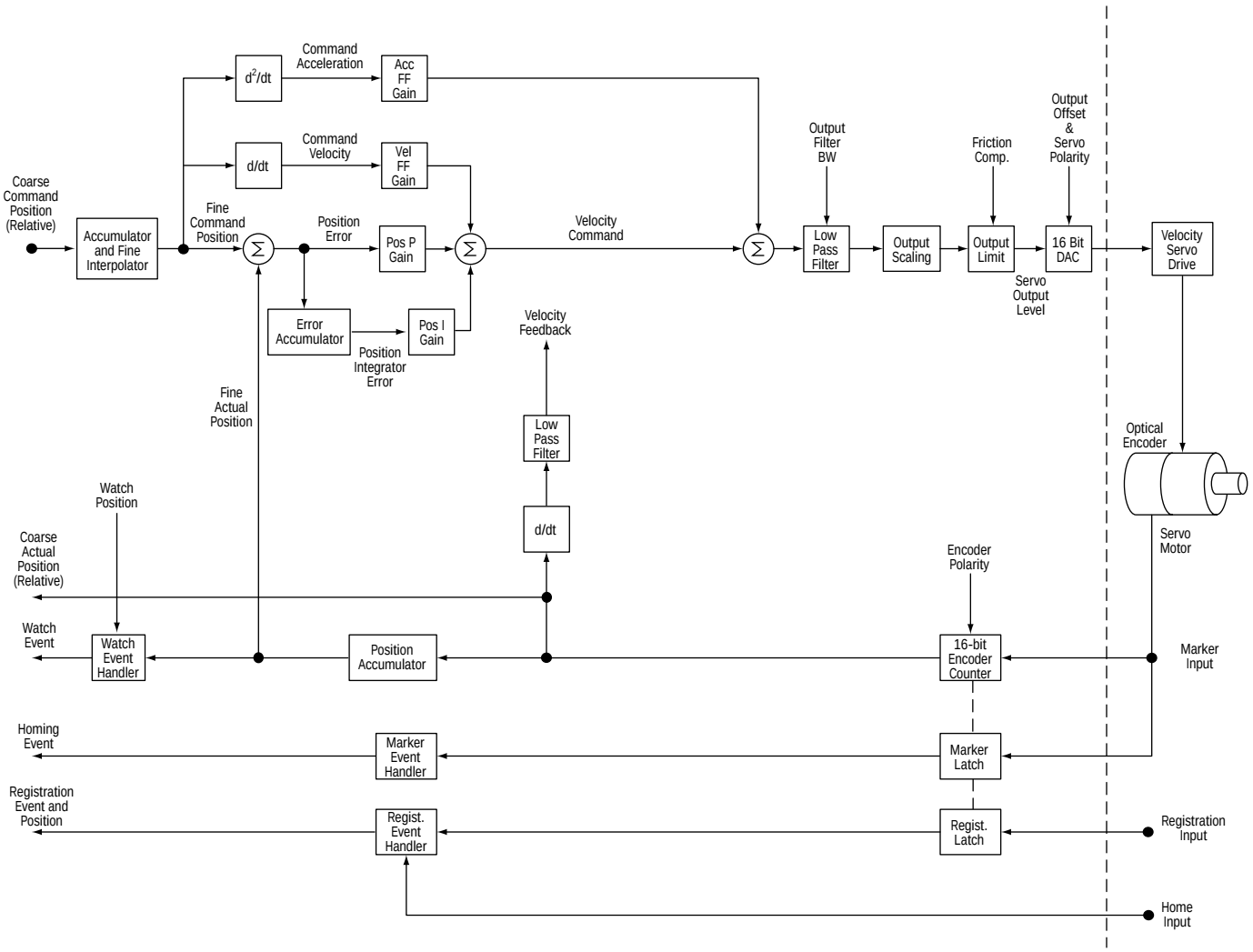
The control block diagrams in this section use the following terms for motion attributes.

Diagram term	Motion attribute name (as used in the GSV and SSV instructions)
Acc FF Gain	AccelerationFeedforwardGain
Vel FF Gain	VelocityFeedforwardGain
Pos P Gain	PositionProportionalGain
Pos I Gain	PositionIntegralGain
Vel P Gain	VelocityProportionalGain
Vel I Gain	VelocityIntegralGain
Output Filter BW	OutputFilterBandwidth
Output Scaling	OutputScaling
Friction Comp	FrictionCompensation
Output Limit	OutputLimit
Output Offset	OutputOffset
Position Error	PositionError
Position Integrator Error	PositionIntegratorError
Velocity Error	VelocityError
Velocity Integrator Error	VelocityIntegratorError
Velocity Feedback	VelocityFeedback
Velocity Command	VelocityCommand
Servo Output Level	ServoOutputLevel
Registration Position	RegistrationPosition
Watch Position	WatchPosition

Using a 1756-M02AE Module With a Torque Servo Drive

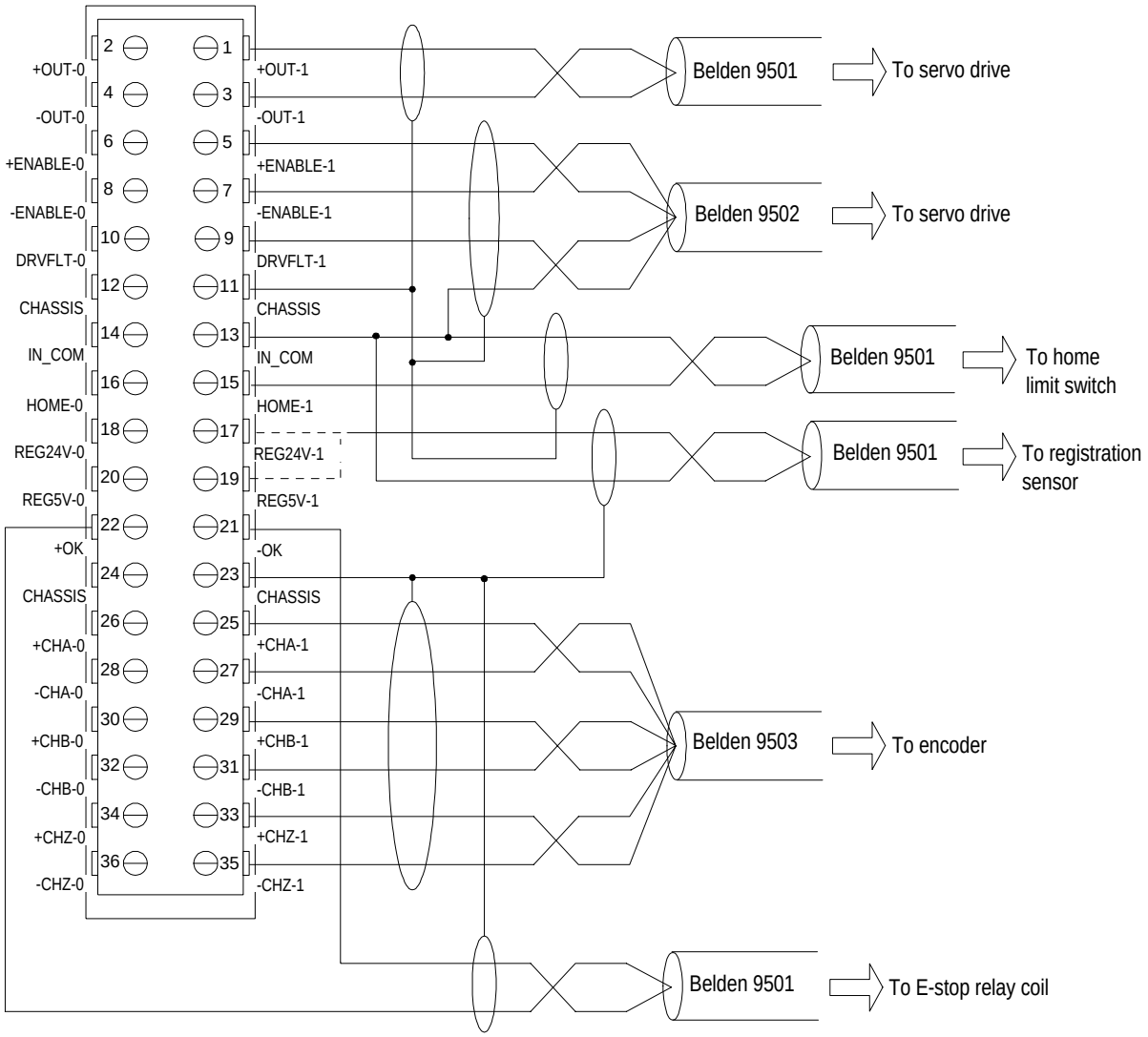


Using a 1756-M02AE Module With a Velocity Servo Drive



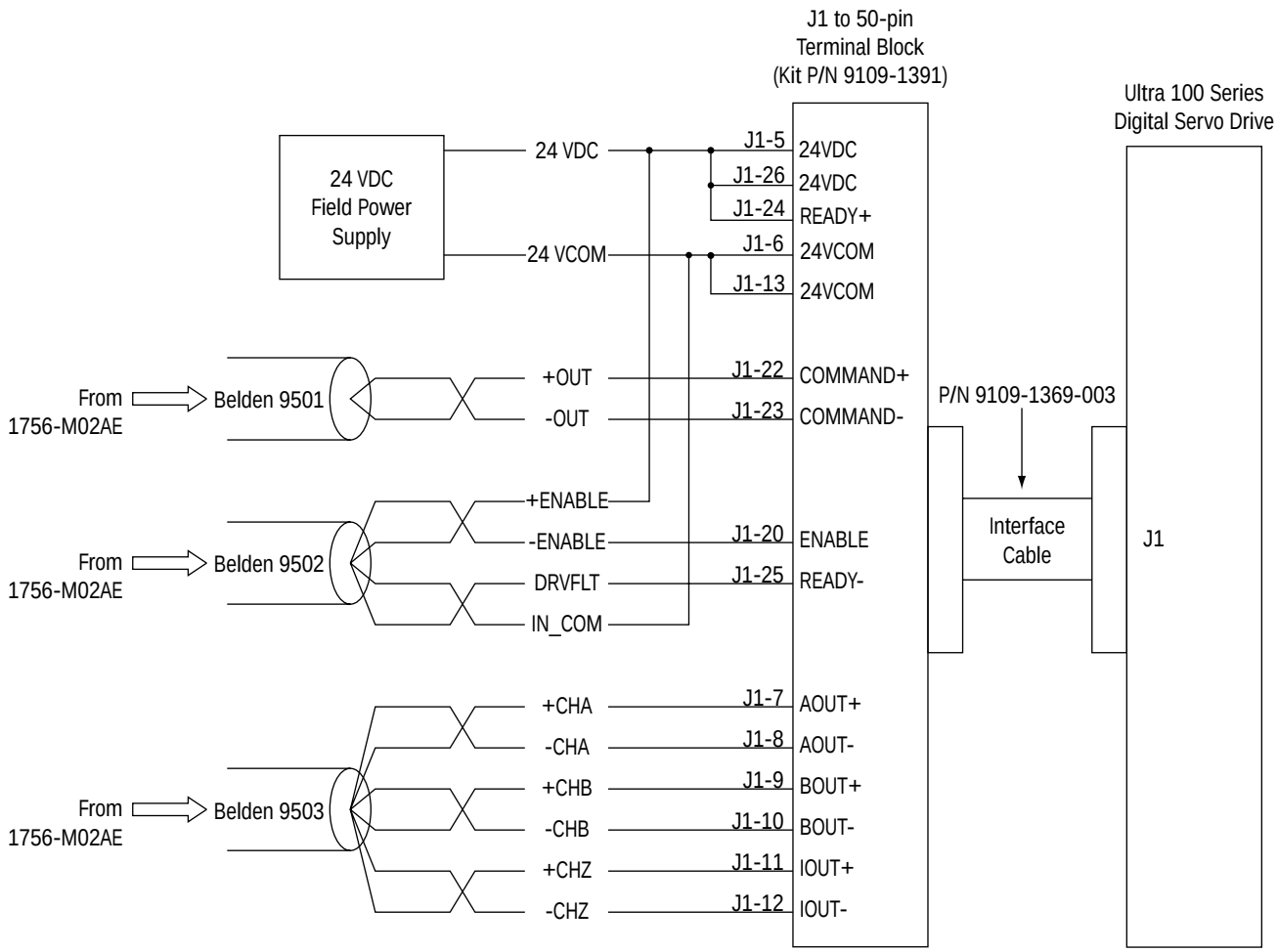
Understanding Wiring Diagrams

Wiring to a Servo Module RTB



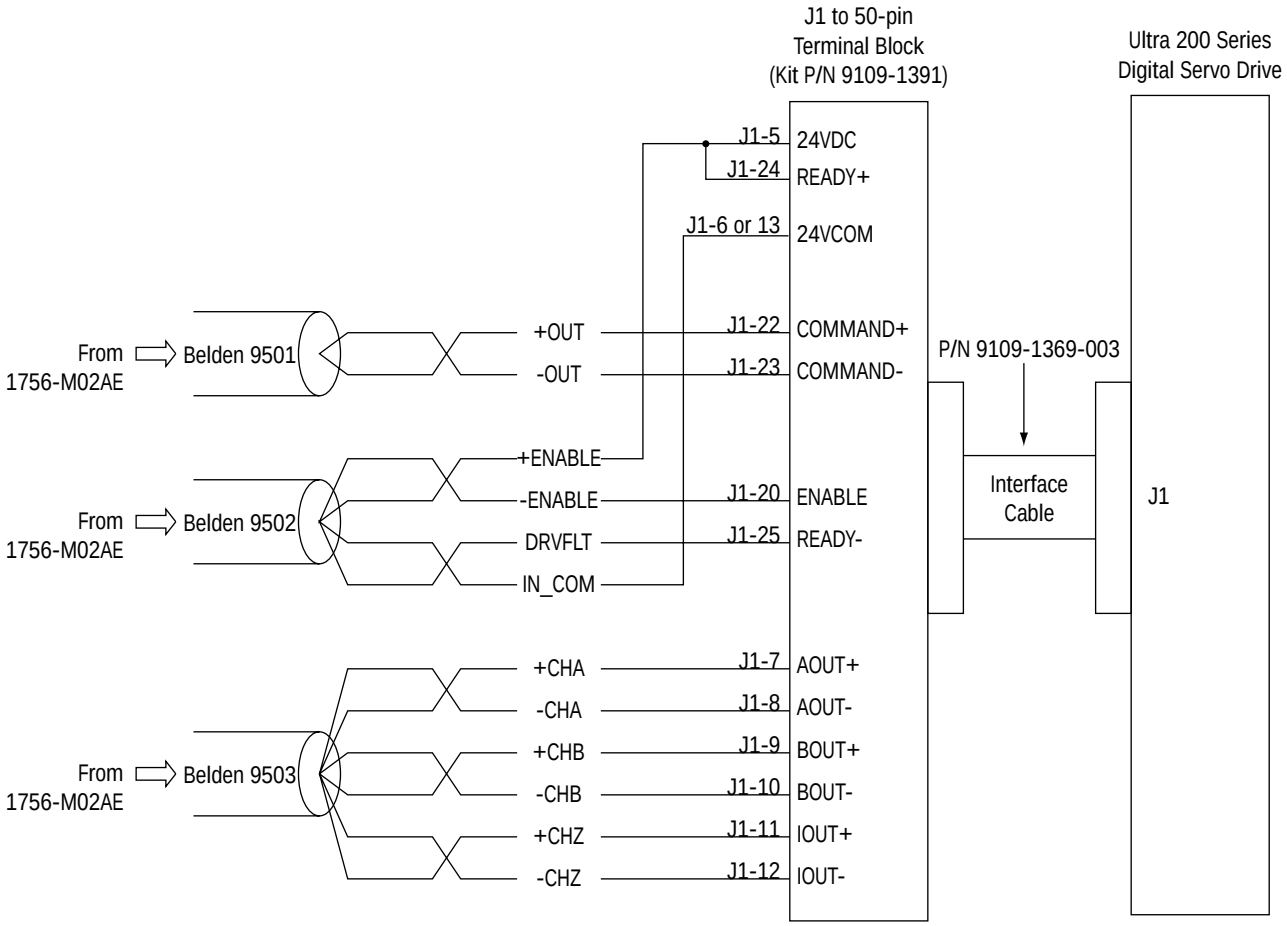
This is a general wiring example illustrating Axis 1 wiring only. Other configurations are possible with Axis 0 wiring identical to Axis 1.

Wiring to an Ultra 100 Series Drive



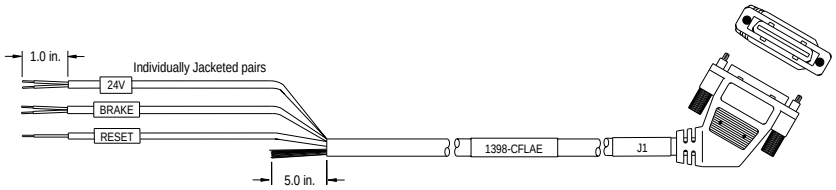
This is a general wiring example only. Other configurations are possible. For more information, refer to the Ultra 100 Series Drive Installation Manual, publication number 1398-5.2.

