



**Allen-Bradley**

## **SoftLogix™ Motion Card**

(Cat.No. 1784-PM02AE, -PM16SE)

**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.

Reproduction of the contents of this copyrighted publication, in whole or part, without written permission of Rockwell Automation, is prohibited.

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.

---

Allen-Bradley is a trademark of Rockwell Automation

RSLogix, SoftLogix, Ultra3000, Ultra5000, and Kinetix 6000 are all trademarks of Rockwell Automation

SERCOS interface is a trademark of the Interests group SERCOS interface e.V. of Stuttgart, Germany

## **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**

**Using This Manual**

Who Should Use This Manual. . . . . 1  
 The Purpose of This Manual. . . . . 1  
 Related Documentation. . . . . 2  
 Rockwell Automation Support . . . . . 2  
     Local Product Support . . . . . 3  
     Technical Product Assistance . . . . . 3  
 On the Web . . . . . 3

**Chapter 1**

**The SoftLogix Motion Control System**

SoftLogix Motion Control. . . . . 5  
 Components of the SoftLogix Motion System. . . . . 6  
     SoftLogix Chassis Monitor . . . . . 7  
     The SoftLogix Controller . . . . . 8  
     The Analog/Encoder Servo Module (1784-PMO2AE) . . . . . 9  
     The 16 Axis SERCOS interface Module (1784-PM16SE). . . . . 9  
     RSLogix 5000 Programming Software . . . . . 9  
 Developing a Motion Control Application Program . . . . . 10  
     Application Program Development . . . . . 10  
     The MOTION\_INSTRUCTION Tag. . . . . 10  
     Motion Status and Configuration Parameters . . . . . 11  
     Modifying Motion Configuration Parameters . . . . . 11  
     Handling Motion Faults. . . . . 12

**Chapter 2**

**SoftLogix Controller**

Introduction . . . . . 13  
 Accessing the New Controller Dialog. . . . . 14  
 Editing Controller Properties . . . . . 17

**Chapter 3**

**Adding and Configuring Your 1784-PMO2AE Motion Module**

Adding the 1784-PMO2AE Module . . . . . 45  
 Editing Your Motion Module Settings . . . . . 51  
 Assigning Additional Motion Modules . . . . . 60

---

	<b>Chapter 4</b>	
<b>Configuring the 1784-PM16SE Card</b>	Adding the 1784-PM16SE.....	61
	SERCOS interface Motion Card Overview .....	66
	Editing 1784-PM16SE Card Properties .....	67
	<b>Chapter 5</b>	
<b>The Motion Group</b>	Creating A Motion Group.....	79
	Editing the Motion Group Properties.....	82
	<b>Chapter 6</b>	
<b>Naming &amp; Configuring Your Motion Axis</b>	Naming an Axis .....	87
	Editing Motion Axis Properties.....	91
	Assigning Additional Motion Axes .....	171
	<b>Chapter 7</b>	
<b>Creating &amp; Configuring Your Coordinate System Tag</b>	Introduction .....	173
	Creating a Coordinate System .....	173
	Editing Coordinate System Properties .....	179
	Right Mouse Click Properties .....	188
	<b>Chapter 8</b>	
<b>Configuring a 1394x-SJTxx-D Digital Servo Drive</b>	1394x-SJTxx-D Digital Servo Drive Overview .....	193
	<b>Chapter 9</b>	
<b>Configuring an Ultra 3000 Drive</b>	Editing the Ultra Drive Properties .....	211
	<b>Chapter 10</b>	
<b>Configuring a Kinetix 6000 Drive</b>	Editing the Kinetix Drive Properties .....	225
	<b>Chapter 11</b>	
<b>Configuring an 8720MC Drive</b>	Editing the 8720MC Drive Properties .....	240

## Chapter 12

### Motion Instructions

Motion State Instructions . . . . .	253
Motion Move Instructions . . . . .	254
Motion Group Instructions . . . . .	255
Motion Event Instructions . . . . .	255
Motion Configuration Instructions . . . . .	255
Motion Coordinated Move Instructions . . . . .	256
Motion Direct Commands . . . . .	257
Accessing Direct Commands . . . . .	257
Supported Commands . . . . .	262
Motion Direct Command Dialog . . . . .	264

## Chapter 13

### Motion Object Attributes

Object Support Attributes . . . . .	267
Motion Object Status Attributes . . . . .	270
Motion Status Attributes . . . . .	271
Motion Status Bit Attributes . . . . .	279
Motion Object Configuration Attributes . . . . .	287
Motion Unit Configuration Attributes . . . . .	287
Motion Conversion Configuration . . . . .	288
Motion Homing Configuration . . . . .	289
Motion Dynamics Configuration . . . . .	299
Servo Status Attributes . . . . .	301
Servo Status Bit Attributes . . . . .	306
Commissioning Status Attributes . . . . .	315
Servo Configuration Attributes . . . . .	319
Servo Limits . . . . .	337
Servo Offsets . . . . .	340
Servo Fault Configuration . . . . .	342
Commissioning Configuration Attributes . . . . .	343
Servo Drive Status Attributes . . . . .	348
Commissioning Status Attributes . . . . .	369
Servo Drive Configuration Attributes . . . . .	372
Motor and Feedback Configuration . . . . .	380
Servo Loop Block Diagrams . . . . .	382
Drive Limits . . . . .	398
Drive Offsets . . . . .	400
Drive Fault Actions . . . . .	401
Drive Power Attributes . . . . .	404
Commissioning Configuration Attributes . . . . .	405

---

## Chapter 14

### Troubleshooting

1784-PM02AE LED Indicator . . . . .	411
SERCOS interface LED Indicators . . . . .	412

### Appendix A

### Specifications and Performance

1784-PM02AE Motion Card Specifications . . . . .	417
1784-M16SE Motion Card Specifications . . . . .	418

### Appendix B

### Wiring Diagrams

About Block Diagrams . . . . .	421
Using a 1784-PM02AE Module With a Torque Servo Drive . . .	422
Using a 1784-PM02AE Module With a Velocity Servo Drive . .	423
Wiring Diagrams . . . . .	423
1784-PM02AE Termination Panel . . . . .	423
Pinouts for Cable 1784-PM02AE-TP0x . . . . .	426
1398 to Termination Panel Wiring Diagram . . . . .	428



## Using This Manual

### Who Should Use This Manual

To use this manual, you should be able to program and operate the Allen-Bradley SoftLogix5800™ controllers to efficiently use your motion control modules. The manual's focus is from the RSLogix 5000 software.

If you need more information about programming and operating the SoftLogix5800 controllers, refer to the *SoftLogix5800 System User Manual*, publication number 1789-UM002.

### The Purpose of This Manual

This manual describes how to setup and configure and troubleshoot your SoftLogix motion cards using the RSLogix 5000 software. The following table shows the contents of each section in this manual:

Section	Contains
Chapter 1 The ControlLogix Motion Control System	Information about the SoftLogix motion control system.
Chapter 2 Controller Properties	Explains how to create and edit your controller.
Chapter 3 Adding and Configuring Your 1784-PM02AE Motion Card	How to add and configure your 1784-PM02AE motion module using the RSLogix™ 5000 programming software.
Chapter 4 Adding and Configuring Your 1784-PM16SE Motion Card	How to add and configure your 1784-PM16SE motion card 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 5000 programming software.
Chapter 7 Creating and Configuring a Coordinate System	How to create and configure a Coordinated System.
Chapter 8 Configuring a 1394C-SJT05/10/22-D Digital Servo Drive	How to add and configure a 1394C Digital Servo Drive.
Chapter 9 Configuring an Ultra3000™ Drive	How to add and configure an Ultra3000 drive
Chapter 10 Configuring the Kinetix 6000™ Drive	How to add and configure a Kinetix 6000 drive
Chapter 11 Configuring the 8720MC High Performance Drive	How to add and configure an 8720MC High Performance Drive
Chapter 12 The Motion Instructions	Information about the motion instructions provided in the RSLogix 5000 programming software.
Chapter 13 Motion Object Attributes	Information about the Motion Object attributes.
Chapter 14 Troubleshooting	Information about troubleshooting your SoftLogix motion control system.

Appendix A Specifications	Specifications and performance guidelines for the motion card.
Appendix B Loop and Interconnect Diagrams	Loop diagrams and wiring diagrams for your SoftLogix motion control system.

## Related Documentation

The following table lists related ControlLogix documentation:

Publication Number	Publication	Description
1784-IN047	<i>Analog Encoder (AE) Servo Module Installation Instructions</i>	Provides instructions for installing, wiring, and troubleshooting your 1784-PM02AE servo card.
1789-UM002	<i>SoftLogix5800 System User Manual</i>	Provides information for using your SoftLogix5800 controller and its components.
1756-RM007	<i>Motion Instructions Set Reference Manual</i>	Provides descriptions of all of the motion instructions used in the RSLogix 5000 software.
1784-IN041	<i>16 Axis SERCOS interface Card Installation Instructions</i>	Provides instructions for installing, wiring, and troubleshooting your 1784-PM16SE SERCOS interface™ card.
1394C-5.20	<i>1394 SERCOS Interface Multi Axis Motion Control System</i>	Information on installation, wiring, and setup for the 1394C-SJTxx-D
1394-IN024	<i>1394 SERCOS Integration Manual</i>	Information on integrating the 1394 drive with the 1784-PM16SE
2098-IN003	<i>Ultra3000 Hardware Installation Manual</i>	Information on Ultra3000 installation
2098-IN001	<i>Ultra5000 Hardware Installation Manual</i>	Information on Ultra5000 installation
2098-IN005	<i>Ultra3000 SERCOS Integration Manual</i>	Information on integrating the Ultra3000 with the 1784-PM16SE SERCOS interface modules
8720MC-UM001	<i>8720MC High Performance Drive User Manual</i>	Information on use of the 8720MC High Performance Drive
2094-IN004	<i>Kinetix 6000 Module Installation Manual</i>	Information on installing the Kinetix 6000 Module
2094-IN001	<i>Kinetix 6000 Installation Manual</i>	Information on installing Kinetix 6000
2094-IN002	<i>Kinetix 6000 Integration Manual</i>	Information on integrating the Kinetix 6000 with SERCOS

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:

1-440-646-5800

## On the Web

For information about Rockwell Automation, visit the following World Wide Web site:

**<http://www.ab.com/support>**



---

## The SoftLogix Motion Control System

This chapter describes the SoftLogix motion control system and its components.

### SoftLogix Motion Control

The Virtual Chassis, SoftLogix controller, 1784-PM02AE servo card, 1784-PM16SE SERCOS interface card, and RSLogix 5000 programming software provide integrated motion control support.

- The Chassis Monitor lets you create, delete, monitor, and configure controllers and communication interface cards such as ControlNet, DeviceNet, 1784-PM02AE and 1784-PM16SE motion cards in the SoftLogix Virtual Chassis.
- The SoftLogix5860 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 1784-PM02AE servo modules. You can use several Logix controllers in each chassis. Each controller can control up to 32 axes of motion.
- The 1784-PM02AE servo module connects to a servo drive and closes a high-speed position and velocity loop. Each SoftLogix controller can support up to 16 1784-PM02AE servo modules. Each 1784-PM02AE module can control up to two axes.
- The 1784-PM16SE SERCOS interface module serves as the interface between one SoftLogix processor and 1 to 16 axes operating in either position or velocity mode. The module has a programmable ring Cycle Period of 0.5 ms, 1 ms, or 2 ms depending on the number of axes and a ring Data Rate of 4 or 8 Mbaud.
- RSLogix 5000 programming software provides complete axis configuration and motion programming support.

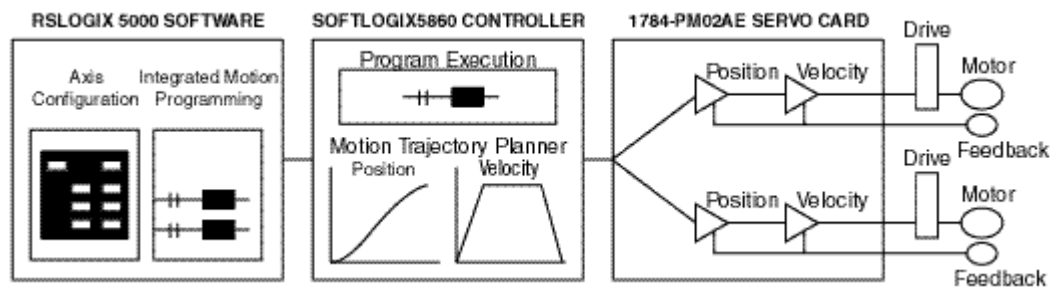


Figure 1.1 SoftLogix System with 1784-PM02AE

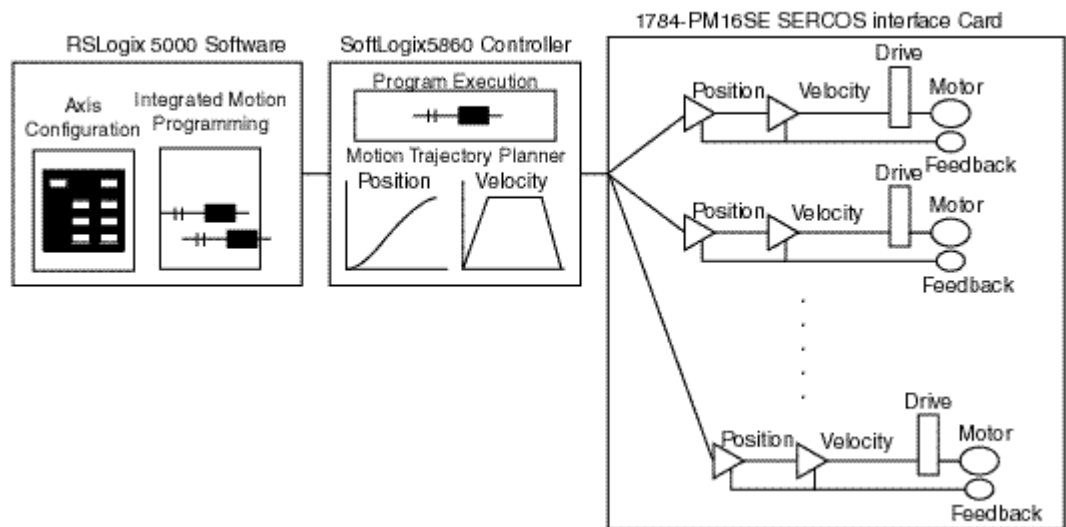
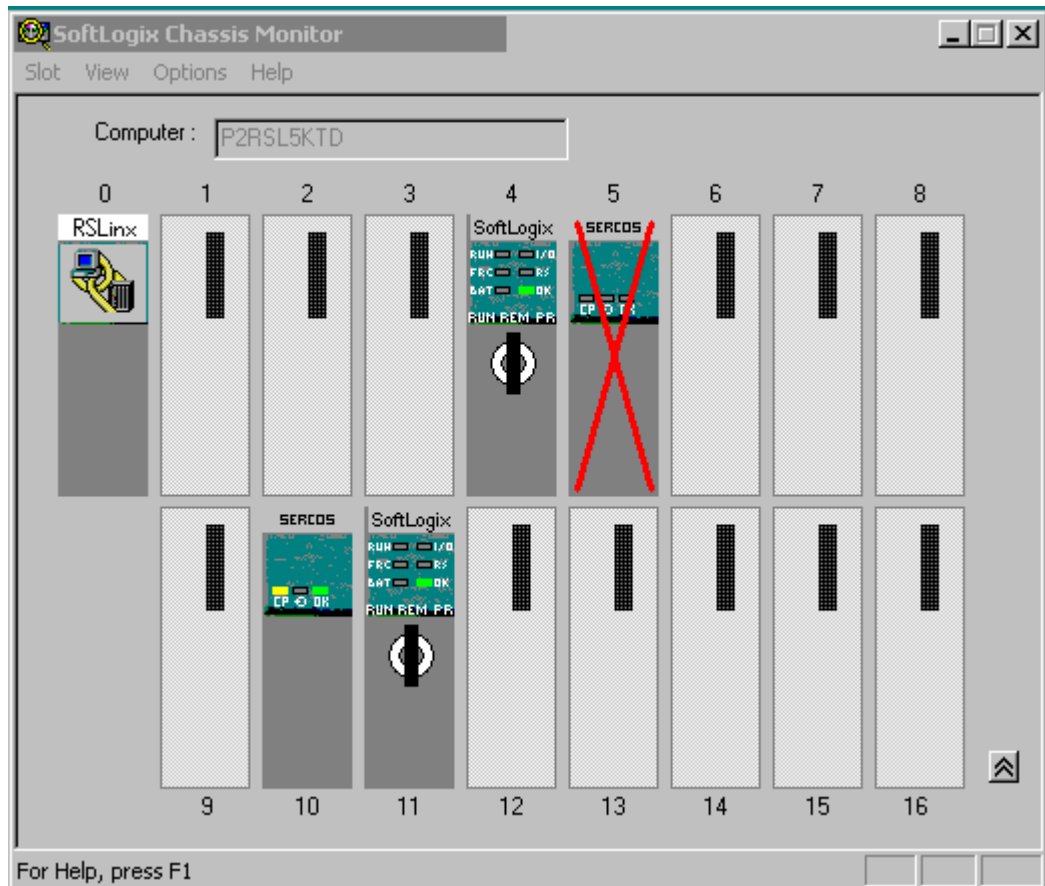


Figure 1.2 SoftLogix System with 1784-PM16SE

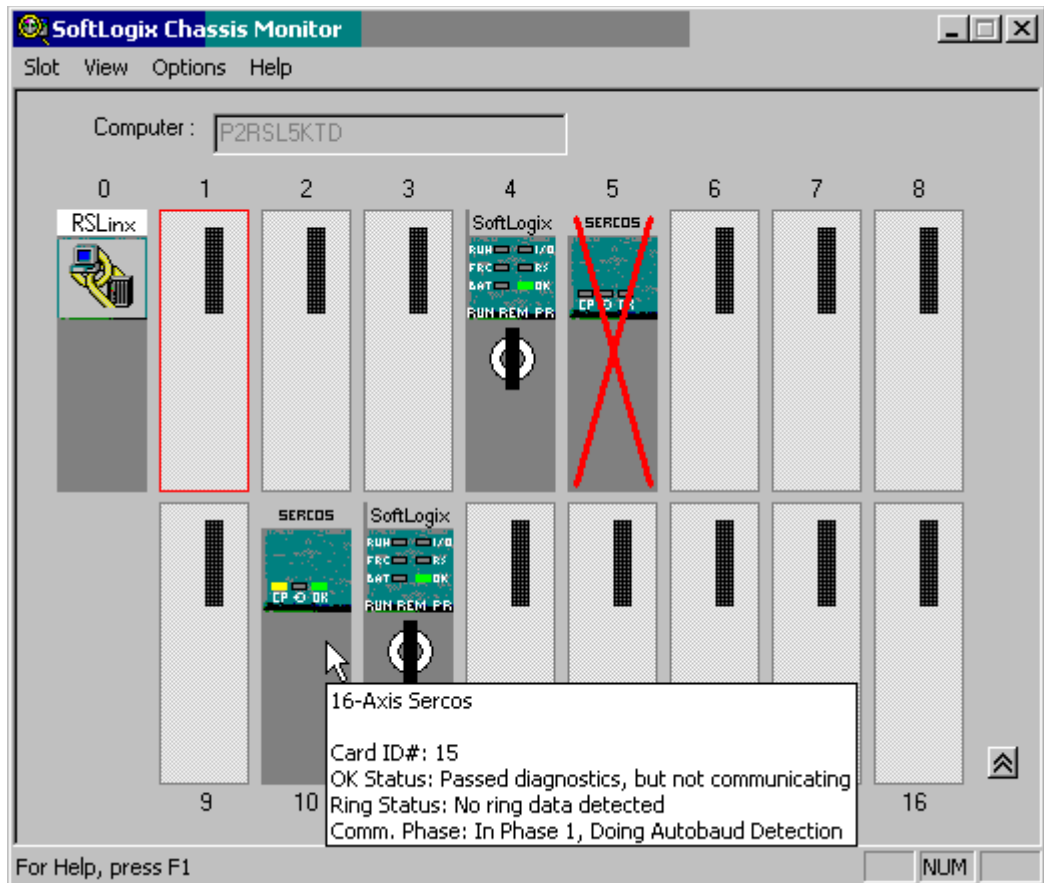
## Components of the SoftLogix Motion System

**SoftLogix Chassis Monitor** It is at the Chassis Monitor where you can change the processor mode, and view system status. After you have created and configured the various modules of your SoftLogix system you can use the Chassis Monitor to display the virtual chassis where you can monitor the simulated LEDs to view status information for your modules.



**Figure 1.3 SoftLogix Chassis Monitor**

Additional information in the form of a tooltip about the modules can be ascertained by placing the mouse over a specific module.



**Figure 1.4 Additional Information for the PM16SE Card**

**The SoftLogix Controller** The SoftLogix controller is the main component in the SoftLogix 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 SoftLogix controllers in each chassis, and each controller can control up to 32 axes of motion.

The SoftLogix controller provides the following motion support:

- 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
- SERCOS support



**The Analog/Encoder Servo Module (1784-PM02AE)** The Analog/Encoder servo module provides an analog/quadrature encoder servo drive interface. The servo module receives configuration and move information from the SoftLogix controller and manages motor position and velocity.

The servo module supports:

- Connection capability for up to two drives
- $\pm 10\text{V}$  analog outputs
- Quadrature encoder inputs
- Home limit switch inputs
- Drive fault inputs
- Drive enable outputs
- 5V or 24V position registration inputs
- 250  $\mu\text{s}$  position and velocity loop updates

**The 16 Axis SERCOS interface Module (1784-PM16SE)** The 16 Axis SERCOS interface modules (1784-PM16SE) serves as a link between the SoftLogix platform and intelligent drives. The communication link between the module and the drive(s) is via IEC 1491 **SER**ial **R**eal-time **C**ommunication **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

**RSLogix 5000 Programming Software** The RSLogix 5000 programming software provides complete programming and commissioning support for the SoftLogix system. RSLogix 5000 is the only programming software needed to fully configure and program SoftLogix motion control systems.

RSLogix 5000 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 31 motion commands

## Developing a Motion Control Application Program

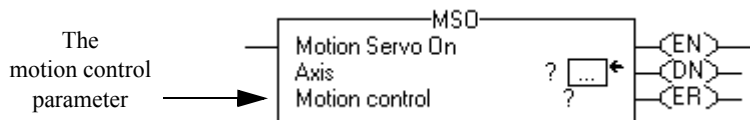
This section provides an introduction to concepts used in developing application programs for motion control. These concepts include:

- 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 SoftLogix 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.



**Figure 1.5 Motion Control Parameter**

**ATTENTION**

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 the *Logix5000 Controller Motion Instruction Set Reference Manual* (1756-RM007).

### 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	<ul style="list-style-type: none"> <li>• Axis faults</li> <li>• Motion status</li> <li>• Servo status</li> </ul>	Refer to the <i>Logix5000 Controller Motion Instruction Set Reference Manual</i> (1756-RM007).
Using the GSV instruction	<ul style="list-style-type: none"> <li>• Actual position</li> <li>• Command position</li> <li>• Actual velocity</li> </ul>	Refer to the <i>Input/Output Instructions</i> chapter of the <i>Logix Controller Instruction Set Reference Manual</i> , publication 1756-RM003.

### 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	<ul style="list-style-type: none"> <li>Do not impact controller operation</li> <li>Should be corrected to optimize execution time and ensure program accuracy</li> </ul>	A Motion Axis Move (MAM) instruction with a parameter out of range
Minor/Major Faults	<ul style="list-style-type: none"> <li>Caused by a problem with the servo loop</li> <li>Can shutdown the controller if you do not correct the fault condition</li> </ul>	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 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 Logix 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 *Logix 5000 Controller User Manual*.

# SoftLogix Controller

## Introduction

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

To 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.
5. Enter the slot number for the controller.

In SoftLogix, 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.

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.

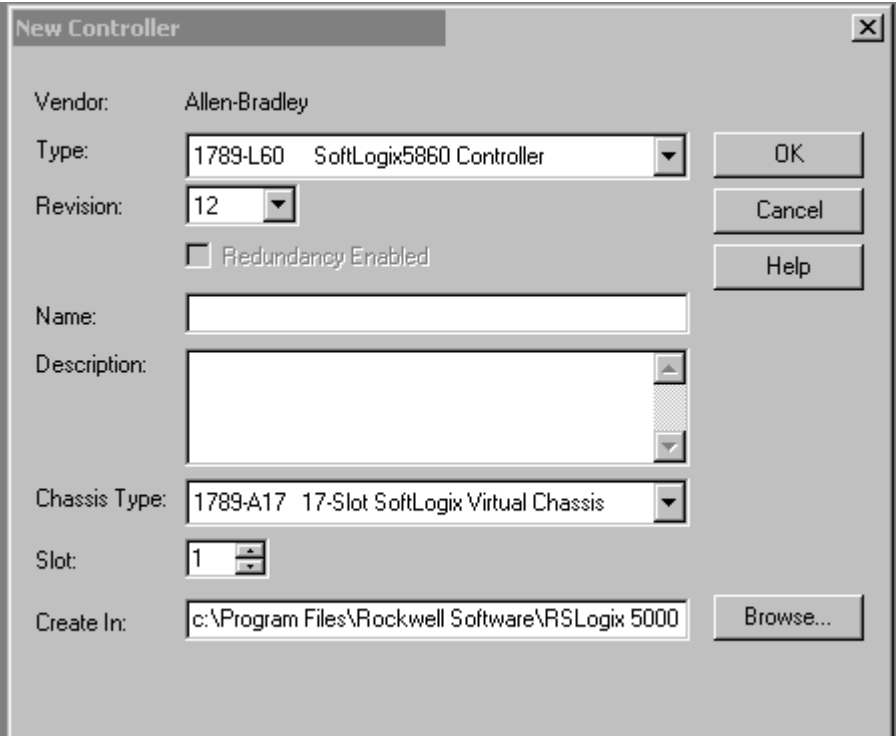
In addition, if you have chosen a FlexLogix controller, 2 FlexBus adapters are created in slots 3 and 4 under the I/O Configuration folder. These 2 folders contain all local I/O for FlexLogix, other than the 2 local slots for communication. The first folder contains all I/O configured on the local Flex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



The image shows a screenshot of the 'New Controller' dialog box. The dialog has a title bar with the text 'New Controller' and a close button (X). The main area contains the following fields and controls:

- Vendor:** Allen-Bradley
- Type:** 1789-L60 SoftLogix5860 Controller (dropdown menu)
- Revision:** 12 (dropdown menu)
- Redundancy Enabled
- Name:** (empty text box)
- Description:** (empty text box)
- Chassis Type:** 1789-A17 17-Slot SoftLogix Virtual Chassis (dropdown menu)
- Slot:** 1 (dropdown menu)
- Create In:** c:\Program Files\Rockwell Software\RSLogix 5000 (text box with a 'Browse...' button to its right)

On the right side of the dialog, there are three buttons: 'OK', 'Cancel', and 'Help'.

**Figure 2.1** New Controller 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.

Choose from:

- 1756-L1 ControlLogix5550 Controller
- 1756-L55 ControlLogix5555 Controller
- 1756-L61 ControlLogix5561 Controller
- 1756-L62 ControlLogix5562 Controller
- 1756-L63 ControlLogix5563 Controller
- 1756-LCE ControlLogixCE Controller
- 1769-L20 CompactLogix5320 Controller
- 1769-L30 CompactLogix5330 Controller
- 1769-L35E CompactLogix5335E Controller
- 1789-L60 SoftLogix5860 Controller
- 1794-L33 FlexLogix5433 Controller
- 1794-L34 FlexLogix5434 Controller
- 618x-LCE PanelViewLogix Controller
- Emulator RSLogix Emulate 5000 Controller
- PowerFlex700S DriveLogix5720 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.

## 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.

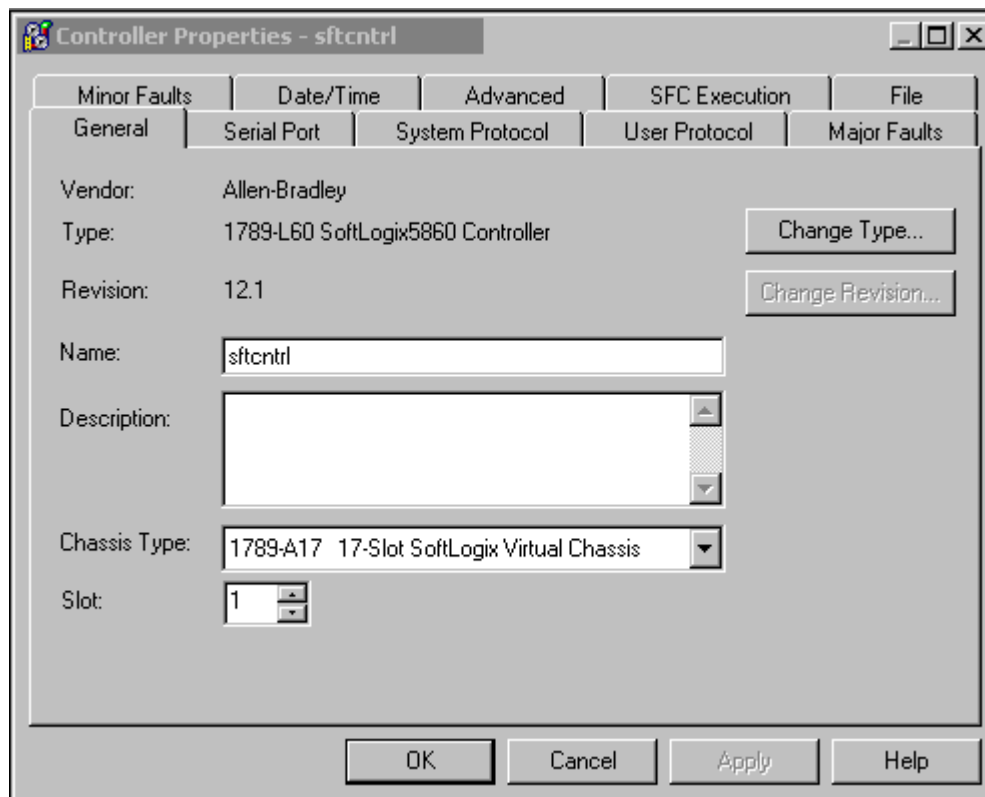


## Editing Controller Properties

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller. The Tabs that appear are governed by the type of the selected 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.



**Figure 2.2 Controller Properties General Tab**

### 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

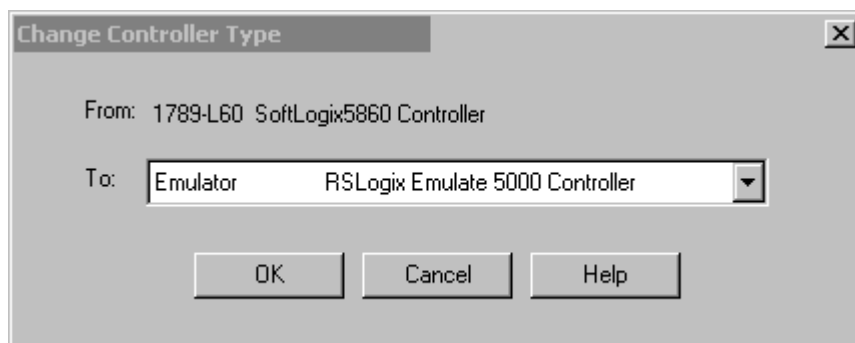
Displays the major and minor firmware revision of the controller. The minor revision is available only when you are online.

## 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

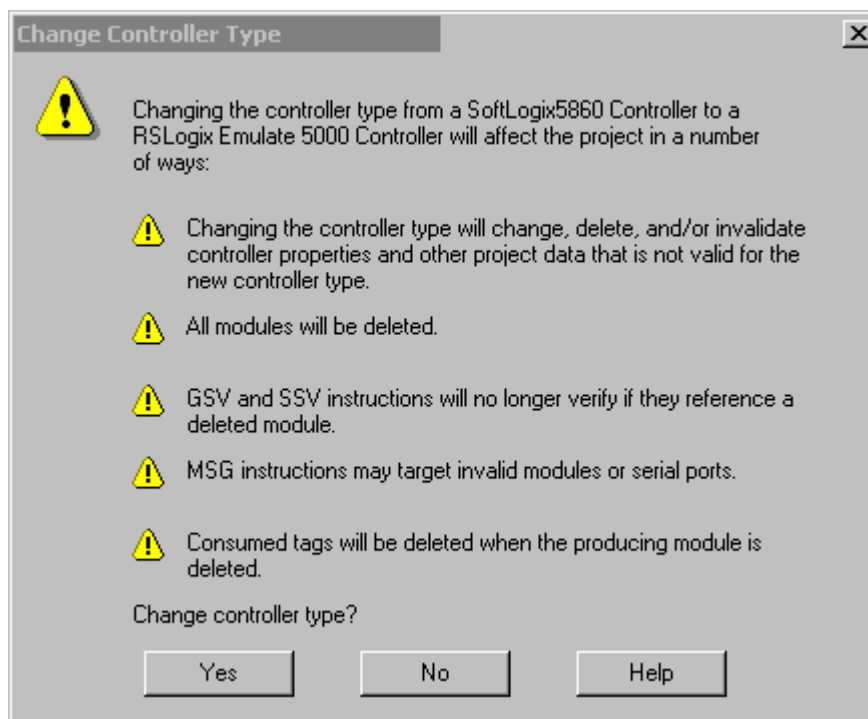
Use this dialog to change your controller to another controller within the same platform (e.g. changing from a 1756-L1 ControlLogix 5550 Controller to a 1756-L53/A ControlLogix 5553 controller).



**Figure 2.3 Change Controller Type Dialog**

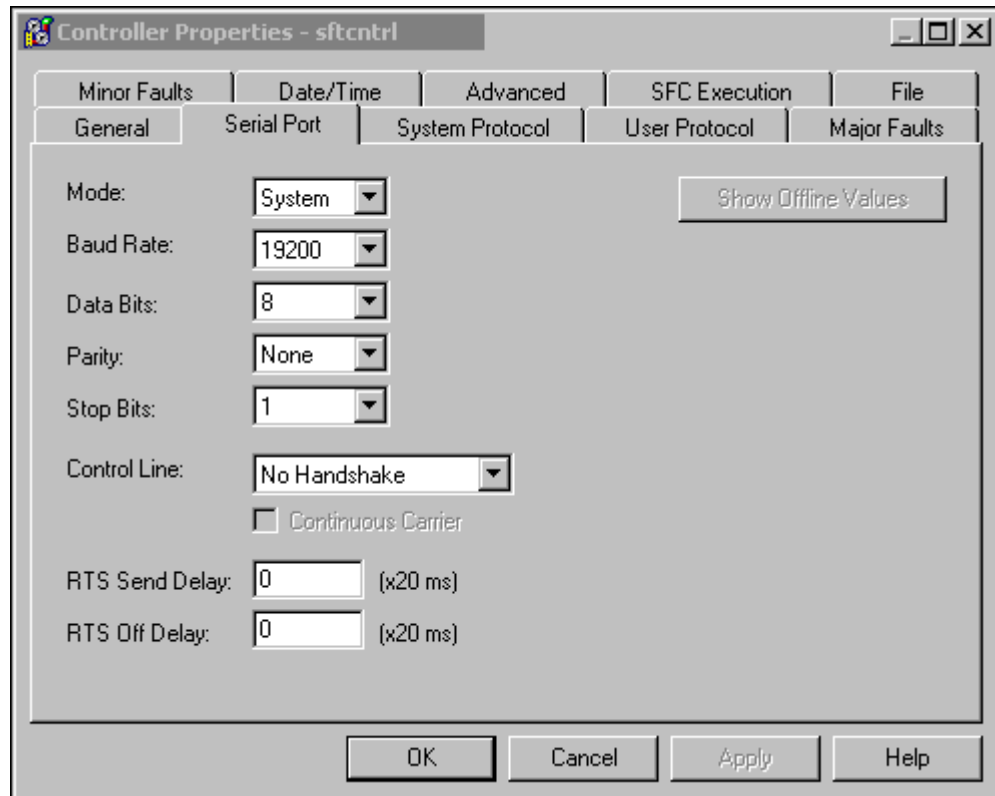
### Select a processor to change to

Choose the controller you wish to change to from the pull-down menu. The list of available controllers includes all controller types except the current processor itself. Once a new controller is selected and the OK button is clicked, a warning message displays. There are two different warning screens depending upon whether the change is within the same platform or to another platform. These warnings inform you that certain settings and properties are changed or deleted based upon the type of controller selected.



**Figure 2.4 Warning Message for Change to a Different Platform**

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



**Figure 2.5 Controller Properties Serial Port Tab**

### 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, 19200 (default), and 38,400.

### 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.

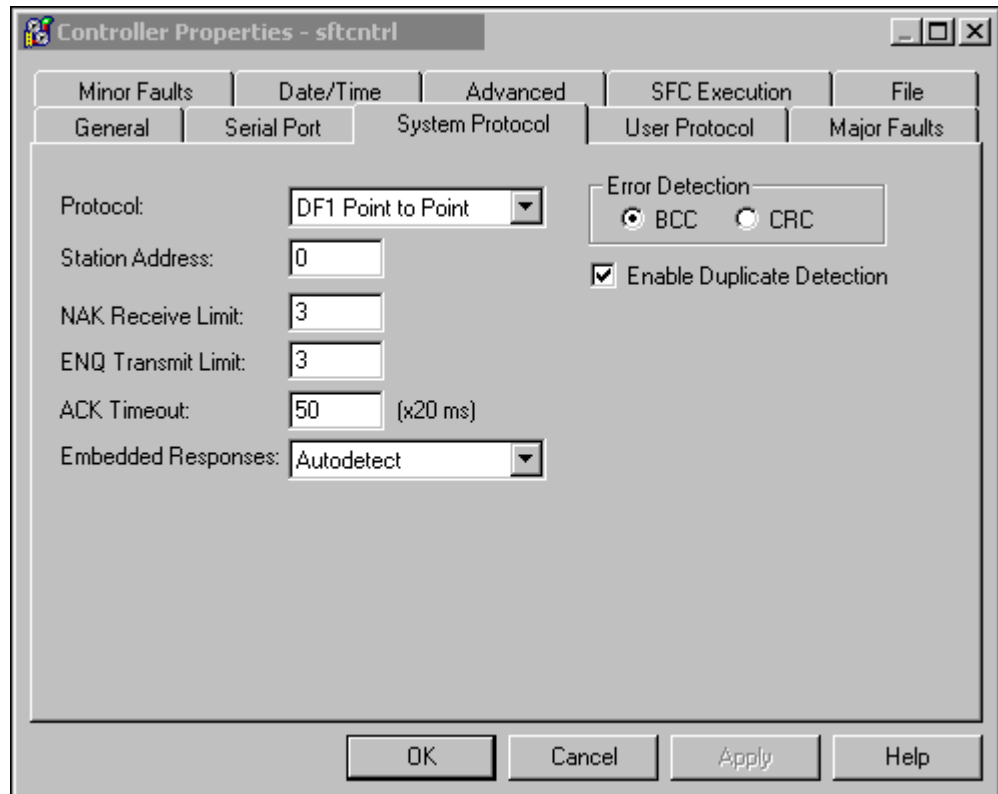
## RTS Send Delay

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.



**Figure 2.6 Controller Properties System Protocol Tab**

### **IMPORTANT**

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**

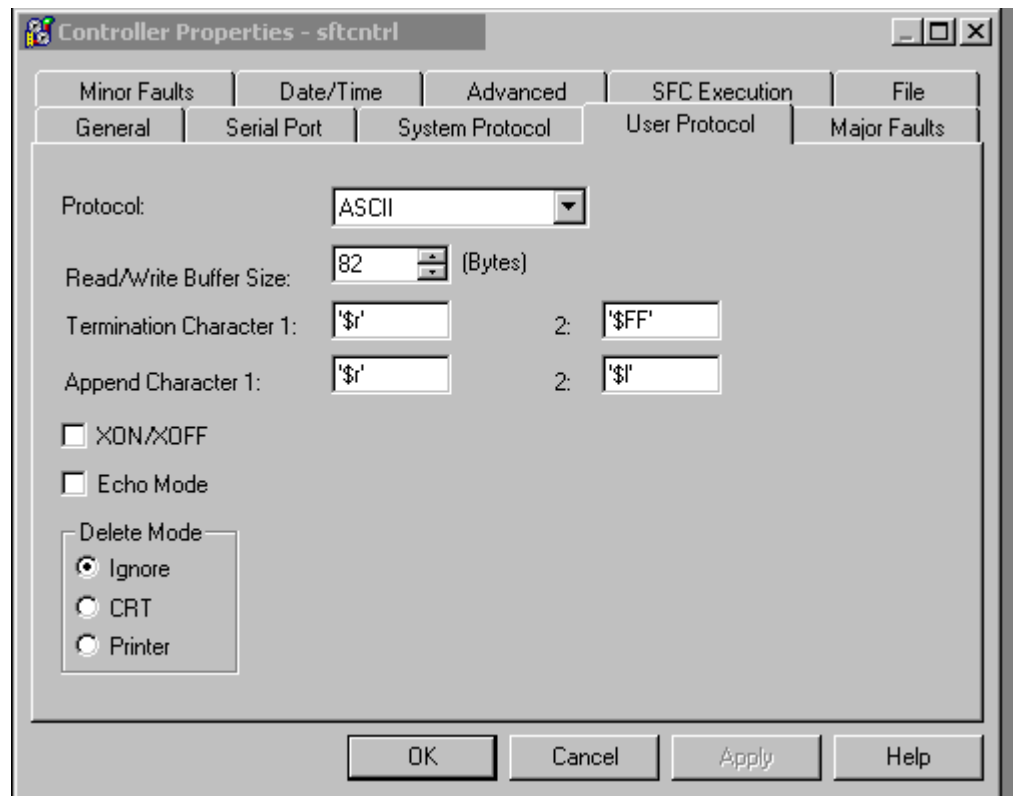
#### **Max Station Address**

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.



**Figure 2.7 Controller Properties User 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.

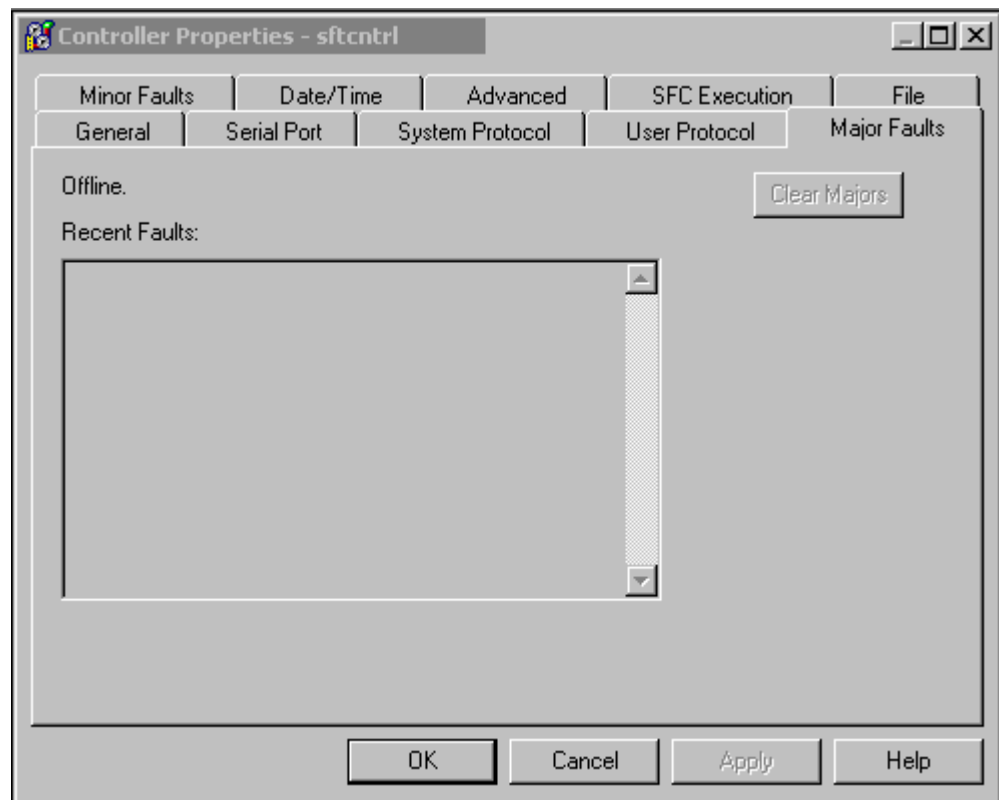
This option is disabled when the Control Line option is configured for Half Duplex.

### 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.



**Figure 2.8 Controller Properties Major Faults Tab**

### Number of Major Faults Since Last Cleared

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

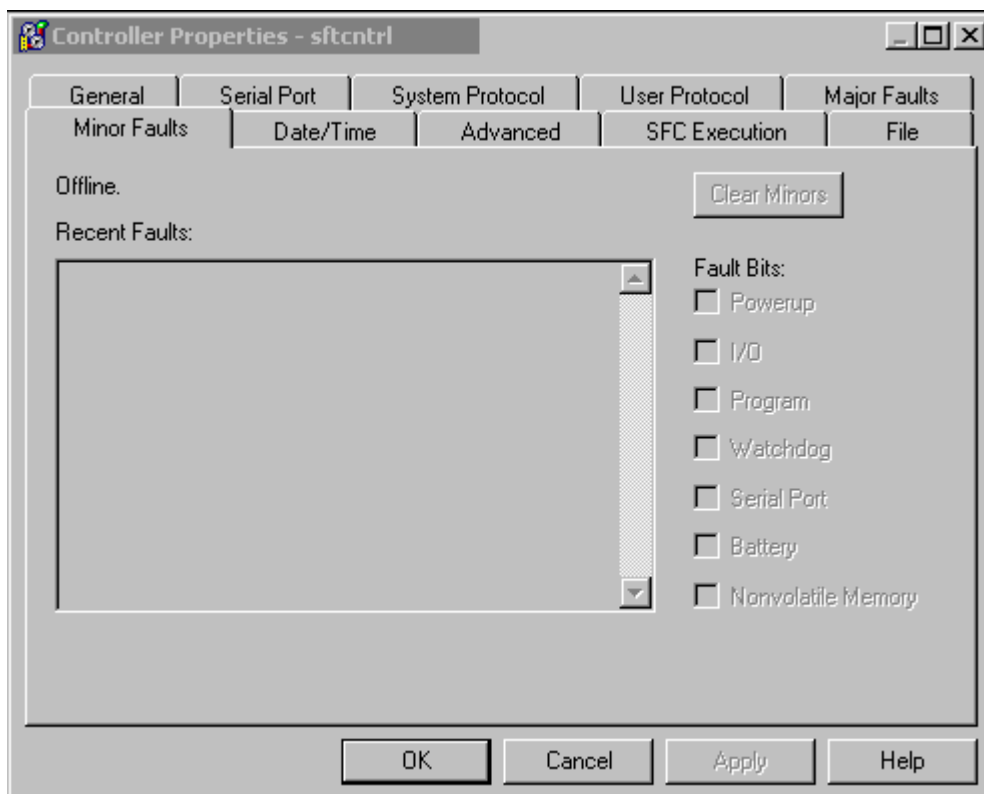
### Recent Faults

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.



**Figure 2.9 Controller Properties Minor Fault Tab**

### 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

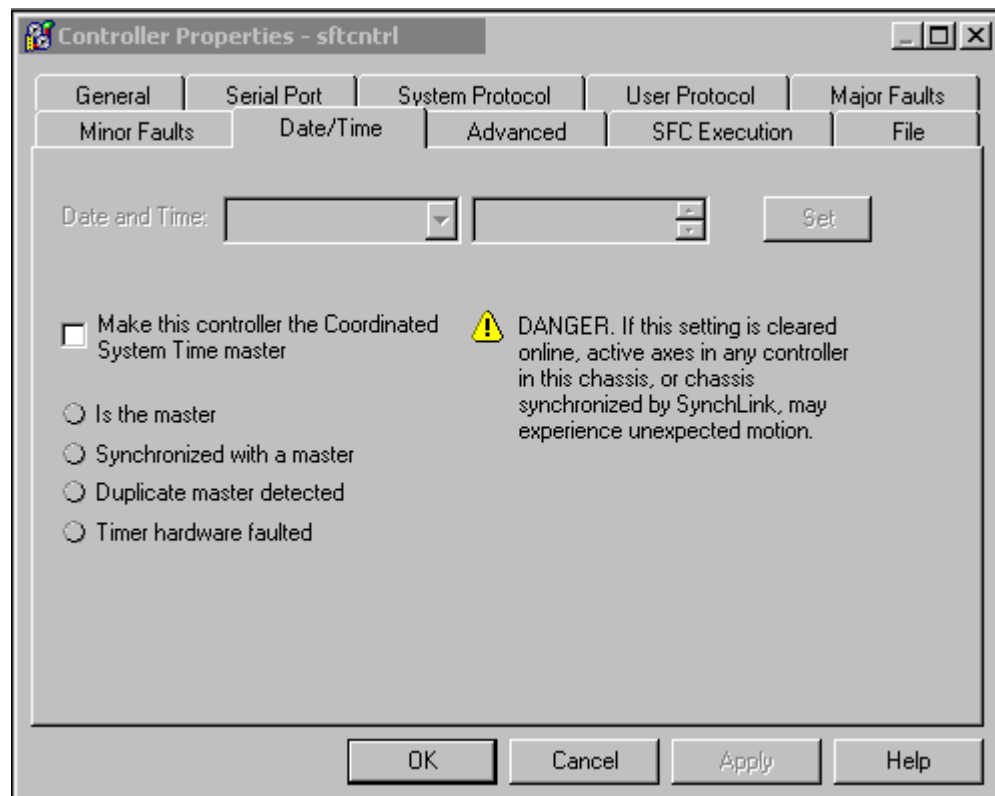
Click on this button to clear the Minor Fault log.

## 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.



**Figure 2.10 Controller Properties Date/Time Tab**

## 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.

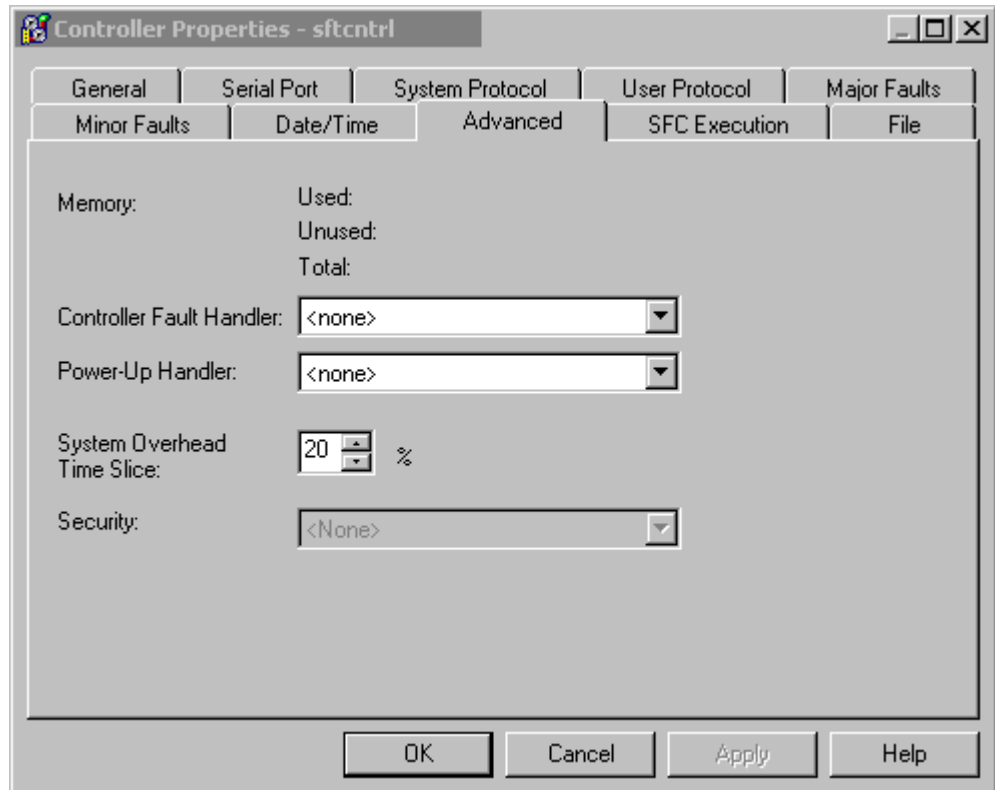
## Status

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.



**Figure 2.11 Controller Properties Advanced Tab**

### 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.

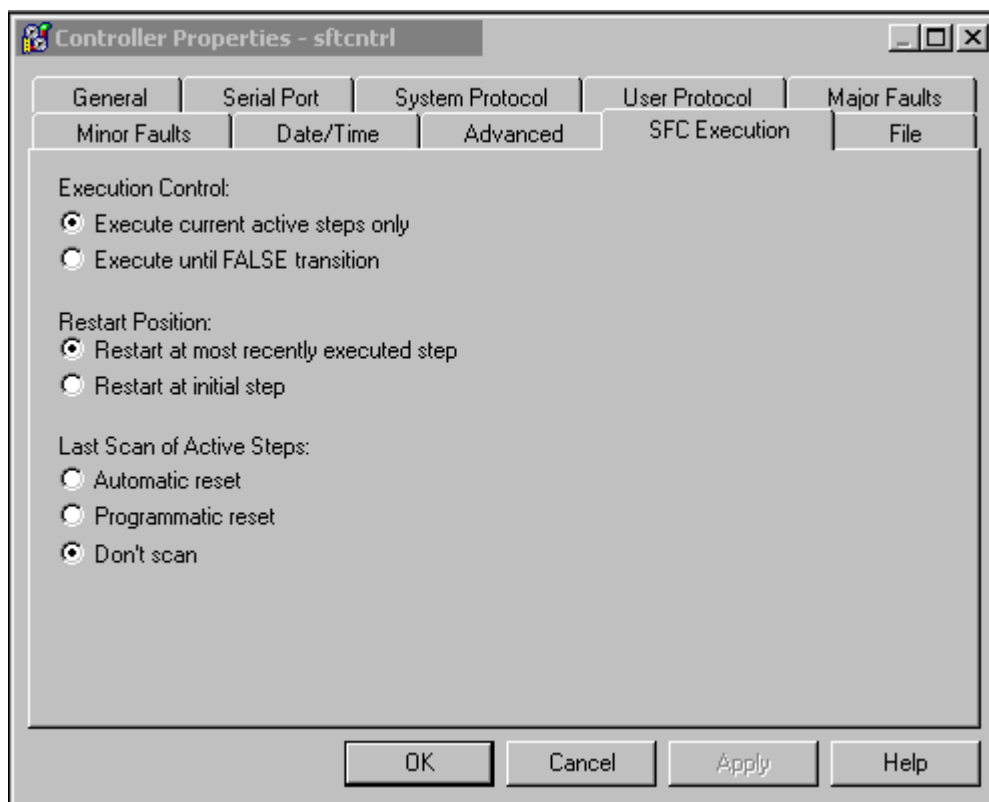
## Power-Up Handler

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.

**SFC Execution Tab** The SFC Execution tab sets the configuration that affects the execution of the SFC (Sequential Function Chart). These settings apply to the entire Controller and therefore affect all SFCs within the Controller.



**Figure 2.12 Controller Properties SFC Execution Tab**

## Execution Control

This determines the execution model for the SFC. Your options are:

- **Execute current active steps only** – Execution control is returned to the controller after processing the active steps, even if the Transitions following the active steps are True.
- **Execute until False transition** – The controller continually processes Steps and Transitions, in a single scan, until a False Transition is found. It then returns to the Controller operating system.

## Restart Position

Your selection here indicates at what point the SFC starts after it has completed executing and been restarted.

- **Restart at most recently executed step** – Restarts at the step where it left off.
- **Restart at initial step** – The SFC re-initializes and starts at the Initial Step.

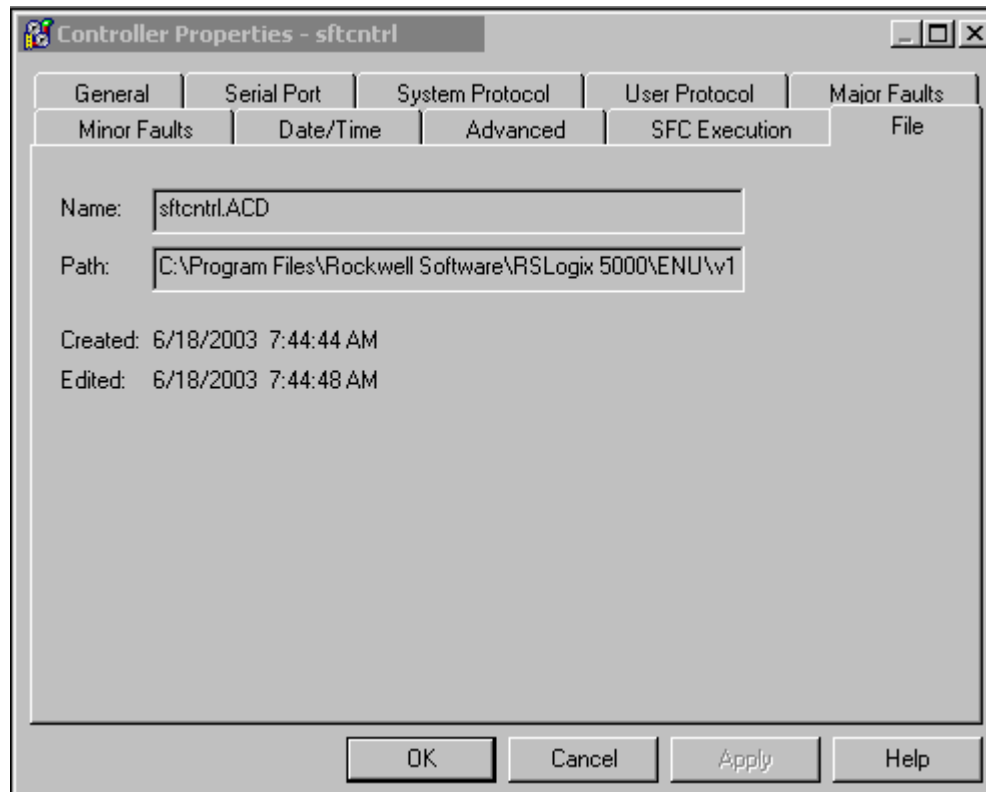
## Last Scan of Active Steps

This determines how the controller performs a final scan of Action logic when the associated Step is deactivated.

- **Automatic reset** – All non-retentive timers and outputs associated with the Step's non-stored Actions are reset when the step is deactivated.
- **Programmatic reset** – All non-retentive timers and outputs associated with the Step's non-stored Actions must be reset manually.
- **Don't scan** – No scan is done on the Action logic when the associated Step is deactivated.

**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.



**Figure 2.13 Controller Properties File Tab**

### **Name**

The name of the project file

### **Path**

The drive and directory of the project file.

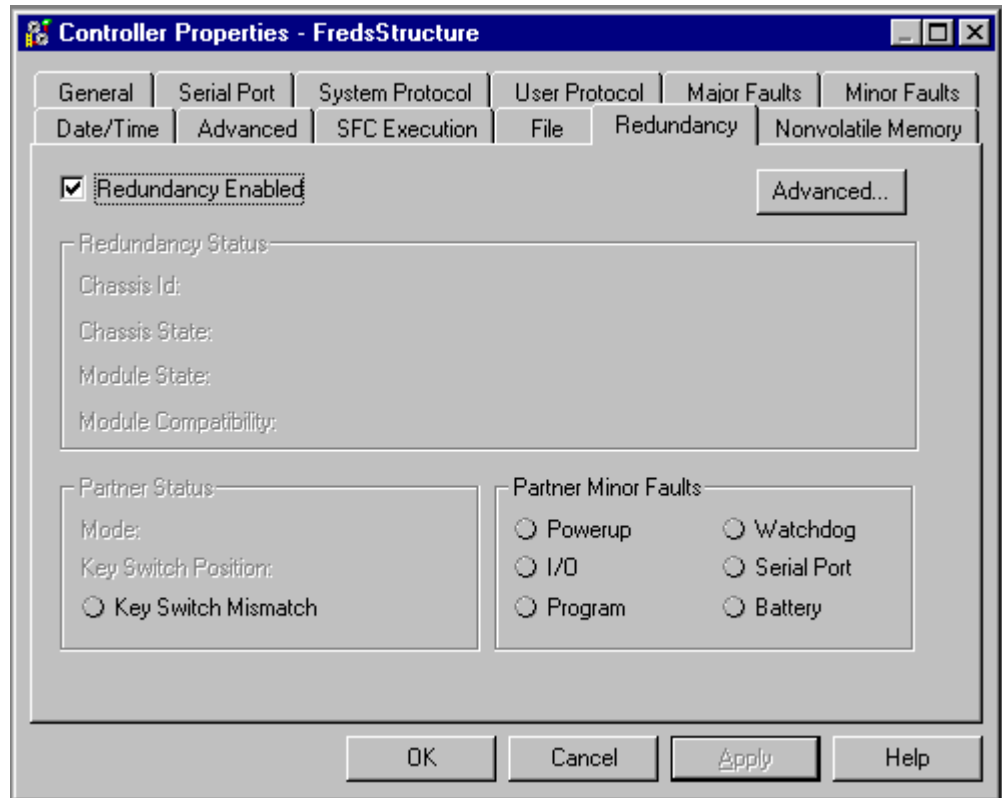
### **Created**

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.

**Redundancy Tab** The Redundancy Tab is only present if the specified processor type and version supports the Redundancy feature. This tab supports the configuration for redundancy.



**Figure 2.14 Controller Properties Redundancy Tab**

### Redundancy Enabled

This checkbox lets you select whether to enable the redundancy feature or not. It can only be selected when the Controller is offline. When it is on-line it provides a valid indication of the redundancy enabled selection.

### Redundancy Status

This section shows the redundancy states.

**Chassis ID** – Displays the chassis identification.

**Chassis State** – This shows the redundancy state of the entire chassis. The possible states include:

- 0 – Unsupported – also displays when the system is offline.
- 1 – Undetermined
- 2 – Primary with Synchronized Secondary
- 3 – Primary with Disqualified Secondary

- 4 – Primary with no partner
- 8 – Synchronized Secondary
- 9 – Disqualified Secondary with partner
- 10 – Disqualified Secondary with no partner

**Module State** – Indicates the redundancy state of the controller. If the controller does not have the redundancy feature or if it is offline, this field is empty and the controls are disabled. The possible states include:

- 0 – Unsupported – also displays when the system is offline.
- 1 – Undetermined
- 2 – Primary with Synchronized Secondary
- 3 – Primary with Disqualified Secondary
- 4 – Primary with no partner
- 6 – Synchronizing Primary
- 7 – Synchronizing Secondary
- 8 – Synchronized Secondary
- 9 – Disqualified Secondary with partner
- 10 – Disqualified Secondary with no partner

**Module Compatibility** – Shows the modules compatibility with its corresponding module in the partner chassis. If the controller does not have the redundancy feature or if it is offline, this field is empty and the controls are disabled. The possible states include:

- 0 – Unsupported – also displays when the system is offline.
- 1 – No compatible Partner
- 2 – Fully compatible Partner

## Partner Status

This section shows information on the status of the Partner module.

**Mode** – Shows the current state of the partner module's mode. If the controller does not have the redundancy feature or if it is offline, this field is empty and the controls are disabled. Valid modes include:

- Faulted
- Run
- Program
- Test
- Unknown – Displayed for any mode that is not one of those listed above. It also displays if the Module Compatibility is not "Fully Compatible Partner" or "Standby Compatible Partner". Also displays when the system is offline.

**Key Switch Position** – Shows the current status of the partner module's keyswitch position. If the controller does not have the redundancy feature or if it is offline, this field is empty and the controls are disabled. Valid modes include:

- Run
- Remote
- Program
- Unknown – Displays when the Module Compatibility is not "Fully Compatible Partner" or "Standby Compatible Partner". Also displays when the system is offline.

**Key Switch Mismatch** – Is activated when there is a mismatch between the key switch position between the primary and partner modules.

## Partner Minor Faults

This section shows the minor faults that are set in the partner controller. These controls are read-only. If the controller does not have the redundancy feature or if it is offline, the radio button is greyed and the text label is disabled. Minor partner faults include:

- Powerup
- I/O
- Program
- Watchdog
- Serial Port
- Battery

## Advanced Button

The Advanced Button displays configuration parameters for retaining test edits when switched to a secondary system and lets you set the percentage of memory that is reserved for the data table.

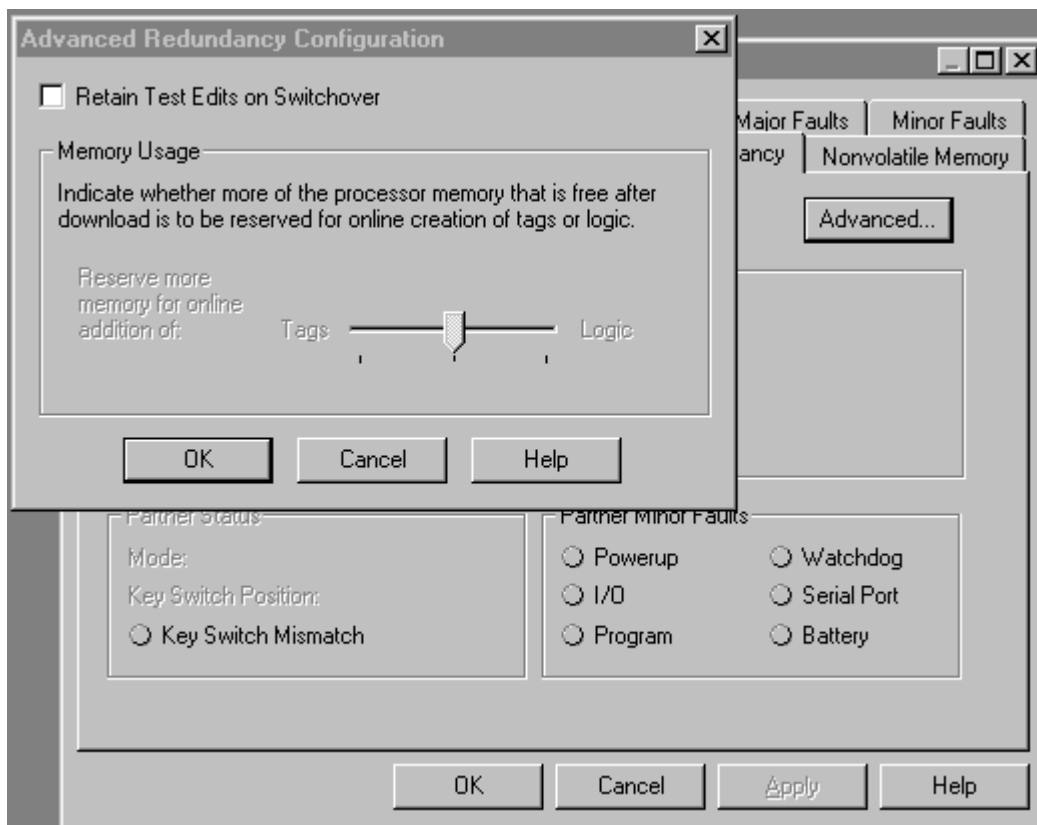


Figure 2.15 Advanced Button from Redundancy Tab

## Retain Test Edits on Switchover

Select the checkbox to allow temporary execution of online edits to be maintained or canceled when a switchover to a secondary system occurs. When enabled, this option prevents the Logix controller from automatically switching back to the unedited version of your program if the primary chassis fails while testing on-line edits.

## Memory Usage

Lets you modify the percentage of memory that is to be reserved in the data table area. This lets you create tags in RUN mode on a synchronized primary system.

**Nonvolatile Memory Tab** The Nonvolatile Memory tab of the Controller Properties dialog serves as the starting point from which you can perform nonvolatile memory operations.

Click on the Load/Store button to access the Nonvolatile Memory Load/Store dialog, from which you can perform the actual operations.

The Nonvolatile Memory tab also provides you with status information that indicates any conditions that might prevent you from loading or storing. Possible status messages include:

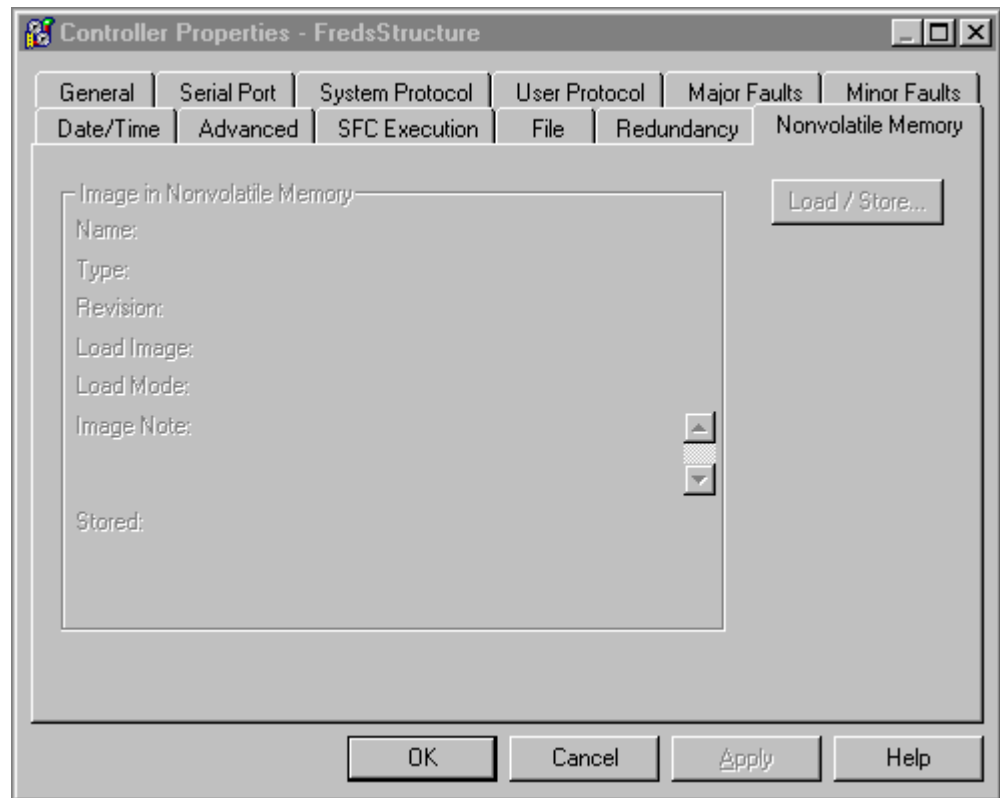
- Nonvolatile memory not present.
- Nonvolatile memory not supported in redundant systems.
- Controller being edited by another user.
- No controller image or stored controller image.
- Controller not in Program mode.
- Stored image revision in nonvolatile memory does not match controller revision.

The Controller Properties dialog displays controller configuration information for the open project and, when online, for the attached controller.

The Nonvolatile Memory tab provides you with access to nonvolatile memory operations.

**Note:** This tab is available only when the selected controller supports nonvolatile memory.





**Figure 2.16 Controller Properties Nonvolatile Memory Tab**

### **Name**

The name of the stored controller image that resides in nonvolatile memory.

### **Type**

The controller type for the image stored in nonvolatile memory. This controller type can be any type that supports nonvolatile memory.

### **Revision**

The firmware revision of the controller when the image in nonvolatile memory was stored. In order for the image in nonvolatile memory to be loaded back into controller memory, the firmware revision of that image must match the revision of the controller.

## Load Image

The condition under which the image stored in nonvolatile memory is loaded back to controller memory. Available conditions include:

- On Corrupt Memory – this will cause a load whenever there is no project in the controller and you turn on or cycle power on the chassis. If you are using a battery the controller, selecting this option performs a load only if the battery has failed to maintain the project during a loss of power.
- On Power Up – this will cause a load whenever you turn on or cycle power on the chassis. If you are using a battery on the controller, selecting this option performs a load even if the battery has maintained the project during loss of power.
- User Initiated – choose this option if you want to load only through RSLogix 5000 software.

Regardless of the currently-selected load option, you can always manually initiate a load as well.

## Load Mode

The mode the controller enters upon loading from nonvolatile memory. Choose from:

- Program (Remote Only)
- Run (Remote Only)

### TIP



The Load Mode only applies if the controller keyswitch is in the remote position when loading. If the controller keyswitch is in Run or Program the controller will be returned to that mode once the load is complete. The controller keyswitch always has priority over the Load Mode.

The Load Mode only applies if the Load Image selection is On Corrupt Memory or On Power Up. If the Load Image is User Initiated, the Load Mode is disabled.

## Image Note

Descriptive information that you entered at the time the image was stored in nonvolatile memory. You may enter up to 128 text characters.

## Stored

The workstation date and time when the image was stored in nonvolatile memory.

## Load/Store

Click on this button to access the Load/Store dialog.

This button is disabled when:

- Nonvolatile memory is not present in the controller.
- The controller is in Run mode.
- Another user has locked the controller.
- Redundancy is enabled for the controller.
- The controller is offline.

If the Load/Store button is disabled, the status bar indicates the reason. Note that the status bar displays only one status at a time, and you must work through each of them to enable the Load/Store button.



## Adding and Configuring Your 1784-PM02AE Motion Module

This chapter describes how to add, configure, and edit your 1784-PM02AE motion module for use in your motion control application.

### Adding the 1784-PM02AE 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.

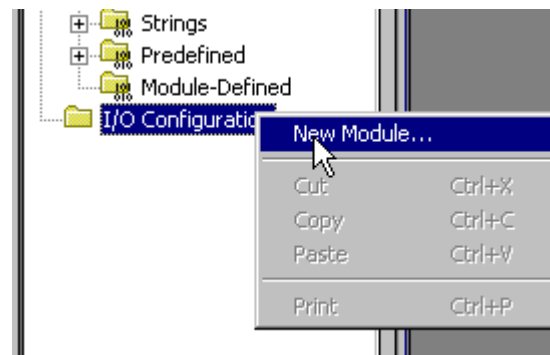


Figure 3.1 Selecting New Module from the Controller Organizer

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

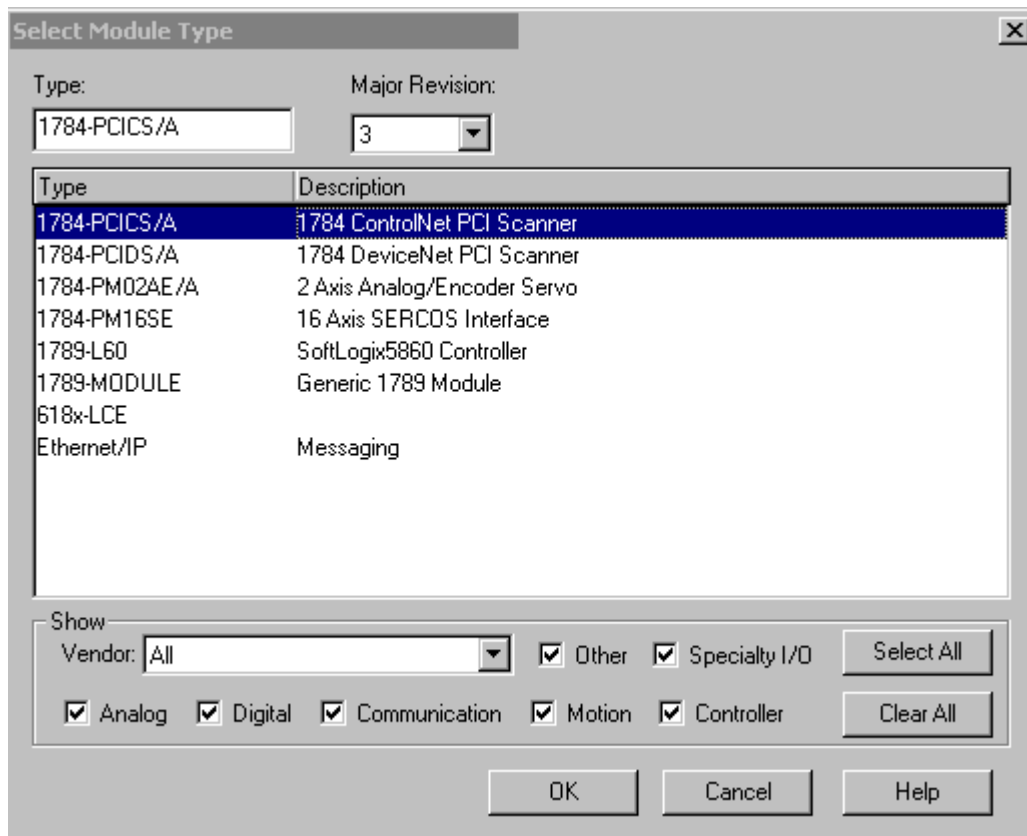
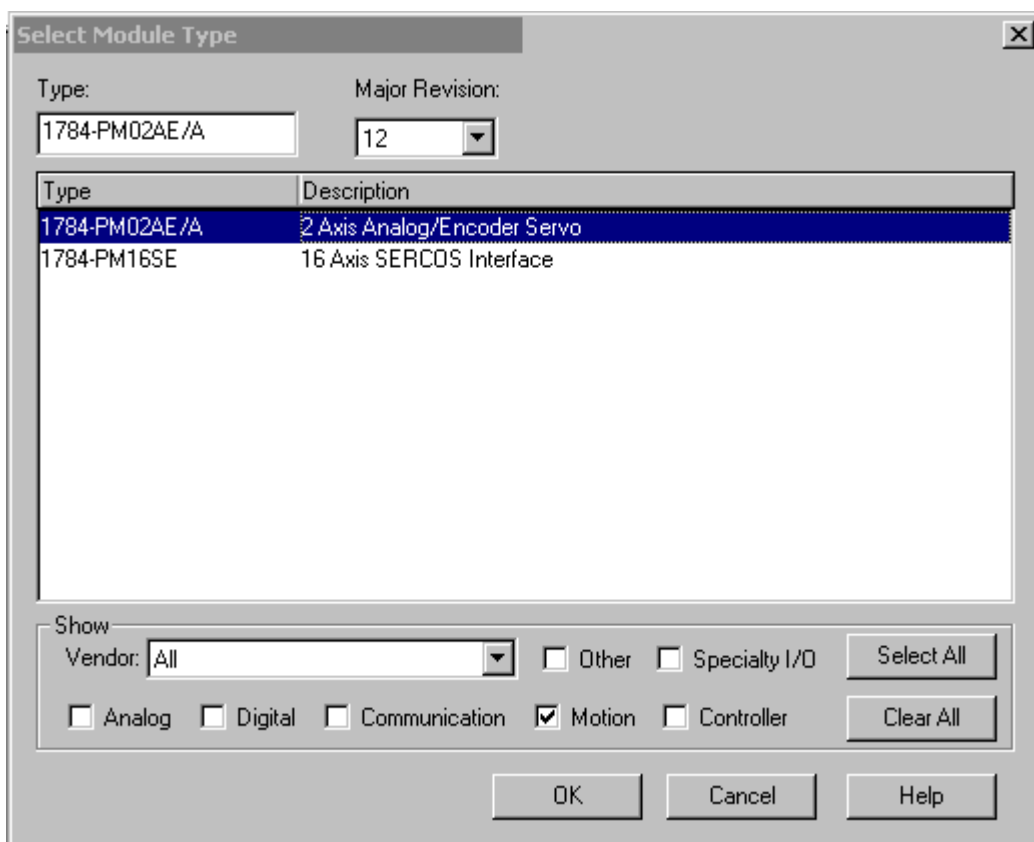


Figure 3.2 Select Module Type Screen Fully Loaded

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



**Figure 3.3 Select Module Type Screen with Motion Options - M02AE Highlighted**

**New Module** Use this dialog to select and create a new module. Highlight the 1784-PM02AE 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 want 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.

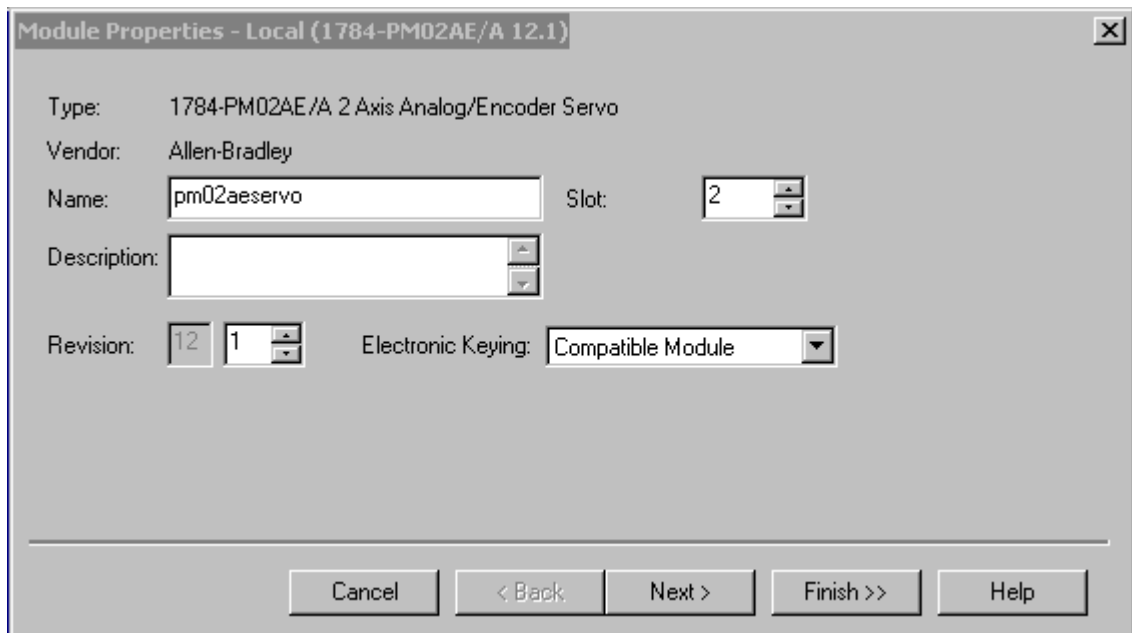
## Clear All

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

- In the *Type* field, select **1784-PM02AE 2 Axis Analog/Encoder Servo**.



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

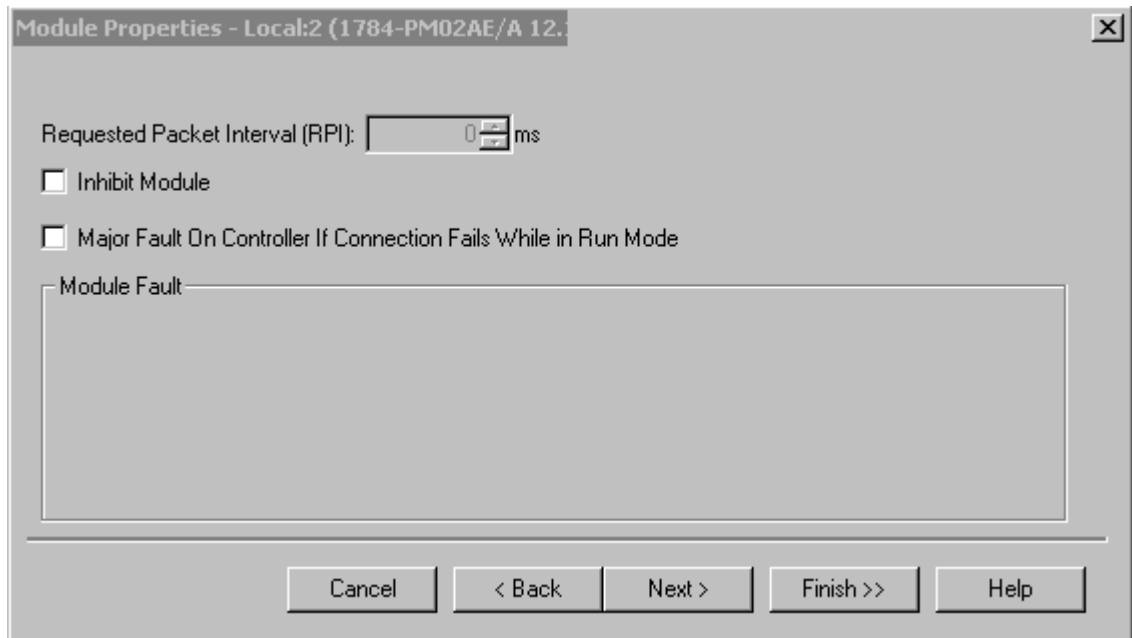


**Figure 3.4 Module Properties Dialog Wizard - Naming the Module**

6. Make entries in the following fields.

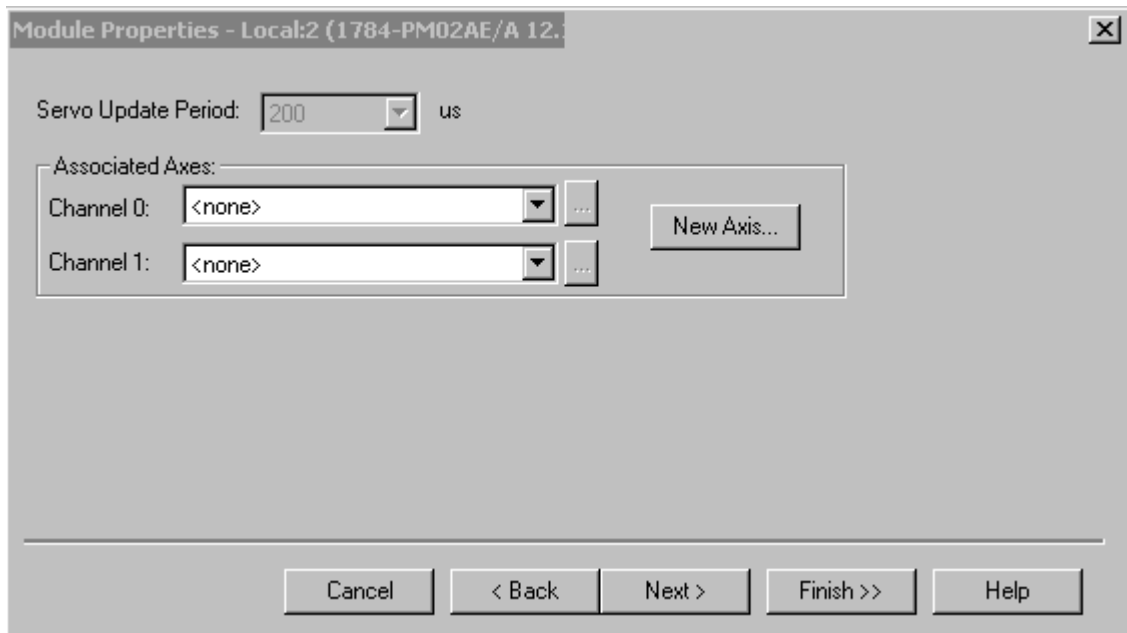
Field	Entry	
Name	Type a name for the servo module. The name can: <ul style="list-style-type: none"> <li>• have a maximum of 40 characters</li> <li>• contain letters, numbers and underscores (_).</li> </ul>	
Slot	Enter the number of the chassis slot that contains your module.	
Description	Type a description for your motion module. This field is optional.	
Electronic keying	Select the electronic keying level.	
	<b>To</b>	<b>Select</b>
	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.