Wiring to an Ultra 200 Series Drive



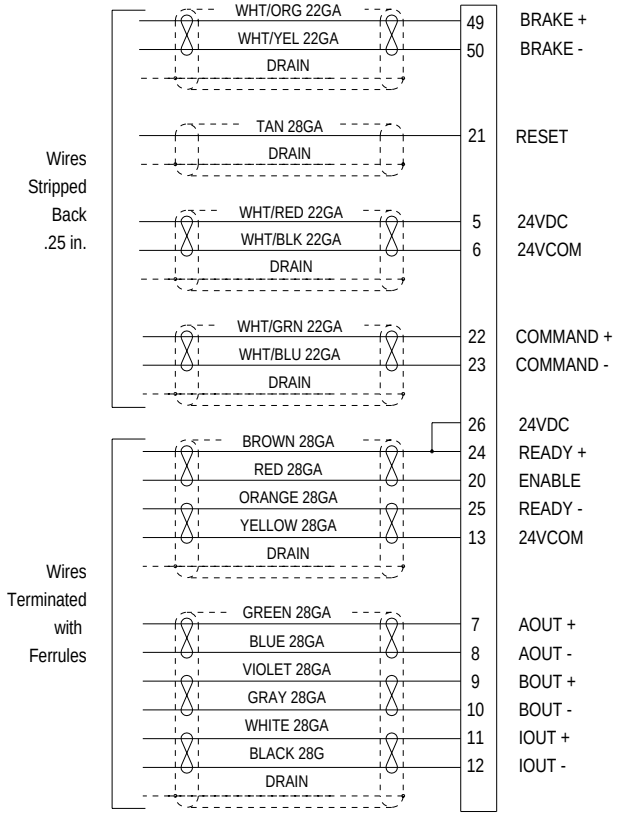
This is a general wiring example only. Other configurations are possible. For more information, refer to the Ultra 200 Series Drive Installation Manual, publication number 1398-5.0.

1398-CFLAExx Cable Diagram

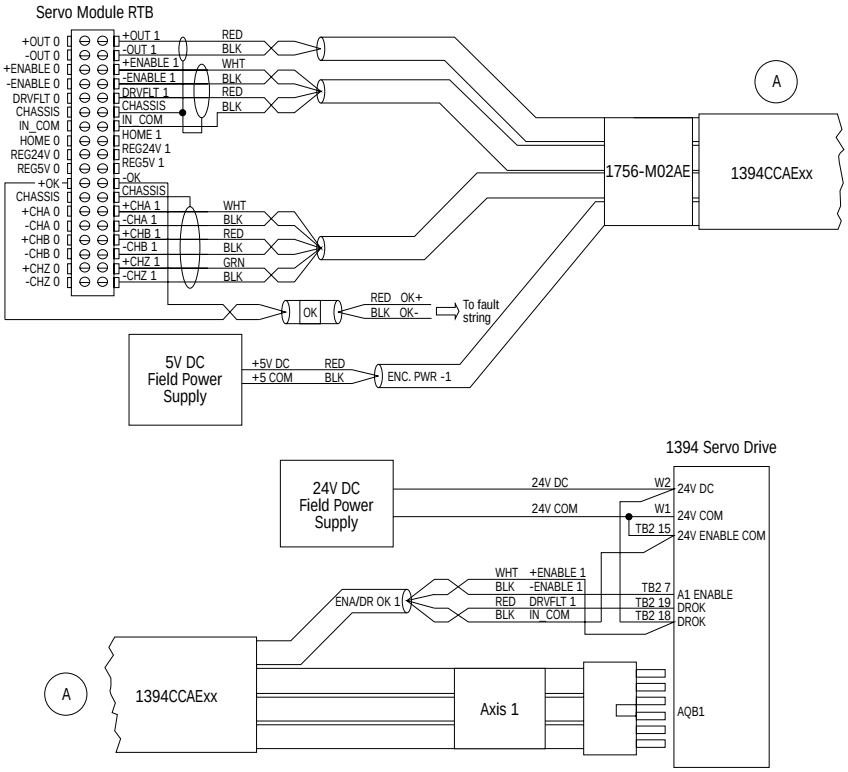


The 1398-CFLAE Cable is available in 10, 25, and 50 foot lengths.

Pinouts for 1398-CFLAExx Cable



Wiring to a 1394 Servo Drive (in Torque Mode only)



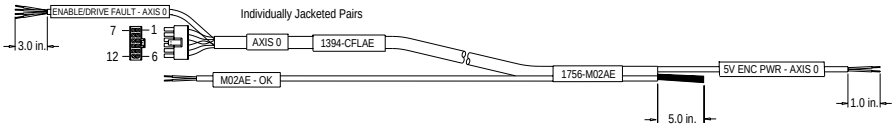
The wiring diagram illustrates Axis 1 wiring only. Other configurations are possible.

The 1394CCAE*xx* cable is wired to connect to torque command reference input pins.

An external +5V power supply is required to power the encoder driver circuit of the 1394 servo drive. Because this connection is shared by all four axis encoder driver circuits, only one connection is needed to the +5V field supply.

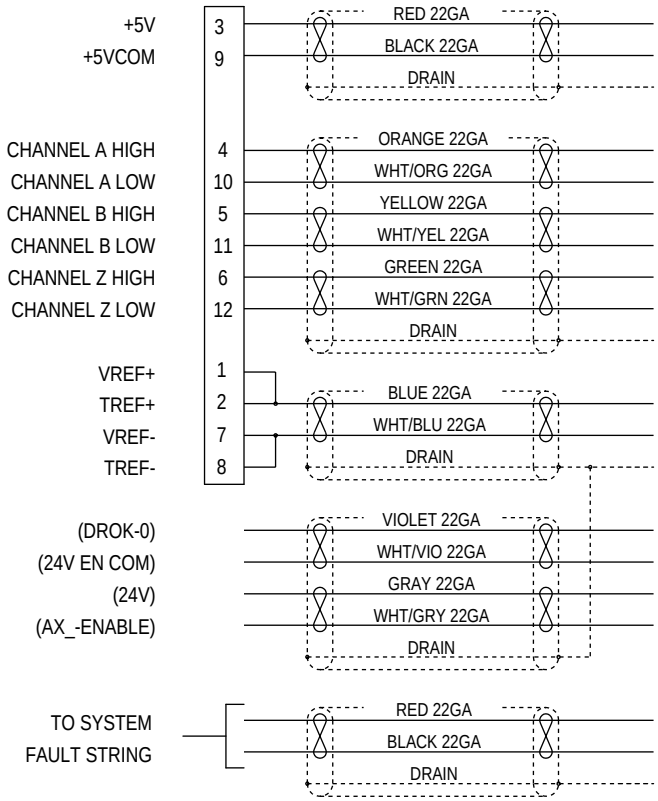
The *xx* in the cable number is the length of the cable. Options are 5, 10, 25, and 50 feet.

The 1394-CFLAE*xx* Cable Wiring Diagram



The 1394-CFLAE cable is available in 1, 3, 8, and 15 meter lengths.

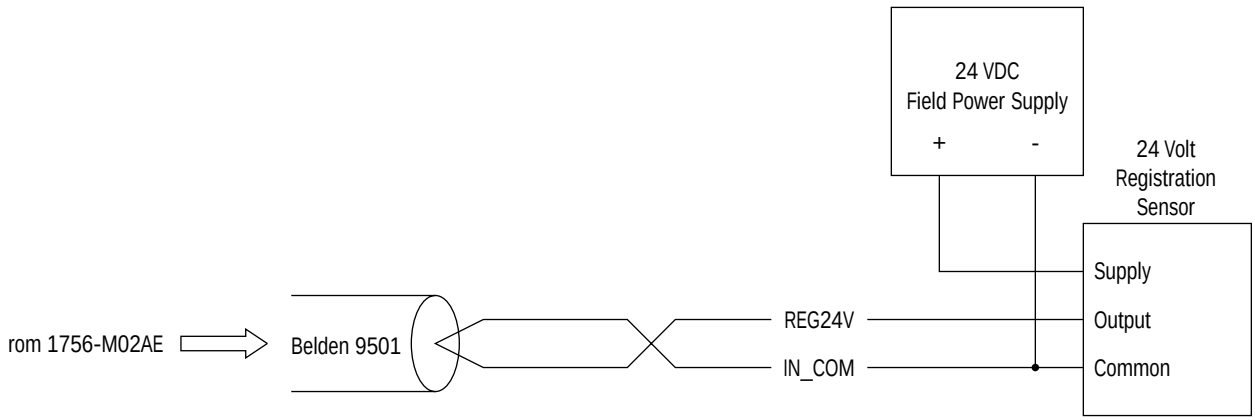
Pinouts for the 1394-CFLAE



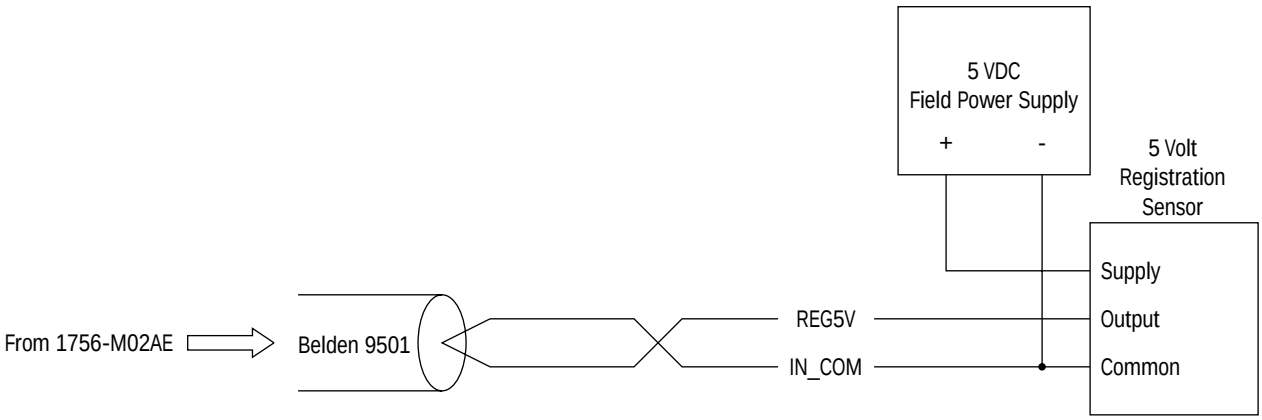
Wiring Registration Sensors

The registration inputs to the servo module can support 24V or 5V registration sensors. These inputs should be wired to receive source current from the sensor. Current sinking sensor configurations are not allowed because the registration input common (IN_COM) is shared with the other 24V servo module inputs.

24V Registration Sensor

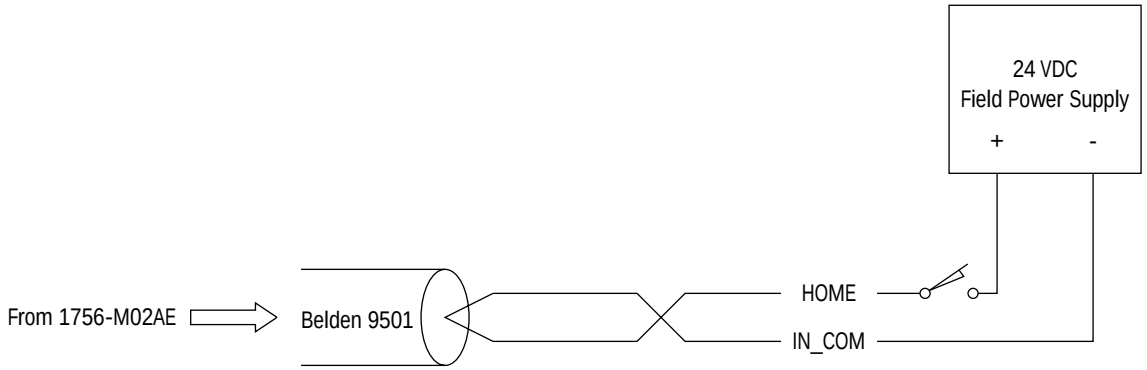


5V Registration Sensor



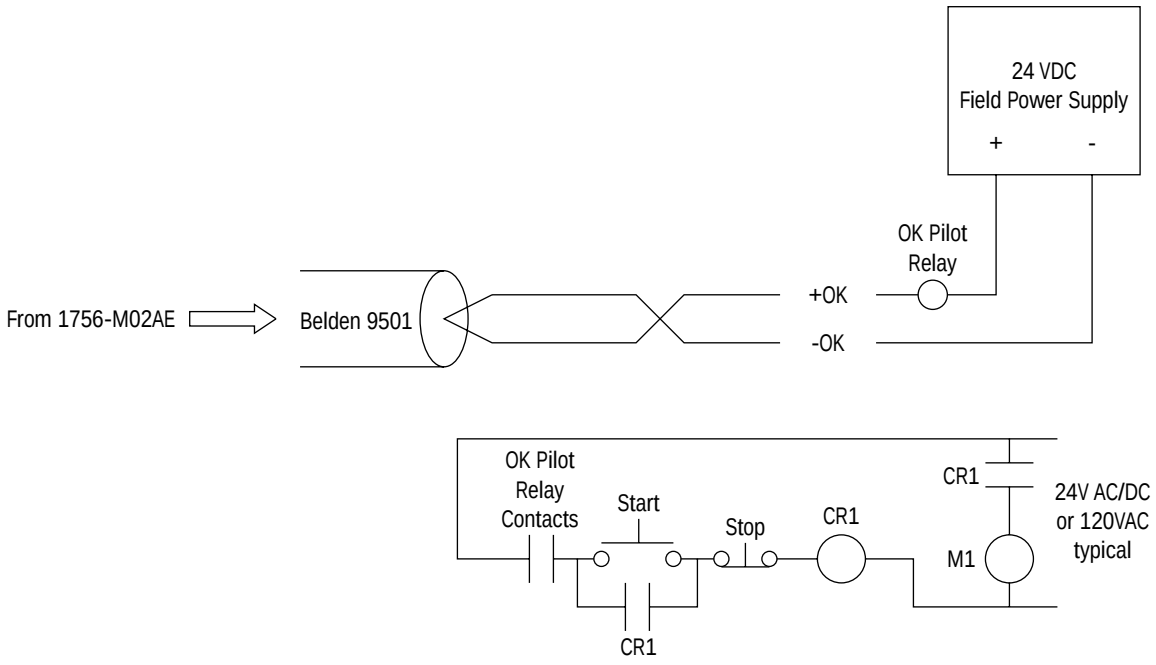
Wiring the Home Limit Switch Input

The home limit switch inputs to the servo module are designed for 24V nominal operation. These inputs should be wired for current sourcing operation.



Wiring the OK Contacts

A set of isolated solid-state OK relay contacts is provided for optional interface to an E-stop string, which controls power to the associated drives. The OK contacts are rated to drive an external 24V pilot relay (for example, Allen-Bradley 700-HA32Z24) whose contacts can be incorporated into the E-Stop string as shown below.


24 VDC
Field Power Supply
+
-
OK Pilot
Relay
From 1756-M02AE
Belden 9501
+OK
-OK
OK Pilot
Relay
Contacts
Start
Stop
CR1
CR1
M1
CR1
24V AC/DC
or 120VAC
typical

The Motion Control Structures

This appendix shows the structures for the AXIS, MOTION_GROUP, MOTION_INSTRUCTION, CAM, and CAM_PROFILE data tags.

AXIS Structures

There are four axis related data types that each have their own structure. The four types are: Axis Consumed, Axis Servo, Axis Servo Drive, and Axis Virtual. The following sections describe the structures for each of these axis data types.

AXIS_CONSUMED Structure

A Consumed Axis is a conduit for axis motion data produced by a motion axis on another Logix processor.

The Axis_Consumed structure has the following status attributes:

Mnemonic:	Data Type:	Description:			
MotionStatus	DINT	The motion status bits for your axis.			
		Bit:	Number:	Data Type:	Description:
		AccelStatus	00	BOOL	Acceleration Status
		DecelStatus	01	BOOL	Deceleration Status
		MoveStatus	02	BOOL	Move Status
		JogStatus	03	BOOL	Jog Status
		GearingStatus	04	BOOL	Gearing Status
		HomingStatus	05	BOOL	Homing Status
		StoppingStatus	06	BOOL	Stopping Status
		AxisHomedStatus	07	BOOL	Homed Status
		PositionCamStatus	08	BOOL	Position Cam Status
		TimeCamStatus	09	BOOL	Time Cam Status
		PositionCamPendingStatus	10	BOOL	Position Cam Pending Status
		TimeCamPendingStatus	11	BOOL	Time Cam Pending Status
		GearingLockStatus	12	BOOL	Gearing Lock Status
		PositionCamLockStatus	13	BOOL	Position Cam Lock Status
		TimeCamLockStatus	14	BOOL	Time Cam Lock Status
		MasterOffsetMoveStatus	15	BOOL	Master Offset Move Status

Mnemonic:	Data Type:	Description:			
AxisStatus	DINT	The status bits for your axis			
		Bit:	Number:	Data Type:	Description:
		ServoActStatus	00	BOOL	Servo Action Status
		DriveEnableStatus	01	BOOL	Drive Enable Status
		ShutdownStatus	02	BOOL	Axis Shutdown Status
		ConfigUpdateInProcess	03	BOOL	Configuration Update in Process
AxisFault	DINT	The axis faults for your axis:			
		Bit:	Number:	Data Type:	Description:
		PhysicalAxisFault	00	BOOL	Physical Axis Fault
		ModuleFault	01	BOOL	Module Fault
		ConfigFault	02	BOOL	Configuration Fault
AxisEvent	DINT	The event status for your axis:			
		Bit:	Number:	Data Type:	Description:
		WatchEvArmStatus	00	BOOL	Watch Event Armed Status
		WatchEvStatus	01	BOOL	Watch Event Status
		RegEvArmStatus	02	BOOL	Registration Event 1 Armed Status
		RegEvStatus	03	BOOL	Registration Event 1 Status
		RegEv2ArmStatus	04	BOOL	Registration Event 2 Armed Status
		RegEv2Status	05	BOOL	Registration Event 2 Status
		HomeEvArmStatus	06	BOOL	Home Event Armed Status
		HomeEvStatus	07	BOOL	Home Event Status
ActualPosition	REAL	Actual Position in Position Units			
StrobeActualPosition	REAL	Strobe Actual Position in Position Units			
StartActualPosition	REAL	Start Actual Position in Position Units			
AverageVelocity	REAL	Average Velocity in Position Units / Sec			
ActualVelocity	REAL	Actual Velocity in Position Units / Sec			
ActualAcceleration	REAL	Actual Acceleration in Position Units / Sec2			
WatchPosition	REAL	Watch Position in Position Units			
RegistrationPosition	REAL	Registration 1 Position in Position Units			
Registration2Position	REAL	Registration 2 Position in Position Units			
Registration1Time	DINT	Registration 1 Time as CST time in microseconds			
Registration2Time	DINT	Registration 2 Time as CST time in microseconds			
InterpolationTime	DINT	Interpolation Time as CST time in microseconds			
InterpolatedActualPosition	REAL	Interpolated Actual Position in Position Units			
MasterOffset	REAL	Master Offset in Master Position Units			
StrobeMasterOffset	REAL	Strobe Master Offset in Master Position Units			
StartMasterOffset	REAL	Start Master Offset in Master Position Units			
CommandPosition	REAL	Command Position in Position Units			
StrobeCommandPosition	REAL	Strobe Command Position in Position Units			
StartCommandPosition	REAL	Start Command Position in Position Units			
CommandVelocity	REAL	Command Velocity in Position Units / Sec			

Mnemonic:	Data Type:	Description:
CommandAcceleration	REAL	Command Acceleration in Position Units / Sec2
InterpolatedCommandPosition	REAL	Interpolated Command Position in Position Units
AccelStatus	BOOL	Set if the axis is currently being commanded to accelerate.
DecelStatus	BOOL	Set if the axis is currently being commanded to decelerate.
MoveStatus	BOOL	Set if a Move motion profile is currently in progress. Cleared when the Move is complete or is superseded by some other motion operation.
JogStatus	BOOL	Set if a Jog motion profile is currently in progress. Cleared when the Jog is complete or is superseded by some other motion operation.
GearingStatus	BOOL	Set if the axis is a slave that is currently Gearing to another axis. Cleared when the gearing operation is stopped or is superseded by some other motion operation.
HomingStatus	BOOL	Set if a Home motion profile is currently in progress. Cleared when the homing operation is stopped or is superseded by some other motion operation.
StoppingStatus	BOOL	Set if there is a stopping process currently in progress. Cleared when the stopping process is complete. Note: The stopping process is used to stop an axis (initiated by an MAS, MGS, Stop Motion fault action, or mode change).
AxisHomedStatus	BOOL	Cleared at power-up or reconnection. Set by the MAH instruction upon successful completion of the configured homing sequence, and later cleared when the axis enters the shutdown state.
PositionCamStatus	BOOL	Set if a Position Cam motion profile is currently in progress. Cleared when the Position Cam is complete or is superseded by some other motion operation.
TimeCamStatus	BOOL	Set if a Time Cam motion profile is currently in progress. Cleared when the Time Cam is complete or is superseded by some other motion operation.
PositionCamPendingStatus	BOOL	Set if a Position Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MAPC instruction with Pending execution selected. This bit is cleared when the current position cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the position cam profile completes, or is superseded by some other motion operation.
TimeCamPendingStatus	BOOL	Set if a Time Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MATC instruction with Pending execution selected. This bit is cleared when the current time cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the time cam profile completes, or is superseded by some other motion operation.
GearingLockStatus	BOOL	Set whenever the slave axis is locked to the master axis in a gearing relationship according to the specified gear ratio. The clutch function of the gearing planner is used to ramp an axis up, or down, to speed in a gearing process (MAG with Clutch selected). This bit is cleared during the intervals where the axis is clutching.
PositionCamLockStatus	BOOL	Set whenever the master axis satisfies the starting condition of a currently active Position Cam motion profile. The starting condition is established by the Start Control and Start Position parameters of the MAPC instruction. This bit is bit is cleared when the current position cam profile completes, or is superseded by some other motion operation. In uni-directional master direction mode, the Position Cam Lock Status bit is cleared when moving in the “wrong” direction and sets when moving in the “correct” direction.
MasterOffsetMoveStatus	BOOL	Set if a Master Offset Move motion profile is currently in progress. This bit is cleared when the Master Offset Move is complete or is superseded by some other motion operation.
ServoActStatus	BOOL	Set when the associated axis is under servo control. Cleared when servo action is disabled.

Mnemonic:	Data Type:	Description:
DriveEnableStatus	BOOL	Set when the Drive Enable output of the associated physical axis is currently enabled. Cleared when physical servo axis Drive Enable output is currently disabled.
ShutdownStatus	BOOL	Set when the associated axis is currently in the Shutdown state. Cleared when the axis is transitioned from the Shutdown state to another state.
ConfigUpdateInProcess	BOOL	The Configuration Update Status Bits attribute provides a method for monitoring the progress of one or more specific module configuration attribute updates initiated by either a Set Attribute List service (which is internal to the firmware) or an SSV in the user program. When such an update is initiated, the ControlLogix processor sets this bit. This bit will remain set until the Set Attribute List reply comes back from the servo module indicating that the data update process was successful. Thus the Configuration Update Status Bits attribute provides a method of waiting until the servo configuration data update to the connected motion module is complete before starting a dependent operation.
PhysicalAxisFault	BOOL	Set when one or more fault conditions have been reported by the physical axis. The specific fault conditions can then be determined through access to the fault attributes of the associated physical axis. A PhysicalAxisFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ModuleFault	DINT	Set when a serious fault has occurred with the motion module associated with the selected axis. Usually a module fault affects all axes associated with the motion module. A module fault generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition. A ModuleFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ConfigFault	BOOL	Set when an update operation targeting an axis configuration attribute of an associated motion module has failed. Specific information concerning the Configuration Fault may be found in the Attribute Error Code and Attribute Error ID attributes associated with the motion module. A ConfigFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ControlSyncFault	BOOL	Set when the Logix controller detects that several position update messages in a row from the motion module have been missed due to a failure of the synchronous communications connection. This condition results in the automatic shutdown of the associated servo module. The Logix controller is designed to “ride-through” a maximum of four missed position updates without issuing a fault or adversely affecting motion in progress. Missing more than four position updates in a row constitutes a problematic condition that warrants shutdown of the servo module. This fault bit is cleared when the connection is reestablished.
WatchEvArmStatus	BOOL	Set when a watch event has been armed through execution of the MAW (Motion Arm Watch) instruction. Cleared when either a watch event occurs or a MDW (Motion Disarm Watch) instruction is executed.
WatchEvStatus	BOOL	Set when a watch event has occurred. Cleared when either another MAW (Motion Arm Watch) instruction or a MDW (Motion Disarm Watch) instruction is executed.
RegEvArmStatus	BOOL	Set when a registration checking has been armed for registration input 1 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEvStatus	BOOL	Set when a registration event has occurred on registration input 1. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEv2ArmStatus	BOOL	Set when a registration checking has been armed for registration input 2 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.

Mnemonic:	Data Type:	Description:
RegEv2Status	BOOL	Set when a registration event has occurred on registration input 2. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
HomeEvArmStatus	BOOL	Set when a home event has been armed through execution of the MAH (Motion Axis Home) instruction. Cleared when a home event occurs.
HomeEvStatus	BOOL	Set when a home event has occurred. Cleared when another MAH (Motion Axis Home) instruction is executed.
OutputCamStatus	DINT	A set of bits* that are set when the Output Cam has been initiated.
OutputCamPendingStatus	DINT	A set of bits* that are set when an Output Cam is waiting for an armed Output Cam to move beyond its cam start/cam end position.
OutputCamLockStatus	DINT	A set of bits* that are set when an Output Cam is locked to the Master Axis.
OutputCamTransitionStatus	DINT	A set of bits* that are set when the transition from the current armed Output Cam to the pending Output Cam is in process.
		* The bit number corresponds with the execution target number. One bit per execution target.

AXIS_SERVO Structure

A servo object represents an axis with full motion planner functionality and integrated configuration support. It is associated with modules that close a servo loop and send an analog command to an external drive, such as a 1756-M02AE module.

The AXIS_SERVO structure contains and following status attributes.

Mnemonic:	Data Type:	Description:
MotionStatus	DINT	The motion status bits for your axis.

Bit:	Number:	Data Type:	Description:
AccelStatus	00	BOOL	Acceleration Status
DecelStatus	01	BOOL	Deceleration Status
MoveStatus	02	BOOL	Move Status
JogStatus	03	BOOL	Jog Status
GearingStatus	04	BOOL	Gearing Status
HomingStatus	05	BOOL	Homing Status
StoppingStatus	06	BOOL	Stopping Status
AxisHomedStatus	07	BOOL	Homed Status
PositionCamStatus	08	BOOL	Position Cam Status
TimeCamStatus	09	BOOL	Time Cam Status
PositionCamPendingStatus	10	BOOL	Position Cam Pending Status
TimeCamPendingStatus	11	BOOL	Time Cam Pending Status
GearingLockStatus	12	BOOL	Gearing Lock Status
PositionCamLockStatus	13	BOOL	Position Cam Lock Status
TimeCamLockStatus	14	BOOL	Time Cam Lock Status
MasterOffsetMoveStatus	15	BOOL	Master Offset Move Status

Mnemonic:	Data Type:	Description:			
AxisStatus	DINT	The status bits for your axis			
		Bit:	Number:	Data Type:	Description:
		ServoActStatus	00	BOOL	Servo Action Status
		DriveEnableStatus	01	BOOL	Drive Enable Status
		ShutdownStatus	02	BOOL	Axis Shutdown Status
		ConfigUpdateInProcess	03	BOOL	Configuration Update in Process
AxisFault	DINT	The axis faults for your axis:			
		Bit:	Number:	Data Type:	Description:
		PhysicalAxisFault	00	BOOL	Physical Axis Fault
		ModuleFault	01	BOOL	Module Fault
		ConfigFault	02	BOOL	Configuration Fault
AxisEvent	DINT	The event status for your axis:			
		Bit:	Number:	Data Type:	Description:
		WatchEvArmStatus	00	BOOL	Watch Event Armed Status
		WatchEvStatus	01	BOOL	Watch Event Status
		RegEvArmStatus	02	BOOL	Registration Event 1 Armed Status
		RegEvStatus	03	BOOL	Registration Event 1 Status
		RegEv2ArmStatus	04	BOOL	Registration Event 2 Armed Status
		RegEv2Status	05	BOOL	Registration Event 2 Status
		HomeEvArmStatus	06	BOOL	Home Event Armed Status
		HomeEvStatus	07	BOOL	Home Event Status
ActualPosition	REAL	Actual Position in Position Units			
StrobeActualPosition	REAL	Strobe Actual Position in Position Units			
StartActualPosition	REAL	Start Actual Position in Position Units			
AverageVelocity	REAL	Average Velocity in Position Units / Sec			
ActualVelocity	REAL	Actual Velocity in Position Units / Sec			
ActualAcceleration	REAL	Actual Acceleration in Position Units / Sec2			
WatchPosition	REAL	Watch Position in Position Units			
RegistrationPosition	REAL	Registration 1 Position in Position Units			
Registration2Position	REAL	Registration 2 Position in Position Units			
Registration1Time	DINT	Registration 1 Time as CST time in microseconds			
Registration2Time	DINT	Registration 2 Time as CST time in microseconds			
InterpolationTime	DINT	Interpolation Time as CST time in microseconds			
InterpolatedActualPosition	REAL	Interpolated Actual Position in Position Units			
MasterOffset	REAL	Master Offset in Master Position Units			
StrobeMasterOffset	REAL	Strobe Master Offset in Master Position Units			
StartMasterOffset	REAL	Start Master Offset in Master Position Units			
CommandPosition	REAL	Command Position in Position Units			
StrobeCommandPosition	REAL	Strobe Command Position in Position Units			
StartCommandPosition	REAL	Start Command Position in Position Units			
CommandVelocity	REAL	Command Velocity in Position Units / Sec			

Mnemonic:	Data Type:	Description:
CommandAcceleration	REAL	Command Acceleration in Position Units / Sec2
InterpolatedCommandPosition	REAL	Interpolated Command Position in Position Units
ServoStatus	DINT	The status bits for your servo loop: Bit: Number: Data Type: Description: - no tag - 00 BOOL Servo Action Status - no tag - 01 BOOL Drive Enable Status - no tag - 02 BOOL Axis Shutdown Status ProcessStatus 03 BOOL Process Status OutLmtStatus 04 BOOL Output Limit Status PosLockStatus 05 BOOL Position Lock Status HomeSwitchStatus 06 BOOL Home Input Status Reg1InputStatus 07 BOOL Registration 1 Input Status Reg2InputStatus 08 BOOL Registration 12Input Status PosOvertravelInputStatus 09 BOOL Positive Overtravel Input Status NegOvertravelInputStatus 10 BOOL Negative Overtravel Input Status DriveFaultInputStatus 11 BOOL Drive Fault Input Status
ServoFault	DINT	The servo fault bits for your servo loop: Bit: Number: Data Type: Description: POtrvlFault 00 BOOL Positive Software Overtravel Fault NOtrvlFault 01 BOOL Negative Software Overtravel Fault PosHardOvertravelFault 02 BOOL Positive Hardware Overtravel Fault NegHardOvertravelFault 03 BOOL Negative Hardware Overtravel Fault FeedbackFault 04 BOOL Feedback Fault FeedbackNoiseFault 05 BOOL Feedback Noise Fault AuxFeedbackFault 06 BOOL Auxiliary Feedback Fault AuxFeedbackNoiseFault 07 BOOL Auxiliary Feedback Noise Fault PosErrorFault 08 BOOL Position Error Fault DriveFault 09 BOOL Drive Fault
ServoModuleFault	DINT	Bit: Number: Data Type: Description: ControlSyncFault 00 BOOL Control Sync Fault ModuleSyncFault 01 BOOL Module Sync Fault TimerEventFault 02 BOOL Timer Event Fault ModuleHardwareFault 03 BOOL Module Hardware Fault
AttributeErrorCode	INT	ASA Error code returned by erred set attribute list service to the module.
AttributeErrorID	INT	Attribute ID associated with non-zero Attribute Error Code.
PositionCommand	REAL	Position Command in Position Units
PositionFeedback	REAL	Position Feedback in Position Units
AuxPositionFeedback	REAL	Auxiliary Position Feedback in Position Units
PositionError	REAL	Position Error in Position Units
PositionIntegratorError	REAL	Position Integrator Error in Position Units - mSec
VelocityCommand	REAL	Velocity Command in Position Units / Sec
VelocityFeedback	REAL	Velocity Feedback in Position Units / Sec
VelocityError	REAL	Velocity Error in Position Units / Sec
VelocityIntegratorError	REAL	Velocity Integrator Error in Position Units – mSec / Sec
AccelerationCommand	REAL	Acceleration Command in Position Units / Sec2

Mnemonic:	Data Type:	Description:
AccelerationFeedback	REAL	Acceleration Feedback in Position Units / Sec2
ServoOutputLevel	REAL	Servo Output Level in Volts
MarkerDistance	REAL	Marker Distance in Position Units
VelocityOffset	REAL	Velocity Offset in Position Units / Sec
TorqueOffset	REAL	Torque Offset from –100% to +100%
AccelStatus	BOOL	Set if the axis is currently being commanded to accelerate.
DecelStatus	BOOL	Set if the axis is currently being commanded to decelerate.
MoveStatus	BOOL	Set if a Move motion profile is currently in progress. Cleared when the Move is complete or is superseded by some other motion operation.
JogStatus	BOOL	Set if a Jog motion profile is currently in progress. Cleared when the Jog is complete or is superseded by some other motion operation.
GearingStatus	BOOL	Set if the axis is a slave that is currently Gearing to another axis. Cleared when the gearing operation is stopped or is superseded by some other motion operation.
HomingStatus	BOOL	Set if a Home motion profile is currently in progress. Cleared when the homing operation is stopped or is superseded by some other motion operation.
StoppingStatus	BOOL	Set if there is a stopping process currently in progress. Cleared when the stopping process is complete. Note: The stopping process is used to stop an axis (initiated by an MAS, MGS, Stop Motion fault action, or mode change).
AxisHomedStatus	BOOL	Cleared at power-up or reconnection. Set by the MAH instruction upon successful completion of the configured homing sequence, and later cleared when the axis enters the shutdown state.
PositionCamStatus	BOOL	Set if a Position Cam motion profile is currently in progress. Cleared when the Position Cam is complete or is superseded by some other motion operation.
TimeCamStatus	BOOL	Set if a Time Cam motion profile is currently in progress. Cleared when the Time Cam is complete or is superseded by some other motion operation.
PositionCamPendingStatus	BOOL	Set if a Position Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MAPC instruction with Pending execution selected. This bit is cleared when the current position cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the position cam profile completes, or is superseded by some other motion operation.
TimeCamPendingStatus	BOOL	Set if a Time Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MATC instruction with Pending execution selected. This bit is cleared when the current time cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the time cam profile completes, or is superseded by some other motion operation.
GearingLockStatus	BOOL	Set whenever the slave axis is locked to the master axis in a gearing relationship according to the specified gear ratio. The clutch function of the gearing planner is used to ramp an axis up, or down, to speed in a gearing process (MAG with Clutch selected). This bit is cleared during the intervals where the axis is clutching.
PositionCamLockStatus	BOOL	Set whenever the master axis satisfies the starting condition of a currently active Position Cam motion profile. The starting condition is established by the Start Control and Start Position parameters of the MAPC instruction. This bit is bit is cleared when the current position cam profile completes, or is superseded by some other motion operation. In uni-directional master direction mode, the Position Cam Lock Status bit is cleared when moving in the “wrong” direction and sets when moving in the “correct” direction.