**Figure 3.5 Module Properties Wizard - Fault Handling**

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.



**Figure 3.6 Module Properties Wizard - Servo Update/Associated Axis**

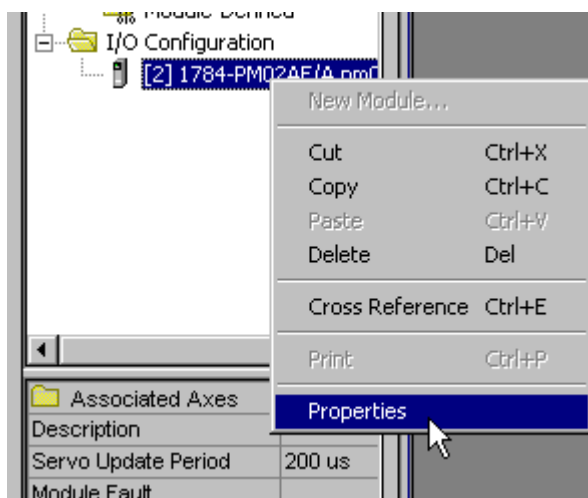
9. This screen lets you associate an axis with the module. Make the appropriate choices for your application. At this point, the rest of the screens are informational only and it would be best to press the Finish button to create the module.

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

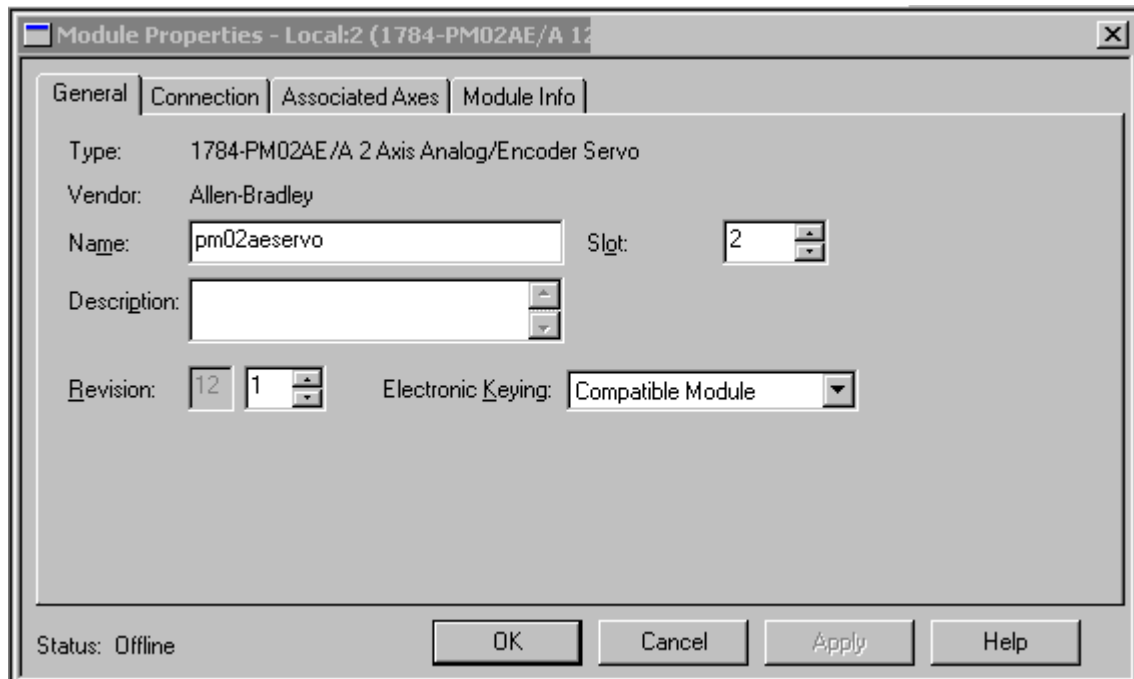
The following section provides explanations of the Motion Module Properties screens. Use these screens to edit the properties of the module when changes need to be made. You can access the Module Properties screen by highlighting the motion module and right clicking the mouse.

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



**Figure 3.7 Controller Organizer - Module Properties Pop up**

This accesses the Module Properties screen. The screen is tabbed to expedite movement to the required dialog.



**Figure 3.8 Module Properties - General Tab**

**General Tab** Use this tab to create/view module properties for 1784-PM02AE 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:

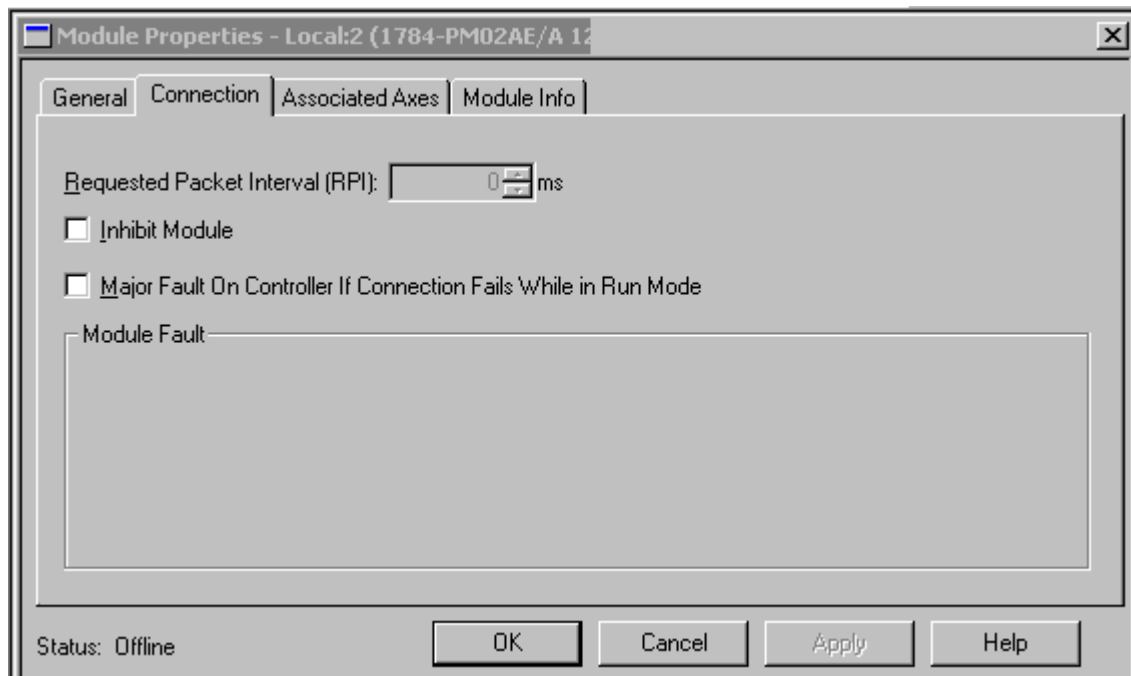
- Exact Match - all of the parameters must match or RSLogix rejects the inserted module.
  - Vendor
  - Product Type
  - Catalog Number
  - Major Revision
  - Minor Revision
- Compatible Module
  - the Module Types, Catalog Number, and Major Revision must match
  - the Minor Revision of the physical 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.



**Figure 3.9 Module Properties - Connection Tab**

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 modules.

### 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	<p>stop communication to a module</p> <p>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.</p> <p>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'.</p> <p>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.</p> <p>If you are a listener (have chosen a "Listen Only" Communications Format), you can not re-configure the module.</p> <p>If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.</p>

### 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 1784-PM02AE motion modules by:

- setting the selected 1784-PM02AE motion module's Servo Update Period
- associating axis tags, of the type AXIS\_SERVO, with channels 0 and 1

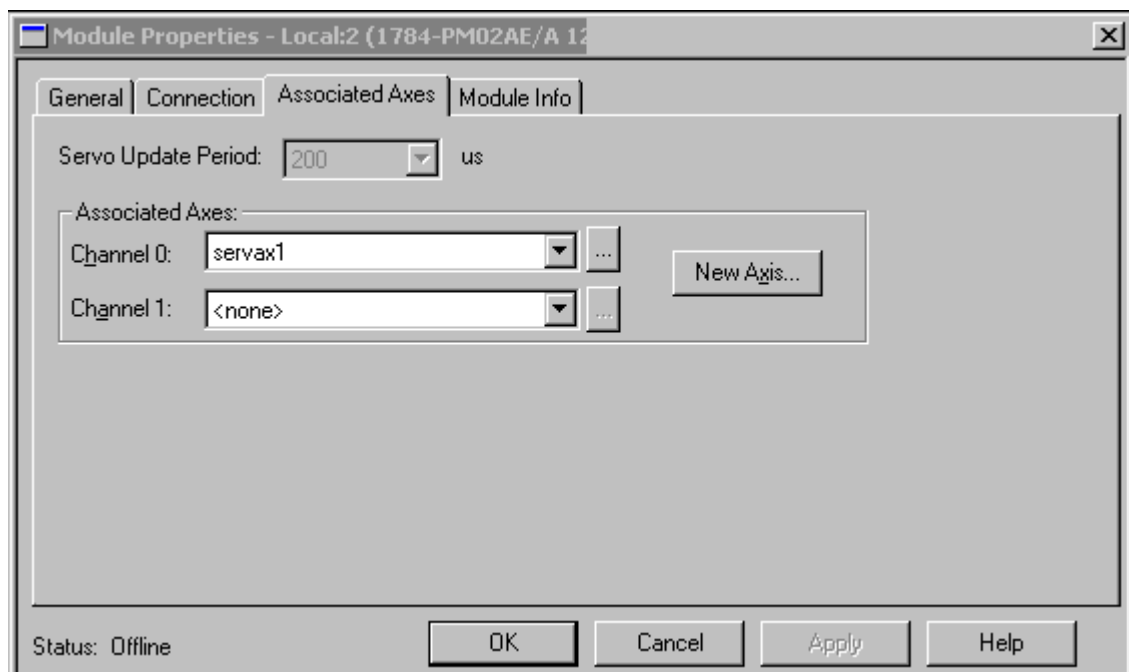


Figure 3.10 Module Properties - Associated Axis Tab



## Servo Update Period

Selects the periodic rate at which the 1784-PM02AE module closes the servo loop for the axis, in microseconds ( $\mu\text{s}$ ).

## 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. See the chapter entitled *Naming & Configuring Your Motion Axis* in this manual for more information on creating axes with RSLogix 5000.

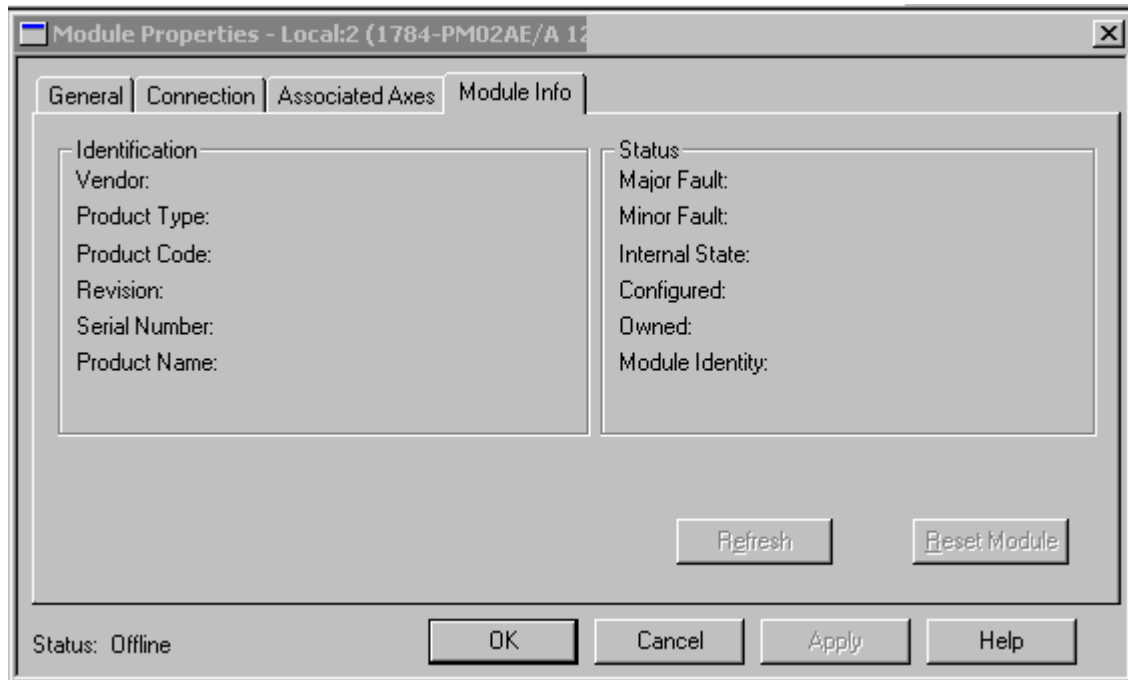
**Module Info Tab** The Module Info tab contains information about the selected module, however, you can click on:

- 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.



**Figure 3.11 Module Properties - Module Info Tab**

### 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: <ul style="list-style-type: none"> <li>• Vendor</li> <li>• Module Type (the combination of Product Type and Product Code for a particular Vendor)</li> <li>• Major Revision</li> </ul>
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.

---

<b>IMPORTANT</b>	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.

## Assigning Additional Motion Modules

You can assign additional modules by repeating the preceding sections. You can assign up to 16 1784-PM02AE modules to each Logix5000 controller. Each module uses a maximum of two axes.

## Configuring the 1784-PM16SE Card

### Adding the 1784-PM16SE

This chapter reviews the necessary steps for configuring the 1784-PM16SE motion card. Much of this information is the same as for adding and configuring the 1784-PM02AE as discussed in the previous chapter.

To configure a 1784-PM16SE motion card:

1. In the Controller Organizer, right mouse click on I/O Configuration.

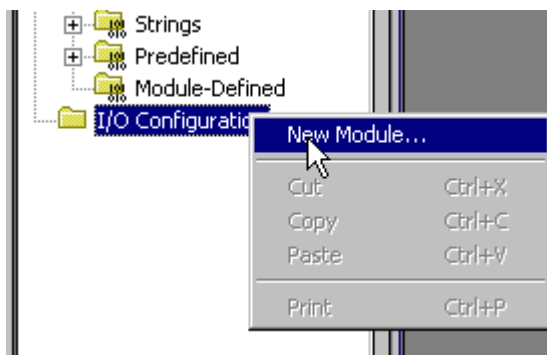


Figure 4.1 Controller Organizer | I/O Configuration | New Module

2. Or in the File menu, select New Component then Module...

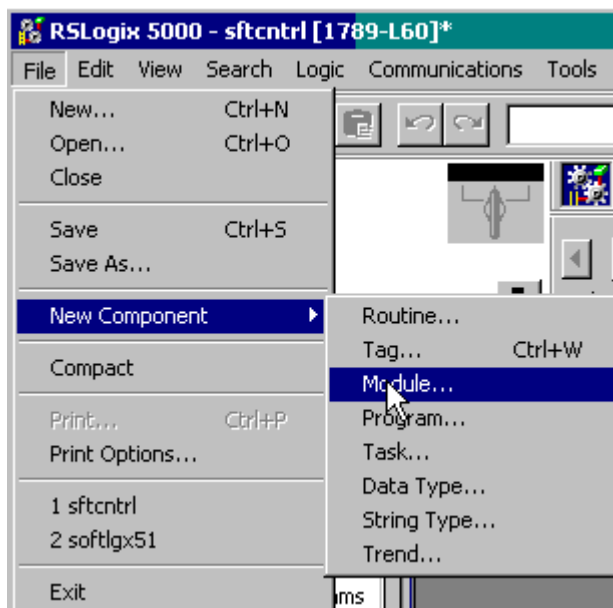
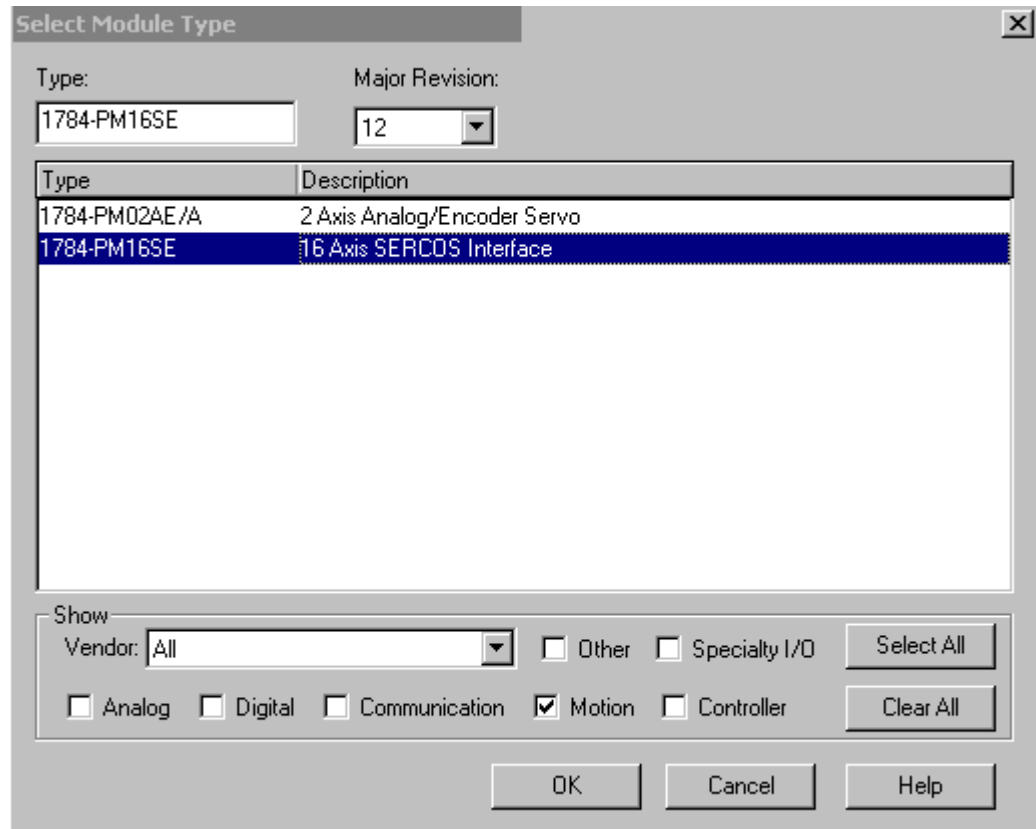


Figure 4.2 File Menu | New Component | Module Selected

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



**Figure 4.3 Select Module Type Screen with Motion Options - 1784-PM16SE Selected**

4. Select 1784-PM16SE.
5. Press the OK button to close the Select Module Type dialog.

6. The Create Module Wizard opens.

The screenshot shows a dialog box titled "Module Properties - Local (1784-PM16SE 12.1)". The fields are as follows:

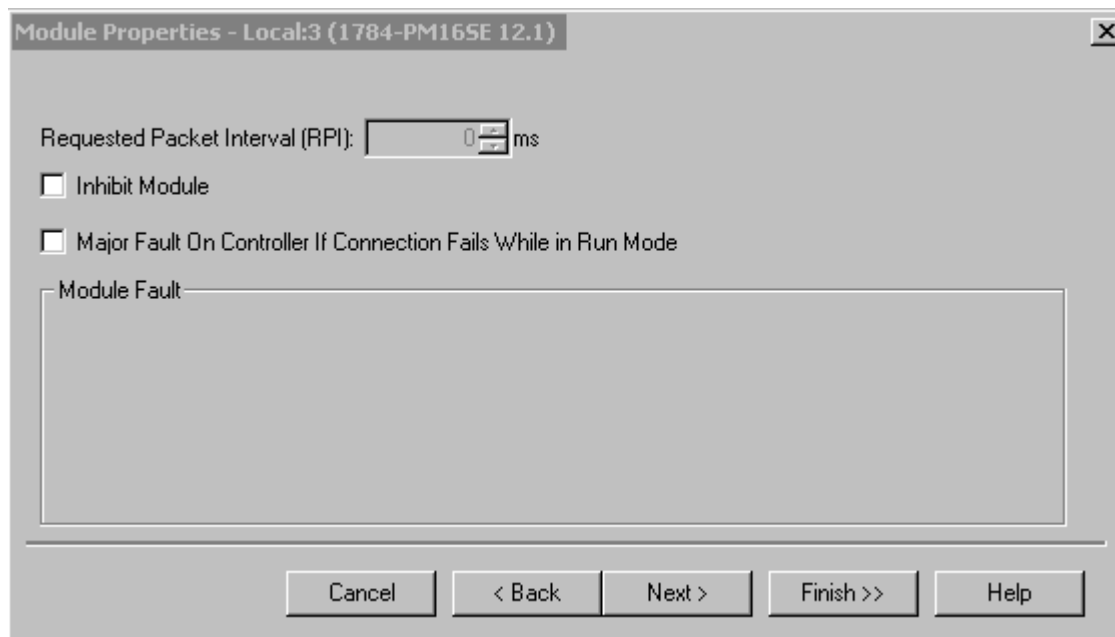
- Type: 1784-PM16SE 16 Axis SERCOS Interface
- Vendor: Allen-Bradley
- Name: pm16sercos
- Slot: 3
- Description: (empty)
- Revision: 12, 1
- Electronic Keying: Compatible Module

Buttons at the bottom: Cancel, < Back, Next >, Finish >>, Help.

**Figure 4.4 Module Properties Wizard Dialog - Name the Module**

7. Name is the only required field that must be entered to create the 1784-PM16SE card. It must conform to the IEC 1131-3 standard. You can also enter a description for the card, select the minor revision number of your card, and select the method for Electronic Keying. Fill in the at least the required Name field and click the Next> button to advance to the next wizard screen to enter Connection information or click on the Finish>> button to create the card. You can then go to the Module Properties screen to edit any values. (See the section titled *SERCOS interface Motion Card Overview* in this chapter for more information on the fields in these screens.)

## 8. The Connection Screen Wizard displays.



**Figure 4.5 Module Properties Wizard Dialog - Connection Screen**

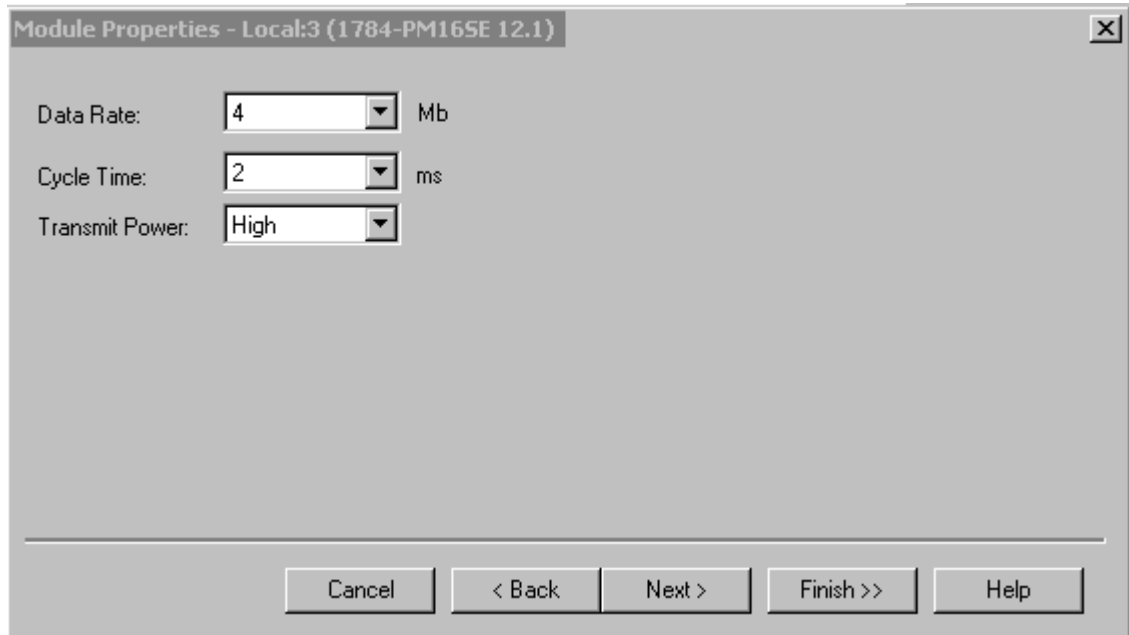
9. On this screen there are no required fields but you can enter how you want to handle connection faults. The Requested Packet Interval (RPI) field does not pertain to the SERCOS interface cards and is greyed out.

Inhibit Module defaults to Unchecked. Click on the check box to inhibit the module.

**Major Fault on Controller ...** check box defaults to uncheck. Click on the box if you want to check this option. Click on Next> to advance the SERCOS interface Create Wizard screen.



10. The SERCOS interface screen displays.



**Figure 4.6 Module Properties Wizard Dialog - SERCOS interface Screen**

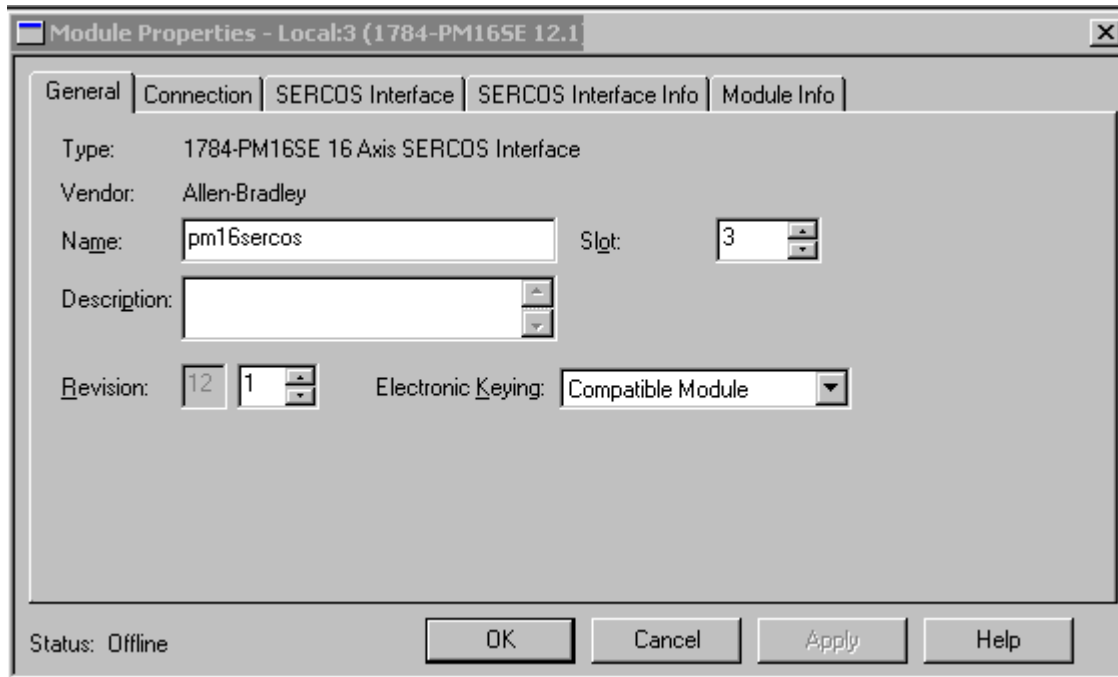
11. On this screen you can enter the Data Rate, SERCOS ring Cycle time, and the transmit power for the SERCOS ring.

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.

12. The 1784-PM16SE motion card appears in the I/O Configuration branch of the Controller Organizer. It can now be put into use or edited as you require.

## SERCOS interface Motion Card Overview

The 1784-PM16SE SERCOS interface motion card has been added. To edit the 1784-PM16SE card properties, go to the I/O Configuration organizer and right click on the 1784-PM16SE card and select Properties from the drop down menu. The tabbed Module Properties screen displays.



**Figure 4.7 Module Properties - General Tab**

The Module Properties screen has the following tabs:

- The General tab references the 1784-PM16SE motion card.
- The Connection tab references the connection of the card to the controller.
- The SERCOS Interface tab is for configuring SERCOS communication settings for the 1784-PM16SE motion card.
- 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 1784-PM16SE Card Properties

**General Tab** Use this tab to create/view module properties for the 1784-PM16SE motion card.

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 card. The name must be IEC 1131-3 compliant. An error message is displayed if you enter an invalid character or a duplicate name. If you exceed the maximum length, the software ignores the extra characters.

### Description

Enter a description for the card 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.

## Slot

Enter the slot number where the card 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). Only available slot numbers are listed by the spin button. However, you can edit the slot number manually. 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 card. The minor revision is used to indicate the firmware revision.

Select the minor revision number of your card.

## Electronic Keying

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

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

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

**Disable Keying** – Controller does not employ keying at all.

---

**ATTENTION**

Changing the Electronic Keying selection may cause the connection to the card 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.

---

**Status** – This is a Read Only field that displays the Controller's current opinion of the card.

*Standby* – A transient state that occurs when shutting down.

*Faulted* – It is unable to communicate with the card. When Faulted is displayed, check the Connection Tab for the fault listing.

*Validating* – A transient state that occurs prior to connecting to the card.

*Connecting* – The state while the connection(s) to the cards are established.

*Running* – The card is communicating and everything is working as expected.

*Shutting Down* – The connections are in the process of closing.

*Inhibited* – The card is prevented from connecting to the controller.

*Waiting* – A connection to this card has not been made due to one of the following reasons.

- Its parent has not yet made a connection to it.
- Its parent is inhibited.
- Its parent is faulted.

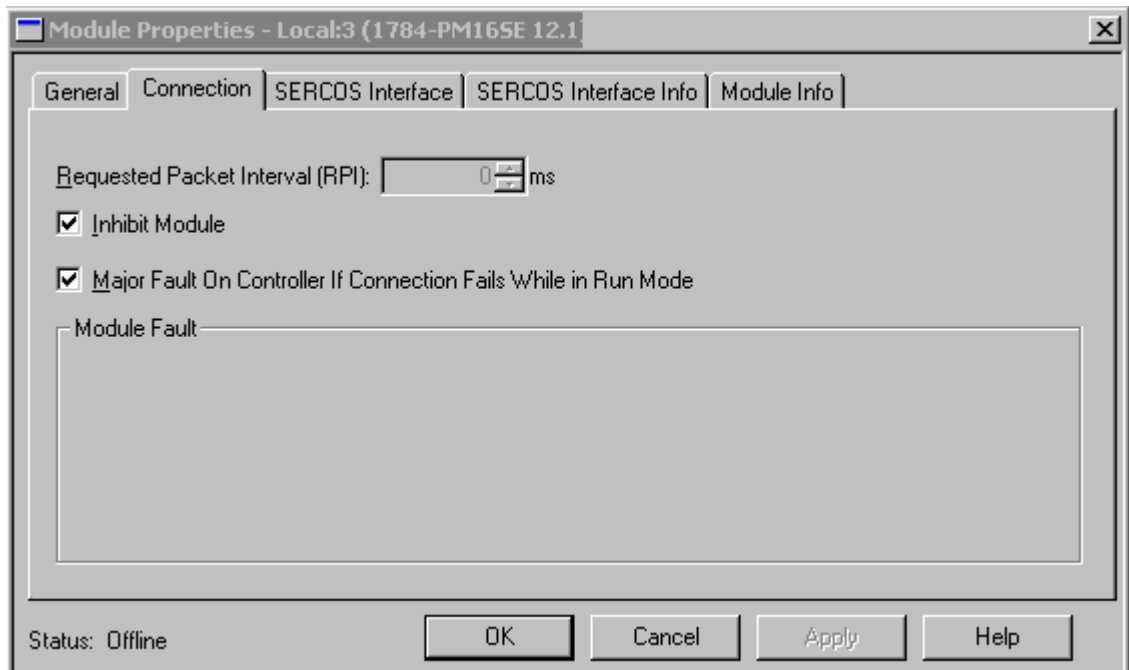
*Offline* – The card is not currently online.

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

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

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

**Connection Tab** The Connection Tab reflects controller to card behavior. This is where you choose to inhibit the card, configure the controller so loss of the connection to this card causes a major fault, and view card faults when online.



**Figure 4.8 Module Properties - Connection Tab**

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

### Requested Packet Interval

This does not apply to motion cards.

### Inhibit Module checkbox

Check/Uncheck this box to inhibit/uninhibit your connection to the card. Inhibiting the card causes the connection to the card to be broken. When a card is inhibited all of the associated axes are not used in the configuration process. The system ignores them as if they were not there and allows configuration and operation of any axis associated to other cards in the group.

#### TIP

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



**ATTENTION**

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



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

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

**Major Fault on Controller if Connection Fails checkbox**

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

**Module Fault**

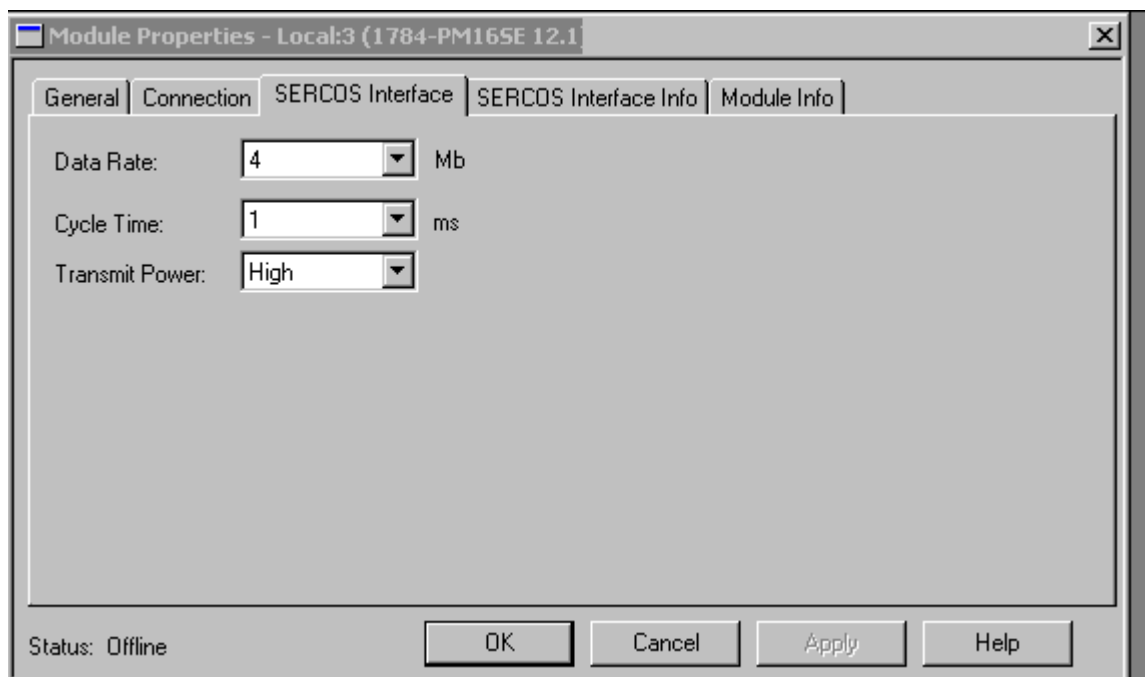
Displays the fault code returned from the controller (related to the card you are configuring) and the text detailing the Module Fault that has occurred.

The following are common categories for errors:

- Connection Request Error - The controller is attempting to make a connection to the card and has received an error. The connection was not made.

- Service Request Error - The controller is attempting to request a service from the card and has received an error. The service was not performed successfully.
- Module Configuration Invalid - The configuration in the card 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 card.

**SERCOS Interface Tab** The SERCOS interface Tab is for configuring the SERCOS ring. It is here where you set the specific Data Rate, Cycle Time, and Transmit Power for the named 1784-PM16SE SERCOS interface card.



**Figure 4.9 Module Properties - SERCOS Interface Tab**

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 1784-PM16SE motion controller.

**TIP**

The settings on this tab are specific to the 1784-PM16SE motion controller.

**Data Rate**

Select the baud rate for the SERCOS ring. Your options are:

- Auto Detect – automatically scans to detect the SERCOS ring baud rate as set by the drive(s).
- 4 Mb – sets the SERCOS ring baud rate to 4 Mb. This value must match the baud rate set on the drives. All drives included in the ring must be set to 4 Mbaud.
- 8 Mb – sets the SERCOS ring baud rate to 8 Mb. This value must match the baud rate set on the drives. All drives included in the ring must be set to 8 Mbaud.

**IMPORTANT**

If drives are set to both 4 and 8 Mbaud rates and the motion card's Data Rate is set to 4 Mbaud, it only detects the 4Mbaud drives and indicates a closed ring. Those drives set to 8 Mbaud are ignored. When the program is run it errs because the drives set to 8 Mbaud are not found.

In the above scenario with the motion card's Data Rate set to 8 Mbaud, it errs with "Wrong baud rate".

**Cycle Time**

This field sets the rate at which drives on the SERCOS ring are updated. Select the update rate for the SERCOS ring:

- 0.5 ms

**Note:** Many drives do not support an update rate of 0.5 ms. Check your drive documentation for appropriate values.

- 1 ms
- 2 ms.

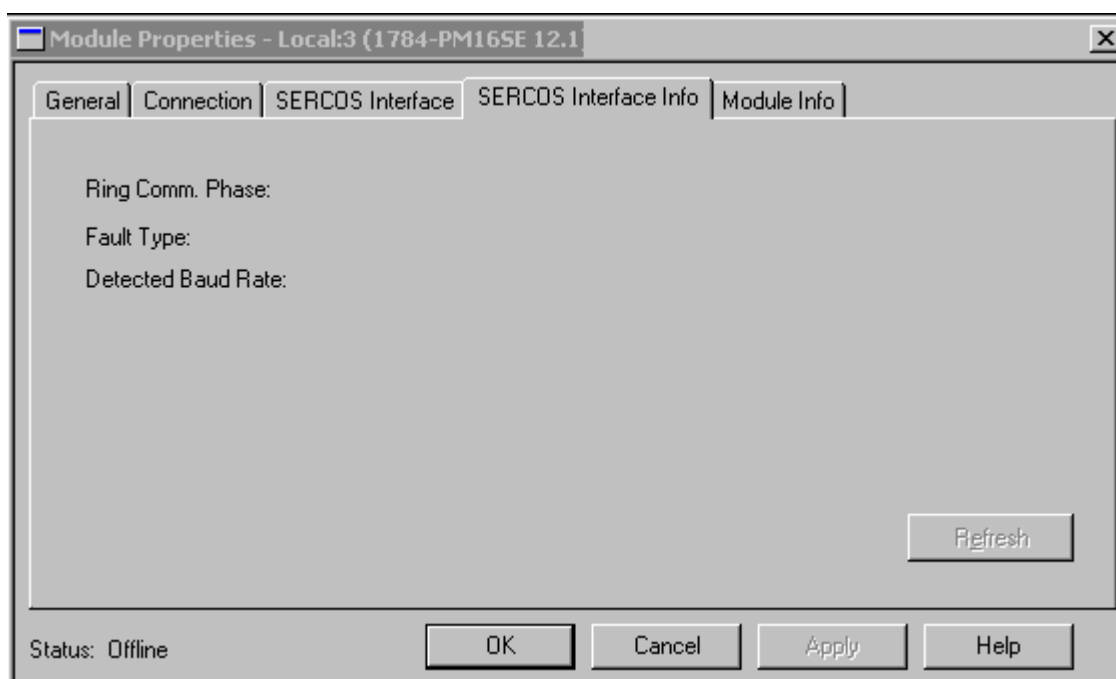
## Transmit Power

Select the optic transmit power range for the SERCOS ring:

- High
- Low

It is recommended that you set to High.

**SERCOS Interface Info Tab** The SERCOS interface Tab is for monitoring the SERCOS ring of the selected 1784-PM16SE while it is on-line. A REFRESH button is available to access the current values.



**Figure 4.10 Module Properties - SERCOS Interface Info Tab**

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
- Loss of received signal
- MST error
- Missed AT
- Excessive AT errors
- Duplicate nodes (not currently supported)
- No nodes
- Wrong ring cycle
- Wrong baud rate
- Link transport fault
- Wrong phase
- Wrong AT number

## Refresh

Click this button to update this page.

**Note:** this information does not refresh automatically.

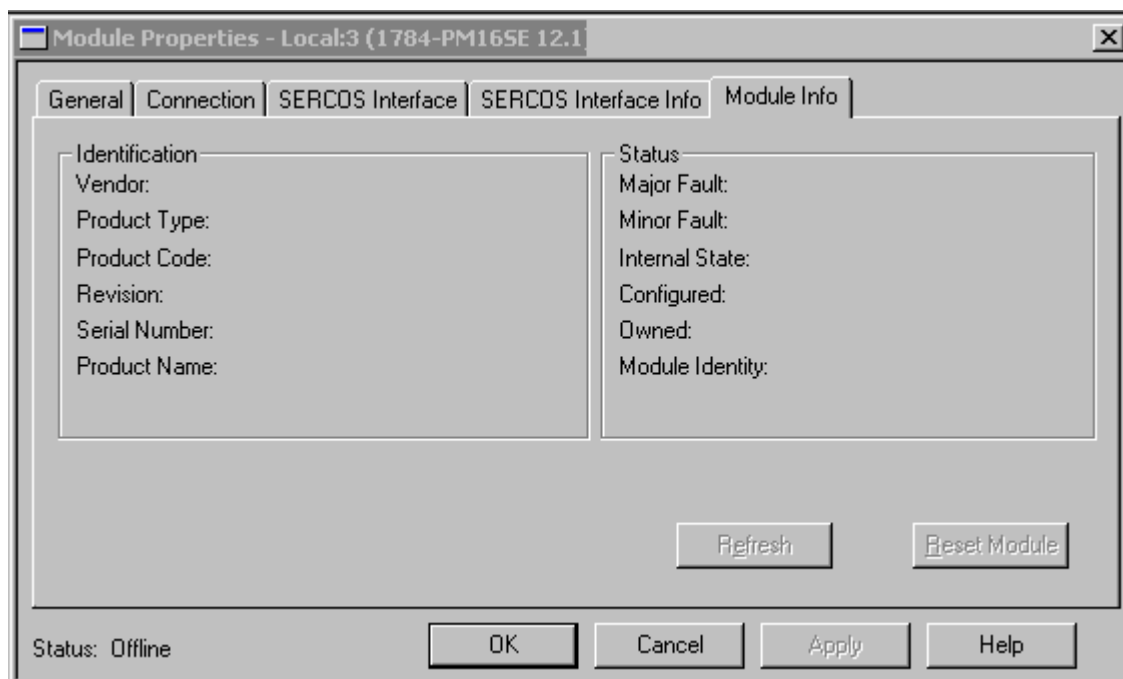
**Module Info Tab** The Module Info tab contains information about the selected card, however, you can click on:

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

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

Use this tab to determine the identity of the card.

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



**Figure 4.11 Module Properties - Module Info Tab**

### Identification

Displays the card's:

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

The name displayed in the Product Name field is read from the card. This name displays the series of the card.

## Major/Minor Fault Status

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

## Internal State Status

This field displays the card'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 card from the card selection tab, this field displays a hexadecimal value. A textual description of this state is only given when the card identity you provide is a match with the actual card.

## Configured

This field displays a yes or no value indicating whether the card has been configured by an owner controller connected to it. Once a card has been configured, it stays configured until the card is reset or power is cycled, even if the owner drops connection to the card.

## Owned

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

## Module Identity

<b>Displays:</b>	<b>If the card in the physical slot:</b>
Match	agrees with what is specified on the General Tab. In order for the Match condition to exist, all of the following must agree: <ul style="list-style-type: none"><li>• Vendor</li><li>• Module Type (the combination of Product Type and Product Code for a particular Vendor)</li><li>• Major Revision</li></ul>
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 card that were specified on the General Tab.

### Refresh

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

### Reset Module

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

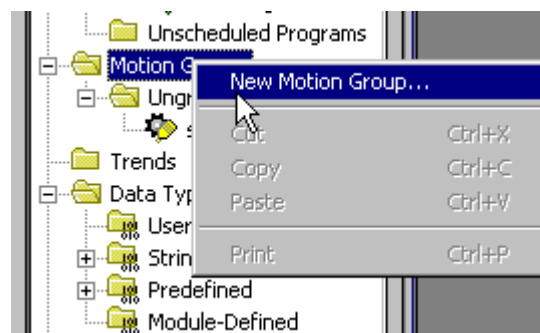
Resetting a card causes all connections to or through the card to be closed, and this may result in loss of control.

## The Motion Group

### Creating A Motion Group

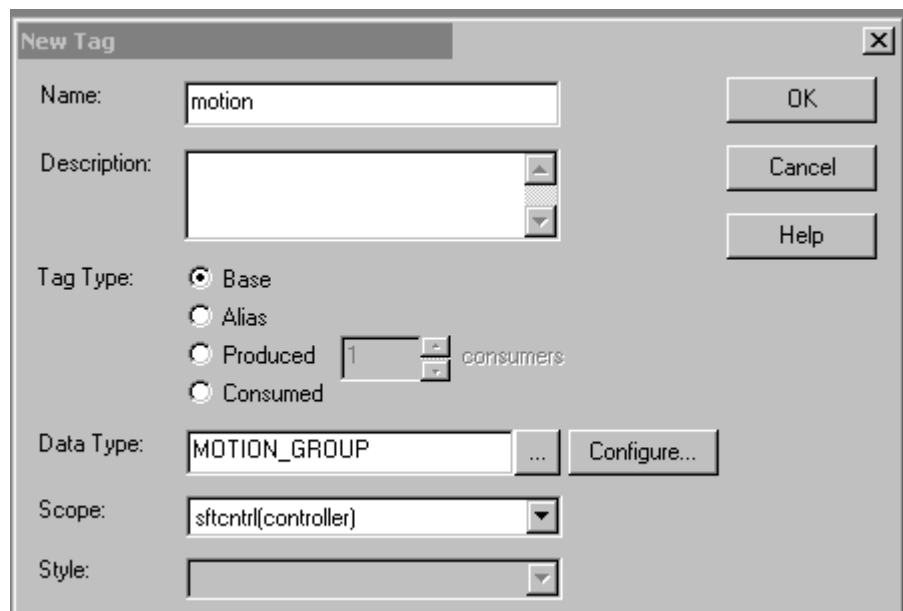
Each .acd program must have one motion group. (There can be only one.) You must create it before an axis can be assigned to the group and have it function within the .acd program.

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



**Figure 5.1 Controller Organizer - New Motion Group Pop-up**

This calls the New Tag window.



**Figure 5.2 New Tag Dialog**

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.

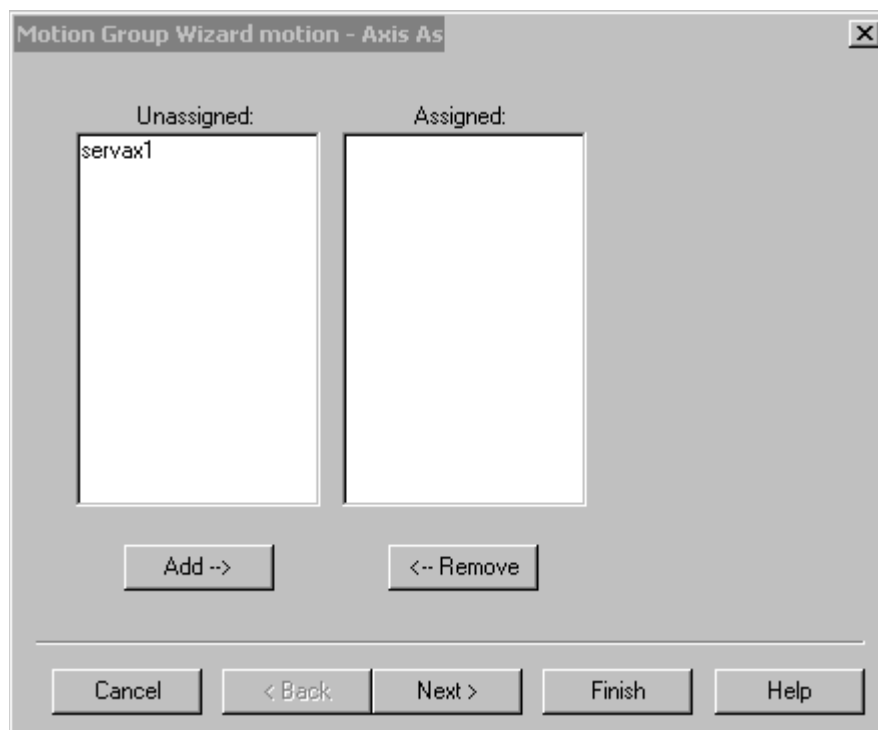
---

7. Click on the Configure button to proceed through the Motion Group Wizard screens to set the properties for the motion group.

If you had clicked on OK instead of the Configure button, it would have created the group and closed the dialog. You would then need to access the Motion Group Properties screen to configure the Motion Group.



The Motion Group Wizard group - Axis Assignment screen displays.



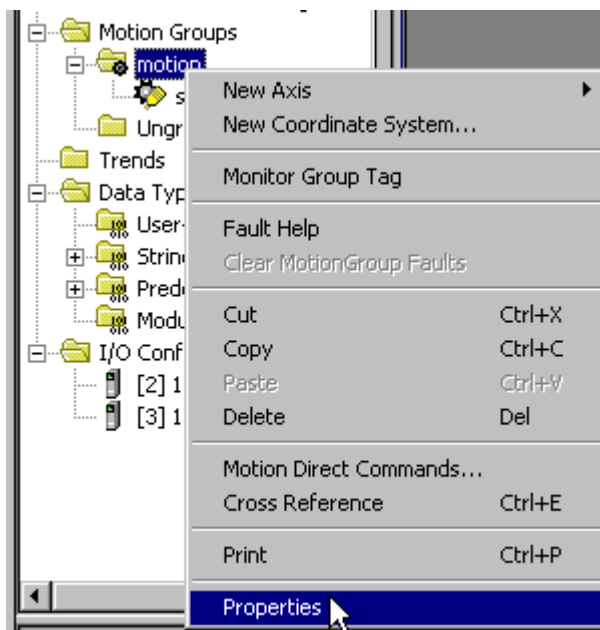
**Figure 5.3 Motion Group Wizard Dialog - Axis Assignment**

Add any existing axes to the group.

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

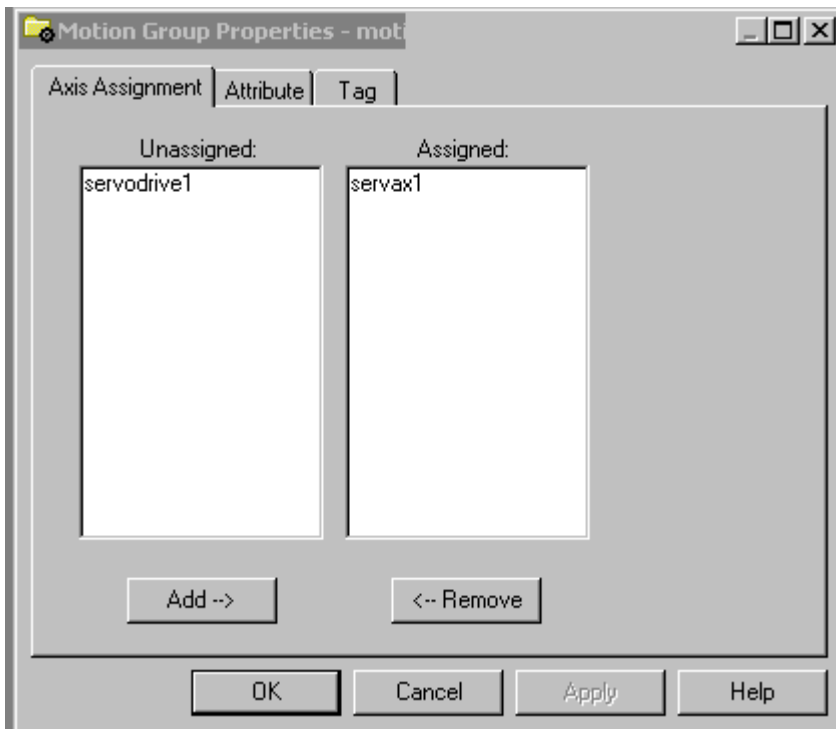
## Editing the Motion Group Properties

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



**Figure 5.4 Controller Organizer | Motion Group| Properties**

The Motion Group Properties tabbed screen displays.



**Figure 5.5 Motion Group Properties - Axis Assignment Tab**

**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.

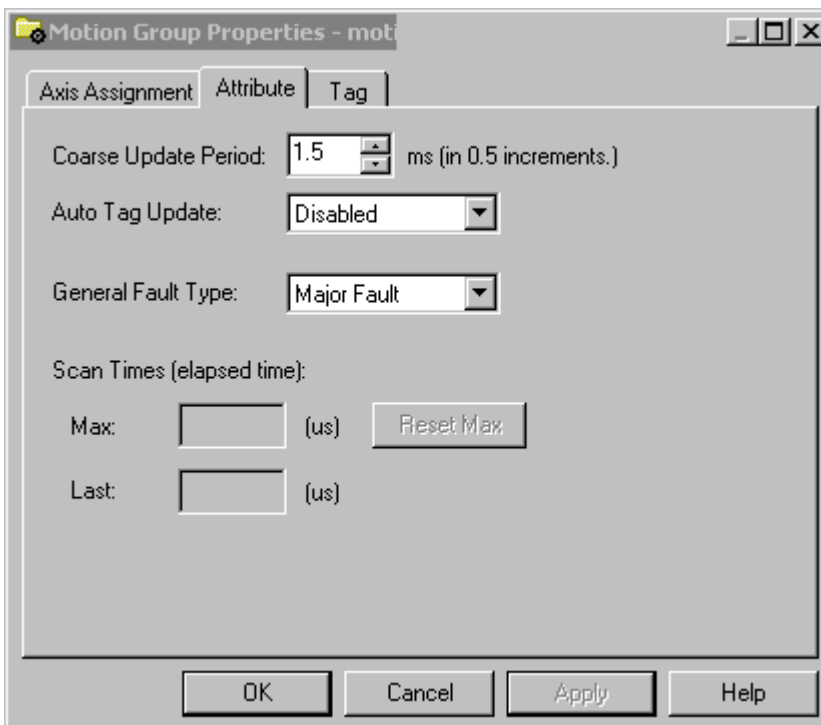
### Add

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.



**Figure 5.6 Motion Group Properties - Attribute Tab**

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 1784-PM02AE or 1784-PM16SE modules when executing motion instructions.

### **Auto Tag Update**

This parameter determines whether or not the axis parameter values are automatically updated during operation. Choose either:

- 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.

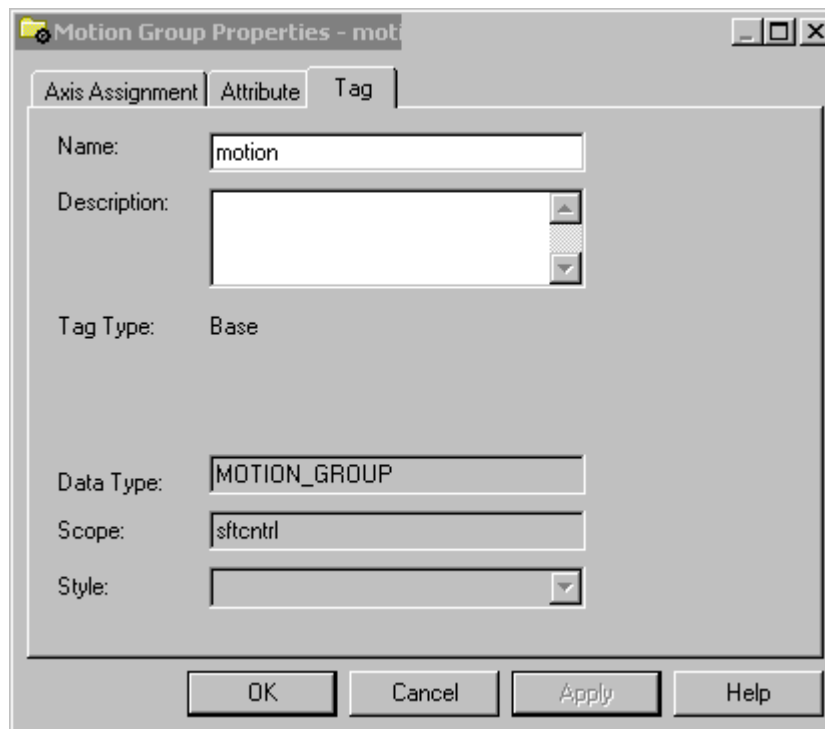
### **Scan Times (elapsed time)**

- 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.



**Figure 5.7 Motion Group Properties - Tag Tab**

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. Be careful while reading this information. Many of the screens appear to be the same (and many are) but some of the screens change in content based on the type of axis. They are labeled where different so read through the entire section to make sure you find the explanations for the type of axis selected.

### 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**.

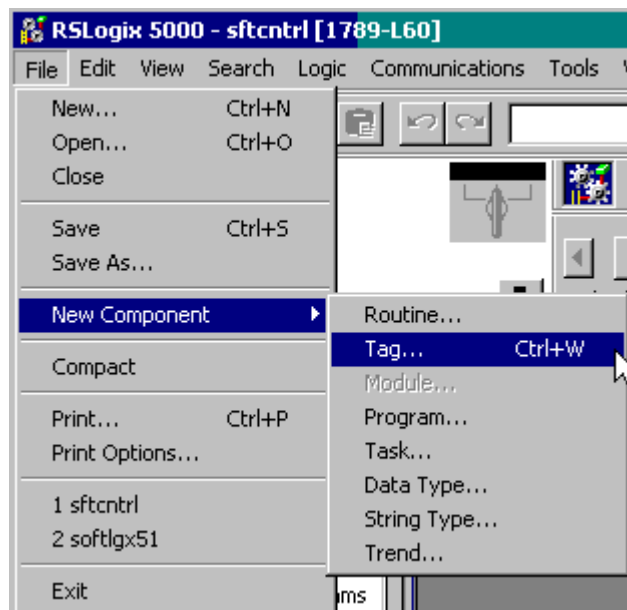
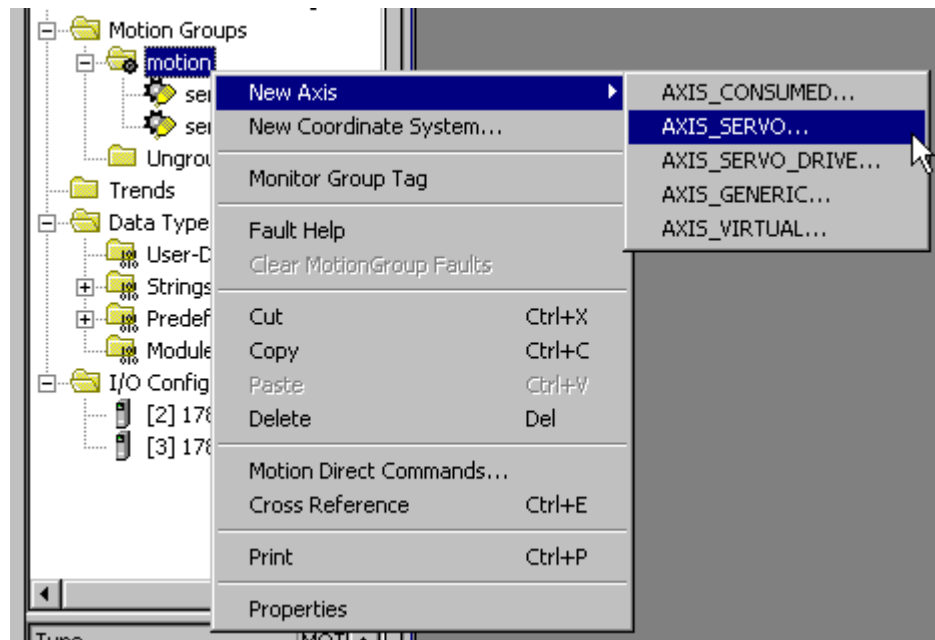


Figure 6.1 File Menu to New Component to Tag

You can also right click on the Motion Group and select New Axis and the type of axis tag you want to create from the menu.



**Figure 6.2 Naming an Axis From Motion Group**

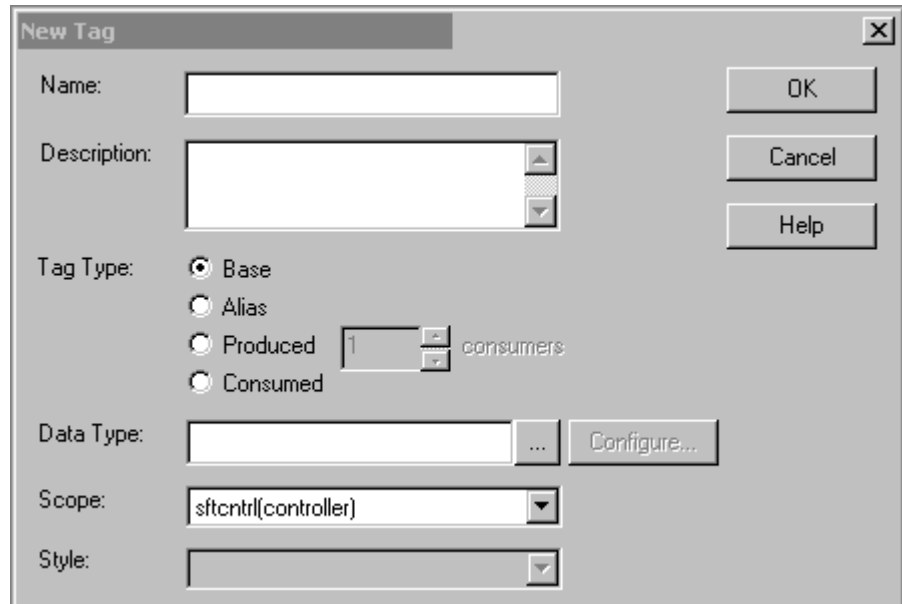
You can also initiate a new axis by right clicking on Ungrouped Axes and selecting the type of axis you want to create.



**Figure 6.3 Naming an Axis From Ungrouped Axes**



The New Tag window appears.



**Figure 6.4 New Tag Dialog**

If you accessed the New Tag window from either Motion Group or Ungrouped Axes, the Data Type is already filled in.

**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.

---

**ATTENTION**

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.	
	<b>To use the axis</b>	<b>Select</b>
	Within the entire program	Controller

## Editing Motion Axis Properties

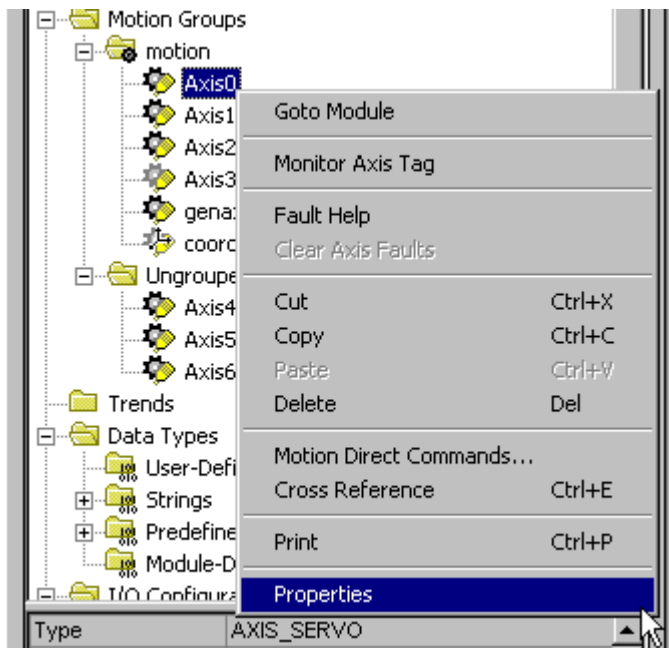
Once you have named your axis in the New Tag window, you must then configure it. You can make your configuring options in the Axis Properties screen. These have a series of Tabs that access a specific dialog for configuring the axis. Make the appropriate entries for each of the fields. An asterisk appears 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



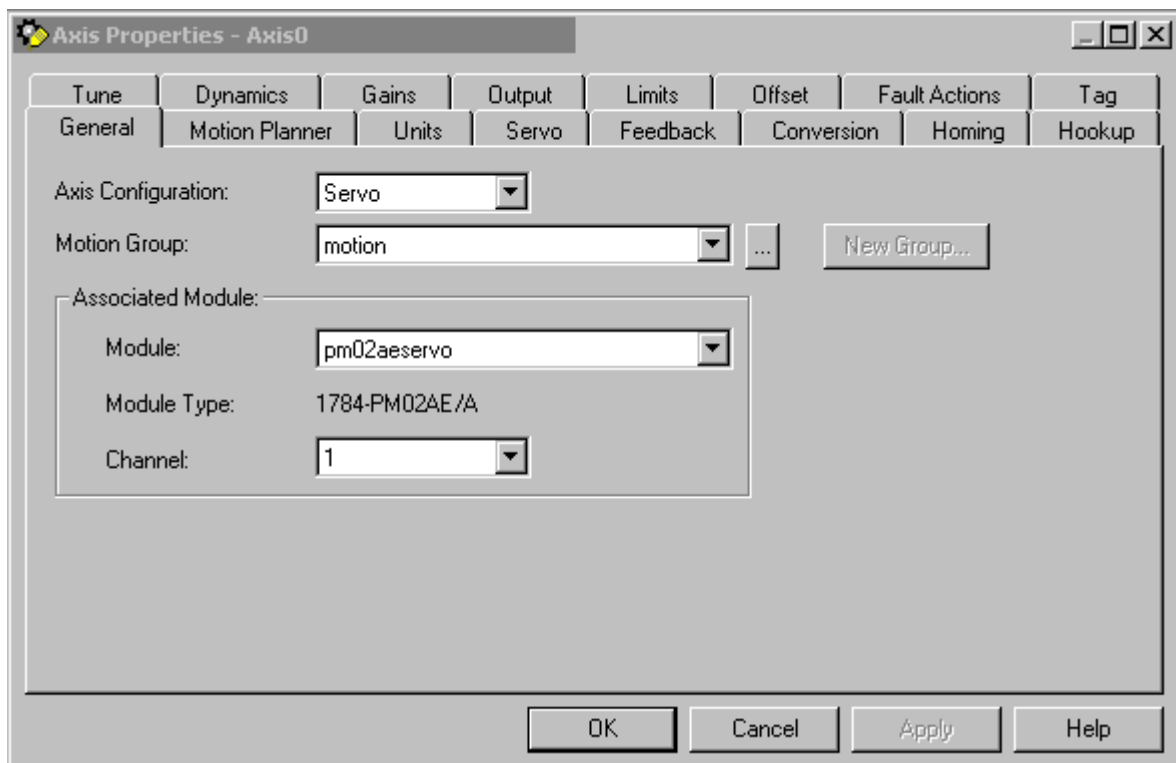
When you configure your axis, some fields may be unavailable (greyed-out) because of choices you made in the New Tag window.

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



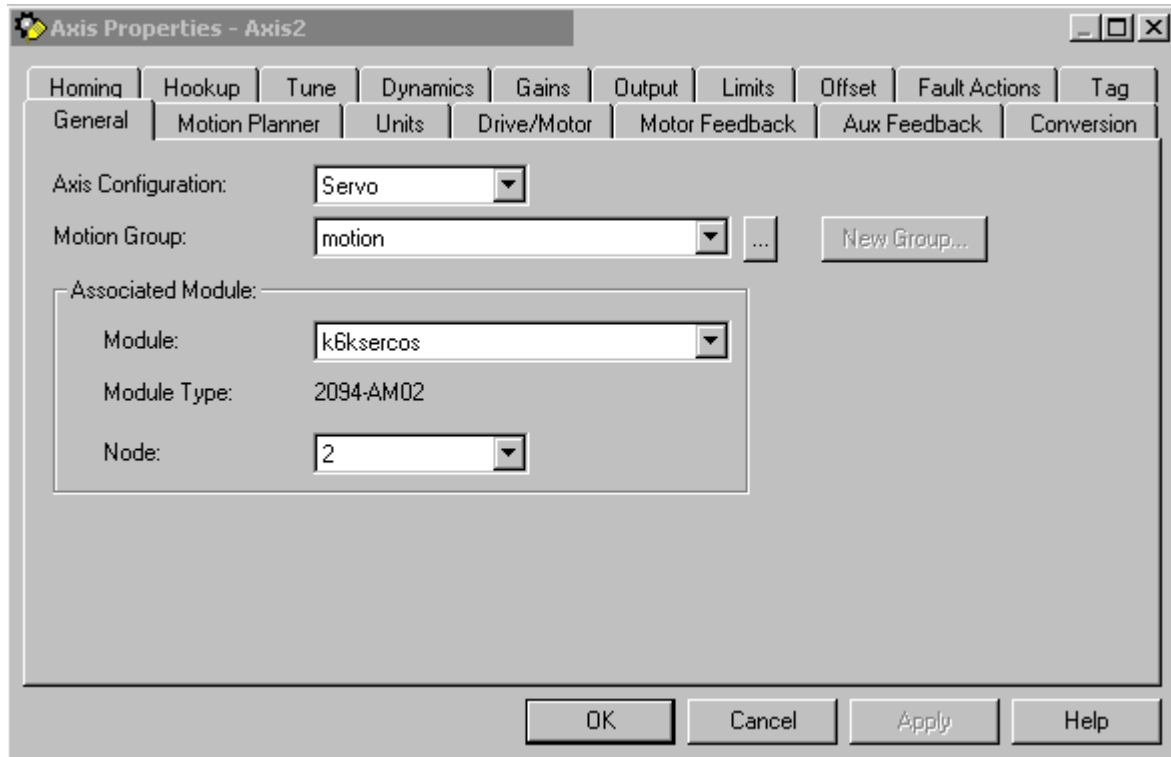
**Figure 6.5 Accessing Axis Properties from Controller Organizer**

The Axis Properties General window appears. The General screen depicted below is for an AXIS\_SERVO data type.



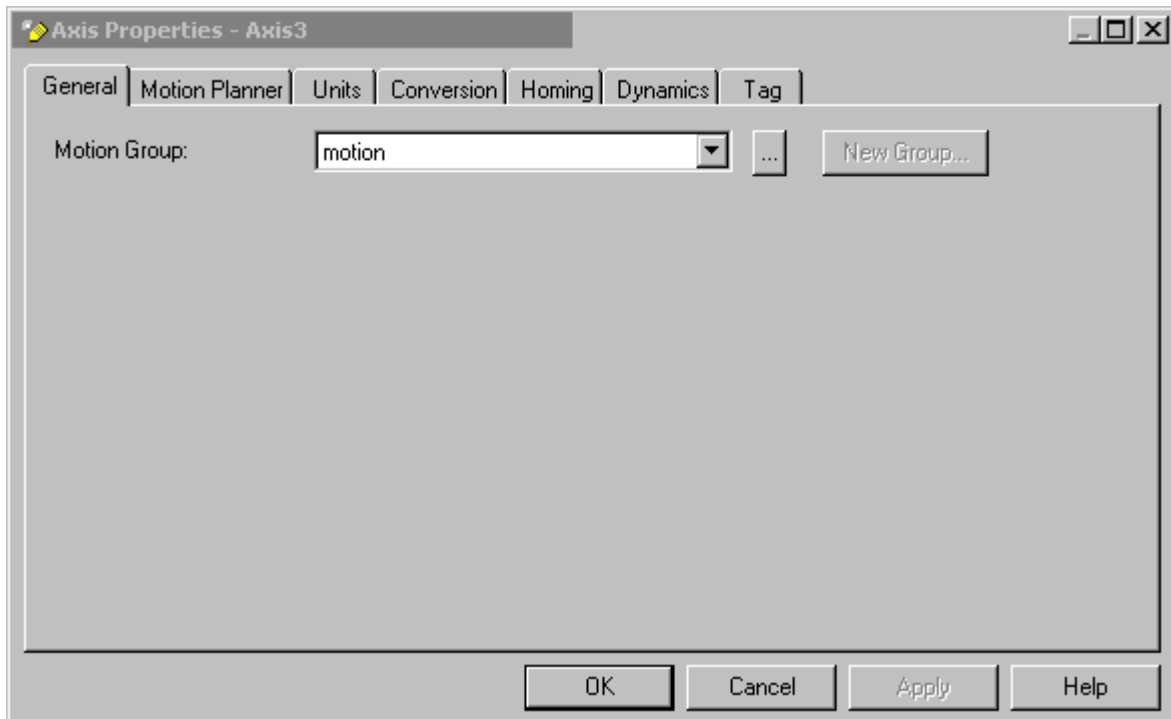
**Figure 6.6 Axis Properties - General Tab for Axis\_Servo**

The General screen shown below is for an AXIS\_SERVO DRIVE Data Type.



**Figure 6.7 Axis Properties - General Tab for Axis\_Servo\_Drive**

The AXIS\_VIRTUAL General Tab is shown below.



**Figure 6.8 Axis Properties - General Tab for Axis\_Virtual**

**General Tab – AXIS\_SERVO** 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 1784-PM02AE motion module.
- Select the channel, 0 or 1, on the 1784-PM02AE 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 1784-PM02AE motion modules. Displays <none> if the axis is not associated with any motion module.

## Channel

Selects and displays the 1784-PM02AE motion module channel - either 0 or 1 - to which the axis is assigned. Disabled when the axis is not associated with any motion module.

## 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 SERCOS Drive.
- View the base node of the associated 1784-PM16SE 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 SERCOS drive to which the axis is associated. Displays <none> if the axis is not associated with any drive.

## Module Type

This read-only field displays the type of SERCOS drive, if any, with which the axis is associated. An axis of the `AXIS_SERVO_DRIVE` data type can be associated only with 1784-PM16SE motion modules. Displays <none> if the axis is not associated with any drive.

## Node

Displays the base node of the associated SERCOS drive. Disabled when the axis is not associated with any drive.



**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.

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



**Figure 6.9 Axis Properties – Motion Planner Tab**

### Motion Planner Tab

The Motion Planner Tab is where you set/edit the number of Output Cam execution targets, the type of stop action to use, enable or disable Master Delay Compensation, enable or disable Master Position Filter, and set the bandwidth for Master Position Filter Bandwidth.

### 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.

## Program Stop Action

Select how a specific axis is stopped 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 braking, 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 braking, this results in the axis coasting to a stop. To recover from this state, a reset instruction must be executed.

## Master Delay Compensation Checkbox

Use this checkbox to Enable/Disable Master Delay Compensation. Master Delay Compensation is used to balance the delay time between reading the master axis command position and applying the associated slave command position to the slave's servo loop. This feature ensures that the slave axis command position accurately tracks the actual position of the master axis i.e. zero tracking error.

Clicking on this box enables Master Delay Compensation. The default setting is Disabled.

If the axis is configured for Feedback only, Master Delay Compensation should be disabled.

## Enable Master Position Filter Checkbox

Use this checkbox to Enable/Disable Master Position Filter. The default is disabled and must be checked to enable position filtering. Master Position Filter, when enabled, effectively filters the specified master axis position input to the slave axis's gearing or position camming operation. The filter smooths out the actual position signal from the master axis, and thus smooths out the corresponding motion of the slave axis.

When this feature is enabled the Master Position Filter Bandwidth field is enabled.

## Master Position Filter Bandwidth

The Master Position Filter Bandwidth field is enabled when the Enable Position Filter checkbox is selected. This field controls the bandwidth for master position filtering. Enter a value in Hz in this field to set the bandwidth to for the Master Position Filter.

### IMPORTANT

A value of zero for Master Position Filter Bandwidth effectively disables the master position filtering.

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

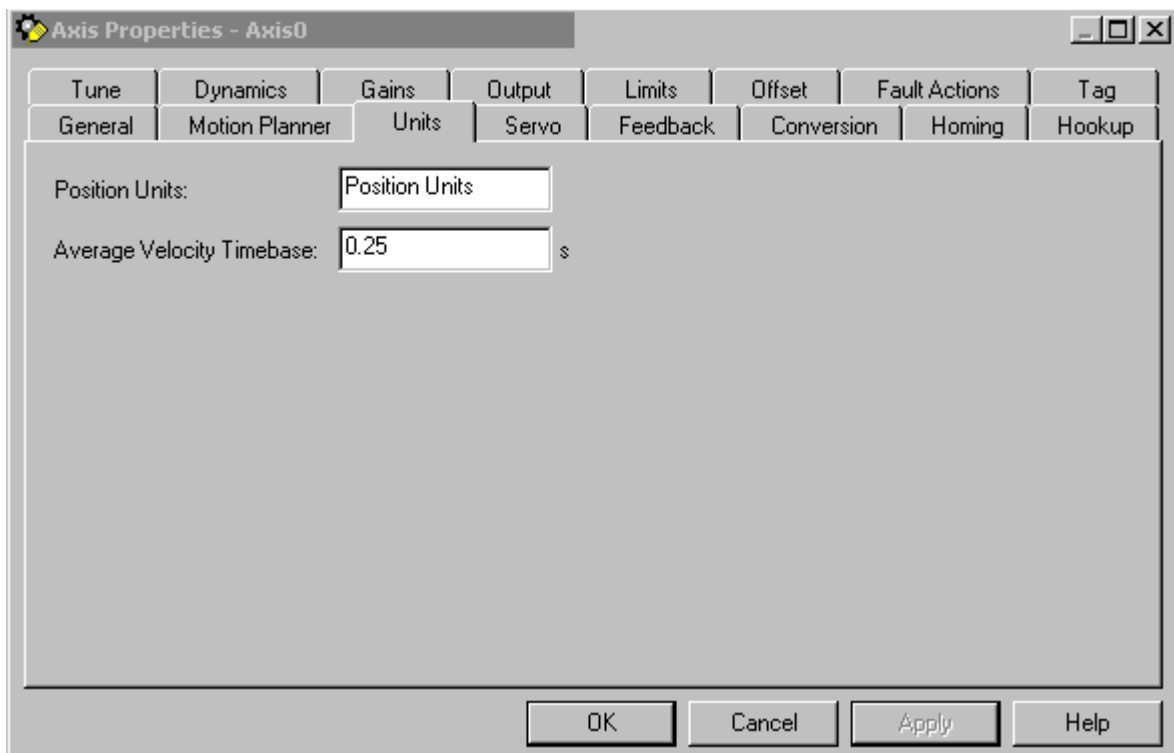


Figure 6.10 Axis Properties - Units Tab

**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.

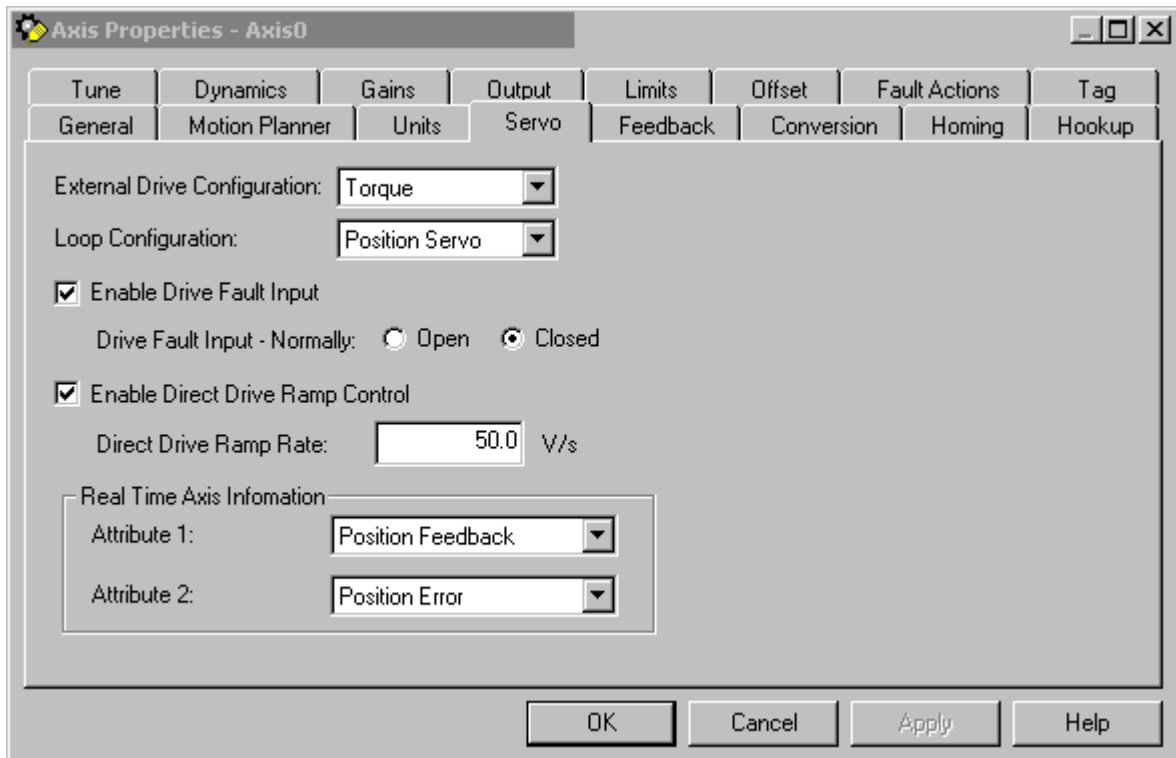
### **Average Velocity Timebase**

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.

**Servo Tab - AXIS\_SERVO** Click on the Servo Tab from the Axis Properties for AXIS\_SERVO to access the Servo dialog.



**Figure 6.11 Axis Properties - Servo Tab for Axis\_Servo**

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 are 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 is static and does not reflect the true value in the servo module.

Click on the Apply button to accept your changes.

**Feedback Tab – (AXIS\_SERVO)** The Feedback Tab allows you to select the type of Feedback used with your Servo axis.



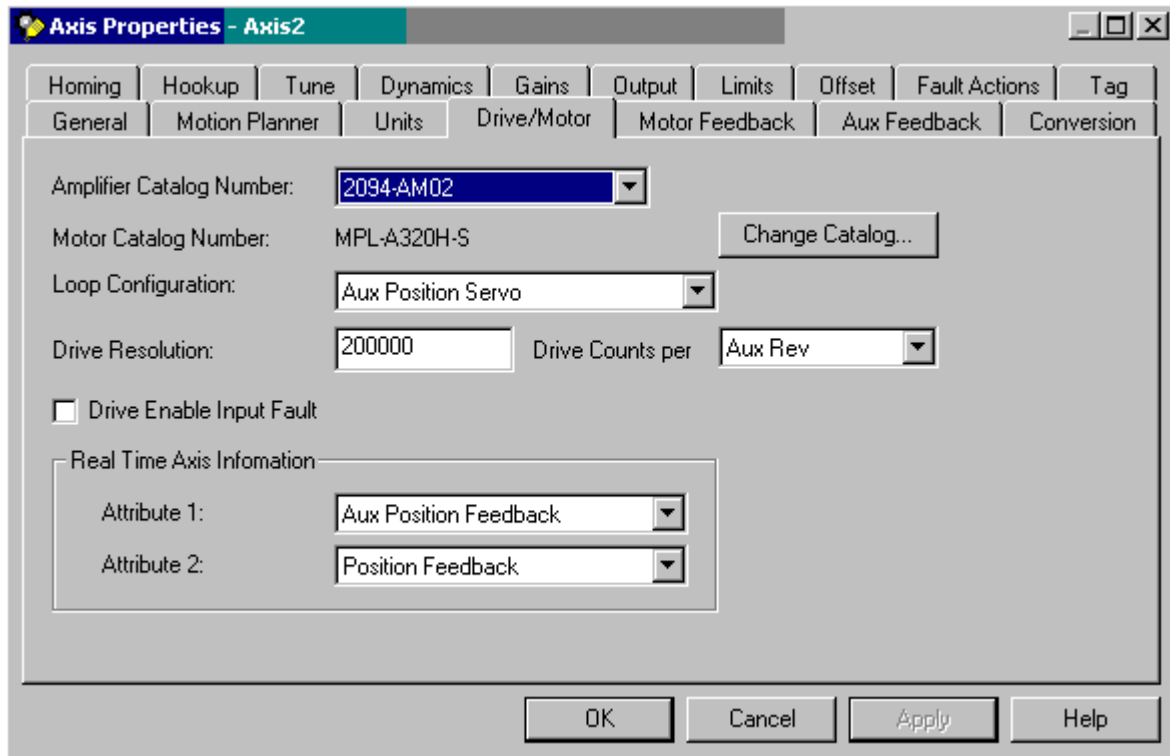
**Figure 6.12 Axis Properties - Feedback Tab for Axis\_Servo**

### **Feedback Type**

Select the appropriate Feedback for your current configuration.



**Drive/Motor Tab - (AXIS\_SERVO\_DRIVE)** Use this tab to configure the servo loop for an AXIS\_SERVO\_DRIVE axis, and open the Change Catalog dialog box.



**Figure 6.13 Axis Properties - Drive Tab for 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.

### Amplifier Catalog Number

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

## 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.

<b>On this tab or dialog:</b>	<b>These attributes are recalculated:</b>
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.

## Loop Configuration

Select the configuration of the servo loop:

- Motor Feedback Only – Displayed when Axis Configuration is Feedback only
- Aux Feedback Only – Displayed when Axis Configuration is Feedback only
- Position Servo
- Auxiliary Position Servo (not applicable to Ultra3000 drives)
- Dual Position Servo
- Dual Command Servo
- Auxiliary Dual Command Servo
- Velocity Servo
- Torque Servo

## Drive Resolution

Type in the number of counts per motor revolution. This value applies to all position data. Valid values range from 1 to  $2^{32} - 1$ . One Least Significant Bit (LSB) for position data equals  $360^\circ / \text{Rotational Position Resolution}$ .

**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/Attribute 2

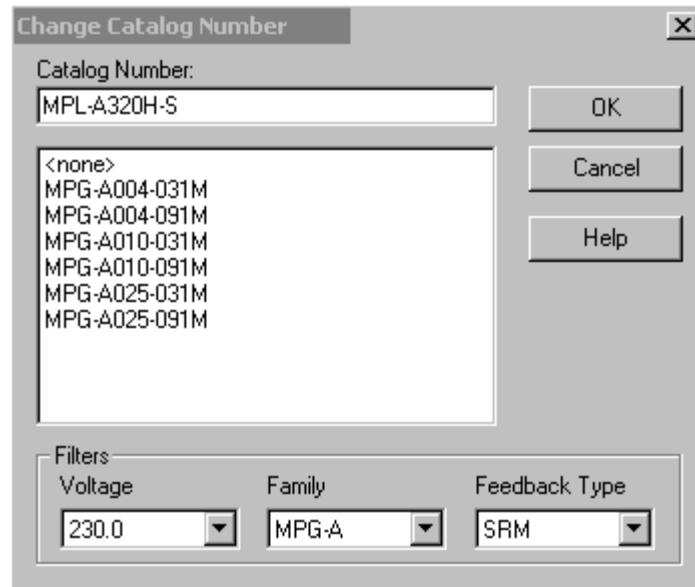
Select up to two axis attributes whose status are transmitted – along with the actual position data – to the Logix processor. The values of the selected attributes can be accessed via the standard GSV or Get Attribute List service.

**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 is static and does not reflect the true value in the servo module.

### Change Catalog...button

The Change Catalog button accesses the motor database and provides for selecting a new motor catalog number. There are three boxes that can be used for refine the selection process.



**Figure 6.14 Change Catalog Screen**

## Catalog Number

Lists the available catalog numbers from the Motor Database based on any selection criteria from the Filters fields.

## Filters

There are three optional Filter fields that allow you to refine your search of the Motor Database. The Filter boxes are defaulted to all.

### Voltage

Lets you select a voltage rating from the pull-down list to broaden or narrow your search. The default is all.