Mnemonic:	Data Type:	Description:
MasterOffsetMoveStatus	BOOL	Set if a Master Offset Move motion profile is currently in progress. This bit is cleared when the Master Offset Move is complete or is superseded by some other motion operation.
ServoActStatus	BOOL	Set when the associated axis is under servo control. Cleared when servo action is disabled.
DriveEnableStatus	BOOL	Set when the Drive Enable output of the associated physical axis is currently enabled. Cleared when physical servo axis Drive Enable output is currently disabled.
ShutdownStatus	BOOL	Set when the associated axis is currently in the Shutdown state. Cleared when the axis is transitioned from the Shutdown state to another state.
ConfigUpdateInProcess	BOOL	The Configuration Update Status Bits attribute provides a method for monitoring the progress of one or more specific module configuration attribute updates initiated by either a Set Attribute List service (which is internal to the firmware) or an SSV in the user program. When such an update is initiated, the ControlLogix processor sets this bit. This bit will remain set until the Set Attribute List reply comes back from the servo module indicating that the data update process was successful. Thus the Configuration Update Status Bits attribute provides a method of waiting until the servo configuration data update to the connected motion module is complete before starting a dependent operation.
PhysicalAxisFault	BOOL	Set when one or more fault conditions have been reported by the physical axis. The specific fault conditions can then be determined through access to the fault attributes of the associated physical axis. A PhysicalAxisFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ModuleFault	BOOL	Set when a serious fault has occurred with the motion module associated with the selected axis. Usually a module fault affects all axes associated with the motion module. A module fault generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition. A ModuleFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ConfigFault	BOOL	Set when an update operation targeting an axis configuration attribute of an associated motion module has failed. Specific information concerning the Configuration Fault may be found in the Attribute Error Code and Attribute Error ID attributes associated with the motion module. A ConfigFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
WatchEvArmStatus	BOOL	Set when a watch event has been armed through execution of the MAW (Motion Arm Watch) instruction. Cleared when either a watch event occurs or a MDW (Motion Disarm Watch) instruction is executed.
WatchEvStatus	BOOL	Set when a watch event has occurred. Cleared when either another MAW (Motion Arm Watch) instruction or a MDW (Motion Disarm Watch) instruction is executed.
RegEvArmStatus	BOOL	Set when a registration checking has been armed for registration input 1 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEvStatus	BOOL	Set when a registration event has occurred on registration input 1. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEv2ArmStatus	BOOL	Set when a registration checking has been armed for registration input 2 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
RegEv2Status	BOOL	Set when a registration event has occurred on registration input 2. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.

Mnemonic:	Data Type:	Description:
HomeEvArmStatus	BOOL	Set when a home event has been armed through execution of the MAH (Motion Axis Home) instruction. Cleared when a home event occurs.
HomeEvStatus	BOOL	Set when a home event has occurred. Cleared when another MAH (Motion Axis Home) instruction is executed.
ProcessStatus	BOOL	Set when there is an axis tuning operation or an axis hookup diagnostic test operation in progress on the associated physical axis.
OutLmtStatus	BOOL	Set when the magnitude of the output of the associated physical servo axis has reached or exceeded the configured Output Limit value.
PosLockStatus	BOOL	Set when the magnitude of the axis position error has become less than or equal to the configured Position Lock Tolerance value for the associated physical axis.
HomeSwitchStatus	BOOL	Set when the current state of the dedicated Home input is active. Cleared when the Home input is inactive.
DriveFaultInputStatus	BOOL	Set when the current state of the Drive Fault input is active. Cleared when the Drive Fault input is inactive.
Reg1InputStatus	BOOL	Set when the current state of the dedicated Registration 1 input is active. Clear when the Registration 1 input is inactive.
Reg2InputStatus	BOOL	Set when the current state of the dedicated Registration 1 input is active. Clear when the Registration 1 input is inactive.
PosOvertravelInputStatus	BOOL	Set when the current state of the dedicated Positive Overtravel input is active. Clear when the Positive Overtravel input is inactive.
NegOvertravelInputStatus	BOOL	Set when the current state of the dedicated Negative Overtravel input is active. Clear when the Negative Overtravel input is inactive.
POtravlFault	BOOL	Set when the axis has traveled, or attempted to travel, beyond the current configured value for Maximum Positive Travel. Cleared when the axis is moved back within this travel limit.
NOtravlFault	BOOL	Set when the axis has traveled, or attempted to travel, beyond the current configured value for Maximum Negative Travel. Cleared when the axis is moved back within this travel limit.
PosHardOvertravelFault	BOOL	Set when the axis has traveled beyond the current positive direction position limits as established by hardware limit switches mounted on the machine. To recover, the axis must be moved back with normal operation limits of the machine and the limit switch reset. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
NegHardOvertravelFault	BOOL	Set when the axis has traveled beyond the current negative direction position limits as established by hardware limit switches mounted on the machine. To recover, the axis must be moved back with normal operation limits of the machine and the limit switch reset. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
FeedbackFault	BOOL	Set for a specific feedback source when one of the following conditions occurs: • The differential electrical signals for one or more of the feedback channels (e.g., A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive; • Loss of feedback "power" or feedback "common" electrical connection between the servo module or drive and the feedback device. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.

Mnemonic:	Data Type:	Description:
FeedbackNoiseFault	BOOL	Set for a specific feedback source when the servo module has detected simultaneous transitions of the feedback A and B channels (called "feedback noise"). Feedback noise is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring, both of which may be able to be seen on an oscilloscope. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
AuxFeedbackFault	BOOL	Set for an auxiliary feedback source when one of the following conditions occurs: • The differential electrical signals for one or more of the feedback channels (e.g., A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive; • Loss of feedback "power" or feedback "common" electrical connection between the servo module or drive and the feedback device.This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
AuxFeedbackNoiseFault	BOOL	Set for an auxiliary feedback source when the servo module has detected simultaneous transitions of the feedback A and B channels (called "feedback noise"). Feedback noise is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring, both of which may be able to be seen on an oscilloscope. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
PosErrorFault	BOOL	Set when the servo has detected that the axis position error has exceeded the current configured value for Position ErrorTolerance. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
DriveFault	BOOL	Set when the external servo drive has detected a fault and has communicated the existence of this fault to the servo module via the Drive Fault input. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
ControlSyncFault	BOOL	Set when the Logix controller detects that several position update messages in a row from the motion module have been missed due to a failure of the synchronous communications connection. This condition results in the automatic shutdown of the associated servo module. The Logix controller is designed to "ride-through" a maximum of four missed position updates without issuing a fault or adversely affecting motion in progress. Missing more than four position updates in a row constitutes a problematic condition that warrants shutdown of the servo module. This fault bit is cleared when the connection is reestablished.
ModuleSyncFault	BOOL	Set when the motion module detects that several position update messages in a row from the ControlLogix processor module have been missed due to a failure of the synchronous communications connection. This condition results in the automatic shutdown of the servo module. The servo module is designed to "ride-through" a maximum of four missed position updates without issuing a fault or adversely affecting motion in progress. Missing more than four position updates in a row constitutes a problematic condition that warrants shutdown of the servo module. This fault bit is cleared when the connection is reestablished.
TimerEventFault	BOOL	Set when the associated servo module has detected a problem with the module's timer event functionality used to synchronize the motion module's servo loop to the master timebase of the Logix rack (i.e. Coordinated System Time). This fault bit can be cleared only by reconfiguration of the motion module.
ModuleHardwareFault	BOOL	Set when the associated servo module has detected a hardware problem that, in general, is going to require replacement of the module to correct.

Mnemonic:	Data Type:	Description:
OutputCamStatus	DINT	A set of bits* that are set when the Output Cam has been initiated.
OutputCamPendingStatus	DINT	A set of bits* that are set when an Output Cam is waiting for an armed Output Cam to move beyond its cam start/cam end position.
OutputCamLockStatus	DINT	A set of bits* that are set when an Output Cam is locked to the Master Axis.
OutputCamTransitionStatus	DINT	A set of bits* that are set when the transition from the current armed Output Cam to the pending Output Cam is in process.
		* The bit number corresponds with the execution target number. One bit per execution target.

AXIS_SERVO_DRIVE Structure

A servo-drive axis object represents an axis with full motion planner functionality and integrated configuration support. It is associated with digital drive interface modules sending digital commands to an external drive, such as a 1756-M08SE (SERCOS). The AXIS_SERVO_DRIVE structure contains the following status and configuration attributes.

Mnemonic:	Data Type:	Description:			
MotionStatus	DINT	The motion status bits for your axis.			
		Bit:	Number:	Data Type:	Description:
		AccelStatus	00	BOOL	Acceleration Status
		DecelStatus	01	BOOL	Deceleration Status
		MoveStatus	02	BOOL	Move Status
		JogStatus	03	BOOL	Jog Status
		GearingStatus	04	BOOL	Gearing Status
		HomingStatus	05	BOOL	Homing Status
		StoppingStatus	06	BOOL	Stopping Status
		AxisHomedStatus	07	BOOL	Homed Status
		PositionCamStatus	08	BOOL	Position Cam Status
		TimeCamStatus	09	BOOL	Time Cam Status
		PositionCamPendingStatus	10	BOOL	Position Cam Pending Status
		TimeCamPendingStatus	11	BOOL	Time Cam Pending Status
		GearingLockStatus	12	BOOL	Gearing Lock Status
		PositionCamLockStatus	13	BOOL	Position Cam Lock Status
		TimeCamLockStatus	14	BOOL	Time Cam Lock Status
		MasterOffsetMoveStatus	15	BOOL	Master Offset Move Status
AxisStatus	DINT	The status bits for your axis			
		Bit:	Number:	Data Type:	Description:
		ServoActStatus	00	BOOL	Servo Action Status
		DriveEnableStatus	01	BOOL	Drive Enable Status
		ShutdownStatus	02	BOOL	Axis Shutdown Status
		ConfigUpdateInProcess	03	BOOL	Configuration Update in Process

Mnemonic:	Data Type:	Description:			
AxisFault	DINT	The axis faults for your axis:			
		Bit:	Number:	Data Type:	Description:
		PhysicalAxisFault	00	BOOL	Physical Axis Fault
		ModuleFault	01	BOOL	Module Fault
		ConfigFault	02	BOOL	Configuration Fault
AxisEvent	DINT	The event status for your axis:			
		Bit:	Number:	Data Type:	Description:
		WatchEvArmStatus	00	BOOL	Watch Event Armed Status
		WatchEvStatus	01	BOOL	Watch Event Status
		RegEvArmStatus	02	BOOL	Registration Event 1 Armed Status
		RegEvStatus	03	BOOL	Registration Event 1 Status
		RegEv2ArmStatus	04	BOOL	Registration Event 2 Armed Status
		RegEv2Status	05	BOOL	Registration Event 2 Status
		HomeEvArmStatus	06	BOOL	Home Event Armed Status
		HomeEvStatus	07	BOOL	Home Event Status
ActualPosition	REAL	Actual Position in Position Units			
StrobeActualPosition	REAL	Strobe Actual Position in Position Units			
StartActualPosition	REAL	Start Actual Position in Position Units			
AverageVelocity	REAL	Average Velocity in Position Units / Sec			
ActualVelocity	REAL	Actual Velocity in Position Units / Sec			
ActualAcceleration	REAL	Actual Acceleration in Position Units / Sec2			
WatchPosition	REAL	Watch Position in Position Units			
RegistrationPosition	REAL	Registration 1 Position in Position Units			
Registration2Position	REAL	Registration 2 Position in Position Units			
Registration1Time	DINT	Registration 1 Time as CST time in microseconds			
Registration2Time	DINT	Registration 2 Time as CST time in microseconds			
InterpolationTime	DINT	Interpolation Time as CST time in microseconds			
InterpolatedActualPosition	REAL	Interpolated Actual Position in Position Units			
MasterOffset	REAL	Master Offset in Master Position Units			
StrobeMasterOffset	REAL	Strobe Master Offset in Master Position Units			
StartMasterOffset	REAL	Start Master Offset in Master Position Units			
CommandPosition	REAL	Command Position in Position Units			
StrobeCommandPosition	REAL	Strobe Command Position in Position Units			
StartCommandPosition	REAL	Start Command Position in Position Units			
CommandVelocity	REAL	Command Velocity in Position Units / Sec			
CommandAcceleration	REAL	Command Acceleration in Position Units / Sec2			
InterpolatedCommandPosition	REAL	Interpolated Command Position in Position Units			

Mnemonic:	Data Type:	Description:			
ServoModuleFault	DINT	Bit:	Number:	Data Type:	Description:
		ControlSyncFault	00	BOOL	Control Sync Fault
		ModuleSyncFault	01	BOOL	Module Sync Fault
		TimerEventFault	02	BOOL	Timer Event Fault
		ModuleHardwareFault	03	BOOL	Module Hardware Fault
AttributeErrorCode	INT	ASA Error code returned by erred set attribute list service to the module.			
AttributeErrorID	INT	Attribute ID associated with non-zero Attribute Error Code.			
PositionCommand	REAL	Position Command in Position Units			
PositionFeedback	REAL	Position Feedback in Position Units			
AuxPositionFeedback	REAL	Auxiliary Position Feedback in Position Units			
PositionError	REAL	Position Error in Position Units			
PositionIntegratorError	REAL	Position Integrator Error in Position Units - mSec			
VelocityCommand	REAL	Velocity Command in Position Units / Sec			
VelocityFeedback	REAL	Velocity Feedback in Position Units / Sec			
VelocityError	REAL	Velocity Error in Position Units / Sec			
VelocityIntegratorError	REAL	Velocity Integrator Error in Position Units – mSec / Sec			
AccelerationCommand	REAL	Acceleration Command in Position Units / Sec2			
AccelerationFeedback	REAL	Acceleration Feedback in Position Units / Sec2			
ServoOutputLevel	REAL	Servo Output Level in Volts			
MarkerDistance	REAL	Marker Distance in Position Units			
VelocityOffset	REAL	Velocity Offset in Position Units / Sec			
TorqueOffset	REAL	Torque Offset from –100% to +100%			
TorqueCommand	REAL	The command when operating in Torque Mode in terms of % rated.			
TorqueFeedback	REAL	The torque feedback when operating in Torque Mode in terms of % rated.			
PosDynamicTorqueLimit	REAL	The currently operative maximum positive torque/current limit magnitude. It should be the lowest value of all torque/current limits in the drive at a given time, including: amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and motor thermal limit.			
NegDynamicTorqueLimit	REAL	The currently operative negative positive torque/current limit magnitude. It should be the lowest value of all torque/current limits in the drive at a given time, including: amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and motor thermal limit.			
MotorCapacity	REAL	The present utilization of motor capacity as a percent of rated capacity.			
DriveCapacity	REAL	The present utilization of drive capacity as a percent of rated capacity.			
PowerCapacity	REAL	The present utilization of the axis power supply as a percent of rated capacity.			
BusRegulatorCapacity	REAL	The present utilization of the axis bus regulator as a percent of rated capacity.			
MotorElectricalAngle	REAL	The present electrical angle of the motor shaft.			
TorqueLimitSource	DINT	The present source (if any) of any torque limiting for the axis.			

Mnemonic:	Data Type:	Description:			
DriveStatus	DINT	The status bits for your servo drive:			
		Bit:	Number:	Data Type:	Description:
		- no tag -	00	BOOL	Servo Action Status
		- no tag -	01	BOOL	Drive Enable Status
		- no tag -	02	BOOL	Axis Shutdown Status
		ProcessStatus	03	BOOL	Process Status
		- no tag -	04	BOOL	(Reserved)
		- no tag -	05	BOOL	(Reserved)
		HomeSwitchStatus	06	BOOL	Home Input Status
		Reg1InputStatus	07	BOOL	Registration 1 Input Status
		Reg2InputStatus	08	BOOL	Registration 12Input Status
		PosOvertravelInputStatus	09	BOOL	Positive Overtravel Input Status
		NegOvertravelInputStatus	10	BOOL	Negative Overtravel Input Status
		EnableInputStatus	11	BOOL	Enable Input Status
		AccelLimitStatus	12	BOOL	Accel Limit Status
		- no tag -	13-15	BOOL	(Reserved)
		VelocityLockStatus	16	BOOL	Velocity Lock Status
		VelocityStandstillStatus	17	BOOL	Velocity Standstill Status
		VelocityThresholdStatus	18	BOOL	Velocity Threshold Status
		TorqueThresholdStatus	19	BOOL	Torque Threshold Status
		TorqueLimitStatus	20	BOOL	Torque Limit Status
		VelocityLimitStatus	21	BOOL	Velocity Limit Status
		PosLockStatus	22	BOOL	Position Lock Status
		-no tag -	23	BOOL	(Reserved)
		-no tag -	24	BOOL	(Reserved)
		-no tag -	25	BOOL	(Reserved)
		-no tag -	26	BOOL	(Reserved)
DriveFault	DINT	The servo fault bits for your servo loop:			
		Bit:	Number:	Data Type:	Description:
		PosSoftOtravelFault	00	BOOL	Positive Software Overtravel Fault
		NegSoftOtravelFault	01	BOOL	Negative Software Overtravel Fault
		PosHardOvertravelFault	02	BOOL	Positive Hardware Overtravel Fault
		NegHardOvertravelFault	03	BOOL	Negative Hardware Overtravel Fault
		MotFeedbackFault	04	BOOL	Feedback Fault
		MotFeedbackNoiseFault	05	BOOL	Feedback Noise Fault
		AuxFeedbackFault	06	BOOL	Auxiliary Feedback Fault
		AuxFeedbackNoiseFault	07	BOOL	Auxiliary Feedback Noise Fault
		- no tag -	08-12	BOOL	(Reserved)
		GroundShortFault	13	BOOL	Ground Short Fault
		DriveHardFault	14	BOOL	Drive Hard Fault
		OverSpeedFault	15	BOOL	Overspeed Fault
		MotorOverloadFault	16	BOOL	Overload Fault
		DriveOvertempFault	17	BOOL	Drive Overtemp Fault
		MotorOvertempFault	18	BOOL	Motor Overtemp Fault
		DriveCoolingFault	19	BOOL	Drive Cooling Fault
		DriveControlVoltageFault	20	BOOL	Drive Control Voltage Fault
		FeedbackFault	21	BOOL	Feedback Fault
		CommutationFault	22	BOOL	Commutation Fault
		DriveOvercurrentFault	23	BOOL	Drive Overcurrent Fault
		DriveOvervoltageFault	24	BOOL	Drive Overvoltage Fault
		DriveUndervoltageFault	25	BOOL	Drive Undervoltage Fault
		PowerPhaseLossFault	26	BOOL	Power Phase Loss Fault
		PosErrorFault	27	BOOL	Position Error Fault
		- no tag -	28	BOOL	SERCOS Commutation Fault

Mnemonic:	Data Type:	Description:
SERCOSErrorCode	INT	Error code returned by SERCOS module indicating source of drive parameter update failure.
AccelStatus	BOOL	Set if the axis is currently being commanded to accelerate.
DecelStatus	BOOL	Set if the axis is currently being commanded to decelerate.
MoveStatus	BOOL	Set if a Move motion profile is currently in progress. Cleared when the Move is complete or is superseded by some other motion operation.
JogStatus	BOOL	Set if a Jog motion profile is currently in progress. Cleared when the Jog is complete or is superseded by some other motion operation.
GearingStatus	BOOL	Set if the axis is a slave that is currently Gearing to another axis. Cleared when the gearing operation is stopped or is superseded by some other motion operation.
HomingStatus	BOOL	Set if a Home motion profile is currently in progress. Cleared when the homing operation is stopped or is superseded by some other motion operation.
StoppingStatus	BOOL	Set if there is a stopping process currently in progress. Cleared when the stopping process is complete. Note: The stopping process is used to stop an axis (initiated by an MAS, MGS, Stop Motion fault action, or mode change).
AxisHomedStatus	BOOL	Cleared at power-up or reconnection. Set by the MAH instruction upon successful completion of the configured homing sequence, and later cleared when the axis enters the shutdown state.
PositionCamStatus	BOOL	Set if a Position Cam motion profile is currently in progress. Cleared when the Position Cam is complete or is superseded by some other motion operation.
TimeCamStatus	BOOL	Set if a Time Cam motion profile is currently in progress. Cleared when the Time Cam is complete or is superseded by some other motion operation.
PositionCamPendingStatus	BOOL	Set if a Position Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MAPC instruction with Pending execution selected. This bit is cleared when the current position cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the position cam profile completes, or is superseded by some other motion operation.
TimeCamPendingStatus	BOOL	Set if a Time Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MATC instruction with Pending execution selected. This bit is cleared when the current time cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the time cam profile completes, or is superseded by some other motion operation.
GearingLockStatus	BOOL	Set whenever the slave axis is locked to the master axis in a gearing relationship according to the specified gear ratio. The clutch function of the gearing planner is used to ramp an axis up, or down, to speed in a gearing process (MAG with Clutch selected). This bit is cleared during the intervals where the axis is clutching.
PositionCamLockStatus	BOOL	Set whenever the master axis satisfies the starting condition of a currently active Position Cam motion profile. The starting condition is established by the Start Control and Start Position parameters of the MAPC instruction. This bit is bit is cleared when the current position cam profile completes, or is superseded by some other motion operation. In uni-directional master direction mode, the Position Cam Lock Status bit is cleared when moving in the “wrong” direction and sets when moving in the “correct” direction.
MasterOffsetMoveStatus	BOOL	Set if a Master Offset Move motion profile is currently in progress. This bit is cleared when the Master Offset Move is complete or is superseded by some other motion operation.
ServoActStatus	BOOL	Set when the associated axis is under servo control. Cleared when servo action is disabled.

Mnemonic:	Data Type:	Description:
DriveEnableStatus	BOOL	Set when the Drive Enable output of the associated physical axis is currently enabled. Cleared when physical servo axis Drive Enable output is currently disabled.
ShutdownStatus	BOOL	Set when the associated axis is currently in the Shutdown state. Cleared when the axis is transitioned from the Shutdown state to another state.
ConfigUpdateInProcess	BOOL	The Configuration Update Status Bits attribute provides a method for monitoring the progress of one or more specific module configuration attribute updates initiated by either a Set Attribute List service (which is internal to the firmware) or an SSV in the user program. When such an update is initiated, the ControlLogix processor sets this bit. This bit will remain set until the Set Attribute List reply comes back from the servo module indicating that the data update process was successful. Thus the Configuration Update Status Bits attribute provides a method of waiting until the servo configuration data update to the connected motion module is complete before starting a dependent operation.
PhysicalAxisFault	BOOL	Set when one or more fault conditions have been reported by the physical axis. The specific fault conditions can then be determined through access to the fault attributes of the associated physical axis. A PhysicalAxisFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ModuleFault	BOOL	Set when a serious fault has occurred with the motion module associated with the selected axis. Usually a module fault affects all axes associated with the motion module. A module fault generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition. A ModuleFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ConfigFault	BOOL	Set when an update operation targeting an axis configuration attribute of an associated motion module has failed. Specific information concerning the Configuration Fault may be found in the Attribute Error Code and Attribute Error ID attributes associated with the motion module. A ConfigFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
WatchEvArmStatus	BOOL	Set when a watch event has been armed through execution of the MAW (Motion Arm Watch) instruction. Cleared when either a watch event occurs or a MDW (Motion Disarm Watch) instruction is executed.
WatchEvStatus	BOOL	Set when a watch event has occurred. Cleared when either another MAW (Motion Arm Watch) instruction or a MDW (Motion Disarm Watch) instruction is executed.
RegEvArmStatus	BOOL	Set when a registration checking has been armed for registration input 1 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEvStatus	BOOL	Set when a registration event has occurred on registration input 1. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEv2ArmStatus	BOOL	Set when a registration checking has been armed for registration input 2 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
RegEv2Status	BOOL	Set when a registration event has occurred on registration input 2. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
HomeEvArmStatus	BOOL	Set when a home event has been armed through execution of the MAH (Motion Axis Home) instruction. Cleared when a home event occurs.
HomeEvStatus	BOOL	Set when a home event has occurred. Cleared when another MAH (Motion Axis Home) instruction is executed.

Mnemonic:	Data Type:	Description:
ControlSyncFault	BOOL	Set when the Logix controller detects that several position update messages in a row from the motion module have been missed due to a failure of the synchronous communications connection. This condition results in the automatic shutdown of the associated servo module. The Logix controller is designed to "ride-through" a maximum of four missed position updates without issuing a fault or adversely affecting motion in progress. Missing more than four position updates in a row constitutes a problematic condition that warrants shutdown of the servo module. This fault bit is cleared when the connection is reestablished.
ModuleSyncFault	BOOL	Set when the motion module detects that several position update messages in a row from the ControlLogix processor module have been missed due to a failure of the synchronous communications connection. This condition results in the automatic shutdown of the servo module. The servo module is designed to "ride-through" a maximum of four missed position updates without issuing a fault or adversely affecting motion in progress. Missing more than four position updates in a row constitutes a problematic condition that warrants shutdown of the servo module. This fault bit is cleared when the connection is reestablished.
TimerEventFault	BOOL	Set when the associated servo module has detected a problem with the module's timer event functionality used to synchronize the motion module's servo loop to the master timebase of the Logix rack (i.e. Coordinated System Time). This fault bit can be cleared only by reconfiguration of the motion module.
ModuleHardwareFault	BOOL	Set when the associated servo module has detected a hardware problem that, in general, is going to require replacement of the module to correct.
ProcessStatus	BOOL	Set when there is an axis tuning operation or an axis hookup diagnostic test operation in progress on the associated physical axis.
HomeInputStatus	BOOL	Set when the current state of the dedicated Home input is active. Cleared when the Home input is inactive.
Reg1InputStatus	BOOL	Set when the current state of the dedicated Registration 1 input is active. Clear when the Registration 1 input is inactive.
Reg2InputStatus	BOOL	Set when the current state of the dedicated Registration 1 input is active. Clear when the Registration 1 input is inactive.
PosOvertravelInputStatus	BOOL	Set when the current state of the dedicated Positive Overtravel input is active. Clear when the Positive Overtravel input is inactive.
NegOvertravelInputStatus	BOOL	Set when the current state of the dedicated Negative Overtravel input is active. Clear when the Negative Overtravel input is inactive.
EnableInputStatus	BOOL	Set when the current state of the dedicated Enable Input is active. Clear when the Enable Input is inactive.
AccelLimitStatus	BOOL	Set when the magnitude of the commanded acceleration to the velocity servo loop input is greater than the configured Velocity Limit.
VelocityLockStatus	BOOL	Set when the magnitude of the physical axis Velocity Feedback is within the configured Velocity Window of the current velocity command.
VelocityStandstillStatus	BOOL	Set when the magnitude of the physical axis Velocity Feedback is within the configured Velocity Standstill Window of zero speed.
VelocityThresholdStatus	BOOL	Set when the magnitude of the physical axis Velocity Feedback is less than the configured Velocity Threshold.
TorqueThresholdStatus	BOOL	Set when the magnitude of the physical axis Torque Feedback is less than the configured Torque Threshold.
TorqueLimitStatus	BOOL	Set when the magnitude of the axis torque command is greater than the configured Torque Limit.

Mnemonic:	Data Type:	Description:
VelocityLimitStatus	BOOL	Set when the magnitude of the commanded velocity to the velocity servo loop input is greater than the configured Velocity Limit.
PosLockStatus	BOOL	Set when the magnitude of the axis position error has become less than or equal to the configured Position Lock Tolerance value for the associated physical axis.
PosSoftOvertravelFault	BOOL	Set when the axis has traveled, or attempted to travel, beyond the current configured value for Maximum Positive Travel. Cleared when the axis is moved back within this travel limit.
NegSoftOvertravelFault	BOOL	Set when the axis has traveled, or attempted to travel, beyond the current configured value for Maximum Negative Travel. Cleared when the axis is moved back within this travel limit.
PosHardOvertravelFault	BOOL	Set when the axis has traveled beyond the current positive direction position limits as established by hardware limit switches mounted on the machine. To recover, the axis must be moved back with normal operation limits of the machine and the limit switch reset. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
NegHardOvertravelFault	BOOL	Set when the axis has traveled beyond the current negative direction position limits as established by hardware limit switches mounted on the machine. To recover, the axis must be moved back with normal operation limits of the machine and the limit switch reset. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
MotFeedbackFault	BOOL	Set for a specific feedback source when one of the following conditions occurs: • The differential electrical signals for one or more of the feedback channels (e.g., A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive. • Loss of feedback "power" or feedback "common" electrical connection between the servo module or drive and the feedback device.This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
MotFeedbackNoiseFault	BOOL	Set for a specific feedback source when the servo module has detected simultaneous transitions of the feedback A and B channels (called "feedback noise"). Feedback noise is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring, both of which may be able to be seen on an oscilloscope. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
AuxFeedbackFault	BOOL	Set for an auxiliary feedback source when one of the following conditions occurs: • The differential electrical signals for one or more of the feedback channels (e.g., A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive; • Loss of feedback "power" or feedback "common" electrical connection between the servo module or drive and the feedback device.This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.

Mnemonic:	Data Type:	Description:
AuxFeedbackNoiseFault	BOOL	Set for an auxiliary feedback source when the servo module has detected simultaneous transitions of the feedback A and B channels (called "feedback noise"). Feedback noise is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring, both of which may be able to be seen on an oscilloscope. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
GroundShortFault		Set for an auxiliary feedback source when one of the following conditions occurs: • The differential electrical signals for one or more of the feedback channels (e.g., A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive. • Loss of feedback "power" or feedback "common" electrical connection between the servo module or drive and the feedback device.This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
DriveHardFault	BOOL	Set when the drive detects a serious hardware fault.
OverspeedFault	BOOL	Set when the speed of the axis as determined from the feedback has exceeded the overspeed limit which is typically set to 150% of configured velocity limit for the motor.
OverloadFault	BOOL	Set when the load limit of the motor/drive has been exceeded and persists. (This attribute is often tied into the IT limit of the drive.)
DriveOvertempFault	BOOL	Set when the drive's temperature exceeds the drive shutdown temperature.
MotorOvertempFault	BOOL	Set when the motor's temperature exceeds the motor shutdown temperature.
DriveCoolingFault	BOOL	Set when the ambient temperature surrounding the drive's control circuitry temperature exceeds the drive ambient shut-down temperature.
DriveControlVoltageFault	BOOL	Set when the power supply voltages associated with the drive circuitry fall outside of acceptable limits.
FeedbackFault	BOOL	Set when one of the feedback sources associated with the drive axis has a problem that prevents the drive from receiving accurate or reliable position information from the feedback device.
CommutationFault	BOOL	Set when the commutation feedback source associated with the drive axis has a problem that prevents the drive from receiving accurate or reliable motor shaft information to perform commutation.
DriveOvercurrentFault	BOOL	Set when drive output current exceeds the predefined operating limits for the drive.
DriveOvervoltageFault	BOOL	Set when drive DC bus voltage exceeds the predefined operating limits for the bus.
DriveUndervoltageFault	BOOL	Set when drive DC bus voltage is below the predefined operating limits for the bus.
PowerPhaseLossFault	BOOL	Set when the drive detects that one or more of the three power line phases is lost from the 3 phase power inputs.
PosErrorFault	BOOL	Set when the servo has detected that the axis position error has exceeded the current configured value for Position ErrorTolerance. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.
OutputCamStatus	DINT	A set of bits* that are set when the Output Cam has been initiated.
OutputCamPendingStatus	DINT	A set of bits* that are set when an Output Cam is waiting for an armed Output Cam to move beyond its cam start/cam end position.

Mnemonic:	Data Type:	Description:
OutputCamLockStatus	DINT	A set of bits* that are set when an Output Cam is locked to the Master Axis.
OutputCamTransitionStatus	DINT	A set of bits* that are set when the transition from the current armed Output Cam to the pending Output Cam is in process.
		* The bit number corresponds with the execution target number. One bit per execution target.

AXIS_VIRTUAL Structure

A virtual axis object is an axis with full motion planner operation, but is not associated with any physical device.