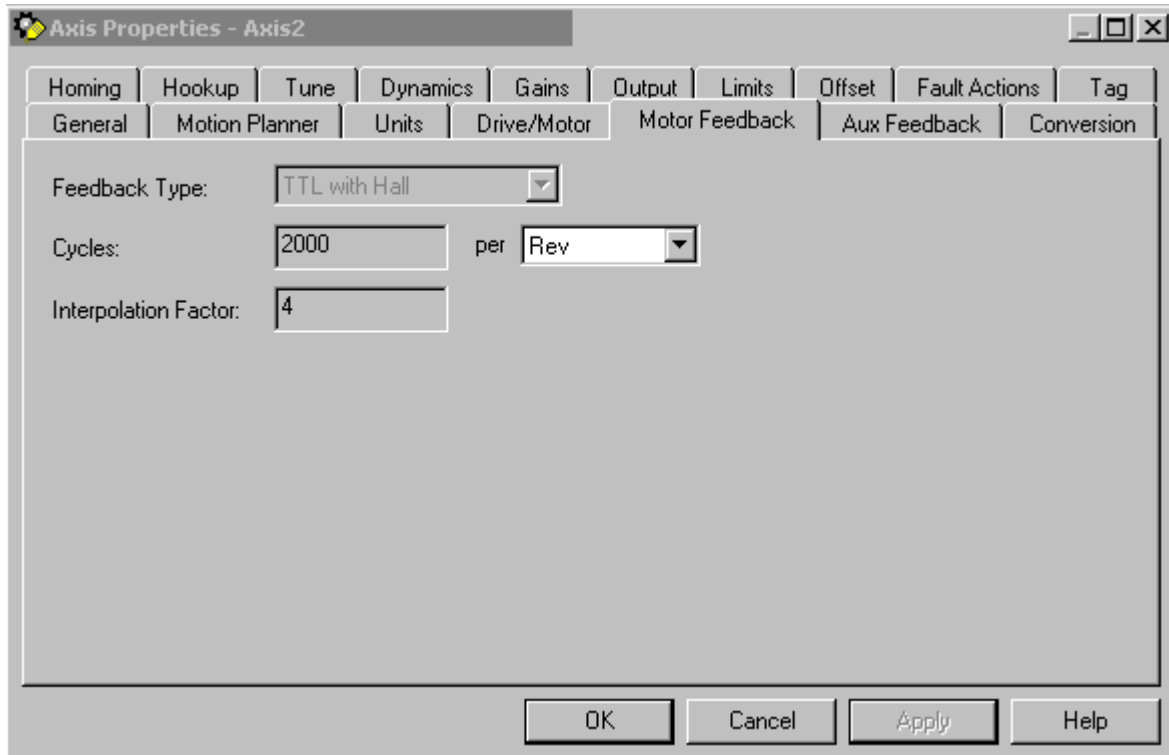
### Family

The Family filter box pull down list lets you narrow your motor search by restricting it to a particular family of motors. The default is all.

### Feedback Type

The Feedback Type filter box pull-down list lets you manipulate your motor search by acceptable Feedback types. The default is all.

**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.



**Figure 6.15 Axis Properties - Motor/Feedback Tab for 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.

### Feedback Type

This field displays the type of feedback associated with the selected motor.

### Cycles

The number of cycles 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.

## Per

The units used to measure the cycles.

## 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.

**Aux Feedback Tab - AXIS\_SERVO\_DRIVE** The Auxiliary Feedback Tab is enabled only if the Drive tab's Loop Configuration field is set to Aux Feedback Only, Aux Position Servo, Dual Position Servo, Dual Command Servo, or Aux Dual Command Servo.

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

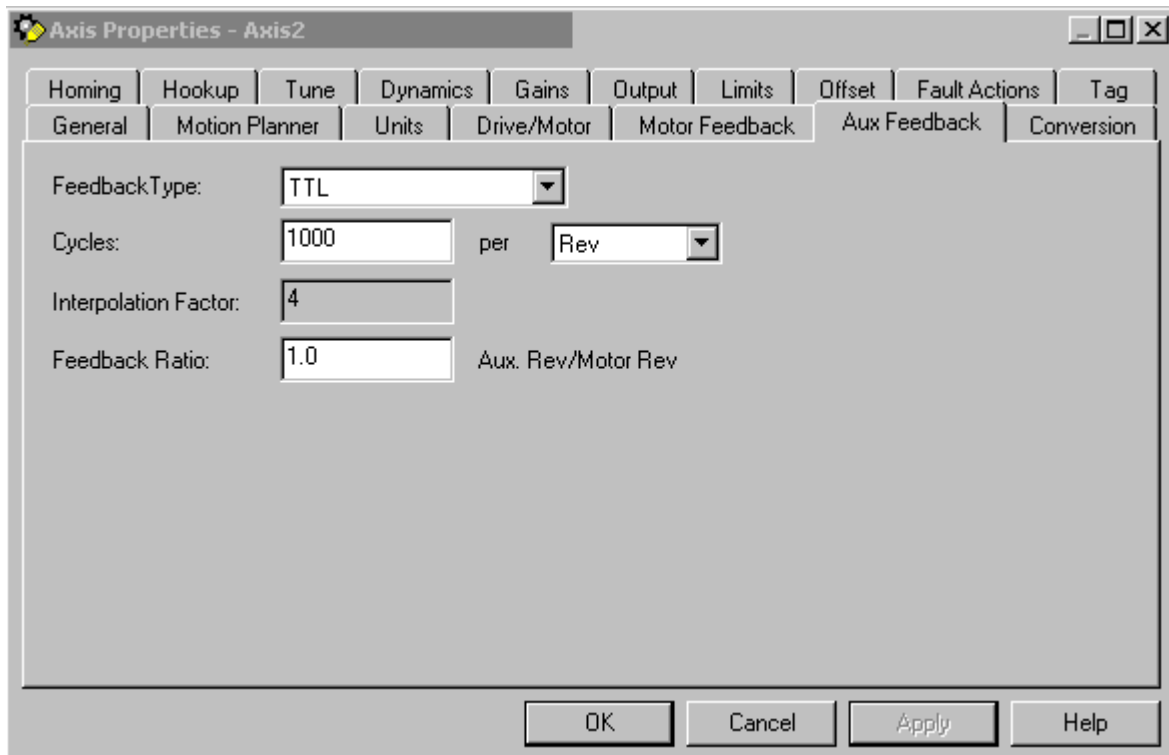


Figure 6.16 Axis Properties - Aux Feedback Tab for Axis\_Servo\_Drive

## Feedback Type

For applications that use auxiliary feedback devices, select the type of auxiliary feedback device type. These are drive dependent.

## Cycles

The number of cycles 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.

## Per

The units used to measure the cycles.

## Interpolation 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.

## Feedback Ratio

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

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

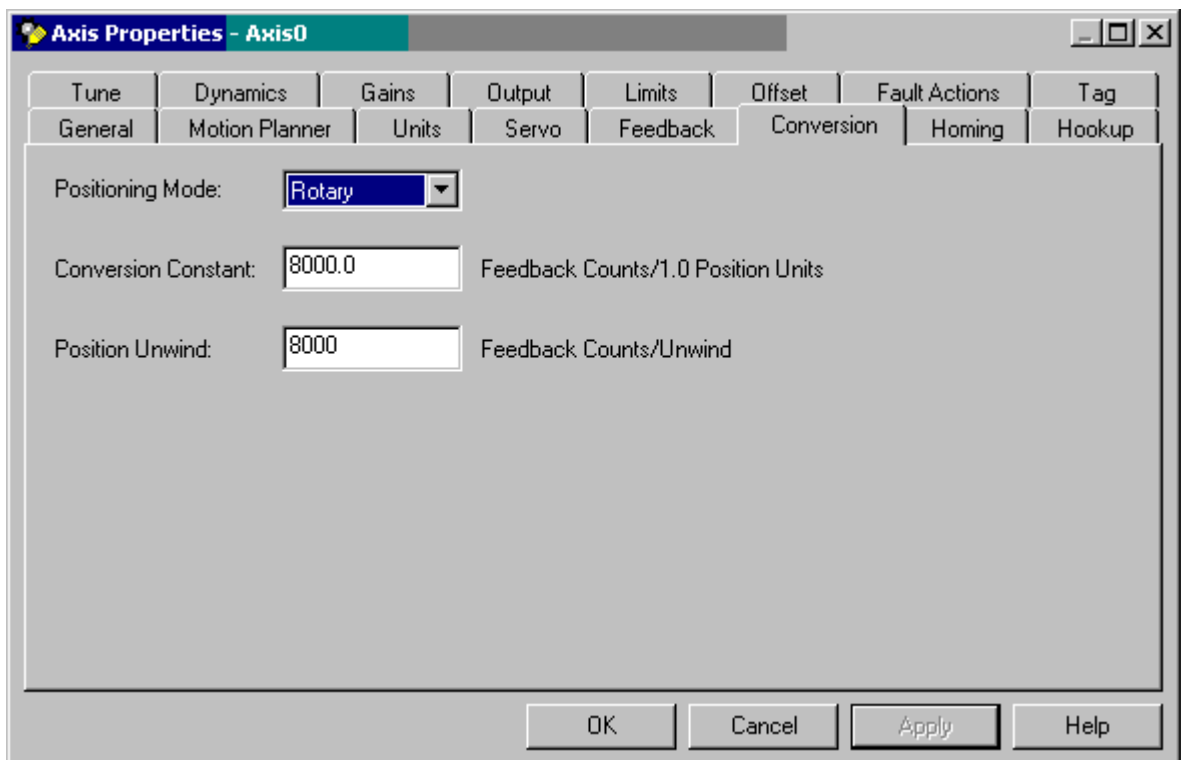
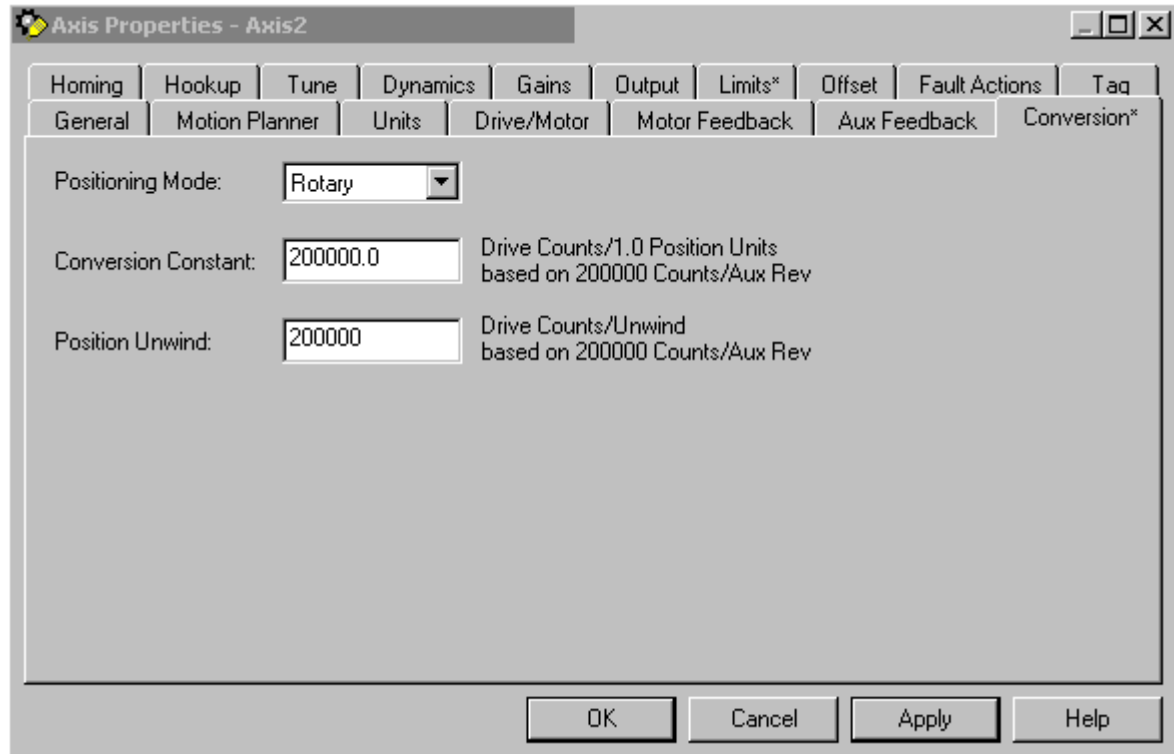


Figure 6.17 Axis Properties - AXIS\_SERVO Conversion Tab

The differences in the appearance of the Conversion Tab screens for the AXIS\_SERVO and AXIS\_SERVO\_DRIVE are the default values for Conversion Constant and Position Unwind and the labels for these values.



**Figure 6.18 Axis Properties - AXIS\_SERVO\_DRIVE Conversion Tab**

**Conversion Tab** Use this tab to view/edit the Positioning Mode, Conversion Constant, and if configured as Rotary, the Unwind values 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. This value can be edited for `AXIS_SERVO`, `AXIS_SERVO_DRIVE` and `AXIS_VIRTUAL`. The options 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 for the `AXIS_SERVO` type and for the `AXIS_SERVO_DRIVE`, 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 (`AXIS_SERVO`), 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.

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 indicates 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.

### Homing Tab - `AXIS_SERVO` and `AXIS_SERVO_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.

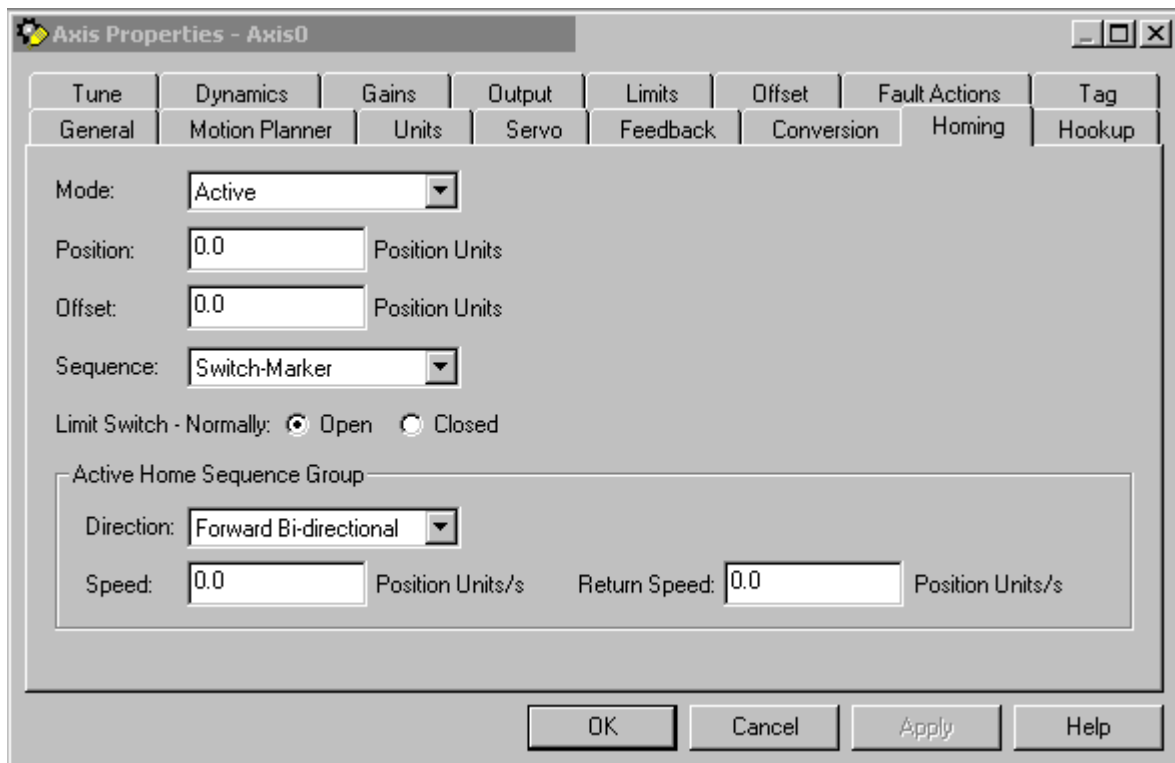


Figure 6.19 Axis Properties - Homing Tab for Axis\_Servo

## 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 is 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 is zero.

## Sequence

Select the event that causes 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

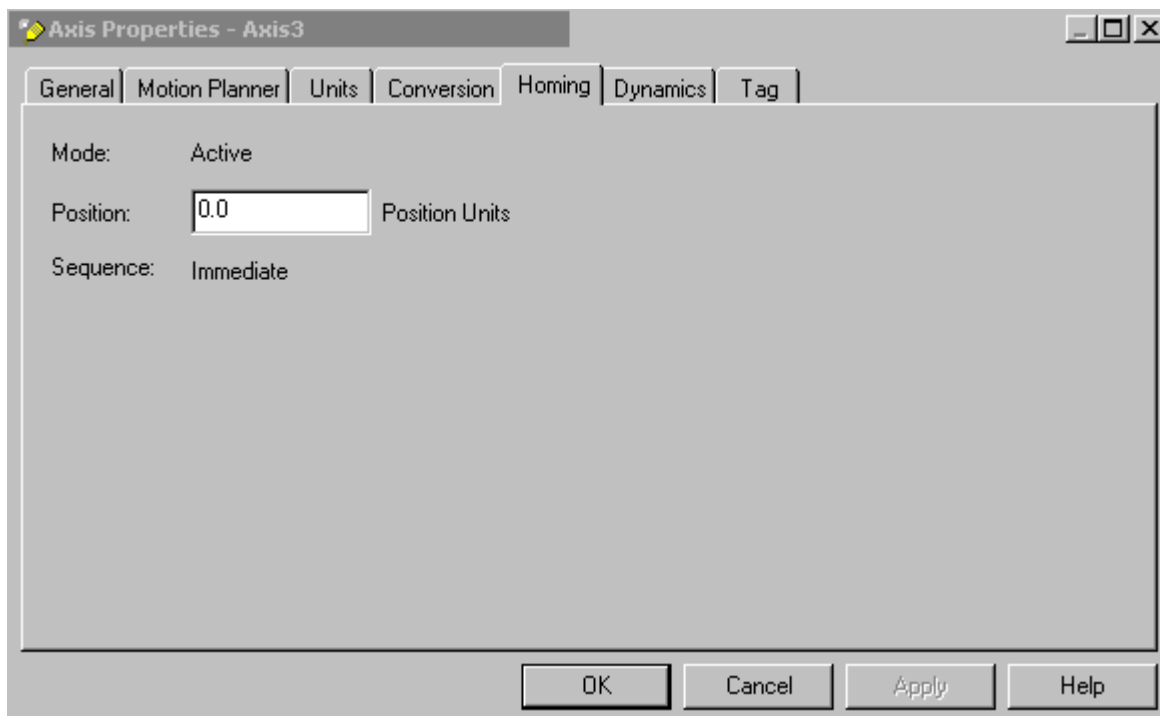
The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.

## 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.



**Figure 6.20 Axis Properties - Homing Tab for Virtual Axis Data Type**

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 is 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.

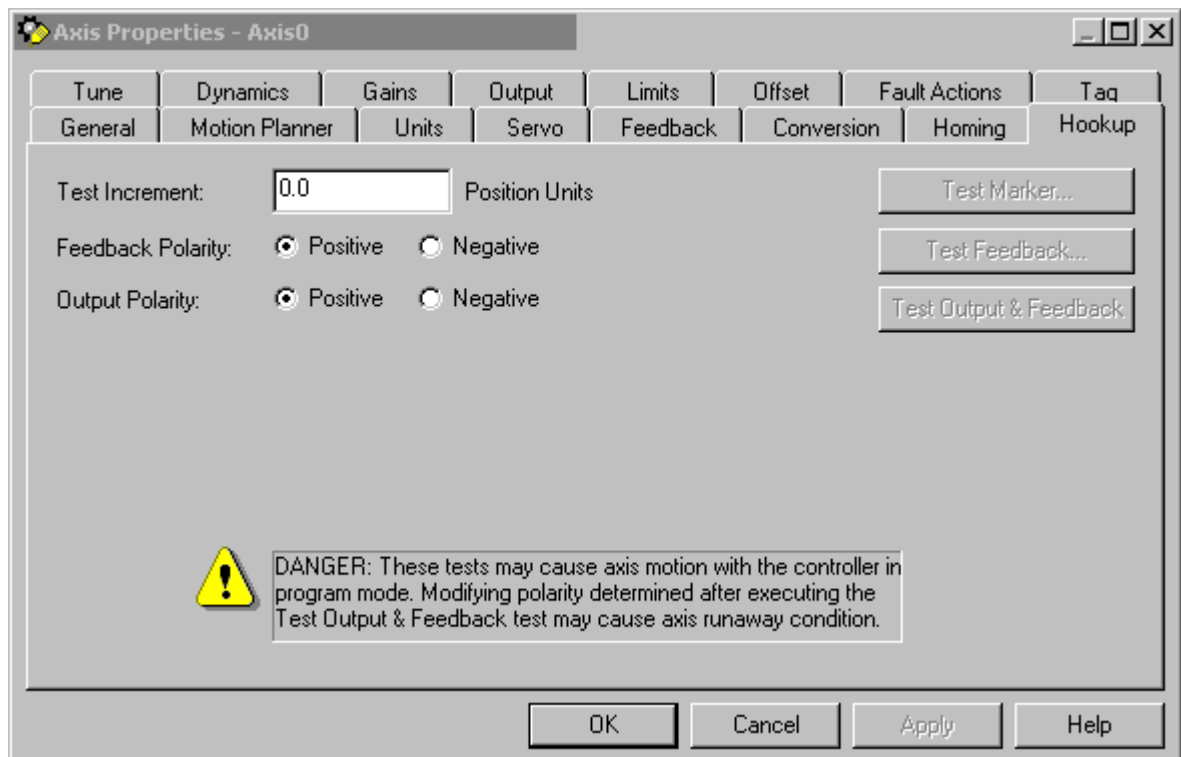
If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.

## 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.



**Figure 6.21 Axis Properties - Hookup Tab for Axis\_Servo**

## 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.

---

**ATTENTION**

Modifying automatically input polarity values by running the Feedback or Output & Feedback Tests can cause a runaway condition resulting in unexpected motion, damage to the equipment, and physical injury or death.

---

## 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

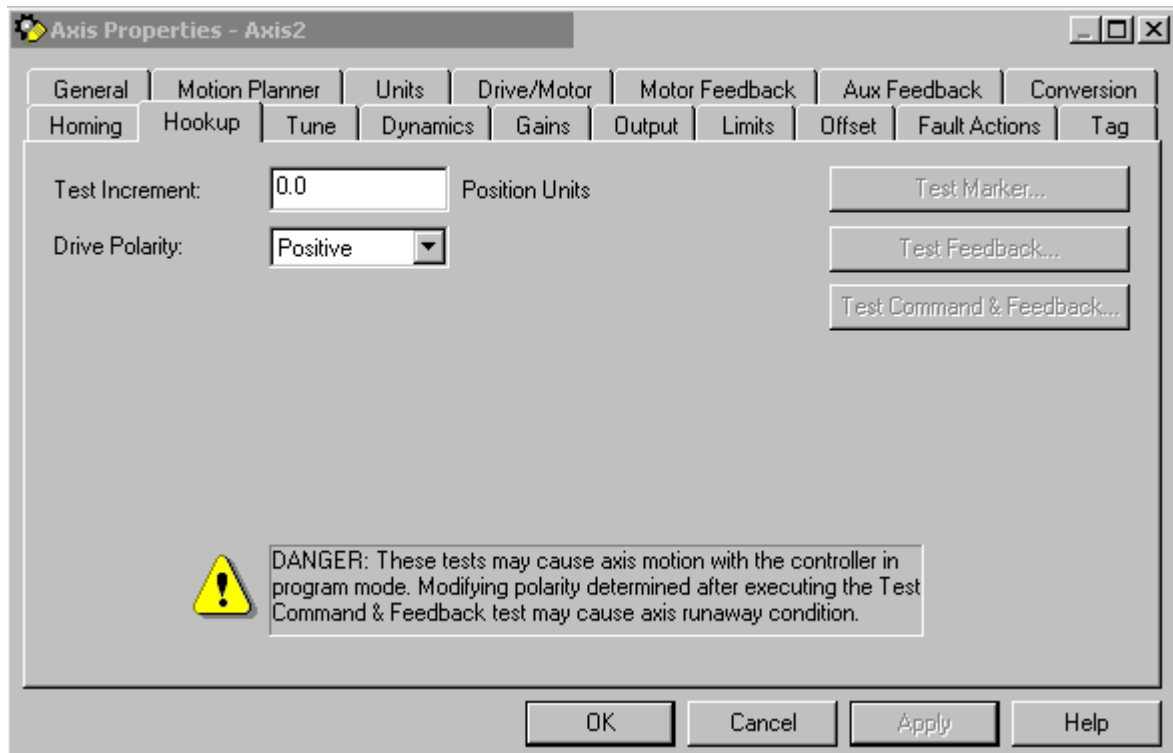
Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

## 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 - 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.



**Figure 6.22 Axis Properties - Hookup Tab for 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.

---

**ATTENTION**

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

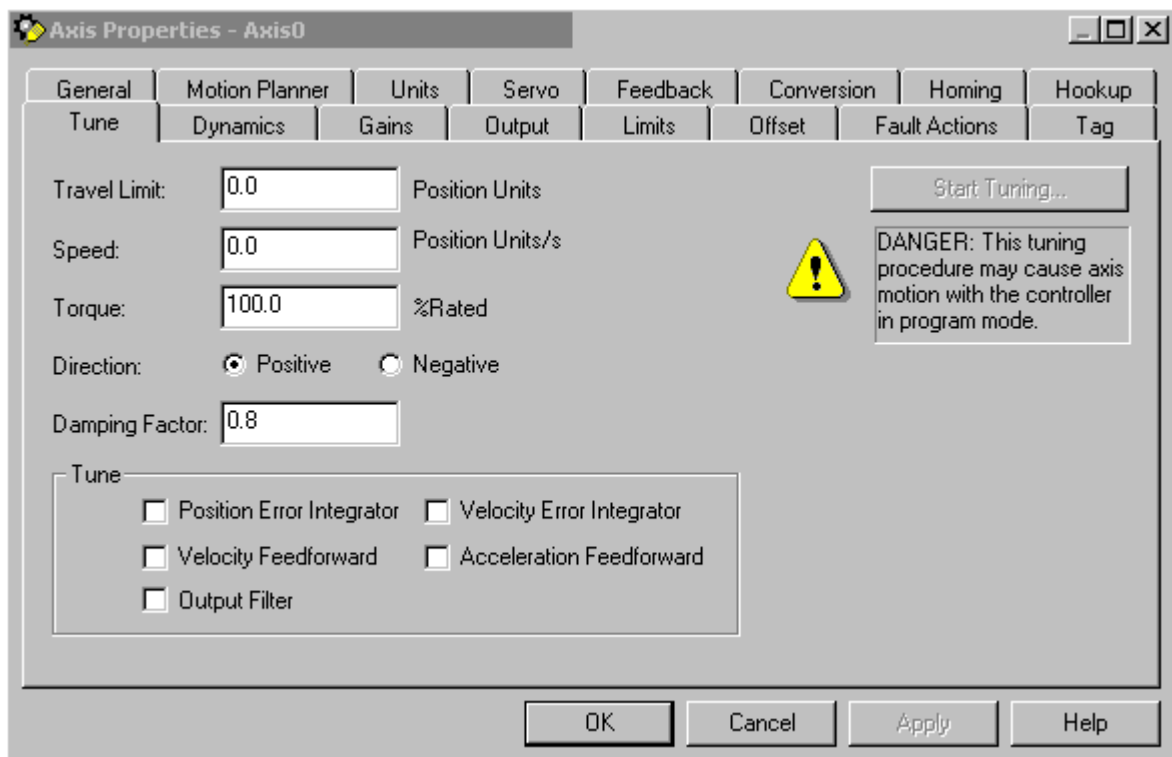
Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

## Test Command & Feedback

Runs the Command & 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.



**Figure 6.23 Axis Properties - Tune Tab for Axis\_Servo**

## Travel Limit

Specifies a limit to the excursion of the axis during the tune test. If the servo module determines that the axis is not able to complete the tuning process before exceeding the tuning travel limit, it terminates 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 are 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.

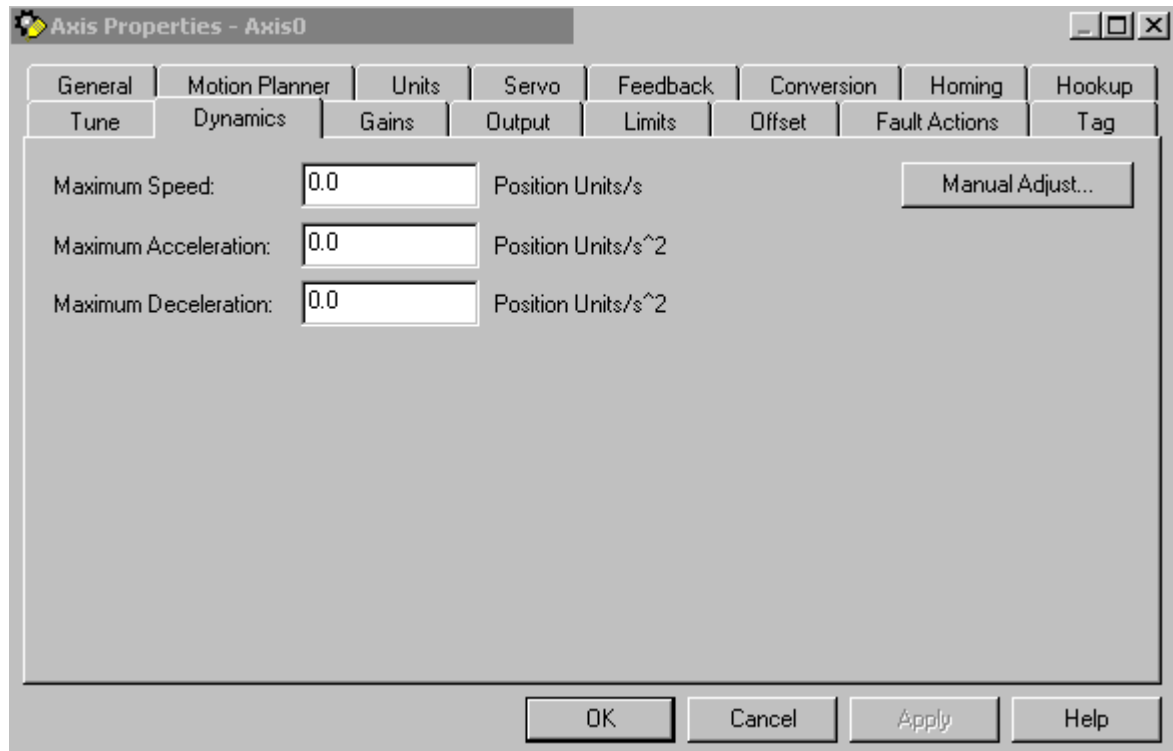
Executing a Tune operation automatically saves all changes to axis properties.

### ATTENTION



This tuning procedure may cause axis motion with the controller in program mode. Unexpected motion may cause damage to the equipment, personal injury, or death.

**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`.



**Figure 6.24 Axis Properties - Dynamics Tab for Axis\_Servo**

### 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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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 “

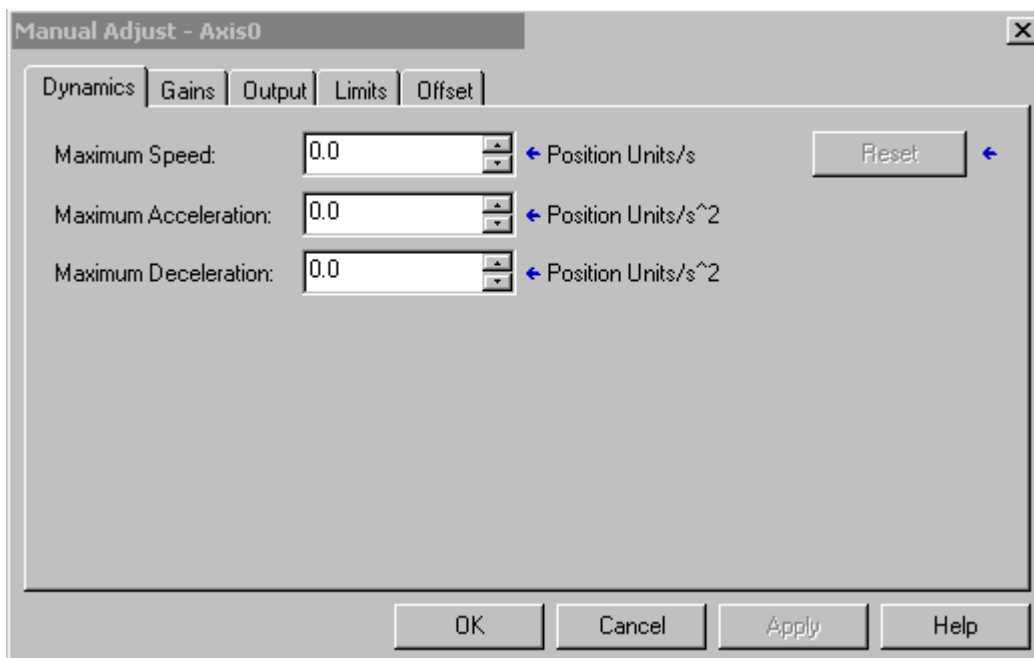
head-room” for the axis to operate at all times within the acceleration limits of the drive and motor. Any change in value, caused by manually changing the spin control, is instantaneously sent to the controller.

### **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.

## Manual Adjust

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



**Figure 6.25 Axis Properties - Dynamics Tab Manual Adjust Screen for Axis\_Servo**

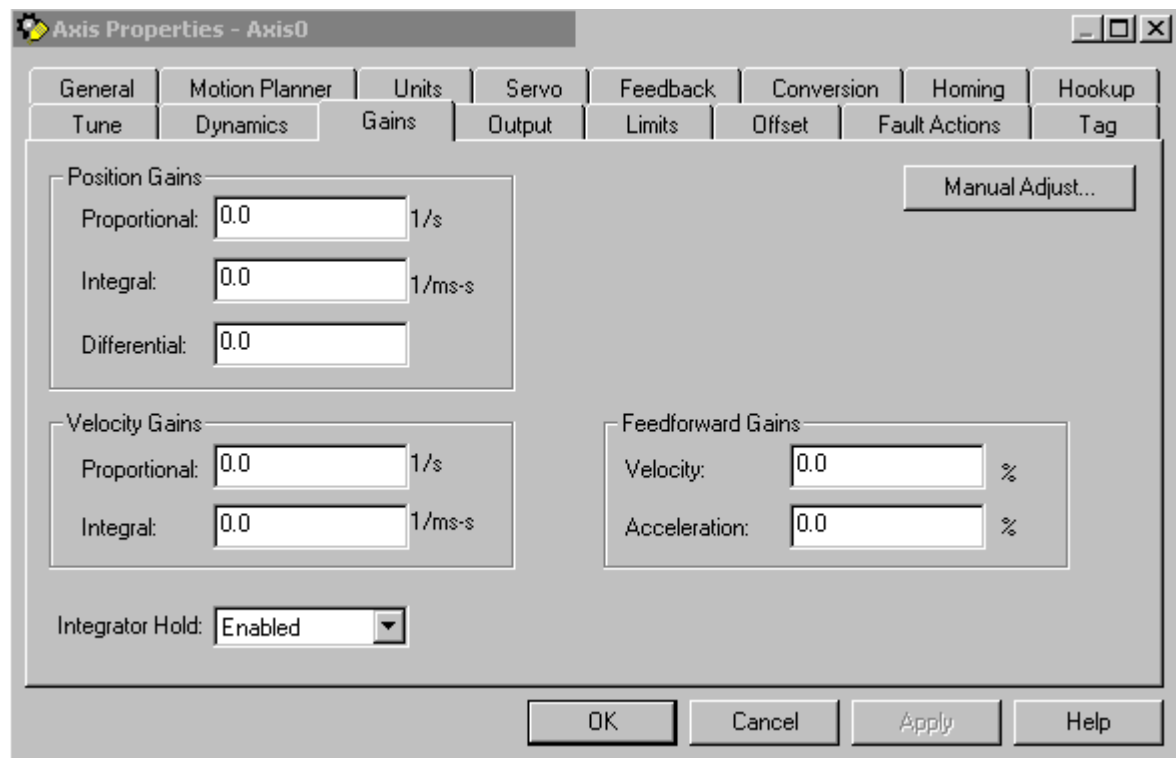
**Note:** The Manual Adjust button is disabled when RSLogix 5000 is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.



**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.



**Figure 6.26 Axis Properties - Gains Tab for Axis\_Servo**

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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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 determines 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:

$$\text{Pos P Gain} = 16.667 * \text{Desired Loop Gain (IPM/mil)}$$

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

$$\text{Pos P Gain} = \text{Bandwidth (Hertz)} * 6.28$$

The typical value for the Position Proportional Gain is  $\sim 100 \text{ Sec}^{-1}$ .

## 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:

$$\text{Pos I Gain} = .025 * 0.001 \text{ Sec/mSec} * (\text{Pos P Gain})^2$$

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

### Proportional (Velocity) Gain

**Note:** This parameter is enabled for all loop types except Torque loop.

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.

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.

$$\text{Velocity P Gain} = \text{Bandwidth (Hertz)} / 6.28$$

The typical value for the Velocity Proportional Gain is 250.

### Integral (Velocity) Gain

**Note:** This parameter is enabled for all loop types except Torque loop.

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  $\sim 15 \text{ mSec}^{-2}$ .

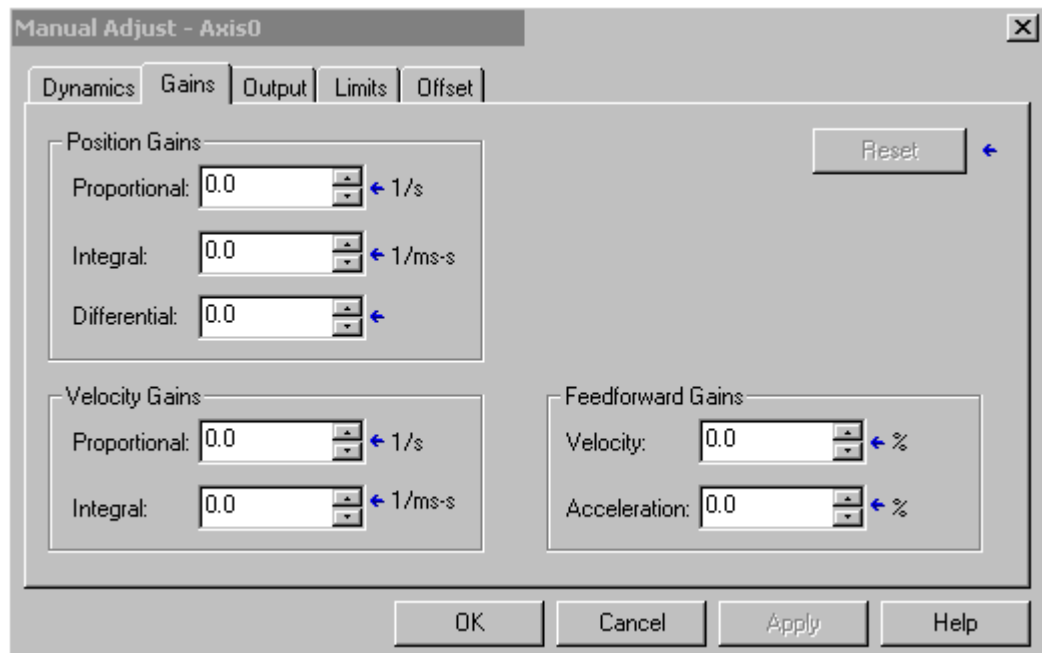
## Integrator Hold

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 Adjust

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



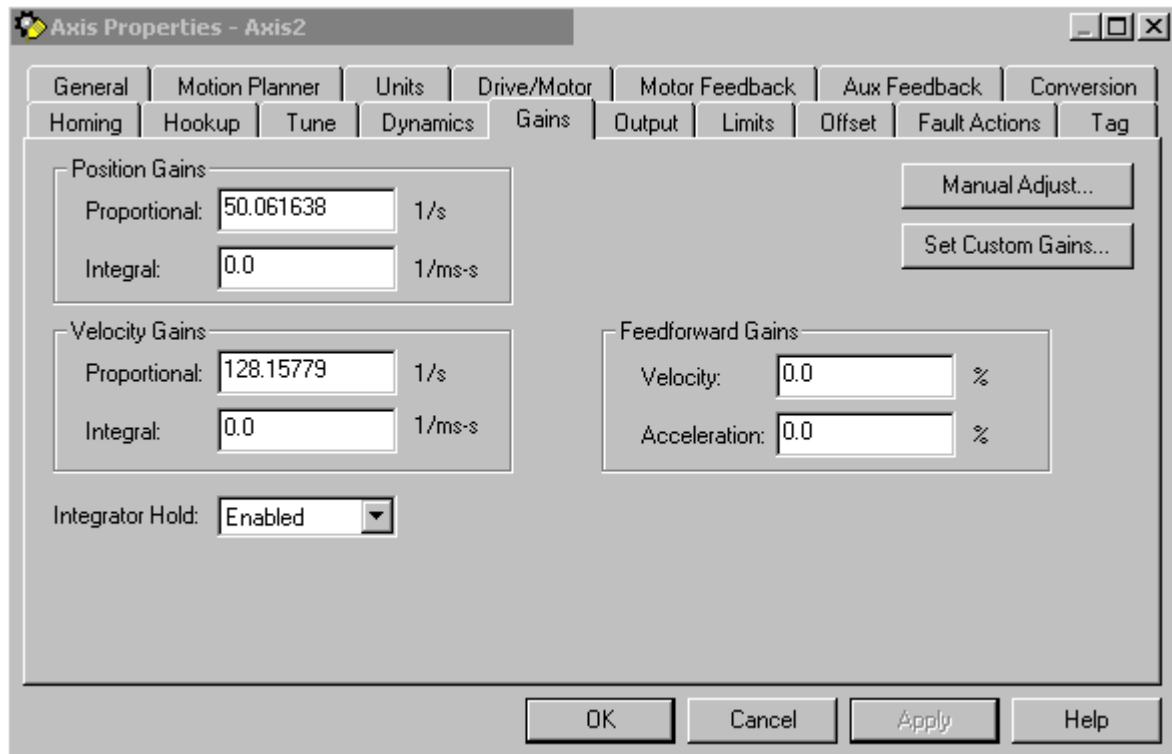
**Figure 6.27 Axis Properties - Gains Tab Manual Adjust Screen for Axis\_Servo**

**Note:** The Manual Adjust 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.



**Figure 6.28 Axis Properties - Gains Tab for 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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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:

$$\text{Pos P Gain} = 16.667 * \text{Desired Loop Gain (IPM/mil)}$$

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

$$\text{Pos P Gain} = \text{Bandwidth (Hertz)} * 6.28$$

The typical value for the Position Proportional Gain is  $\sim 100 \text{ Sec}^{-1}$ .

### 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:

$$\text{Pos I Gain} = .025 * 0.001 \text{ Sec/mSec} * (\text{Pos P Gain})^2$$

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

### 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.

$$\text{Vel P Gain} = \text{Bandwidth (Hertz)} / 6.28$$

The typical value for the Velocity Proportional Gain is ~250 mSec<sup>-1</sup>.

### 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:

$$\text{Vel I Gain} = 0.25 * 0.001 \text{ Sec/mSec} * (\text{Vel P Gain})^2$$

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

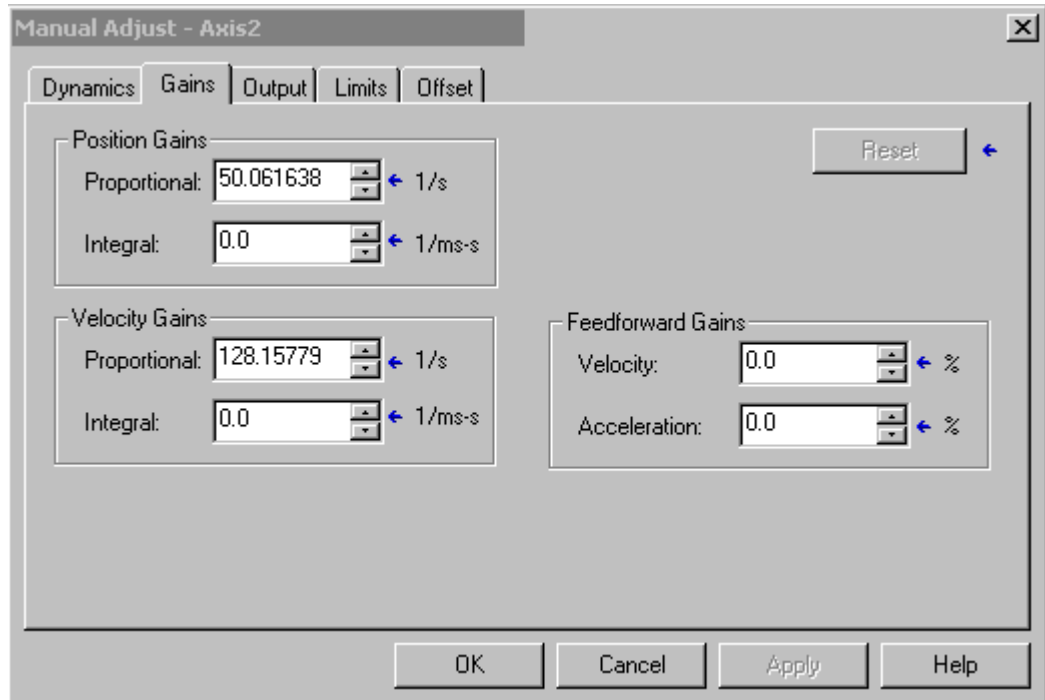
### **Integrator Hold**

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.

## Manual Adjust

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

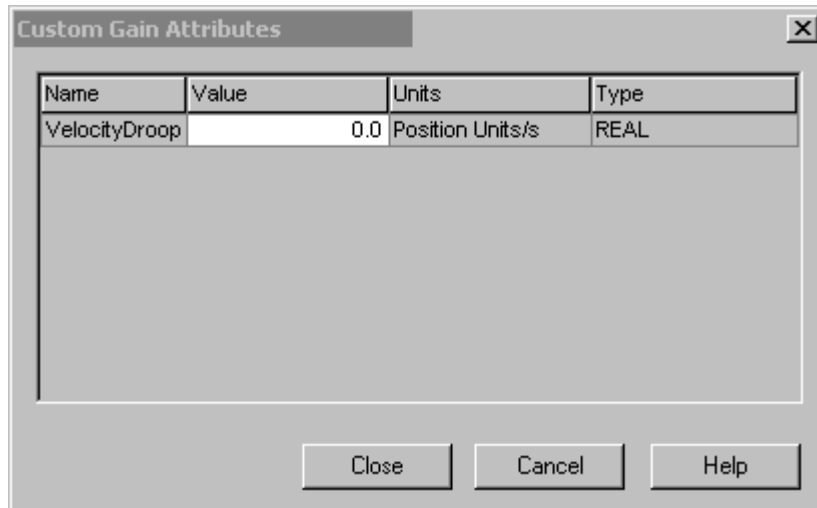


**Figure 6.29 Axis Properties - Gains Tab Manual Adjust Screen for Axis\_Servo\_Drive**

**Note:** The Manual Adjust 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.

### Set Custom Gains

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



**Figure 6.30 Set Custom Gains Dialog from Gains Tab for AXIS\_SERVO\_DRIVE**

At this dialog box you can edit the VelocityDroop attribute.

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.

### Attribute

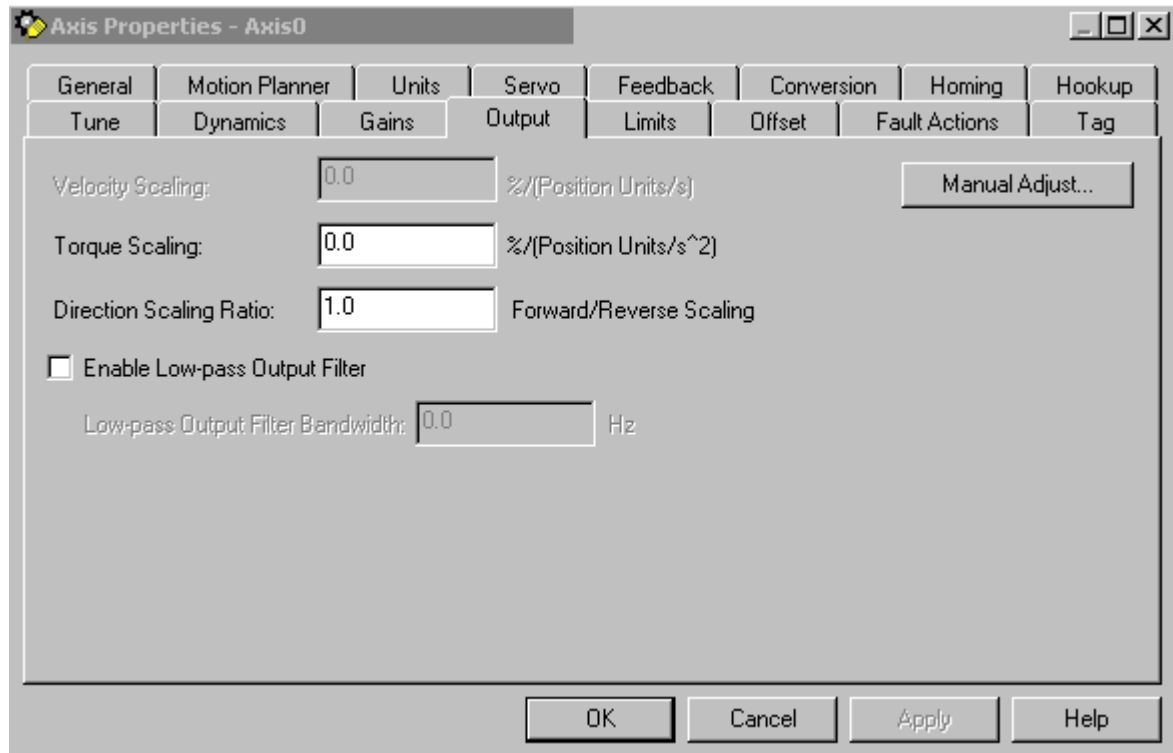
The following attribute value can be monitored and edited in this dialog box.

Attribute	Description
VelocityDroop	<p>This 32-bit unsigned attribute – also referred to as "static gain" – acts as a very slow discharge of the velocity loop integrator. VelocityDroop may be used as a component of an external position loop system where setting this parameter to a higher, non-zero value eliminates servo hunting due to load/stick friction effects. This parameter only has effect if VelocityIntegralGain is not zero. Its value ranges from 0 to <math>2.14748 \times 10^{12}</math>.</p> <p><b>Note:</b> This value is not applicable for Ultra3000 drives.</p>

**Output Tab - AXIS\_SERVO** 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.



**Figure 6.31 Axis Properties - Output Tab for Axis\_Servo**

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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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:

$$\text{Velocity Scaling} = 100\% / (\text{Speed @ } 100\%)$$

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:

$$\text{Velocity Scaling} = 100\% / (83.3 \text{ RPS}) = 1.2\% / \text{Revs Per Second}$$

## 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:

$$\text{Torque Scaling} = 100\% \text{ Rated Torque} / (\text{Acceleration @ } 100\% \text{ Rated Torque})$$

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/Sec<sup>2</sup>, the Torque Scaling attribute value would be calculated as shown below:

$$\text{Torque Scaling} = 100\% \text{ Rated} / (3000 \text{ RPS}^2) = 0.0333\% \text{ Rated} / \text{Revs Per Second}^2$$

**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.

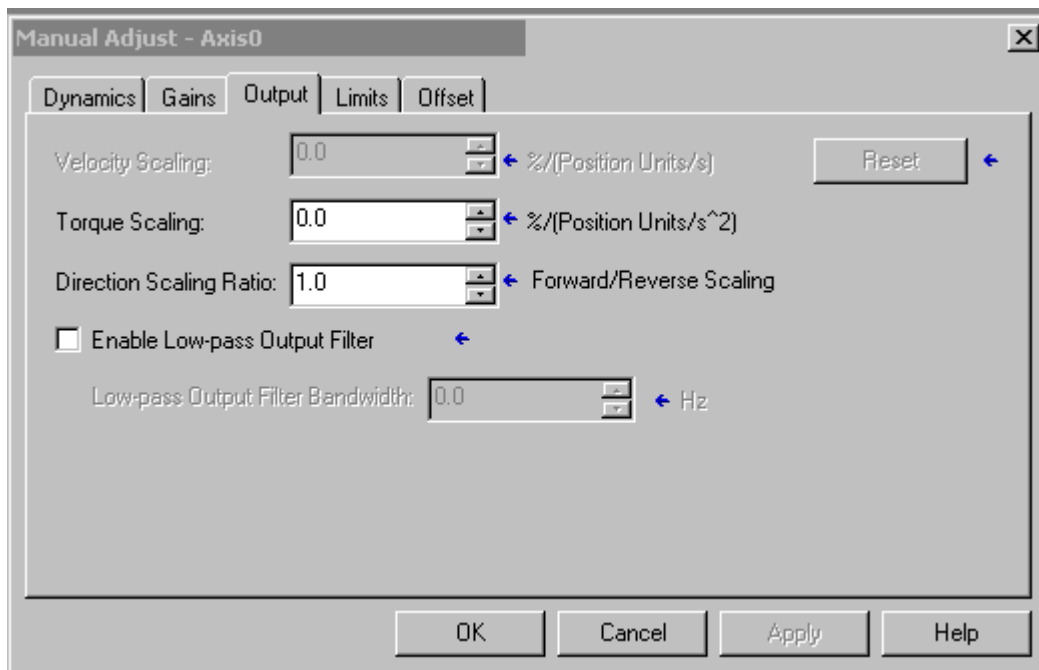
## Low-pass Output Filter Bandwidth

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 is 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 Adjust

Click on this button to access the Output tab of the Manual Adjust dialog for online editing.



**Figure 6.32 Axis Properties - Output Tab Manual Adjust Screen for Axis\_Servo**

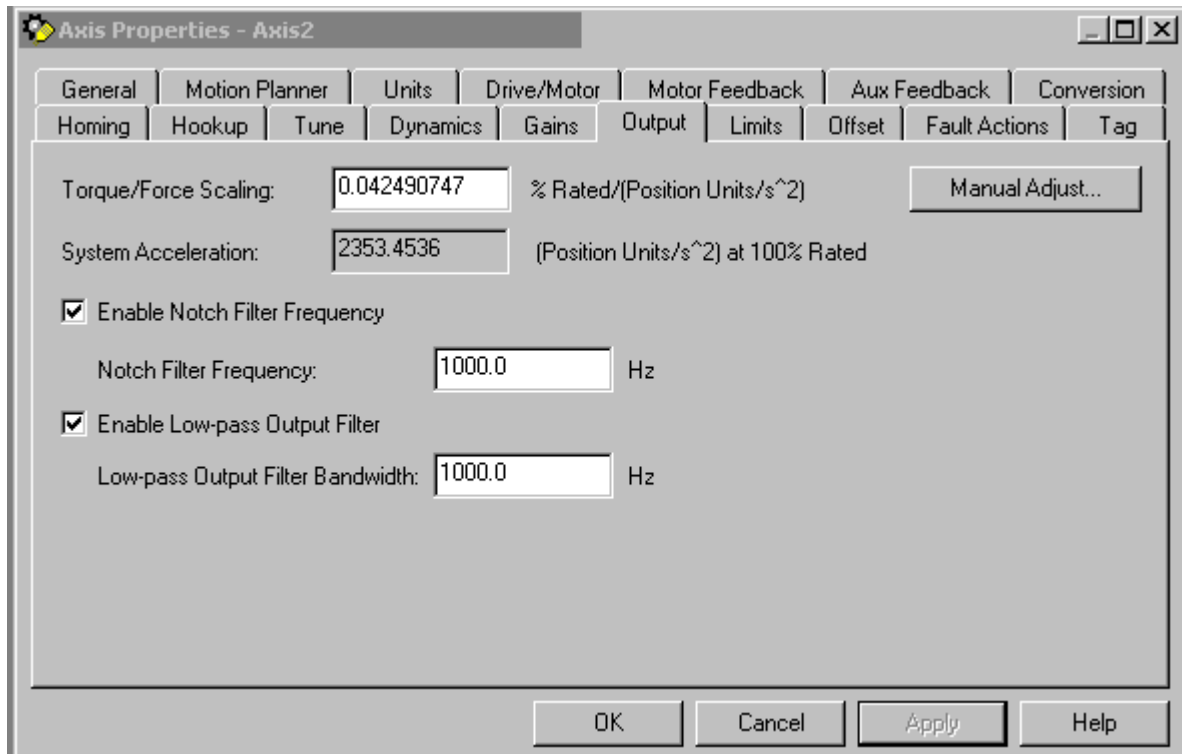
**Note:** The Manual Adjust button is disabled when RSLogix 5000 is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.



**Output Tab (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.



**Figure 6.33 Axis Properties - Output Tab for Axis\_Servo\_Drive**

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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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:

$$\text{Torque Scaling} = 100\% \text{ Rated Torque} / (\text{Acceleration @ } 100\% \text{ Rated Torque})$$

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/Sec<sup>2</sup>, the Torque Scaling attribute value would be calculated as shown below:

$$\text{Torque Scaling} = 100\% \text{ Rated} / (3000 \text{ RPS}^2) = 0.0333\% \text{ Rated} / \text{Revs Per Second}^2$$

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

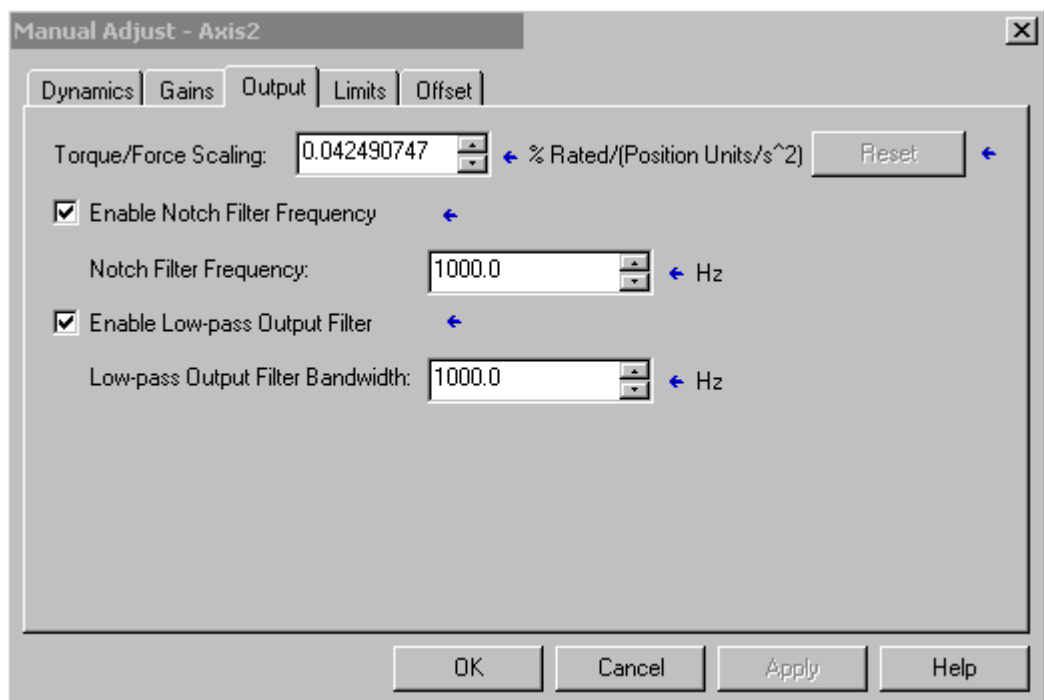
## Low-pass Output Filter Bandwidth

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 is 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 Adjust

Click on this button to open the Output tab of the Manual Adjust dialog for online editing of Torque/Force Scaling, the Notch Filter Frequency, and the Low-pass Output Filter parameters.



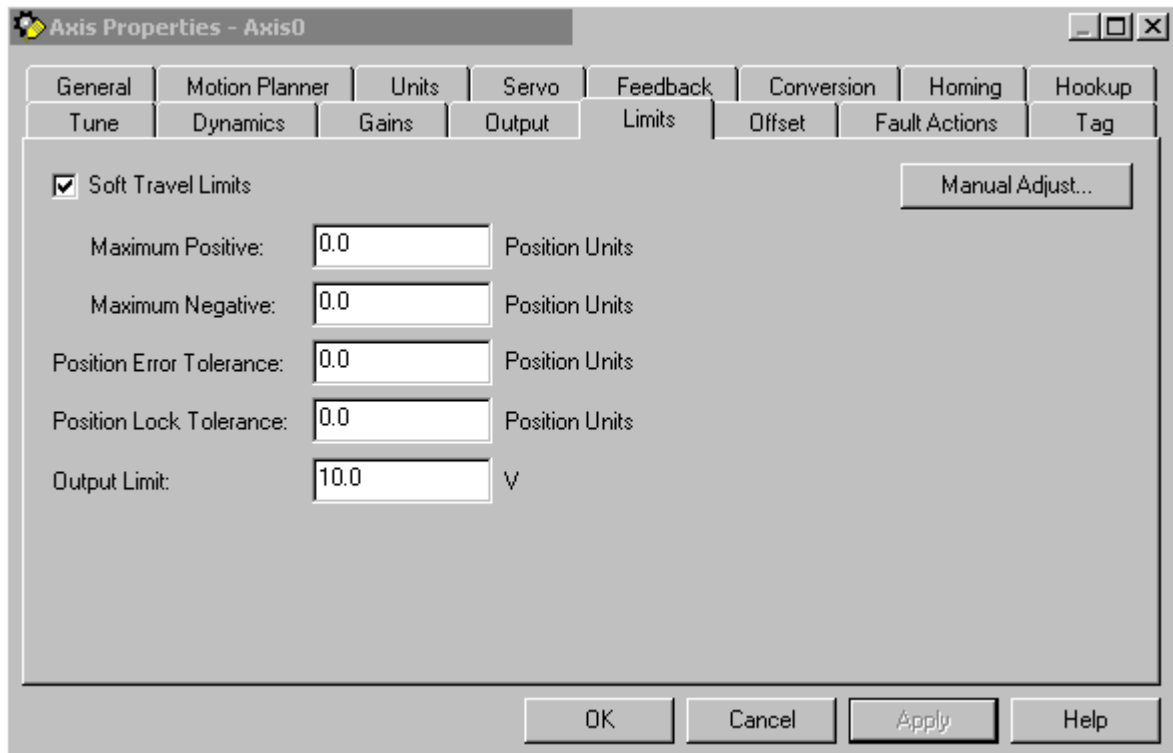
**Figure 6.34 Axis Properties - Output Tab for Axis\_Servo\_Drive**

**Note:** The Manual Adjust 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.



**Figure 6.35 Axis Properties - Limits Tab for Axis\_Servo**

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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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 Error Tolerance**

Specifies how much position error the servo tolerates 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 is 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 accepts 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**

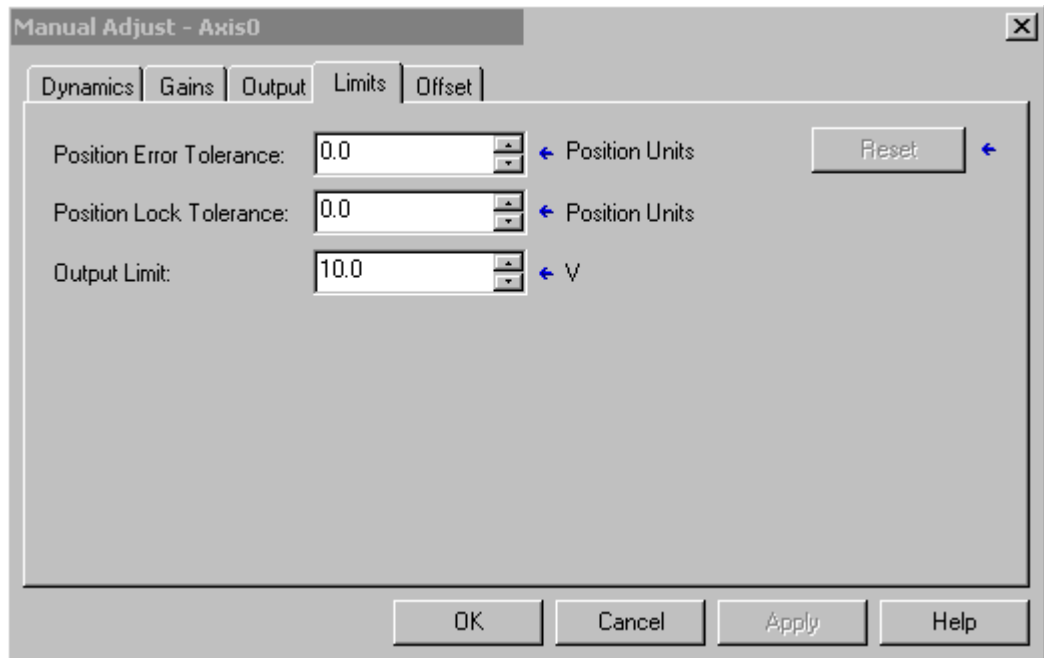
Provides a method of limiting the maximum servo output voltage of a physical axis to a specified level. The servo output for the axis as a function of position servo error, both with and without servo output limiting, is shown below.

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 ever commands 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  $\pm 10$  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  $\pm 7.5$  Volts, set the servo output limit value to 7.5 volts.

## Manual Adjust

Click on this button to open the Limits tab of the Manual Adjust dialog for online editing of the Position Error Tolerance, Position Lock Tolerance, and Output Limit parameters.



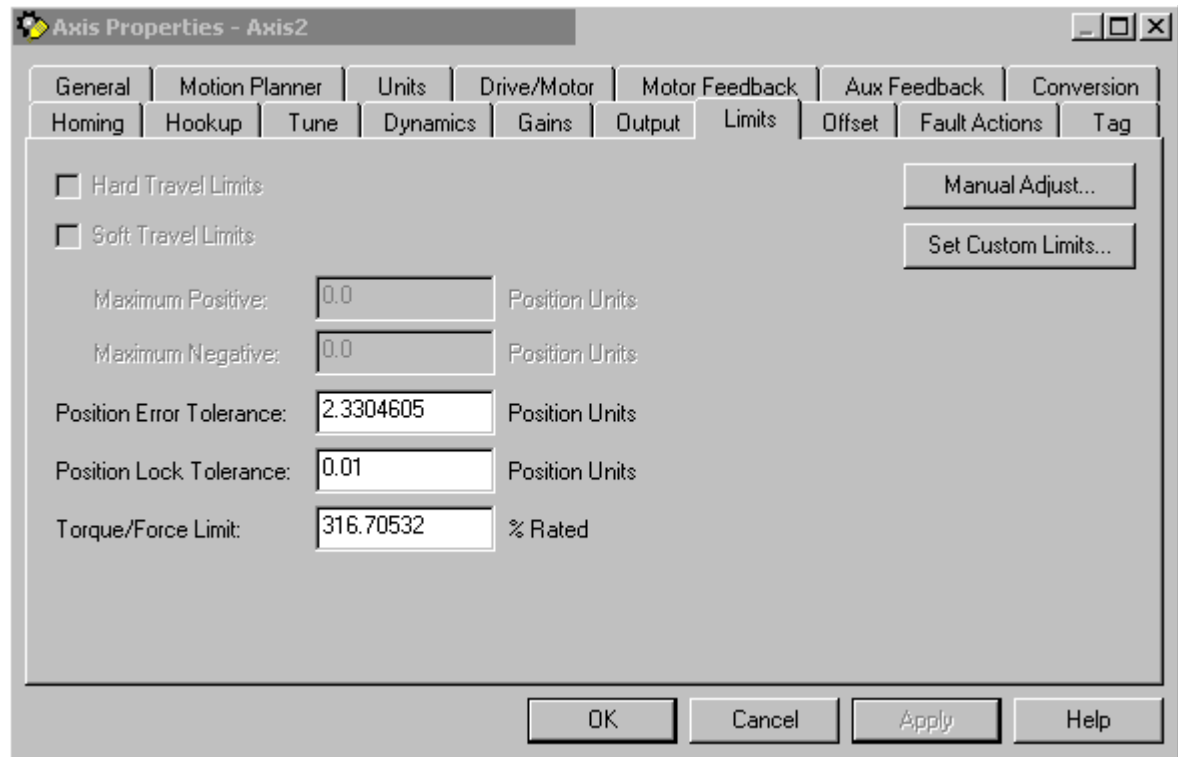
**Figure 6.36 Axis Properties - Limits Tab Manual Adjust Screen for Axis\_Servo**

**Note:** The Manual Adjust 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.



**Figure 6.37 Axis Properties - Limits Tab for Axis\_Servo\_Drive**

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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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 tolerates 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 is 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.

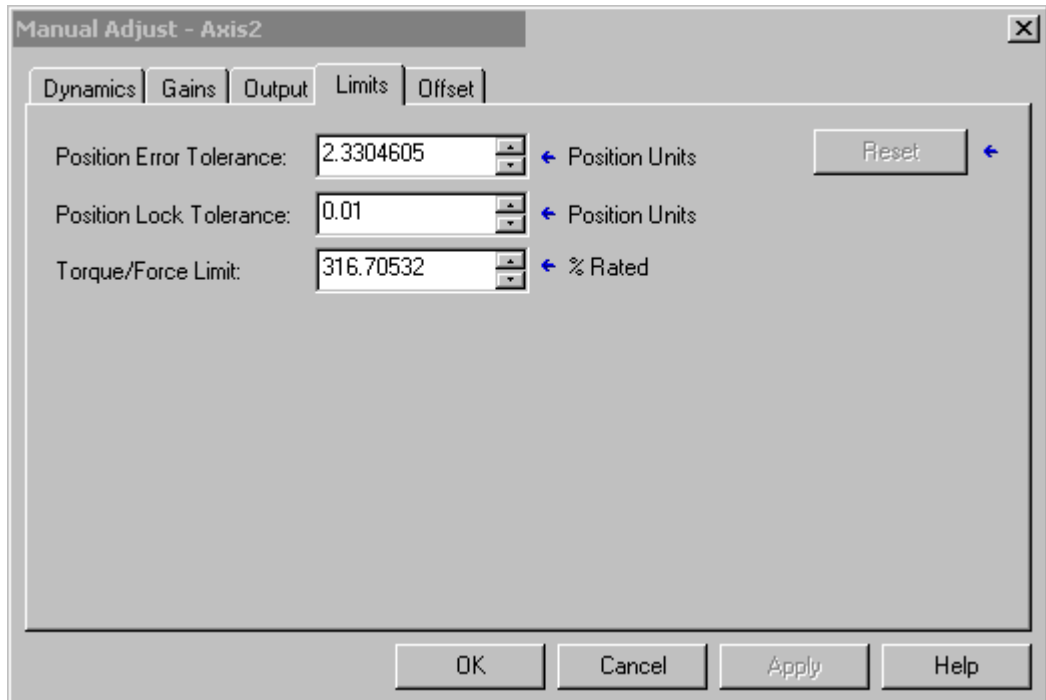
### Position Lock Tolerance

Specifies the maximum position error the servo module accepts 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:

### Manual Adjust

Click on this button to open the Limits tab of the Manual Adjust dialog for online editing of the Position Error Tolerance, Position Lock Tolerance, and Output Limit parameters.

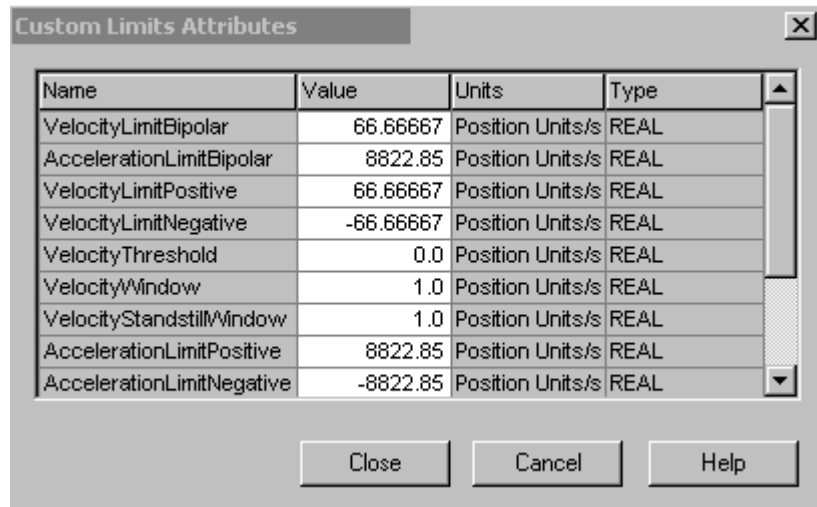


**Figure 6.38 Axis Properties - Limits Tab for Axis\_Servo\_Drive**

**Note:** The Manual Adjust 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.

### Set Custom Limits

Click this button to open the Custom Limit Attributes dialog.



**Figure 6.39 Set Custom Limits Dialog from the Limits Tab for the AXIS\_SERVO\_DRIVE**

From this dialog box you can monitor and edit the limit-related attributes.

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.

### Attributes

The following attribute values can be monitored and edited in this dialog box.

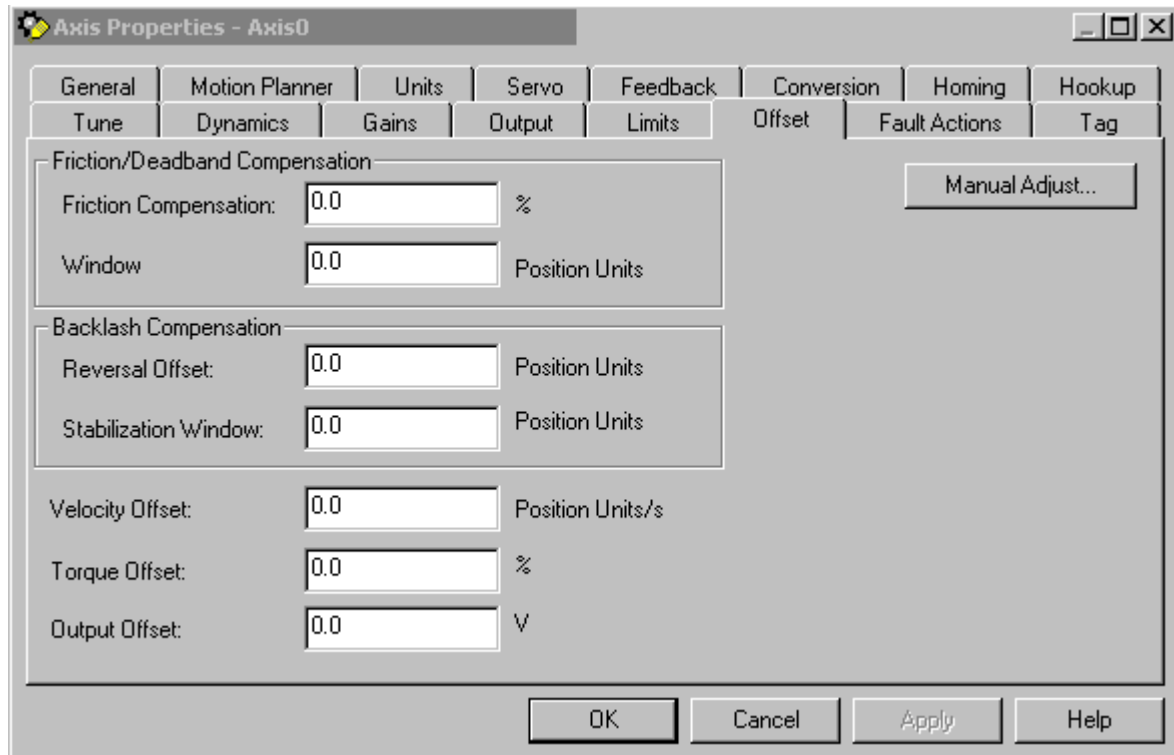
Attribute	Description
VelocityLimitBipolar	This attribute sets the velocity limit symmetrically in both directions. If the command velocity exceeds this value, VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748 \times 10^{12}$ .
AccelerationLimitBipolar	This attribute sets the acceleration and deceleration limits for the drive. If the command acceleration exceeds this value, AccelLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748 \times 10^{15}$ .
TorqueLimitBipolar	This attribute sets the torque limit symmetrically in both directions. When actual torque exceeds this value TorqueLimitStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.

Attribute	Description
VelocityLimitPositive	This attribute displays the maximum allowable velocity in the positive direction. If the velocity limit is exceeded, bit 5 ("Velocity Command Above Velocity Limit") VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748 \times 10^{12}$ .
VelocityLimitNegative	This attribute displays the maximum allowable velocity in the negative direction. If the velocity limit is exceeded, bit 5 ("Velocity Command Above Velocity Limit") VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of $-2.14748 \times 10^{12}$ to 0.
VelocityThreshold	This attribute displays the velocity threshold limit. If the motor velocity is less than this limit, VelocityThresholdStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748 \times 10^{12}$ .
VelocityWindow	This attribute displays the limits of the velocity window. If the motor's actual velocity differs from the command velocity by an amount less than this limit VelocityLockStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748 \times 10^{12}$ .
VelocityStandstillWindow	This attribute displays the velocity limit for the standstill window. If the motor velocity is less than this limit VelocityStandStillStatus of the DriveStatus bit is set. This attribute has a value range of 0 to $2.14748 \times 10^{12}$ .
AccelerationLimitPositive	This attribute limits the maximum acceleration ability of the drive to the programmed value. If the command acceleration exceeds this value, AccelLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748 \times 10^{15}$ .
AccelerationLimitNegative	This attribute limits the maximum acceleration ability of the drive to the programmed value. If the command acceleration exceeds this value, the AccelLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of $-2.14748 \times 10^{15}$ to 0.
TorqueLimitPositive	This attribute displays the maximum torque in the positive direction. If the torque limit is exceeded, the TorqueLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.
TorqueLimitNegative	This attribute displays the maximum torque in the negative direction. If the torque limit is exceeded, the TorqueLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of -1000 to 0.
TorqueThreshold	This attribute displays the torque threshold. If this limit is exceeded, the TorqueThreshold bit of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.

**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.



**Figure 6.40 Axis Properties - Offset Tab for Axis\_Servo**

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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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 can 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.

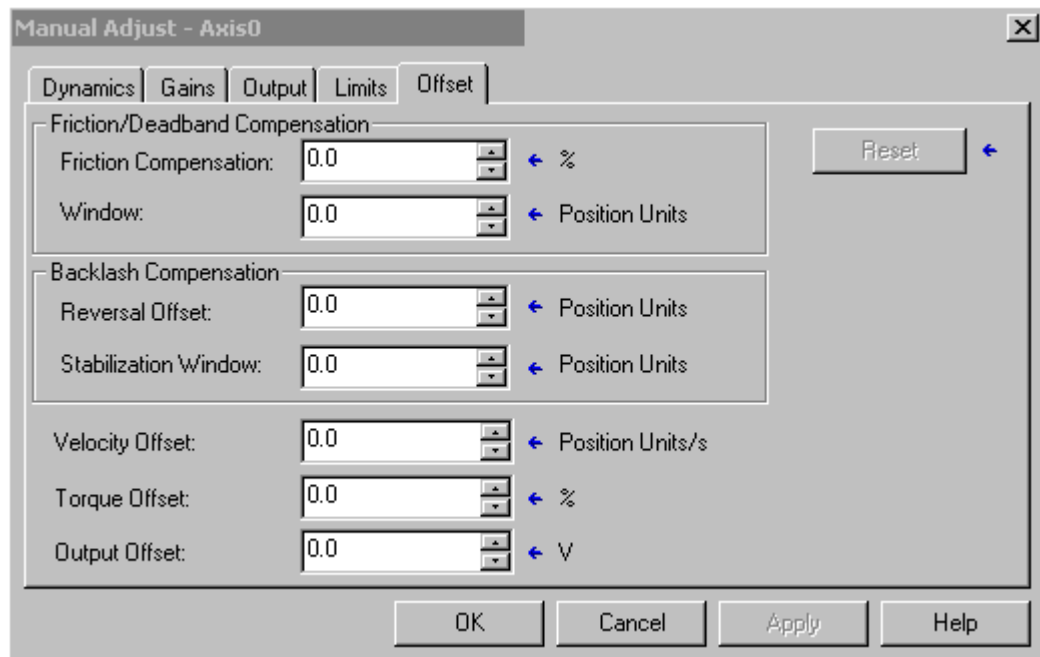
## Output Offset

Corrects the problem of axis “drift”, by adding a fixed voltage value (not to exceed  $\pm 10$  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 cause problems with 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 Adjust

Click on this button to open the Offset tab of the Manual Adjust dialog for online editing of the Friction/Deadband Compensation, Backlash Compensation, Velocity Offset, Torque Offset, and Output Offset parameters.



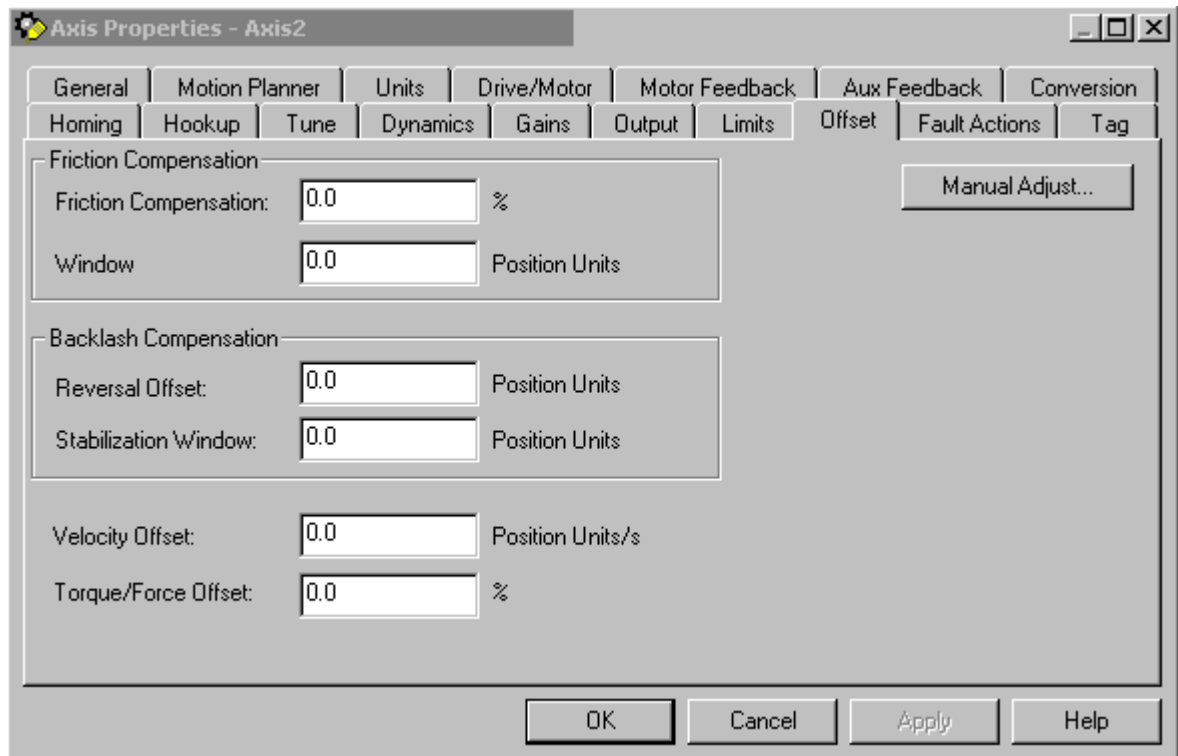
**Figure 6.41 Axis Properties - Offset Tab Manual Adjust Screen for Axis\_Servo**

**Note:** The Manual Adjust 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.



**Figure 6.42 Axis Properties - Offset Tab for Axis\_Servo\_Drive**

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 Adjust dialog: click on the Manual Adjust button to open the Manual Adjust 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 can cause the axis to "dither", i.e. move rapidly back and forth about the commanded position.

### **Velocity Offset**

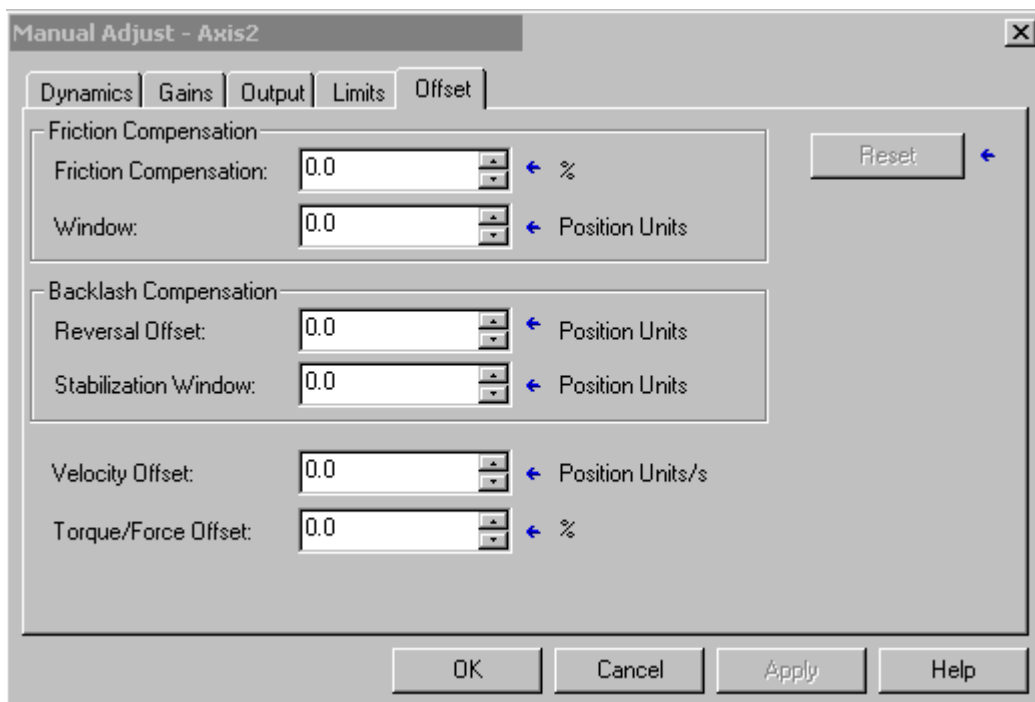
Provides a dynamic velocity correction to the output of the position servo loop, in position units per second. Because the position servo loop output value is updated synchronously every Coarse Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block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

Click on this button to open the Offset tab of the Manual Adjust dialog for online editing of the Friction/Deadband Compensation, Backlash Compensation, Velocity Offset, Torque Offset, and Output Offset parameters.