The AXIS_VIRTUAL structure contains the following status attributes:

Mnemonic:	Data Type:	Description:			
MotionStatus	DINT	The motion status bits for your axis.			
		Bit:	Number:	Data Type:	Description:
		AccelStatus	00	BOOL	Acceleration Status
		DecelStatus	01	BOOL	Deceleration Status
		MoveStatus	02	BOOL	Move Status
		JogStatus	03	BOOL	Jog Status
		GearingStatus	04	BOOL	Gearing Status
		HomingStatus	05	BOOL	Homing Status
		StoppingStatus	06	BOOL	Stopping Status
		AxisHomedStatus	07	BOOL	Homed Status
		PositionCamStatus	08	BOOL	Position Cam Status
		TimeCamStatus	09	BOOL	Time Cam Status
		PositionCamPendingStatus	10	BOOL	Position Cam Pending Status
		TimeCamPendingStatus	11	BOOL	Time Cam Pending Status
		GearingLockStatus	12	BOOL	Gearing Lock Status
		PositionCamLockStatus	13	BOOL	Position Cam Lock Status
		TimeCamLockStatus	14	BOOL	Time Cam Lock Status
		MasterOffsetMoveStatus	15	BOOL	Master Offset Move Status
AxisStatus	DINT	The status bits for your axis			
		Bit:	Number:	Data Type:	Description:
		ServoActStatus	00	BOOL	Servo Action Status
		DriveEnableStatus	01	BOOL	Drive Enable Status
		ShutdownStatus	02	BOOL	Axis Shutdown Status
		ConfigUpdateInProcess	03	BOOL	Configuration Update in Process
AxisFault	DINT	The axis faults for your axis:			
		Bit:	Number:	Data Type:	Description:
		PhysicalAxisFault	00	BOOL	Physical Axis Fault
		ModuleFault	01	BOOL	Module Fault
		ConfigFault	02	BOOL	Configuration Fault

Mnemonic:	Data Type:	Description:
AxisEvent	DINT	The event status for your axis: Bit: — Number: — Data Type: — Description: WatchEvArmStatus — 00 — BOOL — Watch Event Armed Status WatchEvStatus — 01 — BOOL — Watch Event Status RegEvArmStatus — 02 — BOOL — Registration Event 1 Armed Status RegEvStatus — 03 — BOOL — Registration Event 1 Status RegEv2ArmStatus — 04 — BOOL — Registration Event 2 Armed Status RegEv2Status — 05 — BOOL — Registration Event 2 Status HomeEvArmStatus — 06 — BOOL — Home Event Armed Status HomeEvStatus — 07 — BOOL — Home Event Status
ActualPosition	REAL	Actual Position in Position Units
StrobeActualPosition	REAL	Strobe Actual Position in Position Units
StartActualPosition	REAL	Start Actual Position in Position Units
AverageVelocity	REAL	Average Velocity in Position Units / Sec
ActualVelocity	REAL	Actual Velocity in Position Units / Sec
ActualAcceleration	REAL	Actual Acceleration in Position Units / Sec2
WatchPosition	REAL	Watch Position in Position Units
RegistrationPosition	REAL	Registration 1 Position in Position Units
Registration2Position	REAL	Registration 2 Position in Position Units
Registration1Time	DINT	Registration 1 Time as CST time in microseconds
Registration2Time	DINT	Registration 2 Time as CST time in microseconds
InterpolationTime	DINT	Interpolation Time as CST time in microseconds
InterpolatedActualPosition	REAL	Interpolated Actual Position in Position Units
MasterOffset	REAL	Master Offset in Master Position Units
StrobeMasterOffset	REAL	Strobe Master Offset in Master Position Units
StartMasterOffset	REAL	Start Master Offset in Master Position Units
CommandPosition	REAL	Command Position in Position Units
StrobeCommandPosition	REAL	Strobe Command Position in Position Units
StartCommandPosition	REAL	Start Command Position in Position Units
CommandVelocity	REAL	Command Velocity in Position Units / Sec
CommandAcceleration	REAL	Command Acceleration in Position Units / Sec2
InterpolatedCommandPosition	REAL	Interpolated Command Position in Position Units
AccelStatus	BOOL	Set if the axis is currently being commanded to accelerate.
DecelStatus	BOOL	Set if the axis is currently being commanded to decelerate.
MoveStatus	BOOL	Set if a Move motion profile is currently in progress. Cleared when the Move is complete or is superseded by some other motion operation.
JogStatus	BOOL	Set if a Jog motion profile is currently in progress. Cleared when the Jog is complete or is superseded by some other motion operation.

Mnemonic:	Data Type:	Description:
GearingStatus	BOOL	Set if the axis is a slave that is currently Gearing to another axis. Cleared when the gearing operation is stopped or is superseded by some other motion operation.
HomingStatus	BOOL	Set if a Home motion profile is currently in progress. Cleared when the homing operation is stopped or is superseded by some other motion operation.
StoppingStatus	BOOL	Set if there is a stopping process currently in progress. Cleared when the stopping process is complete. Note: The stopping process is used to stop an axis (initiated by an MAS, MGS, Stop Motion fault action, or mode change).
AxisHomedStatus	BOOL	Cleared at power-up or reconnection. Set by the MAH instruction upon successful completion of the configured homing sequence, and later cleared when the axis enters the shutdown state.
PositionCamStatus	BOOL	Set if a Position Cam motion profile is currently in progress. Cleared when the Position Cam is complete or is superseded by some other motion operation.
TimeCamStatus	BOOL	Set if a Time Cam motion profile is currently in progress. Cleared when the Time Cam is complete or is superseded by some other motion operation.
PositionCamPendingStatus	BOOL	Set if a Position Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MAPC instruction with Pending execution selected. This bit is cleared when the current position cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the position cam profile completes, or is superseded by some other motion operation.
TimeCamPendingStatus	BOOL	Set if a Time Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MATC instruction with Pending execution selected. This bit is cleared when the current time cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the time cam profile completes, or is superseded by some other motion operation.
GearingLockStatus	BOOL	Set whenever the slave axis is locked to the master axis in a gearing relationship according to the specified gear ratio. The clutch function of the gearing planner is used to ramp an axis up, or down, to speed in a gearing process (MAG with Clutch selected). This bit is cleared during the intervals where the axis is clutching.
PositionCamLockStatus	BOOL	Set whenever the master axis satisfies the starting condition of a currently active Position Cam motion profile. The starting condition is established by the Start Control and Start Position parameters of the MAPC instruction. This bit is bit is cleared when the current position cam profile completes, or is superseded by some other motion operation. In uni-directional master direction mode, the Position Cam Lock Status bit is cleared when moving in the “wrong” direction and sets when moving in the “correct” direction.
MasterOffsetMoveStatus	BOOL	Set if a Master Offset Move motion profile is currently in progress. This bit is cleared when the Master Offset Move is complete or is superseded by some other motion operation.
ServoActStatus	BOOL	Set when the associated axis is under servo control. Cleared when servo action is disabled.
DriveEnableStatus	BOOL	Set when the Drive Enable output of the associated physical axis is currently enabled. Cleared when physical servo axis Drive Enable output is currently disabled.
ShutdownStatus	BOOL	Set when the associated axis is currently in the Shutdown state. Cleared when the axis is transitioned from the Shutdown state to another state.

Mnemonic:	Data Type:	Description:
ConfigUpdateInProcess	BOOL	The Configuration Update Status Bits attribute provides a method for monitoring the progress of one or more specific module configuration attribute updates initiated by either a Set Attribute List service (which is internal to the firmware) or an SSV in the user program. When such an update is initiated, the ControlLogix processor sets this bit. This bit will remain set until the Set Attribute List reply comes back from the servo module indicating that the data update process was successful. Thus the Configuration Update Status Bits attribute provides a method of waiting until the servo configuration data update to the connected motion module is complete before starting a dependent operation.
PhysicalAxisFault	BOOL	Set when one or more fault conditions have been reported by the physical axis. The specific fault conditions can then be determined through access to the fault attributes of the associated physical axis. A PhysicalAxisFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ModuleFault	BOOL	Set when a serious fault has occurred with the motion module associated with the selected axis. Usually a module fault affects all axes associated with the motion module. A module fault generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition. A ModuleFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
ConfigFault	BOOL	Set when an update operation targeting an axis configuration attribute of an associated motion module has failed. Specific information concerning the Configuration Fault may be found in the Attribute Error Code and Attribute Error ID attributes associated with the motion module. A ConfigFault can be set as either a Major Fault or a Non Major Fault in the Attribute tab of the associated Motion Group properties dialog box.
WatchEvArmStatus	BOOL	Set when a watch event has been armed through execution of the MAW (Motion Arm Watch) instruction. Cleared when either a watch event occurs or a MDW (Motion Disarm Watch) instruction is executed.
WatchEvStatus	BOOL	Set when a watch event has occurred. Cleared when either another MAW (Motion Arm Watch) instruction or a MDW (Motion Disarm Watch) instruction is executed.
RegEvArmStatus	BOOL	Set when a registration checking has been armed for registration input 1 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEvStatus	BOOL	Set when a registration event has occurred on registration input 1. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
RegEv2ArmStatus	BOOL	Set when a registration checking has been armed for registration input 2 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
RegEv2Status	BOOL	Set when a registration event has occurred on registration input 2. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
HomeEvArmStatus	BOOL	Set when a home event has been armed through execution of the MAH (Motion Axis Home) instruction. Cleared when a home event occurs.
HomeEvStatus	BOOL	Set when a home event has occurred. Cleared when another MAH (Motion Axis Home) instruction is executed.
OutputCamStatus	DINT	A set of bits* that are set when the Output Cam has been initiated.
OutputCamPendingStatus	DINT	A set of bits* that are set when an Output Cam is waiting for an armed Output Cam to move beyond its cam start/cam end position.

Mnemonic:	Data Type:	Description:
OutputCamLockStatus	DINT	A set of bits* that are set when an Output Cam is locked to the Master Axis.
OutputCamTransitionStatus	DINT	A set of bits* that are set when the transition from the current armed Output Cam to the pending Output Cam is in process.
		* The bit number corresponds with the execution target number. One bit per execution target.

Servo Configuration Update Status Bits attributes

You can use the servo configuration update status bits attributes to monitor the progress of servo configuration attribute updates, which are initiated by an SSV instruction in your application program.

When the SSV instruction initiates an update, the controller sets the update status bit associated with the attribute. The update status bit remains set until the servo module indicates that the data update was successful.

For example, if you use an SSV instruction to change the PositionProportionalGain attribute of an axis and follow it with logic based on the completion of the SSV instruction, you can check for the resetting of the .PosPGainStatus bit to ensure that the servo module attribute is updated.

The following is a list of the servo configuration update status bits attributes.

Variable	Data Type	Description
.AccFfGainStatus	BOOL	The status of an update to the *AccelerationFeedforwardGain* attribute.
.AxisTypeStatus	BOOL	The status of an update to the *AxisType* attribute.
.DriveFaultActStatus	BOOL	The status of an update to the *DriveFaultAction* attribute.
.EncLossFaultActStatus	BOOL	The status of an update to the *EncoderLossFaultAction* attribute.
.EncNsFaultActStatus	BOOL	The status of an update to the *EncoderNoiseFaultAction* attribute.
.FricCompStatus	BOOL	The status of an update to the *FrictionCompensation* attribute.
.MaxNTrvlStatus	BOOL	The status of an update to the *MaximumNegativeTravel* attribute.
.MaxPTrvlStatus	BOOL	The status of an update to the *MaximumPositiveTravel* attribute.
.OutFiltBWStatus	BOOL	The status of an update to the *OutputFilterBandwidth* attribute.
.OutLimitStatus	BOOL	The status of an update to the *OutputLimit* attribute.
.OutOffsetStatus	BOOL	The status of an update to the *OutputOffset* attribute.
.OutScaleStatus	BOOL	The status of an update to the *OutputScaling* attribute.
.PosErrorFaultActStatus	BOOL	The status of an update to the *PositionErrorFaultAction* attribute.
.PosErrorTolStatus	BOOL	The status of an update to the *PositionErrorTolerance* attribute.
.PosIGainStatus	BOOL	The status of an update to the *PositionIntegralGain* attribute.
.PosLockTolStatus	BOOL	The status of an update to the *PositionLockTolerance* attribute.

Variable	Data Type	Description
.PosPGainStatus	BOOL	The status of an update to the *PositionProportionalGain* attribute.
.PosUnwindStatus	BOOL	The status of an update to the *PositionUnwind* attribute.
.POtrvlFactActStatus	BOOL	The status of an update to the *SoftOvertravelFaultAction* attribute.
.VelFfGainStatus	BOOL	The status of an update to the *VelocityFeedforwardGain* attribute.
.VelIGainStatus	BOOL	The status of an update to the *VelocityIntegralGain* attribute.
.VelPGainStatus	BOOL	The status of an update to the *VelocityProportionalGain* attribute.

The MOTION_GROUP Structure

The MOTION_GROUP structure contains status and configuration information for your motion group. There is one MOTION_GROUP structure per controller. You can directly access this information in your motion control program. For example, if you want to use the DriveFault attribute for MOTION_GROUP, you would use MOTION_GROUP.DriveFault to gain access to the attribute.

The bits in the MOTION_GROUP structure are set when any axis in the group experiences the conditions required to set the bit. For example, if one axis in a group of ten axes developed the conditions to set the .POtrvlFault bit, the controller would set the .POtrvlFault bit in the MOTION_GROUP structure.

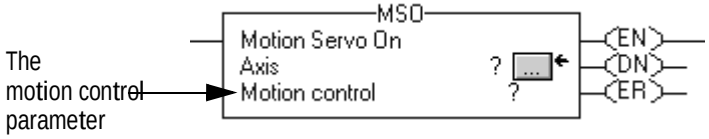
.

Mnemonic:	Data Type:	Description:			
.GroupStatus	DINT	The status bits for the group.			
		Bit:	Number:	Data Type:	Description:
		.InhibStatus	00	BOOL	inhibit status
		.GroupSynced	01	BOOL	synchronization status
.MotionFault	DINT	The motion fault bits for the group.			
		Bit:	Number:	Data Type:	Description:
		.ACAsyncConnFault	00	BOOL	asynchronous connection fault
		.ACSyncConnFault	01	BOOL	synchronous connection fault (controller declared)
.ServoFault	DINT	The servo-module fault bits for the group.			
		Bit:	Number:	Data Type:	Description:
		.POtrvlFault	00	BOOL	positive overtravel fault
		.NOtrvlFault	01	BOOL	negative overtravel fault
		.PosErrorFault	02	BOOL	position error fault
		.EncCHALossFault	03	BOOL	encoder channel A loss fault
		.EncCHBLossFault	04	BOOL	encoder channel B loss fault
		.EncCHZLossFault	05	BOOL	encoder channel Z loss fault
		.EncNsFault	06	BOOL	encoder noise fault
		.DriveFault	07	BOOL	drive fault
		Bit:	Number:	Data Type:	Description:
		.SyncConnFault	00	BOOL	synchronous connection fault (servo declared)
		.HardFault	01	BOOL	servo hardware fault

Mnemonic:	Data Type:	Description:			
.GroupFault	DINT	The fault bits for the group.			
		Bit:	Number:	Data Type:	Description:
		.GroupOverlapFault	00	BOOL	group overlap fault
		.CSTLossFault	01	BOOL	The controller has lost synchronization with the CST master
		.GroupTaskLoadingFault	02	BOOL	The group coarse update period is too low, user application tasks are not getting enough time to execute.

The MOTION_INSTRUCTION Structure

The controller uses the MOTION_INSTRUCTION tag (structure) to store status information during the execution of motion instructions. Every motion instruction has a motion control parameter that requires a MOTION_INSTRUCTION tag for this purpose.

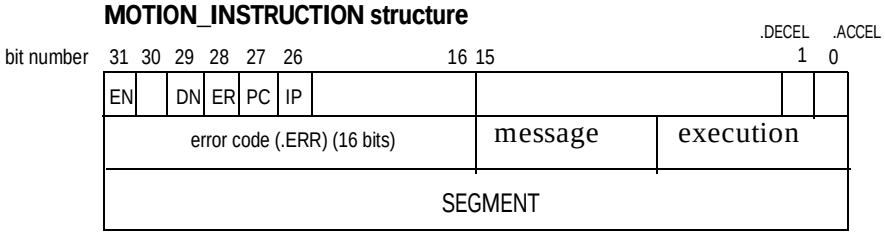


WARNING

Tags used for the motion control attribute of instructions should only be used once. Re-use of the motion control attribute in other instructions can cause unintended operation of the control variables.

The structure of the motion instruction structure is shown below:

MOTION_INSTRUCTION structure



Mnemonic:	Data Type:	Description:
.EN	BOOL	The enable bit indicates that the instruction is enabled (the rung-in and rung-out condition is true).
.DN	BOOL	The done bit indicates that all calculations and messaging (if any) are complete.
.ER	BOOL	The error bit indicates when the instruction is used illegally.
.IP	BOOL	The in process bit indicates that a process is being executed.
.PC	BOOL	The process complete bit indicates that the operation is complete. The .DN bit sets after an instruction has completed execution. The .PC bit sets when the initiated process has completed.
.ACCEL	BOOL	The .ACCEL bit indicates that the velocity has increased for the individual instruction that it is tied to i.e jog, move, gearing
.DECEL	BOOL	The .DECEL bit indicates that the velocity has decreased for the individual instruction that it is tied to i.e jog, move, gearing.
.ERR	INT	The error value contains the error code associated with a motion function. See page 1-8.
.STATUS	SINT	The message status value indicates the status condition of any message associated with the motion function. See page 1-10.
.STATE	SINT	The execution status value keeps track of the execution state of a function. Many motion functions have several steps and this value tracks these steps. See page 1-10.
.SEGMENT	DINT	A segment is the distance from one point up to but, not including the next point. A .SEGMENT gives the relative position by segment number as the Cam is executing.