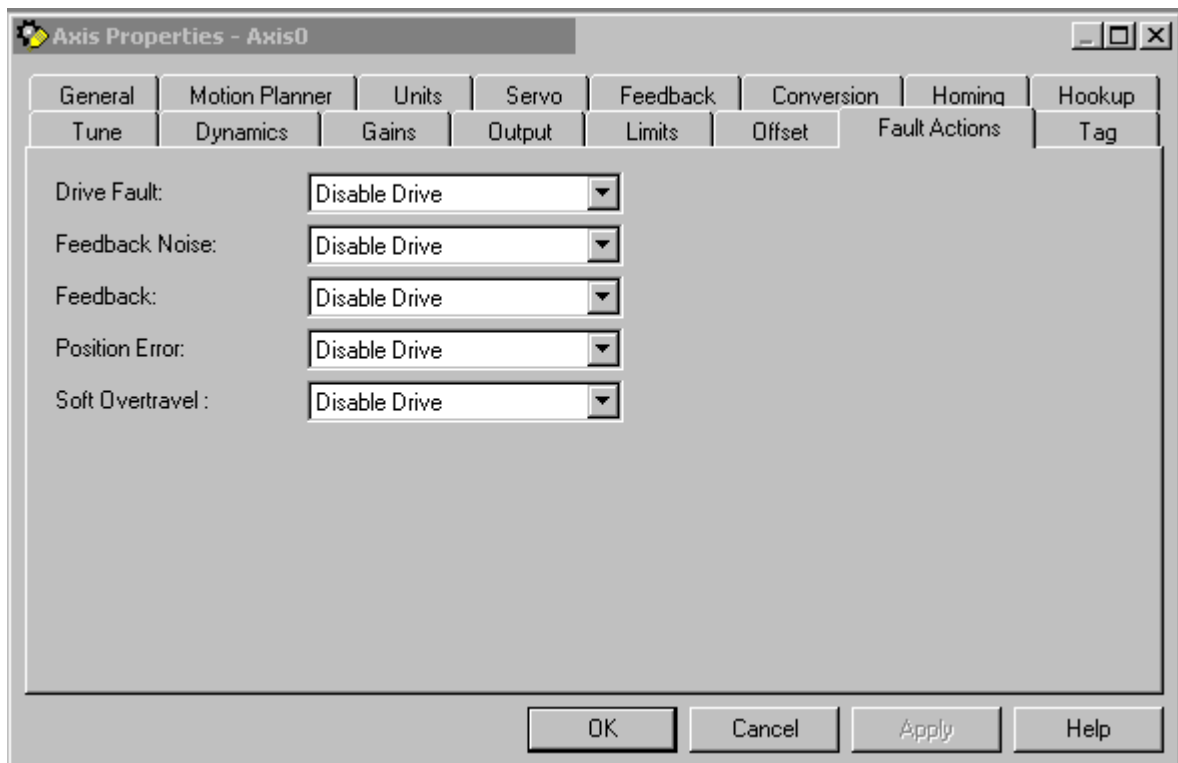
**Figure 6.43 Axis Properties - Offset Tab Manual Adjust Screen for Axis\_Servo\_Drive**

**Note:** The Manual Adjust 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 are 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.



**Figure 6.44 Axis Properties - Fault Actions Tab for 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. 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.

---

**ATTENTION**

Selecting the wrong fault action for your application can cause a dangerous condition resulting in unexpected motion, damage to the equipment, and physical injury or death. 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.

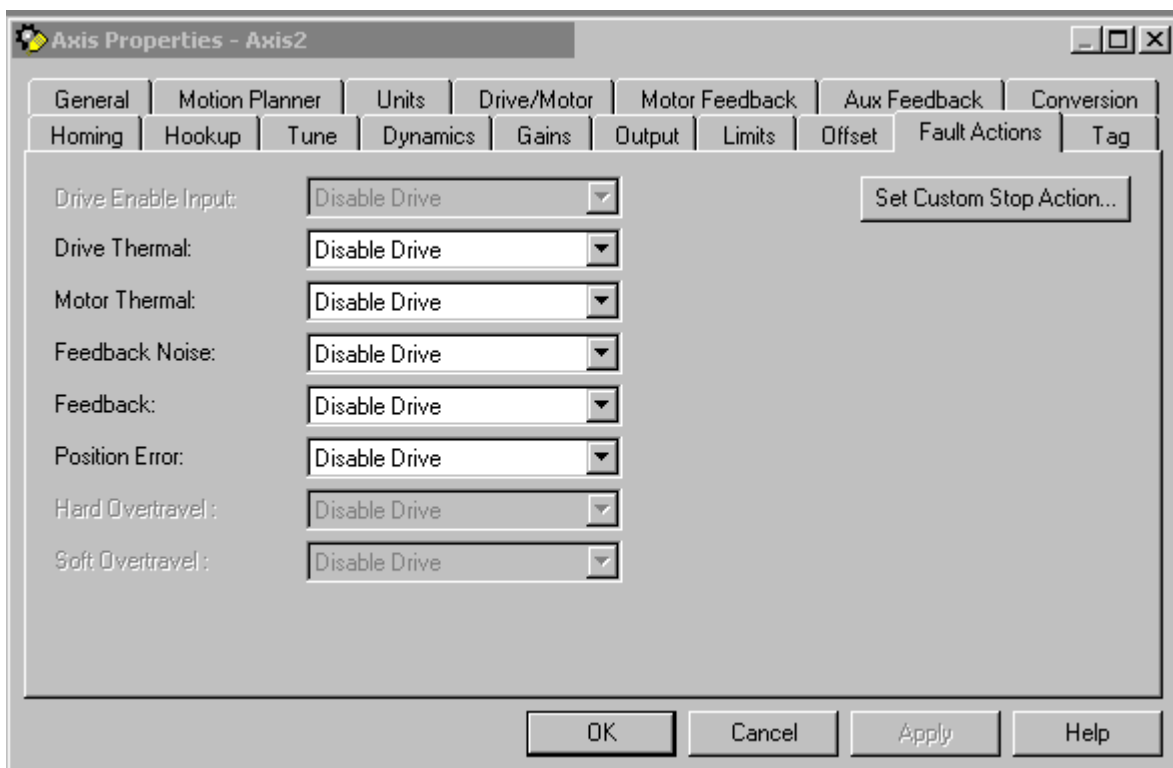
**Soft Overtravel**

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 are 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.



**Figure 6.45 Axis Properties - Fault Action Tab for 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. 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, it brings the axis to a stop by applying the Stopping Torque for up to the Stopping Time Limit. During this period the servo is active but no longer tracking the command reference from logix. Once the axis is stopped (or the stopping limit is exceeded) the servo and power structure are disabled.
- **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.

---

**ATTENTION**

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.

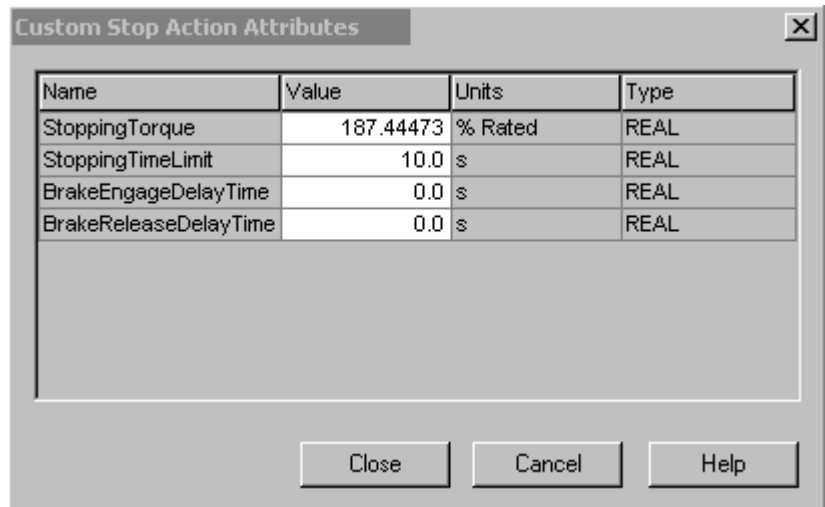
### **Soft Overtravel**

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.



**Figure 6.46 Set Custom Stop Action Dialog From Fault Actions Tab for the AXIS\_SERVO\_DRIVE**

Use this dialog to monitor and edit the Stop Action-related attributes.

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.

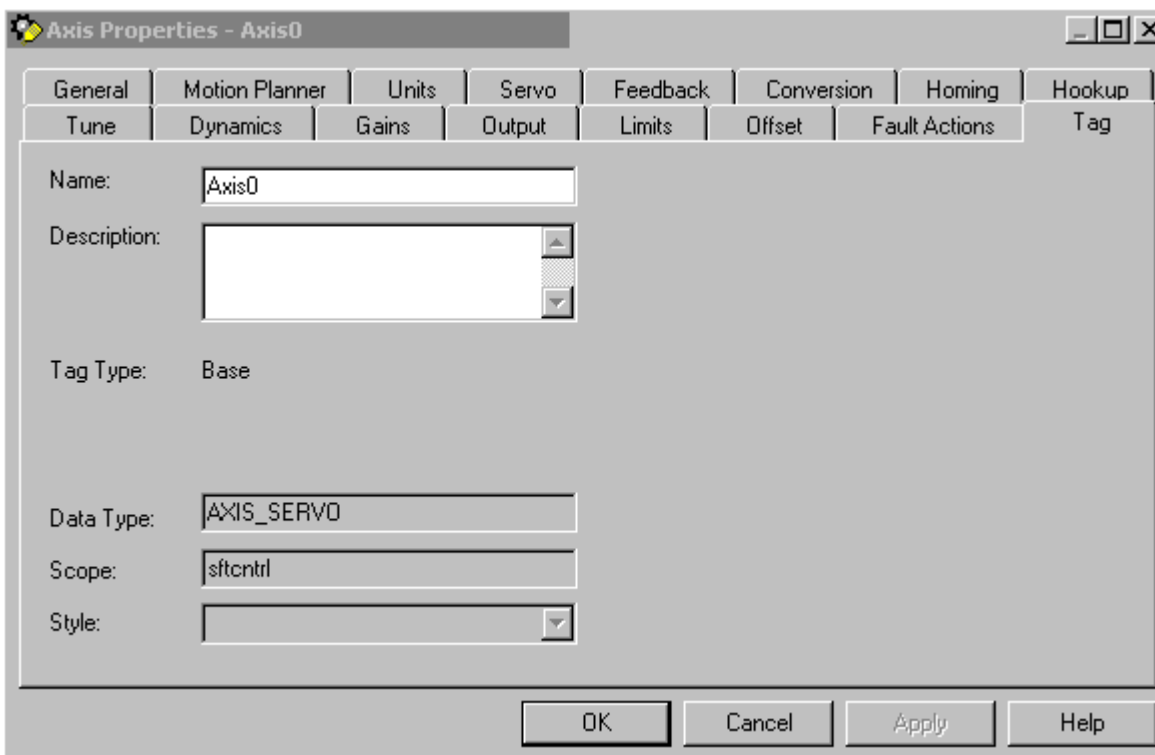
### Attributes

The following attribute, or parameter, values can be monitored and edited in this dialog box.

Attribute	Description
StoppingTorque	This attribute displays the amount of torque available to stop the motor. This attribute has a value range of 0 to 1000.
StoppingTimeLimit	This attribute displays the maximum amount of time that the drive amplifier remains enabled while trying to stop. It is useful for very slow velocity rate change settings. This attribute has a value range of 0 to 6553.5.
BrakeEngageDelayTime	When servo axis is disabled and the drive decelerates to a minimum speed, the drive maintains torque until this time has elapsed. This time allows the motor's brake to be set. This attribute has a value range of 0 to 6.5535.

Attribute	Description
BrakeReleaseDelayTime	When the servo axis is enabled , the drive activates the torque to the motor but ignores the command values from the Logix controller until this time has elapsed. This time allows the motor’s brake to release. This attribute has a value of 0 to 6.5535.

**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.



**Figure 6.47 Axis Properties - Tag Tab**

**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.

### **Data Type**

Displays the axis data type of 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.

## **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.



# Creating & Configuring Your Coordinate System Tag

## Introduction

The Coordinate System tag is used to set the attribute values to be used by the Multi-Axis Coordinated Motion instructions in your motion applications. The Coordinate System tag must exist before you can run any of the Multi-Axis Coordinated Motion instructions. This is where you introduce the COORDINATE\_SYSTEM data type, associate the Coordinate System to a Motion Group, associate the axes to the Coordinate System, set the dimension, and define the values later used by the operands of the Multi-Axis Motion Instructions. The values for Coordination Units, Maximum Speed, Maximum Acceleration, Maximum Deceleration, Actual Position Tolerance, and Command Position Tolerance are all defined by the information included when the Coordinate System tag is configured. This chapter describes how to name, configure, and edit your Coordinate System tag.

## Creating a Coordinate System

Creating a coordinate system adds it to your application. There are four ways in which you can initiate the creation of a coordinate system. The first way is to go to the **File** pull-down menu, select **New Component**, and then select **Tag**.

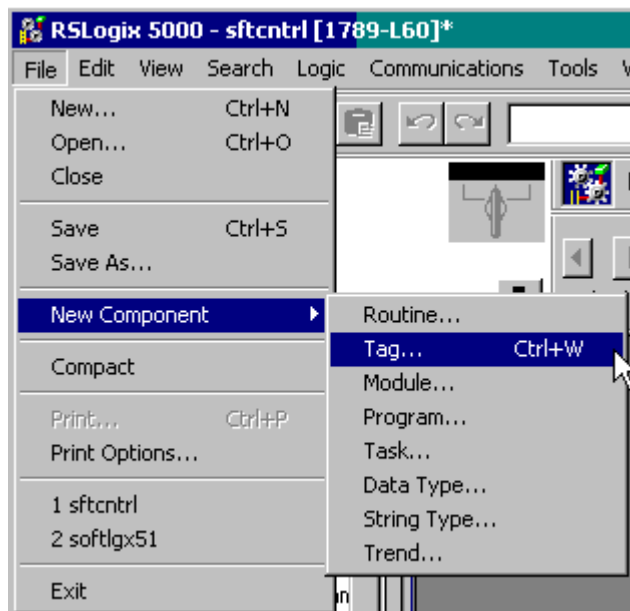


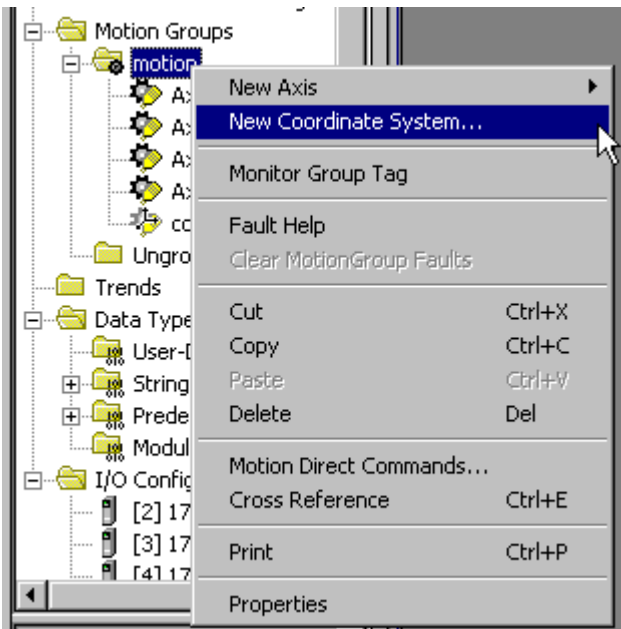
Figure 7.1 File Menu to New Component to Tag

The second way is to go the Controller organizer and right click on Controller Tags and select New Tag from the pop-up menu.



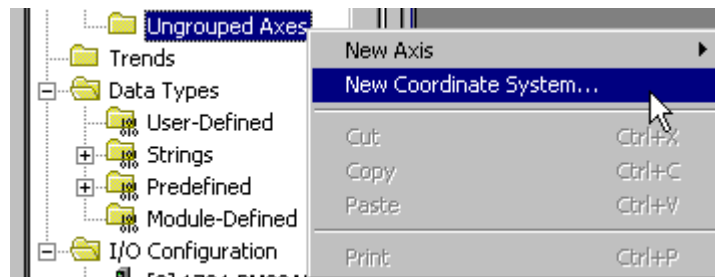
**Figure 7.2 Accessing the New Tag Menu From The Controller Tag**

The third way also employs the right mouse click method. Right click on the Motion Group in the Controller Organizer and select New Coordinate System from the menu.



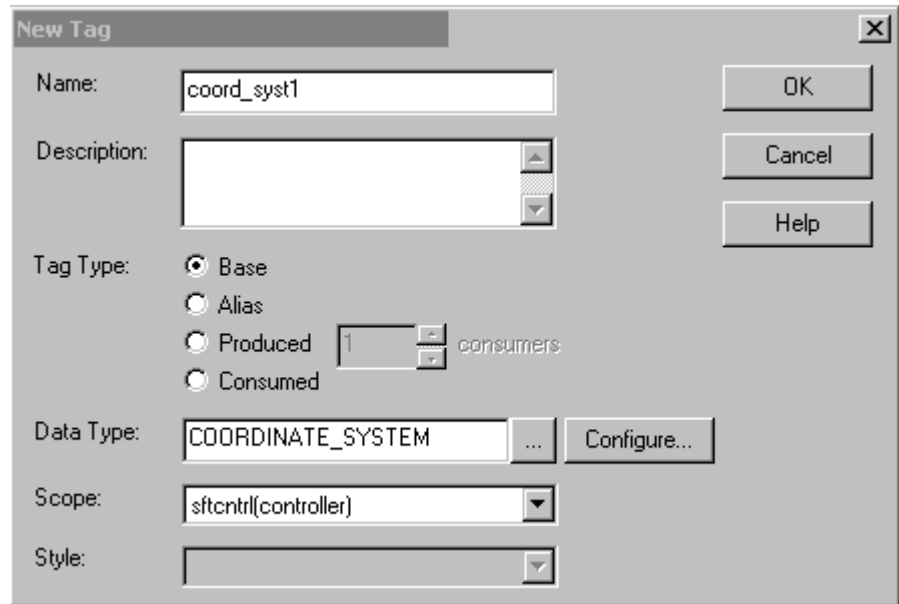
**Figure 7.3 Creating a New Coordinate System From Motion Group**

The final way to create a new coordinate system tag is by right clicking on Ungrouped Axes and selecting New Coordinate System from the menu.



**Figure 7.4 Creating a Coordinate System From Ungrouped Axes**

Regardless of the method you use the New Tag window appears.



**Figure 7.5 New Tag Dialog**

The method used to access the New Tag Dialog determines how much of the dialog is already filled in when the window displays. If you accessed the New Tag window from either Motion Group or Ungrouped Axes, the Data Type fills in automatically.

### Entering Tag Information

A tag allows you to allocate and reference data stored in the controller. A tag can be a single element, array, or a structure. With `COORDINATE_SYSTEM` selected as the Data Type, there are only two 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 coordinate system tag.

Use this dialog to create new tags.

You can create base tags and alias tags while the controller is either online or offline, as long as the new tag is verified. However, tags created online can only be created in the Ungrouped Axes folder and cannot be used for motion at that time.

### New Tag Parameters

The following parameters appear on the New Tag dialog when you are creating a base tag or an alias tag.

Make entries in the following fields.

Field	Entry
Name	Type a name for the coordinate system tag. The name can have a maximum of 40 characters containing letters, numbers and underscores (_).
Description	Type a description for your motion axis for annotation purposes. This field is optional.
Tag Type	Click on the radio button for the type of tag to create. The only legal choices are Tag and Alias. Selecting either Produced or Consumed generates an error when the OK button is pressed.
Alias For	This field only displays when Alias is selected for Tag Type. Enter the name of the related Base Tag.
Data type	Enter COORDINATE_SYSTEM.
Scope	A Coordinate System tag can only be created at the controller scope.

### Name

Enter a relevant name for the new tag. The name can be up to 40 characters and can be composed of letters, numbers, or underscores (\_).

### Description

Enter a description of the tag. This is an optional field and is used for annotating the tag.

### Tag Type

For a Coordinate System you may choose either Base or Alias for the Tag Type. Click on the appropriate radio button for 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.

### Alias For:

If you selected Alias as the Tag Type the **Alias For:** field displays. Enter the name of the associated Base Tag.



## Data Type

In the Data Type field select COORDINATE\_SYSTEM if you entered from either method that did not fill this field automatically.

## Scope

Enter the Scope for the tag. A Coordinated System Tag can only be Controller Scope.

## Style

The Style parameter is not activated. No entry for this field is possible.

After the information for the tag is entered, you have two options. You can either press the OK button to create the tag or you can press the Configure Button located next to the Data Type field to use the Wizard screens to enter the values for the Coordinate System Tag.

Pressing the OK button, creates the tag and automatically places it in the Ungrouped Axes folder or the Motion Group if the tag was initiated from the Motion Group menu.

Pressing the Configure button next to the Data Type field invokes the Coordinate System Tag Wizard to let you continue to configure the Coordinate System tag.

## Coordinate System Wizard Screens

The Coordinate System Wizard screens walk you through the process of configuring a Coordinate System. These are the same screens that appear when you access Coordinate System Properties but instead of appearing as tabbed screens they advance you through the process by individual screens. At the bottom of each screen are a series of buttons. To advance to the next screen click on the Next button and the information you entered is saved and you advance to the next wizard screen. To end your progression through the Wizard screens click on the Finish button. The information entered to this point is saved and the Coordinate System is stored in the Controller Organizer under either the Ungrouped Axes folder or the Motion Group (if a motion group has been associated with the coordinate system).

It is not necessary to use the Wizard screens to configure your Coordinate System. Once it has been created, you can access the Coordinate System Properties screen and enter the information for the Coordinate System. See the section entitled “Editing Coordinate System Properties” later in this manual for detailed information about entering configuration information.

### **General Wizard Screen**

The General screen lets you associate the tag to a Motion Group, enter the Coordinate System Type, select the Dimension for the tag (i.e. the number of associated axes), enter the associated axis information, and select whether or not to update Actual Position values of the Coordinate System automatically during operation. This screen has the same fields as the General Tab found under Coordinate System Properties.

### **Units Wizard Screen**

The Units screen is where you determine the units that define the coordinate system. At this screen you define the Coordination Units and the Conversion Ratios. This screen has the same fields as the Units Tab found under Coordinate System Properties.

### **Dynamics Wizard Screen**

The Dynamics screen is for entering the Vector values used for Maximum Speed, Maximum Acceleration, and Maximum Deceleration. It is also used for entering the Actual and Command Position Tolerance values. This screen has the same fields as the Dynamics Tab found under Coordinate System Properties.

### **Manual Adjust Button**

The Manual Adjust button is inactive when creating a Coordinate System tag via the Wizard screens. It is active on the Dynamics Tab of the Coordinate System Properties screen. It is described in detail in the “Editing Coordinate System Properties” later in this chapter.

### **Tag Wizard Screen**

The Tag screen lets you rename your Tag, edit your description and review the Tag Type, Data Type and Scope information.

The only fields that are editable on the Tag screen are the Name and Description fields. These are the same fields as on the New Tag screen and the Coordinate System Properties Tag Tab.

## Editing Coordinate System Properties

Once you have created your Coordinate System in the New Tag window, you must then configure it. If you did not use the Wizard screens available from the Configure button on the New Tag screen, you can make your configuration selections from the Coordinate System Properties screen. You can also use the Coordinate System Properties screens to edit an existing Coordinate System tag. These have a series of Tabs that access a specific dialog for configuring the different facets of the Coordinate System. 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 save your selections.

**TIP** When you configure your Coordinate System, 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 coordinate system to edit and select Coordinate System Properties from the drop down menu. The Coordinate System Properties General window appears. The name of the Coordinate System tag that is being edited appears in the title bar to the right of Coordinate System Properties. The General screen is shown below.

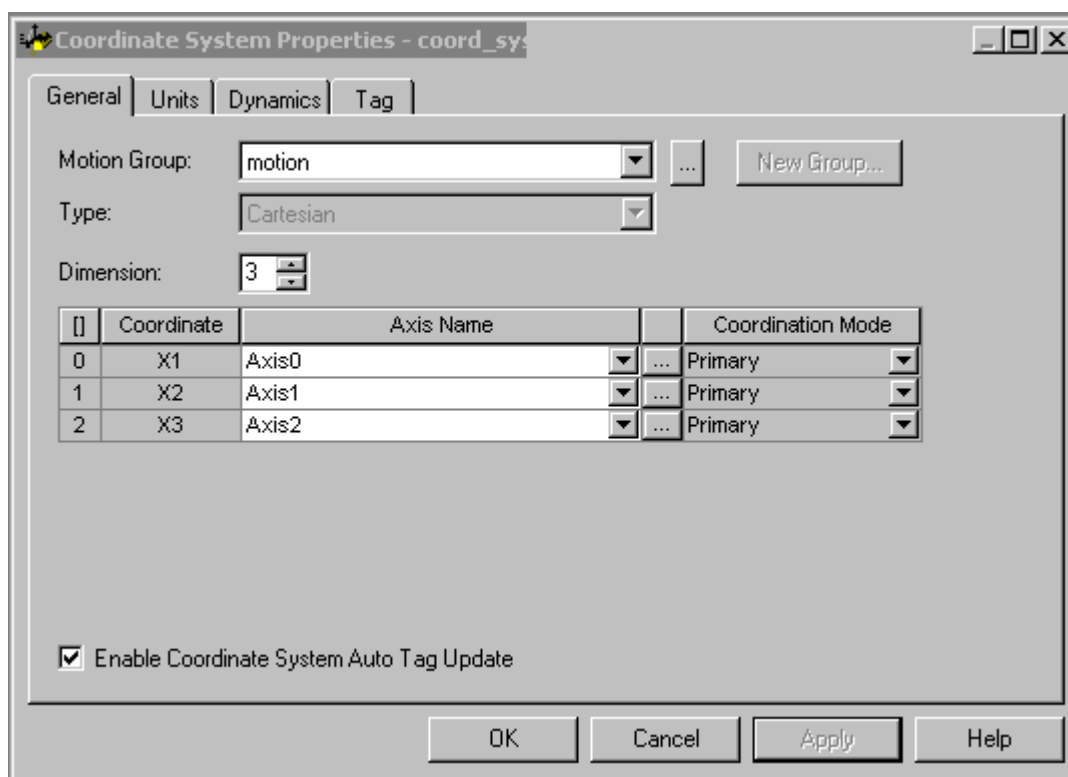


Figure 7.6 Coordinate System Properties - General Tab

**General Tab** Use this tab to do the following for a coordinate system:

- Assign the coordinate system, or terminate the assignment of a coordinate system, to a Motion Group.
- Change the number of dimension i.e. the number of axes.
- Assign axes to the coordinate system tag.
- Enable/Disable automatic updating of the tag.

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

### **Motion Group**

Selects and displays the Motion Group to which the Coordinate System is associated. A Coordinate System 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 coordinate system 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 Motion Group properties. If no Motion Group is assigned to this coordinate system, this button is disabled (grayed out).

### **New Group button**

The 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.

### **Type**

This read-only field displays the type of coordinate system. It currently only supports a Cartesian system therefore the field automatically fills with Cartesian and it cannot be edited.

### **Dimension**

Enter the dimension, i.e. the number of axes, that this coordinated system is to support. The options are 1, 2, or 3 in keeping with its support of a maximum of three axes. Changes in the Dimension spin box also reflect in the Axis Grid by either expanding or contracting the number of fields available. Data is set back to the defaults for any axis that is removed from the Axis Grid due to reducing the Dimension field.

## Axis Grid

The Axis Grid is where you associate axes to the Coordinate System. There are five columns in the Axis Grid that provide information about the axes in relation to the Coordinate System.

### [ ] (Brackets)

The Brackets column displays the indices in tag arrays used with the current coordinate system. The tag arrays used in multi-axis coordinated motion instructions map to axes using these indices.

### Coordinate

The text in this column X1, X2, or X3 (depending on the entry to the Dimension field) is used as a cross reference to the axes in the grid. For a Cartesian system the mapping is simple.

### Axis Name

The Axis Name column is a list of combo boxes (the number is determined by the Dimension field) used to assign axes to the coordinate system. The pulldown lists display all of the Base Tag axes defined in the project. (Alias Tag axes do not display in the pull down list.) They can be axes associated with the motion group, axes associated with other coordinated systems, or axes from the Ungrouped Axes folder. Select an axis from the pulldown list. The default is <none>. It is possible to assign fewer axes to the coordinate system than the Dimension field allows, however, you will receive a warning when you verify the coordinate system and if left in that state, the instruction generates a run-time error. You can only assign an axis once in a coordinate system. Ungrouped axes also generate a runtime error.

### Ellipsis Button (...)

The Ellipsis buttons in this column take you to the Axis Properties pages for the axis listed in the row. See the “Creating and Configuring Your Motion Axis” chapter in this manual for information about the Axis Properties page.

### Coordination Mode

The Coordination Mode column indicates the axes that are used in the velocity vector calculations. Only Primary axes are used in these calculations. Currently the only option is Primary. Therefore this column is automatically filled in as Primary and cannot be edited.

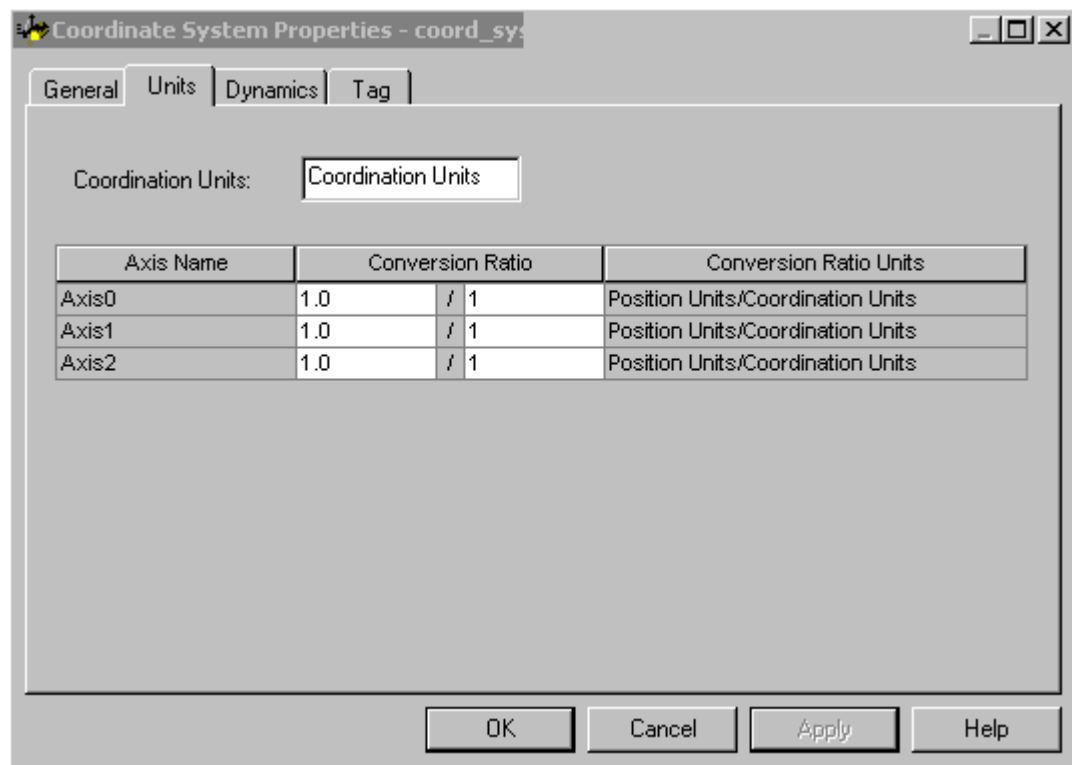
## Enable Coordinate System Auto Tag Update

The Enable Coordinate System Auto Tag Update checkbox lets you determine whether or not the Actual Position values of the current coordinated system are automatically updated during operation. Click on the checkbox to enable this feature. The Coordinate System Auto Tag Update feature can ease your programming burden if you would need to add GSV statements to the program in order to get the desired result. However, by enabling this feature the Coarse Update rate is increased. Whether to use the Coordinate System Auto Tag Update feature depends upon the trade-offs between ease in programming and increase in execution time. Some users may want to enable this feature in the initial programming of their system to work out the kinks and then disable it and enter the GSV statements to their program to lower their execution time.

**Note:** Enabling this feature may result in some performance penalty.

Press **Apply** to implement your entries or cancel to not save the new entries.

To edit the Units properties, select the Units tab to access the Coordinate System Properties Units dialog.



**Figure 7.7 Coordinate System Properties - Units Tab**

**Units Tab** The Units Tab of the Coordinate System Properties is where you determine the units that define the coordinate system. This screen is where you define the Coordination Units and the Conversion Ratios.

### Coordination Units

The Coordination Units field lets you define the units to be used for measuring and calculating motion related values such as position, velocity, and the like. The coordination units do not need to be the same for each coordinate system. Enter units that are relevant to your application and maximize ease of use. When you change the Coordination Units, the second portion of the Coordination Ratio Units automatically changes to reflect the new units. Coordination Units is the default.

### Axis Grid

The Axis Grid of the Units page displays the axis names associated with the Coordinate System, the conversion ratio, and the units used to measure the conversion ratio.

#### Axis Name

The Axis Name column contains the names of the axes assigned to the Coordinate System in the General screen. These names appear in the order that they were configured into the current coordinate system. This column is not editable from this screen.

#### Conversion Ratio

The Conversion Ratio column defines the relationship of axis position units to coordination units for each axis. For example: If the position units for an axis is in millimeters and the axis is associated with a coordinate system whose units are in inches, then the conversion ratio for this axis/coordinate system association is 25.4/1 and can be specified in the appropriate row of the Axis Grid.

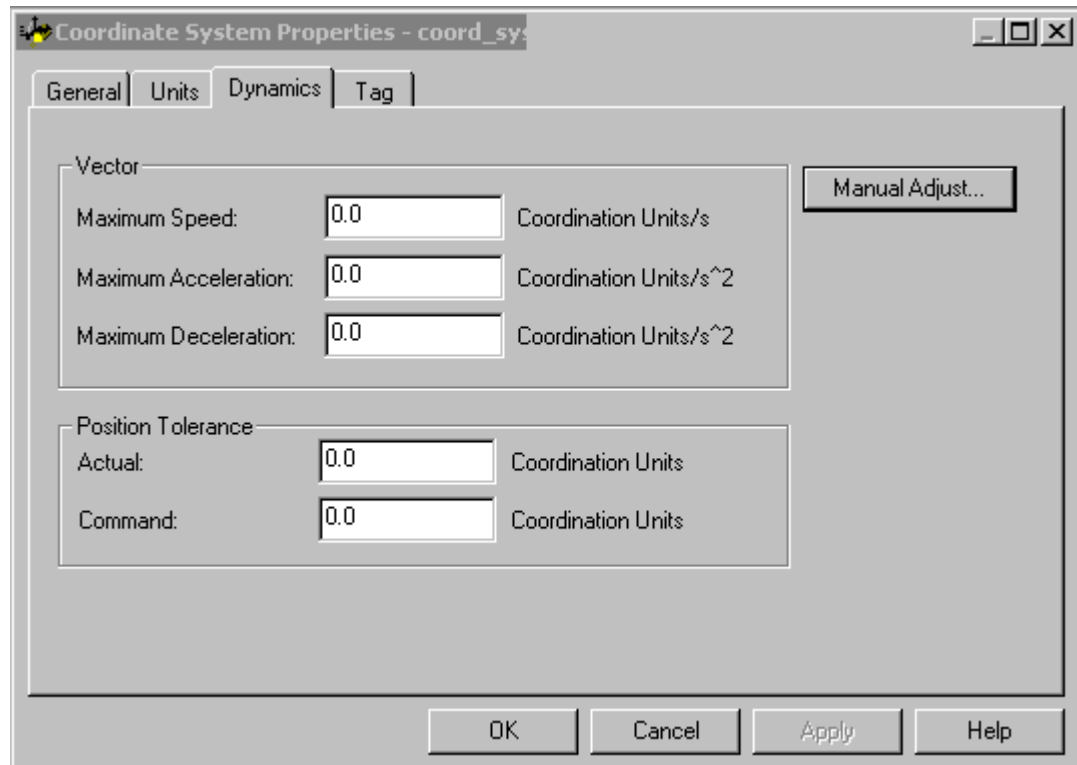
**Note:** The numerator can be entered as a float or an integer. The denominator must be entered as an integer only.

#### Conversion Ratio Units

The Conversion Ratio Units column displays the axis position units to coordination units used. The Axis Position units are defined in the Axis Properties – Units screen and the coordination units are defined in Coordinated System Properties – Units screen. These values are dynamically updated when changes are made to either axis position units or coordination units.

Click on the Apply button to preserve your edits or Cancel to discard your changes.

Click on the Dynamics Tab to access the Coordinate System Properties Dynamics dialog.



**Figure 7.8 Coordinate System Properties - Dynamics Tab**

**Dynamics Tab** The Dynamics dialog of the Coordinate System is for entering the Vector values used for Maximum Speed, Maximum Acceleration, and Maximum Deceleration. It is also used for entering the Actual and Command Position Tolerance values.

### Vector Box

In the Vector box, values are entered for Maximum Speed, Maximum Acceleration, and Maximum Deceleration and are used by the Coordinated Motion instructions in calculations when their operands are expressed as percent of Maximum. The Coordination Units to the right of the edit boxes automatically change when the coordination units are redefined at the Units screen.



### **Maximum Speed**

Enter the value for Maximum Speed to be used by the Coordinated Motion instructions in calculating vector speed when speed is expressed as a percent of maximum.

### **Maximum Acceleration**

Enter the value for Maximum Acceleration to be used by the Coordinated Motion instructions to determine the acceleration rate to apply to the coordinate system vector when acceleration is expressed as a percent of maximum.

### **Maximum Deceleration**

Enter the value for Maximum Deceleration to be used by the Coordinated Motion instructions to determine the deceleration rate to apply to the coordinate system vector when deceleration is expressed as a percent of maximum. The Maximum Deceleration value must be a non zero value to achieve any motion using the coordinate system.

## **Position Tolerance Box**

In the Position Tolerance Box, values are entered for Actual and Command Position Tolerance values. See the Logix5000 Motion Instruction Set Reference Manual (1756-RM007) for more information regarding the use of Actual and Command Position Tolerance.

### **Actual**

Enter the value in coordination units, for Actual Position to be used by Coordinated Motion instructions when they have a Termination Type of Actual Tolerance.

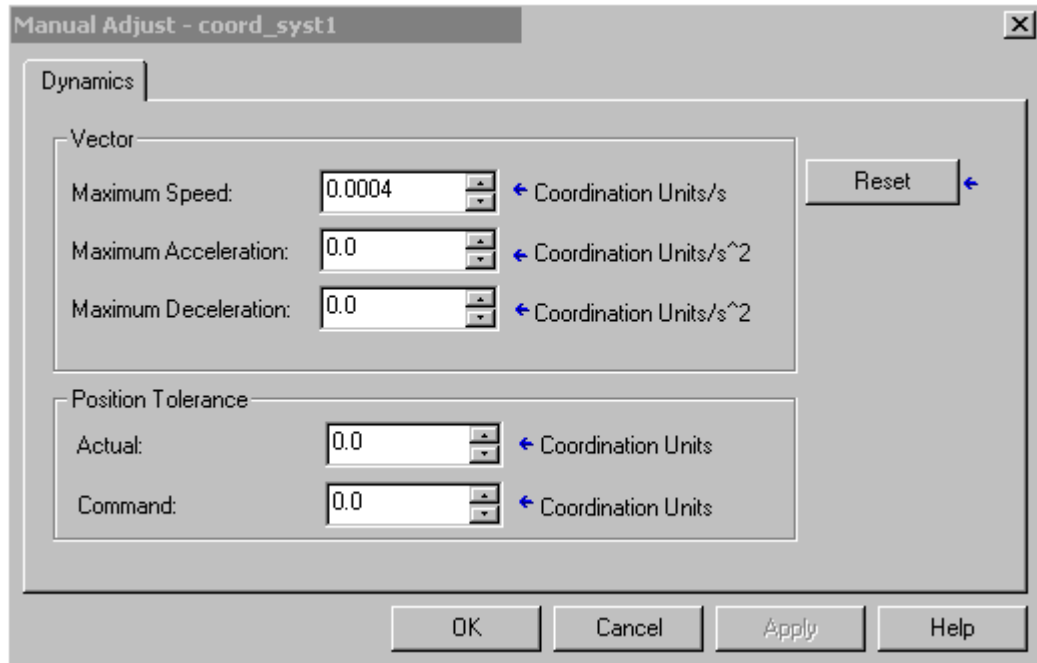
### **Command**

Enter the value in coordination units, for Command Position to be used by Coordinated Motion instructions when they have a Termination Type of Command Tolerance.

## **Manual Adjust Button**

The Manual Adjust button on the Coordinate System Dynamics Tab accesses the Manual Adjust Properties dialog. The Manual Adjust button is enabled only when there are no pending edits on the properties dialog.

**Dynamics Tab Manual Adjust** At this screen you can make changes to the Vector and Position Tolerance values. See the explanations for the Vector and Position Tolerance fields in the explanation of the Dynamics Tab earlier in this chapter.



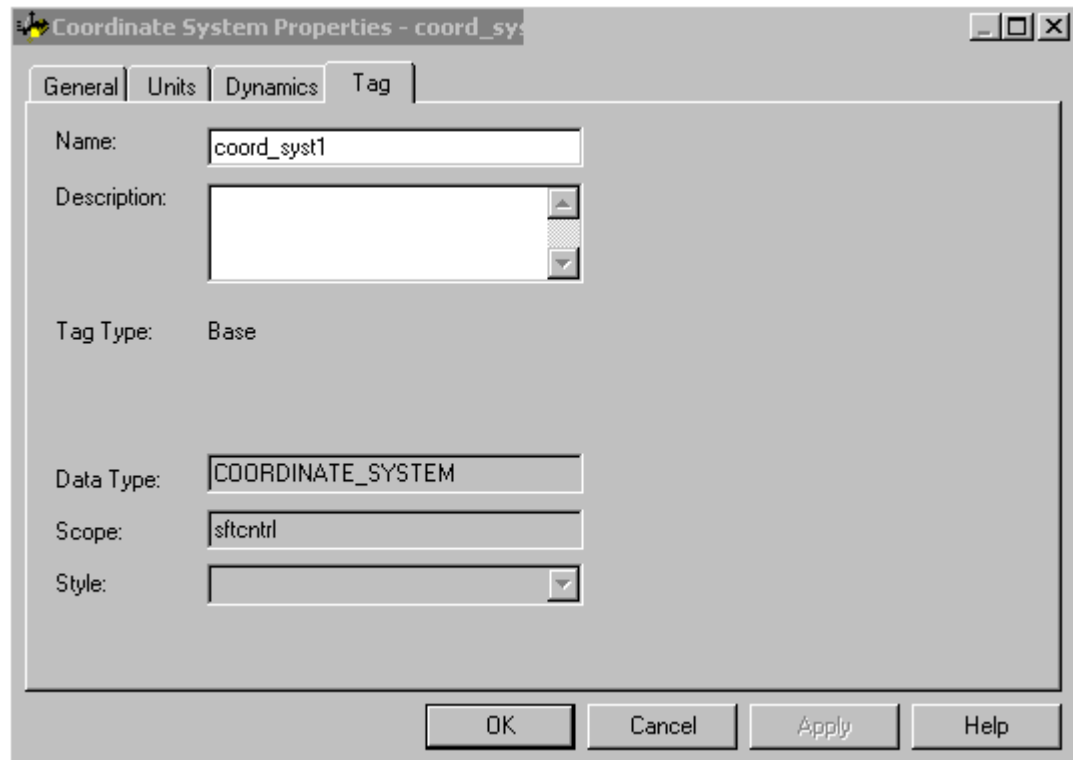
**Figure 7.9 Coordinate System Properties - Manual Adjust Screen of Dynamics Tab**

These changes can be made either on or off line. The blue arrows to the right of the fields indicate that they are immediate commit fields. This means that the values in those fields are immediately updated to the controller if on-line or to the project file if off line.

### Reset Button

The Reset Button reloads the values that were present at the time this dialog was entered. The blue arrow to the right of the Reset button means that the values are immediately reset when the Reset button is clicked.

**Tag Tab** The Tag Tab is for reviewing your Tag information and renaming the tag or editing the description.



**Figure 7.10 Coordinate System Properties - Tag Tab**

Use this tab to modify the name and description of the coordinate system. 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 the tag at this time. The name can be up to 40 characters and can include letters, numbers, and underscores (\_). When you rename a tag, the new name replaces the old one in the Controller Organizer after click on the OK or Apply button.

### Description

Displays the description of the current tag, if any is available. You can edit this description. The edited description replaces the existing description when you click on either the OK or Apply button.

## Tag Type

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

- Base
- Alias

The field is not editable and is for informational purposes only.

## Data Type

Displays the data type of the current Coordinate System tag which is always COORDINATE\_SYSTEM. This field cannot be edited and is for informational purposes only.

## Scope

Displays the scope of the current Coordinate System tag. The scope for a Coordinate System tag can only be controller scope. This field is not editable and is for informational purposes only.

## Style

Not applicable.

## Right Mouse Click Properties

Right mouse clicking on a specific Coordinate System launches the following pop-up menu.

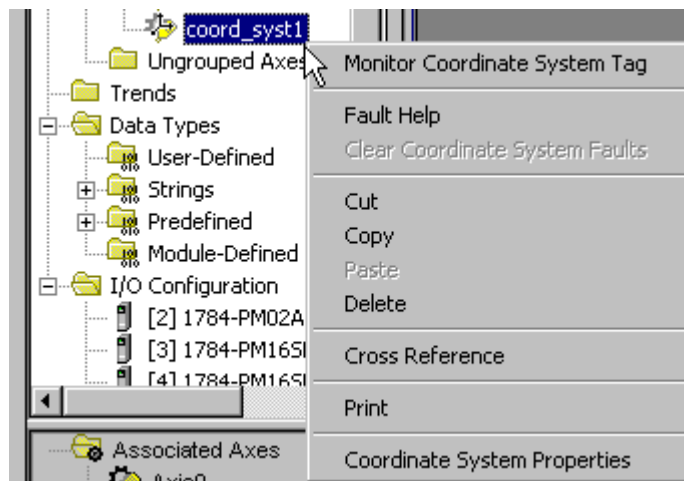


Figure 7.11 Right Click Pop-Up Menu

The menu has the following options:

- Monitor Coordinate System Tag – launches the data monitor with focus on the coordinate system tag from which the monitor was launched.
- Fault Help – launches on-line help to assist in understanding and correcting system faults.
- Clear Coordinate System Faults – clears all system faults associated with this coordinate system tag. This option is grayed out (inactive) if there are no faults associated with the selected coordinate system.
- Cut – cuts the coordinated system from its folder.
- Copy – copies the selected coordinated system and all of its properties.
- Paste – is never active from the right mouse click menu when initiated from the coordinate system tag. It only becomes active when initiated from a right mouse click on the Ungrouped Axes folder or Motion Group when a coordinate system has been Cut or Copied.
- Delete – removes the coordinate system from the Motion Group Tag or Ungrouped Axes folder.
- Cross Reference – launches the Cross Reference screen which lists all references associated with the selected coordinate system tag.
- Print – sends tag information to the printer.
- Coordinate System Properties – launches the tabbed Coordinate System Properties screen.

### **Cut, Copy, Paste, and Delete Behavior**

A Coordinate System tag can be cut or copied from either a Motion Group Tag or the Ungrouped Axes folder. Once cut or copied it can be pasted into either a Motion Group Tag or the Ungrouped Folder.

### **Copy/Paste**

A Copy/Paste operation implies creation of a new coordinate system tag. The new tag has the exact same properties as its host. It is automatically given a new name when pasted to its new location. The new name is the same as the old one but with a one added after the last existing character. For example: Copying and pasting the coordinate system tag `coord_syst2` would create a new tag with the name `coord_syst21`. Subsequent copying and pasting of the same tag would increment the name by one on the last digit i.e. `coord_syst22`, `coord_syst23`, `coord_syst24`, etc. It can be pasted into the same motion group tag or into the Ungrouped Folder. A maximum of 32 Coordinate System tags can be created.

### **Cut/Paste**

A Cut/Paste operation is used for moving the Coordinate System tag from either a Motion Group Tag to the Ungrouped Axes folder or vice versa. When a Cut/Paste operation is performed on a tag being moved from a Motion Group tag to the Ungrouped Axes folder it unassigns the coordinate system tag from the motion group. Likewise when it moves to the Motion Group tag it becomes assigned to the Motion group tag.

### **Delete**

Delete removes the Coordinate System tag from a Motion Group Tag or the Ungrouped Axes folder. If a Motion Group is deleted, all coordinate system tags associated with that motion group are unassigned and placed in the Ungrouped Axes folder.

## Configuring a 1394x-SJTxx-D Digital Servo Drive

To configure a 1394x-SJTxx-D drive module:

1. In the Controller Organizer, in the I/O Configuration branch, select a 1784-PM16SE motion module.
2. In the File menu, select New Component then Module.

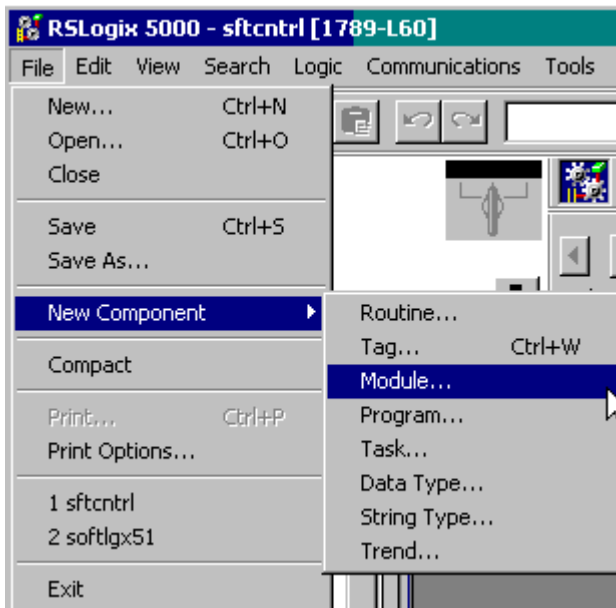
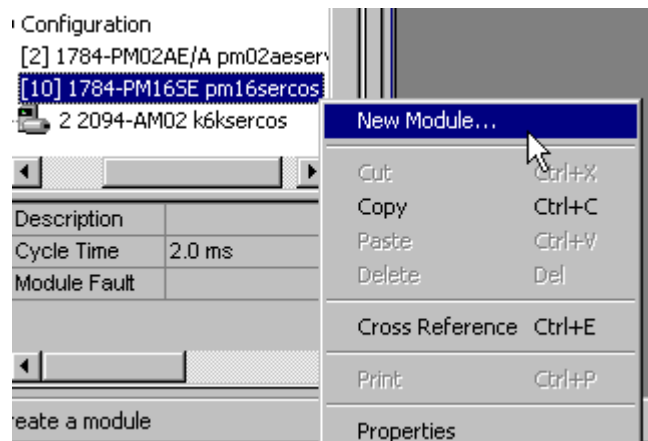


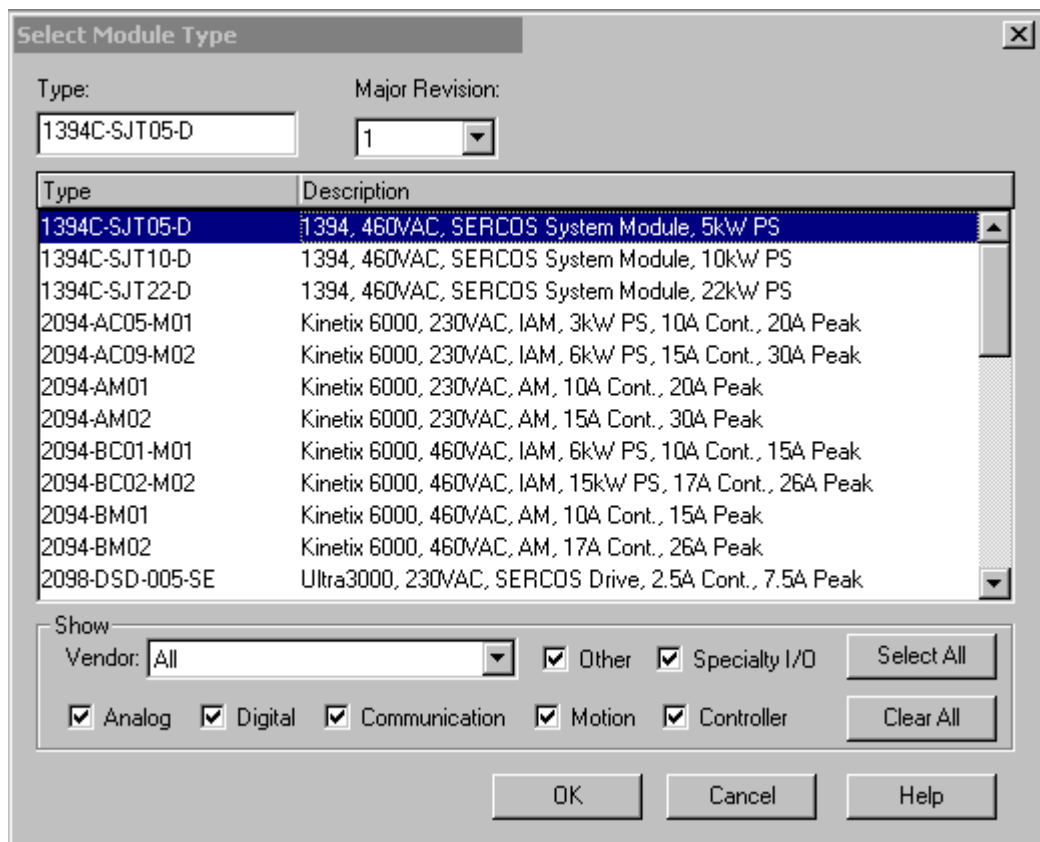
Figure 8.1 File Menu to New Component to Module

3. You can also right click on a selected 1784-PM16SE module and select New Module from the pop up menu.



**Figure 8.2 Right Click on 1784-PM16SE**

4. In the Select Module Type dialog, select the desired 1394x-SJTxx-D drive module.



**Figure 8.3 Select Module Type Screen**

5. Press the OK button to close the Select Module Type dialog. The Module Properties wizard opens.



- Fill in the required parameters for each page, then click the Next> button.

The screenshot shows a dialog box titled "Module Properties - pm16sercos (1394C-SJT05-D)". The fields are as follows:

- Type: 1394C-SJT05-D 1394, 460VAC, SERCOS System Module, 5kW PS
- Vendor: Allen-Bradley
- Name: sercos1394
- Base Node: 10
- Description: (empty)
- Revision: 1
- Electronic Keying: Compatible Module

At the bottom of the dialog are five buttons: Cancel, < Back, Next >, Finish >>, and Help.

**Figure 8.4 Module Properties Wizard Dialog - Naming the Drive**

- When you complete the last page, click the Finish> button. A new drive module displays beneath the selected 1784-PM16SE motion module.

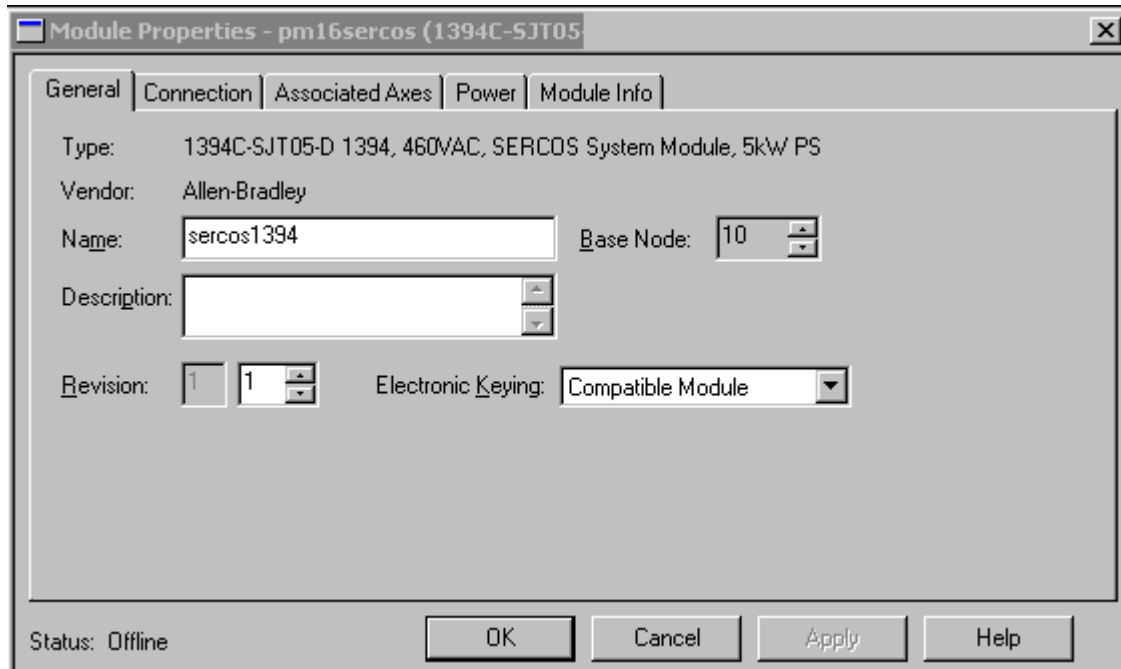
## 1394x-SJTxx-D Digital Servo Drive Overview

The 1784-PM16SE 16 Axis SERCOS interface motion module can be connected to any of three drives:

- 1394x-SJT05-D 5 KW digital servo drive
- 1394x-SJT10-D 10 KW digital servo drive
- 1394x-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 1784-PM16SE SERCOS interface module can support up to 32 axes.

The module for a 1394x-SJTxx-D drive has 5 tabs:



**Figure 8.5 Module Properties - General Tab**

- General tab
- Connection tab
- Axes Association
- Power tab
- Module Info tab.

**General Tab** Use this tab to enter the module properties for 1394x-SJTxx-D digital servo drive modules.

**IMPORTANT**

To create any one of the 1394x-SJT modules, the parent module must be a 1784-PM16SE 16 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 1394x-SJTxx-D digital servo drive 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.

### **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 rejects the connection.
  - Vendor
  - Product Type
  - Catalog Number
  - Major Revision
  - Minor Revision
- **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 softwareor the inserted module rejects the connection
- **Disable Keying** – Controller does not employ keying at all.

---

**ATTENTION**

Changing the Electronic Keying selection 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.

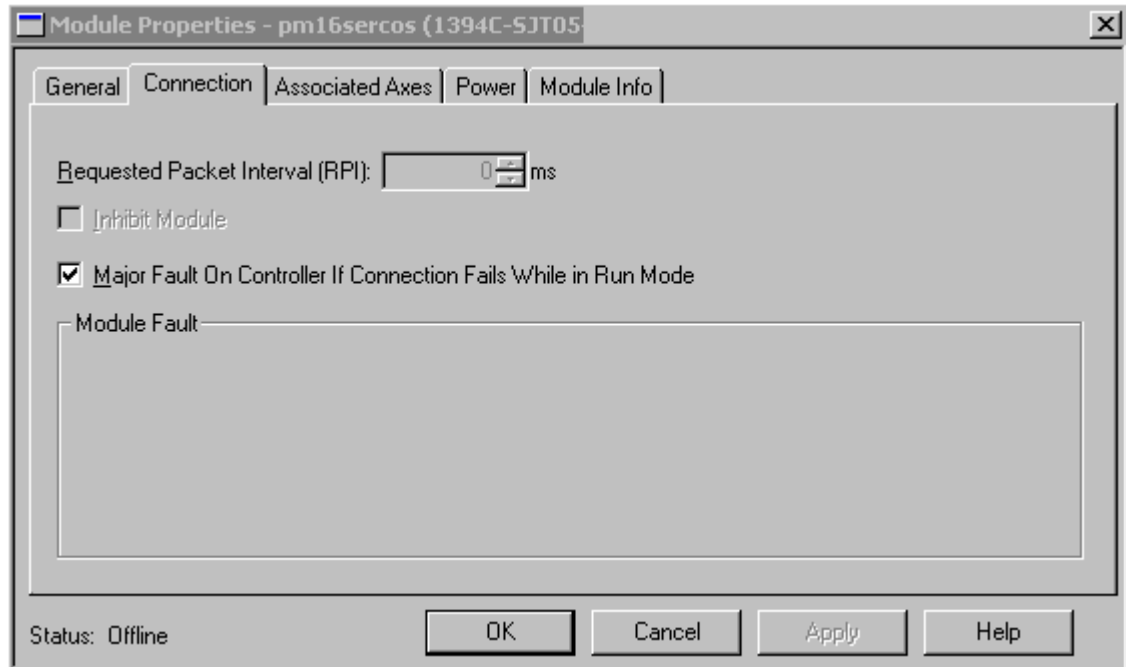
---

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.



**Figure 8.6 Module Properties - Connection Tab**

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

This field is disabled for all motion modules (e.g., 1784-PM02AE, 1784-PM16SE, and all 1394-, Ultra3000, Kinetix 6000, and 8720 modules).

### 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.

**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 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 1394x-SJTxx-D drive module by associating up to four AXIS\_SERVO\_DRIVE axis tags with configured axis modules.

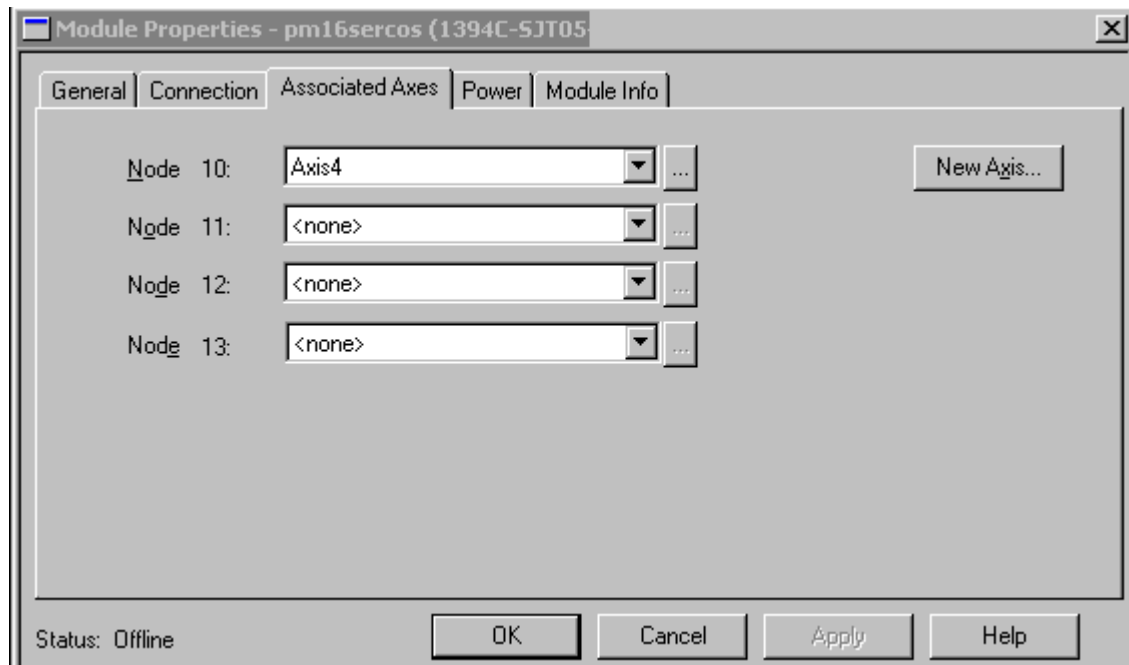


Figure 8.7 Module Properties - Associated Axis Tab

### **Node X0**

Represents Axis 0 on the 1784-PM16SE 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 1784-PM16SE 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 1784-PM16SE 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**

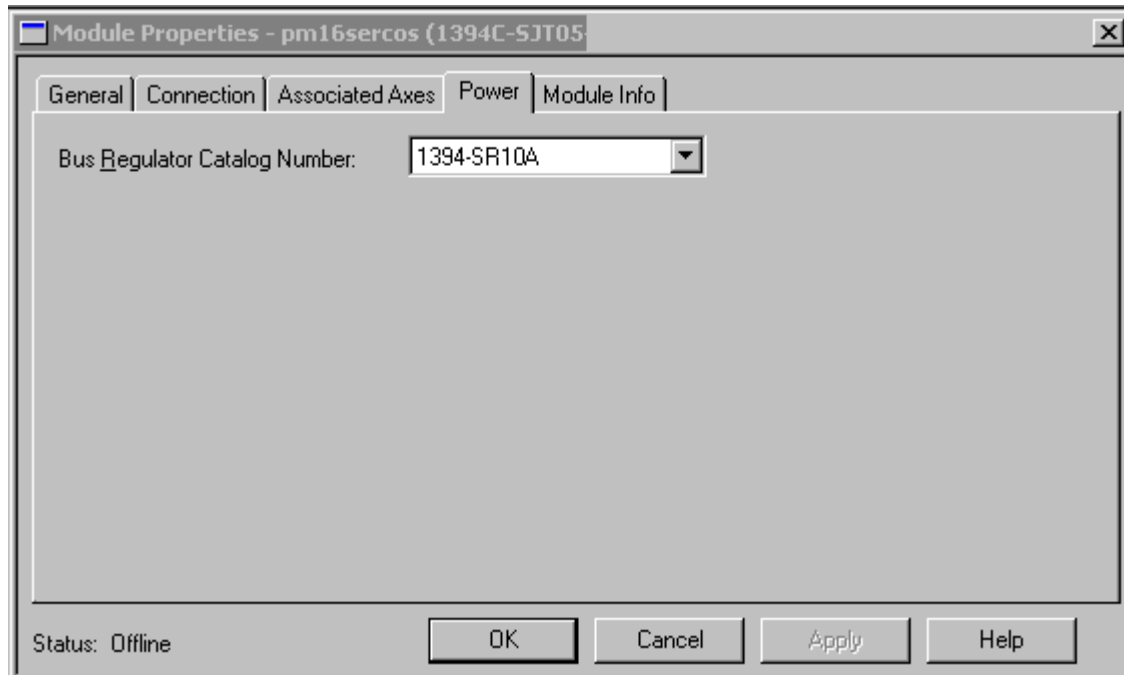
Represents Axis 3 on the 1784-PM16SE SERCOS module. The node number is the sum of the Base Node set in the General page of this dialog box (X0) and the axis number (3). This field allows you to associate an AXIS\_SERVO\_DRIVE tag with Axis 3. This field transitions to a read only state while online. 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 1394x-SJTxx-D drive module.



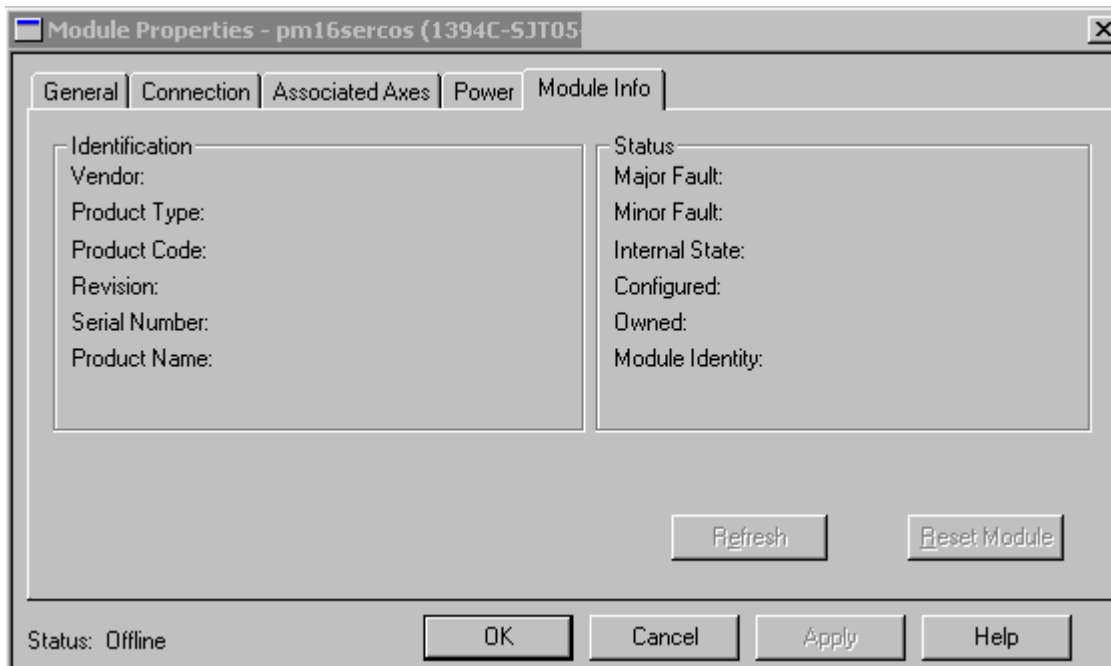
**Figure 8.8 Module Properties - Power Tab**

### Bus Regulator ID

Select the catalog number that describes bus regulator device used by the 1394x-SJTxx-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 1394x-SJTxx-D drive module. It also allows you to refresh a module and reset a module to its power-up state.



**Figure 8.9 Module Properties - Module Info Tab**

The information on this tab is not displayed if you are:

- offline, or
- currently creating a module

#### TIP



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.

---

**ATTENTION**

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

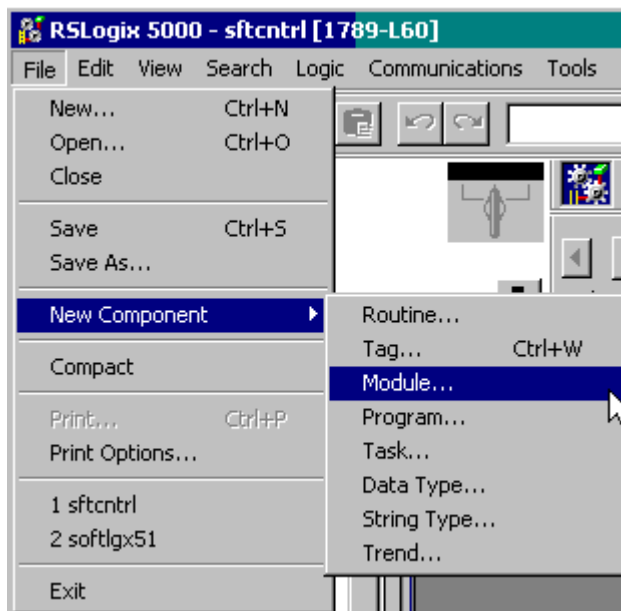


## Configuring an Ultra 3000 Drive

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 1784-PM16SE motion module.
2. In the File menu, select New Component then Module.

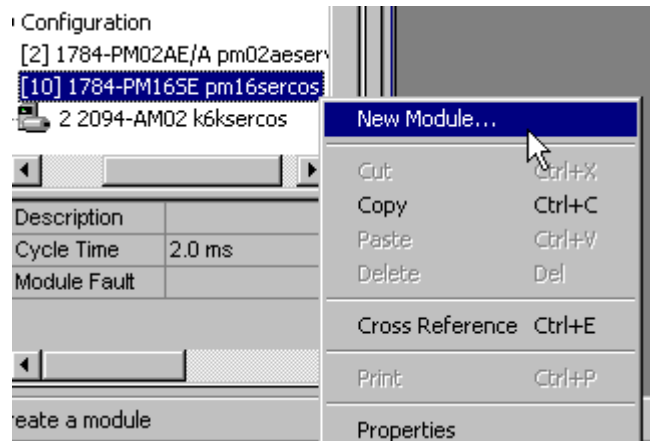


**Figure 9.1 File Menu - New Component - Module**

OR

3. Right click on the selected 1784-PM16SE in the I/O Configuration branch of the Controller Organizer.

4. Select New Module from the pop up menu.



**Figure 9.2 New Module Selection from Pop Up Menu**

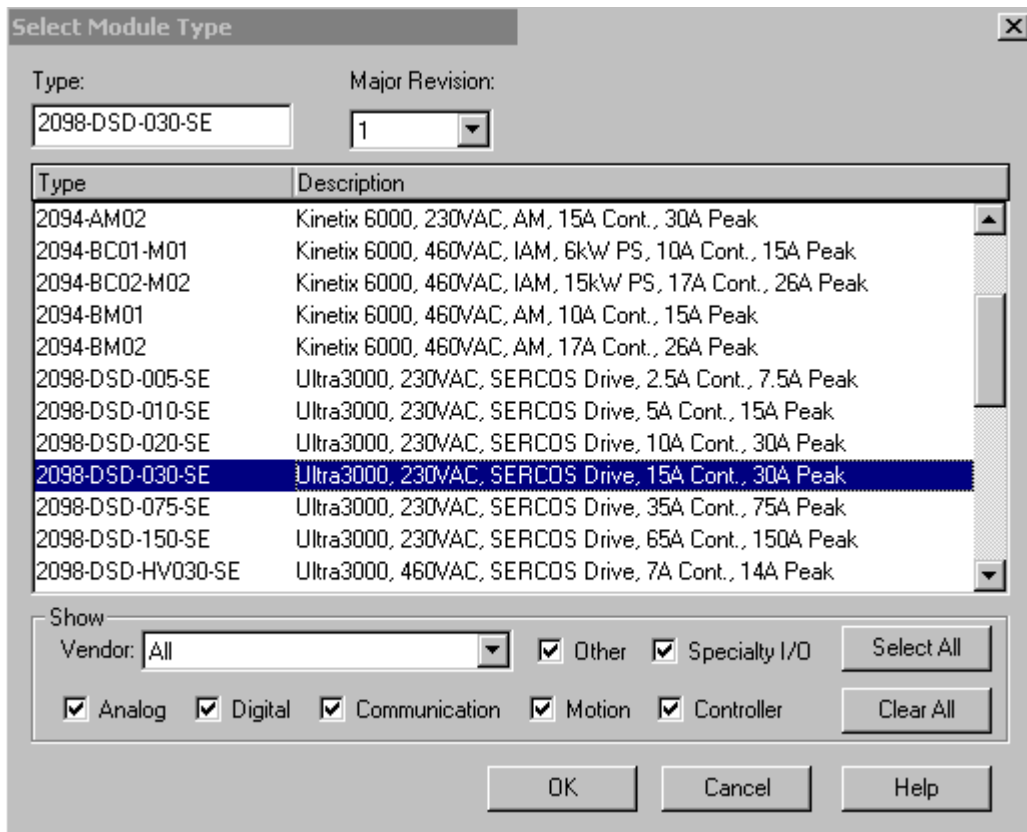
The following fields are displayed only if you are viewing this tab through the Create wizard.

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.



**Figure 9.3 Select Module Type Window**

5. In the Select Module Type dialog, select the desired drive module. The Ultra drives begin with the 2098 prefix.

- Press the OK button to close the Select Module Type dialog. The Ultra Drive Create Wizard Module Properties dialog opens.

The screenshot shows the 'Module Properties - pm16sercos (2098-DSD-030-)' dialog box. The fields are filled as follows:

- Type: 2098-DSD-030-SE Ultra3000, 230VAC, SERCOS Drive, 15A Cont., 30A Peak
- Vendor: Allen-Bradley
- Name: sercos2098ultra3
- Node: 1
- Description: (empty)
- Revision: 1
- Electronic Keying: Compatible Module

Buttons at the bottom: Cancel, < Back, Next >, Finish >>, Help.

**Figure 9.4 Module Properties Wizard Dialog - Naming the Drive**

- 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.

The screenshot shows the 'Module Properties - sercos1 (2098-DSD-030-SE 1.1)' dialog box. The fields are filled as follows:

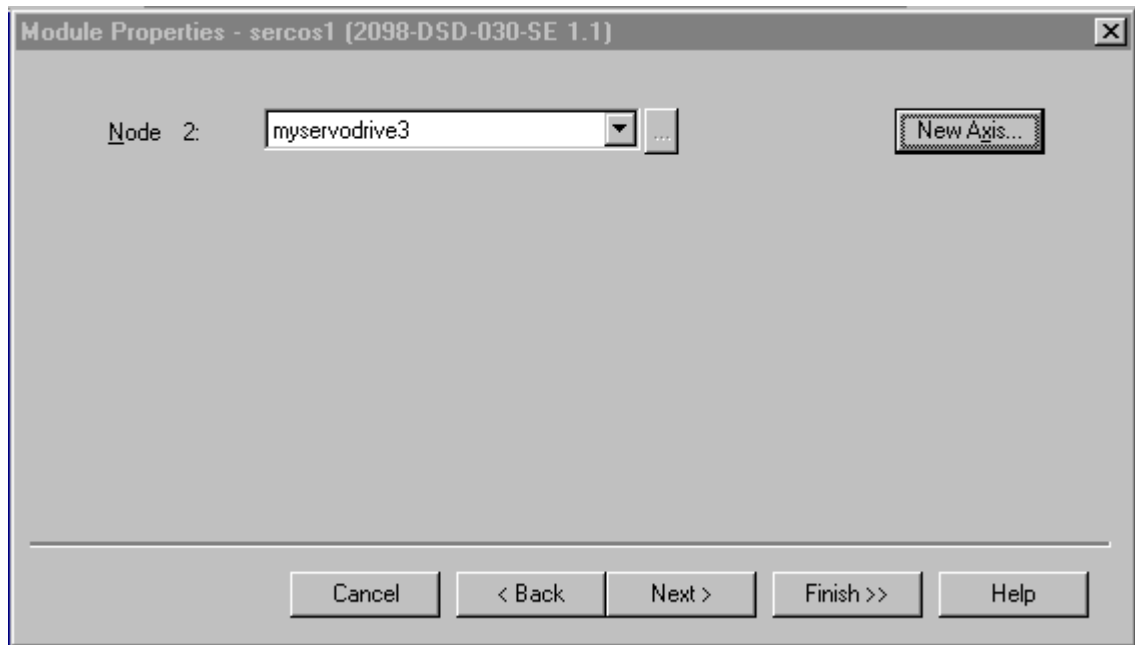
- Requested Packet Interval (RPI): 0 ms (2.0 - 750.0 ms)
- Inhibit Module
- Major Fault On Controller If Connection Fails While in Run Mode
- Module Fault: (empty text area)

Buttons at the bottom: Cancel, < Back, Next >, Finish >>, Help.

**Figure 9.5 Module Properties Wizard Dialog - Connection**

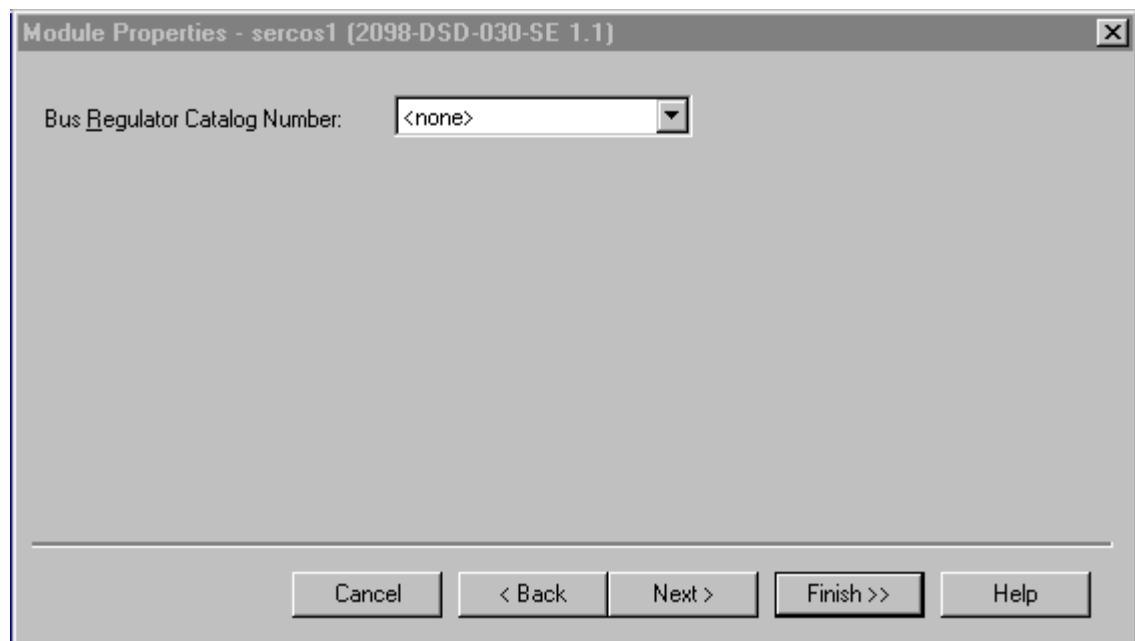


8. Fill in the required information and click on the Next button to advance to the next screen.



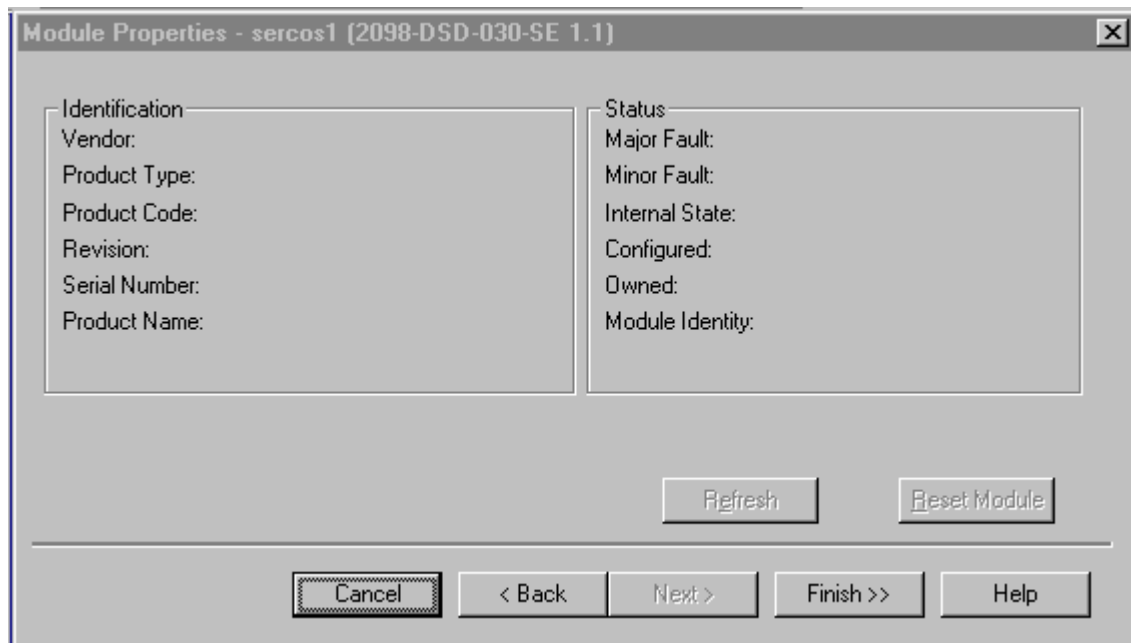
**Figure 9.6 Module Properties Wizard Dialog - Axis Information**

9. Fill in the node information for the drive and press Next.



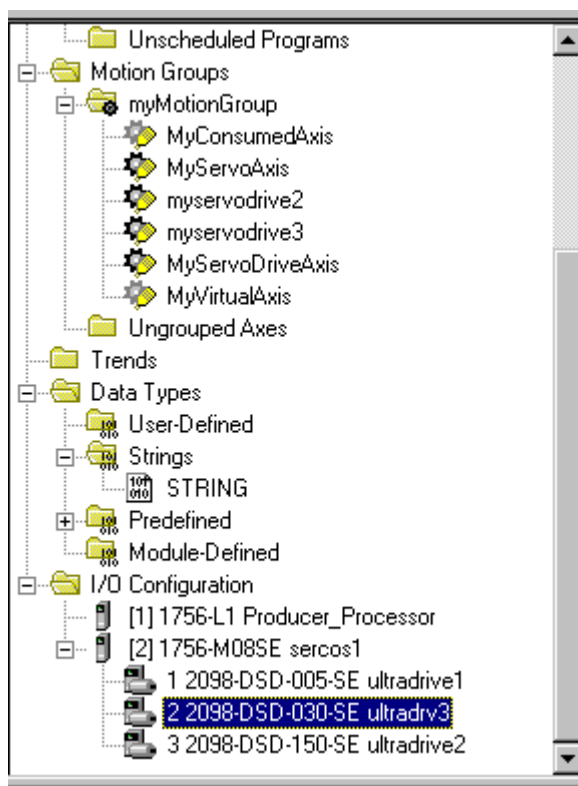
**Figure 9.7 Module Properties - Bus Information**

10. There is no Bus regulator for the Ultra3000. Press Next to continue.



**Figure 9.8 Module Properties - Module Information**

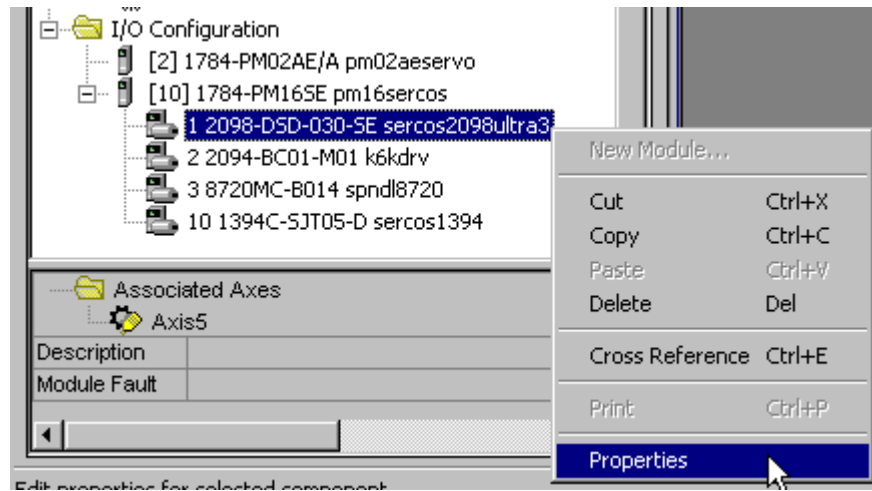
11. When you complete the last page, click the Finish> button. A new drive module displays beneath the selected 1784-PM16SE motion module.



**Figure 9.9 Controller Organizer - New Drive**

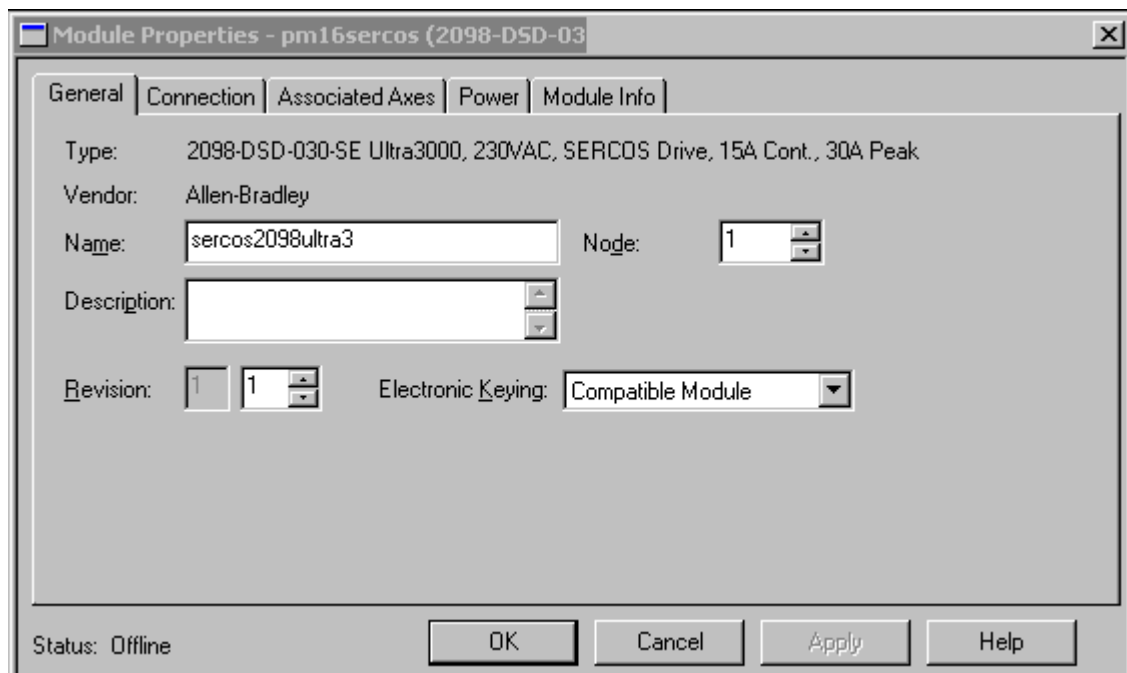
## Editing the Ultra Drive Properties

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.



**Figure 9.10 Accessing the Properties of the Drive**

The Module Properties screen displays.



**Figure 9.11 Module Properties - General Tab**

**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** – Controller does not employ keying at all.

---

**ATTENTION**

Changing the 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

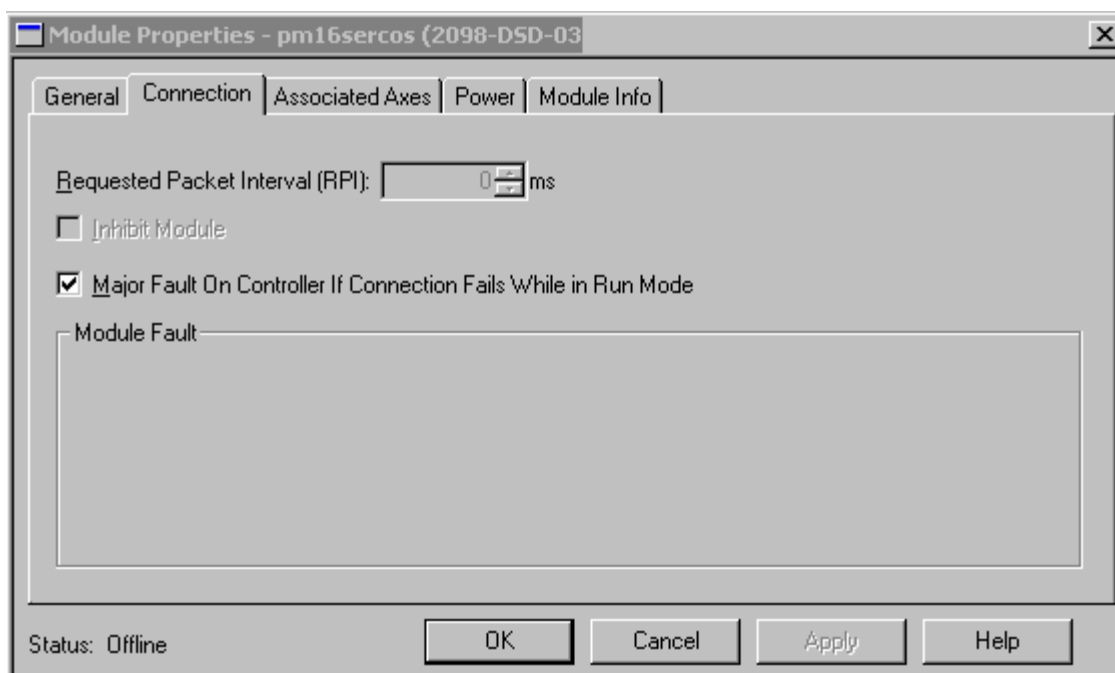
This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

### 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: <ul style="list-style-type: none"> <li>• its parent has not yet made a connection to it</li> <li>• its parent is inhibited</li> <li>• its parent is faulted</li> </ul>
Offline	You are not online.

**Connection Tab** Use this tab to define controller to module behavior.



**Figure 9.12 Module Properties - Connection Tab**

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

This field is disabled for all motion modules (e.g., 1784-PM02AE, 1784-PM16SE, and all 1394-, Ultra3000, Kinetix 6000, and 8720 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.

---

**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 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 <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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'.</li> <li>• 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.</li> <li>• If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.</li> </ul>

### 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

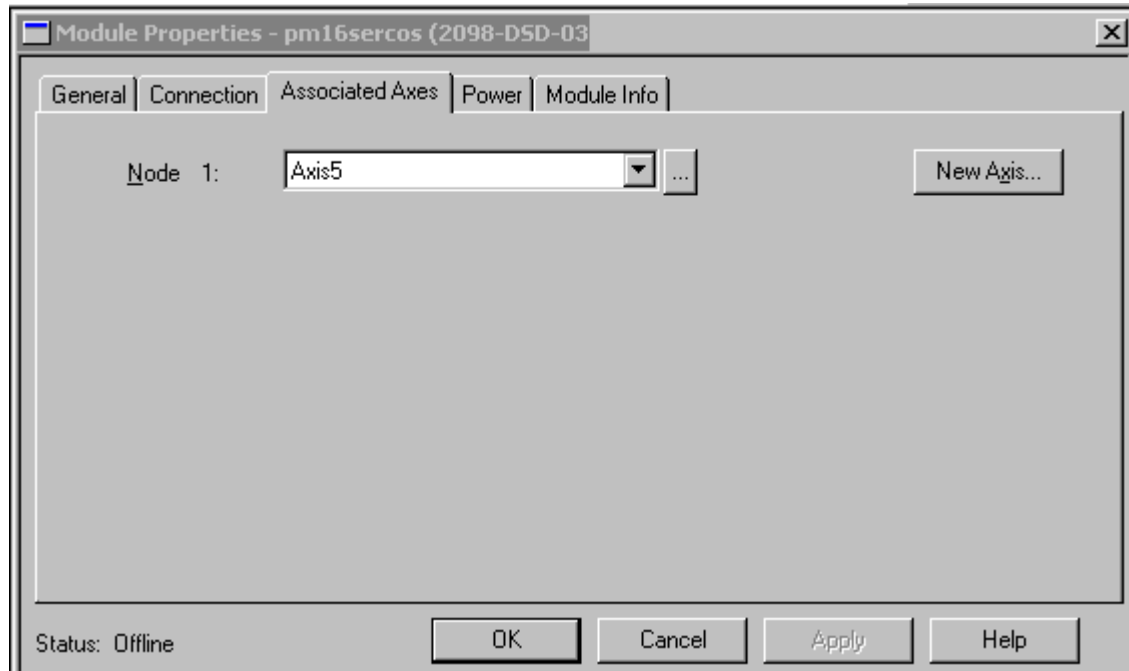
Displays the fault code returned from the controller (related to the module you are configuring) and the text detailing the Module Fault that has occurred.

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 1784-PM16SE motion module by associating axis tags (of the type `AXIS_SERVO_DRIVE`) with nodes available on the module.



**Figure 9.13 Module Properties - Associated Axes Tab**

### 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.

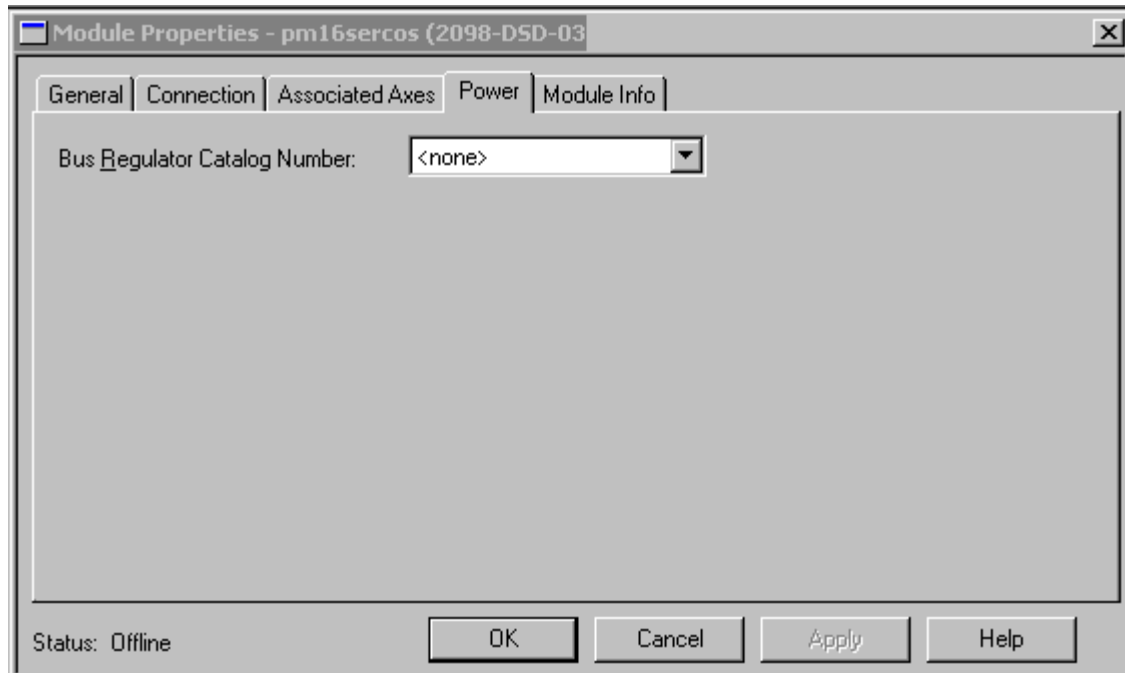
### Ellipsis (...)

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.



**Figure 9.14 Module Properties - Power Tab**

**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>.

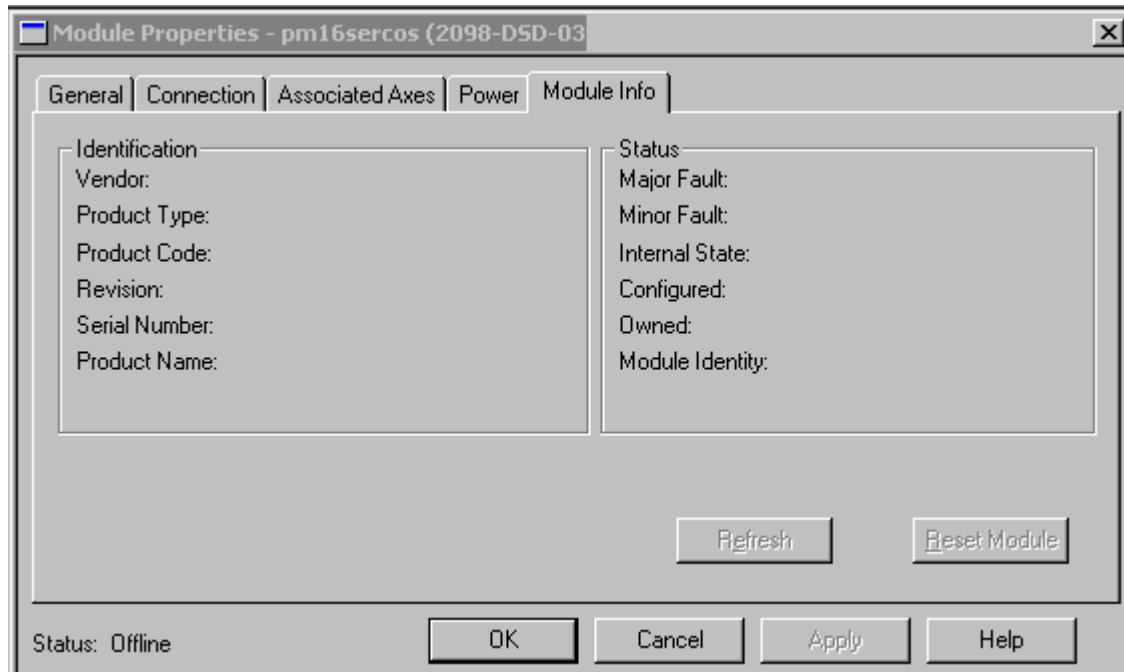
### Bus Regulator ID

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 Tab

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.



**Figure 9.15 Module Properties - Module Info**

### 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

<b>If you are configuring a:</b>	<b>This field displays one of the following:</b>
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: <ul style="list-style-type: none"> <li>• Vendor</li> <li>• Module Type (the combination of Product Type and Product Code for a particular Vendor)</li> <li>• Major Revision</li> </ul>
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.

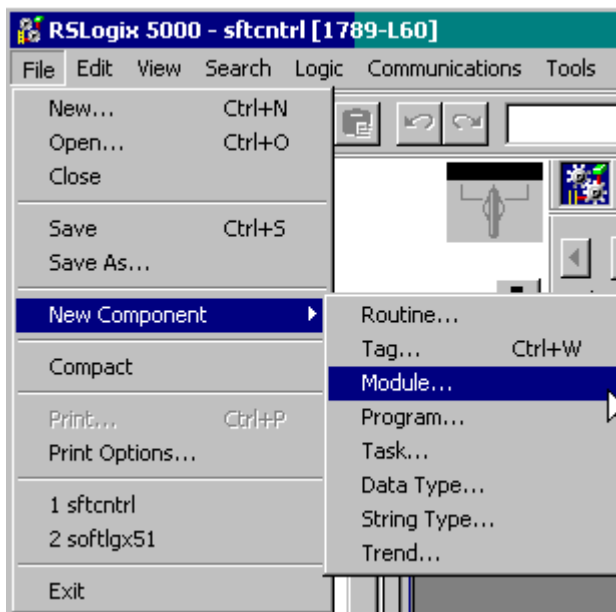
## Configuring a Kinetix 6000 Drive

The Kinetix 6000 Digital Servo Drive with fiber optic SERCOS interface simplifies the integration of the Kinetix 6000 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 Kinetix 6000 drive module by either of two methods.

The first method:

1. In the Controller Organizer, in the I/O Configuration branch, select a 1784-PM16SE motion module.
2. In the File menu, select New Component then Module.

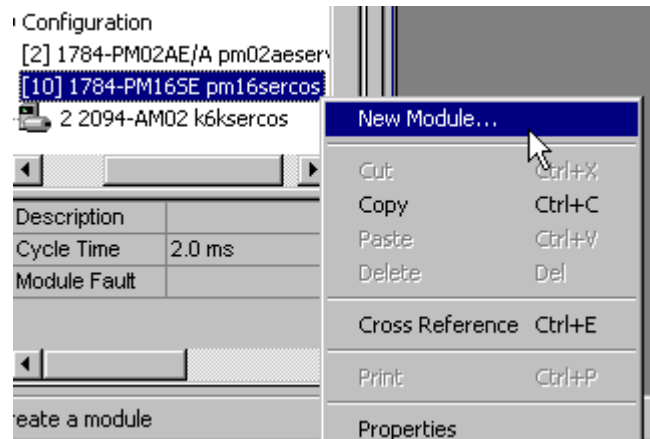


**Figure 10.1 File Menu - New Component - Module**

The second method:

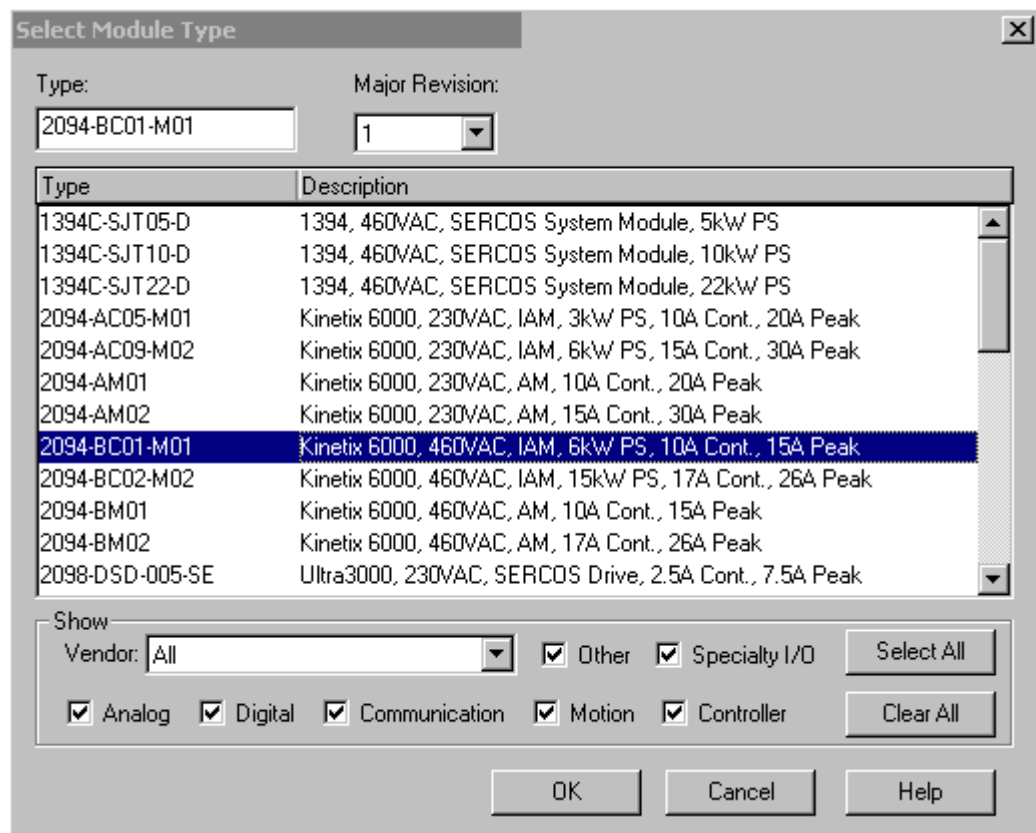
1. Right click on the selected 1784-PM16SE in the I/O Configuration branch of the Controller Organizer.

2. Select New Module from the pop up menu.



**Figure 10.2 New Module Selection from Pop Up Menu**

The Select Module Type dialog displays.

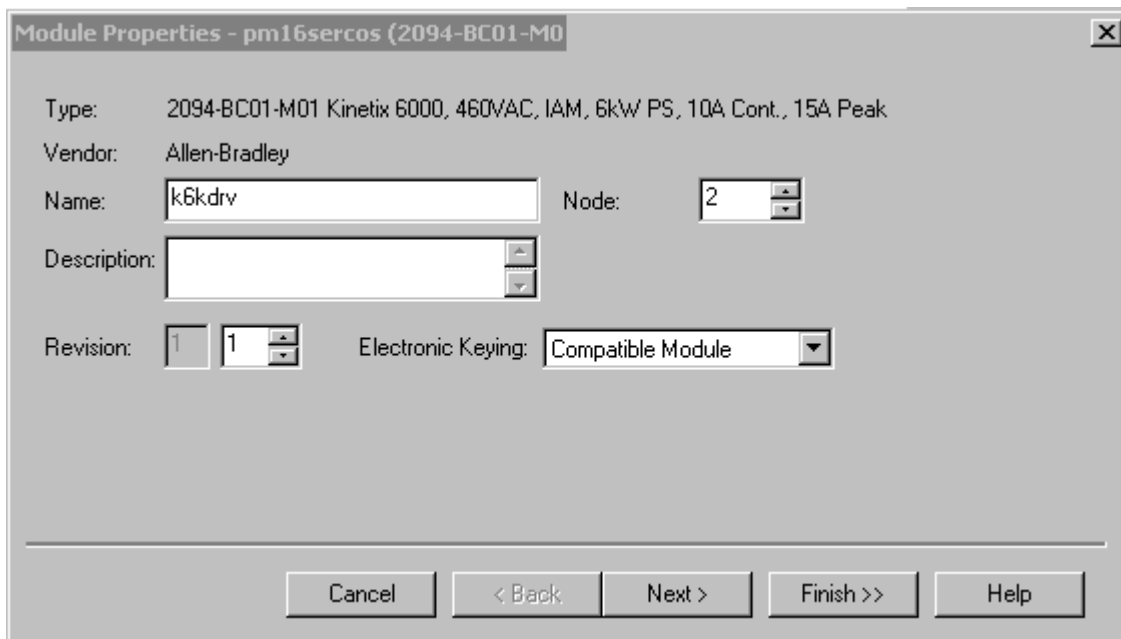


**Figure 10.3 Select Module Type Window**

3. In the Select Module Type dialog, select the desired drive module. The Kinetix 6000 drives begin with the 2094 prefix.



4. Press the OK button to close the Select Module Type dialog. The Kinetix 6000 Drive Create Wizard Module Properties dialog opens.

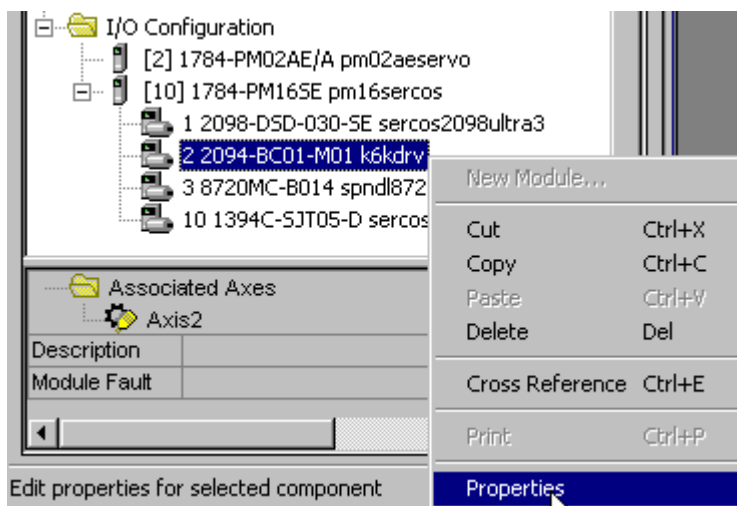


**Figure 10.4 Module Properties Wizard Dialog - Naming the Drive**

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 or click on the Finish>> button to add the drive.

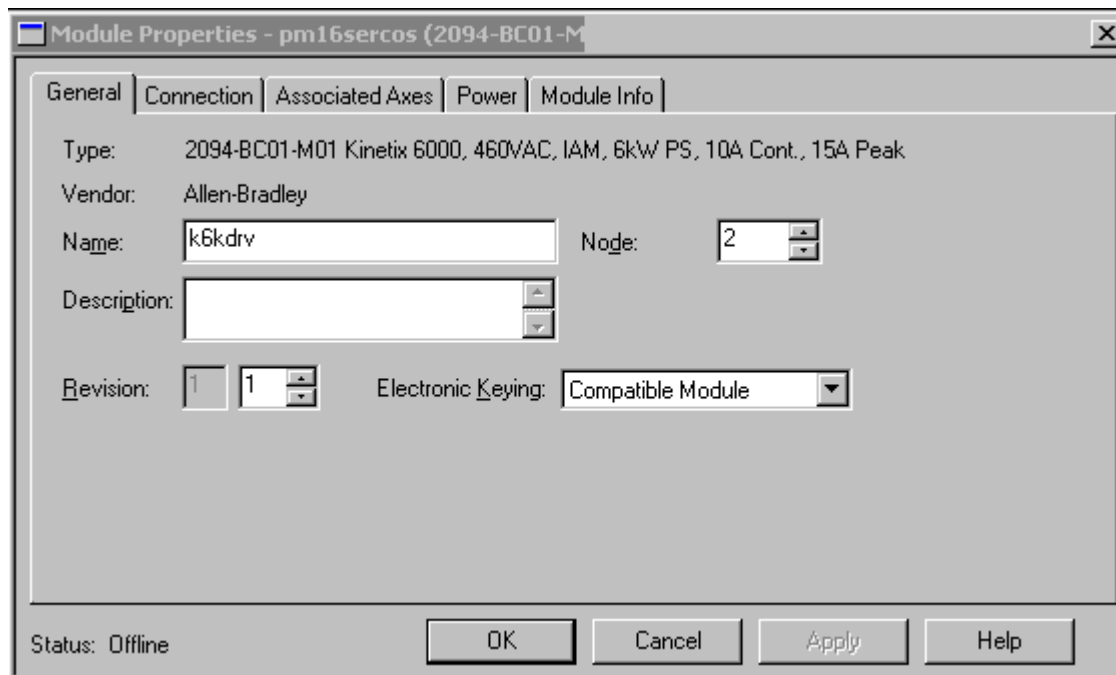
## Editing the Kinetix Drive Properties

The Module Properties for any of the Kinetix 6000 drives can be edited by highlighting the drive to be edited, right click with the mouse and select Properties.



**Figure 10.5 Accessing the Properties of the Drive**

The Module Properties screen displays.



**Figure 10.6 Module Properties - General Tab**

**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 a name for 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 the extra character(s).

## Node

Enter the SERCOS node number of the drive module. Valid values include those nodes not already in use. You can determine the SERCOS node number by checking the position of the rotary switch on the associated drive. IAM has node switch which specifies remaining slot location node addresses.

## 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 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** – the controller does not employ keying at all.

**ATTENTION**

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

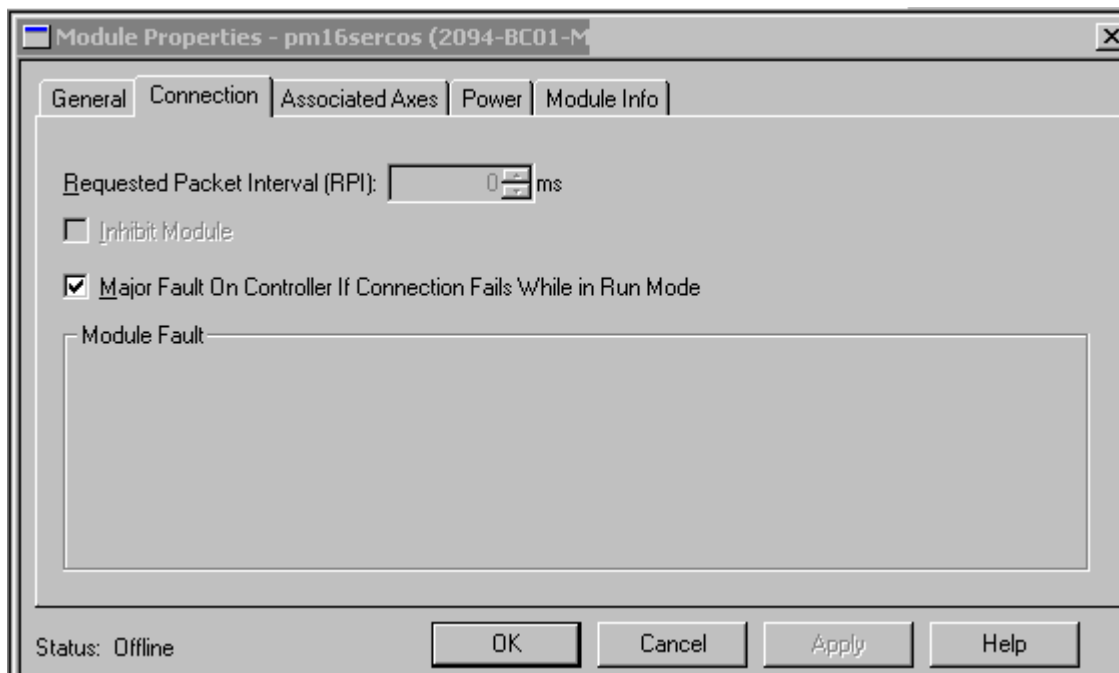
Be extremely cautious when using this option; if used incorrectly, this option can lead to personal injury or death, property damage or economic loss.

**Status**

Displays the status the controller has about the module:

<b>This status:</b>	<b>Indicates:</b>
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: <ul style="list-style-type: none"> <li>• its parent has not yet made a connection to it</li> <li>• its parent is inhibited</li> <li>• its parent is faulted</li> </ul>
Offline	You are not online.

**Connection Tab** Use this tab to define controller to module behavior.



**Figure 10.7 Module Properties - Connection Tab**

On this tab, you can:

- Requested Packet Interval – does not pertain to this drive.
- 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

This field is disabled for all motion modules (e.g., 1784-M02AE, 1784-PM16SE, and all 1394- and Ultra3000, Kinetix 6000, and 8720 drives).

### 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.

**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 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 <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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'.</li> <li>• 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.</li> <li>• If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.</li> </ul>

### 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 (Kinetix 6000 Drives)** Use this tab to configure the selected 1784-PM16SE motion module by associating axis tags (of the type AXIS\_SERVO\_DRIVE) with nodes available on the module.

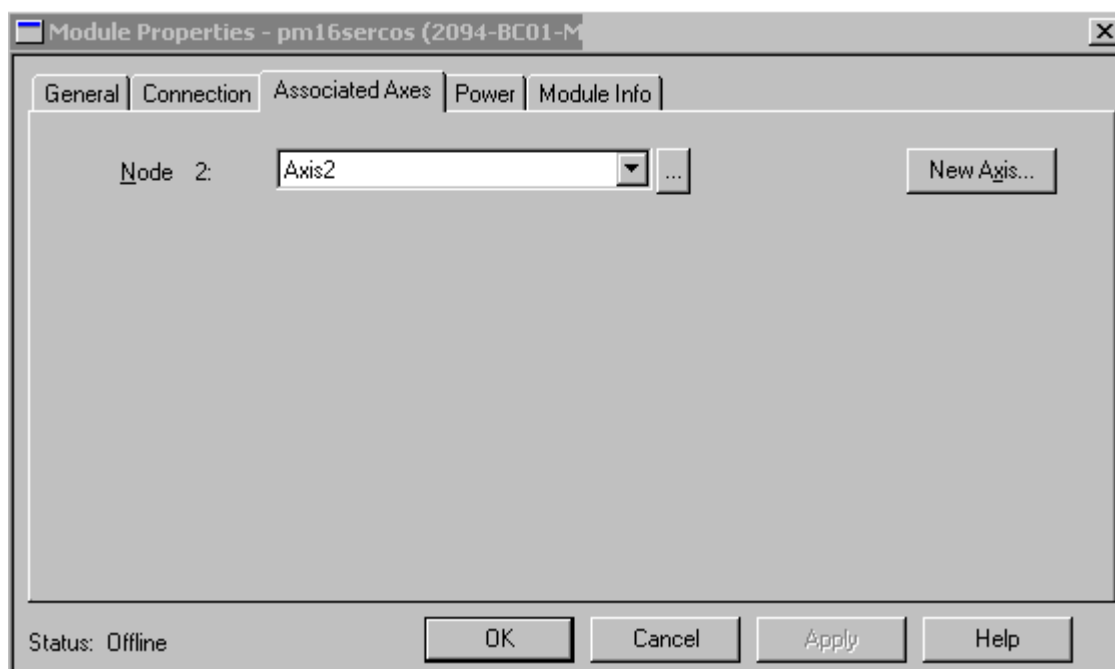


Figure 10.8 Module Properties - Associated Axes Tab

## Node

Displays the selected node of the Kinetix 6000 drive, as entered 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.

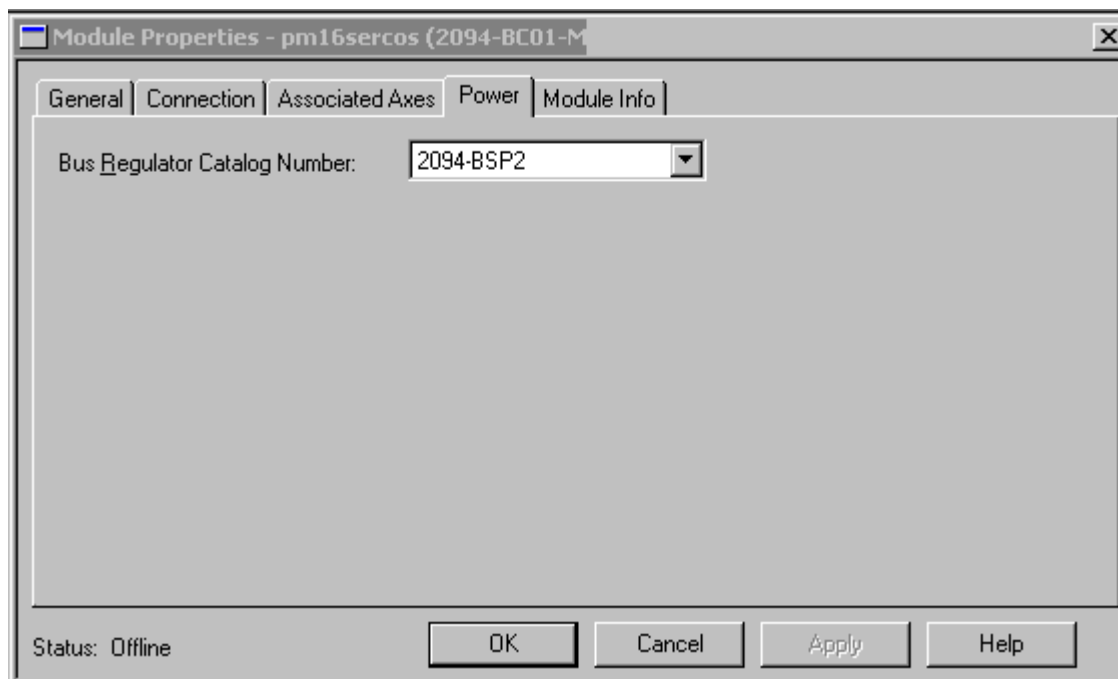
## Ellipsis (...)

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 - Kinetix Drive** Use this tab to select a bus regulator for your Kinetix 6000 drive.



**Figure 10.9 Module Properties - Power Tab**

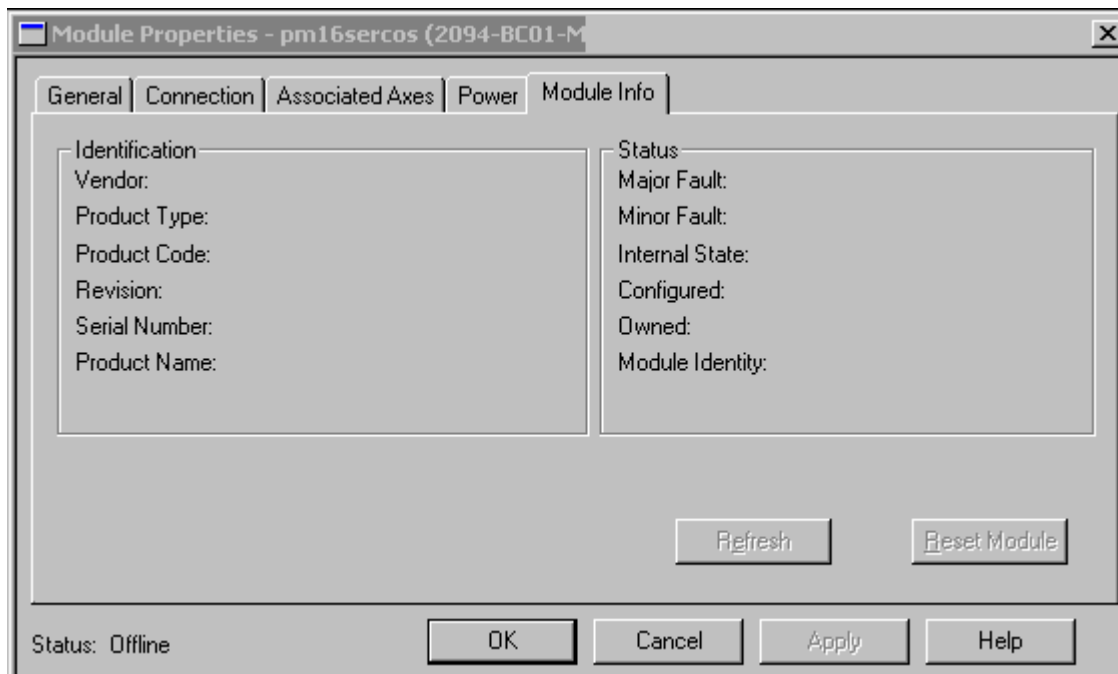
## Bus Regulator Catalog Number

Select the catalog number that describes the bus regulator device used by the Kinetix 6000 drive module.



## Module Info Tab

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.



**Figure 10.10 Module Properties - Module Info**

### 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

<b>If you are configuring a:</b>	<b>This field displays one of the following:</b>
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: <ul style="list-style-type: none"> <li>• Vendor</li> <li>• Module Type (the combination of Product Type and Product Code for a particular Vendor)</li> <li>• Major Revision</li> </ul>
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.

## Configuring an 8720MC Drive

The Allen-Bradley 8720MC Drive System is a family of products designed to satisfy a wide range of machine tool spindle and power servo applications. For applications which do not require line regeneration, Allen-Bradley offers five 380 to 460 VAC input high performance digital drives with current outputs ranging from 21 to 48 amperes. For applications requiring line regeneration, the same five drives plus an additional 14 amp drive can be connected to a regenerative power supply via a 750V DC common bus interface. The complete family includes a set of twelve drive amplifiers capable of controlling a family of motors ranging in power from 5.5 to 93 kW.

The 8720MC Digital Servo Drive with fiber optic SERCOS interface simplifies the integration of the 8720MC with the ControlLogix architecture by providing single point drive commissioning through RSLogix 5000 software and reducing the control wiring to a single fiber optic cable.

You can initiate the configuration of an 8720MC drive module by either of two methods:

1. In the Controller Organizer, in the I/O Configuration branch, select a 1784-PM16SE motion module.
2. In the File menu, select New Component then Module.

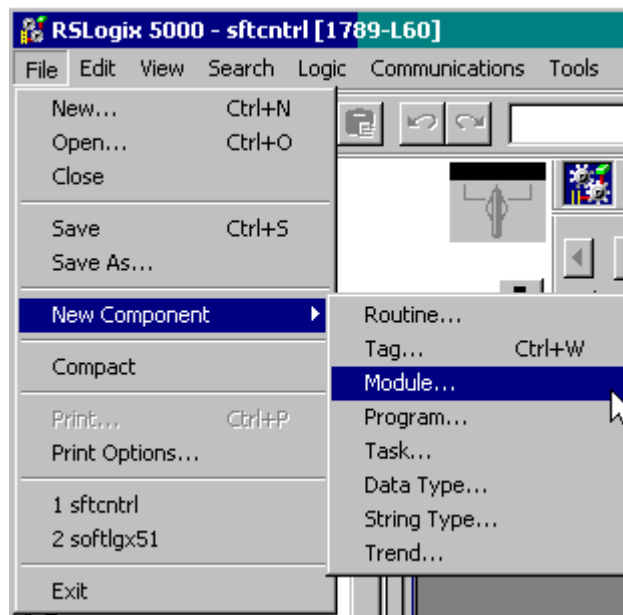
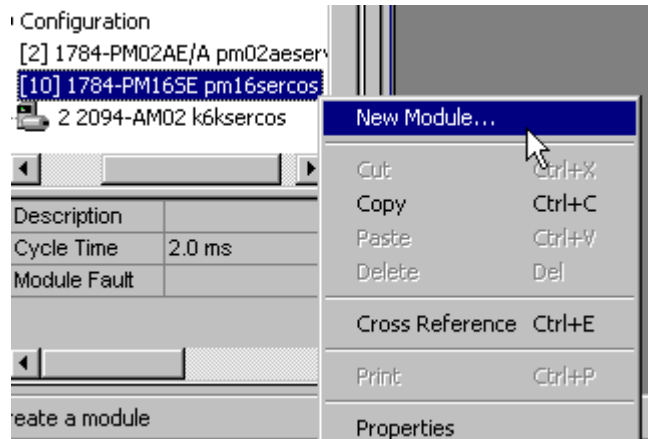


Figure 11.1 File Menu - New Component - Module

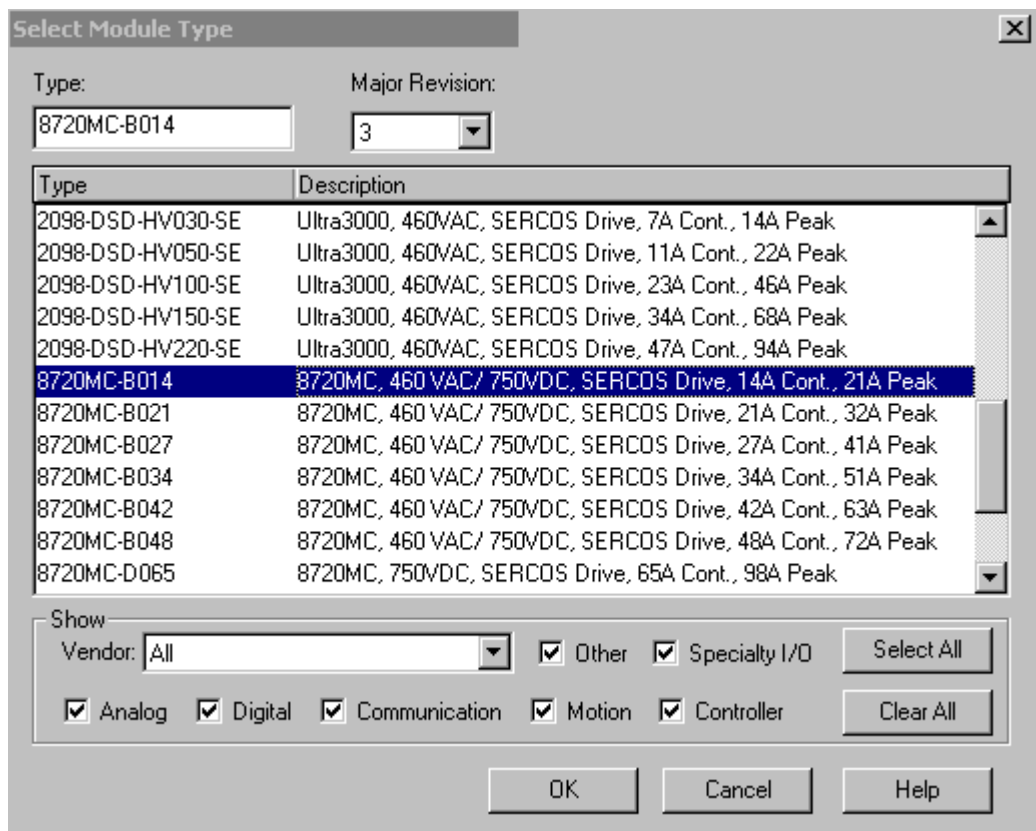
OR

1. Right click on the selected or 1784-PM16SE module in the I/O Configuration branch of the Controller Organizer.
2. Select New Module from the pop up menu.



**Figure 11.2 New Module Selection from Pop Up Menu**

The Select Module Type dialog displays.



**Figure 11.3 Select Module Type Window**

3. In the Select Module Type dialog, select the desired drive module. The 8720MC drives begin with the 8720MC prefix.
4. Press the OK button to close the Select Module Type dialog. The 8720MC Drive Create Wizard Module Properties dialog opens.

The screenshot shows a dialog box titled "Module Properties - pm16sercos (8720MC-B014)". The fields are as follows:

- Type: 8720MC-B014 8720MC, 460 VAC/ 750VDC, SERCOS Drive, 14A Cont., 21A Peak
- Vendor: Allen-Bradley
- Name:
- Node:
- Description:
- Revision:
- Electronic Keying:

At the bottom, there are five buttons: Cancel, < Back, Next >, Finish >>, and Help.

**Figure 11.4 Module Properties Wizard Dialog - Naming the Drive**

5. You must fill in a name for the drive; this is a required field. Fill in the responses for the other parameters as needed.

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.

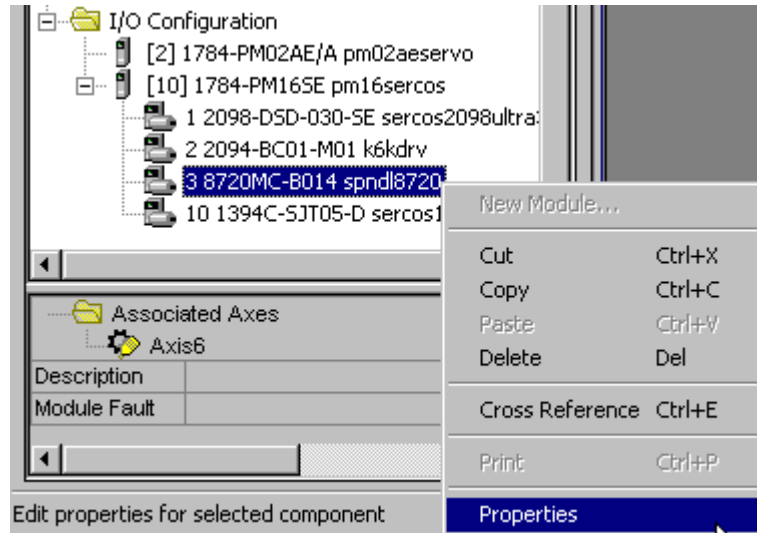
<Back – Click this button to view the previous Create wizard page.

Finish>> – Click this button to close the Create wizard.

Click on the Finish>> button to create the drive list it in the Controller Organizer.

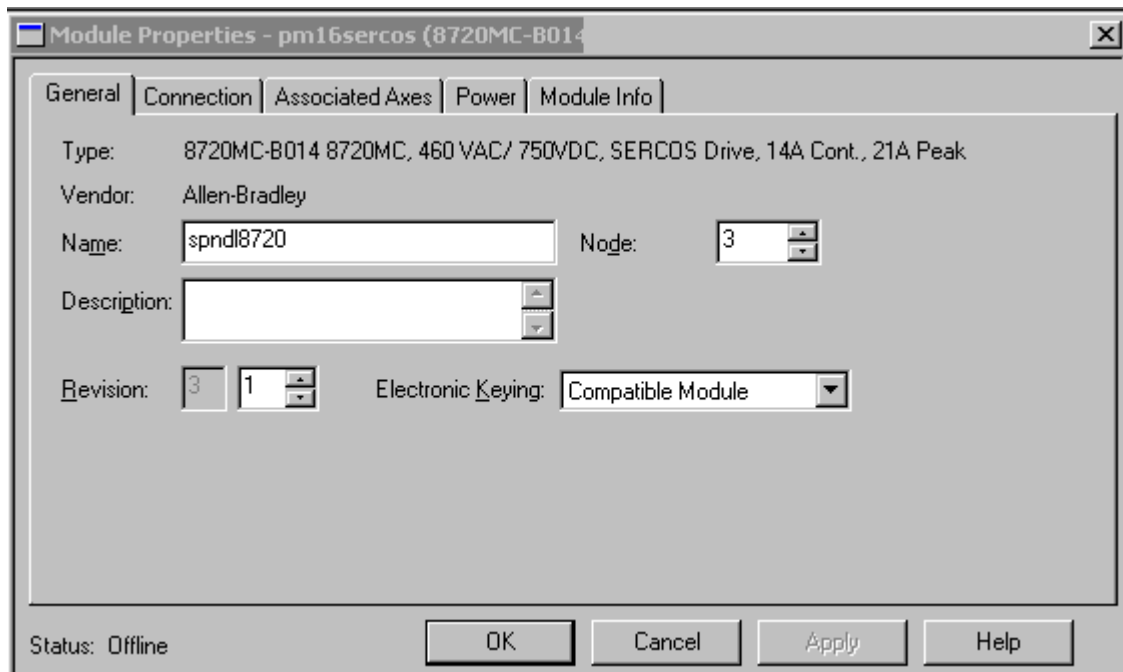
## Editing the 8720MC Drive Properties

The Module Properties for any of the 8720MC drives can be edited by highlighting the drive to be edited, right click with the mouse and selecting Properties.



**Figure 11.5 Accessing the Properties of the Drive**

The Module Properties screen displays.



**Figure 11.6 Module Properties - General Tab**

**General Tab** The General Tab is where you edit the basic values for the 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.

## 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** – does not employ keying at all.

---

**ATTENTION**

Changing the 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

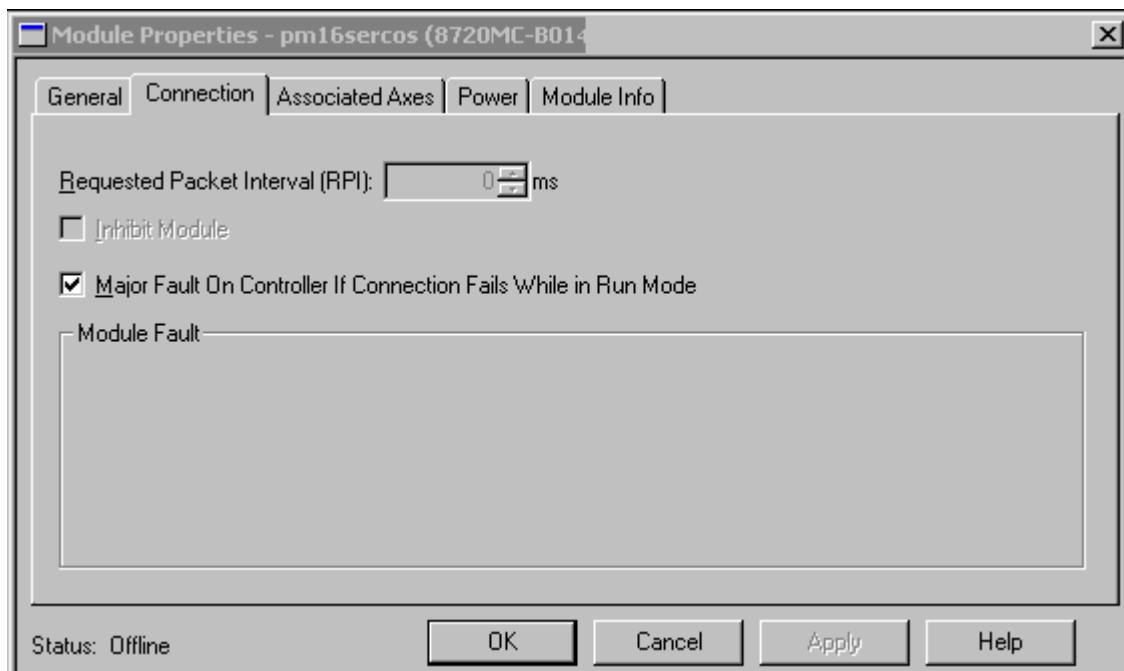
This feature prevents the inadvertent insertion of the wrong module in the wrong slot.

### 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: <ul style="list-style-type: none"> <li>• its parent has not yet made a connection to it</li> <li>• its parent is inhibited</li> <li>• its parent is faulted</li> </ul>
Offline	You are not online.

**Connection Tab** Use this tab to define controller to module behavior.



**Figure 11.7 Module Properties - Connection Tab**

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

Does not apply to this setup. Field is greyed out.

**Note:** This field is disabled for all motion modules (e.g., 1784-PM02AE, 1784-PM16SE modules and all 1394-, Ultra3000, Kinetix 6000, and 8720MC drives).

### 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.

---

**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 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 <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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'.</li> <li>• 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.</li> <li>• If you uninhibit a module while online and a fault condition occurs, a connection is not made to the module.</li> </ul>

### 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.

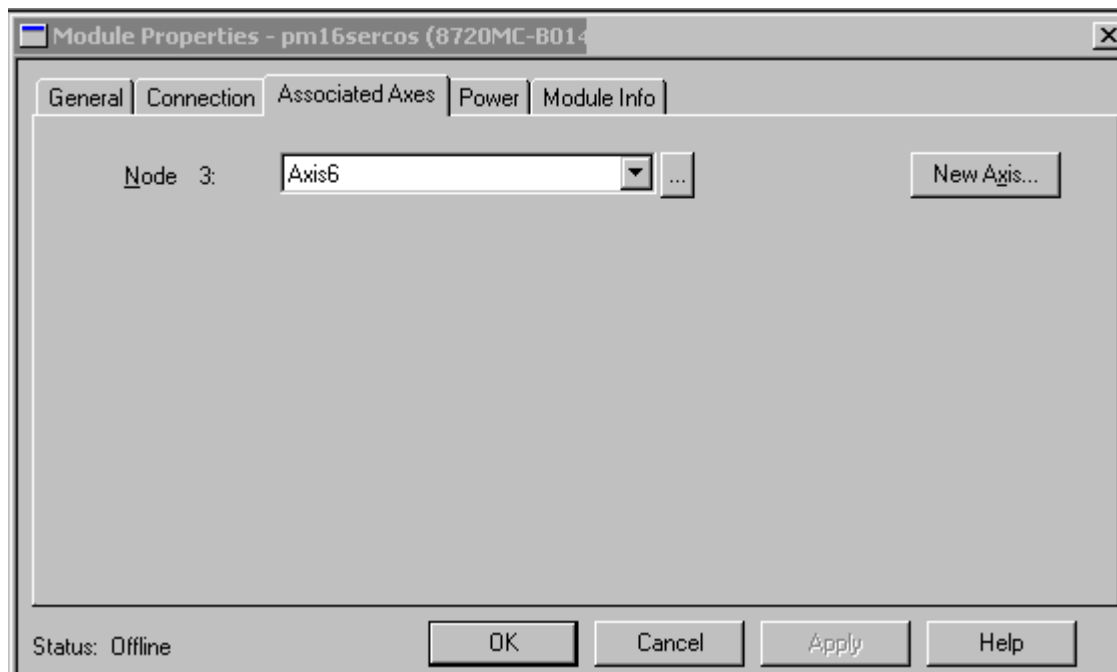
### Module Fault

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 (8720MC Drives)** Use this tab to configure the selected 1784-PM16SE motion module by associating axis tags (of the type AXIS\_SERVO\_DRIVE) with nodes available on the module.



**Figure 11.8 Module Properties - Associated Axes Tab**

### Node

Displays the selected node of the 8720MC drive, as selected on the General tab. This field allows you to associate an AXIS\_SERVO\_DRIVE tag with the driver's node. This field is read-only while you are online.

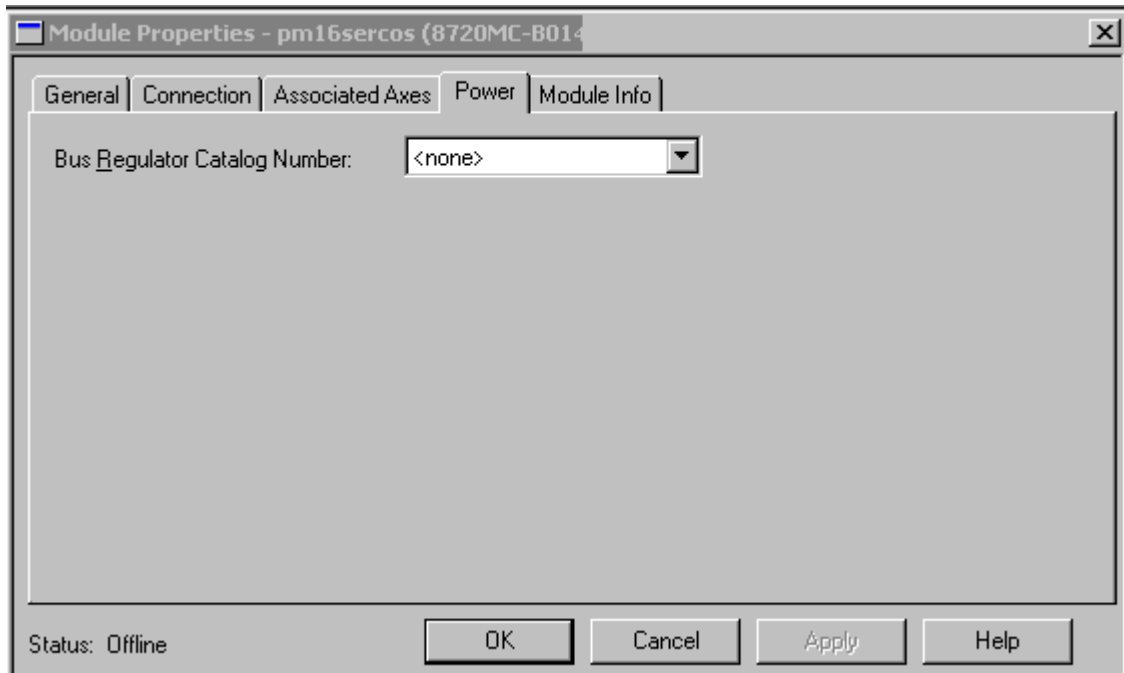
### Ellipsis (...)

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. See the chapter titled *Naming & Configuring Your Motion Axis* for the steps on how to configure a motion axis.

**Power Tab - 8720MC Drive** Use this tab to select a bus regulator for your drive module.



**Figure 11.9 Module Properties - Power Tab**

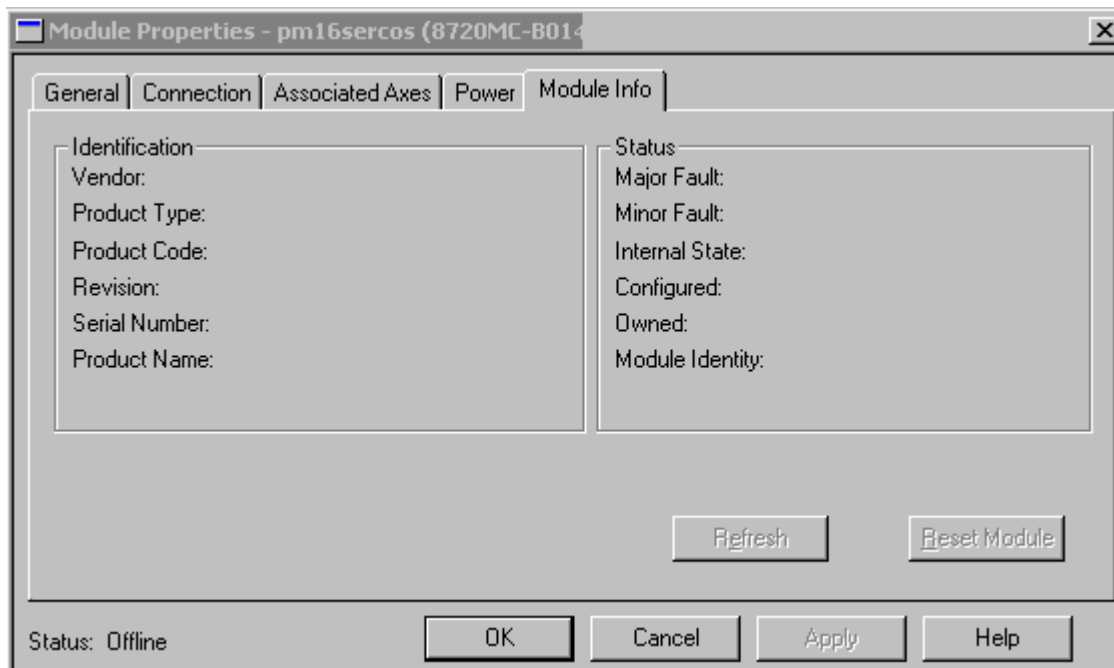
**Note:** The Power Tab does not apply to the 8720MC SERCOS drives.

### **Bus Regulator ID**

**Note:** This parameter does not apply to the 8720MC SERCOS drives. The only available selection in the pull-down menu is <none>.

## Module Info Tab

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.



**Figure 11.10 Module Properties - Module Info**

### 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.

## Major/Minor Fault Status

<b>If you are configuring a:</b>	<b>This field displays one of the following:</b>
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: <ul style="list-style-type: none"> <li>• Vendor</li> <li>• Module Type (the combination of Product Type and Product Code for a particular Vendor)</li> <li>• Major Revision</li> </ul>
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 (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.

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 motion instructions for RSLogix 5000 programming software.

The motion instructions for the RSLogix 5000 programming software consist of six 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.
- **Motion Coordinated Move instructions** – to control multi-axis coordinated motion.
- **Motion Direct Commands.**

For more information about motion instructions, refer to the *Logix5000 Controller Motion Instruction Set Reference Manual* (publication 1756-RM007).

### Motion State Instructions

Motion state instructions directly control or change the operating state of an axis.

The motion state instructions are:

Instruction	Abbreviation	Description
Motion Servo On	MSO	Enables the servo drive and activates the axis servo loop
Motion Servo Off	MSF	Disables the servo drive and deactivates the axis servo loop
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.

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 close.
Motion Direct Drive On	MDO	Enables the servo drive and sets the servo output voltage of an axis
Motion Direct Drive Off	MDF	Disables the servo drive and sets the servo output voltage to the output offset voltage
Motion Axis Fault Reset	MAFR	Clears all motion faults

For more information about motion state instructions, refer to the *Motion State Instructions* chapter of the *Logix5000 Controller Motion Instruction Set Reference Manual*, publication 1756-RM007.

## Motion Move Instructions

Motion move instructions control all aspects of axis position.

The motion move instructions are:

Instruction	Abbreviation	Description
Motion Axis Stop	MAS	Initiates a controlled stop of any motion process on an axis
Motion Axis Home	MAH	Homes an axis
Motion Axis Jog	MAJ	Initiates a jog motion profile for an axis
Motion Axis Move	MAM	Initiates a move profile for an axis
Motion Axis Gear	MAG	Enables electronic gearing between two axes
Motion Change Dynamics	MCD	Changes the speed, acceleration rate, or deceleration rate of a move profile or jog profile in progress
Motion Redefine Position	MRP	Changes the command or actual position of an axis
Motion Calculate Cam Profile	MCCP	Calculates a Cam Profile based on an array of cam points.
Motion Axis Position Cam	MAPC	Performs electronic camming between any two axes designated in the specified Cam Profile.
Motion Axis Time Cam	MATC	Performs electronic camming as a function of time designated in the specified Cam Profile.

For more information about motion state instructions, refer to the *Motion Move Instructions* chapter of *Logix5000 Controller Motion Instruction Set Reference Manual*, publication 1756-RM007.

## Motion Group Instructions

Motion group instructions initiate action on all axes in a group.

The motion group instructions are:

Instruction	Abbreviation	Description
Motion Group Stop	MGS	Initiates a stop of motion on a group of axes
Motion Group Shutdown	MGSD	Forces all the axes in a group into the shutdown operating state
Motion Group Shutdown Reset	MGSR	Transitions a group of axes from the shutdown operating state to the axis ready operating state
Motion Group Strobe Position	MGSP	Latches the current command and actual positions of all the axes in a group

For more information about motion state instructions, refer to the *Motion Group Instructions* chapter of *Logix5000 Controller Motion Instruction Set Reference Manual*, publication 1756-RM007.

## 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
Motion Arm Watch Position	MAW	Arms watch-position event checking for an axis
Motion Disarm Watch Position	MDW	Disarms watch-position event checking for an axis
Motion Arm Registration	MAR	Arms servo module registration event checking for an axis
Motion Disarm Registration	MDR	Disarms servo module registration event checking for an axis
Motion Arm Output Cam	MAOC	Arms an Output Cam for a particular Axis and Output as determined by the operands for the instruction.
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.

For more information about motion state instructions, refer to the *Motion Event Instructions* chapter of *Logix5000 Controller Motion Instruction Set Reference Manual*, publication 1756-RM007.

## 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
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.
Motion Run Axis Tuning	MRAT	Commands the servo module to run a tuning motion profile for an axis
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.
Motion Run Hookup Diagnostic	MRHD	Commands the servo module to run one of three diagnostic tests on an axis

For more information about motion state instructions, refer to the *Motion Configuration Instructions* chapter of *Logix5000 Controller Motion Instruction Set Reference Manual*, publication 1756-RM007.

## Motion Coordinated Move Instructions

The multi-axis coordinated move motion instructions are the vehicle for performing linear and circular moves in single and multi-dimensional Cartesian space.

The motion configuration instructions are:

Instruction	Abbreviation	Description
Motion Coordinated Linear Move	MCLM	Initiates a single or multi-dimensional linear coordinated move for the specified axes within a Cartesian coordinate system.
Motion Coordinated Circular Move	MCCM	Initiates a two- or three-dimensional circular coordinated move for the specified axes within a Cartesian coordinate system.
Motion Coordinated Change Dynamics	MCCD	Initiates a change in path dynamics for coordinate motion active on the specified coordinate system.
Motion Coordinated Stop	MCS	Initiates a controlled stop of the specified coordinate motion profile taking place on the designated coordinate system.
Motion Coordinated Shutdown	MCS D	Initiates a controlled shutdown of all of the axes of the specified coordinate system.
Motion Coordinated Shutdown Reset	MCS R	Initiates a reset of all of the axes of the specified coordinate system from the shutdown state to the axis ready state and clear the axis faults.



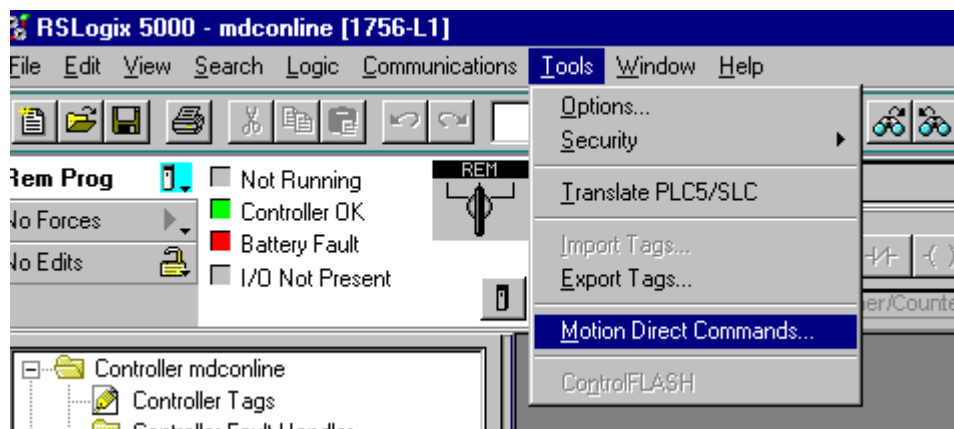
## Motion Direct Commands

The Motion Direct Commands feature lets you issue motion commands while you are online without having to write or execute an application program. Motion Direct Commands are particularly useful when you are commissioning or debugging a motion application. During commissioning, you can configure an axis and monitor the behavior using Trends in the Controller Organizer. Use of Motion Direct Commands can “fine-tune” the system with or without load to optimize its performance. When in the testing and or debugging cycle, you can issue Motion Direct Commands to establish or reestablish conditions such as Home. Often during initial development or enhancement to mature applications you need to test the system in small manageable areas. These areas can include - Home to establish initial conditions, Incrementally Move to a physical position, and monitor system dynamics under specific conditions.

## Accessing Direct Commands

The Motion Direct Command dialog can be accessed from the Tools pull-down of the Main Menu, by right clicking on the Group in the Controller Organizer, and by right clicking on an Axis in the Controller Organizer. The point of entry determines the look of the opening dialog and the default values that are set.

**From the Main Menu** You can access the Motion Direct Commands dialog directly from the Tool pull-down of the Main Menu.



**Figure 12.1 Main Menu | Tools Pull-down | Motion Direct Commands**

When you access the Motion Direct Commands dialog from the Tools pull-down, it defaults to the MSO command and the Axis field is defaulted to a question mark (?).

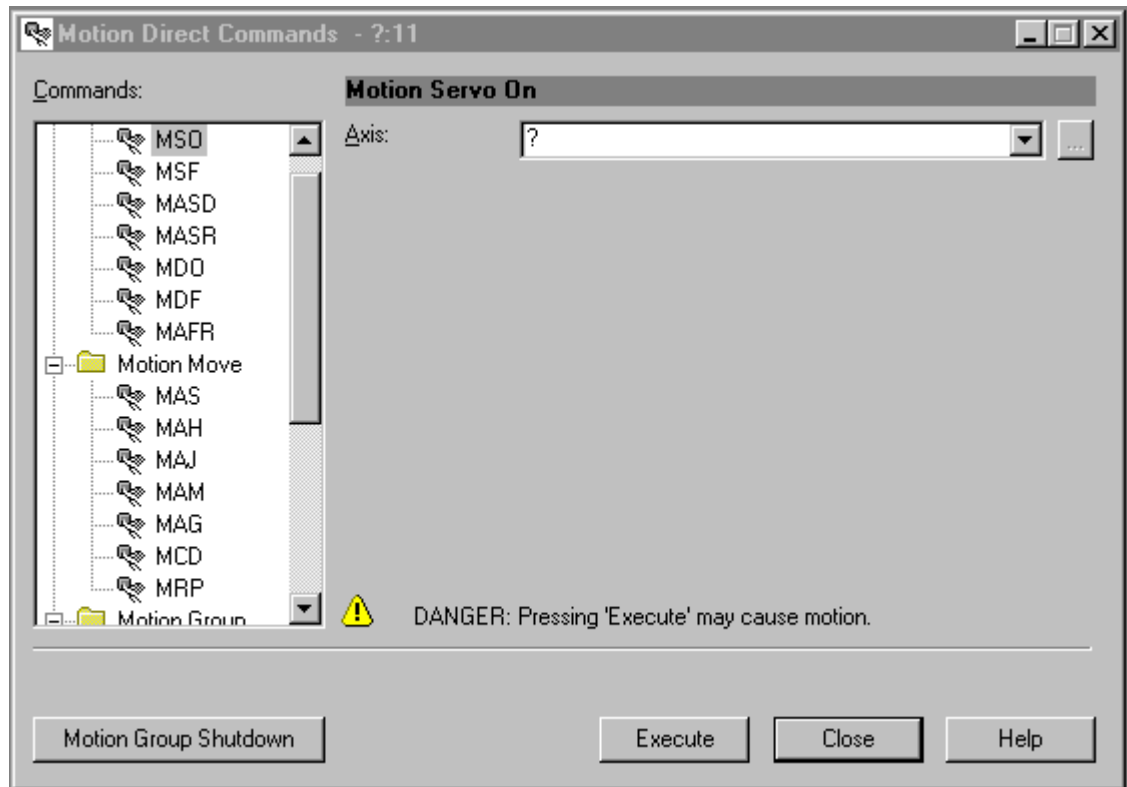


Figure 12.2 Motion Direct Command Dialog from Tool Menu

**From Group in the Controller Organizer** You can access the Motion Direct Commands by right clicking on the Group in the Controller Organizer. This is the recommended way when you want to invoke a Motion Group Instruction.

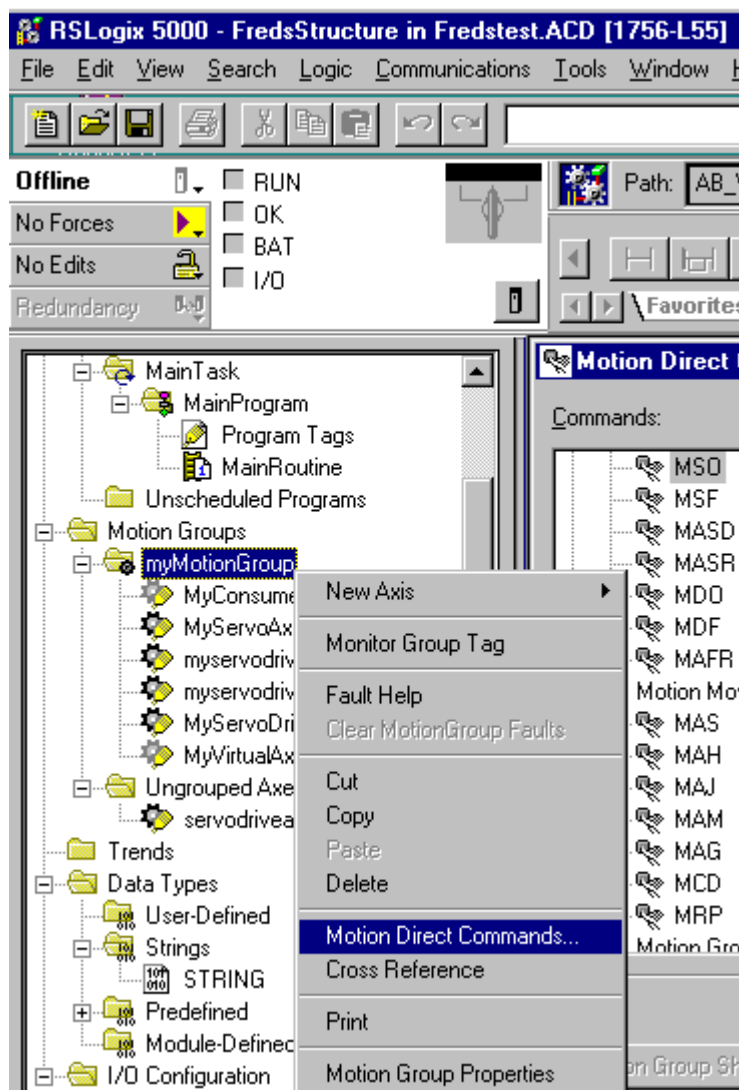


Figure 12.3 Controller Organizer | Group | Motion Direct Commands

When the Motion Direct Commands dialog is accessed from the Motion Group in the Controller Organizer, the Motion Group field defaults to the group you right clicked on and the MGS command is the default selection.

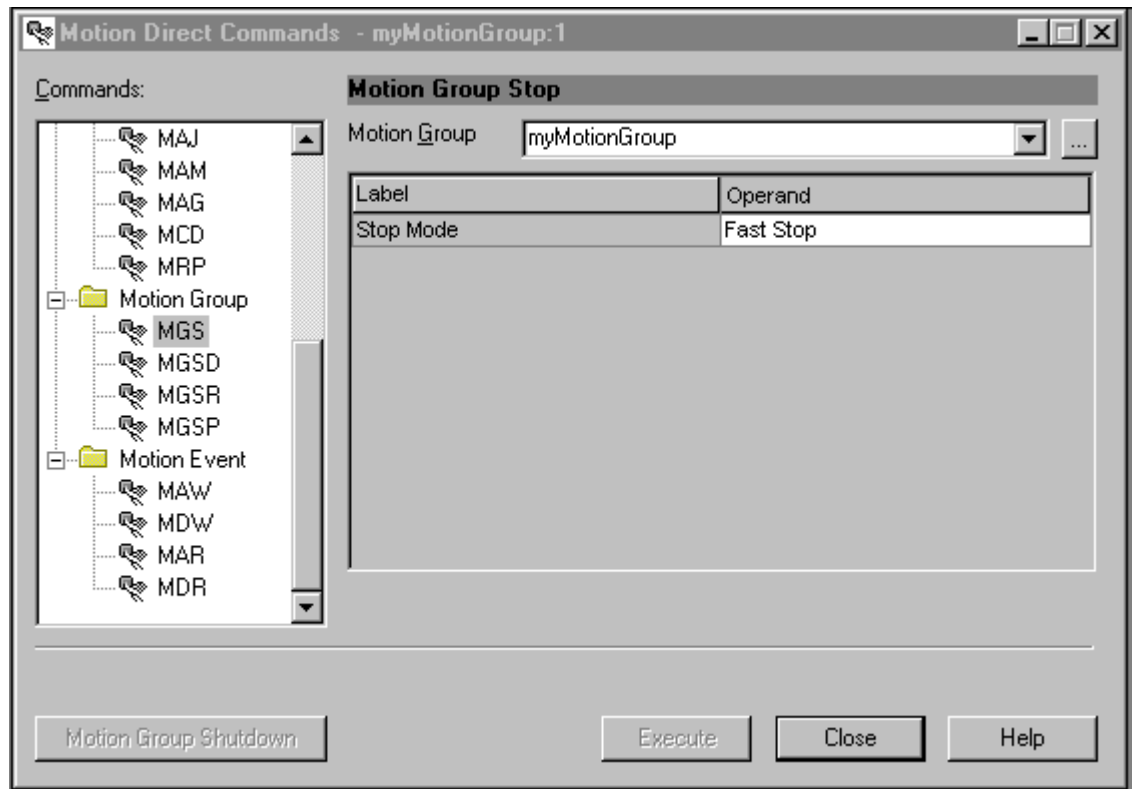
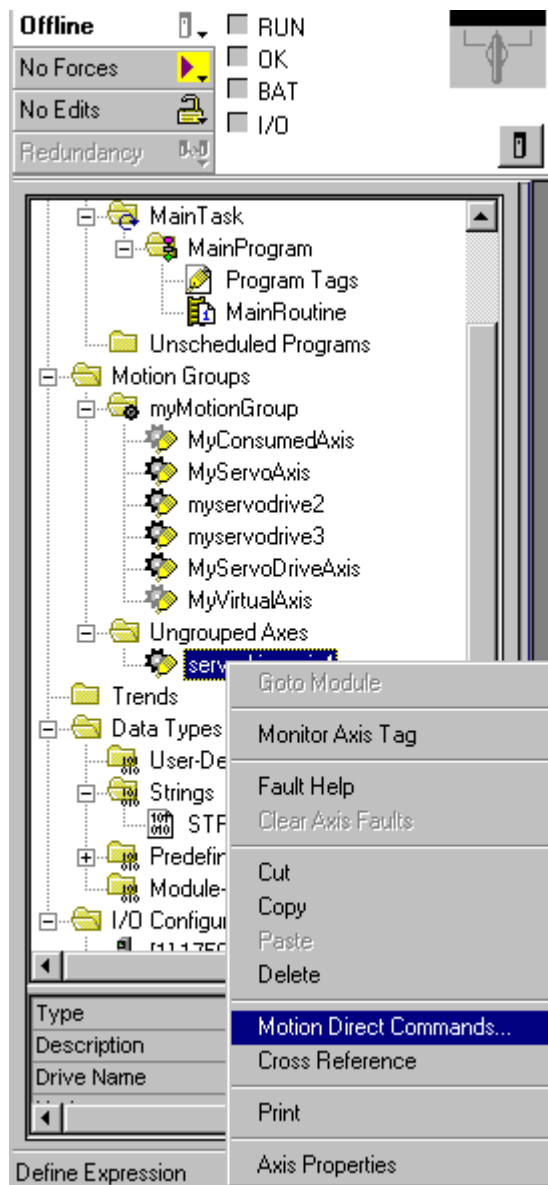


Figure 12.4 Motion Direct Command Dialog from Motion Group

**From Axis in the Controller Organizer** You can access the Motion Direct Commands by right clicking on an Axis in the Controller Organizer. This is the recommended way when you want to invoke a Motion Instruction for an axis.



**Figure 12.5 Controller Organizer | Axis | Motion Direct Commands**

When the Motion Direct Commands dialog is accessed from an Axis in the Controller Organizer, the Axis field defaults to the axis you right clicked on and the MSO command is the default selection.

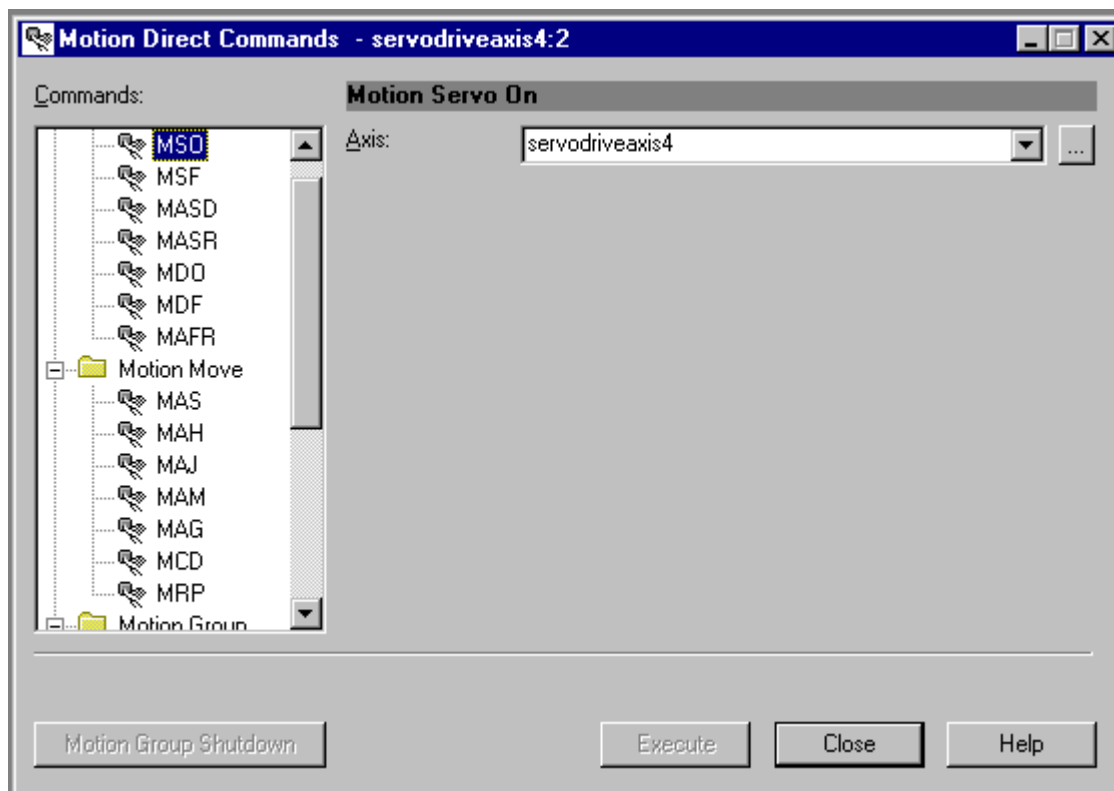


Figure 12.6 Motion Direct Command Dialog from Axis

## Supported Commands

The list of instructions supported by the Motion Direct Commands feature include:

### Motion State

Command	Description
MSO	Enable the servo drive and activate the axis servo loop.
MSF	Disable the servo drive and deactivate the axis servo loop.
MASD	Force an axis into the shutdown operating state. Once the axis is in the shutdown operating state, the controller blocks any instructions that initiate axis motion.
MASR	Change 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 close.

Command	Description
MDO	Enable the servo drive and set the servo output voltage of an axis.
MDF	Disable the servo drive and set the servo output voltage to the output offset voltage.
MAFR	Clear all motion faults for an axis.

### Motion Move

Command	Description
MAS	Initiate a controlled stop of any motion process on an axis.
MAH	Home an axis.
MAJ	Initiate a jog motion profile for an axis.
MAM	Initiate a move profile for an axis.
MAG	Provide electronic gearing between any two axes
MCD	Change the speed, acceleration rate, or deceleration rate of a move profile or a jog profile in progress.
MRP	Change the command or actual position of an axis.

### Motion Group

Command	Description
MGS	Initiate a stop of motion on a group of axes.
MGSD	Force all axes in a group into the shutdown operating state.
MGSR	Transition a group of axes from the shutdown operating state to the axis ready operating state.
MGSP	Latch the current command and actual position of all axes in a group.

### Motion Event

Command	Description
MAW	Arm watch-position event checking for an axis.
MDW	Disarm watch-position event checking for an axis.
MAR	Arm servo-module registration-event checking for an axis.
MDR	Disarm servo-module registration-event checking for an axis.

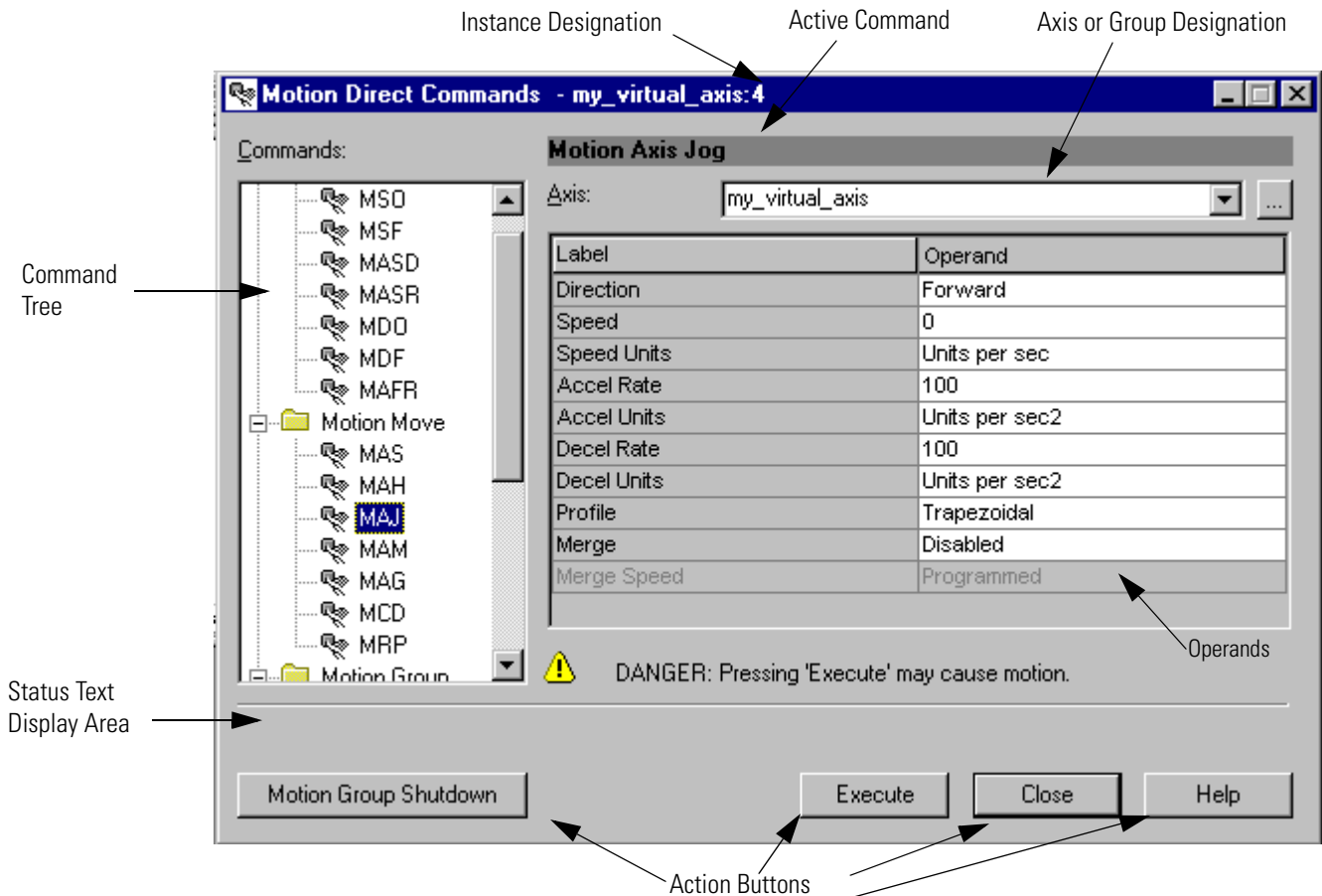
For more information about the use and operation of Motion Direct Commands see the *Logix5000 Controller Motion Instruction Set Reference Manual*, publication number 1756-RM007.

## Motion Direct Command Dialog

The Motion Direct Commands dialog is similar in position and behavior to other dialogs in RSLogix5000. The dialog can be accessed when the system is either off-line or on-line.

### Motion Direct Command Dialog On-line

In order to execute a Motion Direct Command, you must be on-line. The on-line dialog has the Motion Group Shutdown and Execute buttons active. If you click on either of these, action is taken immediately.



**Figure 12.7 Motion Direct Command Dialog (on-line)**

When the Motion Direct Command dialog is opened, focus is given to the Command Tree. In the Command list, you can either type the mnemonic and the list advances to the closest match or you can scroll down the list to select a command. Click on the desired command and its dialog displays.



At the top of the dialog, in the title bar, there is a number at the end of the axis or group that the command is being applied upon. This is the Instance reference number. This number increases by one every time a command is accessed for that axis or group. The number is cleared when you execute RSLogix.

Located at the bottom of the dialog are the following buttons: Motion Group Shutdown, Execute, Close, and Help.