Error codes (.ERR)

Error Code:	Error Message	Description:
1	Reserved Error Code 1	Reserved for future use.
2	Reserved Error Code 2	Reserved for future use
3	Execution Collision	The instruction tried to execute while another instance of this instruction was executing. This can occur when the controller executes a messaging instruction without checking the .DN bit of the preceding instruction.
4	Servo On State Error	The instruction tried to execute on an axis with a closed servo loop.
5	Servo Off State Error	The instruction tried to execute on an axis with a servo loop that is not closed.
6	Drive On State Error	The axis drive is enabled.
7	Shutdown State Error	The axis is in the shutdown state.
8	Illegal Axis Type	The configured axis type is not correct.
9	Overtravel Condition	The instruction tried to execute in a direction that aggravates the current overtravel condition.
10	Master Axis Conflict	The master axis reference is the same as the slave axis reference.
11	Axis Not Configured	The axis is not configured.
12	Servo Message Failure	Messaging to the servo module failed.
13	Parameter Out Of Range	The instruction tried to use a parameter that is outside the range limit.
14	Tune Process Error	The instruction cannot apply the tuning parameters because of an error in the run tuning instruction.
15	Test Process Error	The instruction cannot apply the diagnostic parameters because of an error in the run diagnostic test instruction.
16	Home In Process Error	The instruction tried to execute with homing in progress.
17	Axis Mode Not Rotary	The instruction tried to execute a rotary move on an axis that is not configured for rotary operation.
18	Axis Type Unused	The axis type is configured as unused.
19	Group Not Synchronized	The motion group is not in the synchronized state. This could be caused by a missing servo module or a misconfiguration.
20	Axis In Faulted State	The axis is in the faulted state.
21	Group In Faulted State	The group is in the faulted state.
22	Axis In Motion	An MSO (Motion Servo On) or MAH (Motion Axis Home) instruction was attempted while the axis was in motion.
23	Illegal Dynamic Change	An instruction attempted an illegal change of dynamics.
24	Illegal AC Mode Op	The controller attempted to execute an MDO, MSO, MAH, MAJ, MAM, MCD, MAPC, MATC, MAG, MRAT, or MRHD instruction when the controller was in the test mode.
25	Illegal Instruction	You attempted to execute an instruction that is not correct.
26	Illegal Cam Length	The cam array is of an illegal length.
27	Illegal Cam Profile Length	The cam profile array is of an illegal length.
28	Illegal Cam Type	You have an illegal segment type in the cam element.
29	Illegal Cam Order	You have an illegal order of cam elements.
30	Cam Profile Being Calculated	You tried to execute a cam profile while it is being calculated.
31	Cam Profile Being Used	The cam profile array you tried to execute is in use.

Error Code:	Error Message	Description:
32	Cam Profile Not Calculated	The cam profile array you tried to execute has not been calculated.
33	Position Cam Not Enabled	It attempted to execute an MAH instruction without a position cam in process.
34	Registration in Progress	A MAH instruction is trying to start while a registration is already running.
35	Illegal Execution Target	Either the Logix controller or the Output Cam module does not support the specified Output Cam, axis, input or output.
36	Illegal Output Cam	Either the size of the Output Cam array is not supported or the value of one of its members is out of range.
37	Illegal Output Compensation	Either the size of the Output Compensation array is not supported or the value of one of its members is out of range.
38	Illegal Axis Data Type	The axis data type is illegal. It is incorrect for the operation.
39	Process Conflict	You have a conflict in your process. Test and Tune cannot be run at the same time.
40	Drive Locally Disabled	You are trying to run a MSO or MAH instruction when the drive is locally disabled.
41	Illegal Homing Config	The Homing configuration is illegal. You have an absolute homing instruction when the Homing sequence is not immediate.

Message status (.STATUS)

Message Status:	Description:
0x0	The message was successful.
0x1	The module is processing another message.
0x2	The module is waiting for a response to a previous message.
0x3	The response to a message failed.
0x4	The module is not ready for messaging.

Execution status (.STATE)

The execution status is always set to 0 when the controller sets the .EN bit for a motion instruction. Other execution states depend on the motion instruction.

Profile Segment (.SEGMENT)

A segment is the distance from one point up to but, not including the next point. A .SEGMENT instruction gives the relative position by segment number as the Cam is executing.

CAM Structure

The Cam data type consists of slave and master point pairs as well as an interpolation type. Since there is no association with a specific axis position or time, the point values are unit-less. The interpolation type can be specified for each segment as either linear or cubic. The format of the cam array element is shown in the following table.

Mnemonic:	Data Type:	Description:
MASTER	REAL	The x value of the point.
SLAVE	REAL	The y value of the point.
Segment Type	DINT	The type of interpolation. Value: Description 0 linear. 1 cubic.

CAM_PROFILE Structure

The CAM_PROFILE data type is an array of coefficients representing a calculated cam profile that can be used as input to a time cam or position cam instruction. The only element available to the user is Status which is defined in the following table.

Mnemonic:	Data Type:	Description:
Status	DINT	The status parameter is used to indicate that the Cam Profile array element has been calculated. If execution of a camming instruction is attempted using an uncalculated element in a Cam Profile, the instruction produces an error. Value: Description 0 Cam profile element has not been calculated. 1 Cam profile element is being calculated. 2 Cam profile element has been calculated. n Cam profile element has been calculated and is currently being used by (n-2) MAPC and MATC instructions.

OUTPUT_CAM Structure

The OUTPUT_CAM data type is an array that defines the specifics for each Output Cam element. The OUTPUT_CAM contains the following members.

Mnemonic	Data Type:	Description:
OutputBit	DINT	You must select an output bit within the range of 0 to 31. A selection of less than 0 or greater than 31 results in an Illegal Output Cam error and the cam element is not considered.
LatchType	DINT	The Latch Type determines how the corresponding output bit is set. A value of less than 0 or greater than 3 results in an Illegal Output Cam error and a latch type of Inactive is used. Value: — Description 0 = Inactive — The output bit is not changed. 1 = Position — The output bit is set when the axis enters the compensated cam range. 2 = Enable — The output bit is set when the enable bit becomes active. 3 = Position and Enable — The output bit is set when the axis enters the compensated cam range and the enable bit becomes active.
UnlatchType	DINT	The Unlatch Type determines how the output bit is reset. Selecting a value less than 0 or greater than 5 results in an Illegal Output Cam error and an unlatch type of Inactive is used. Value: — Description 0 = Inactive — The output bit is not changed. 1 = Position — The output bit is reset when the axis leaves the compensated cam range. 2 = Duration — The output bit is reset when the duration expires. 3 = Enable — The output bit is reset when the enable bit becomes inactive. 4 = Position and Enable — The output bit is reset when the axis leaves the compensated cam range or the enable bit becomes inactive. 5 = Duration and Enable — The output bit is reset when the duration expires or the enable bit becomes inactive.
Left	REAL	The left cam position along with the right cam position define the cam range of the Output Cam element. The left and right cam positions specify the latch or unlatch positions of the output bit when the latch or unlatch type is set to **Position** or **Position and Enable** with the enable bit active. If the left position is less than the Cam Start position or greater than the Cam End position, an Illegal Output Cam error is returned and the cam element is not considered.
Right	REAL	The right cam position along with the left cam position define the cam range of the Output Cam element. The right and left cam positions specify the latch or unlatch positions of the output bit when the latch or unlatch type is set to **Position** or **Position and Enable** with the enable bit active. If the right position is less than the Cam Start position or greater than the Cam End position, an Illegal Output Cam error is returned and the cam element is not considered.
Duration	REAL	Duration specifies the time in seconds between latching and unlatching when the UnlatchType is **Duration** or **Duration and Enable** with the enable bit active. A value less than or equal to 0 results in an Illegal Output Cam error and the cam element is not considered.
EnableType	DINT	This defines the source and polarity of the specified EnableBit when LatchType or UnlatchType is **Enable**, **Position and Enable** or **Duration and Enable**. A value of less than 0 or greater than 31 results in an Illegal Output Cam error and the cam element is not considered. Value: — Description 0 = Input — The enable bit is in the Input parameter. 1 = Inverted Input — The enable bit is in the input parameter and is active low. 2 = Output — The enable bit is in the Output parameter. 3 = Inverted Output — The enable bit is in the Output parameter and is active low.
EnableBit	DINT	The value of the Enable Bit selected must be between 0 and 31 when LatchType or UnlatchType is **Enable**, **Position and Enable** or **Duration and Enable**. A value of less than 0 or greater than 31 results in an Illegal Output Cam error and the cam element is not considered.

OUTPUT_COMPENSATION Structure

The OUTPUT_COMPENSATION data type defines the details for each output bit by setting the characteristics of each actuator. OUTPUT_COMPENSATION contains the following members:

Mnemonic	Data Type:	Description:
Offset	REAL	Offset provides position compensation for both the latch and unlatch operations.
LatchDelay	REAL	Latch delay, programmed in seconds, provides time compensation for the latch operation.
UnlatchDelay	REAL	Unlatch delay, programmed in seconds, provides time compensation for the unlatch operation.
Mode	DINT	The Mode determines the behavior of the output bit. The following four mode options are available. A value of less than 0 or greater than 3 results in an Illegal Output Compensation error. Value: — Description 0 = Normal — The output bit is set for the latch operation and is reset for the unlatch operation. 1 = Inverted — The output bit is reset for the latch operation and is set for the unlatch operation. 2 = Pulsed — The output bit is set for the latch operation and for the on-duty state of the pulse and is reset for the unlatch operation and for the off-duty state of the pulse. 3 = Inverted and Pulsed — The output bit is reset for the latch operation and for the on-duty state of the pulse and is set for the unlatch operation and for the off-duty state of the pulse.
CycleTime	REAL	Pulse time in seconds. If mode is **Pulsed** or **Inverted and Pulsed**, and CycleTime is less than or equal to 0, an Illegal Output Compensation error results.
DutyCycle	REAL	The percent of CycleTime in which the pulse is to be turned on (on-duty). A value of 50 represents 50% on-duty. A value of less than 0 or greater than 100 returns an Illegal Output Compensation error.

The Motion Attributes

This appendix describes the motion attributes, their data types, and their access rules.

The Logix5550 controller stores motion status and configuration information in the AXIS and MOTION_GROUP objects. To directly access this information, you can select the object (AXIS or MOTION_GROUP) and select the attribute. You can also use the GSV and SSV instructions to access these objects. See *Input/Output Instructions* in the Logix5550 Controller Instruction Set Reference Manual, publication 1756-6.4.1 for more information about the GSV and SSV instructions.

Motion Instance Variables

To use the motion instance variables, choose AXIS from the object list of the GSV and SSV instructions.

When an attribute is marked with an asterisk (*), it means that the attribute is located in both the ControlLogix controller and in the motion module. When you use an SSV instruction to write one of these values, the controller will automatically update the copy in the module. However, this process is not immediate. To be sure that the new value has been updated in the module, use an interlock mechanism using the boolean bits in the Servo Configuration Update Status Bits of the AXIS structure.

For example, if you perform an SSV instruction on the PositionLockTolerance, the PositionLockTolStatus of the Axis tag will be set until an update to the module is successful. Therefore, the logic following the SSV could wait on this bit resetting before continuing in the program.

Variable	Data Type	Access	Description
* AccelerationFeedforwardGain	REAL	GSV SSV	The value used to provide the torque command output to generate the command acceleration.
ActualPosition	REAL	GSV	The actual position of your axis.
ActualVelocity	REAL	GSV	The actual velocity of your axis. The internal resolution limit of the actual velocity is 1 encoder count per coarse update.
AverageVelocity	REAL	GSV	The average velocity of your axis.

Variable	Data Type	Access	Description
AverageVelocityTimebase	REAL	GSV SSV	The timebase of the average velocity of your axis.
AxisConfigurationState	SINT	GSV	The state of the axis configuration.
* AxisType	INT	GSV SSV	The type of axis that you are using. **Value: Meaning:** 0 unused axis 1 position-only axis 2 servo axis 3 consumed axis 4 virtual axis
CommandPosition	REAL	GSV	The command position of your axis.
CommandVelocity	REAL	GSV	The command velocity of your axis. The internal resolution limit on the command velocity is 0.00001 encoder counts per coarse update.
ConversionConstant	REAL	GSV SSV	The conversion factor used to convert from your units to feedback counts.
DampingFactor	REAL	GSV SSV	The value used in calculating the maximum position servo bandwidth during the execution of the Motion Run Axis Tuning (MRAT) instruction.
* DriveFaultAction	SINT	GSV SSV	The operation performed when a drive fault occurs. **Value: Meaning:** 0 shutdown the axis 1 disable the drive 2 stop the commanded motion 3 change the status bit only
EffectiveInertia	REAL	GSV	The inertia value for the axis as calculated from the measurements the controller made during the last Motion Run Axis Tuning (MRAT) instruction.
* EncoderLossFaultAction	SINT	GSV SSV	The operation performed when an encoder loss fault occurs. **Value: Meaning:** 0 shutdown the axis 1 disable the drive 2 stop the commanded motion 3 change the status bit only
* EncoderNoiseFaultAction	SINT	GSV SSV	The operation performed when an encoder noise fault occurs. **Value: Meaning:** 0 shutdown the axis 1 disable the drive 2 stop the commanded motion 3 change the status bit only
* FrictionCompensation	REAL	GSV SSV	The fixed output level used to compensate for static friction.
GroupInstance	DINT	GSV	The instance number of the motion group that contains your axis.
HomeMode	SINT	GSV SSV	The homing mode for your axis. **Value: Meaning:** 0 passive homing 1 active homing (default)
HomePosition	REAL	GSV SSV	The homing position of your axis.

Variable	Data Type	Access	Description
HomeReturnSpeed	REAL	GSV SSV	The homing return speed of your axis.
HomeSequenceType	SINT	GSV SSV	The homing sequence type for your axis. **Value:** **Meaning:** 0 immediate homing 1 switch homing 2 marker homing 3 switch-marker homing (default)
HomeSpeed	REAL	GSV SSV	The homing speed of your axis.
INSTANCE	DINT	GSV	The instance number of the axis.
MapTableInstance	DINT	GSV	The I/O map instance of the servo module. This attribute can only be set if you did not assign the axis to a group or if you assigned it to a group in the group inhibit mode.
MaximumAcceleration	REAL	GSV SSV	The maximum acceleration of your axis. The controller automatically sets the maximum acceleration value to approximately 85% of the tuning acceleration determined by the Motion Apply Axis Tune (MAAT) instruction.
MaximumDeceleration	REAL	GSV SSV	The maximum deceleration of your axis. The controller automatically sets the maximum deceleration value to approximately 85% of the tuning deceleration determined by the Motion Apply Axis Tune (MAAT) instruction.
* MaximumNegativeTravel	REAL	GSV SSV	The maximum negative travel limit. This value is always less than the MaximumPositiveTravel value.
* MaximumPositiveTravel	REAL	GSV SSV	The maximum positive travel limit. This value is always greater than the MaximumNegativeTravel value.
MaximumSpeed	REAL	GSV SSV	The maximum speed of your axis. The controller automatically sets the maximum speed value to the tuning speed determined by the Motion Apply Axis Tune (MAAT) instruction.
ModuleChannel	SINT	GSV	The module channel of your servo module. This attribute can only be set if you did not assign the axis to a group or if you assigned it to a group in the group inhibit mode.
MotionConfigurationBits	DINT	GSV SSV	The motion configuration bits for your axis. **Bit:** **Meaning:** 0 home direction reverse 1 home switch normally closed 2 home marker edge negative
MotionFaultBits	DINT	AXIS structure	The motion fault bits for your axis. **Bit:** **Bit Name:** **Meaning:** 0 ACAsyncConnFault asynchronous connection fault 1 ACSyncConnFault synchronous connection fault

Variable	Data Type	Access	Description
MotionStatusBits	DINT	AXIS structure	The motion status bits for your axis. **Bit:** **Bit Name:** **Meaning:** 0 AccelStatus velocity increase 1 DecelStatus velocity decrease 2 MoveStatus move motion profile in progress 3 JogStatus jog motion profile in progress 4 GearingStatus axis is gearing to another axis 5 HomingStatus home motion profile in progress 6 StoppingStatus stopping process in progress 7 AxisHomedStatus absolute position ref established 8 PositionCamStatus Pcam in progress 9 TimeCamStatus Tcam in progress 10 PositionCamPendingStatus Pcam profile waiting for another to end 11 TimeCamPendingStatus Tcam profile waiting for another to end 12 GearingLockedStatus clutching to a new gear speed 13 PositionCamLockStatus master axis meets Pcam condition
MotorEncoderTestIncrement	REAL	GSV SSV	The amount of motion that is necessary to initiate the Motion Run Hookup Diagnostic (MRHD) test.
* OutputFilterBandwidth	REAL	GSV SSV	The bandwidth of the servo low-pass digital output filter.
* OutputLimit	REAL	GSV SSV	The value of the maximum servo output voltage of your axis.
* OutputOffset	REAL	GSV SSV	The value used to offset the effects of the cumulative offsets of the servo module DAC output and the servo drive input.
* OutputScaling	REAL	GSV SSV	The value used to convert the output of the servo loop into the equivalent voltage to the drive. For a velocity servo drive, the output scaling is: $\frac{10Volts}{Speedat10Volts \times ConversionConstant}$ For a torque servo drive, the output scaling is: $\frac{10Volts}{Accelerationat10Volts \times ConversionConstant}$
PositionError	REAL	GSV	The difference between the actual and command position of an axis. You can use this value to drive the motor to where the actual position equals the command position.
* PositionErrorFaultAction	SINT	GSV SSV	The operation performed when a position error fault occurs. **Value:** **Meaning:** 0 shutdown the axis 1 disable the drive 2 stop the commanded motion 3 change the status bit only