### **Motion Group Shutdown Button**

The Motion Group Shutdown button is located to the left of the screen to avoid accidental invoking of this command when you really want to execute the command accessed from the Command tree. Clicking on this button causes the Motion Group Shutdown instruction to execute. If you click on the Motion Group Shutdown button and it is successfully executed, a Result message is displayed in the results window below the dialog. Since the use of this button is an abrupt means of stopping motion, an additional message is displayed in the error text field. The message "MOTION GROUP SHUTDOWN executed!" is displayed with the intention of giving greater awareness of the execution of this command. If the command fails then an error is indicated as per normal operation. (See Error Conditions later in this chapter.)

There is space above the Motion Group Shutdown button and below the line where status text is displayed when a command is executed.

### **Execute Button**

Clicking the Execute button verifies the operands and initiates the current Motion Direct Command. Verification and error messages display as the

### **Close Button**

To end a Motion Direct Command session, click on the Close button. The data is not saved and the command is not executed. It acts the same as a Cancel button.

### **Help Button**

Click on the Help button to access the on-line Help.



## Motion Object Attributes

The Motion Object Attributes are included in this manual to provide you with a greater understanding of how the system works. Your familiarity with these attributes allows you to take greater advantage of the flexibility inherent in the RSLogix software. The Axis Object Interface Attributes comprise all the axis object attributes that are used by RSLogix 5000 to support the interface to the axis object including configuration attributes used in customizing many of the configuration screens and motion instructions that operate on the axis object.

### Object Support Attributes

The following attributes are used by software to establish the interfaces and structure of the motion axis object instance.

**Axis Instance** The Axis Instance attribute returns the instance number of an axis. Major fault records generated for an axis contain only the instance of the offending axis. This attribute is used to determine if this was the offending axis; i.e. if the instance number matches.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Axis Instance	DINT	Instance Number assigned to Axis

**Group Instance** The Assigned Group Instance attribute is used to determine what motion group object instance the axis is assigned to.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Assigned Group Instance	DINT	Instance Number of Group assigned to Axis

**Map Instance** The axis is associated to a specific motion compatible module by specifying the instance of the map entry representing the module.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Map Instance	DINT	I/O Map Instance Number. It is 0 for virtual and consumed Data Types.

**Module Channel** The axis is associated to a specific channel on a motion compatible module by specifying the Module Channel attribute.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Module Channel	SINT	Zero based channel number of the module. 0xff, indicates unassigned.

**Module Class Code** The ASA class code of the object in the motion module which is supporting motion; e.g., 0xAF is the ASA object ID of the “Servo Module Axis Object” residing in the 1784-PM02AE module.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Module Class Code	DINT	ASA Object class code of the motion engine in the module; e.g., 0xAF for the M02AE module.

**C2C Map Instance** When the Axis Data Type attribute is specified to be ‘Consumed’ then this axis is associated to the consumed data by specifying both the C2C Map Instance and the C2C Connection Instance. For all other Axis Data Types if this axis is to be produced then this attribute is set to 1 (one) to indicate that the connection is off of the local controller’s map instance.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	C2C Map Instance	SINT	Producer/Consumed axis’s associated C2C map instance

**C2C Connection Instance** When Axis Data Type is specified to be ‘Consumed’ then this axis is associated to the consumed data by specifying both the C2C Map Instance and the C2C Connection Instance. This attribute is the connection instance under the C2C map instance, which provides the axis data being sent to it from another axis via a C2C connection.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	C2C Connection Instance	SINT	Producer/Consumed axis's associated C2C connection instance in reference to the C2C map instance

For all other Axis Data Types if this axis is to be produced then this attribute is set to the connection instance under the local controller's map instance (1) that will be used to send the remote axis data via the C2C connection.

**Memory Use** RSLogix 5000 software uses this attribute to create axis instances in I/O memory for axes that are either to be produced or consumed. The Memory Use attribute can only be set as part of an axis create service and is used to control which controller memory the object instance is created in.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Memory Use	INT	Controller memory space where instance exists. 105 (0x69) = I/O space 106 (0x6a) = Data Table space

**Memory Usage** The Memory Use attribute can be used to determine the amount of memory the created instance consumes in bytes.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Memory Usage	DINT	Amount of memory consumed for this instance (in bytes)

**Axis Data Type** The Axis Data Type attribute and is used to determine which data template, memory format, and set of attributes are created and applicable for this axis instance. This attribute can only be set as part of an axis create service.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Data Type	SINT	Associated motion axis tag data type: 0 = Feedback 1 = Consumed 2 = Virtual 3 = Generic 4 = Servo 5 = Servo Drive.

**Axis Configuration State** The Axis Configuration State attribute is used for debugging purposes to indicate where in the axis configuration state-machine this axis presently is. Even consumed and virtual axes utilize this attribute.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Configuration State	SINT	State of the axis configuration state machine

**Axis State** The Axis State attribute indicates the operating state of the axis. Possible states are axis-ready, direct drive control, servo control, axis faulted and axis shutdown.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis State	SINT	Axis State: 0 = Axis Ready 1 = Direct Drive Control 2 = Servo Control 3 = Axis Faulted 4 = Axis Shutdown

**Output Cam Execution Targets** The Output Cam Execution Targets attribute is used to specify the number of Output Cam nodes attached to the axis. This attribute can only be set as part of an axis create service and dictates how many Output Cam Nodes are created and associated to that axis. Each Output Cam Execution Target requires approximately 5.4k bytes of data table memory to store persistent data. With four Output Cam Execution Targets per axis, an additional 21.6k bytes of memory is required for each axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Output Cam Execution Targets	DINT	Represents the number of Output Cam nodes attached to this axis. Valid range = 0-8 with default of 0.

The ability to configure the number of Output Cam Execution Targets for a specific axis reduces the memory required per axis for users who do not need Output Cam functionality, or only need 1 or 2 Output Cam Execution Targets for a specific axis. Each axis can be configured differently.

## Motion Object Status Attributes

The Motion Status Attributes are comprised of all Motion Axis Object variables that are “read-only”, i.e. attributes that you can get from the axis object, but not set. This general category does not include attributes specific to the software interface, which are covered in the section under Software Interface Attributes.

## Motion Status Attributes

The Motion Status Attributes associated with the Axis Object provide access to the current and historical position velocity, and acceleration information of the axis. These values may be used as part of the user program to implement sophisticated real time computations associated with motion control applications. A list of all Motion Status Attributes is shown in the tables below.

For all of the Motion Status attributes to return a meaningful value, the 'Conversion Constant' Axis Configuration Attribute must be established. Furthermore, attributes having velocity units (Position Units / Sec) must also have a valid coarse update period which is established through association with a fully configured Motion Group Object.

All Motion Status attributes support Direct Tag Access via RSLogix5000 software. Thus, a Motion Status attribute may be directly referenced in a program as <axis tag name>.<motion status tag name>. An example of this might be FeedAxis.ActualPosition.

To avoid the unnecessary processor effort associated with real-time conversion of certain Motion Status tags that are not of interest to the user, it is necessary to explicitly activate real-time update of these attributes via the Auto Tag Update attribute of the associated motion group. A subset of the Motion Status attributes must have the Auto Tag Update attribute enabled or the tag value will be forced to zero. The following Motion Status attributes are affected:

- Actual Position
- Actual Velocity
- Actual Acceleration
- Master Offset
- Command Position
- Command Velocity
- Command Acceleration
- Average Velocity.

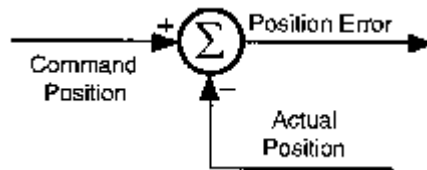
**Actual Position** Actual Position is the current absolute position of an axis, in the configured Position Units of that axis, as read from the feedback transducer. Note, however, that this value is based on data reported to the ControlLogix Processor as part of an ongoing synchronous data transfer process which results in a delay of one coarse update period. Thus, the Actual Position value that is obtained is the actual position of the axis one coarse update period ago.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Actual Position	REAL	Position Units

**Command Position** Command Position is the desired or commanded position of a physical axis, in the configured Position Units of that axis, as generated by the controller in response to any previous motion Position Control instruction. Command Position data is transferred by the ControlLogix Processor to a physical axis as part of an ongoing synchronous data transfer process which results in a delay of one coarse update period. Thus, the Command Position value that is obtained is the command position that will be acted upon by the physical servo axis one coarse update period from now.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Command Position	REAL	Position Units

The figure below shows the relationship between Actual Position, Command Position, and Position Error for an axis with an active servo loop. Actual Position is the current position of the axis as measured by the feedback device (e.g., encoder). Position error is the difference between the Command and Actual Positions of the servo loop, and is used to drive the motor to make the actual position equal to the command position.



**Figure 13.1 Actual and Command Position Error Relationship**

Command position is useful when performing motion calculations and incremental moves based on the current position of the axis while the axis is moving. Using command position rather than actual position avoids the introduction of cumulative errors due to the position error of the axis at the time the calculation is performed.

**Strobe Position** Strobe Actual Position, and Strobe Command Position are used to simultaneously store a snap-shot of the actual, command position and master offset position of an axis when the MGSP (Motion Group Strobe Position) instruction is executed. The values are stored in the configured Position Units of the axis. Refer to the AC Motion Instruction Specification for a detailed description of the MGSP instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Strobe Actual Position	REAL	Position Units
GSV	Strobe Command Position	REAL	Position Units



Since the MGSP instruction simultaneously stores the actual and command positions for all axes in the specified group of axes, the resultant Strobe Actual Position and Strobe Command Position values for different axes can be used to perform real time calculations. For example, the Strobe Actual Positions can be compared between two axis to provide a form of “slip compensation” in web handling applications.

**Start Position** Whenever a new motion planner instruction starts for an axis (for example, using a MAM instruction), the value of the axis command position and actual position is stored at the precise instant the motion begins. These values are stored as the Start Command Position and Start Actual Position respectively in the configured Position Units of the axis.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Start Actual Position	REAL	Position Units
GSV	Start Command Position	REAL	Position Units

Start Positions are useful to correct for any motion occurring between the *detection* of an event and the action *initiated* by the event. For instance, in coil winding applications, Start Command Positions can be used in an expression to compensate for overshooting the end of the bobbin before the gearing direction is reversed. If you know the position of the coil when the gearing direction was *supposed* to change, and the position at which it *actually* changed (the Start Command Position), you can calculate the amount of overshoot, and use it to correct the position of the wire guide relative to the bobbin.

**Average Velocity** Average Velocity is the current speed of an axis in the configured Position Units per second of the axis. Unlike the Actual Velocity attribute value, it is calculated by averaging the actual velocity of the axis over the configured Average Velocity Timebase for that axis. Average velocity is a signed value with the sign indicating the direction the axis is currently moving.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Average Velocity	REAL	Position Units / Sec

The resolution of the Average Velocity variable is determined by the current value of the Averaged Velocity Timebase parameter, and the configured Conversion Constant (feedback counts per Position Unit) for the axis. The greater the Average Velocity Timebase value, the better the speed resolution, but the slower the response to changes in speed.

The Average Velocity resolution in Position Units per second may be calculated using the equation below.

$$\frac{1}{\text{Averaged Velocity Timebase [Seconds]} \times K} \left[ \frac{\text{Feedback Counts}}{\text{Position Unit}} \right]$$

For example, on an axis with position units of inches and a conversion constant (K) of 20000, an averaged velocity time-base of 0.25 seconds results in an average velocity resolution of:

$$\frac{1}{0.25 \times 20000} = 0.0002 \frac{\text{Inches}}{\text{Second}} = 0.012 \frac{\text{Inches}}{\text{Minute}}$$

The minimum Average Velocity Timebase value is Coarse Update period defined by the associated Motion Group Object. See the *Motion Configuration Attribute* section of this document for more information on setting the Averaged Velocity Timebase and the Conversion Constant parameters.

**Actual Velocity** Actual Velocity is the current instantaneously measured speed of an axis, in the configured axis Position Units per second. It is calculated as the current increment to the actual position per coarse update interval. Actual Velocity is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Actual Velocity	REAL	Position Units / Sec

Actual Velocity is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the GSV/SSV resolution limit on actual velocity is 1 feedback counts per coarse update.

**Command Velocity** Command Velocity is the commanded speed of an axis, in the configured axis Position Units per second, as generated by any previous motion instructions. It is calculated as the current increment to the command position per coarse update interval. Command Velocity is a signed value—the sign (+ or -) depends on which direction the axis is being commanded to move.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Command Velocity	REAL	Position Units / Sec

Command Velocity is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the GSV/SSV resolution limit on command velocity is 0.00001 feedback counts per coarse update.

**Actual Acceleration** Actual Acceleration is the current instantaneously measured acceleration of an axis, in the configured axis Position Units per second per second. It is calculated as the current increment to the actual velocity per coarse update interval. Actual Acceleration is a signed value — the sign (+ or -) depends on which direction the axis is currently accelerating.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Actual Acceleration	REAL	Position Units / Sec <sup>2</sup>

Actual Acceleration is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the GSV/SSV resolution limit on actual velocity is 1 feedback counts per coarse update period per coarse update period.

**Command Acceleration** Command Acceleration is the commanded speed of an axis, in the configured axis Position Units per second per second, as generated by any previous motion instructions. It is calculated as the current increment to the command velocity per coarse update interval. Command Acceleration is a signed value—the sign (+ or -) depends on which direction the axis is being commanded to move.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Command Acceleration	REAL	Position Units / Sec <sup>2</sup>

Command Acceleration is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the GSV/SSV resolution limit on command velocity is 0.00001 feedback counts per coarse update period per coarse update period.

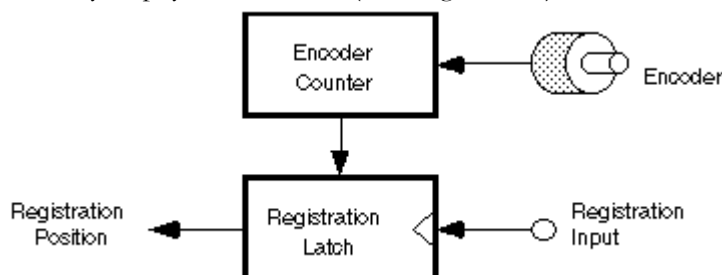
**Watch Position** Watch Position is the current set-point position of an axis, in the configured axis Position Units, as set up in the last, most recently executed, MAW (Motion Arm Watch) instruction for that axis. Refer to the AC Motion Instruction Specification for a detailed description of the MAW instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Watch Position	REAL	Position Units

**Registration Position** Two registration position attributes are provided to independently store axis position associated with two different registration input events. The Registration Position value is the absolute position of a physical or virtual axis (in the position units of that axis) at the occurrence of the most recent registration event for that axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Registration 1 Position	REAL	Position Units
GSV	Registration 2 Position	REAL	Position Units

The figure below shows how the registration position is latched by the registration input when a registration event occurs. The latching mechanism can be implemented in the controller software (soft registration) or, for greater accuracy, in physical hardware (hard registration).



**Figure 13.2 Registration Position**

The Registration Latch mechanism is controlled two Event Control instructions, MAR (Motion Arm Registration) and MDR (Motion Disarm Registration). Refer to the AC Motion Instruction Specification for a detailed description of these instructions.

The accuracy of the registration position value, saved as a result of a registration event, is a function of the delay in recognizing the specified transition (typically 1  $\mu$ sec. for hardware registration) and the speed of the axis during this time. The uncertainty in the registration position is the distance traveled by the axis during this interval as shown by the equation below:

$$\text{Uncertainty} = \text{Axis Speed} \left[ \frac{\text{Position Units}}{\text{Second}} \right] \times \text{Delay}$$

Use the formula given above to calculate the maximum registration position error for the expected axis speed. Alternatively, you can calculate the maximum axis speed for a specified registration accuracy by re-arranging this formula as shown below:

$$\text{Maximum Speed} \left[ \frac{\text{Position Units}}{\text{Second}} \right] = \frac{\text{Desired Accuracy} [\text{Position Units}]}{\text{Delay}}$$

**Registration Time** The two Registration Time values contain the lower 32-bits of CST time at which their respective registration events occurred. Units for this attribute are in microseconds.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Registration 1 Time	DINT	Lower 32 bits of CST time
GSV	Registration 2 Time	DINT	Lower 32 bits of CST time

**Interpolation Time** Interpolated Time is the 32-bit CST time used to calculate the interpolated positions. When this attribute is updated with a valid CST value, the Interpolated Actual Position and Interpolated Command Position values are automatically calculated.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Interpolation Time	DINT	CST time to interpolate to

**Interpolated Actual Position** Interpolated Actual Position is the interpolation of the actual position, based on past axis trajectory history, at the time specified by the “Interpolated Time” attribute.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Interpolated Actual Position	REAL	Position Units

**Interpolated Command Position** Interpolated Command Position is the interpolation of the commanded position, based on past axis trajectory history, at the time specified by the “Interpolated Time” attribute.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Interpolated Command Position	REAL	Position Units

**Master Offset** The Master Offset is the position offset that is currently applied to the master side of the position cam. The Master Offset is returned in master position units. The Master Offset will show the same unwind characteristic as the position of a linear axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Master Offset	REAL	Master Position Units

**Strobe Master Offset** The Strobe Master Offset is the position offset that was applied to the master side of the position cam when the last Motion Group Strobe Position (MGSP) instruction was executed. The Strobe Master Offset is returned in master position units. The Strobe Master Offset will show the same unwind characteristic as the position of a linear axis.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Strobe Master Offset	REAL	Master Position Units

**Start Master Offset** The Start Master Offset is the position offset that was applied to the master side of the position cam when the last Motion Axis Move (MAM) instruction with the move type set to “Absolute Master Offset” or “Incremental Master Offset” was executed. The Start Master Offset is returned in master position units. The Start Master Offset will show the same unwind characteristic as the position of a linear axis.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Start Master Offset	REAL	Master Position Units

## Motion Status Bit Attributes

This section describes the various Motion Axis Object status bit attributes.

### Motion Status Bits

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motion Status Bits	DINT	Direct Access Entire DINT - <b>MotionStatus</b> 0: Acceleration Status - <b>AccelStatus</b> 1: Deceleration Status - <b>DecelStatus</b> 2: Move Status - <b>MoveStatus</b> 3: Jog Status - <b>JogStatus</b> 4: Gearing Status - <b>GearingStatus</b> 5: Homing Status - <b>HomingStatus</b> 6: Stopping Status - <b>StoppingStatus</b> 7: Homed Status - <b>HomedStatus</b> 8: Position Cam Status - <b>PositionCamStatus</b> 9: Time Cam Status - <b>TimeCamStatus</b> 10: Position Cam Pending Status - PositionCamPendingStatus 11: Time Cam Pending Status - TimeCamPendingStatus 12: Gearing Lock Status - <b>GearingLockStatus</b> 13: Position Cam Lock Status - PositionCamLockStatus 14: <i>Reserved (Time Cam Lock Status)</i> - 15: Master Offset Move Status - MasterOffsetMoveStatus 16-31: <i>Reserved</i>

### Acceleration/Deceleration Status

The Acceleration and Deceleration Status bit attributes can be used to determine if the axis is currently being commanded to accelerate or decelerate. If neither bit is set then the axis is running at steady state velocity or at rest.

### Move Status

The Move Status bit attribute is set if a Move motion profile is currently in progress. As soon as the Move is complete or superseded by some other motion operation, the Move Status bit is cleared.

### **Jog Status**

The Jog Status bit attribute is set if a Jog motion profile is currently in progress. As soon as the Jog is complete or superseded by some other motion operation, the Jog Status bit is cleared.

### **Gearing Status**

The Gearing Status bit attribute is set if the axis is currently Gearing to another axis. As soon as the gearing operation is stopped or superseded by some other motion operation, the Gear Status bit is cleared.

### **Homing Status**

The Homing Status bit attribute is set if a Home motion profile is currently in progress. As soon as the Home is complete or superseded by some other motion operation, the Home Status bit is cleared.

### **Stopping Status**

The Stopping Status bit attribute is set if there is a stopping process currently in progress. As soon as the stopping process is complete, the Stopping Status bit is cleared. The stopping process is used to stop an axis (initiated by an MAS, MGS, MGPS, Stop Motion fault action, or mode change). This bit is no longer associated with the gearing Clutch bit (MAG with Clutch selected) which, for I4B, has been explicitly named the Gearing Lock Status bit.

### **Homed Status**

The Homed Status bit attribute is cleared at power-up or reconnection. The bit is set to 1 by the MAH instruction upon successful completion of the configured homing sequence. This bit would be later cleared if the axis entered the shutdown state.

### **Position Cam Status**

The Position Cam Status bit attribute is set if a Position Cam motion profile is currently in progress. As soon as the Position Cam is complete or superseded by some other motion operation, the Position Cam Status bit is cleared.

### **Time Cam Status**

The Time Cam Status bit attribute is set if a Time Cam motion profile is currently in progress. As soon as the Time Cam is complete or superseded by some other motion operation, the Time Cam Status bit is cleared.



### **Position Cam Pending Status**

The Position Cam Pending Status bit attribute is set if a Position Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing a MAPC instruction with Pending execution selected. As soon as the current position cam profile completes, initiating the start of the pending cam profile, the Position Cam Pending bit is cleared. This bit is also cleared if the position cam profile completes, or superseded by some other motion operation.

### **Time Cam Pending Status**

The Time Cam Pending Status bit attribute is set if a Time Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing a MATC instruction with Pending execution selected. As soon as the current time cam profile completes, initiating the start of the pending cam profile, the Time Cam Pending bit is cleared. This bit is also cleared if the time cam profile completes, or superseded by some other motion operation.

### **Gearing Lock Status**

The Gearing Lock Status bit attribute is 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). During the intervals where the axis is clutching, the Gearing Lock Status bit is clear.

### **Position Cam Lock Status**

The Position Cam Lock Status bit attribute is set whenever the master axis satisfies the starting condition of an active Position Cam motion profile. The starting condition is set by the Start Control and Start Position parameters of the MAPC instruction. If the current position cam profile ends, or is superseded by another motion operation, the Position Cam Lock bit is cleared. In uni-directional master direction mode, the Position Cam Lock Status bit clears when moving in the “wrong” direction and sets when moving in the correct direction.

### **Master Offset Move Status**

The Master Offset Move Status bit attribute is set if a Master Offset Move motion profile is currently in progress. As soon as the Master Offset Move is complete or superseded by some other motion operation, the Master Offset Move Status bit is cleared.

## Axis Status Bit Attributes

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Status Bits	DINT	Direct Access Entire DINT - <b>AccelStatus</b> 0: Servo Action Status - <b>ServoActionStatus</b> 1: Drive Enable Status - <b>DriveEnableStatus</b> 2: Axis Shutdown Status - <b>ShutdownStatus</b> 3: Configuration Update in Process - <b>ConfigUpdateInProcess</b> 4-31: <i>Reserved</i>

### Servo Action Status

The Servo Action Status bit attribute is set when the associated axis is under servo control. If the bit is not set then servo action is disabled.

### Drive Enable Status

The Drive Enable Status bit attribute is set when the Drive Enable output of the associated physical axis is currently enabled. If the bit is not set then physical servo axis Drive Enable output is currently disabled.

### Shutdown Status

The Shutdown Status bit attribute is set when the associated axis is currently in the Shutdown state. As soon as the axis is transitioned from the Shutdown state to another state, the Shutdown Status bit is cleared.

### Configuration Update in Process

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 or an SSV in the user program. As soon as such an update is initiated, the ControlLogix processor sets the “Configuration Update in Process” bit. The 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.

**Axis Fault Bit Attributes** All of the fault bit attributes defined below can be handled by the ControlLogix processor as a Major Fault by configuring the associated Group Object's "General Fault Type Mechanism" attribute accordingly. Otherwise any specific fault handling must be done as part of the user program.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Fault Bits	DINT	Direct Access Entire DINT - <b>AxisFault</b> 0: Physical Axis Fault - <b>PhysicalAxisFault</b> 1: Module Fault - <b>ModuleFault</b> 2: Configuration Fault - <b>ConfigFault</b>

### Physical Axis Fault

If the Physical Axis Fault bit is set, it indicates that one or more fault conditions have been reported by the physical axis. The specific fault conditions can then be determined by accessing the fault attributes of the associated physical axis.

### Module Fault

The Module Fault bit attribute is set when a serious fault has occurred on the motion module associated with the selected axis. A module fault affects all axes associated with the motion module. It generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition.

### Configuration Fault

The Configuration Fault bit is 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.

**Module Fault Bit Attribute** The Module Fault Bit attribute is a collection of faults that have module scope as opposed to axis scope. Besides being a valid attribute for axes of data type Servo and Servo Drive, this attribute is also valid for a consumed axis data type. In this case, however, the module is the producing Logix processor rather than a motion module such as the 1784-PM02AE or 1784-PM16SE.

The fault bits are updated every coarse update period of the consuming Logix processor. The fault bit attributes defined below can be handled by the Logix processor as a Major Fault by configuring the associated Group Object's "General Fault Type Mechanism" attribute accordingly. Otherwise any specific fault handling must be done as part of the user program.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Module Fault Bits	DINT	Direct Access Entire DINT - <b>ModuleFaults</b> 0: Control Sync Fault - <b>ControlSyncFault</b> 1-31: <i>Reserved</i>

### Control Sync Fault

The Control Sync Fault bit attribute is set when the Logix controller detects that several position update messages in a row from the producing controller have been missed due to a failure of the controller-to-controller communications connection. This condition results in the automatic shutdown of the associated servo module. The consuming 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. The Control Sync Fault bit is cleared when the connection is reestablished.

### Axis Event Bit Attributes

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
Template	Axis Event Bits	DINT	Direct Access Entire DINT - <b>AxisEvent</b> 0: Watch Event Armed Status - <b>WatchEventArmedStatus</b> 1: Watch Event Status - <b>WatchEventStatus</b> 2: Registration Event 1 Armed Status - <b>RegEvent1ArmedStatus</b> 3: Registration Event 1 Status - <b>RegEvent1Status</b> 4: Registration Event 2 Armed Status - <b>RegEvent2ArmedStatus</b> 5: Registration Event 2 Status - <b>RegEvent2Status</b> 6: Home Event Armed Status - <b>HomeEventArmedStatus</b> 7: Home Event Status - <b>HomeEventStatus</b> 8-31: <i>Reserved</i>

### **Watch Event Armed Status**

The Watch Event Armed Status bit attribute is set when a watch event has been armed through execution of the MAW (Motion Arm Watch) instruction. This bit is cleared when either a watch event occurs or a MDW (Motion Disarm Watch) instruction is executed.

### **Watch Event Status**

The Watch Event Status bit attribute is set when a watch event has occurred. This bit is cleared when either another MAW (Motion Arm Watch) instruction or a MDW (Motion Disarm Watch) instruction is executed.

### **Registration 1 Event Armed Status**

The Registration 1 Event Armed Status bit attribute is set when a registration checking has been armed for registration input 1 through execution of the MAR (Motion Arm Registration) instruction. This bit is cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.

### **Registration 1 Event Status**

The Registration 1 Event Status bit attribute is set when a registration event has occurred on registration input 1. This bit is cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.

### **Registration 2 Event Armed Status**

The Registration 2 Event Armed Status bit attribute is set when a registration checking has been armed for registration input 2 through execution of the MAR (Motion Arm Registration) instruction. This bit is cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.

### **Registration 2 Event Status**

The Registration 2 Event Status bit attribute is set when a registration event has occurred on registration input 2. This bit is cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.

### **Home Event Armed Status**

The Home Event Armed Status bit attribute is set when a home event has been armed through execution of the MAH (Motion Axis Home) instruction. This bit is cleared when a home event occurs.

## Home Event Status

The Home Event Status bit attribute is set when a home event has occurred. This bit is cleared when another MAH (Motion Axis Home) instruction is executed.

**Output Cam Status** The Output Cam Status bit is set when an Output Cam has been initiated. The Output Cam Status bit is reset when the cam position moves beyond the cam start or cam end position in “Once” execution mode with no Output Cam pending or when the Output Cam is terminated by a MDOC instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Output Cam Status	DINT	Set of Output Cam Status bits

**Output Cam Pending Status** The Output Cam Pending Status bit is set if an Output Cam is currently pending the completion of another Output Cam. This would be initiated by executing an MAOC instruction with Pending execution selected. As soon as this output cam is armed, being triggered when the currently executing Output Cam has completed, the Output Cam Pending bit is cleared. This bit is also cleared if the Output Cam is terminated by a MDOC instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Output Cam Pending Status	DINT	Set of Output Cam Pending Status bits

**Output Cam Lock Status** The Output Cam Lock Status bit is set when an Output Cam has been armed. This would be initiated by executing an MAOC instruction with Immediate execution selected, when a pending output cam changes to armed, or when the axis approaches or passes through the specified axis arm position. As soon as this output cam current position moves beyond the cam start or cam stop position, the Output Cam Lock bit is cleared. This bit is also cleared if the Output Cam is terminated by a MDOC instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Output Cam Lock Status	DINT	Set of Output Cam Lock Status bits

**Output Cam Transition Status** The Output Cam Transition Status bit is set when a transition between the currently armed and the pending Output Cam is in process. Therefore, each Output Cam controls a subset of Output Bits. The Output Cam Transition Status bit is reset, when the transition to the pending Output Cam is complete or when the Output Cam is terminated by a MDOC instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Output Cam Transition Status	DINT	Set of Output Cam Transition Status bits

## Motion Object Configuration Attributes

The following sections define in more detail the behavior of all the various configuration attributes associated with the Motion axis Object. The attributes are, by definition, have read-write access. The Servo Object Configuration Attributes are divided into five categories: Motion General Configuration, Motion Units, Motion Conversion, Motion Homing, Motion Dynamics, and Motion Instruction attributes. These categories correspond roughly to the organization of the RSLogix 5000 Axis Properties pages.

## Motion Unit Configuration Attributes

**Average Velocity Timebase** The Average Velocity Timebase attribute is used to specify the desired time in seconds to be used for calculating the Average Velocity of the axis. When the Average Velocity Value is requested, the value is computed by taking the total distance traveled by the axis in the amount of time given by the Average Velocity Timebase and dividing this value by the timebase.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Average Velocity Timebase	REAL	Sec

The Average Velocity Timebase value should be large enough to filter out the small changes in velocity which would otherwise result in a “noisy” velocity value, but small enough to track significant changes in axis velocity. Typically, a value between 0.25 and 0.5 seconds works well for most applications

## Motion Conversion Configuration

**Conversion Constant** To allow axis position to be displayed and motion to be programmed in the position units specified by the Position Unit string attribute, a Conversion Constant must be established for each axis. The Conversion Constant, sometimes known as the K constant, allows the Axis Object to convert the axis position units into feedback counts and vice versa. Specifically, K is the number of feedback counts per Position Unit.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Conversion Constant	REAL	Counts / Position Unit Range: 0.1 - 1e <sup>12</sup> Default: 8000.0

Note that the 1784-PM02AE encoder based servo module uses 4X encoder feedback decoding (both edges of channel A and B are counted). The count direction is determined from both the direction of the edge and the state of the opposite channel. Channel A leads channel B for increasing count. This is the most commonly used decode mode with incremental encoders, since it provides the highest resolution.

For example, suppose this servo axis utilizes a 1000 line encoder in a motor coupled directly to a 5 pitch lead screw (5 turns per inch). With a user defined Position Unit of Inches, the conversion constant is calculated as shown below:

$$K = 1000 \text{ Lines/Rev} * 4 \text{ Counts/Line} * 5 \text{ Revs/Inch} = 20,000 \text{ Counts/Inch.}$$

### ATTENTION



If 'Conversion Constant' is changed it invalidates all of the settable attributes with "Position Unit" conversions in "Values" column. To be valid the 'Conversion Constant' must be set to the desired value prior to setting (including defaulting) any of the affected attributes.



**Rotary Axis** When the Rotary Axis attribute is set true (1), it 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 physical revolution. The number of encoder counts per physical revolution of the axis is specified by the Position Unwind attribute. If the Rotary Axis attribute is false (0), indicating linear operation, the maximum total linear excursion is limited to 1 Billion feedback counts before rolling over to zero.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Rotary Axis	SINT	0 = Linear 1 = Rotary

**Position Unwind** If the axis is configured as a rotary axis by setting the corresponding Rotary Axis bit Servo Configuration Bit word, a value for the Position Unwind attribute is required. This is the value used to perform automatic electronic unwind of the rotary axis. Electronic unwind allows infinite position range for rotary axes by subtracting the unwind value from both the actual and command position every time the axis makes a complete revolution. To avoid accumulated error due to round-off with irrational conversion constants, the unwind value is requested in units feedback counts per axis revolution and must be an integer.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Unwind	DINT	Counts per Revolution

For example, suppose that a given axis is configured as a Rotary Axis with Position Units of “Degrees” and 10 feedback counts per degree. It is desired to unwind the axis position after every revolution. In this case, the Position Unwind attribute should be set to 3600 since there are 3600 feedback counts ( $10 * 360$ ) per revolution of the axis.

## Motion Homing Configuration

**Home Mode** There are currently three Homing Modes supported by the Motion Axis Object, active, passive, and absolute. Active homing is the most common homing procedure for physical servo axes.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Home Mode	SINT	Enumeration: 0 = passive 1 = active (default) 2 = absolute

## Active

When active homing is chosen as the homing mode, the desired homing sequence is then selected by specifying whether or not a home limit switch and/or the encoder marker is used for this axis. Active homing sequences always use the trapezoidal velocity profile. The Home Sequence attribute section below describes the available active homing sequences.

## Passive

Passive homing redefines the current 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, as described below, except that no motion is commanded—the controller just *waits* for the switch and marker events to occur.

## Absolute

If the motion axis hardware supports an absolute feedback device, a Homing Mode of “absolute” may be used. The only valid Home Sequence for an absolute Homing Mode is “immediate”. In this case, 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. Prior to execution of the absolute homing process via the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled.

## Home Sequence and Home Direction

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Home Direction	SINT	0 = uni-directional forward 1 = bi-directional forward 2 = unidirectional reverse 3 = bi-directional reverse
SSV/GSV	Home Sequence	SINT	0 = immediate (default) 1 = switch 2 = marker 3 = switch then marker

## **Active Homing**

Active homing sequences, with the exception of the “Immediate” home sequence type, employ trapezoidal jog velocity profiles to move the axis while waiting for a homing event to occur. When “Active” is the configured Home Mode, the Home Sequence attribute is used to specify whether or not a home limit switch and/or the feedback device marker is to be used for the homing events. The Home Direction attribute determines the directional behavior of jog profiles associated with the specified homing sequence. Uni-directional and Bi-directional refer whether or not the jog is to reverse direction after detecting the homing event. Forward and Reverse refer to the direction of the initial jog during the homing process. The available active homing sequences are described in detail below with the assumption that the Home Direction is always forward

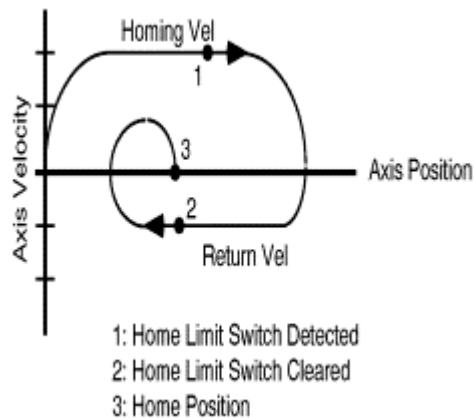
### **Active Immediate Home**

This is the simplest active homing sequence type. 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 and the Home Offset attribute is not applicable.

### **Active Bi-directional Home with Switch**

This active homing sequence is useful when an encoder marker is not available.

When this sequence is performed, the axis moves in the specified Home Direction at the specified Home Speed until the home limit switch is detected. The axis then decelerates to a stop and then moves in the opposite direction at the specified Home Return Speed until the home limit switch is cleared. When the home limit switch is cleared, axis position is immediately redefined to be equal to the Home Position and the axis decelerates to a stop. If Home Offset is non-zero, then the Home Position will be offset from the point where the home switch is cleared by this value. Once the axis decelerates to a stop, the controller then moves the axis back to the Home Position at the Home Return Speed using a trapezoidal move profile. If the axis is configured as a Rotary Axis, the move back to the Home Position takes the shortest path (i.e., no more than  $\frac{1}{2}$  revolution). The motions for this active homing sequence are shown below.



**Figure 13.3 Active Bi-Directional Homing Sequence**

If the controller detects that the state of the home switch at the start of the homing sequence is active, the controller immediately reverses the homing direction and begins the return leg of the homing sequence.

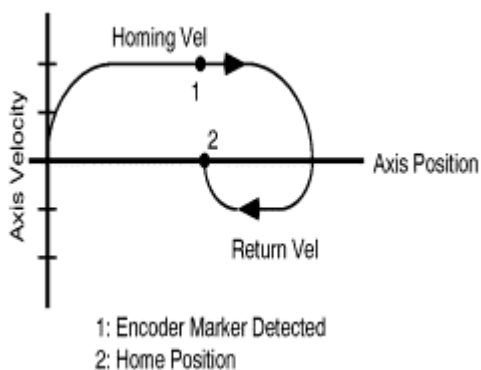
Neglecting the mechanical uncertainty of the home limit switch, the accuracy of this homing sequence depends on the time uncertainty in detecting the home limit switch transitions. The position uncertainty of the home position is the product of the maximum time for the control to detect the home limit switch ( $\sim 10$  milliseconds) and the specified Home Return Speed. For this reason, the Home Return Speed is often made significantly slower than the Home Speed.

For example, if a Home Return Speed of 0.1 inches per second (6 IPM) is specified, the uncertainty of the home position is calculated as shown below:

$$\text{Uncertainty} = 0.1 \text{ Inch/Sec} * 0.01 \text{ Sec} = 0.001 \text{ Inch.}$$

### Active Bi-directional Home with Marker

This active homing sequence is useful for single turn rotary and linear encoder applications since these have only one encoder marker for full axis travel. When this sequence is performed, the axis moves in the specified Home Direction at the specified Home Speed until the marker is detected. The Home Position is then assigned to the axis position corresponding to the marker location, and the axis decelerates to a stop. If Home Offset is non-zero, then the Home Position will be offset from the point where the marker is detected by this value. The controller then moves the axis back to the Home Position at the specified Home Return Speed using a trapezoidal move profile. If the axis is configured as a Rotary Axis, the move back to the Home Position takes the shortest path (i.e., no more than  $\frac{1}{2}$  revolution). The axis behavior for this homing sequence is shown below.



**Figure 13.4 Active Bi-Directional Home with Marker**

The accuracy of this homing sequence depends only on the time delay in detecting the marker transition. The position uncertainty of the home position is the product of the maximum delay for the control to detect the marker pulse ( $\sim 1$  microsecond) and the specified Home Speed.

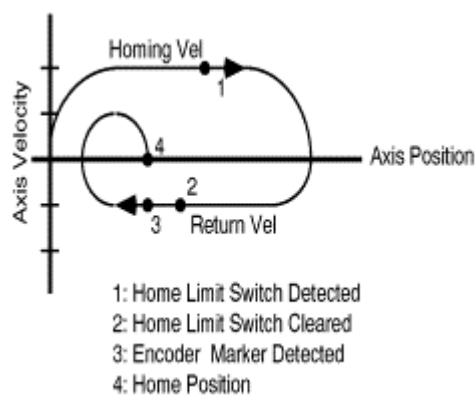
For example, if a Home Speed of 1 inches per second (60 IPM) is specified, the uncertainty of the home position is calculated as shown below:

$$\text{Uncertainty} = 1 \text{ Inch/Sec} * 0.000001 \text{ Sec} = 0.000001 \text{ Inch.}$$

Clearly, using the marker pulse as part of the homing sequence results in a tremendous increase in absolute homing accuracy over just employing mechanical limit switches.

### Active Bi-directional Home with Switch then Marker

This is the most precise active homing sequence available. When this sequence is performed, the axis moves in the specified Home Direction at the specified Home Speed until the home limit switch is detected. The axis then decelerates to a stop and moves in the opposite direction at the specified Home Return Speed until the home limit switch is cleared. After clearing the home limit switch, the axis continues in the same direction at the Home Return Speed until the first encoder marker is detected. The Home Position is assigned to the axis position at the moment that the marker is detected, and the axis then decelerates to a stop. If Home Offset is non-zero, then the Home Position will be offset from the point where the marker is detected by this value. The controller then moves the axis back to the Home Position at the specified Home Return Speed using a trapezoidal move profile. If the axis is configured as a Rotary Axis, the move back to the Home Position takes the shortest path (i.e., no more than  $\frac{1}{2}$  revolution). Axis behavior for this homing sequence is shown below.



**Figure 13.5 Active Home Bi-directional with Limit Switch and Marker**

If the controller detects that the state of the home switch at the start of the homing sequence is active, the controller immediately reverses the homing direction and begins the return leg of the homing sequence.

### Active Uni-directional Home with Switch

This active homing sequence is useful for when an encoder marker is not available and either uni-directional motion is required or proximity switch is being used.

When this sequence is performed in the Active Homing Mode, the axis moves in the specified Home Direction at the specified Home Speed until the home switch is detected. The Home Position is assigned to the axis position at the moment that the limit switch is detected. If Home Offset is non-zero, then the Home Position will be offset from the point where the switch is detected by this value. The controller then continues to move the axis to the Home Position at the specified Home Speed using a trapezoidal move profile. By setting a Home Offset greater than the deceleration distance, unidirectional motion to the Home Position is insured. However, if the Home Offset value is less than the deceleration distance, then the axis is simply decelerated to a stop. The axis does NOT reverse direction to move to the Home Position. In this case, the PC-bit leg of the associated MAH instruction is not set when the IP-bit leg is cleared.

In the case where this homing sequence is performed on a rotary axis and the Home Offset value is less than the deceleration distance when the home event is detected, the control automatically adds one or more revolutions to the move distance. This guarantees the resulting move to the Home Position is unidirectional.

### **Active Uni-directional Home with Marker**

This active homing sequence is useful for single turn rotary and linear encoder applications when uni-directional motion is required.

When this sequence is performed in the Active Homing Mode, the axis moves in the specified Home Direction at the specified Home Speed until the marker is detected. The Home Position is assigned to the axis position at the moment that the marker is detected. If Home Offset is non-zero, then the Home Position will be offset from the point where the marker was detected by this value. The controller then continues to move the axis to the Home Position at the specified Home Speed using a trapezoidal move profile. By setting a Home Offset greater than the deceleration distance, unidirectional motion to the Home Position is insured. However, if the Home Offset value is less than the deceleration distance, then the axis is simply decelerated to a stop. The axis does NOT reverse direction to move to the Home Position. In this case, the PC-bit leg of the associated MAH instruction is not set when the IP-bit leg is cleared.

In the case where this homing sequence is performed on a rotary axis and the Home Offset value is less than the deceleration distance when the home event is detected, the control automatically adds one or more revolutions to the move distance. This guarantees the resulting move to the Home Position is unidirectional.

### **Active Uni-directional Home with Switch then Marker**

This active homing sequence is useful for multi-turn rotary applications when uni-directional motion is required.

When this sequence is performed in the Active Homing Mode, the axis moves in the specified Home Direction at the specified Home Speed until the home switch is detected. The axis continues in the same direction at the Home Speed until the first marker event is detected. The Home Position is assigned to the axis position at the precise position where the marker was detected, and the axis then decelerates to a stop. If Home Offset is non-zero, then the Home Position will be offset from the point where the marker was detected by this value. The controller then continues to move the axis to the Home Position at the specified Home Speed using a trapezoidal move profile. By setting a Home Offset greater than the deceleration distance, unidirectional motion to the Home Position is insured. However, if the Home Offset value is less than the deceleration distance, then the axis is simply decelerated to a stop. The axis does NOT reverse direction to move to the Home Position. In this case, the PC-bit leg of the associated MAH instruction is not set when the IP-bit leg is cleared.

In the case where this homing sequence is performed on a rotary axis and the Home Offset value is less than the deceleration distance when the home event is detected, the control automatically adds one or more revolutions to the move distance. This guarantees the resulting move to the Home Position is unidirectional.

### **Passive Homing**

#### **Passive Immediate Home**

This is the simplest passive homing sequence type. When this sequence is performed, the controller immediately assigns the Home Position to the current axis actual position. This homing sequence produces no axis motion.

#### **Passive Home with Switch**

This passive homing sequence is useful for when an encoder marker is not available or a proximity switch is being used.

When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the home switch is detected. The Home Position is assigned to the axis position at the moment that the limit switch is detected. If Home Offset is non-zero, then the Home Position will be offset from the point where the switch is detected by this value.

#### **Passive Home with Marker**

This passive homing sequence is useful for single turn rotary and linear encoder applications.



When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the marker is detected. The home position is assigned to the axis position at the precise position where the marker was detected. If Home Offset is non-zero, then the Home Position will be offset from the point where the switch is detected by this value.

### Passive Home with Switch then Marker

This passive homing sequence is useful for multi-turn rotary applications.

When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the home switch and then the first encoder marker is detected. The home position is assigned to the axis position at the precise position where the marker was detected. If Home Offset is non-zero, then the Home Position will be offset from the point where the switch is detected by this value.

## Home Configuration Bits

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Home Configuration Bits	DINT	0: (Reserved) 1: Home Switch Normally Closed 2: (Reserved) 3-31: <i>Reserved</i>

### Home Switch Normally Closed

The Home Switch Normally Closed bit attribute determines the normal state of the home limit switch used by the homing sequence. The normal state of the switch is its state prior to being engaged by the axis during the homing sequence. For example, if the Home Switch Normally Closed bit is set (true) then the condition of the switch prior to homing is closed. When the switch is engaged by the axis during the homing sequence, the switch is opened, which constitutes a homing event.

**Home Position** The Home Position is the desired absolute position for the axis after the specified homing sequence has been completed. After an active homing sequence has completed, the axis is left at the specified Home Position.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Home Position	REAL	Position Units

In most cases, Home Position is set to zero, although any value, within the Maximum Positive and Negative Travel limits of the axis (if enabled), may also be used. (A description of the Maximum Positive and Negative Travel configuration attributes may be found in the Servo and Drive Axis Object specifications). For a rotary axis, the Home Position is constrained to be a positive number less than the Position Unwind value divided by the Conversion Constant.

When configured for absolute Homing Mode, the Home Position value is applied directly to the absolute feedback device to establish an absolute position reference for the system.

**Home Offset** When applied to an active or passive Homing Mode, using a non-immediate Home Sequence, the Home Offset is the desired position offset of the axis Home Position from the position at which the home event occurred. The Home Offset is applied at the end of the specified homing sequence before the axis moves to the Home Position. In most cases, Home Offset is set to zero. .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Home Offset	REAL	Position Units

After an active bi-directional homing sequence has completed, the axis is left at the specified Home Position. If the Home Offset is non-zero, the axis will then be offset from the marker or home switch event point by the Home Offset value. If the Home Offset is zero, the axis will sit right “on top of” the marker or home switch point.

**Home Speed** The Home Speed attribute controls the speed of the jog profile used in the first leg of an active homing sequence as described in the above discussion of the Home Sequence Type attribute.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Home Speed	REAL	Position Units / Sec

**Home Return Speed** The Home Return Speed attribute controls the speed of the jog profile used after the first leg of an active bi-directional homing sequence as described in the above discussion of the Home Sequence Type attribute. .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Home Return Speed	REAL	Position Units / Sec

## Motion Dynamics Configuration

**Maximum Speed** The value of the Maximum Speed attribute is used by various motion instructions (e.g. MAJ, MAM, MCD, etc.) to determine the steady-state speed of the axis. These instructions all have the option of specifying speed as a percent of the Maximum Speed attribute value for the axis. The Maximum Speed value for the axis is automatically set to the Tune Speed by the MAAT (Motion Apply Axis Tune) instruction. This value is typically set to ~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. .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Maximum Speed	REAL	Position Units / Sec

**Maximum Acceleration/Deceleration** The Maximum Acceleration and Deceleration attribute values are frequently used by motion instructions such as MAJ, MAM, MCD, etc., to determine the acceleration and deceleration rates to apply to the axis. These instructions all have the option of specifying acceleration and deceleration as a percent of the Maximum Acceleration and Maximum Deceleration attributes for the axis. The Maximum Acceleration and Maximum Deceleration values for the axis are automatically set to ~85% of the measured Tune Acceleration and Tune Deceleration by the MAAT (Motion Apply Axis Tune) instruction. If set manually, these values should typically be set to ~85% of the maximum acceleration and maximum deceleration rate of the axis. This provides sufficient “head-room” for the axis to operate at all times within the acceleration and deceleration limits of the drive and motor.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Maximum Acceleration	REAL	Position Units / Sec <sup>2</sup>
SSV/GSV	Maximum Deceleration	REAL	Position Units / Sec <sup>2</sup>

**Programmed Stop Mode** The Programmed Stop Mode attribute value determines how a specific axis will stop when the ControlLogix processor undergoes a critical processor mode change or when an explicit MGS (Motion Group Stop) instruction executed with its stop mode set to 'programmed'. There are currently four modes defined for the ControlLogix processor: Program Mode, Run Mode, Test Mode and Faulted Mode. Any mode change into or out of program mode (prog->run, prog->test, run->prog & test->prog) will initiate a programmed stop for every axis owned by that processor. Each individual axis can have its own Programmed Stop Mode configuration independent of other axes. Three methods of stopping a given axis are currently supported.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Programmed Stop Mode	SINT	0 = Fast Stop (default) 1 = Fast Disable 2 = Hard Disable 3 = Fast Shutdown 4 = Hard Shutdown

### Fast Stop

When the Programmed Stop Mode attribute is configured for 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.

### Fast Disable

When the Programmed Stop Mode attribute is configured for 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 disabled, and Servo Action disabled

### Hard Disable

When configured for Hard Disable, the axis is immediately disabled, i.e. Drive Enable disabled, Servo Action disabled, but the OK contact is left closed. Unless the drive is configured to provide some form of dynamic braking, this results in the axis coasting to a stop.

### Fast Shutdown

When configured for Fast Shutdown, the axis is decelerated to a stop as with Fast Stop but, once the axis motion is stopped, the axis is placed in the Shutdown state, i.e. Drive Enable disabled, servo action disabled, and the OK contact opened. To recover from the Shutdown state requires execution of one of the axis or group Shutdown Reset instructions (MASR or MGSR).

## Hard Shutdown

When configured for Hard Shutdown, the axis is immediately placed in the Shutdown state, i.e. Drive Enable disabled, Servo Action disabled, and the OK contact opened. Unless the drive is configured to provide some form of dynamic braking, this results in the axis coasting to a stop. To recover from the Shutdown state requires execution of one of the axis or group Shutdown Reset instructions (MASR or MGSR).

## Servo Status Attributes

The following sections define in more detail the behavior of the various status attributes associated with the Servo specific behavior of the Motion Axis Object. Status attributes are, by definition, read access only. The following Servo specific Status Attributes are divided into 3 categories: Servo Status attributes, Servo Commissioning Status attributes, and Servo Status Bit attributes.

**Servo Status Attributes** The list of Servo Status Attributes associated with the Axis Object provides access to the servo module resident information for the axis. These values may be used as part of the user program to perform real time measurements of servo operation. A list of all Servo Status Attributes is shown in the tables below.

Since Servo Status Attributes values are resident in the axis' servo module, these values need to be transferred to the ControllLogix processor module on a regular basis. To avoid unnecessary communication traffic transferring data that is not of interest, it is necessary to explicitly activate transfer of the specific Servo Status Attribute data from the servo module using the Axis Info Select attributes. Thus, a Servo Status Attribute value is **ONLY** valid if the attribute has been selected by one of the Axis Info Select attributes. Otherwise the Servo Status Attribute value is forced to zero.

In order for the position unit-based servo status attributes to return a meaningful value, the 'Conversion Constant' Axis Configuration Attribute must be established. Furthermore, attributes having velocity or acceleration units (Position Units / Sec) must also have a valid coarse update period which is established through association with a fully configured Motion Group Object.

Each of the Servo Status Attributes appears in the following Servo block diagram.

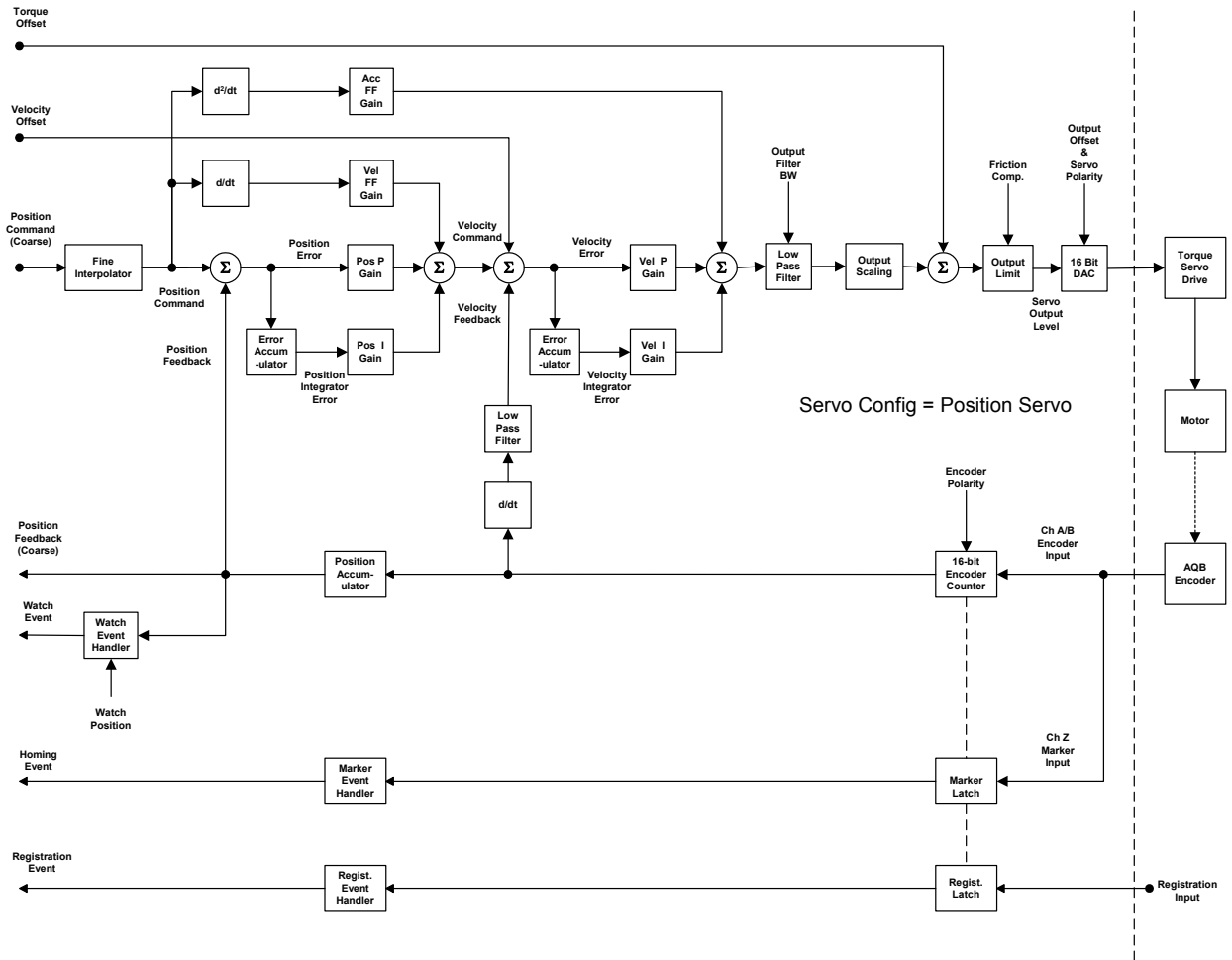


Figure 13.6 Servo Loop with Servo Attributes

**Position Command** Position Command is the current value of the Fine Command Position into the position loop summing junction, in configured axis Position Units. Within the active servo loop, the Position Command value is used to control the position of the axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Command	REAL	Position Units

**Position Feedback** Position Feedback is the current value of the Fine Actual Position into the position loop summing junction, in configured axis Position Units. Within the servo loop, the Position Feedback represents the current position of the axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Feedback	REAL	Position Units

**Aux Position Feedback** Aux Position Feedback is the current value of the position feedback coming from the auxiliary feedback input.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Aux Position Feedback	REAL	Position Units

**Position Error** Position Error is the difference, in configured axis Position Units, between the command and actual positions of a servo axis. For an axis with an active servo loop, position error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Error	REAL	Position Units

**Position Integrator Error** Position Integrator Error is the running sum of the Position Error, in the configured axis Position Units, for the specified axis. For an axis with an active servo loop, the position integrator error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Integrator Error	REAL	Position Units - mSec

**Velocity Command** Velocity Command is the current velocity reference to the velocity servo loop, in the configured axis Position Units per Second, for the specified axis. The Velocity Command value, hence, represents the output of the outer position control loop. Velocity Command is not to be confused with Command Velocity, which represents the rate of change of Command Position input to the position servo loop.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Command	REAL	Position Units / Sec

**Velocity Feedback** Velocity Feedback is the actual velocity of the axis as estimated by the servo module, in the configured axis Position Units per Second. The Estimated Velocity value is computed by applying a 1 KHz low-pass filter to the change in actual position over the servo update interval. Velocity Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Feedback	REAL	Position Units / Sec

**Velocity Error** Velocity Error is the difference, in configured axis Position Units per Second, between the commanded and actual velocity of a servo axis. For an axis with an active velocity servo loop, velocity error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Error	REAL	Position Units / Sec

**Velocity Integrator Error** Velocity Integrator Error is the running sum of the Velocity Error, in the configured axis Position Units per Second, for the specified axis. For an axis with an active velocity servo loop, the velocity integrator error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Integrator Error	REAL	Position Units – mSec / Sec

**Acceleration Command** Acceleration Command is the current acceleration reference to the output summing junction, in the configured axis Position Units per Second<sup>2</sup>, for the specified axis. The Acceleration Command value, hence, represents the output of the inner velocity control loop. Acceleration Command is not to be confused with Command Velocity, which represents the rate of change of Command Position input to the position servo loop.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Acceleration Command	REAL	Position Units / Sec <sup>2</sup>



**Acceleration Feedback** Acceleration Feedback is the actual velocity of the axis as estimated by the servo module, in the configured axis Position Units per Second<sup>2</sup>. The Estimated Acceleration is calculated by taking the difference in the Estimated Velocity over the servo update interval. Acceleration Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Acceleration Feedback	REAL	Position Units / Sec <sup>2</sup>

**Servo Output Level** Servo Output Level is the current voltage level of the servo output of the specified axis. The Servo Output Level can be used in drilling applications, for example, where the servo module is interfaced to an external Torque Loop Servo Drive, to detect when the drill bit has engaged the surface of the work piece.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Servo Output Level	REAL	Volts

**Marker Distance** Marker Distance is the distance between the axis position at which a home switch input was detected and the axis position at which the marker event was detected. This value is useful in aligning a home limit switch relative to a feedback marker pulse to provide repeatable homing operation.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Marker Distance	REAL	Position Units

## Servo Status Bit Attributes

This section describes the various Servo Axis Object status bit attributes.

### Servo Status Bit Attributes

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Servo Status Bits	DINT	Direct Access Entire DINT - <b>ServoStatus</b> 0: Servo Action Status - <i>No Tag</i> 1: Drive Enable Status - <b>No Tag</b> 2: Axis Shutdown Status - <i>No Tag</i> 3: Process Status - <b>ProcessStatus</b> 4: Output Limit Status - <b>OutputLimitStatus</b> 5: Position Lock Status - <b>PositionLockStatus</b> 6: Home Input Status - <b>HomeInputStatus</b> 7: Registration 1 Input Status - <b>Reg1Input Status</b> 8: Registration 2 Input Status - <b>Reg2InputStatus</b> 9: Positive Overtravel Input Status - <b>PosOvertravellInputStatus</b> 10: Negative Overtravel Input Status - <b>NegOvertravellInputStatus</b> 11-15: <i>Reserved</i> 16-31: <i>Reserved</i>

### Servo Action Status

The Servo Action Status bit attribute is set when servo action is currently enabled on the associated axis. If the bit is not set then servo action is disabled.

### Drive Enable Status

The Drive Enable Status bit attribute is set when the Drive Enable output of the associated physical axis is currently enabled. If the bit is not set then physical servo axis Drive Enable output is currently disabled.

### Shutdown Status

The Shutdown Status bit attribute is set when the associated axis is currently in the Shutdown state. As soon as the axis is transitioned from the Shutdown state to another state, the Shutdown Status bit is cleared.

## **Process Status**

The Process Status bit attribute is set when there is an axis tuning operation or an axis hookup diagnostic test operation in progress on the associated physical axis.

## **Output Limit Status**

The Output Limit Status bit attribute is set when the magnitude of the output of the associated physical servo axis has reached or exceeded the configured Output Limit value. If this bit is not set then the magnitude of the servo output is within the configured Output Limit value.

## **Position Lock Status**

The Position Lock Status bit attribute is 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. If this bit is not set then the magnitude of the axis position error is greater than the configured Position Lock Tolerance value.

## **Home Input Status**

The Home Input Status bit attribute represents the current state of the dedicated Home input. This bit is set if the Home input is active and clear if inactive.

## **Registration 1/2 Input Status**

The Registration Input 1 and Registration Input 1 Status bit attributes represent the current state of the corresponding dedicated Registration input. This bit is set if the registration input is active and clear if inactive.

## **Positive Overtravel Input Status**

The Positive Overtravel Input Status bit attribute represents the current state of the dedicated Positive Overtravel input. This bit is set if the Positive Overtravel input is active and clear if inactive.

## **Negative Overtravel Input Status**

The Negative Overtravel Input Status bit attribute represents the current state of the dedicated Negative Overtravel input. This bit is set if the Negative Overtravel input is active and clear if inactive.

## Axis Control Bit Attributes

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Control Bits	DINT	0:Abort Process Request 1: Shutdown Request 3:Zero DAC Request 4-14: <i>Reserved</i> 15:Change Cmd Reference 16-31: <i>Reserved</i>

### Abort Process Request

When the Abort Process bit is set, the servo module disables any active process, such as a tuning or test process.

### Shutdown Request

When the Shutdown Request bit is set, the servo module forces the axis into the shutdown state which opens the OK contact and zeroes the DAC output.

### Zero DAC Request

When the Zero DAC Request bit is set, the servo module forces the DAC output for the axis to zero volts. This bit only has an affect if the axis is in the Direct Drive State with the drive enabled but no servo action.

### Abort Home Request

When the Abort Home Request bit is set, any active homing procedures are cancelled.

### Abort Event Request

When the Abort Event Request bit is set, any active registration or watch event procedures are cancelled.

### Change Cmd Reference

The Change Command Reference Request bit attribute is set when the Logix processor has switched to a new position coordinate system for command position. The servo module processor uses this bit when processing new command position data from the Logix processor to account for the offset implied by the shift in the reference point. The bit is cleared when the Servo module acknowledges completion of the reference position change by clearing its Change Position Reference bit.

## Axis Response Bit Attributes

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Control Bits	DINT	0:Abort Process Acknowledge 1: Shutdown Acknowledge 2:Zero DAC Acknowledge 3:Abort Home Acknowledge 4:Abort Event Acknowledge 5-14: <i>Reserved</i> 15:Change Pos Reference 16-31: <i>Reserved</i>

### Abort Process Acknowledge

When the Abort Process Acknowledge bit is set, the servo module acknowledges that the tuning or test process has been aborted.

### Shutdown Request Acknowledge

When the Shutdown Acknowledge bit is set, the servo module acknowledges that the axis has been forced into the shutdown state.

### Zero DAC Request Acknowledge

When the Zero DAC Acknowledge bit is set, the servo module acknowledges that the DAC output for the axis has been set to zero volts.

### Abort Home Acknowledge

When the Abort Home Acknowledge bit is set, the servo module acknowledges that the active home procedure has been aborted.

### Abort Event Acknowledge

When the Abort Home Acknowledge bit is set, the servo module acknowledges that the active registration or watch position event procedure has been aborted.

### Change Position Reference

The Change Position Reference bit attribute is set when the Servo loop has switched to a new position coordinate system. The Logix processor uses this bit when processing new position data from the servo module to account for the offset implied by the shift in the reference point. The bit is cleared when the Logix processor acknowledges completion of the reference position change by clearing its Change Cmd Reference bit.

**Servo Fault Bit Attributes** The Servo Fault Bits attribute is a collection of all fault attributes that are associated with the servo axis. Servo Fault Bit attributes are passed from a servo module to the controller via a 32-bit value in the Synchronous Input connection axis data structure. Thus, these fault bits are updated every coarse update period.

All of the fault bit attributes defined below can be handled by the ControlLogix processor as a Major Fault by configuring the associated Group Object's "General Fault Type Mechanism" attribute accordingly. Otherwise any specific fault handling must be done as part of the user program.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Servo Fault Bits	DINT	Direct Access Entire DINT - <b>ServoFault</b> 0: Positive Soft Overtravel Fault <b>-PosSoftOvertravelFault</b> 1: Negative Soft Overtravel Fault <b>-NegSoftOvertravelFault</b> 2: Positive Hard Overtravel Fault <b>-PosHardOvertravelFault</b> 3: Negative Hard Overtravel Fault <b>-NegHardOvertravelFault</b> 4: Feedback Fault <b>-FeedbackFault</b> 5: Feedback Noise Fault <b>-FeedbackNoiseFault</b> 6: Auxiliary Feedback Fault <b>-AuxFeedbackFault</b> 7: Auxiliary Feedback Noise Fault <b>-AuxFeedbackNoiseFault</b> 8: Position Error Fault <b>-PositionErrorFault</b> 9: Drive Fault <b>-DriveFault</b> 13-31: <i>Reserved</i>

### Positive/Negative Soft Overtravel Status

If either the Positive Overtravel Status or Negative Overtravel Status bit attributes are set it indicates that the axis has traveled, or attempted to travel, beyond the current configured values for Maximum Positive Travel or Maximum Negative Travel, respectively. As soon as the axis is moved back within these travel limits, the corresponding Overtravel Status bit is cleared.

## Positive/Negative Hardware Overtravel Faults

If either the Positive Hard Overtravel Status or Negative Hard Overtravel Status bit attributes are set it indicates that the axis has traveled beyond the current 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.

## Feedback Loss Fault

If the Feedback Loss Fault bit is set for a specific feedback source, it indicates that one of the following conditions occurred:

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.

## Feedback Noise Fault

If the Feedback Noise Fault bit attribute is set for a specific feedback source, it indicates that simultaneous transitions of the feedback A and B channels has been detected by the servo module which is referred to generally as feedback noise. Feedback noise (shown below) 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.

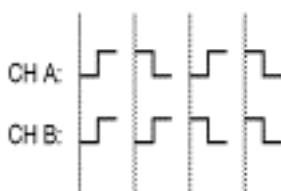


Figure 13.7 Feedback Noise

For example, loss of channel quadrature for an encoder can be caused by physical misalignment of the feedback transducer components, or excessive capacitance (or other delays) on the encoder signals. Proper grounding and shielding techniques can usually cure radiated noise problems. See the 1784-PM02AE Installation and Setup manual for information on grounding and shielding. 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.

### **Position Error Fault**

If the Position Error Fault bit attribute is set it indicates that the servo has detected that the axis position error has exceeded the current configured value for Position Error Tolerance. 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.

### **Drive Fault**

If the Drive Fault bit attribute is set it indicates that the external servo drive has detected a fault and has indicated such 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.

### **Module Fault Bit Attributes**

The Module Fault Bit attribute is a collection of all faults that have module scope as opposed to axis scope. Generally, these module faults are reflected by all axes supported by the given servo module.

Module Fault attribute information is passed from a physical module or device to the controller via an 8-bit value contained in the header of the Synchronous Input connection assembly. Thus, these fault bits are updated every coarse update period by the Motion Task. The module's map driver should also monitor module Faults so module fault conditions can be reflected to the user through the Module Properties dialog.



All of the fault bit attributes defined below can be handled by the ControlLogix processor as a Major Fault by configuring the associated Group Object's "General Fault Type Mechanism" attribute accordingly. Otherwise any specific fault handling must be done as part of the user program.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Servo Module Fault Bits	DINT	Direct Access Entire DINT - <b>ServoModuleFault</b> 0: Control Sync Fault <b>-ControlSyncFault</b> 1: Module Sync Fault <b>-ModuleSyncFault</b> 2: Timer Event Fault <b>-TimerEventFault</b> 3: Module Hardware Fault <b>-ModuleHardwareFault</b> 4-31: <i>Reserved</i>

### **Control Sync Fault**

The Control Sync Fault bit attribute is 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. The Synchronous Connection Fault bit is cleared when the connection is reestablished.

### **Module Sync Fault**

The Module Sync Fault bit attribute is 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. The Synchronous Connection Fault bit is cleared when the connection is reestablished.

### Timer Event Fault

If the Timer Event Fault bit attribute is set it indicates that 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). The Timer Event Fault bit can only be cleared by reconfiguration of the motion module.

### Module Hardware Fault

If the Module Hardware Fault bit attribute is set it indicates that the associated servo module has detected a hardware problem that, in general, is going to require replacement of the module to correct.

### Attribute Error Code

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Attribute Error Code	INT	ASA Error code returned by erred set attribute list service to the module.

When an Axis Configuration Fault occurs, one or more axis parameters associated with a servo module or device has not been successfully updated to match the value of the corresponding parameter of the local controller. The fact that the configuration of the servo axis no longer matches the configuration of the local controller is a serious fault and results in the shutdown of the faulted axis. The Attribute Error Code is reset to zero by reconfiguration of the motion module.

Axis Configuration Fault information is passed from the servo module or device to the controller via a 16-bit ASA status word contained in the Set Attribute List service response received by the controller. A Set Attribute List service to the motion module can be initiated by a software Set Attribute List service to the controller, or by an SSV instruction within the controller's program, referencing a servo attribute. Various routines that process responses to motion services are responsible for updating these attributes.

The Set and Get service responses provide a status response with each attribute that was processed. That status value is defined by ASA as follows: UINT16, Values 0-255 (0x00-0xFF) are reserved to mirror common service status codes. Values 256 – 65535 are available for object/class attribute specific errors.

### Attribute Error ID

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Attribute Error ID	INT	Attribute ID associated with non-zero Attribute Error Code.

The Attribute Error ID is used to retain the ID of the servo attribute that returned a non-zero attribute error code resulting in an Axis Configuration Fault. The Attribute Error ID defaults to zero and, after a fault has occurred may be reset to zero by reconfiguration of the motion module.

## Commissioning Status Attributes

The list of Commissioning Status Attributes associated with the Axis Object provides access to attributes associated with the state of various motion instruction generated commissioning processes. Motion instructions involved in commissioning an axis are MRAT (Motion Run Axis Tune) and MRHD (Motion Run Hookup Diagnostic). Commissioning Status Attributes are primarily used by external software (e.g., RSLogix5000) to implement the Test and Tuning dialogs associated with the axis configuration tool. However, these same attributes may also be used as part of the user program to implement a “built-in” axis test and tuning procedure. A list of all Commissioning Status Attributes are shown in the tables below.

In order for the position unit-based attributes to return a meaningful value, the ‘Conversion Constant’ Axis Configuration Attribute must be established. Furthermore, attributes having time units (Position Units / Sec) must also have a valid coarse update period which is established through association with a fully configured Motion Group Object.

**Test Status** The Test Status attribute returns status of the last run MRHD (Motion Run Hookup Diagnostic) instruction that initiates a hookup diagnostic process on the targeted servo module axis. The Test Status attribute can be used to determine when the MRHD initiated operation has successfully completed. Conditions may occur, however, that make it impossible for the control to properly perform the operation. When this is the case, the test process will be automatically aborted and a test fault reported that is stored in the Test Status output parameter.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Test Status	INT	Enumeration: 0 = test process successful 1 = test in progress 2 = test process aborted by user 3 = test process time-out fault (~2 seconds) 4 = test process failed due to servo fault

**Test Direction Forward** The Test Direction Forward attribute reports the direction of axis travel during hookup test as seen by the servo module during the last test process initiated by a MRHD (Motion Run Hookup Test) instruction. A Test Direction value of 1 (true) indicates that the direction of motion as observed by the servo module was in the forward (or positive) direction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Test Direction Forward	SINT	

**Tune Status** The Tune Status attribute returns status of the last run MRAT (Motion Run Axis Tuning) instruction that initiates a tuning process on the targeted servo module axis. The Tune Status attribute can, thus, be used to determine when the MRAT initiated operation has successfully completed. Conditions may occur, however, that make it impossible for the control to properly perform the operation. When this is the case, the tune process will be automatically aborted and a tune fault reported that is stored in the Tune Status output parameter.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Status	INT	0 = tune process successful 1 = tune in progress 2 = tune process aborted by user 3 = tune process time-out fault 4 = tune process failed due to servo fault 5 = axis reached Tuning Travel Limit 6 = axis polarity set incorrectly

**Tune Acceleration/Deceleration Time** The Tune Acceleration Time and Tune Deceleration Time attributes return acceleration and deceleration time in seconds for the last run MRAT (Motion Run Axis Tune) instruction. These values are used to calculate the Tune Acceleration and Tune Deceleration attributes.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Acceleration Time	REAL	Sec
GSV	Tune Deceleration Time	REAL	Sec

**Tune Acceleration/Deceleration** The Tune Acceleration Time and Tune Deceleration attributes return the measured acceleration and deceleration values for the last run MRAT (Motion Run Axis Tuning) instruction. These values are used, in the case of an external torque servo drive configuration, to calculate the Tune Inertia value of the axis, and are also typically used by a subsequent MAAT (Motion Apply Axis Tune) to determine the tuned values for the Maximum Acceleration and Maximum Deceleration attributes.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Acceleration	REAL	Position Units / Sec <sup>2</sup>
GSV	Tune Deceleration	REAL	Position Units / Sec <sup>2</sup>

**Tune Speed Scaling** The Tune Speed Scaling attribute returns the axis drive scaling factor measured during the last executed MRAT (Motion Run Axis Tune) instruction. This value is only applicable to axes configured for interface to an external velocity servo drive. In this case, the Tune Speed Scaling attribute value is directly applied to the Velocity Scaling attribute by a subsequent MAAT (Motion Apply Axis Tune) instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Speed Scaling	REAL	% / KiloCounts Per Sec

**Tune Rise Time** The Tune Rise Time attribute returns the axis rise time as measured during the last executed MRAT (Motion Run Axis Tune) instruction. This value is only applicable to axes configured for interface to an external velocity servo drive. In this case, the Tune Rise Time attribute value is used to calculate the Tune Velocity Bandwidth.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Rise Time	REAL	Sec

**Tune Inertia** When the axis is configured for interface to a external torque servo drive, the Tune Inertia value represents the total inertia for the axis as calculated from the measurements made during the last MRAT (Motion Run Axis Tune) initiated tuning process. In actuality, the units of Tune Inertia are not industry standard inertia units but rather in terms of percent (%) of full-scale servo output per MegaCounts/Sec<sup>2</sup> of feedback input. In this sense it represents the input gain of torque servo drive. These units represent a more useful description of the inertia of the system as seen by the servo controller. The Tune Inertia value is used by the MAAT (Motion Apply Axis Tune) instruction to calculate the Torque Scaling attribute.

If the Tune Inertia value exceeds 100 %Rated/MegaCounts Per Second<sup>2</sup>, performance of the digital servo loop may be compromised due to excessive digitization noise associated with the velocity estimator. This noise is amplified by the Torque Scaling gain which is related to the Tune Inertia factor and passed on to the torque output of the drive. A high Tune Inertia value can, thus, result in excitation of mechanical resonances and also result in excessive heating of the motor due to high torque ripple. The only solution to this problem is to lower the loop bandwidths and optionally apply some output filtering.

Since the Tune Inertia value represents a measure of the true system inertia, this situation can occur when driving a high inertia load relative to the motor, i.e. a high inertia mismatch. But it can also occur when working with a drive that is undersized for the motor or with a system having low feedback resolution. In general, the lower the Tune Inertia the better the performance of the digital servo loops will approximate that of an analog servo system.

The Logix tuning algorithm addresses excessive noise by managing quantization noise levels. The product of the Tune Inertia (% Rated/MCPS) and the Velocity Servo Bandwidth (Hertz) can be calculated to directly determine quantization noise levels. Based on this product, the tuning algorithm can take action to limit high frequency noise injection to the motor.

For motors with a Tune Inertia Bandwidth product of 1000 or greater, the LP Filter is applied with a Filter Bandwidth of 5x the Velocity Servo Bandwidth in Hertz. This limits the amount of phase lag introduced by the LP filter to ~12 degrees which is relatively small when compared to the 30 to 60 degrees of phase margin for a typical tuned servo system. With a tuned LP filter Bandwidth value of 200 Hz, you can expect the high frequency quantization noise in the 1 KHz range to be attenuated roughly by a factor of 5.

When the Tune Inertia Bandwidth product reaches 4000 or greater, the LP filter alone is not enough to manage the quantization noise level. The tune algorithm begins to taper the system bandwidth by the ratio of  $4000 / (\text{Tune Inertia} * \text{Vel Servo Bandwidth})$ . This holds the quantization noise level at a fixed value, independent of the Tune Inertia Bandwidth product.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Inertia	REAL	% / MegaCounts Per Sec <sup>2</sup>

## Servo Configuration Attributes

The following sections define in more detail the behavior of all the various configuration attributes associated with the Servo Axis Object. The attributes, by definition, have read-write access. The Servo Object Configuration Attributes are divided into five categories: Servo Configuration, Servo Gains, Servo Limits, Servo Offsets, and Servo Commissioning attributes. These categories correspond roughly to the organization of the RSLogix 5000 Axis Properties pages.

Each of the following Servo Configuration attributes are associated with corresponding attributes contained in the Servo Axis Object associated with servo module such as the 1784-PM02AE 2-Axis Servo module. When any of these attributes are modified by a Set Attribute List service or an SSV instruction within the user program, the local processor value for the attribute is immediately changed and a Set Attribute List service to the servo module is initiated to update the working value stored in the servo module. The progress of this update can be monitored, if necessary, within the user program through the Servo Configuration Update Bits status attribute.

**Servo Configuration** The following Servo Configuration attributes provide basic servo loop configuration information.

**Axis Type** The Axis Type attribute is used to establish the intended use of the axis. If the axis is intended for full servo operation than a value of “2” is required. If only the position information from the feedback interface is of interest, than a Axis Type should be set to “1”. Finally, if the axis is unused in the application, which is a common occurrence when there are an odd number of axes in the system, then the Axis Type associated with the unused axis should be set to “0”. Axis Type is not only used to qualify many operations associated with the axis servo loop, it also controls the behavior of the servo module’s Axis Status LEDs. An Axis Type of “1” (Feedback Only) results in the DRIVE LED being blanked, while a value of “0” (Unused) blanks both the FDBK and DRIVE LEDs.

External software (RSLogix5000) also uses the current configured value for Axis Type to control the look of many of the tab dialogs associated with the axis configuration tool.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Axis Type	INT	0 = unused 1 = feedback only 2 = servo

**Servo Loop Configuration** The Servo Loop Configuration attribute determines the specific configuration of the servo loop topology when the Axis Type is set to “servo”. When the Axis Type is set to “feedback only”, the Servo Loop Configuration is used to select which feedback port is to be used. Initial release of this object however will not support the auxiliary feedback port.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Servo Loop Configuration	INT	Enumeration: 0 = custom 1 = feedback only 2 = aux. feedback only 3 = position servo 4 = aux. position servo 5 = dual position servo 6 = dual command servo 7 = aux. dual command servo 8 = velocity servo 9 = torque servo

**External Drive Type** When the application requires the servo module axis to interface with an external velocity servo drive, the External Drive Type should be configured for “velocity servo drive”. This disables the servo module’s GSV/SSV digital velocity loop. If the External Drive Type attribute is set to “torque servo drive” the servo module’s GSV/SSV digital velocity loop is active. This configuration is the required configuration for interfacing to a torque loop servo drive.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	External Drive Type	DINT	0 = torque servo drive 1 = velocity servo drive



## Fault Configuration Bits

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Servo Fault Configuration Bits	DINT	Bit Field: 0: Soft Overtravel Checking 1: Drive Fault Checking 2: Drive Fault Normally Closed (S) 3-31: <i>Reserved</i> Soft Overtravel Checking for Linear Axis Only; Change to Rotary or Overtravel Checking requires Home range checks.

### Soft Overtravel Checking

When the Soft Overtravel Checking bit is set it enables a periodic test that monitors the current position of the axis and issues a Positive Overtravel Fault or Negative Overtravel Fault if ever the axis position travels outside the configured travel limits. The travel limits are determined by the configured values for the Maximum Positive Travel and Maximum Negative Travel attributes. This software overtravel check is not a substitute, but rather a supplement, for hardware overtravel fault protection which uses hardware limit switches to directly stop axis motion at the drive and deactivate power to the system. If the Soft Overtravel Checking bit is clear (default), then no software overtravel checking is done. Software overtravel checking is only available for a linear servo axes.

### Hard Overtravel Checking

When the Hard Overtravel Checking bit is set it enables a periodic test that monitors the current state of the positive and negative overtravel limit switch inputs and issues a Positive Hard Overtravel Fault or Negative Hard Overtravel Fault if ever the axis position travels activates the limit switch inputs. If the Hard Overtravel Checking bit is clear (default), then no overtravel limit switch input checking is done.

Hardware overtravel checking is only available for a linear servo axes.

### Drive Fault Checking

The 1784-PM02AE servo module provides a dedicated drive fault input for each axis. These inputs may be connected to fault outputs on the external drive (if provided) to notify the servo module of a fault in the drive itself. Set the Drive Fault Checking bit if you are using the servo module's drive fault input, and then specify the drive fault contact configuration of the amplifier's drive fault output as described below.

## Drive Fault Normally Closed

The Drive Fault Normally Closed bit attribute controls the sense of the Drive Fault input to the servo module. If this bit is set (true) then during normal (fault-free) operation of the drive, the Drive Fault input should be active, i.e. 24 Volts. If a drive fault occurs, the drive will open its drive fault output contacts and remove 24 Volts from the servo module's Drive Fault input generating an axis Drive Fault condition. This is the default "fail-safe" configuration. In some cases it may be necessary to clear the Drive Fault Normally Closed bit to interface with a drive system that closes its contacts when faulted. This is generally not recommended for "fail-safe" operation.

## Axis Info Select

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Info Select 1 Axis Info Select 2	DINT	0 = None (default) 1 = Position Command 2 = Position Feedback 3 = Aux Position Feedback 4 = Position Error 5 = Position Int. Error 6 = Velocity Command 7 = Velocity Feedback 8 = Velocity Error 9 = Velocity Int. Error 10 = Accel. Command 11 = Accel. Feedback 12 = Servo Output Level 13 = Marker Distance

**Servo Polarity Bits** Axis Info Select attributes are used to enable periodic data updates for selected servo status attributes. This method of accessing servo status data is designed to reduce the flow of unnecessary data for the Servo module. By selecting the servo status attribute of interest from the enumerated list, this attribute's value is transmitted along with the actual position data to the Logix processor. Thus, the servo status data update time is precisely the coarse update period. Note, 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 is static and does not reflect the true value in the servo module.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Servo Polarity Bits	DINT	0: Feedback Polarity Negative 1: Servo Polarity Negative 2-31: <i>Reserved</i>

### **Feedback Polarity Negative**

This Feedback Polarity Negative bit attribute controls the polarity of the encoder feedback and, when properly configured, insures that when the axis is moved in the user defined positive direction that the axis Actual Position value increases. This bit can be configured automatically using the MRHD and MAHD motion instructions.

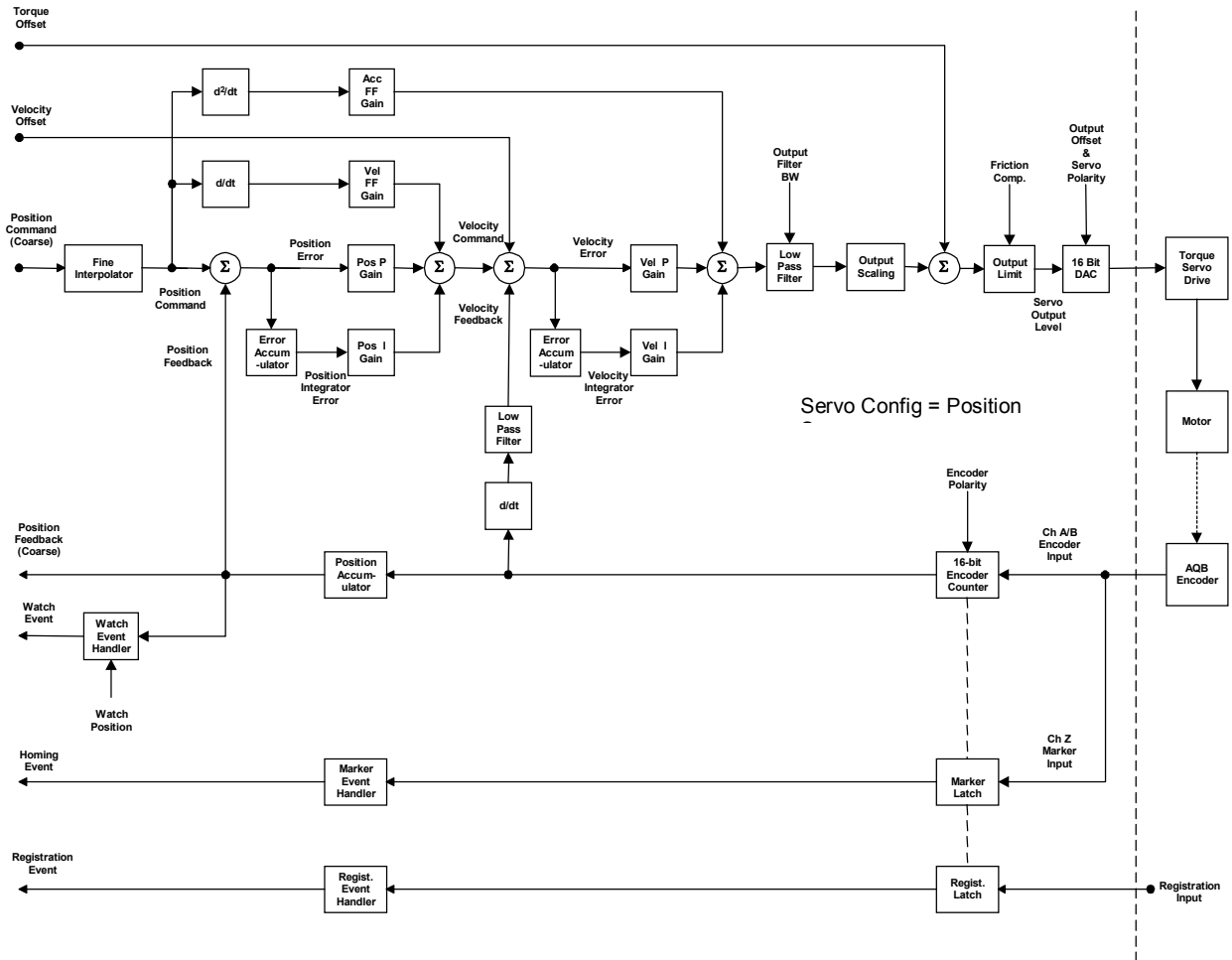
### **Servo Polarity Negative**

This Servo Polarity Negative bit attribute controls the polarity of the servo output to the drive. When properly configured along with the Feedback Polarity Negative bit, it ensures 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.

**Servo Loop Block Diagrams** The following section illustrates the various servo loop configurations that are supported with the first release of this object. Which of these servo loop topologies is in effect depends on the current settings of the of the Servo Loop Configuration and External Drive Type attributes.

**Position Servo with Torque Servo Drive**

This configuration provides full position servo control using an external torque loop servo drive. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.



**Figure 13.8 Position Servo with Torque Servo Drive**

**Velocity Servo with Torque Servo Drive**

This configuration provides velocity servo control using an external torque loop servo drive. Synchronous input data to the servo loop includes Velocity Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Velocity Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.

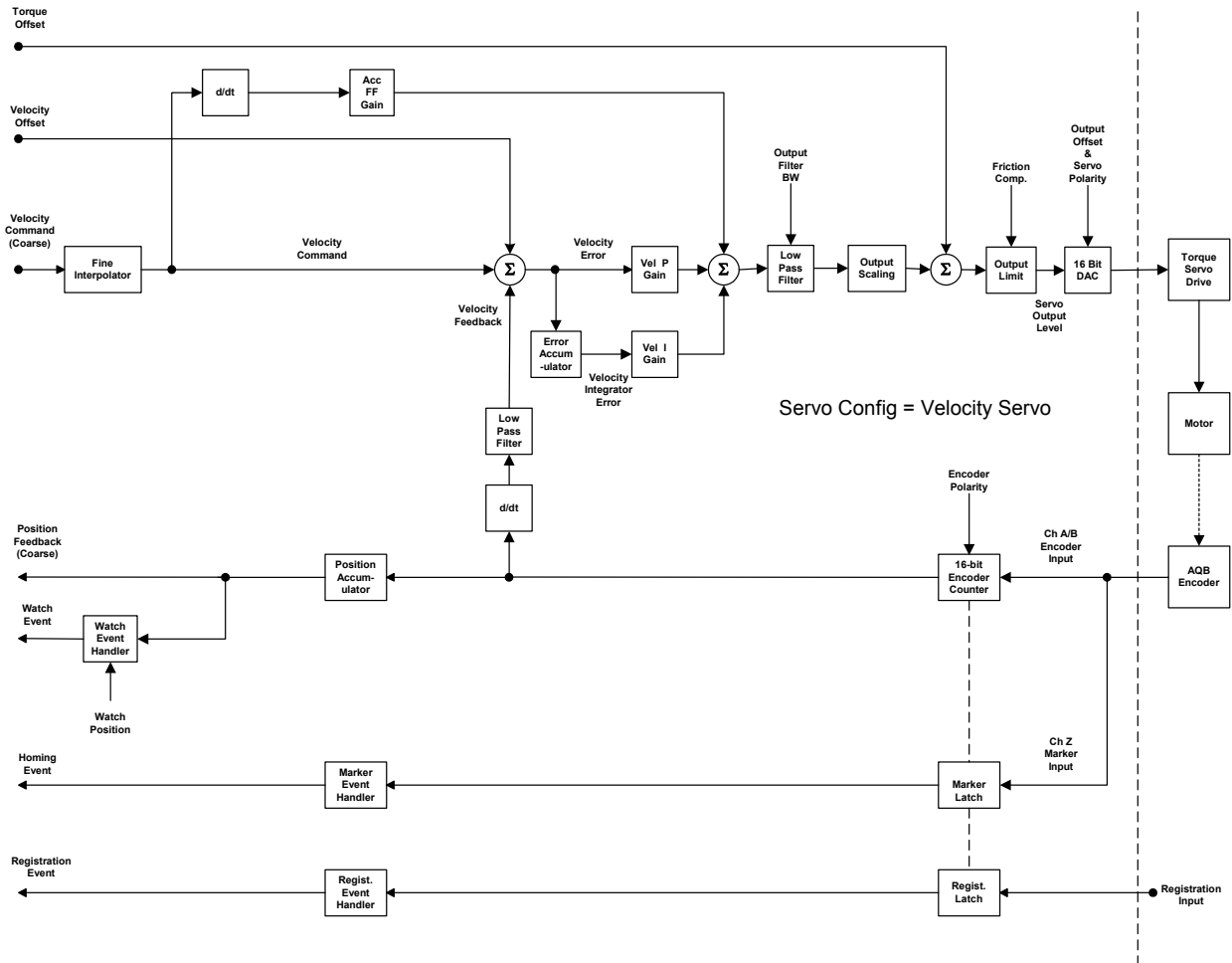
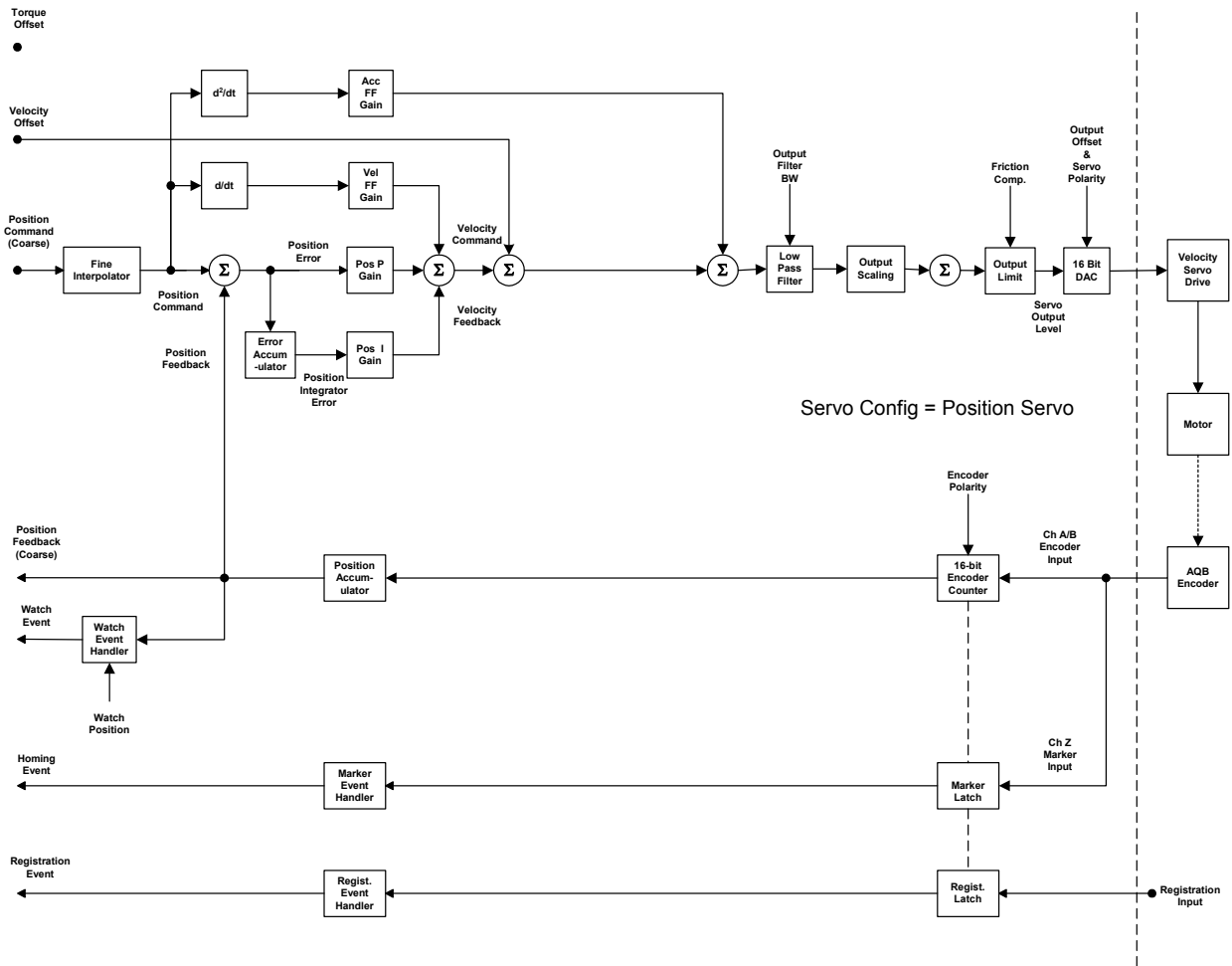


Figure 13.9 Velocity Servo with Torque Servo Drive

**Position Servo with Velocity Servo Drive**

This configuration provides full position servo control using an external velocity loop servo drive. Note that in this configuration the servo module does not close the velocity loop, but rather the drive does. Synchronous input data to the servo loop includes Position Command and Velocity Offset. (Torque Offset is ignored.) These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The Velocity Offset attribute can be changed programmatically via SSV instructions which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.



**Figure 13.10 Position Servo with Velocity Servo Drive**

**Velocity Servo with Velocity Servo Drive**

This configuration provides velocity servo control using an external velocity loop servo drive. In this configuration the servo module does not close the velocity loop, but rather the drive does. Synchronous input data to the servo loop includes Velocity Command and Velocity Offset. (Torque Offset is ignored.) These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The Velocity Offset attribute can be changed programmatically via SSV instructions which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.

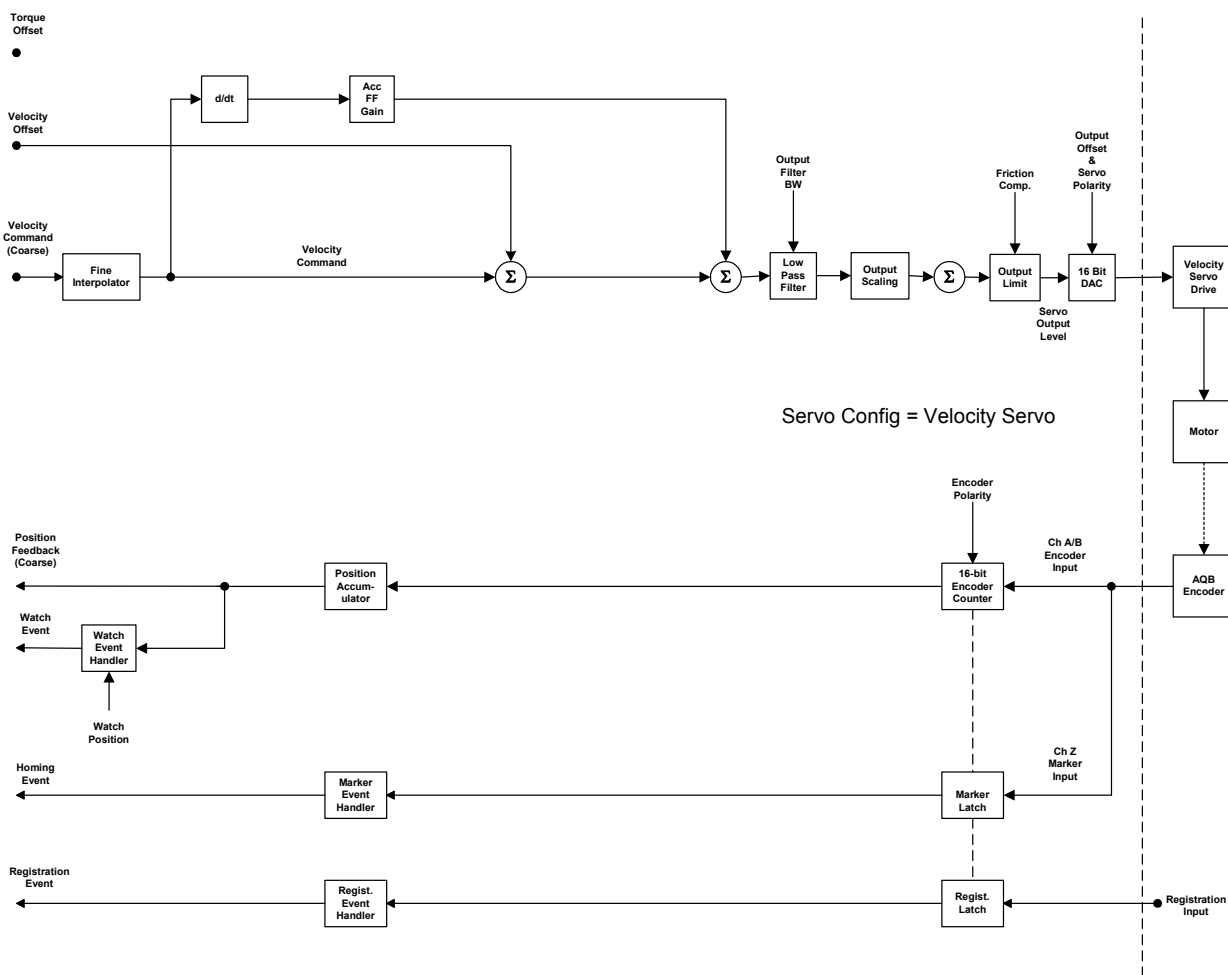


Figure 13.11 Velocity Servo with Velocity Servo Drive

**Servo Gains** The 1784-PM02AE 2-Axis Servo 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, again with proportional and integral gains for each axis. These gains provide software control over the servo dynamics, and allow the servo system to be completely stabilized. Unlike analog servo controllers, these digitally set gains do not drift. Furthermore, once these gains are set for a particular system, another servo module programmed with these gain values will operate identically to the original one.

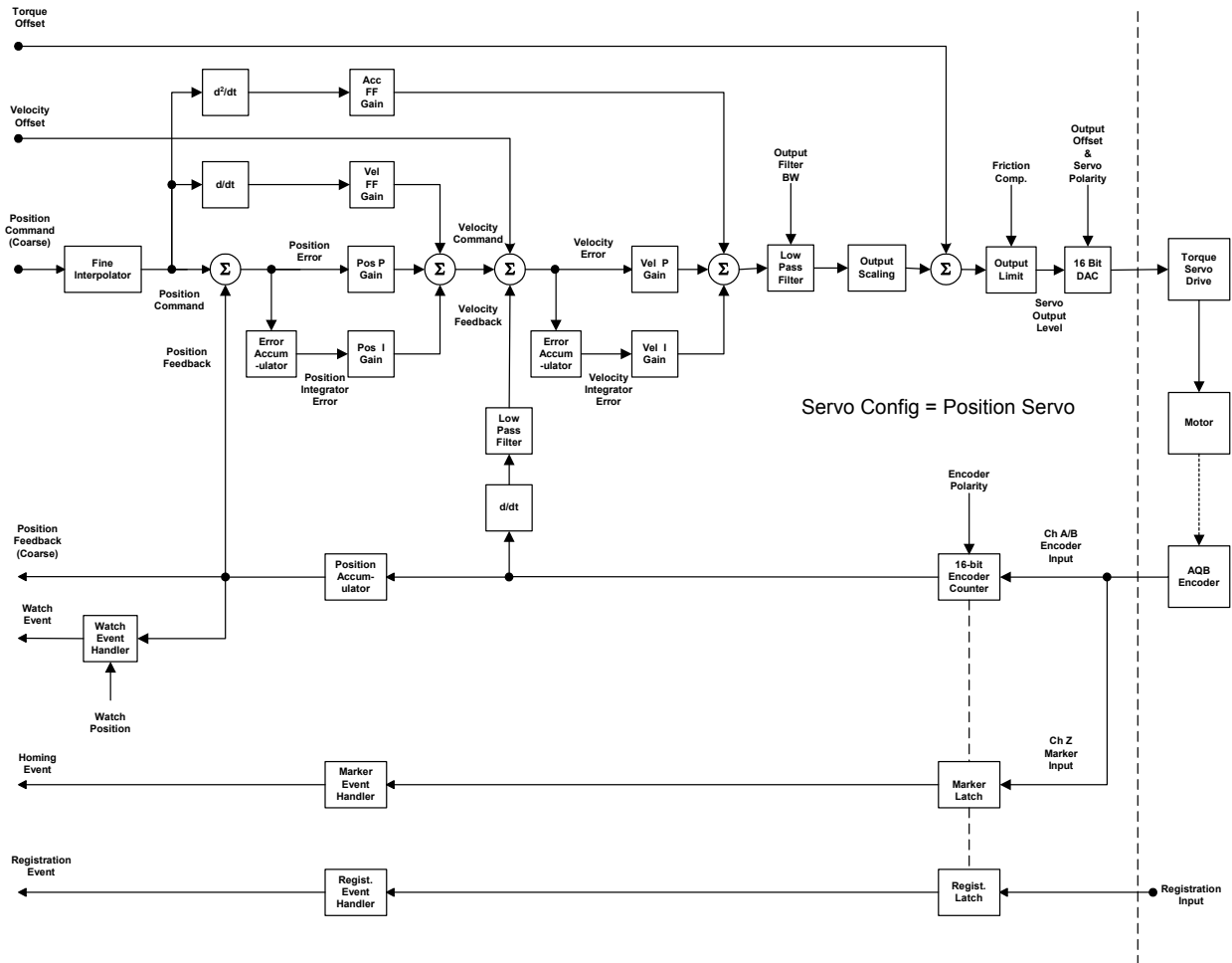


Figure 13.12 Servo Gains



**Velocity Feedforward Gain** Servo Drives require non-zero command input to generate steady-state axis acceleration or velocity. To provide the non-zero output from the 1784-PM02AE, a non-zero position or velocity error would need to be present. This dynamic error while moving is called “following error”. We ideally want zero following error all the time. This could be achieved through use of the position integral gain controls as described above, but typically the response time of the integrator action is too slow to be effective. An alternative approach that has superior dynamic response is to use Velocity and Acceleration Feedforward.

The Velocity Feedforward Gain attribute is used to provide the Velocity Command output necessary to generate the commanded velocity. It does this by scaling the current Command Velocity by the Velocity Feedforward Gain and adding it as an offset to the Velocity Command generated by the position loop control elements. With this done, the position loop control elements do not need to generate much of a contribution to the Velocity Command, hence the Position Error value is significantly reduced. 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. One thing that may force a smaller Velocity Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot. If necessary, the Velocity Feedforward Gain may be "tweaked" from the 100% value by running a simple user program that jogs the axis in the positive direction and monitor the Position Error of the axis during the jog. Increase the Velocity Feedforward Gain until the Position Error at constant speed is as small as possible, but still positive. If the Position Error at constant speed is negative, the actual position of the axis is *ahead* of the command position. If this occurs, decrease the Velocity Feedforward Gain such that the Position Error is again positive. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Velocity Feedforward Gain	REAL	%

**Acceleration Feedforward Gain** The Acceleration Feedforward Gain attribute is used to provide the Torque Command output necessary to generate the commanded acceleration. It does this by scaling the current Command Acceleration by the Acceleration Feedforward Gain and adding 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. One thing that may force a smaller Acceleration Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot.

If necessary, the Acceleration Feedforward Gain may be "tweaked" from the 100% value by running a simple user program that jogs the axis in the positive direction and monitors the Position Error of the axis during the jog. Usually Acceleration Feedforward is used in tandem with Velocity Feedforward to achieve near zero following error during the entire motion profile. To fine tune the Acceleration Feedforward Gain, the Velocity Feedforward Gain must first be optimized using the procedure described above. While capturing the peak Position Error during the acceleration phase of the jog profile, increase the Acceleration Feedforward Gain until the peak Position Error is as small as possible, but still positive. If the peak Position Error during the acceleration ramp is negative, the actual position of the axis is *ahead* of the command position during the acceleration ramp. If this occurs, decrease the Acceleration Feedforward Gain such that the Position Error is again positive. To be thorough the same procedure should be done for the deceleration ramp to verify that the peak Position Error during deceleration is acceptable. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Acceleration Feedforward Gain	REAL	%

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

**Position Proportional Gain** The Position Error is multiplied by the Position Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Increasing this gain value increases the bandwidth of the position servo loop and results in greater “static stiffness” of the axis which is a measure of the corrective force that is applied to an axis for a given 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.

A well-tuned system will move and stop quickly or "smartly" and exhibit little or no "ringing" during constant velocity or when the axis stops. If the response time is poor, or the motion "sloppy" or slow, the proportional gain may need to be increased. If excessive ringing or overshoot is observed when the motor stops, the proportional gain may need to be decreased.

While the Pos P Gain is typically established by the automatic servo tuning procedure, the Pos P gain 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. Refer to Output Scaling attribute description for an explanation of how the Output Scaling factor can be calculated. Once this is done the Pos P Gain can be computed based on either the desired loop gain or the desired bandwidth of the position servo system.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Proportional Gain	REAL	1/Sec

### Loop Gain Method

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.

$$\text{Pos P Gain} = 16.667 * \text{Desired Loop Gain (IPM/mil)}$$

Thus, according to an old machine tool rule of thumb, a loop gain of 1 IPM/mil (Pos P gain = 16.7 Sec<sup>-1</sup>) provides stable positioning for virtually any axis. In general, however, modern position servo systems typically run much tighter than this. The typical value for the Position Proportional Gain is ~100 Sec<sup>-1</sup>.

### Bandwidth Method

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

$$\text{Pos P Gain} = \text{Bandwidth (Hertz)} / 6.28$$

In general, however, modern position servo systems typically run with at least a unity gain bandwidth of ~16 Hertz. The typical value for the Position Proportional Gain is  $\sim 100 \text{ Sec}^{-1}$ .

### Maximum Bandwidth

There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and torque loops of the system and the desired damping of the system,  $Z$ . These limitations may be expressed as follows:

$$\text{Bandwidth (Pos)} = 0.25 * 1/Z^2 * \text{Bandwidth (Vel)} = 0.25 * 1/Z^2 * \text{Bandwidth (Torque)}$$

For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor,  $Z$ , is 0.8, the velocity bandwidth is approximately 40 Hz and the position bandwidth is 16 Hz. Based on these numbers the corresponding proportional gains for the loops can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.

**Position Integral Gain** Position Integral Gain, or Pos I Gain, improves the steady-state positioning performance of the system. By using Position Integral Gain, it is possible to achieve accurate axis positioning despite the presence of such disturbances as static friction or gravity. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

Every servo update the current Position Error is accumulated in variable called the Position Integral Error. This value is multiplied by the Position Integral Gain to produce a component to the Velocity Command that attempts to correct for the position error. The characteristic of Pos I Gain correction, however, is that any non-zero Position Error will accumulate in time to generate enough force to make the correction. This attribute of Pos I Gain makes it invaluable in applications where positioning accuracy or tracking accuracy is critical. The higher the Pos I Gain value the faster the axis is driven to the zero Position Error condition. Unfortunately, Pos I Gain control is intrinsically unstable. Too much Pos I Gain will result in axis oscillation and servo instability.

If the axis is configured for an external velocity loop servo drive, the Pos I Gain should be zero—most analog velocity loop servo amplifiers have integral gain of their own and will not tolerate *any* amount of Pos I Gain in the position loop without producing severe oscillations. If Pos I Gain is necessary for the application, the velocity integrator in the drive must be disabled.

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. Another common case is when performing certain motion. -. When the Integrator Hold Enable attribute is set, 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, 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. Refer to Output Scaling attribute description for an explanation of how the Output Scaling factor can be calculated. 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:

$$\text{Pos I Gain} = 0.25 * 0.001 \text{ Sec/mSec} * (\text{Pos P Gain})^2$$

Assuming a Pos P Gain value of  $100 \text{ Sec}^{-1}$  this results in a Pos I Gain value of  $2.5 \sim 0.1 \text{ mSec}^{-1} \cdot \text{Sec}^{-1}$

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Position Integral Gain	REAL	1/mSec-Sec

**Velocity Proportional Gain** When configured for a torque (current) loop servo drive, the servo module's digital velocity loop provides damping without the requirement for an analog tachometer. The 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 the damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. The velocity loop also allows higher effective position loop gain values to be used, however, too much Velocity Proportional Gain leads to high frequency instability and resonance effects. Note that units for Velocity Proportional Gain are identical to that of the Position Proportional Gain making it easy to perform classic inches/min/mil calculations to determine static stiffness or damping.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Velocity Proportional Gain	REAL	1/Sec

## Maximum Bandwidth

There are limitations to the maximum bandwidth that can be achieved for the velocity loop based on the dynamics of the torque loop of the servo drive and the desired damping of the system,  $Z$ . These limitations may be expressed as follows:

$$\text{Bandwidth (Velocity)} = 0.25 * 1/Z^2 * \text{Bandwidth (Torque)}$$

For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor,  $Z$ , is 0.8, the velocity bandwidth is approximately 40 Hz. Based on this number the corresponding gains for the loop can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.

The velocity loop in the motion controller is not used when the servo module is configured for a velocity loop servo drive. Establishing the Velocity Proportional Gain is not required in this case.

The typical value for the Velocity Proportional Gain is 250.

**Velocity Integral Gain** When configured for a torque (current) loop servo drive, every servo update the current Velocity Error is also accumulated in 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 characteristic of Vel I Gain correction, however, is that any non-zero Velocity Error will accumulate in time to generate enough force to make the correction. This attribute of Vel I Gain makes it invaluable in applications where velocity accuracy is critical. 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 will result 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. Another common case is when performing certain motion. When the Integrator Hold Enable attribute is set, 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, Velocity Integral Gain is the better choice.

The typical value for the Velocity Integral Gain is  $\sim 15 \text{ mSec}^{-1}\text{-Sec}^{-1}$ .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Integral Gain	REAL	1/mSec-Sec

**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 using a velocity servo drive, the software velocity loop in the servo module is disabled. In this case the Velocity Scaling value can be calculated by the following formula:

$$\text{Velocity Scaling} = 100 \% / (\text{Speed @ } 100 \%)$$

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 Torque Scaling attribute value would be calculated as shown below.

$$\text{Velocity Scaling} = 100 \% / (83.3 \text{ RPS}) = 1.2 \% / \text{Revs Per Second}$$

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Scaling	REAL	% / Position Units 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. In fact, the Torque Scaling value, when properly established, represents the inertia of the system and is related to the Tune Inertia attribute value by a factor of the Conversion Constant. The Torque Scaling value is typically established by the MAAT instruction as part of the controller’s automatic tuning procedure but the value can be manually calculated, if necessary, using the following guidelines.

$$\text{Torque Scaling} = 100 \% \text{ Rated Torque} / (\text{Acceleration @ } 100\% \text{ Rated Torque})$$

For example, if this axis is using position units of motor revolutions (revs), and that with 100% rated torque applied to the motor, the motor accelerates at a rate of 3000 Revs/Sec<sup>2</sup>, the Torque Scaling attribute value would be calculated as shown below.

$$\text{Torque Scaling} = 100 \% \text{ Rated} / (3000 \text{ RPS}^2) = 0.0333 \% \text{ Rated} / \text{Revs Per Second}^2$$

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Torque Scaling	REAL	% / Position Units Per Second <sup>2</sup>

**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.



**Output LP Filter Bandwidth** The Output Filter Bandwidth attribute controls the bandwidth of the servo's low-pass digital output filter. The programmable low-pass output filter is bypassed if the configured Output Filter Bandwidth for this filter is set to zero (the default). This output filter can be used to filter out, or reduce, high frequency variation of the servo module output to the drive. The lower the Output Filter Bandwidth, the greater the attenuation of these high frequency components of the output signal. Unfortunately, since the low-pass filter adds lag to the servo loop which pushes the system towards instability, decreasing the Output Filter Bandwidth usually requires lowering the Position or Velocity Proportional Gain of the system 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.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Output LP Filter Bandwidth	REAL	Hertz

**Integrator Hold Enable** When the Integrator Hold Enable attribute value is configured TRUE, the servo loop temporarily disables any enabled integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion. When the Integrator Hold Enable attribute value is FALSE, all active integrators are always enabled.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Integrator Hold Enable	SINT	

## Servo Limits

This section covers the various servo attributes that either apply limits to various servo loop real-time parameters, such as position and output voltage, or are used in limit checks of servo loop parameters like position error.

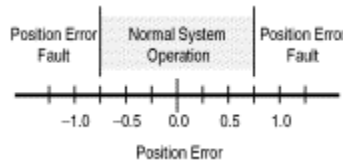
**Maximum Positive/Negative Travel** The Axis Object provides configurable software travel limits via the Maximum Positive and Negative Travel attributes. If the axis is configured for software overtravel limit checking by setting the Soft Overtravel Bit in the Servo Configuration Bit word, and the axis passes outside these maximum travel limits, a Software Overtravel Fault is issued.

When software overtravel checking is enabled, appropriate values for the maximum travel in both the Maximum Positive and Maximum Negative Travel attributes need to be established with Maximum Positive Travel always greater than Maximum Negative Travel. Both of these values are specified in the configured Position Units of the axis.

**Note:** The software travel limits are not enabled until the selected homing sequence is completed.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Maximum Positive Travel	REAL	Position Units
SSV/GSV	Maximum Negative Travel	REAL	Position Units

**Position Error Tolerance** The Position Error Tolerance parameter specifies how much position error the servo tolerates before issuing a Position Error Fault. Like the position lock tolerance, the position error tolerance is interpreted as a  $\pm$  quantity. For example, specifying a position error tolerance of 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 below:



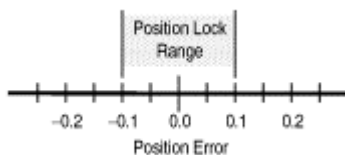
**Figure 13.13 Position Error**

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Error Tolerance	REAL	Position Units

The self tuning routine sets the position error tolerance to twice the following error at maximum speed based on the measured response of the axis. 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** The Position Lock Tolerance attribute value specifies how much position error the servo module tolerates when giving a true Position Locked Status indication. When used in conjunction with the Position Locked Status bit, it is a useful parameter to control positioning accuracy. The Position Lock Tolerance value should be set, in Position Units, to the desired positioning accuracy of the axis.

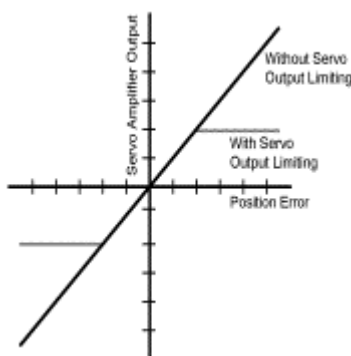
The position lock tolerance value is interpreted as a  $\pm$  quantity. For example, if your position units are Inches, specifying a position lock tolerance of 0.01 provides a minimum positioning accuracy of  $\pm 0.01$  inches as shown in the following diagram.



**Figure 13.14 Position Lock**

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Lock Tolerance	REAL	Position Units

**Output Limit** The Output Limit attribute 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.



**Figure 13.15 Servo Output Limit**

The servo output limit may be used as a software current or torque limit if you are using a servo drive in torque (current) 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  $\pm 10$  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  $\pm 7.5$  Volts, set the servo output limit value to 7.5 volts.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Output Limit	REAL	Volts Range: 0.0 - 10.0

## Servo Offsets

This section covers the various servo attributes that provide offsets to real-time servo loop operation.

**Friction Compensation** It is not unusual for an axis to have enough static friction, so called “sticktion”, that even with a significant position error, refuses to budge. Of course, integral gain can be used to generate enough output to the drive to correct the error, but this approach may not be responsive enough for the application. An alternative is to use Friction Compensation to break sticktion in the presence of a non-zero position error. This is done by adding, or subtracting, a fixed output level, called Friction Compensation, to the Servo Output value based on its current sign.

The Friction Compensation value should be just under the value that would break the sticktion. A larger value will result in the Axis to “dither”, a phenomena describing a rapid back and forth motion of the axis centered on the commanded position.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Friction Compensation	REAL	% Range: - 0% to 100%

**Velocity Offset** Velocity Offset compensation can be used to correct to provide a dynamic velocity correction to the output of the position servo loop. Since this value is updated synchronously every Coarse Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block programming.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Offset	REAL	Position Units per sec.

**Torque Offset** Torque Offset compensation can be used to provide a dynamic torque command correction to the output of the velocity servo loop. Since this value is updated synchronously every Coarse Update Period, the Torque Offset can be tied into custom outer control loop algorithms using Function Block programming.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Torque Offset	REAL	% Range: -100% to 100%

**Output Offset** Another common situation when interfacing an external Servo Drive, particularly for velocity servo drives, is 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.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Output Offset	REAL	Volts Range: +/-10

Output offset compensation can be used to correct this problem by adding a fixed value, called Output Offset, to the Servo Output. This value is chosen to achieve near zero drive velocity when the uncompensated Servo Output value is zero.

## Servo Fault Configuration

**Servo Fault Actions** Each axis can be configured to respond to each of the five types of servo faults in any one of four different ways. This flexibility is important because motion control applications differ widely in their fault action requirements.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Soft Overtravel Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Hard Overtravel Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Position Error Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Feedback Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Feedback Noise Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Drive Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only

### Shutdown

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. 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 which 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. Shutdown is the most severe action to a fault and it is usually used for faults which could endanger the machine or the operator if power is not removed as quickly as possible.

### Stop Command

If a fault action is set to Stop Command, 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. Once 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, motion faults must be handled by the application program. In general, this setting should *only* be used in applications where the standard fault actions are not appropriate. The recommended setting of the fault action configuration parameters—suitable for most applications—are provided as defaults.

## Commissioning Configuration Attributes

The Axis Object provides sophisticated automatic test tuning instructions, which allow it to determine proper settings for the servo loop attributes for each axis. These include not only the polarities, the gains, and also the maximum acceleration, deceleration, and velocity parameters.

Usually, the servo loop parameters need only be tested and tuned once when the motion controller is first integrated into the machine or when the machine is being commissioned at start-up. However, if the load on any axis changes significantly or if the motor or servo amplifier is replaced for any reason, it may be necessary to re-test and re-tune the servo loop parameters.

The Commissioning Configuration Attributes shown in the table below are used to control the axis test and tuning processes that are initiated by the MRHD and MRAT instructions. Therefore, these values should be established before the MRHD or MRAT instructions are executed.

**Test Increment** The Motor Feedback Test Increment attribute is used in conjunction with the MRHD (Motion Run Hookup Diagnostic) instruction to determine the amount of motion that is necessary to satisfy the MRHD initiated test process. This value is typically set to approximately a quarter of a revolution of the motor.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Test Increment	REAL	Position Units

**Tuning Travel Limit** The Tuning Travel Limit attribute is used in conjunction with the MRAT (Motion Run Axis Tuning) instruction to limit the excursion of the axis during the test. If, while performing the tuning motion profile, the servo module determines that the axis will not be able to complete the tuning process before exceeding the Tuning Travel Limit, the servo module will terminate the tuning profile and report that the Tuning Travel Limit was exceeded via the Tune Status attribute. This does not mean that the Tuning Travel Limit was actually exceeded, but that had the tuning process gone to completion that the limit would have been exceeded.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Travel Limit	REAL	Position Units

**Tuning Speed** The Tuning Speed attribute value determines the maximum speed of the MRAT (Motion Run Axis Tune) initiated tuning motion profile. This attribute should be set to the desired maximum operating speed of the motor prior to running the MRAT instruction. The reason for doing this is that the tuning procedure will measure maximum acceleration and deceleration rates based on ramps to and from the Tuning Speed. Thus, the accuracy of the measured acceleration and deceleration capability is reduced by tuning at a speed other than the desired operating speed of the system.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Speed	REAL	Position Units / Sec



**Tuning Torque** The Tuning Torque attribute value determines the maximum torque of the MRAT (Motion Run Axis Tune) initiated tuning motion profile. This attribute should be set to the desired maximum safe torque level prior to running the MRAT instruction. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system. 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. Note that the extrapolation error increases as the Tuning Torque value decreases.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Torque	REAL	%

**Damping Factor** The Damping Factor attribute value is used in calculating the maximum Position Servo Bandwidth (see below) during execution of the MRAT (Motion Run Axis Tune) instruction. In general the Damping Factor attribute controls the dynamic response of the servo axis. When gains are tuned using a small damping factor (like 0.7), a step response test performed on the axis would demonstrate under-damped behavior with velocity overshoot. A gain set generated using a larger damping factor, like 1.0, would produce a system step response that have no overshoot but have a significantly lower servo bandwidth. The default value for the Damping Factor of 0.8 should work fine for most applications.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Damping Factor	REAL	

**Drive Model Time Constant** The value for the Drive Model Time Constant represents lumped model time constant for the drives current loop used by the MRAT instruction to calculate the Maximum Velocity and Position Servo Bandwidth values. The Drive Model Time Constant is the sum of the drive's current loop time constant, the feedback sample period, and the time constant associated with the velocity feedback filter. This value is set to a default value when the axis is configured based on the specific servo module selection. This value is only used by MRAT when the axis is configured for an External Torque Servo Drive.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Drive Model Time Constant	REAL	Sec

**Velocity Servo Bandwidth** The value for the Velocity Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the velocity servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Velocity Servo Bandwidth is the better the dynamic performance of the system. A maximum value for the Velocity Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction will result in dynamic response in keeping with the current value of the Damping Factor described above. Alternatively, the responsiveness of the system can be “softened” by reducing the value of the Velocity Servo Bandwidth before executing the MAAT instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Servo Bandwidth	REAL	Hertz

There are practical limitations to the maximum Velocity Servo Bandwidth for the velocity servo loop based on the drive system and, in some cases, the desired damping factor of the system, Z. Exceeding these limits could result in an unstable servo operation. These bandwidth limitations may be expressed as follows:

For an external velocity loop servo drive,

$$\text{Max Velocity Servo Bandwidth (Hz)} = 0.159 * 2 / \text{Tune Rise Time}$$

For an external torque loop servo drive,

$$\text{Max Velocity Servo Bandwidth (Hz)} = 0.159 * 0.25 * 1 / Z^2 * 1 / \text{Drive Model Time Constant}$$

The factor of 0.159 represents the 1/2PI factor required to convert Radians per Second units to Hertz.

**Position Servo Bandwidth** The value for the Position Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the position servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Position Servo Bandwidth is the better the dynamic performance of the system. A maximum value for the Position Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction will result in dynamic response in keeping with the current value of the Damping Factor described above. Alternatively, the responsiveness of the system can be “softened” by reducing the value of the Position Servo Bandwidth before executing the MAAT instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Servo Bandwidth	REAL	Hertz

There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and current loops of the servo system and the desired damping of the system,  $Z$ . Exceeding these limits could result in an unstable system. These bandwidth limitations may be expressed as follows:

$$\text{Max Position Bandwidth (Hz)} = 0.25 * 1/Z^2 * \text{Velocity Bandwidth (Hz)}$$

For example, if the maximum bandwidth of the velocity servo loop is 40 Hz and the damping factor,  $Z$ , is 0.8, the maximum the maximum position bandwidth is 16 Hz. Based on these numbers the corresponding proportional gains for the loops can be computed.

### Tuning Configuration Bits

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Configuration Bits	DINT	0: Tuning Direction Reverse 1: Tune Position Error Integrator 2: Tune Velocity Error Integrator 3: Tune Velocity Feedforward 4: Tune Acceleration Feedforward 5: Tune Output Low-Pass Filter 6-31: <i>Reserved</i>

### Tuning Direction Reverse

The Tune Direction Reverse bit attribute determines the direction of the tuning motion profile initiated by the MRAT (Motion Run Axis Tune) instruction. If this bit is set (true), motion is initiated in the reverse (or negative) direction.

### **Tune Position Error Integrator**

The Tune Position Error Integrator bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction will calculate a value for the Position Integral Gain. If this bit is clear (false) the value for the Position Integral Gain will be set to zero.

### **Tune Velocity Error Integrator**

The Tune Velocity Error Integrator bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction calculates a value for the Velocity Integral Gain. If this bit is clear (false) the value for the Velocity Integral Gain is set to zero.

### **Tune Velocity Feedforward**

The Tune Velocity Feedforward bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction calculates a value for the Velocity Feedforward Gain. If this bit is clear (false) the value for the Velocity Feedforward Gain is set to zero.

### **Tune Acceleration Feedforward**

The Tune Acceleration Feedforward bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction calculates a value for the Acceleration Feedforward Gain. If this bit is clear (false) the value for the Acceleration Feedforward Gain is set to zero.

### **Tune Output Low-Pass Filter**

The Tune Output Low-Pass Filter bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction calculates a value for the Output Filter Bandwidth. If this bit is clear (false) the value for the Output Filter Bandwidth is set to zero which disables the filter.

## **Servo Drive Status Attributes**

The following sections define the behavior of the various status attributes associated with the Servo Drive specific behavior of the Motion Axis Object. Status attributes are, by definition, read access only. The following Servo specific Status Attributes are divided into 3 categories: Drive Status attributes, Drive Commissioning Status attributes, and Drive Status Bit attributes.

### **Drive Status Attributes**

The list of Drive Status Attributes associated with the Motion Axis Object provides access to servo drive resident information for the axis. These values may be used as part of the user program to perform real time measurements of drive operation. A list of all Drive Status Attributes is shown in the table below.

Since Drive Status Attributes values are resident in the drive, these values need to be transferred to the ControlLogix processor module on a regular basis. To avoid unnecessary communication traffic transferring data that is not of interest, it is necessary to explicitly activate transfer of the specific Drive Status Attribute data from the drive using the Axis Info Select attributes. Thus, a Servo Status Attribute value is **ONLY** valid if the attribute has been selected by one of the Axis Info Select attributes. Otherwise the Drive Status Attribute value is forced to zero.

In order for the above position unit-based attributes to return a meaningful value, the 'Conversion Constant' Axis Configuration Attribute must be established. Furthermore, attributes having velocity or acceleration units (e.g. Position Units / Sec) must also have a valid coarse update period which is established through association with a fully configured Motion Group Object.

Each of the Drive Status Attributes appears in the following Servo block diagram.

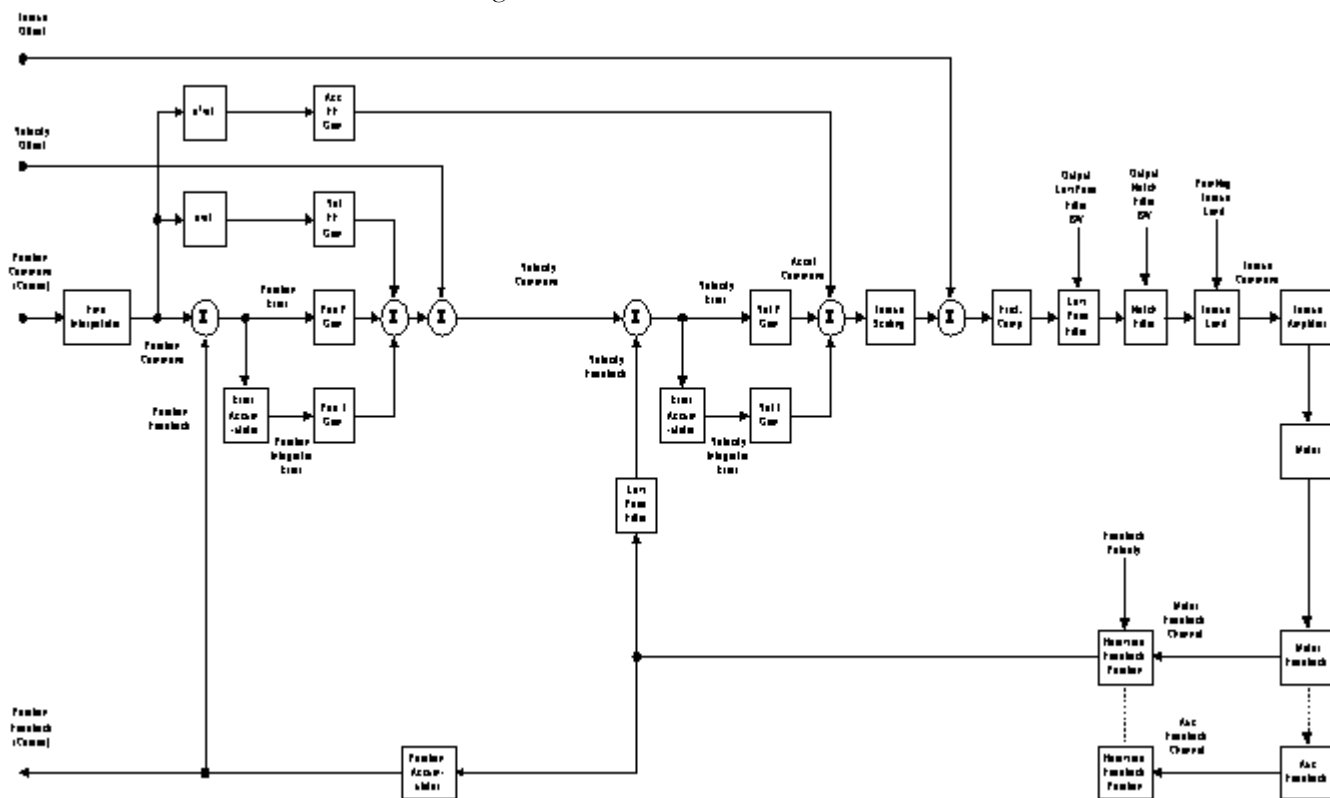


Figure 13.16 Motor Position Servo Loop Diagram

**Position Command** Position Command is the current value of the Fine Command Position into the position loop summing junction, in configured axis Position Units. Within the active servo loop, the Position Command value is used to control the position of the axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Command	REAL	Position Units

**Position Feedback** Position Feedback is the current value of the Fine Actual Position into the position loop summing junction, in configured axis Position Units. Within the servo loop, the Position Feedback represents the current position of the axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Feedback	REAL	Position Units

**Aux Position Feedback** Aux Position Feedback is the current value of the position feedback coming from the auxiliary feedback input.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Aux Position Feedback	REAL	Position Units

**Position Error** Position Error is the difference, in configured axis Position Units, between the command and actual positions of a drive axis. For an axis with an active servo loop, position error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Error	REAL	Position Units

**Position Integrator Error** Position Integrator Error is the running sum of the Position Error, in the configured axis Position Units, for the specified axis. For an axis with an active servo loop, the position integrator error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position. .

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Position Integrator Error	REAL	Position Units - mSec

**Velocity Error** Velocity Error is the difference, in configured axis Position Units per Second, between the commanded and actual velocity of a drive axis. For an axis with an active velocity servo loop, velocity error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Error	REAL	Position Units / Sec

**Velocity Integrator Error** Velocity Integrator Error is the running sum of the Velocity Error, in the configured axis Position Units per Second, for the specified axis. For an axis with an active velocity servo loop, the velocity integrator error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Integrator Error	REAL	Position Units – mSec / Sec

**Velocity Command** Velocity Command is the current velocity reference to the velocity servo loop, in the configured axis Position Units per Second, for the specified axis. The Velocity Command value, hence, represents the output of the outer position control loop. Velocity Command is not to be confused with Command Velocity which represents the rate of change of Command Position input to the position servo loop.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Command	REAL	Position Units / Sec

**Velocity Feedback** Velocity Feedback is the actual velocity of the axis as estimated by the SERCOS module, in the configured axis Position Units per second. The estimated velocity is generated by applying a 1 KHz low-pass filter to the change in actual position over the servo update interval. Velocity Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Velocity Feedback	REAL	Position Units / Sec

**Acceleration Command** Acceleration Command is the current acceleration reference to the output summing junction, in the configured axis Position Units per Second<sup>2</sup>, for the specified axis. The Acceleration Command value, hence, represents the output of the inner velocity control loop. Acceleration Command is not to be confused with Command Velocity, which represents the rate of change of Command Position input to the position servo loop.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Acceleration Command	REAL	Position Units / Sec <sup>2</sup>

**Acceleration Feedback** Acceleration Feedback is the actual velocity of the axis as estimated by the servo module, in the configured axis Position Units per Second<sup>2</sup>. The Estimated Acceleration is calculated by taking the difference in the Estimated Velocity over the servo update interval. Acceleration Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Acceleration Feedback	REAL	Position Units / Sec <sup>2</sup>

**Marker Distance** Marker Distance is the distance between the axis position at which a home switch input was detected and the axis position at which the marker event was detected. This value is useful in aligning a home limit switch relative to a feedback marker pulse to provide repeatable homing operation.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Marker Distance	REAL	Position Units

**Torque Command** This is the command value when operating in torque mode.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Torque Command	REAL	%Rated

**Torque Feedback** This is the torque feedback value when operating in torque mode.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Torque Feedback	REAL	%Rated



**Pos./Neg. Dynamic Torque Limit** These parameters represent the currently operative maximum positive and negative torque/current limit magnitude. Each value should be the lowest value of all torque/current limits in the drive at a given time. These limits include the amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and the motor thermal limit.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Pos. Dynamic Torque Limit Neg. Dynamic Torque Limit	REAL	%Rated

**Motor Capacity** This parameter displays the present utilization of motor capacity as a percent of rated capacity.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motor Capacity	REAL	%

**Drive Capacity** This parameter displays the present utilization of drive capacity as a percent of rated capacity.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Drive Capacity	REAL	%

**Power Capacity** This parameter displays the present utilization of the axis power supply as a percent of rated capacity.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Power Capacity	REAL	%

**Bus Regulator Capacity** This parameter displays the present utilization of the axis bus regulator as a percent of rated capacity.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Bus Regulator Capacity	REAL	%

**Motor Electrical Degrees** This parameter is the present electrical angle of the motor shaft.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motor Electrical Angle	REAL	Degrees

**DC Bus Voltage** This parameter is the present voltage on the DC Bus of the drive.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	DC Bus Voltage	DINT	Volts

**Torque Limit Source** This parameter displays the present source (if any) of any torque limiting for the axis.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Torque Limit Source	DINT	0 = Not Limited 1 = Neg.e Torque Limit 2 = Pos. Torque Limit 3 = Amp Peak Limit 4 = Amp I(t) Limit 5 = Bus Regulator Limit 6 = Bipolar Torque Limit 7 = Motor Peak Limit 8 = Motor I(t) Limit 9 = Voltage Limit

## Drive Status Bit Attributes

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Drive Status Bits	DINT	Direct Access Entire DINT - <b>DriveStatus</b> 0: Servo Action Status - <b>No Tag</b> 1: Drive Enable Status - <b>No Tag</b> 2: Axis Shutdown Status - <b>No Tag</b> 3: Process Status - <b>ProcessStatus</b> 4: <i>Reserved</i> 5: <i>Reserved</i> 6: Home Input Status - <b>HomeInputStatus</b> 7: Registration 1 Input Status - <b>Reg1InputStatus</b> 8: Registration 2 Input Status - <b>Reg2InputStatus</b> 9: Positive Overtravel Input Status - <b>PosOvertravelInputStatus</b> 10: Negative Overtravel Input Status - <b>NegOvertravelInputStatus</b> 11: Enable Input Status - <b>EnableInputStatus</b> 12: Acceleration Limit Status - <b>AccelLimitStatus</b> 13: Absolute Reference Status - <b>AbsoluteReferenceStatus</b> 14: <i>Reserved</i> 15: <i>Reserved</i> 16: Velocity Lock Status - <b>VelocityLockStatus</b> 17: Velocity Standstill Status - <b>VelocityStandstillStatus</b> 18: Velocity Threshold - <b>VelocityThresholdStatus</b> 19: Torque Threshold - <b>TorqueThresholdStatus</b> 20: Torque Limit Status - <b>TorqueLimitStatus</b> 21: Velocity Limit Status - <b>VelocityLimitStatus</b> 22: Position Lock Status - <b>PositionLockStatus</b> 23: Power Limit Status - <b>PowerLimitStatus</b> 24: <i>Reserved</i> 25: Lower Velocity Threshold Status - <b>LowVelocityThresholdStatus</b> 26: High Velocity Threshold Status - <b>HighVelocityThresholdStatus</b> 27-31: <i>Reserved</i>

### **Servo Action Status**

The Servo Action Status bit attribute is set when servo loops on the associated with the axis is currently enabled and able to follow command. If the bit is not set then servo action is disabled.

### **Drive Enable Status**

The Drive Enable Status bit attribute is set when the drive's power structure associated with the axis has been activated. If the bit is not set then drive's power structure is currently deactivated.

### **Shutdown Status**

The Shutdown Status bit attribute is set when the associated axis is currently in the Shutdown state. As soon as the axis is transitioned from the Shutdown state to another state, the Shutdown Status bit is cleared.

### **Process Status**

The Process Status bit attribute is set when there is an axis tuning operation or an axis hookup diagnostic test operation in progress on the associated physical axis.

### **Home Input Status**

The Home Input Status bit attribute represents the current state of the dedicated Home input. This bit is set if the Home input is active and clear if inactive.

### **Registration 1/2 Input Status**

The Registration Input 1 and Registration Input 1 Status bit attributes represent the current state of the corresponding dedicated Registration input. This bit is set if the registration input is active and clear if inactive.

### **Positive Overtravel Input Status**

The Positive Overtravel Input Status bit attribute represents the current state of the dedicated Positive Overtravel input. This bit is set if the Positive Overtravel input is active and clear if inactive.

### **Negative Overtravel Input Status**

The Negative Overtravel Input Status bit attribute represents the current state of the dedicated Negative Overtravel input. This bit is set if the Negative Overtravel input is active and clear if inactive.

### **Enable Input Status**

The Enable Input Status bit attribute represents the current state of the dedicated Enable input. This bit is set if the Enable input is active and clear if inactive.

### **Acceleration Limit Status**

The Acceleration Limit Status bit attribute is set when the magnitude of the commanded acceleration to the velocity servo loop input is greater than the configured Velocity Limit.

### **Absolute Reference Status**

The Absolute Reference Status bit attribute is set after an absolute homing procedure. The bit will remain set unless the drive resets its configuration parameters to default values or an active or passive home is performed on the axis. If the bit is clear, it indicates that the reported position of the axis has not been, or is no longer, referenced to the absolute machine reference system established by an absolute homing procedure.

### **Velocity Lock Status**

The Velocity Lock Status bit attribute is set when the magnitude of the physical axis Velocity Feedback is within the configured Velocity Window of the current velocity command.

### **Velocity Standstill Status**

The Velocity Standstill Status bit attribute is set when the magnitude of the physical axis Velocity Feedback is within the configured Velocity Standstill Window of zero speed.

### **Velocity Threshold**

The Velocity Threshold Status bit attribute is set when the magnitude of the physical axis Velocity Feedback is less than the configured Velocity Threshold.

### **Torque Limit Status**

The Torque Limit Status bit attribute is set when the magnitude of the axis torque command is greater than the configured Torque Limit.

### **Velocity Limit Status**

The Velocity Limit Status bit attribute is set when the magnitude of the commanded velocity to the velocity servo loop input is greater than the configured Velocity Limit.

### Position Lock Status

The Position Lock Status bit attribute is 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. If this bit is not set then the magnitude of the axis position error is greater than the configured Position Lock Tolerance value.

### Power Limit Status

The Power Limit Status bit attribute is set when the magnitude of the actual supplied power is greater than the configured Power Threshold.

### Axis Control Bit Attributes

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Control Bits	DINT	0: Abort Process Request 1: Shutdown Request 2: <i>Reserved</i> 3: Abort Home Request 4: Abort Event Request 5-14: <i>Reserved</i> 15: Change Cmd Reference 16-31: <i>Reserved</i>

#### Abort Process

When the Abort Process bit is set, any active tuning or test process on the drive axis is aborted

#### Shutdown Request

When the Shutdown Request bit is set, the drive axis is forced into the shutdown state.

#### Abort Home Request

When the Abort Home Request bit is set, any active homing procedures are cancelled.

#### Abort Event Request

When the Abort Event Request bit is set, any active registration or watch event procedures are cancelled.

## Change Cmd Reference

The Change Command Reference bit attribute is set when the Logix processor has switched to a new position coordinate system for command position. The servo drive processor uses this bit when processing new command position data from the Logix processor to account for the offset implied by the shift in the reference point. The bit is cleared when the drive axis acknowledges completion of the reference position change by clearing its Change Position Reference bit.

## Axis Response Bit Attributes

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Response Bits	DINT	0: Abort Process Acknowledge 1: Shutdown Acknowledge 2: <i>Reserved</i> 3: Abort Home Acknowledge 4: Abort Event Acknowledge 5-14: <i>Reserved</i> 15: Change Pos Reference 16-31: <i>Reserved</i>

### Abort Process Acknowledge

When the Abort Process Acknowledge bit is set, the servo module acknowledges that the tuning or test process has been aborted.

### Shutdown Request Acknowledge

When the Shutdown Acknowledge bit is set, the servo module acknowledges that the axis has been forced into the shutdown state.

### Abort Home Acknowledge

When the Abort Home Acknowledge bit is set, the servo module acknowledges that the active home procedure has been aborted.

### Abort Event Acknowledge

When the Abort Event Acknowledge bit is set, the servo module acknowledges that the active registration or watch position event has been aborted.

### **Change Pos Reference**

The Change Position Reference bit attribute is set when the Servo loop has switched to a new position coordinate system. The Logix processor uses this bit when processing new position data from the servo drive to account for the offset implied by the shift in the reference point. The bit is cleared when the Logix processor acknowledges completion of the reference position change by clearing its Change Cmd Reference bit.

**Drive Fault Bit Attributes** All of the fault bit attributes defined below can be handled by the ControlLogix processor as a Major Fault by configuring the associated Group Object's "General Fault Type Mechanism" attribute accordingly. Otherwise any specific fault handling must be done as part of the user program.



GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Drive Fault Bits	DINT	Direct Access Entire DINT - <b>DriveFaults</b> 0: Positive Soft Overtravel Fault <b>-PosSoftOvertravelFault</b> 1: Negative Soft Overtravel Fault <b>-NegSoftOvertravelFault</b> 2: Positive Hard Overtravel Fault <b>-PosHardOvertravelFault</b> 3: Negative Hard Overtravel Fault <b>-NegHardOvertravelFault</b> 4: Feedback Fault <b>-FeedbackFault</b> 5: Feedback Noise Fault <b>-FeedbackNoiseFault</b> 6: Auxiliary Feedback Fault <b>-AuxFeedbackFault</b> 7: Auxiliary Feedback Noise Fault <b>-AuxFeedbackNoiseFault</b> 8 - 12: (reserved) 13: Ground Short Fault <b>-GroundShortFault</b> 14: Drive Hardware Fault <b>-DriveHardFault</b> 15: Overspeed Fault <b>-OverspeedFault</b> 16: Overload Fault <b>-OverloadFault</b> 17: Drive Overtemperature Fault <b>-DriveOvertempFault</b> 18: Motor Overtemperature Fault <b>-MotorOvertempFault</b> 19: Drive Cooling Fault <b>-DriveCoolingFault</b> 20: Drive Control Voltage Fault <b>-DriveControlVoltageFault</b> 21: Feedback Fault <b>-Feedback Fault</b> 22: Commutation Fault <b>-CommutationFault</b> 23: Drive Overcurrent Fault <b>-DriveOvercurrentFault</b> 24: Drive Overvoltage Fault <b>-DriveOvervoltageFault</b> 25: Drive Undervoltage Fault <b>-DriveUndervoltageFault</b> 26: Power Phase Loss Fault <b>-PowerPhaseLossFault</b> 27: Position Error Fault <b>-PositionErrorFault</b> 28: Sercos Fault <b>-SERCOSFault</b> 29: Overtravel Fault <b>-No Tag</b> 30-31: <i>Reserved</i>

### Positive/Negative Software Overtravel Faults

If either the Positive Soft Overtravel Status or Negative Soft Overtravel Status bit attributes are set it indicates that the axis has traveled, or attempted to travel, beyond the current configured values for Maximum Positive Travel or Maximum Negative Travel, respectively. As soon as the axis is moved back within these travel limits, the corresponding Overtravel Status bit is cleared.

### Positive/Negative Hardware Overtravel Faults

If either the Positive Hard Overtravel Status or Negative Hard Overtravel Status bit attributes are set it indicates that the axis has traveled beyond the current 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.

### Position Error Fault

If the Position Error Fault bit attribute is set it indicates that the servo has detected that the axis position error has exceeded the current configured value for Position Error Tolerance. 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.

### Feedback 1 or Feedback 2 Fault

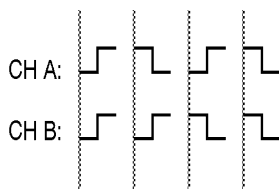
If the Feedback Fault bit is set for a specific feedback source, it indicates that one of the following conditions occurred:

- The differential electrical signals for one or more of the feedback channels (e.g., A+ and A-, B+ and B-, or Z+ and Z- for an encoder) 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 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.

## Feedback 1 or Feedback 2 Noise Fault

If the Feedback Noise Fault bit attribute is set for a specific feedback source, it indicates that simultaneous transitions of the feedback A and B channels has been detected by the servo module which is referred to generally as feedback noise. When the feedback device is an encoder, feedback noise (shown below) 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.



**Figure 13.17 Channel Quadrature**

For example, loss of channel quadrature for an encoder can be caused by physical misalignment of the feedback transducer components, or excessive capacitance (or other delays) on the encoder signals. Proper grounding and shielding techniques can usually cure radiated noise problems. 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.

## Ground Short Fault

When the drive detects an imbalance in the D.C bus supply current, the Ground Short Fault bit is set, indicating that current is flowing through an improper ground connection.

## Drive Hardware Fault

The Drive Hardware Fault bit is set when the drive detects a serious hardware fault.

## Overspeed Fault

The Overspeed Fault bit is 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.

## Overload Fault

When the load limit of the motor/drive is first exceeded, the Overload warning bit is set. If, however, the condition persists, the Overload fault is set. Often this bit is tied into the IT limit of the drive.

### **Drive Overtemperature Fault**

The Drive Overtemperature Fault bit is set when the drive's temperature exceeds the drive shutdown temperature.

### **Motor Overtemperature Fault**

The Motor Overtemperature Fault bit is set when the motor's temperature exceeds the motor shutdown temperature.

### **Drive Cooling Fault**

The Drive Cooling Fault bit is set when the ambient temperature surrounding the drive's control circuitry temperature exceeds the drive ambient shut-down temperature.

### **Drive Control Voltage Fault**

The Drive Control Voltage Fault bit is set when the power supply voltages associated with the drive circuitry fall outside of acceptable limits.

### **Feedback Fault**

The Feedback Fault bit is 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.

### **Commutation Fault**

The Commutation Fault bit is 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.

### **Drive Overcurrent Fault**

The Drive Overcurrent Fault bit is set when drive output current exceeds the predefined operating limits for the drive.

### **Drive Overvoltage Fault**

The Drive Overvoltage Fault bit is set when drive DC bus voltage exceeds the predefined operating limits for the bus.

### **Drive Undervoltage Fault**

The Drive Undervoltage Fault bit is set when drive DC bus voltage is below the predefined operating limits for the bus.

## Power Phase Loss Fault

The Power Phase Loss Fault bit is set when the drive detects that one or more of the three power line phases is lost from the 3 phase power inputs.

## SERCOS Fault

The SERCOS Fault bit is set when either a requested SERCOS procedure fails to execute properly or the associated drive node has detected a SERCOS communication fault.

**Module Fault Bit Attributes** The Module Fault Bit attribute is a collection of all faults that have module scope as opposed to axis scope. Generally, these module faults are reflected by all axes supported by the associated SERCOS module.

Module Fault attribute information is passed from a physical module or device to the controller via an 8-bit value contained in the header of the Synchronous Input connection assembly. Thus, these fault bits are updated every coarse update period by the Motion Task. The module's map driver should also monitor module Faults so module fault conditions can be reflected to the user through the Module Properties dialog.

All of the fault bit attributes defined below can be handled by the ControlLogix processor as a Major Fault by configuring the associated Group Object's "General Fault Type Mechanism" attribute accordingly. Otherwise any specific fault handling must be done as part of the user program.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Module Fault Bits	DINT	Direct Access Entire DINT - <b>ModuleFaults</b> 0: Control Sync Fault - <b>ControlSyncFault</b> 1: Module Sync Fault - <b>ModuleSyncFault</b> 2: Timer Event Fault - <b>TimerEventFault</b> 3: Module Hardware Fault - <b>ModuleHardwareFault</b> 4: SERCOS Communications Fault - <b>SERCOSRingFault</b> 5-31: <i>Reserved</i>

### **Control Sync Fault**

The Control Sync Fault bit attribute is 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. The Synchronous Connection Fault bit is cleared when the connection is reestablished.

### **Module Sync Fault**

The Module Sync Fault bit attribute is 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. The Synchronous Connection Fault bit is cleared when the connection is reestablished.

### **Timer Event Fault**

If the Timer Event Fault bit attribute is set it indicates that 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). The Timer Event Fault bit can only be cleared by reconfiguration of the motion module.

### **Module Hardware Fault**

If the Module Hardware Fault bit attribute is set it indicates that the associated servo module has detected a hardware problem that, in general, is going to require replacement of the module to correct.

### **SERCOS Ring Fault**

The SERCOS Ring Fault bit is set when the SERCOS module detects that a problem has occurred on the SERCOS ring; i.e., the light has been broken or a drive has been powered down.

**Drive Warning Bit Attributes** All of the warning bit attributes defined below are not supported in the initial release of this object.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Drive Warning Bits	DINT	Direct Access Entire DINT - <b>DriveWarnings</b> 0: Drive Overload Warning <b>-DriveOverloadWarning</b> 1: Drive Overtemperature Warning <b>-DriveOvertempWarning</b> 2: Motor Overtemperature Warning <b>-MotorOvertempWarning</b> 3: Cooling Error Warning <b>-CoolingErrorWarning</b> 4-31: <i>Reserved</i>

### Overload Warning

When the load limit of the motor is exceeded, the Overload Warning bit is set. If the condition persists, an Overload Fault will occur. This warning bit gives the control program an opportunity to reduce motor loading to avoid a future shutdown situation.

### Drive Overtemperature Warning

When the over-temperature limit of the drive is exceeded, the Drive Overtemperature Warning bit is set. If the condition persists, a Drive Overtemperature Fault will occur. This warning bit gives the control program an opportunity to reduce motor loading, or increasing drive cooling, to avoid a future shutdown situation.

### Motor Overtemperature Warning

When the over-temperature limit of the motor is exceeded, the Motor Overtemperature Warning bit is set. If the condition persists, an Motor Overtemperature Fault will occur. This warning bit gives the control program an opportunity to reduce motor loading, or increasing motor cooling, to avoid a future shutdown situation.

### Cooling Error Warning

When the ambient temperature limit inside the drive enclosure is exceeded, the Cooling Error Warning bit is set. If the condition persists, a Cooling Error Fault will occur. This warning bit gives the control program an opportunity to increasing drive cooling, to avoid a future shutdown situation.

**Attribute Error Code** When an Axis Configuration Fault occurs, one or more axis parameters associated with a SERCOS module or drive has not been successfully updated to match the value of the corresponding parameter of the local controller. The fact that the configuration of the drive axis no longer matches the configuration of the local controller is a serious fault and results in the shutdown of the faulted axis. The Attribute Error Code is reset to zero by reconfiguration of the motion module.

Axis Configuration Fault information is passed from the SERCOS module or device to the controller via a 16-bit ASA status word contained in the Set Attribute List service response received by the controller. A Set Attribute List service to the motion module can be initiated by a software Set Attribute List service to the controller, or by an SSV instruction within the controller's program, referencing a servo attribute. Various routines that process responses to motion services are responsible for updating these attributes.

The Set and Get service responses provide a status response with each attribute that was processed. That status value is defined by ASA as follows: UINT16, Values 0-255 (0x00-0xFF) are reserved to mirror common service status codes. Values 256 – 65535 are available for object/class attribute specific errors.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Attribute Error Code	INT	ASA Error code returned by erred set attribute list service to the module.

**Attribute Error ID** The Attribute Error ID is used to retain the ID of the servo attribute that returned a non-zero attribute error code resulting in an Axis Configuration Fault. The Attribute Error ID defaults to zero and, after a fault has occurred may be reset to zero by reconfiguration of the motion module.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Attribute Error ID	INT	Attribute ID associated with non-zero Attribute Error Code.

**SERCOS Error Code** The SERCOS Error Code value can be used to identify the source of the drive parameter update failure that resulted in the Axis Configuration Fault. The error codes for this attribute are derived from the IEC-1394 SERCOS Interface standard. .

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	SERCOS Error Code	INT	Error code returned by SERCOS module indicating source of drive parameter update failure.



## Commissioning Status Attributes

The list of Commissioning Status Attributes associated with the Axis Object provides access to attributes associated with the state of various motion instruction generated commissioning processes. Motion instructions involved in commissioning an axis are MRAT (Motion Run Axis Tune) and MRHD (Motion Run Hookup Diagnostic) which are described in detail in the AC Motion Instruction Specification. Commissioning Status Attributes are primarily used by external software (e.g. RSLogix 5000) to implement the Test and Tuning dialogs associated with the axis configuration tool. However, these same attributes may also be used as part of the user program to implement a “built-in” axis test and tuning procedure.

In order for position unit-based attributes to return a meaningful value, the ‘Conversion Constant’ Axis Configuration Attribute must be established. Furthermore, attributes having time units (Position Units / Sec) must also have a valid coarse update period which is established through association with a fully configured Motion Group Object.

**Test Status** The Test Status attribute returns status of the last run MRHD (Motion Run Hookup Diagnostic) instruction that initiates a hookup diagnostic process on the targeted SERCOS module axis. The Test Status attribute can, thus, be used to determine when the MRHD initiated operation has successfully completed. Conditions may occur, however, that make it impossible for the control to properly perform the operation. When this is the case, the test process will be automatically aborted and a test fault reported that is stored in the Test Status output parameter. .

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Test Status	INT	0 = test process successful 1 = test in progress 2 = test process aborted by user 3 = test process time-out fault (~2 seconds) 4 = test failed – servo fault 5 = test failed – insufficient test increment 6 = test failed – wrong polarity 7 = test failed – missing signal 8 = test failed – device comm error 9 = test failed – feedback config error 10 = test failed – motor wiring error

**Test Direction Forward** The Test Direction Forward attribute reports the direction of axis travel during hookup test as seen by the servo module during the last test process initiated by a MRHD (Motion Run Hookup Test) instruction. A Test Direction value of 1 (true) indicates that the direction of motion as observed by the SERCOS drive was in the forward (or positive) direction.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Test Direction Forward	SINT	

**Tune Status** The Tune Status attribute returns status of the last run MRAT (Motion Run Axis Tuning) instruction that initiates a tuning process on the targeted SERCOS module axis. The Tune Status attribute can, thus, be used to determine when the MRAT initiated operation has successfully completed. Conditions may occur, however, that make it impossible for the control to properly perform the operation. When this is the case, the tune process will be automatically aborted and a tune fault reported that is stored in the Tune Status output parameter. .

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Status	INT	Enumeration: 0 = tune process successful 1 = tune in progress 2 = tune process aborted by user 3 = tune process time-out fault 4 = tune process failed due to drive fault 5 = axis reached Tuning Travel Limit 6 = axis polarity set incorrectly 7 = tune measurement fault 8 = tune configuration fault

**Tune Acceleration/Deceleration Time** The Tune Acceleration Time and Tune Deceleration Time attributes return acceleration and deceleration time in seconds for the last run MRAT (Motion Run Axis Tune) instruction. These values are used to calculate the Tune Acceleration and Tune Deceleration attributes.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Acceleration Time	REAL	Sec
GSV	Tune Deceleration Time	REAL	Sec

**Tune Acceleration/Deceleration** The Tune Acceleration Time and Tune Deceleration attributes return the measured acceleration and deceleration values for the last run MRAT (Motion Run Axis Tuning) instruction. These values are used, in the case of an external torque servo drive configuration, to calculate the Tune Inertia value of the axis, and are also typically used by a subsequent MAAT (Motion Apply Axis Tune) to determine the tuned values for the Maximum Acceleration and Maximum Deceleration attributes.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Acceleration	REAL	Position Units / Sec <sup>2</sup>
GSV	Tune Deceleration	REAL	Position Units / Sec <sup>2</sup>

**Tune Inertia** The Tune Inertia value represents the total inertia for the axis as calculated from the measurements made during the last MRAT (Motion Run Axis Tune) initiated tuning process. In actuality, the units of Tune Inertia are not industry standard inertia units but rather in terms of percent (%) of rated drive output per MegaCounts/Sec<sup>2</sup> of feedback input. In this sense it represents the input gain of torque servo drive. These units represent a more useful description of the inertia of the system as seen by the servo controller. The Tune Inertia value is used by the MAAT (Motion Apply Axis Tune) instruction to calculate the Torque Scaling.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Tune Inertia	REAL	% / MegaCounts Per Sec <sup>2</sup>

If the Tune Inertia value exceeds 100 %Rated/MegaCounts Per Second<sup>2</sup>, performance of the digital servo loop may be compromised due to excessive digitalization noise associated with the velocity estimator. This noise is amplified by the Torque Scaling gain which is related to the Tune Inertia factor and passed on to the torque output of the drive. A high Tune Inertia value can, thus, result in excitation of mechanical resonances and also result in excessive heating of the motor due to high torque ripple. The only solution to this problem is to lower the loop bandwidths and optionally apply some output filtering.

Since the Tune Inertia value represents a measure of the true system inertia, this situation can occur when driving a high inertia load relative to the motor, i.e. a high inertia mismatch. But it can also occur when working with a drive that is undersized for the motor or with a system having low feedback resolution. In general, the lower the Tune Inertia the better the performance of the digital servo loops will approximate that of an analog servo system.

Enhancements have been made to the Logix tuning algorithm to address excessive noise issues by managing quantization noise levels. The product of the Tune Inertia (% Rated/MCPS) and the Velocity Servo BW (Hertz) can be calculated to directly determine quantization noise levels. Based on this product, the tuning algorithm can take action to limit high frequency noise injection to the motor. These are the actions that have been implemented:

- For motors with a Tune Inertia BW product of 1000 or more, the LP Filter will be applied with a Filter BW of 5x the Velocity Servo Bandwidth in Hertz. This will limit the amount of phase lag introduced by the LP filter to ~12 degrees which is relatively small compared to the 30 to 60 degrees of phase margin that we have for a typical tuned servo system. With a typical tuned LP filter BW value of 200 Hz, we can expect the high frequency quantization noise in the 1 KHz range to be attenuated roughly by a factor of 5.
- When the Tune Inertia BW product reaches 4000 or more, the LP filter alone is not going to be enough to manage the quantization noise level. So the tune algorithm will begin to taper the system bandwidth by the ratio of  $4000 / (\text{Tune Inertia} * \text{Vel Servo BW})$ . This will hold the quantization noise level at a fixed value, independent of the Tune Inertia BW product. For example, Dave's 420 motor with a Tune Inertia value of 213 and a Vel Servo BW of 41 Hz (8733 Inertia BW product) will tune with a Pos P Gain of 46 and a Vel P Gain of 117 and LP Filter BW of 93. This he has found to be a good noise free gain set.

## Servo Drive Configuration Attributes

The following sections define the behavior of all the various configuration attributes associated with the Servo Drive data type of the Motion Axis Object. The attributes, by definition, have read-write access. The Servo Drive Configuration Attributes are divided into seven categories: Drive Configuration, Motor and Feedback, Drive Gains, Drive Limits, Drive Offsets, Drive Power, and Drive Commissioning attributes. These categories correspond roughly to the organization of the RSLogix 5000 Axis Properties pages.

Many of the following Drive Configuration attributes are associated with corresponding attributes contained in the SERCOS Axis Object associated with the 1784-PM16SE 16-Axis SERCOS interface module. When any of these attributes are modified by a Set Attribute List service or an SSV instruction within the user program, the local processor value for the attribute is immediately changed and a Set Attribute List service to the SERCOS interface module is initiated to update the working value stored in the drive. The progress of this update can be monitored, if necessary, within the user program through the Configuration Update in Process bit of the Axis Status Bits attribute.

**Drive Configuration** The Drive Configuration attributes provide basic drive configuration information. These parameters are used to determine the specific drive, axis type, servo configuration, as well as determine drive polarity and fault handling behavior.

**Drive Axis ID** The Drive Axis ID attribute contains the ASA Product Code of the drive amplifier associated with the axis. If the Product Code does not match that of the actual drive amplifier, an error is generated during the configuration process.

Attribute ID	External Access	GSV/SSV Access	Attribute Name	Data Type	Values
0x120	Set*/Get	GSV	Drive Axis ID	INT	Product Code of Drive Amplifier

**Axis Type** The Axis Type attribute is used to establish the intended use of the axis. If the axis is intended for full servo operation than a value of “2” is required. If only the position information from the feedback interface is of interest, than a Axis Type should be set to “1”. Finally, if the axis is unused in the application, which is a common occurrence when there are an odd number of axes in the system, then the Axis Type associated with the unused axis should be set to “0”. Axis Type is not only used to qualify many operations associated with the axis servo loop, it also controls the behavior of the SERCOS module’s Axis Status LEDs. An Axis Type of “1” (Position Only) results in the DRIVE LED being blanked, while a value of “0” (Unused) blanks both the FDBK and DRIVE LEDs.

External software (RSLogix 5000) also uses the current configured value for Drive Axis Configuration to control the look of many of the tab dialogs associated with the axis configuration tool.

Axis Type attribute can be set ONLY if Axis is not assigned to a group.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Type	INT	0 = unused 1 = feedback only 2 = servo

**Servo Loop Configuration** The Servo Loop Configuration attribute determines the specific configuration of the servo loop topology when the Drive Axis Configuration is set to “servo”. The Servo Loop Configuration establishes several advanced drive configuration attributes that are part of the SERCOS Interface standard.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Servo Loop Configuration	INT	0 = custom 1 = feedback only 2 = aux. feedback only 3 = position servo 4 = aux. position servo 5 = dual position servo 6 = dual command servo 7 = aux. dual command servo 8 = velocity servo 9 = torque servo

**Advanced Servo Configuration Attributes** The advanced attributes map directly to SERCOS IDNs. For a detailed description of these attributes refer to the corresponding IDN descriptions found in the SERCOS Interface standard. Since these attributes are automatically configured based on the current Servo Loop Configuration, the user need not be concerned with manually configuring each of these attributes.

---

**ATTENTION**


Changing the auto-configured values of the above advanced attributes can result in unpredictable motion behavior.

---

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Primary Operation Mode	INT	Bit Map: x000 = no mode x001 = torque servo x010 = vel servo x011 = pos servo w/ fdbk1 x100 = pos servo w/ fdbk2 x101 = pos servo w/ fdbk1 and 2 x110 = (reserved) x111 = no servo
GSV	Telegram Type	INT	Enumeration: 0 = no cyclic data 1 = trq cmd 2 = vel cmd, vel fbk 3 = vel cmd, pos fbk 4 = pos cmd, pos fbk 5 = pos/vel cmd, pos fbk and vel fbk 6 = vel cmd 7 = applic. Telegram (default)
GSV	AT Configuration list	Struct { INT; DINT [16]}	Struct {length; data[ ]}
GSV	MDT Configuration list	Struct { INT; DINT [4]}	Struct {length; data[ ]}  its

### Fault Configuration Bits

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Fault Configuration Bits	DINT	Bit Field: 0: Soft Overtravel Checking 1: Hard Overtravel Checking  1-31: <i>Reserved</i> Overtravel Checking for Linear Axis Only; Change to Rotary or Overtravel Checking requires Home range checks.

## Soft Overtravel Checking

When the Soft Overtravel Checking bit is set it enables a periodic test that monitors the current position of the axis and issues a Positive Soft Overtravel Fault or Negative Soft Overtravel Fault if ever the axis position travels outside the configured travel limits. The travel limits are determined by the configured values for the Maximum Positive Travel and Maximum Negative Travel attributes. This software overtravel check is not a substitute, but rather a supplement, for hardware overtravel fault protection which uses hardware limit switches to directly stop axis motion at the drive and deactivate power to the system. If the Soft Overtravel Checking bit is clear (default), then no software overtravel checking is done.

Software overtravel checking is only available for a linear servo axes.

## Hard Overtravel Checking

When the Hard Overtravel Checking bit is set it enables a periodic test that monitors the current state of the positive and negative overtravel limit switch inputs and issues a Positive Hard Overtravel Fault or Negative Hard Overtravel Fault if ever the axis position travels activates the limit switch inputs. If the Hard Overtravel Checking bit is clear (default), then no overtravel limit switch input checking is done.

Hardware overtravel checking is only available for a linear servo axes.

## Drive Scaling Bits

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Drive Scaling Bits	DINT	Bit Field: 0: Standard/Custom Scaling 1-31: <i>Reserved</i>

## Custom Scaling

The Custom Scaling bit attribute is used to enable custom scaling using the various scaling parameters defined by the SERCOS Interface standard. When the bit is clear (default), the scaling parameters are all set based on the preferred Rockwell Automation SERCOS drive scaling factors.



**Advanced Scaling Attributes** The advanced attributes are derived from the Drive Scaling Bits attribute and map directly to SERCOS IDNs. Thus, for a detailed description of these attributes refer to the corresponding IDN descriptions found in the SERCOS Interface standard. Since these attributes are automatically configured to appropriate defaults, the user need not be concerned with manually configuring each of these attributes.

**ATTENTION**

Changing the auto-configured values of the above advanced attributes can result in unpredictable motion behavior.

**Rotational Position Resolution**

You may want to specifically configure for fractional unwind applications or multi-turn absolute applications requiring cyclic compensation. When the Unwind value for a rotary application does not work out to be an integer value, the Rotational Position Scaling attribute may be modified to a value which is integer divisible by the Unwind value.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Position Data Scaling	INT	Default: rotary axis in Degrees
GSV	Position Data Scaling Factor	DINT	(see IEC 1491)
GSV	Position Data Scaling Exp.	INT	(see IEC 1491)
GSV	Rotational Pos. Resolution	DINT	Drive Units per Rev
GSV	Velocity Data Scaling	INT	Default: rotary axis in RPM
GSV	Velocity Data Scaling Factor	DINT	(see IEC 1491)
GSV	Velocity Data Scaling Exp.	INT	(see IEC 1491)
GSV	Accel Data Scaling	INT	Default: rotary axis in Rad/sec. <sup>2</sup>
GSV	Accel Data Scaling Factor	DINT	(see IEC 1491)
GSV	Accel Data Scaling Exp.	INT	(see IEC 1491)
GSV	Torque/Force Data Scaling	INT	Default: %
GSV	Torque Data Scaling Factor	DINT	(see IEC 1491)
GSV	Torque Data Scaling Exp.	INT	(see IEC 1491)

## Drive Polarity

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Drive Polarity	DINT	Enumeration: 0 = Custom Polarity 1 = Positive Polarity 2 = Negative Polarity

### Custom Polarity

Custom Polarity is used to enable custom polarity configurations using the various polarity parameters defined by the SERCOS Interface standard.

### Positive/Negative Polarity

Positive and Negative Polarity bit attribute determine the overall polarity of the servo loop of the drive. All the advanced polarity parameters are automatically set based on whether the Drive Polarity is configured as Positive or Negative. Proper wiring guarantees that the servo loop is closed with negative feedback. However there is no such 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 is configured automatically using the MRHD and MAHD motion instructions. Refer to the Logix Motion Instruction Specification for more information on these hookup diagnostic instructions.

**Advanced Polarity Attributes** The Advanced Polarity Attributes are derived from the Drive Polarity Bits attribute and map directly to SERCOS IDNs. Thus, for a detailed description of these attributes refer to the corresponding IDN descriptions found in the SERCOS Interface standard. Since these attributes are automatically configured to appropriate values based on the current Drive Polarity Bits settings, the user need not be concerned with manually configuring each of these attributes. Generally, all command bits are set according to the current Command Polarity bit value, and the feedback bits are set according to the current Feedback Polarity bit setting.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
GSV	Position Polarity	INT	Position Polarity Bit Map: 0: pos cmd 1: additive pos cmd 2: pos feedback 1 3: pos feedback 2 4: use position limits 5: use under/over-flow Polarity bits: bit = 0 – non-inverted bit = 1 – inverted
GSV	Velocity Polarity	INT	Velocity Polarity Bit Map: 0: vel cmd 1: additive vel cmd 2: vel feedback Polarity bits: bit = 0 – non-inverted bit = 1 – inverted
GSV	Torque Polarity	INT	Torque Polarity Bit Map: 0: torque cmd 1: additive torque cmd 2: torque feedback Polarity bits: bit = 0 – non-inverted bit = 1 – inverted

**Axis Info Select** Axis Info Select attributes are used to enable periodic data updates for selected drive status attributes. This method of accessing drive status data is designed to reduce the flow of unnecessary data for the SERCOS module. By selecting the drive status attribute of interest from the enumerated list, this attribute's value is transmitted along with the actual position data to the Logix processor. Thus, the drive status data update time is precisely the coarse update period. Note, if a GSV is done to one of these drive status attributes without the having selected this attribute via the Axis Info Select attribute, the attribute value is static and does not reflect the true value in the drive.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Axis Info Select 1 Axis Info Select 2	DINT	0 = None (default) 1 = Position Command 2 = Position Feedback 3 = Aux Position Feedback 4 = Position Error 5 = Position Int. Error 6 = Velocity Command 7 = Velocity Feedback 8 = Velocity Error 9 = Velocity Int. Error 10 = Accel. Command 11 = Accel. Feedback 12 = (reserved) 13 = Marker Distance 14 = Torque Command 15 = Torque Feedback 16 = Pos Dynamic Torque Limit 17 = Neg Dynamic Torque Limit 18 = Motor Capacity 19 = Drive Capacity 20 = Power Capacity 21 = Bus Regulator Capacity 22 = Motor Electrical Angle 23 = Torque Limit Source 24 = DC Bus Voltage

## Motor and Feedback Configuration

This section covers the various drive attributes that provide motor and feedback device configuration information.

**Motor ID** The Motor ID attribute contains the enumeration of the specific A-B motor catalog number associated with the axis. If the Motor ID does not match that of the actual motor, an error is generated during the drive configuration process.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motor ID	INT	

**Motor Data** The Motor Data attribute is a structure with a length element and an array of bytes that contains important motor configuration information needed by the drive to operate the motor. The length element represents the number of valid data elements in the data array. The meaning of data within the data array is understood only by the drive. The block of data stored in the Motor Data attribute is derived at configuration time from an RSLogix 5000 motor database file.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motor Data	Struct { INT; SINT [256]}	(Struct {length; data[ ]})

**Feedback Type** The Motor and Aux Feedback Type attributes are used to provide the A-B drive with specific device configuration information for the motor mounted or auxiliary feedback device.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motor Feedback Type Aux Feedback Type	INT	

**Feedback Resolution** The Motor and Aux Feedback Resolution attributes are used to provide the A-B drive with the resolution of the associated feedback device in cycles. These parameters also provide the Logix controller and drive with critical information to compute conversion constants used to convert drive units to feedback counts.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motor Feedback Resolution Aux Feedback Resolution	DINT	Cycles per Rev (rotary device) or Grid Constant (linear device)

**Aux Feedback Ratio** The Aux Feedback Ratio attribute represents the quantitative relationship between auxiliary feedback device and the motor. For a rotary auxiliary feedback device, this attributes value should be the turns ratio between the auxiliary feedback device and the motor shaft. The Aux Feedback Ratio attribute is used in range limit and default value calculations during configuration based on the selected motor's specifications. The value is also used by the drive when running the dual feedback servo loop configuration.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Aux Feedback Ratio	FLOAT	Aux Revs per Motor Rev

**Advanced Feedback Attributes** The Advanced Feedback Attributes map directly to SERCOS IDNs. Thus, for a detailed description of these attributes refer to the corresponding IDN descriptions found in the SERCOS Interface standard. Since these attributes are automatically configured to appropriate values based on the current Drive Polarity Bits settings, the user need not be concerned with manually configuring each of these attributes. Generally, all command bits are set according to the current Feedback Type value.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Motor Feedback Configuration Aux Feedback Configuration	INT	Bit map: 0: Feedback type 0 – rotational 1 – linear scale 1: Distance coding 0 – no 1 – yes 2: Feedback res. 0 – metric 1 – inches 3: Direction polarity 0 – not inverted 1 – inverted 4: Marker quantity 0 – one pulse 1 – cyclic pulses 5: Distance coding 0 – counting pos. 1 – counting neg. 6: Measuring system 0 – relative (incr.) 1 – absolute 7: Absolute Usage 0 – absolute 1 – relative (incr.) 8-15: <i>Reserved</i>

**Servo Loop Block Diagrams** The following section illustrates the various servo loop configurations that are supported with this object. Which of these servo loop topologies is in effect depends on the current settings of the of the Servo Loop Configuration and External Drive Type attributes.