Variable	Data Type	Access	Description
* PositionErrorTolerance	REAL	GSV SSV	The amount of position error that the servo tolerates before issuing a position error fault.
* PositionIntegralGain	REAL	GSV SSV	The value used to achieve accurate axis positioning despite disturbances such as static friction and gravity.
PositionIntegratorError	REAL	GSV	The sum of the position error for an axis. You can use this value to drive the motor to where the actual position equals the command position.
PositionLockTolerance	REAL	GSV SSV	The amount of position error that the servo module tolerates when giving a true position locked status indication.
* PositionProportionalGain	REAL	GSV SSV	The value the controller multiples with the position error to correct for the position error.
PositionServoBandwidth	REAL	GSV SSV	The unity gain bandwidth that the controller uses to calculate the gains for a Motion Apply Axis Tuning (MAAT) instruction.
* PositionUnwind	DINT	GSV SSV	The value used to perform the automatic unwind of the rotary axis.
ProgrammedStopMode	SINT	GSV SSV	The type of stop to perform on your axis. **Value:** **Meaning:** 0 fast stop 1 fast shutdown 2 hard shutdown
RegistrationPosition	REAL	GSV	The registration position for your axis. You can use the following equation to determine the maximum registration position error based on your axis speed: $MaximumSpeed\left(\frac{PositionUnits}{Seconds}\right) = \frac{Accuracy(PositionUnits)}{0.000001Seconds}$
* ServoConfigurationBits	DINT	GSV SSV	The servo configuration bits for your servo loop. **Bit:** **Meaning:** 0 rotary axis 1 external velocity servo drive 2 encoder polarity negative 3 servo polarity negative 4 soft overtravel checking 5 position error checking 6 encoder loss fault checking 7 encoder noise fault checking 8 drive fault checking 9 drive fault normally closed

Variable	Data Type	Access	Description
ServoConfigurationUpdateBits	DINT	AXIS structure	The servo configuration status bits for your servo loop. **Bit:** **Bit Name:** **Meaning:** 0 AxisTypeStatus axis type 1 PosUnwndStatus position unwind 2 MaxPTrvlStatus maximum positive travel 3 MaxNTrvlStatus maximum negative travel 4 PosErrorTolStatus position error tolerance 5 PosLockTolStatus position lock tolerance 6 PosPGainStatus position proportional gain 7 PosIGainStatus position integral gain 8 VelFfGainStatus velocity feedforward gain 9 AccFfGainStatus acceleration feedforward gain 10 VelPGainStatus velocity proportional gain 11 VelIGainStatus velocity integral gain 12 OutFiltBwStatus output filter bandwidth 13 OutScaleStatus output scaling 14 OutLimitStatus output limit 15 OutOffsetStatus output offset 16 FricCompStatus friction compensation 17 POtrvlFaultActStatus soft overtravel fault action 18 PosErrorFaultActStatus position error fault action 19 EncLossFaultActStatus encoder loss fault action 20 EncNsFaultActStatus encoder noise fault action 21 DriveFaultActStatus drive fault action 22 ServoConfigBitsStatus update to Servo config bits
ServoEventBits	DINT	AXIS structure	The servo event bits for your servo loop. **Bit:** **Bit Name:** **Meaning:** 0 WatchEvArmStatus watch event armed 1 WatchEvStatus watch event 2 RegEvArmStatus registration event armed 3 RegEvStatus registration event 4 HomeEvArmStatus home event armed 5 HomeEvStatus home event
ServoFaultBits	DINT	AXIS structure	The servo fault bits for your servo loop. **Bit:** **Bit Name:** **Meaning:** 0 POtrvlFault positive overtravel fault 1 NOtrvlFault negative overtravel fault 2 PosErrorFault position error fault 3 EncCHALossFault encoder channel A loss fault 4 EncCHBLossFault encoder channel B loss fault 5 EncCHZLossFault encoder channel Z loss fault 6 EncNsFault encoder noise fault 7 DriveFault drive fault 8 SyncConnFault synchronous connection fault 9 HardFault servo hardware fault
ServoOutputLevel	REAL	GSV	The output voltage level for your axis servo loop.

Variable	Data Type	Access	Description
ServoStatusBits	DINT	AXIS structure	The status bits for your servo loop. **Bit:** **Bit Name:** **Meaning:** 0 ServoActStatus servo action 1 DriveEnableStatus drive enable 2 OutLmtStatus output limit 3 PosLockStatus position lock 5 HomeSwitchStatus state of home input switch 13 TuneStatus tuning process 14 TestStatus test diagnostic 15 ShutdownStatus axis shutdown
ServoStatusUpdateBits	DINT	GSV SSV	The servo status update bits for your axis. **Bit:** **Meaning:** 0 position error update 1 position integrator error update 2 velocity error update 3 velocity integrator error update 4 velocity command update 5 velocity feedback update 6 servo output level update
* SoftOvertravelFaultAction	SINT	GSV SSV	The operation performed when a soft overtravel fault occurs. **Value:** **Meaning:** 0 shutdown the axis 1 disable the drive 2 stop the commanded motion 3 change the status bit only
StartActualPosition	REAL	GSV	The actual position of your axis when new commanded motion starts for the axis. You can use this value to correct for any motion occurring between the detection of an event and the action initiated by the event.
StartCommandPosition	REAL	GSV	The command position of your axis when new commanded motion starts for the axis. You can use this value to correct for any motion occurring between the detection of an event and the action initiated by the event.
StrobeActualPosition	REAL	GSV	The actual position of an axis when the Motion Group Strobe Position (MGSP) instruction executes.
StrobeCommandPosition	REAL	GSV	The command position of an axis when the Motion Group Strobe Position (MGSP) instruction executes.
TestDirectionForward	BOOL	GSV	The direction of axis travel during the Motion Run Hookup Diagnostic (MRHD) instruction as seen by the servo module. **Value:** **Meaning:** 0 negative (reverse) direction 1 positive (forward) direction
TestStatus	UINT16	GSV	The status of the last Motion Run Hookup Diagnostic (MRHD) instruction. **Value:** **Meaning:** 0 test process successful 1 test in progress 2 test process aborted by the user 3 test exceeded 2-second time-out 4 test process failed due to servo fault 5 insufficient test increment

Variable	Data Type	Access	Description
TuneAcceleration	REAL	GSV	The acceleration value measured during the last Motion Run Axis Tuning (MRAT) instruction.
TuneAccelerationTime	REAL	GSV	The acceleration time in seconds measured during the last Motion Run Axis Tuning (MRAT) instruction.
TuneDeceleration	REAL	GSV	The deceleration value measured during the last Motion Run Axis Tuning (MRAT) instruction.
TuneDecelerationTime	REAL	GSV	The deceleration time in seconds measured during the last Motion Run Axis Tuning (MRAT) instruction.
TuneRiseTime	REAL	GSV	The axis rise time in seconds measured during the last Motion Run Axis Tuning (MRAT) instruction. This value only applies to axes that you configure to work with an external velocity servo drive.
TuneSpeedScaling	REAL	GSV	The axis drive scaling factor measured during the last Motion Run Axis Tuning (MRAT) instruction. This value only applies to axes that you configure to work with an external velocity servo drive.
TuneStatus	UINT16	GSV	The status of the last Motion Run Axis Tuning (MRAT) instruction. **Value: Meaning:** 0 tune process successful 1 tuning in progress 2 tune process aborted by user 3 tune exceeded 2-second time-out 4 tune process failed due to servo fault 5 axis reached tuning travel limit 6 axis polarity set incorrectly 7 tune speed is too small to make measurements
TuneVelocityBandwidth	REAL	GSV	The bandwidth of the drive as calculated from the measurements made during the last Motion Run AxisTuning (MRAT) instruction.
TuningConfigurationBits	DINT	GSV SSV	The tuning configuration bits for your axis. **Bit: Meaning:** 0 tuning direction (0=forward, 1=reverse) 1 tune position error integrator 2 tune velocity error integrator 3 tune velocity feedforward 4 acceleration feedforward 5 tune velocity low-pass filter
TuningSpeed	REAL	GSV SSV	The maximum speed reached by the Motion Run Axis Tuning (MRAT) instruction.
TuningTravelLimit	REAL	GSV SSV	The travel limit used by the Motion Run Axis Tuning (MRAT) instruction to limit the action of the axis during tuning.
VelocityCommand	REAL	GSV	The current velocity reference to the velocity servo loop for an axis.
VelocityError	REAL	GSV	The difference between the commanded and actual velocity of a servo axis. You can use this value to drive the motor to where the velocity feedback equals the velocity command.
VelocityFeedback	REAL	GSV	The actual velocity of your axis as estimated by the servo module. To estimate the velocity, the servo module applies a 1 kHz low-pass filter to the change in actual position in one update interval.

Variable	Data Type	Access	Description
* VelocityFeedforwardGain	REAL	GSV SSV	The value used to provide the velocity command output to generate the command velocity.
* VelocityIntegralGain	REAL	GSV SSV	The value that the controller multiplies with the VelocityIntegratorError value to correct the velocity error.
VelocityIntegratorError	REAL	GSV	The sum of the velocity error for a specified axis. You can use this value to drive the motor to where the velocity feedback equals the velocity command.
* VelocityProportionalGain	REAL	GSV SSV	The value that the controller multiplies with the VelocityError to correct the velocity error.
WatchPosition	REAL	GSV	The watch position of your axis.

Instruction Timing

This appendix describes motion instruction timing types.

Motion instructions use three types of timing sequences

Timing type	Description
Immediate	The instruction completes in one scan.
Message	The instruction completes over several scans because the instruction sends messages to the servo module.
Process	The instruction could take an indefinite amount of time to complete.

Immediate Type Instructions

Immediate type motion instructions execute to completion in one scan. If the controller detects an error during the execution of these instructions, the error status bit sets and the operation ends.

Examples of immediate type instructions include the:

- Motion Change Dynamics (MCD) instruction
- Motion Group Strobe Position (MGSP) instruction

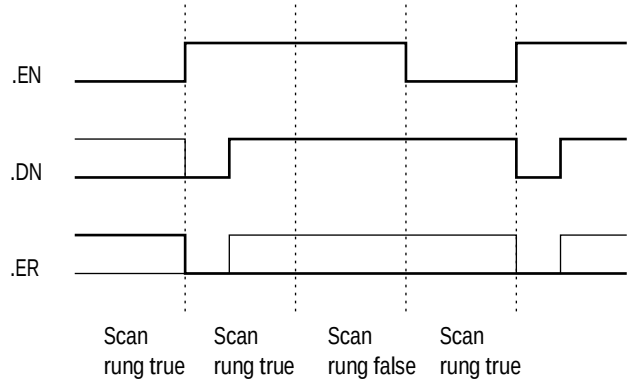
Immediate instructions work as follows:

1. When the rung that contains the motion instruction becomes true, the controller:
   - Sets the enable (EN) bit.
   - Clears the done (DN) bit.
   - Clears the error (ER) bit.
2. The controller executes the instruction completely.
3.

If the controller	Then
Does not detect an error when the instruction executes	The controller sets the .DN bit.
Detects an error when the instruction executes	The controller sets the .ER bit and stores an error code in the control structure.

4. The next time the rung becomes false after either the .DN or .ER bit sets, the controller clears the .EN bit.

5. The controller can execute the instruction again when the rung becomes true.



Message Type Instructions

Message type motion instructions send one or more messages to the servo module.

Examples of message type instructions include the:

- Motion Direct Drive On (MDO) instruction
- Motion Redefine Position (MRP) instruction

Message type instructions work as follows:

1. When the rung that contains the motion instruction becomes true, the controller:
   - Sets the enable (EN) bit.
   - Clears the done (DN) bit.
   - Clears the error (ER) bit.
2. The controller begins to execute the instruction by setting up a message request to the servo module.

The remainder of the instruction executes in parallel to the program scan.

3. The controller checks if the servo module is ready to receive a new message.
4. The controller places the results of the check in the message status word of the control structure.

5. When the module is ready, the controller constructs and transmits the message to the module.

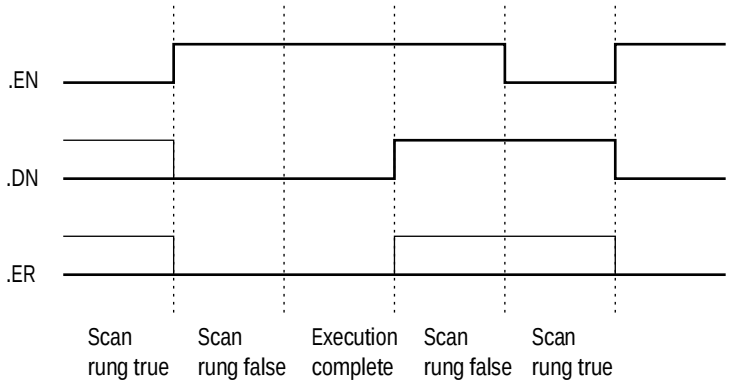
This process may repeat several times if the instruction requires multiple messages.

6.

If the controller	Then
Does not detect an error when the instruction executes	The controller sets the .DN bit.
Detects an error when the instruction executes	The controller sets the .ER bit and stores an error code in the control structure.

7. The next time the rung becomes false after either the .DN or .ER bit sets, the controller clears the .EN bit.

8. When the rung becomes true, the controller can execute the instruction again.



Process Type Instructions

Process type motion instructions initiate motion processes that can take an indefinite amount of time to complete.

Examples of process type instructions include the:

- Motion Arm Watch Position (MAW) instruction
- Motion Axis Move (MAM) instruction

Process type instructions work as follows:

1. When the rung that contains the motion instruction becomes true, the controller:
   - Sets the enable (.EN) bit.
   - Clears the done (.DN) bit.
   - Clears the error (.ER) bit.
   - Clears the process complete (.PC) bit.
2. The controller initiates the motion process.
3.

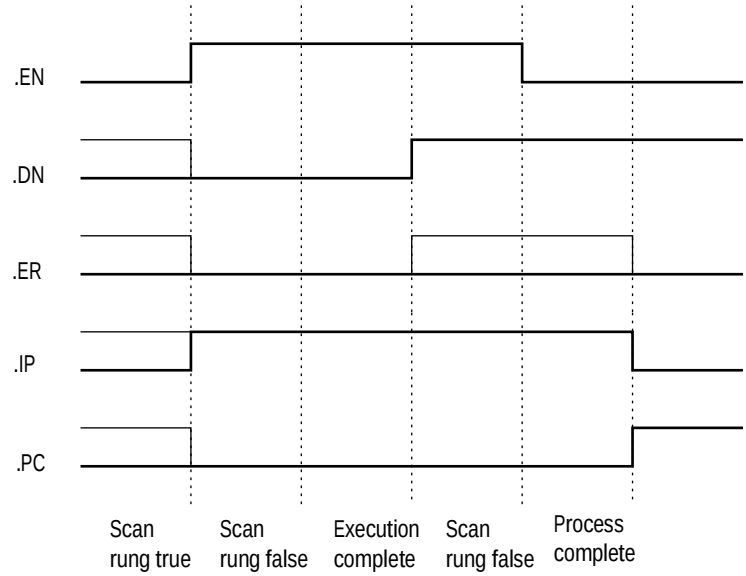
If	Then the controller
The controller does not detect an error when the instruction executes	• Sets the .DN bit. • Sets the in process (.IP) bit.
The controller detects an error when the instruction executes	• Sets the .ER bit. • Stores an error code in the control structure. • Does not change the .IP and .PC bits.
The controller detects another instance of the motion instruction	Clears the .IP bit for that instance.
The motion process reaches the point where the instruction can be executed again	Sets the .DN bit. For some process type instructions, like MAM, this will occur on the first scan. For others, like MAH, the .DN bit will not be set until the entire homing process is complete.
One of the following occurs during the motion process: • The motion process completes • Another instance of the instruction executes • Another instruction stops the motion process • A motion fault stops the motion process	• Sets the .DN bit. • Sets the .PC bit. • Clears the .IP bit.

4. Once the initiation of the motion process completes, the program scan can continue.

The remainder of the instruction and the control process continue in parallel with the program scan.

5. The next time the rung becomes false after either the .DN bit or the .ER bit sets, the controller clears the .EN bit.

6. When the rung becomes true, the instruction can execute again.



Fault Handling

This appendix describes motion errors and faults.

Handling Motion Faults

Two types of motion faults exist.

Type	Description	Example
Errors	• Do not impact controller operation • Should be corrected to optimize execution time and ensure program accuracy	A Motion Axis Move (MAM) instruction with a parameter out of range
Minor/ Major	• Caused by a problem with the servo loop • Can shutdown the controller if you do not correct the fault condition	The application exceeded the PositionErrorTolerance value

Errors

Executing a motion instruction within an application program can generate errors. The MOTION_INSTRUCTION tag has a field that contains the error code. For more information on error codes for individual instructions, refer to the motion instruction chapters in the Logix5550 Controller Instruction Set Reference Manual.

Minor/Major Faults

Several faults can occur that are not caused by motion instructions. For example, a loss of encoder feedback or actual position exceeding an overtravel limit will cause faults. The motion faults are considered Type 11 faults with error codes from 1 to 32. For more information about motion error codes, refer to *Handling Controller Faults* in the Logix5550 Controller User Manual.

TIP You can configure a fault as either minor (non major) or major by using the Axis Wizard-Group window.

For more information about handling faults, see *Handling Controller Faults* in the Logix5550 Controller User Manual, publication 1756-6.5.12.

Symbols

Numerics

A

N

O

P

R

S

T

U

W

Reach us now at www.rockwellautomation.com

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

Americas Headquarters, 1201 South Second Street, Milwaukee, WI 53201-2496, USA, Tel: (1) 414 382-2000, Fax: (1) 414-382-4444
European Headquarters SA/NV, Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1756-UM006B-EN-P- July 2001
Supersedes Publication 1756-UM006A-EN-P - April 2001
PN 957564-02