**Motor Position Servo** The Motor Position Servo configuration provides full position servo control using only the motor mounted feedback device to provide position and velocity feedback. This servo configuration is a good choice in applications where smoothness and stability are more important than positioning accuracy. Positioning accuracy is limited due to the fact that the controller has no way of compensating for non-linearity in the mechanics external to the motor. Note that the motor mounted feedback device also provides motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.

Servo Config = Motor Position Servo

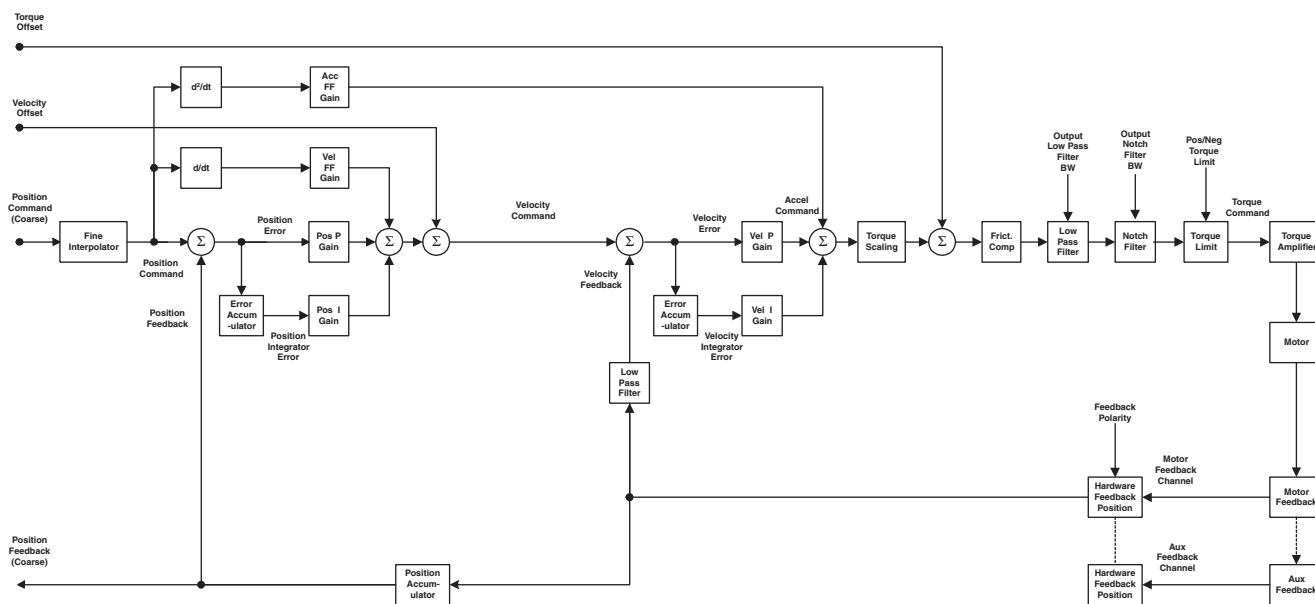


Figure 13.18 Motor Position Servo

### Auxiliary Position Servo

The Auxiliary Position Servo configuration provides full position servo control using an auxiliary (i.e., external to the motor) feedback device to provide position and velocity feedback. This servo configuration is a good choice in applications positioning accuracy is important. The smoothness and stability may be limited, however, due to the mechanical non-linearities external to the motor. Note, that the motor mounted feedback device is still required to provide motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.

Servo Config = Aux Position Servo

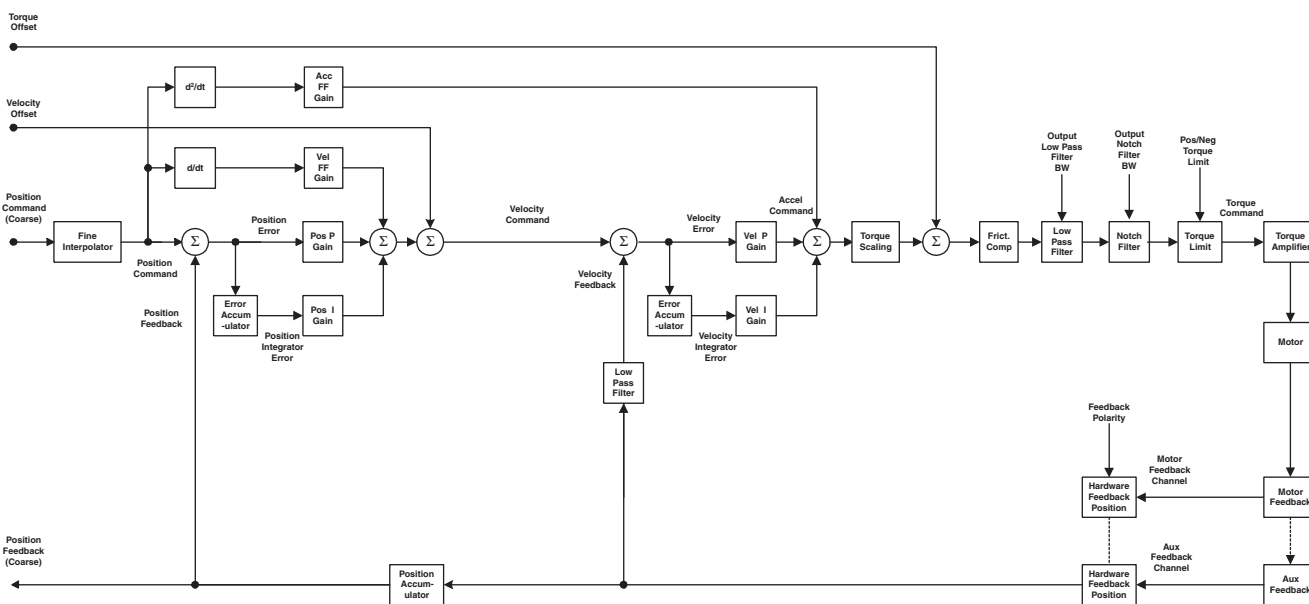


Figure 13.19 Auxiliary Position Servo

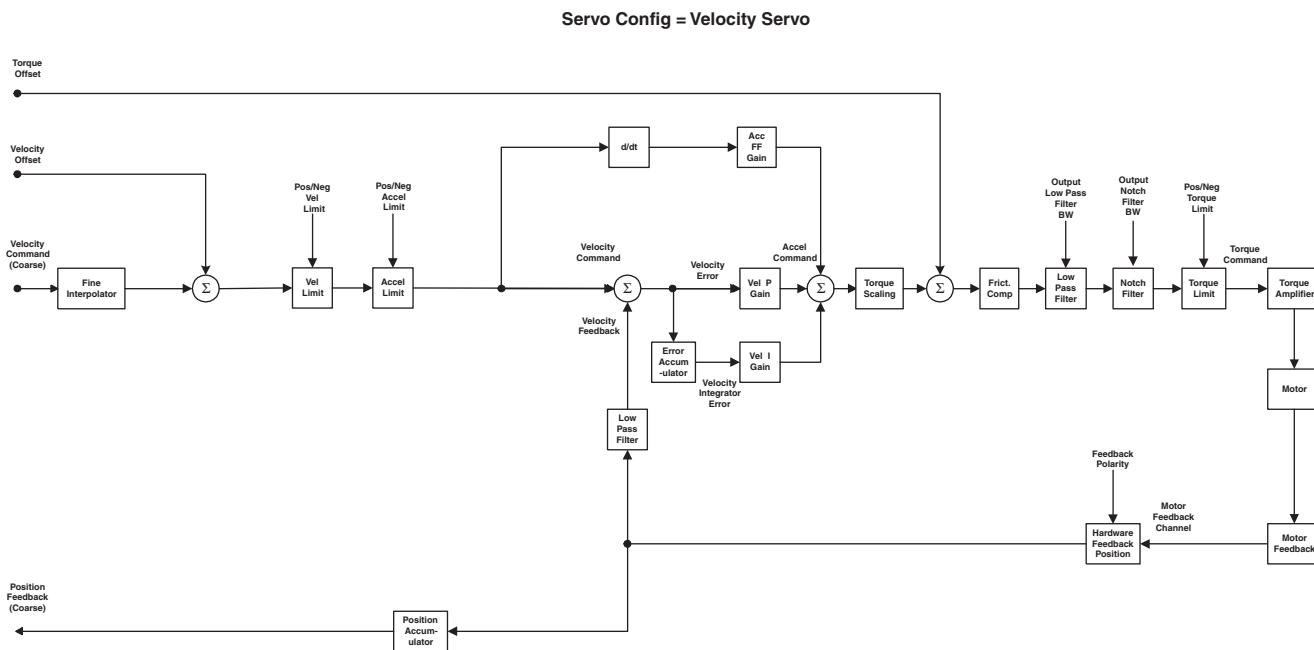






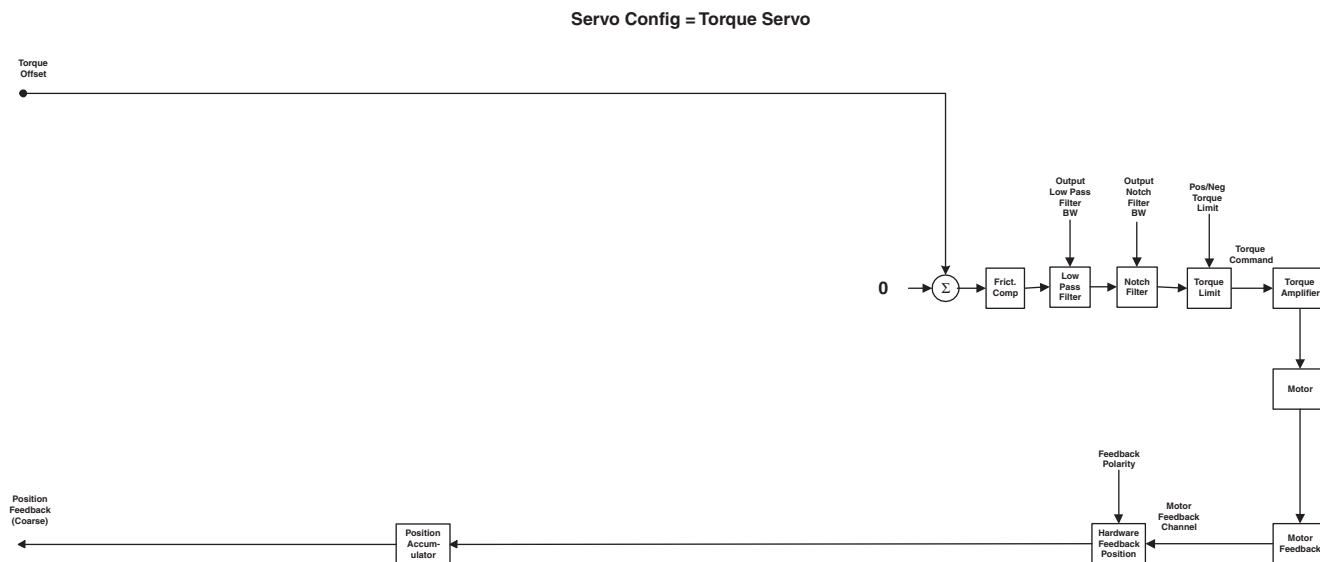


**Velocity Servo** The Velocity Servo configuration provides velocity servo control using the motor mounted feedback device. Synchronous input data to the servo loop includes Velocity Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Velocity Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.



**Figure 13.23 Velocity Servo**

**Torque Servo** The Torque Servo configuration provides torque servo control using only the motor mounted feedback device for commutation. Synchronous input data to the servo loop includes only the Torque Offset. This value is updated at the coarse update rate of the associated motion group. The Torque Offset value is derived from the current value of the corresponding attribute. This offset attribute may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom “outer” control loop capability.



**Figure 13.24 Torque Servo**

**Drive Gains** Rockwell Automation servo drives use Nested Digital Servo Control Loop such as shown in the block diagrams above, consisting typically of a position loop with proportional, integral and feed-forward gains around a digitally synthesized inner velocity loop, again with proportional and integral gains for each axis. These gains provide software control over the servo dynamics, and allow the servo system to be completely stabilized. Unlike analog servo controllers, these digitally set gains do not drift. Furthermore, once these gains are set for a particular system, another SERCOS module programmed with these gain values will operate identically to the original one.

**Position Proportional Gain** The Position Error is multiplied by the Position Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Increasing this gain value increases the bandwidth of the position servo loop and results in greater “static stiffness” of the axis which is a measure of the corrective force that is applied to an axis for a given 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. .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Proportional Gain	REAL	1/Sec

A well-tuned system moves and stops quickly or "smartly" and exhibits little or no "ringing" during constant velocity or when the axis stops. If the response time is poor, or the motion "sloppy" or slow, the proportional gain may need to be increased. If excessive ringing or overshoot is observed when the motor stops, the proportional gain may need to be decreased.

While the Pos P Gain is typically established by the automatic servo tuning procedure, the Pos P gain 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. Refer to Torque Scaling attribute description for an explanation of how the Torque Scaling factor can be calculated. Once this is done the Pos P Gain can be computed based on either the desired loop gain or the desired bandwidth of the position servo system.

### Loop Gain Method

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.

$$\text{Pos P Gain} = 16.667 * \text{Desired Loop Gain (IPM/mil)}$$

According to an old machine tool rule of thumb, a loop gain of 1 IPM/mil (Pos P gain =  $16.7 \text{ Sec}^{-1}$ ) provides stable positioning for virtually any axis. In general, however, modern position servo systems typically run much tighter than this. The typical value for the Position Proportional Gain is  $\sim 100 \text{ Sec}^{-1}$ .

### Bandwidth Method

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

$$\text{Pos P Gain} = \text{Bandwidth (Hertz)} / 6.28$$

In general, modern position servo systems typically run with a unit gain bandwidth of ~16 Hertz. The typical value for the Position Proportional Gain is  $\sim 100 \text{ Sec}^{-1}$ .

### Maximum Bandwidth

There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and torque loops of the system and the desired damping of the system,  $Z$ . These limitations may be expressed as follows:

$$\text{Bandwidth (Pos)} = 0.25 * 1/Z^2 * \text{Bandwidth (Vel)} = 0.25 * 1/Z^2 * \text{Bandwidth (Torque)}$$

For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor,  $Z$ , is 0.8, the velocity bandwidth is approximately 40 Hz and the position bandwidth is 16 Hz. Based on these numbers the corresponding proportional gains for the loops can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.

**Position Integral Gain** Position Integral Gain, or Pos I Gain, improves the steady-state positioning performance of the system. By using Position Integral Gain, it is possible to achieve accurate axis positioning despite the presence of such disturbances as static friction or gravity. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability. .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Integral Gain	REAL	1/mSec-Sec

Every servo update the current Position Error is accumulated in a variable called the Position Integral Error. This value is multiplied by the Position Integral Gain to produce a component to the Velocity Command that attempts to correct for the position error. The characteristic of Pos I Gain correction, however, is that any non-zero Position Error will accumulate in time to generate enough force to make the correction. This attribute of Pos I Gain makes it invaluable in applications where positioning accuracy or tracking accuracy is critical. The higher the Pos I Gain value the faster the axis is driven to the zero Position Error condition. Unfortunately, Pos I Gain control is intrinsically unstable. Too much Pos I Gain will result in axis oscillation and servo instability.

If the axis is configured for an external velocity loop servo drive, the Pos I Gain should be zero—most analog velocity loop servo amplifiers have integral gain of their own and will not tolerate *any* amount of Pos I Gain in the position loop without producing severe oscillations. If Pos I Gain is necessary for the application, the velocity integrator in the drive must be disabled.

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. Another common case is when performing certain motion. When the Integrator Hold Enable attribute is set, 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, 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. Refer to Torque Scaling attribute description for an explanation of how the Torque Scaling factor can be calculated. 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:

$$\text{Pos I Gain} = 0.25 * 0.001 \text{ Sec/mSec} * (\text{Pos P Gain})^2$$

Assuming a Pos P Gain value of  $100 \text{ Sec}^{-1}$  this results in a Pos I Gain value of  $2.5 \sim 0.1 \text{ mSec}^{-1} \cdot \text{Sec}^{-1}$

### Velocity Feedforward Gain

Servo Drives require non-zero command input to generate steady-state axis acceleration or velocity. To provide the non-zero output from the drive to the motor, a non-zero position or velocity error needs to be present. We call this dynamic error while moving “following error”. The non-zero following error condition is a situation you are trying to avoid. Ideally you want zero following error -- all the time. This could be achieved through use of the position integral gain controls as described above, but typically the response time of the integrator action is too slow to be effective. An alternative approach that has superior dynamic response is to use Velocity and Acceleration Feedforward.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Feedforward Gain	REAL	%

The Velocity Feedforward Gain attribute is used to provide the Velocity Command output necessary to generate the commanded velocity. It does this by scaling the current command velocity (derivative of command position) by the Velocity Feedforward Gain and adding it as an offset to the Velocity Command generated by the position loop control elements. With this done, the position loop control elements do not need to generate much of a contribution to the Velocity Command, hence the Position Error value is significantly reduced. 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. One thing that may force a smaller Velocity Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot. If necessary, the Velocity Feedforward Gain may be "tweaked" from the 100% value by running a simple user program that jogs the axis in the positive direction and monitor the Position Error of the axis during the jog. Increase the Velocity Feedforward Gain until the Position Error at constant speed is as small as possible, but still positive. If the Position Error at constant speed is negative, the actual position of the axis is *ahead* of the command position. If this occurs, decrease the Velocity Feedforward Gain such that the Position Error is again positive. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

**Acceleration Feedforward Gain** The Acceleration Feedforward Gain attribute is used to provide the Torque Command output necessary to generate the commanded acceleration. It does this by scaling the current Command Acceleration by the Acceleration Feedforward Gain and adding 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 control effort, hence the Position and/or Velocity Error values are significantly reduced. 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.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Acceleration Feedforward Gain	REAL	%

The Acceleration Feedforward Gain attribute is used to provide the Torque Command output necessary to generate the commanded acceleration. It does this by scaling the current Command Acceleration by the Acceleration Feedforward Gain and adding 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 control effort, hence the Position and/or Velocity Error values are significantly reduced. 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 torque loops with non-infinite loop gain and other application considerations. One thing that may force a smaller Acceleration Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot.

If necessary, the Acceleration Feedforward Gain may be "tweaked" from the 100% value by running a simple user program that jogs the axis in the positive direction and monitors the Position Error of the axis during the jog. Usually Acceleration Feedforward is used in tandem with Velocity Feedforward to achieve near zero following error during the entire motion profile. To fine-tune the Acceleration Feedforward Gain, the Velocity Feedforward Gain must first be optimized using the procedure described above. While capturing the peak Position Error during the acceleration phase of the jog profile, increase the Acceleration Feedforward Gain until the peak Position Error is as small as possible, but still positive. If the peak Position Error during the acceleration ramp is negative, the actual position of the axis is *ahead* of the command position during the acceleration ramp. If this occurs, decrease the Acceleration Feedforward Gain such that the Position Error is again positive. To be thorough the same procedure should be done for the deceleration ramp to verify that the peak Position Error during deceleration is acceptable. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

### Velocity Proportional Gain

The standard RA SERCOS drive's digital velocity loop provides damping without the requirement for an analog tachometer. The Velocity Error is multiplied by the Velocity Proportional Gain to produce a Torque Command that ultimately attempts to correct for the velocity error, creating the damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. The velocity loop also allows higher effective position loop gain values to be used, however, too much Velocity Proportional Gain leads to high frequency instability and resonance effects. Note that units for Velocity Proportional Gain are identical to that of the Position Proportional Gain making it easy to perform classic calculations to determine damping and bandwidth.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Proportional Gain	REAL	1/Sec

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

$$\text{Vel P Gain} = \text{Bandwidth (Hertz)} / 6.28$$

In general, modern velocity servo systems typically run with a unit gain bandwidth of ~40 Hertz. The typical value for the Velocity Proportional Gain is ~250 Sec<sup>-1</sup>.

### Maximum Bandwidth

There are limitations to the maximum bandwidth that can be achieved for the velocity loop based on the dynamics of the inner torque loop of the system and the desired damping of the system, Z. These limitations may be expressed as follows:

$$\text{Bandwidth (Velocity)} = 0.25 * 1/Z^2 * \text{Bandwidth (Torque)}$$

For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor, Z, is 0.8, the velocity bandwidth is approximately 40 Hz. Based on this number the corresponding gains for the loop can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.

**Velocity Integral Gain** When configured for a torque (current) loop servo drive, every servo update the current Velocity Error is also accumulated in 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 characteristic of Vel I Gain correction, however, is that any non-zero Velocity Error will accumulate in time to generate enough force to make the correction. This attribute of Vel I Gain makes it invaluable in applications where velocity accuracy is critical. 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 will result in axis oscillation and servo instability.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Integral Gain	REAL	1/mSec

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. Another common case is when performing certain motion. When the Integrator Hold Enable attribute is set, 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, 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. Refer to Torque Scaling attribute description for an explanation of how the Torque Scaling factor can be calculated. 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:

$$\text{Vel I Gain} = 0.25 * 0.001 \text{ Sec/mSec} * (\text{Vel P Gain})^2$$

Assuming a Vel P Gain value of  $0.25 \text{ Sec}^{-1}$  this results in a Vel I Gain value of  $\sim 15.6 \text{ mSec}^{-1} \cdot \text{Sec}^{-1}$

**Output LP Filter Bandwidth** The Output LP (Low Pass) Filter Bandwidth attribute controls the bandwidth of the drives low-pass digital output filter. The programmable low-pass output filter is bypassed if the configured Output LP Filter Bandwidth for this filter is set to zero (the default). This output filter can be used to filter out, or reduce, high frequency variation of the drive output to the motor. The lower the Output LP Filter Bandwidth, the greater the attenuation of these high frequency components of the output signal. Unfortunately, since the low-pass filter adds lag to the servo loop which pushes the system towards instability, decreasing the Output LP Filter Bandwidth usually requires lowering the Position or Velocity Proportional Gain of the system to maintain stability.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Output LP Filter Bandwidth	REAL	Hertz

The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

**Output Notch Filter Frequency** The Output Notch Filter Frequency attribute controls the center frequency of the drive's digital notch filter. Currently implemented as a 2<sup>nd</sup> order digital filter with a fixed Q, the Notch Filter provides approximately 40DB of output attenuation at the Notch Filter Frequency. The programmable notch filter is bypassed if the configured Output Notch Filter Frequency for this filter is set to zero (the default). This output notch filter is particularly useful in attenuating mechanical resonance phenomena.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Output Notch Filter Frequency	REAL	Hertz

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.

**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. In fact, the Torque Scaling value, when properly established, represents the inertia of the system and is related to the Tune Inertia value by a factor of the Conversion Constant. The Torque Scaling value is typically established by the drive’s automatic tuning procedure but the value can be manually calculated, if necessary, using the following guidelines.

$$\text{Torque Scaling} = 100 \% \text{ Rated Torque} / (\text{Acceleration @ 100\% Rated Torque})$$

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Torque Scaling	REAL	%Rated/ Position Units Per Second <sup>2</sup>

For example, if this axis is using position units of motor revolutions (revs), and that with 100% rated torque applied to the motor, the motor accelerates at a rate of 3000 Revs/Sec<sup>2</sup>, the Torque Scaling attribute value would be calculated as shown below.

$$\text{Torque Scaling} = 100 \% \text{ Rated} / (3000 \text{ RPS}^2) = 0.033 \% \text{ Rated/ Revs Per Second}^2$$

**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.

**Integrator Hold Enable** When the Integrator Hold Enable attribute value is configured TRUE, the servo loop temporarily disables any enabled integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion. When the Integrator Hold Enable attribute value is FALSE, all active integrators are always enabled. .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Integrator Hold Enable	SINT	0 = disabled 1 = enabled

**Advanced Drive Gain Attributes** The above advanced attributes map directly to SERCOS IDNs. Thus, for a detailed description of these attributes refer to the corresponding IDN descriptions found in the SERCOS Interface standard or the AB SERCOS Drive PISD. Since these attributes are automatically configured to reasonable default values, manual configuration by the user is not required unless motivated by a specific application requirement.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Droop	REAL	Position Units / sec.

## Drive Limits

This section covers the various drive attributes that either apply limits to various servo loop real-time parameters, such as position and output voltage, or are used in limit checks of servo loop parameters like position error.

**Maximum Positive/Negative Travel** The Axis Object provides configurable software travel limits via the Maximum Positive and Negative Travel attributes. If the axis is configured for software overtravel limit checking by setting the Soft Overtravel Bit in the Drive Configuration Bit word, and the axis passes outside these maximum travel limits, a Software Overtravel Fault is issued.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Maximum Positive Travel	REAL	Position Units
SSV/GSV	Maximum Negative Travel	REAL	Position Units

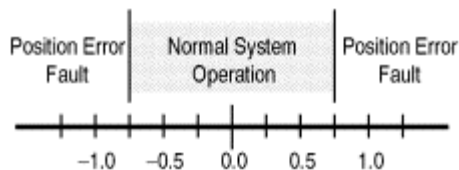
When software overtravel checking is enabled, appropriate values for the maximum travel in both the Maximum Positive and Maximum Negative Travel attributes need to be established with Maximum Positive Travel always greater than Maximum Negative Travel. Both of these values are specified in the configured Position Units of the axis.

**Note:** The software travel limits are not enabled until the selected homing sequence is completed.

**Position Error Tolerance** The Position Error Tolerance parameter specifies how much position error the drive tolerates before issuing a Position Error Fault. Like the position lock tolerance, the position error tolerance is interpreted as a  $\pm$  quantity.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Error Tolerance	REAL	Position Units

For example, specifying a position error tolerance of 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 below:



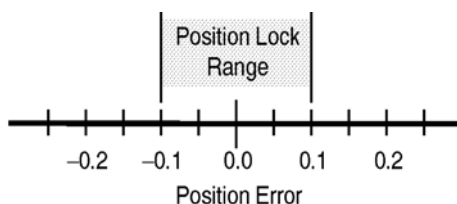
**Figure 13.25 Position Error**

The self tuning routine sets the position error tolerance to twice the following error at maximum speed based on the measured response of the axis. 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** The Position Lock Tolerance attribute value specifies how much position error the SERCOS module tolerates when giving a true Position Locked Status indication. When used in conjunction with the Position Locked Status bit, it is a useful parameter to control positioning accuracy. The Position Lock Tolerance value should be set, in Position Units, to the desired positioning accuracy of the axis.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Lock Tolerance	REAL	Position Units

The position lock tolerance value is interpreted as a  $\pm$  quantity. For example, if your position units are Inches, specifying a position lock tolerance of 0.01 provides a minimum positioning accuracy of  $\pm 0.01$  inches as shown below.



**Figure 13.26 Position Lock Range**

**Advanced Drive Limits** The advanced attributes map directly to SERCOS IDNs. Thus, for a detailed description of these attributes refer to the corresponding IDN descriptions found in the SERCOS Interface standard. Since these attributes are automatically configured to reasonable default values, manual configuration by the user is not required unless motivated by a specific application requirement.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Velocity Limit (Bipolar)	REAL	Position Units / sec.
SSV/GSV	Acceleration Limit (Bipolar)	REAL	Position Units / sec. <sup>2</sup>
SSV/GSV	Torque Limit (Bipolar)	REAL	%Rated
SSV/GSV	Velocity Limit (Positive)	REAL	Position Units / sec.
SSV/GSV	Velocity Limit (Negative)	REAL	Position Units / sec.
SSV/GSV	Velocity Threshold	REAL	Position Units / sec.
SSV/GSV	Velocity Window	REAL	Position Units / sec.
SSV/GSV	Velocity Standstill Window	REAL	Position Units / sec.
SSV/GSV	Acceleration Limit (Pos.)	REAL	Position Units / sec. <sup>2</sup>
SSV/GSV	Acceleration Limit (Neg.)	REAL	Position Units / sec. <sup>2</sup>
SSV/GSV	Torque Limit (Positive)	REAL	%Rated
SSV/GSV	Torque Limit (Negative)	REAL	%Rated
SSV/GSV	Torque Threshold	REAL	%Rated

## Drive Offsets

This section covers the various drive attributes that provide offsets to real-time servo drive loop operation.

**Friction Compensation** It is not unusual for an axis to have enough static friction, so called “sticktion”, that even with a significant position error, refuses to budge. Of course, integral gain can be used to generate enough output to the drive to correct the error, but this approach may not be responsive enough for the application. An alternative is to use Friction Compensation to break sticktion in the presence of a non-zero position error. This is done by adding, or subtracting, a fixed output level, called Friction Compensation, to the Servo Output value based on its current sign.

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Friction Compensation	REAL	% Rated

The Friction Compensation value should be just under the value that would break the sticktion. A larger value will result in the Axis to “dither”, a phenomena describing a rapid back and forth motion of the axis centered on the commanded position.



**Velocity Offset** Velocity Offset compensation can be used to correct to provide a dynamic velocity correction to the output of the position servo loop. Since this value is updated synchronously every Coarse Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block programming. .

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Offset	REAL	Position Units per sec.

**Torque Offset** Torque Offset compensation can be used to provide a dynamic torque command correction to the output of the velocity servo loop. Since this value is updated synchronously every Coarse Update Period, the Torque Offset can be tied into custom outer control loop algorithms using Function Block programming.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Torque Offset	REAL	% Rated

## Drive Fault Actions

Each axis can be configured to respond to each of the five types of drive faults in any one of four different ways. This flexibility is important because motion control applications differ widely in their fault action requirements.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Soft Overtravel Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Hard Overtravel Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Position Error Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Feedback Loss Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only

<b>GSV/SSV Access</b>	<b>Attribute Name</b>	<b>Data Type</b>	<b>Values</b>
SSV/GSV	Feedback Noise Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Drive Thermal Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only
SSV/GSV	Motor Thermal Fault Action	SINT	0 = shutdown 1 = disabled drive 2 = stop command 3 = status only

### Shutdown

If a fault action is set to Disable Drive, then when the associated fault occurs, axis servo action is immediately disabled, the servo output is zeroed, and the appropriate drive enable output is deactivated. Furthermore, this fault action opens the OK contact associated with the SERCOS 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 which 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 output is zeroed, and the appropriate drive enable output is deactivated. Shutdown is the most severe action to a fault and it is usually used for faults which could endanger the machine or the operator if power is not removed as quickly as possible.

### Stop Command

If a fault action is set to Stop Command, 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 SERCOS modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults, since it is relatively easy to recover from. Once 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, motion faults must be handled by the application program. In general, this setting should *only* be used in applications where the standard fault actions are not appropriate.

The recommended setting of the fault action configuration parameters—suitable for most applications—are provided as defaults.

**Advanced Stop Action Attributes** The advanced attributes map directly to SERCOS IDNs. For a detailed description of these attributes refer to the corresponding IDN descriptions found in the SERCOS Interface standard. Since these attributes are automatically configured based on the current Drive Configuration, the user need not be concerned with manually configuring each of these attributes.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Stopping Torque	REAL	% Rated
SSV/GSV	Stopping Time Limit	REAL	Sec

**Brake Engage Delay** The Brake Engage Delay attribute controls the amount of time that the drive continues to apply torque to the motor after the brake output is changed to engage the brake. This gives time for the brake to engage.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Brake Engage Delay Time	REAL	Sec

Below is the sequence of events associated with engaging the brake:

1. Disable axis is initiated (via MSF or drive disable fault action)
2. Decel to zero speed using configured Stopping Torque.
3. Zero speed or Stopping Time Limit is reached.
4. Turn brake output off to engage the brake.
5. Wait Brake Engage Delay Time.
6. Disable the drive power structure.

If the axis is shutdown through either a fault action or motion instruction the drive power structure is disabled immediately and the brake is engaged immediately.

**Brake Release Delay** The Brake Release Delay attribute controls the amount of time that the drive holds of tracking command reference changes after the brake output is changed to release the brake. This gives time for the brake to release.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Brake Release Delay Time	REAL	Sec

Below is the sequence of events associated with engaging the brake.

1. Enable axis is initiated (via MSO or MAH).
2. Drive power structure enabled.
3. Turn brake output on to release the brake.
4. Wait Brake Release Delay Time.
5. Track command reference. (ServoActionStatus bit sets.)

The drive shall not release the brake unless there is holding torque.

## Drive Power Attributes

Two key drive configuration attributes are used to verify that the actual drive has the proper power supply and bus regulator hardware.

**Power Supply ID** The Power Supply ID attribute contains the enumeration of the specific A-B Power Supply or System Module catalog numbers associated with the axis. If the Power Supply ID does not match that of the actual supply hardware, an error is generated during the drive configuration process.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Power Supply ID	INT	

**Bus Regulator ID** The Bus Regulator ID attribute contains the enumeration of the specific A-B Bus Regulator or System Shunt catalog numbers associated with the axis. If the Bus Regulator ID does not match that of the actual bus regulator or shunt hardware, an error is generated during the drive configuration process.

GSV/SSV Access	Attribute Name	Data Type	Values
GSV	Bus Regulator ID	INT	

## Commissioning Configuration Attributes

The Axis Object provides sophisticated automatic test tuning instructions, which allow it to determine proper settings for the servo loop attributes for each axis. These include not only the polarities, the gains, and also the maximum acceleration, deceleration, and velocity parameters.

The servo loop parameters need only be tested and tuned once when the motion controller is first integrated into the machine or when the machine is being commissioned at start-up. However, if the load on any axis changes significantly or if the motor or drive amplifier is replaced for any reason, it may be necessary to re-test and re-tune the servo loop parameters.

The Commissioning Configuration Attributes are used to control the axis test and tuning processes that are initiated by the MRHD and MRAT instructions. These values should be established before the MRHD or MRAT instructions are executed.

**Test Increment** The Test Increment attribute is used in conjunction with the MRHD (Motion Run Hookup Diagnostic) instruction to determine the amount of motion that is necessary to satisfy the MRHD initiated test process. This value is typically set to approximately a quarter of a revolution of the motor.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Test Increment	REAL	Position Units

**Tuning Travel Limit** The Tuning Travel Limit attribute is used in conjunction with the MRAT (Motion Run Axis Tuning) instruction to limit the excursion of the axis during the test. If, while performing the tuning motion profile, the SERCOS module determines that the axis will not be able to complete the tuning process before exceeding the Tuning Travel Limit, the SERCOS module will terminate the tuning profile and report that the Tuning Travel Limit was exceeded via the Tune Status attribute. This does not mean that the Tuning Travel Limit was actually exceeded, but that had the tuning process gone to completion that the limit would have been exceeded.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Travel Limit	REAL	Position Units

**Tuning Speed** The Tuning Speed attribute value determines the maximum speed of the MRAT (Motion Run Axis Tune) initiated tuning motion profile. This attribute should be set to the desired maximum operating speed of the motor prior to running the MRAT instruction. The reason for doing this is that the tuning procedure will measure maximum acceleration and deceleration rates based on ramps to and from the Tuning Speed. Thus, the accuracy of the measured acceleration and deceleration capability is reduced by tuning at a speed other than the desired operating speed of the system.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Speed	REAL	Position Units / Sec

**Tuning Torque** The Tuning Torque attribute value determines the maximum torque of the MRAT (Motion Run Axis Tune) initiated tuning motion profile. This attribute should be set to the desired maximum safe torque level prior to running the MRAT instruction. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system. 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. Note that the extrapolation error increases as the Tuning Torque value decreases.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Torque	REAL	%

**Damping Factor** The Damping Factor attribute value is used in calculating the maximum Position Servo Bandwidth (see below) during execution of the MRAT (Motion Run Axis Tune) instruction. In general the Damping Factor attribute controls the dynamic response of the drive axis. When gains are tuned using a small damping factor (like 0.7), a step response test performed on the axis would demonstrate under-damped behavior with velocity overshoot. A gain set generated using a larger damping factor, like 1.0, would produce a system step response that have no overshoot but have a significantly lower servo bandwidth. The default value for the Damping Factor of 0.8 should work fine for most applications.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Damping Factor	REAL	

**Drive Model Time Constant** The value for the Drive Model Time Constant represents lumped model time constant for the drive's current loop used by the MRAT instruction to calculate the Maximum Velocity and Position Servo Bandwidth values. The Drive Model Time Constant is the sum of the drive's current loop time constant, the feedback sample period, and the time constant associated with the velocity feedback filter. This value is set to a default value when the axis is configured based on the specific drive amplifier and motor feedback selection. Since the bandwidth of the velocity feedback filter is determined by the resolution of the feedback device, the value for the Drive Model Time Constant will be smaller when high resolution feedback devices are selected.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Drive Model Time Constant	REAL	Sec

**Velocity Servo Bandwidth** The value for the Velocity Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the velocity servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Velocity Servo Bandwidth is the better the dynamic performance of the system. A maximum value for the Velocity Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction results in dynamic response in keeping with the current value of the Damping Factor described above. . Alternatively, the responsiveness of the system can be "softened" by reducing the value of the Velocity Servo Bandwidth before executing the MAAT instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Velocity Servo Bandwidth	REAL	Hertz

There are practical limitations to the maximum Velocity Servo Bandwidth for the velocity servo loop based on the drive system and the desired damping factor of the system,  $Z$ . Exceeding these limits could result in an unstable servo operation. These bandwidth limitations may be expressed as follows:

$$\text{Max Velocity Servo Bandwidth (Hz)} = 0.159 * 0.25 * 1/Z^2 * 1/\text{Drive Model Time Constant}$$

The factor of 0.159 represents the  $1/2\text{PI}$  factor required to convert Radians per Second units to Hertz.

**Position Servo Bandwidth** The value for the Position Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the position servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Position Servo Bandwidth the better the dynamic performance of the system. A maximum value for the Position Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction will result in dynamic response in keeping with the current value of the Damping Factor described above. Alternatively, the responsiveness of the system can be “softened” by reducing the value of the Position Servo Bandwidth before executing the MAAT instruction.

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Position Servo Bandwidth	REAL	Hertz

There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and current loops of the servo system and the desired damping of the system,  $Z$ . Exceeding these limits could result in an unstable system. These bandwidth limitations may be expressed as follows:

$$\text{Max Position Bandwidth (Hz)} = 0.25 * 1/Z^2 * \text{Velocity Bandwidth (Hz)}$$

For example, if the maximum bandwidth of the velocity servo loop is 40 Hz and the damping factor,  $Z$ , is 0.8, the maximum the maximum position bandwidth is 16 Hz. Based on these numbers the corresponding proportional gains for the loops can be computed.



## Tuning Configuration Bits

GSV/SSV Access	Attribute Name	Data Type	Values
SSV/GSV	Tuning Configuration Bits	DINT	0: Tuning Direction Reverse 1: Tune Position Error Integrator 2: Tune Velocity Error Integrator 3: Tune Velocity Feedforward 4: Tune Acceleration Feedforward 5: Tune Output Low-Pass Filter 6-31: <i>Reserved</i>

### Tuning Direction Reverse

The Tune Direction Reverse bit attribute determines the direction of the tuning motion profile initiated by the MRAT (Motion Run Axis Tune) instruction. If this bit is set (true), motion is initiated in the reverse (or negative) direction.

### Tune Position Error Integrator

The Tune Position Error Integrator bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction will calculate a value for the Position Integral Gain. If this bit is clear (false) the value for the Position Integral Gain will be set to zero.

### Tune Velocity Error Integrator

The Tune Velocity Error Integrator bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction will calculate a value for the Velocity Integral Gain. If this bit is clear (false) the value for the Velocity Integral Gain will be set to zero.

### Tune Velocity Feedforward

The Tune Velocity Feedforward bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction will calculate a value for the Velocity Feedforward Gain. If this bit is clear (false) the value for the Velocity Feedforward Gain will be set to zero.

### Tune Acceleration Feedforward

The Tune Acceleration Feedforward bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction will calculate a value for the Acceleration Feedforward Gain. If this bit is clear (false) the value for the Acceleration Feedforward Gain will be set to zero.

### **Tune Output Low-Pass Filter**

The Tune Output Low-Pass Filter bit attribute determines whether or not the MAAT (Motion Apply Axis Tune) instruction will calculate a value for the Output Filter Bandwidth. If this bit is clear (false) the value for the Output Filter Bandwidth is set to zero which disables the filter.

## Troubleshooting

This chapter describes how to troubleshoot your SoftLogix motion control system using the LED indicators.

### 1784-PM02AE LED Indicator Status LED

There is one bi-color LED to indicate the status of the Servo card. It is visible through the PCI slot at the rear of the computer.

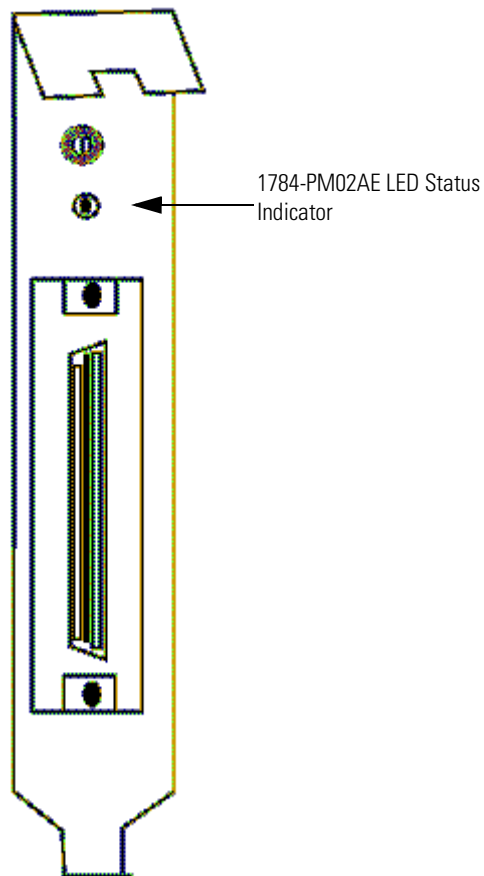


Figure 14.1 1784-PM02AE LED Status Indicator

At start up the LED goes through a sequence of color changes:

- At power up LED is Green.
- When SoftLogix driver starts the LED turns Red.
- At download of a valid program the LED is Green.

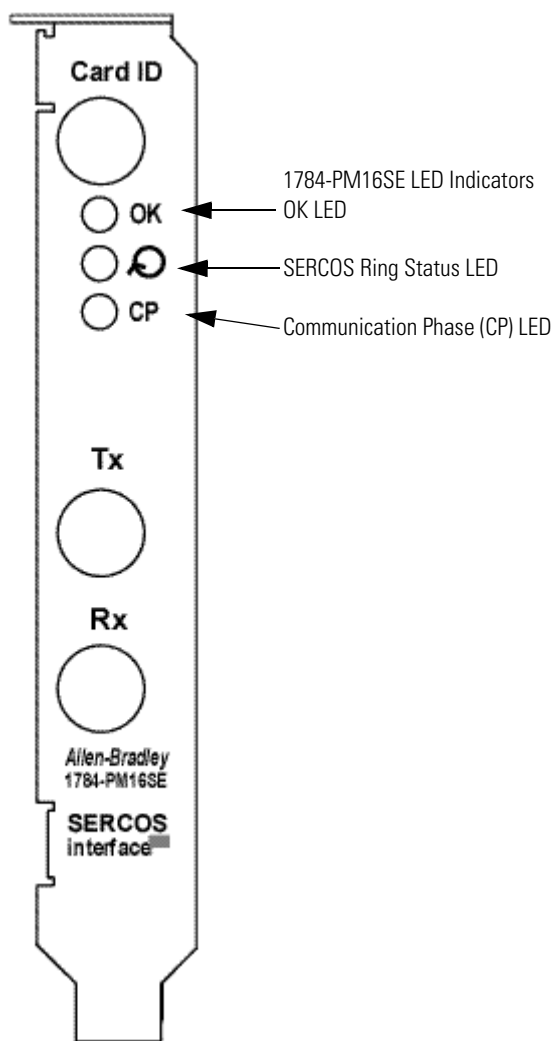
During regular operation the LED is GREEN when the Servo card is functioning normally and RED when a fault situation occurs.

## **SERCOS interface LED Indicators**

### **Checking the LED Indicators**

The card provides three bi-color LED indicators to show the state of the system. The LEDs are located on the mounting bracket of the 1784-PM16SE card. The LED on the top, marked by OK, indicates the present health of the card. Just below the OK LED is the SERCOS Ring Status LED. This is marked with a ring icon and displays the status of the SERCOS network.

Below the SERCOS Ring Status LED is the CP LED. The CP LED shows the communication phase.



**Figure 14.2 LEDs on 1784-PM16SE Mounting Bracket**

### The OK LED

During power up, the card completes an indicator test. The OK indicator is solid red at startup and then turns to flashing green if the card passes all of its self tests.

### Card Status Using the OK Indicator

<b>If the OK LED displays:</b>	<b>Then the card status is:</b>	<b>Take this action:</b>
Off	The card is not operating.	<ul style="list-style-type: none"> <li>• Apply power.</li> <li>• Verify the card is completely inserted into the PCI slot.</li> </ul>
Flashing green light	The card has passed internal diagnostics, but has not established active communications.	<ul style="list-style-type: none"> <li>• None, if you have not configured the card.</li> </ul>
Solid green light	<ul style="list-style-type: none"> <li>• Data is being exchanged.</li> <li>• The card is in the normal operating state.</li> </ul>	None. The card is ready for action.
Flashing red light	<ul style="list-style-type: none"> <li>• A major recoverable failure has occurred.</li> <li>• An NVS update is in progress.</li> </ul>	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.	<ul style="list-style-type: none"> <li>• Reboot the card.</li> <li>• If the solid red persists, replace the card.</li> </ul>

### The SERCOS Ring Status LED

The SERCOS Ring Status LED shows the status of the SERCOS ring.

### SERCOS Ring Status

<b>If SERCOS Ring Status LED displays:</b>	<b>Then the ring status is:</b>	<b>Take this action:</b>
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 card has detected a setup or configuration fault with the ring.	Check your system setup and configuration as follows: <ul style="list-style-type: none"> <li>• Ensure drive and axes addresses are correct.</li> <li>• Remove excess axes from ring.</li> <li>• Make sure application program has selected the proper Ring Cycle Period and Baud Rate.</li> </ul>
Solid red light	The card has detected a hardware or installation fault with the ring.	Check your system hardware and installation as follows: <ul style="list-style-type: none"> <li>• Make sure all cables are properly installed.</li> <li>• Make sure cable is of the correct type and length.</li> <li>• Make sure application program has configured the card's ring transmit level to High when using specified cables.</li> <li>• Make sure the drive's transmit levels are set appropriately.</li> <li>• Inspect cables for degradation.</li> <li>• Inspect drives for any faults and correct them.</li> </ul>
Off	The card has detected no ring data on its receiver or has not successfully completed phase 2.	Check your system and installation as follows: <ul style="list-style-type: none"> <li>• Make sure all cables are properly installed</li> <li>• Inspect cable for degradation and breakage.</li> <li>• Inspect drives for faults.</li> </ul>
Flashing green light	The ring, drive, or axes are not configured but, at least one has been identified.	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.

### The Communication Phase LED

The Communication Phase (CP) LED provides information as to what phase the SERCOS link is in.

**SERCOS Communication Phase Status Using the CP Indicator**

<b>If the CP LED displays:</b>	<b>Then the card status is:</b>
Solid Orange	In Phase -1: Autobaud detection in progress.
Off	In Phase 0: looking for a closed ring.
Flashing Red	In Phase 1: looking for active nodes.
Alternating Red/Green	In Phase 2: configuring nodes for communication.
Flashing Green	In Phase 3: configuring device specific parameters
Solid Green	In Phase 4: configured and active.




## Specifications and Performance

This appendix shows specifications and performance guidelines for the motion modules.



### 1784-PM02AE Motion Card Specifications

Number of Nodes	2 axes maximum
Card location	PCI Bus slot of PC
Card keying	Electronic
Power dissipation	5W (1A @ 5V from PC)
Encoder input	
Type	Incremental AB quadrature with marker
Mode	4X quadrature
Rate	4 MHz maximum
Electrical interface	Optically isolated 5V differential
Voltage range	
On state	3.4V to 5.0V
Off state	0V to 1.8V
Input impedance	531 Ohms differential
Registration inputs	
Type	Optically isolated, current sinking input
24V input voltage	+24V dc nominal
Maximum	26.4V
Minimum on	18.5V
Maximum off	3.5V
Input impedance	
24V input	1.2 kOhms
Response time	1 $\mu$ s
(position latched)	
Drive Faults and Home 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	$\pm$ 10V
Voltage resolution	16 bits
Load	5.6 kOhms resistive minimum
Maximum offset	25 mV
Gain error	$\pm$ 4%

OK and Enable Outputs Type Operating voltage Maximum Operating current	Solid-state isolated relay contacts +24V dc nominal (Class 2 source) 26.4V 75 mA
Environmental Specifications Operational Temperature Storage temperature Relative Humidity	0 to 40° C (32 to 104° F) -40 to 85° C (-40 to 185°F) 30 to 95%
Shock Operating Non-operating	10G 30G
Vibration Operating Non-operating	1G, 5Hz - 500Hz 2G, 5Hz - 500Hz
Agency certification (when product or packaging is marked)	<p>UL508 – Industrial Control Equipment</p>  UL1604 – Class I, Division 2, Groups A, B, C, D Hazardous location <p>CE marked for all applicable directives</p>

## 1784-M16SE Motion Card Specifications

Number of Nodes	16 axes maximum
Card location	32 bit 33MHz primary PCI Bus slot
Card keying	3.3 / 5 VDC universal
Power Dissipation	7.5 W maximum
Backplane Current	5 VDC @ 1.065 A and 3.3 VDC @ 0.050 A <sup>1</sup>
Environmental Specifications Slot Temperature Storage temperature Relative Humidity	0-60° C (32-140° F) -40 to 85°C (-40 to 185°F) 30 to 95% non condensing IEC 68-2-30
Shock Operating Non-operating	30g peak acceleration, 11(±1)ms pulse width 50g peak acceleration, 11(±1)ms pulse width
Vibration	2g @ 10-150Hz 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 (-67 to 185°F) F-SMA standard screw-type connector 2.5 cm

<p>Glass Fiber Optic Specifications</p> <p>Transmission Range</p> <p>Core Diameter</p> <p>Cladding Diameter</p> <p>Cable Attenuation</p> <p>Operating Temperature</p> <p>Connector</p> <p>Bend Radius</p>	<p>1-200 meters</p> <p>200<math>\mu</math>m <math>\pm</math> 4<math>\mu</math>m</p> <p>230<math>\mu</math>m +0/-10<math>\mu</math>m</p> <p>6.0 dB/km @ 820nm</p> <p>-20 to 85° C</p> <p>F-SMA standard screw-type connector</p> <p>2.5 cm</p>
<p>SERCOS</p> <p>SERCOS Class</p> <p>Data Rate</p> <p>Operating cycle</p> <p>1784-PM16SE @ 4 Mbaud</p> <p>1784-PM16SE @ 8 Mbaud</p>	<p>Class B (Position or Velocity)</p> <p>4 Mbits or 8 Mbits per second</p> <p>0.5 ms for up to 2 axes</p> <p>1 ms for up to 4 axes</p> <p>2 ms for up to 8 axes</p> <p>No support for 9 to 16 axes</p> <p>0.5 ms for up to 4 axes</p> <p>1 ms for up to 8 axes</p> <p>2 ms for up to 16 axes</p>
<p>Agency certification (when product or packaging is marked)</p>	<p> UL Recognized Component Industrial Control Equipment for Class I, Division 2, Groups A, B, C, D Hazardous Locations, certified for US and Canada</p> <p> EN50081-2; Industrial Emissions EN61000-6-2; Industrial Immunity</p>



## Wiring Diagrams

This appendix shows the loop interconnect diagrams for common motion configurations.

### About Block Diagrams

The control block diagrams in this section use the following terms for motion attributes.

<b>Diagram term</b>	<b>Motion attribute name (as used in the GSV and SSV instructions)</b>
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 1784-PM02AE Module With a Torque Servo Drive

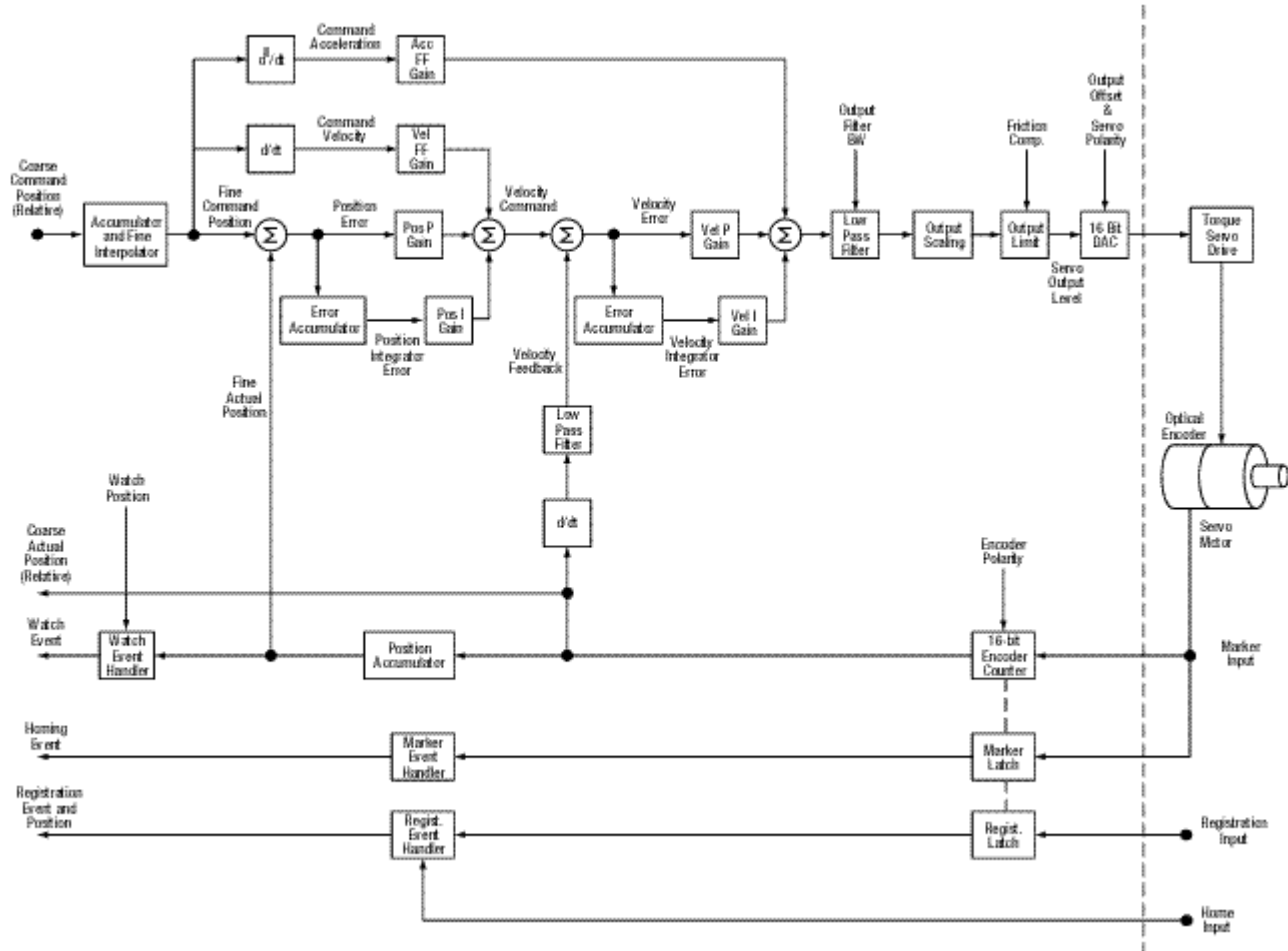


Figure B.1 Torque Servo Drive

## Using a 1784-PM02AE Module With a Velocity Servo Drive

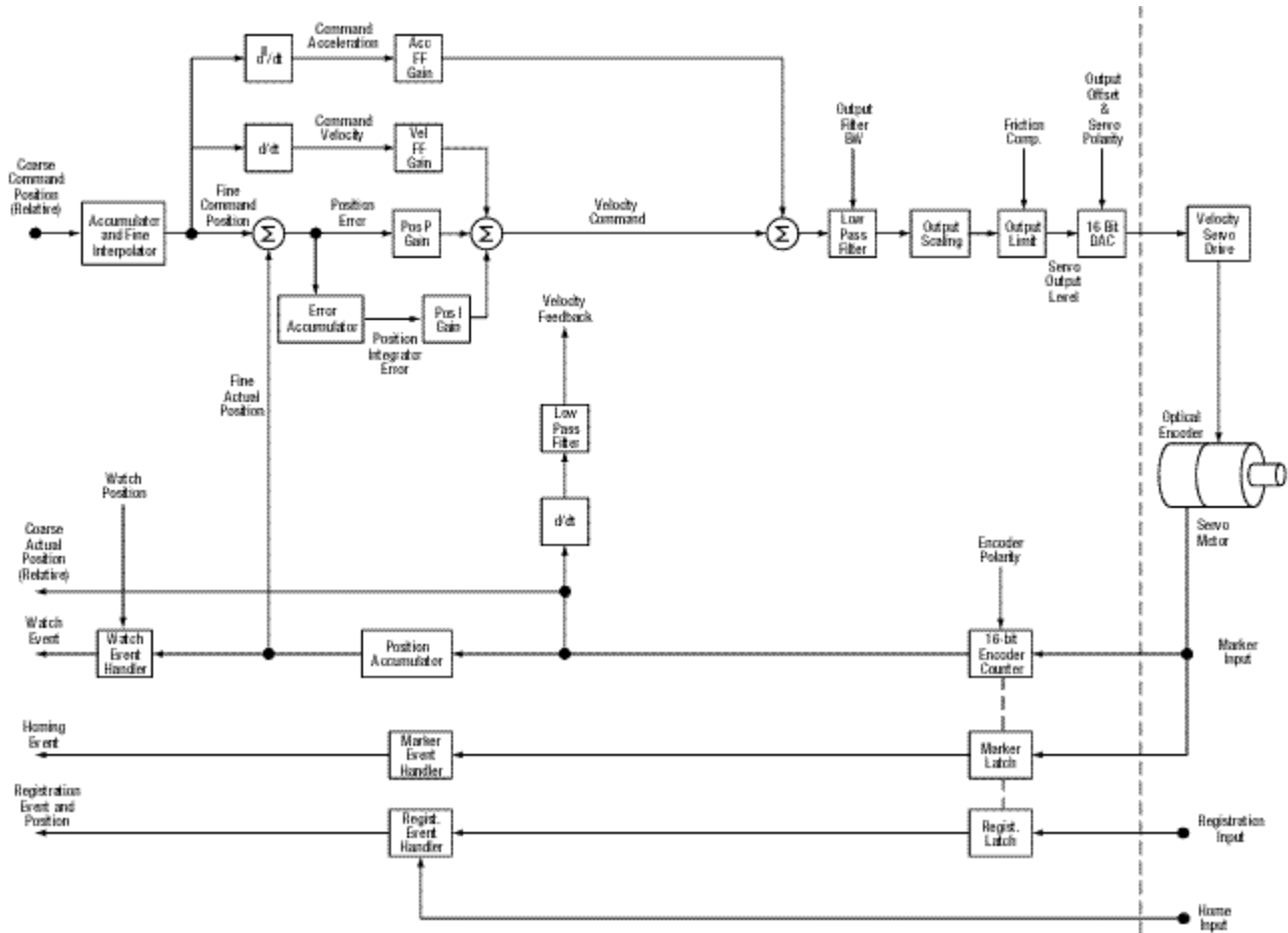


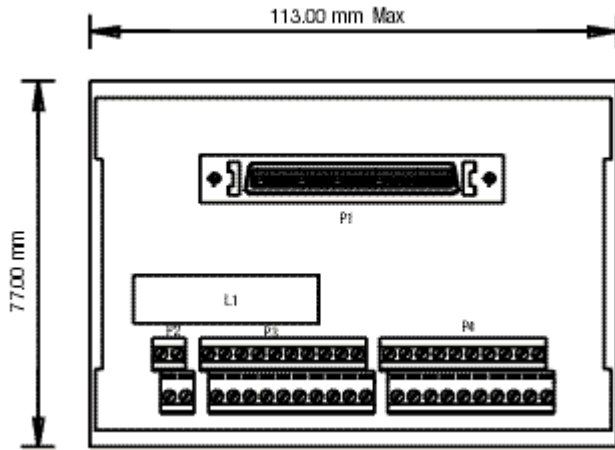
Figure B.2 Velocity Servo Drive

## Wiring Diagrams

### 1784-PM02AE Termination Panel

The termination panel is used in conjunction with the 1784-PM02AE card to facilitate the wiring of drives and encoders for use with the card. Because the card is installed inside the PC computer cabinet it would be difficult to access

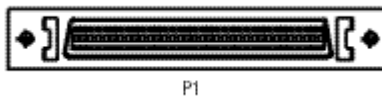
for wiring drives, encoders, etc. A termination panel, mounted separately from the card, allows for easier access to the two axis terminals.



**Figure B.3 Termination Panel for 1784-PM02AE**

**P1**

The connection marked P1 is for the cable from the PCI 2 Axis Servo card. It accepts a straight 68 way Mini D shielded plug with a spring latch. Through this connection the termination panel is connected to the PCI card by a 1 meter or 3 meter premade cable. The cable is shipped with the termination panel and shares its catalog number, 1784 - PM02AE-TP0x where x represents the length of the cable.



**Figure B.4 P1 Connector for 1784-PM02AE-TP0x Cable**

**P2**

The P2 connection is for wiring the encoder power. Those marked B0 and B1 are for the Encoder power and those marked A0 and A1 are the 0 volt connections.

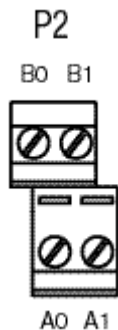




Figure 5 P2 Connector

P2	Function	P2	Function
A0	Encoder 0V	B0	Encoder Power
A1	Encoder 0V	B1	Encoder Power

### P3 and P4

The P3/P4 receptacles are for wiring the axes. P3 is Axis 0 and P4 is Axis 1.

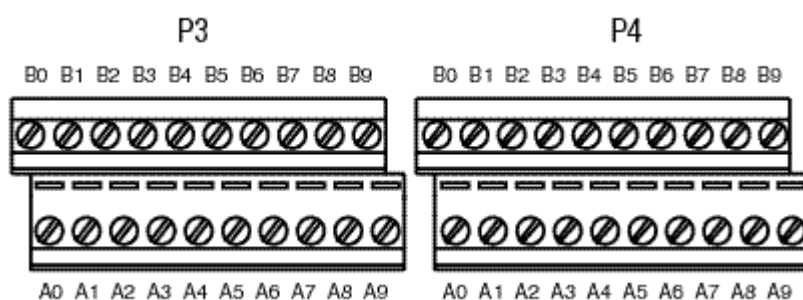
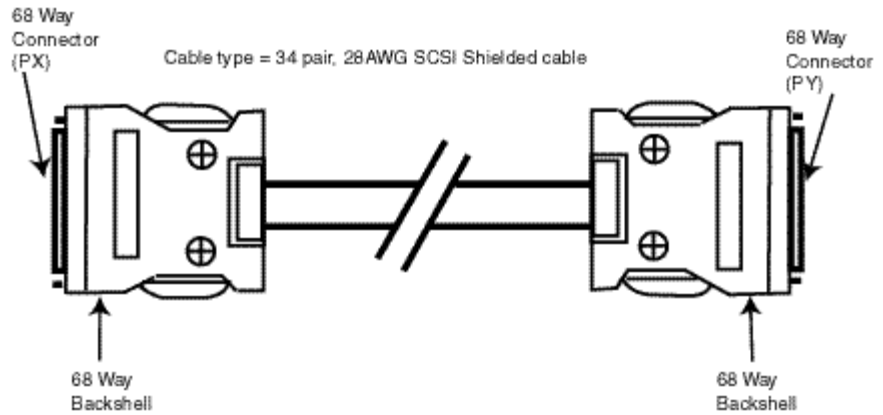


Figure B.6 P3 and P4 Connectors

P3 & P4	Function	P3 & P4	Function
A0	DRVFLT	B0	+CHA
A1	Home Area	B1	-CHA
A2	Reg 1	B2	+CHB
A3	Reg 2	B3	-CHB
A4	OK	B4	+CHZ
A5	IN_COM	B5	-CHZ
A6	Enable+	B6	Shield
A7	Enable-	B7	+Out
A8	Encoder 0V	B8	-Out
A9	Chassis	B9	Encoder Pwr

### 1784-PM02AE-TP0x Cable

The 1784-PM02AE card is connected to the termination panel via a premade 34 pair, 28 AWG SCSI shielded cable. The cable is available in lengths of 1 meter and 3 meters.



**Figure B.7 1784-PM02AE-TP0x Cable**

Catalog Numbers for premade Servo card to termination panel cables.

Allen-Bradley Catalog Number	Length in meters
1784-PM02AE-TP01	1m
1784-PM02AE-TP03	3m

### Pinouts for Cable 1784-PM02AE-TP0x

Pin (PX)	Pin Pair Number	Pin Description	Pin (PY)
1	Pair 1	+CH A Feedback Input Axis 0	1
35		-CH A Feedback Input Axis 0	35
2	Pair 2	+CH B Feedback Input Axis 0	2
36		-CH B Feedback Input Axis 0	36
3	Pair 3	+CH Z Feedback Input Axis 0	3
37		-CH Z Feedback Input Axis 0	37
4	Pair 4	+OUT, Axis 0	4
38		-OUT, Axis 0	38
5	Pair 5	DRVFLT, Axis 0	5
49		HOME, Axis 0	49
6	Pair 6	REG1, Axis 0	6
40		REG2, Axis 0	40
7	Pair 7	+ENABLE, Axis 0	7
41		-ENABLE, Axis 0	41

Pin (PX)	Pin Pair Number	Pin Description	Pin (PY)
8	Pair 8	OK 1	8
42		IN_COM, Axis 0	42
9	Pair 9	IN_COM, Axis 0	9
43		IN_COM, Axis 1	43
10	Pair 10	+CH A Feedback Input Axis 1	10
44		- CH A Feedback Input Axis 1	44
11	Pair 11	+CH B Feedback Input Axis 1	11
45		-CH B Feedback Input Axis 1	45
12	Pair 12	+CH Z Feedback Input Axis 1	12
46		-CH Z Feedback Input Axis 1	46
13	Pair 13	+OUT, Axis 1	13
47		-OUT, Axis 1	47
14	Pair 14	DRVFLT, Axis 1	14
48		HOME, Axis 1	48
15	Pair 15	REG1, Axis 1	15
49		REG2, Axis 1	49
16	Pair 16	+ENABLE, Axis 1	16
50		-ENABLE, Axis 1	50
17	Pair 17	OK 2	17
51		IN_COM, Axis 1	51

### Sync Cable

Up to four (4) 1784-PM02AE cards may be installed in a single computer provided there are enough PCI Bus slots available. A sync cable is required to keep the cards synchronized. This is a ribbon cable with four connectors that fit to the cards.

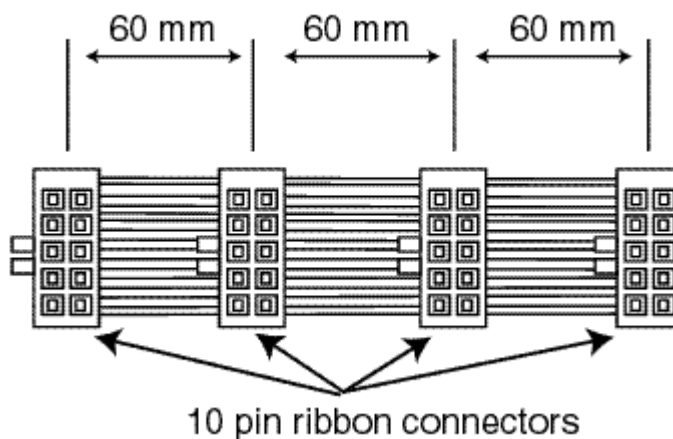
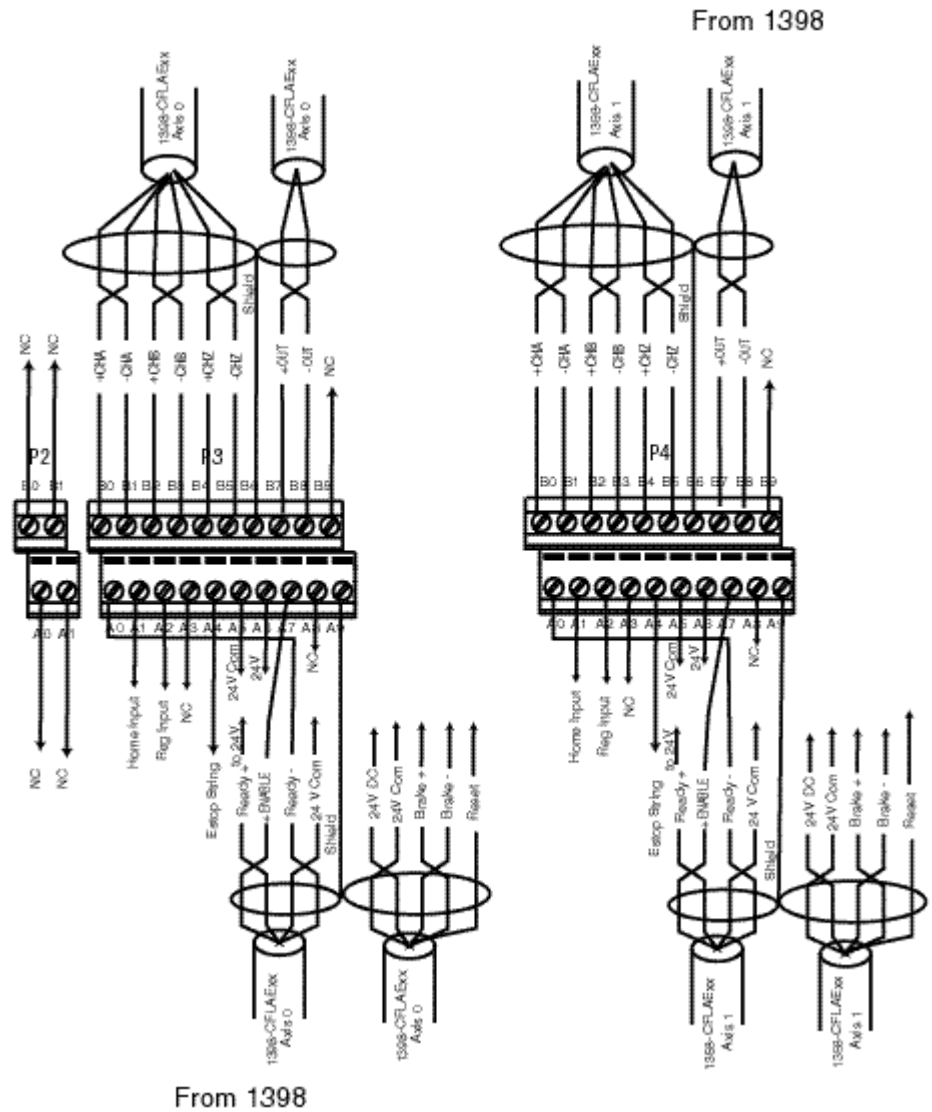


Figure B.8 1784-PMCSY4 Cable

**1398 to Termination Panel Wiring Diagram**

This is a general wiring example of the 1398 to the termination panel.



**Figure B.9 Wiring from a 1398 to the Termination Panel**

This is a general wiring example only. Other configurations are possible.

## Symbols

(Brackets) 181

## Numerics

1394C Drive module

Associated Axes Tab 199

New Axis button 200

Node X0 200

Node X1 200

Node X2 200

Node X3 200

Connection Tab 197

Inhibit Module Checkbox 198

Major Fault on Controller if Connection Fails Checkbox  
198

Module Fault 199

Connection Request Error 199

Electronic Keying Mismatch 199

Module Configuration Invalid 199

Service Request Error 199

Requested Packet Interval 197

General Tab 194

Base Node 195

Description 195

Electronic Keying 196

Compatible Module 196

Disable Keying 196

Exact Match 196

Name 195

Revision 195

Type 195

Vendor 195

Module Info tab 202

(16#xxxx) unknown 203

Configured 203

Internal State Status 203

Major/Minor Fault Status 203

Module Identity 204

Owned 203

Product Name 203

Refresh 204

Reset Module 204

Power Tab 201

Bus Regulator ID 201

1394x-SJTxx Digital Servo Drive Overview 193

1756-M02AE servo module

Adding to a program 45, 87

Specifications 417

1756-M08SE Properties

General Tab

Electronic Keying

Disable Keying 68

Type 67

1756-M16SE

Configuring 61

1784-M16SE Motion Card Specifications 418

1784-PM02AE

Assigning Additional Motion Modules 60

1784-PM02AE Module Properties

Associated Axes Tab 56

Channel 0 57

Channel 1 57

New Axis button 57

Servo Update Period 57

Connection Tab 54

Inhibit Module checkbox 55

Major Fault on Controller if Connection Fails checkbox 55

Module Fault 56

Requested Packet Interval 54

General Tab 52

Description 53

Electronic Keying 53

Name 52

Revision 53

Slot 53

Type 52

Vendor 52

Module Info Tab 57

Configured 59

Identification 58

Internal State Status 59

Major/Minor Fault Status 59

Module Identity 60

Owned 59

Refresh 60

Reset Module 60

1784-PM02AE Motion Card Specifications 417

1784-PM02AE servo module

Adding to a program 45

Additional modules and axes 60

Configuring a motion axis 45

1784-PM16S 61

1784-PM16SE

Adding the module 61

configuring module 61

Motion Module Overview 66

Properties

Connection Tab 70

Inhibit Module checkbox 70

Major Fault On Controller 71

Module Fault 71

Requested Packet Interval 70

General Tab 67

- Description 67
  - Electronic Keying 68
    - Compatible Module 68
    - Exact Match 68
  - Name 67
  - Revision 68
  - Slot 68
  - Status 69
  - Vendor 67
  - Module Info Tab 75
    - Configured 77
    - Identification 76
    - Internal State Status 77
    - Major/Minor Fault Status 77
    - Module Identity 78
    - Owned 77
    - Refresh 78
    - Reset Module 78
  - SERCOS Interface Info Tab 74
    - Fault Type 75
    - Refresh 75
    - Ring Comm. Phase 74
  - SERCOS Interface Tab 72
    - Cycle Time 73
    - Data Rate 73
    - Transmit Power 74
  - 8720MC Drive
    - Configuring 237
    - Properties 240
      - Associated Axes Tab 246
        - Ellipsis (...) 246
        - New Axis 246
        - Node 246
      - Connection Tab 243
        - Inhibit Module 244
        - Major Fault on Controller 245
        - Module Fault 245
          - Connection Request Error 245
          - Electronic Keying Mismatch 245
          - Module Configuration Invalid 245
          - Service Request Error 245
      - General Tab 240
        - Description 241
        - Electronic Keying 242
          - Compatible Modules 242
          - Disable Keying 242
          - Exact Match 242
        - Name 241
        - Node 241
        - Revision 241
        - Status 243
        - Type 241
        - Vendor 241
  - Module Info Tab 248
    - Configured 250
    - Identification 249
    - Internal State Status 249
    - Major/Minor Fault Status 249
    - Module Identity 250
    - Owned 250
    - Reset Module 250
  - Module InfoTab
    - Refresh 251
  - Power Tab 247
    - Bus Regulator ID 247
- ## A
- AAxis Properties
    - Servo Tab (AXIS\_SERVO)
      - Enable Drive Fault Input 103
  - Adding the 1756-PM02AE Module 45
  - Adding the 1784-PM02AE Module
    - New Module 47
      - Clear All 48
      - Description 48
      - Major Revision 48
      - Select All 48
      - Show 48
      - Type 47, 48
  - Application program
    - Developing 10
  - Assigning in an application program
    - Additional modules 60
  - Axis Properties
    - Aux Feedback Tab (AXIS\_SERVO\_DRIVE) 110
      - Cycles 111
      - Feedback Ratio 111
      - Feedback Type 110
      - Interpolation Factor 111
      - Per 111
    - Conversion Tab 112
      - Conversion Constant 113
      - Position Unwind 113
      - Positioning Mode 113
        - Linear 113
        - Rotary 113
    - Drive Tab (AXIS\_SERVO\_DRIVE)
      - Amplifier Catalog Number 105
      - Attribute 1/Attribute 2 107
      - Drive Resolution 107
      - Loop Configuration 106
      - Real Time Axis Information 107
    - Drive/Motor Tab (AXIS\_SERVO\_DRIVE) 105
      - Change Catalog Button 107
      - Catalog Number 108

- Filters 108
  - Family 108
  - Feedback Type 108
  - Voltage 108
- Dynamics Tab 126
  - Manual Adjust 128
  - Maximum Acceleration 127
  - Maximum Deceleration 127
  - Maximum Velocity 127
- Editing 91
- Fault Actions Tab - AXIS\_SERVO
  - Drive Fault 164
- Fault Actions Tab (AXIS\_SERVO) 163
  - Feedback Loss 165
  - Feedback Noise 164
  - Position Error 165
  - Soft Overtravel 165
- Fault Actions Tab (AXIS\_SERVO\_DRIVE) 166
  - Drive Thermal 167
  - Feedback 168
  - Feedback Noise 168
  - Hard Overtravel 168
  - Motor Thermal 168
  - Position Error 168
  - Set Custom Stop Action 169
    - Attributes 169
  - Soft Overtravel 168
- Gains Tab - AXIS\_SERVO\_DRIVE
  - Proportional (Velocity) Gain 132
- Gains Tab (AXIS\_SERVO) 129
  - Acceleration Feedforward 130
  - Integral (Position) Gain 131
  - Integral (Velocity) Gain 132
  - Integrator Hold 133
  - Manual Adjust 133
  - Proportional (Position) Gain 131
  - Proportional (Velocity) Gain 132
  - Velocity Feedforward 130
- Gains Tab (AXIS\_SERVO\_DRIVE) 134
  - Acceleration Feedforward 135
  - Integral (Position) Gain 136
  - Integral (Velocity) Gain 137
  - Integrator Hold 138
  - Manual Tune 139
  - Proportional (Position) Gain 136
  - Proportional (Velocity) Gain 137
  - Set Custom Gains 140
    - Attribute 140
  - Velocity Feedforward 135
- General Tab (AXIS\_SERVO) 94
  - Assigned Motion Group 94
  - Axis Configuration 94
  - Channel 95
  - Ellipsis (...) button 94
  - Module 95
  - Module Type 95
  - New Group button 95
- General Tab (AXIS\_SERVO\_DRIVE) 95
  - Assigned Motion Group 96
  - Axis Configuration 96
  - Ellipsis (...) button 96
  - Module 96
  - Module Type 96
  - New Group button 96
  - Node 96
- General Tab (AXIS\_VIRTUAL) 97
  - Assigned Motion Group 97
  - Ellipsis (...) button 97
  - New Group button 97
- Homing Tab (AXIS\_SERVO, AXIS\_SERVO\_DRIVE) 114
  - Direction 116
  - Homing Configurations 117
  - Limit Switch 116
  - Mode 115
    - Active 115
    - Passive 115
  - Offset 115
  - Position 115
  - Return Speed 117
  - Sequence 116
  - Speed 117
- Homing Tab (AXIS\_SERVO\_DRIVE)
  - Mode
    - Absolute 115
- Homing Tab (AXIS\_VIRTUAL) 118
  - Mode 118
  - Position 118
  - Sequence 119
- Hookup Tab (AXIS\_SERVO) 119
  - Feedback Polarity 120
  - Output Polarity 120
  - Test Feedback 121
  - Test Increment 119
  - Test Marker 120
  - Test Output & Feedback 121
- Hookup Tab (AXIS\_SERVO\_DRIVE) 121
  - Drive Polarity 122
  - Test Feedback 122
  - Test Increment 122
  - Test Marker 122
  - Test Output & Feedback 123
- Limits Tab (AXIS\_SERVO) 148
  - Manual Adjust 151
  - Maximum Negative 149
  - Maximum Positive 149
  - Output Limit 150

- Position Error Tolerance 149
- Position Lock Tolerance 150
- Soft Travel Limits 149
- Limits Tab (AXIS\_SERVO\_DRIVE) 152
  - Hard Travel Limits 153
  - Manual Adjust 154
  - Maximum Negative 153
  - Maximum Positive 153
  - Position Error Tolerance 153
  - Position Lock Tolerance 154
  - Set Custom Limits 155
    - Attributes 155
  - Soft Travel Limits 153
- Motion Planner Tab 98
  - Enable Master Position Filter Checkbox 100
  - Master Delay Compensation Checkbox 99
  - Master Position Filter Bandwidth 100
  - Output Cam Execution Targets 98
  - Program Stop Action 99
    - Fast Disable 99
    - Fast Shutdown 99
    - Fast Stop 99
    - Hard Disable 99
    - Hard Shutdown 99
- Motor/Feedback Tab (AXIS\_SERVO\_DRIVE) 109
  - (Motor) Catalog Number 106
  - (Motor) Cycles 109
  - (Motor) Feedback Type 109
  - (Motor) Interpolation Factor 110
  - Per 110
- Offset Tab (AXIS\_SERVO) 157
  - Friction Compensation 158
  - Manual Adjust 159
  - Output Offset 159
  - Torque Offset 158
  - Velocity Offset 158
- Offset Tab (AXIS\_SERVO\_DRIVE) 160
  - Friction Compensation 161
  - Manual Adjust 162
  - Torque Offset 161
  - Velocity Offset 161
- Output Tab (AXIS\_SERVO\_DRIVE) 145
  - Enable Low-pass Output Filter 146
  - Enable Notch Filter 146
  - Low-pass Output Filter Bandwidth 147
  - Manual Adjust 147
  - Notch Filter 146
  - Torque Scaling 146
- Output Tab (SERVO\_AXIS) 141
  - Enable Low-pass Output Filter 143
  - Low-pass Output Filter Bandwidth 143
  - Manual Adjust 144
  - Torque Scaling 142

- Velocity Scaling 142
- Servo Tab (AXIS\_SERVO) 102
  - Drive Fault Input 103
  - External Drive Configuration 103
  - Loop Configuration 103
  - Real Time Axis Information 103
    - Attribute 1/Attribute 2 103
- Tag Tab 170
  - Data Type 171
  - Description 170
  - Name 170
  - Scope 171
  - Style 171
  - Tag Type 171
- Tune Tab (AXIS\_SERVO, AXIS\_SERVO\_DRIVE) 123
  - Damping Factor 124
  - Direction 124
  - Speed 124
  - Start Tuning 125
  - Torque 124
  - Travel Limit 123
  - Tune 125
- Units Tab 101
  - Average Velocity Timebase 101
  - Position Units 101
- Axis Tag types
  - alias tag 89, 175
  - base tag 89, 175
  - produced tag 89

## C

- Catalog 108
- Change Controller Type 19
  - Select a processor 19
- Commissioning Configuration Attributes 343, 405
  - Damping Factor 345, 407
  - Drive Model Time Consta 407
  - Drive Model Time Constant 345
  - Position Servo Bandwidth 347, 408
  - Test Increment 344, 405
  - Tuning Configuration Bits 347, 409
    - Tune Acceleration Feedforward 348, 409
    - Tune Output Low-Pass Filter 348, 410
    - Tune Position Error Integrator 348, 409
    - Tune Velocity Error Integrator 348, 409
    - Tune Velocity Feedforward 348, 409
    - Tuning Direction Reverse 347, 409
  - Tuning Speed 344, 406
  - Tuning Torque 345, 406
  - Tuning Travel Limit 344, 405
  - Velocity Servo Bandwidth 346, 407



- Commissioning Status Attributes 315, 369
    - Test Direction Forward 316, 370
    - Test Status 316, 369
    - Tune Acceleration/Deceleration 317, 371
    - Tune Acceleration/Deceleration Time 317, 370
    - Tune Inertia 318, 371
    - Tune Rise Time 317
    - Tune Speed Scaling 317
    - Tune Status 316, 370
  - Configuring a 1394C-SJT05/10/22-D Digital Servo Drive 191
    - consumed tag 89
  - Coordinate System Properties
    - Dynamics Tab 184
      - Manual Adjust 186
        - Reset Button 186
      - Manual Adjust Button 185
      - Position Tolerance Box 185
        - Actual 185
        - Command 185
      - Vector Box 184
        - Maximum Acceleration 185
        - Maximum Deceleration 185
        - Maximum Speed 185
    - Editing 179
    - General Tab 180
      - Axis Grid 181
        - Axis Name 181
        - Coordinate 181
        - Coordination Mode 181
        - Ellipsis Button (...) 181
      - Dimension 180
      - Ellipsis button 180
      - Enable Coordinate System Auto Tag Update 182
      - Motion Group 180
      - New Group button 180
      - Type 180
    - Tag Tab 187
      - Data Type 188
      - Description 187
      - Name 187
      - Scope 188
      - Style 188
      - Tag Type 188
    - Units Tab 183
      - Axis Grid 183
        - Axis Name 183
        - Conversion Ratio 183
        - Conversion Ratio Units 183
      - Coordination Units 183
  - Coordinate System Wizard Screens 177
    - Dynamics Wizard Screen 178
    - General Wizard Screen 178
    - Manual Adjust Button 178
    - Tag Wizard Screen 178
    - Units Wizard Screen 178
  - Creating a Coordinate System 173
    - Entering Tag Information 175
      - Parameters 175
        - Alias For 176
        - Data Type 177
        - Description 176
        - Name 176
        - Scope 177
        - Style 177
        - Tag Type 176
          - Base 176
  - Creating A Motion Group 79
- ## D
- Developing a Motion Control Application Program 10
  - Diagrams
    - Wiring 423
      - 1398 to Termination Panel Wiring Diagram 428
      - 1784-PM02AE Sync Cable 427
      - 1784-PM02AE Termination Panel 423
        - 1784-PM02AE-TPOx Cable 426
          - P1 424
          - P2 424
          - P3 and P4 425
  - Direct Commands
    - Accessing 257
      - From Axis 261
      - From Group 259
      - From the Main Menu 257
    - Dialog 264
      - On-line 264
        - Close Button 265
        - Execute Button 265
        - Help Button 265
        - Motion Group Shutdown Button 265
    - Supported Commands 262
      - Motion Event 263
      - Motion Group 263
      - Motion Move 263
      - Motion State 262
  - Drive Fault Actions 401
    - Advanced Stop Action Attributes 403
    - Brake Engage Delay 403
    - Brake Release Delay 404
    - Disable Drive 402
    - Shutdown 402
    - Status Only 403
    - Stop Command 402

- Drive Fault Inputs 321
  - Drive Limits 398
    - Advanced Drive Limits 400
    - Maximum Positive/Negative Travel 398
    - Position Error Tolerance 398
    - Position Lock Toleranc 399
  - Drive Offsets 400
    - Friction Compensation 400
    - Torque Offset 401
    - Velocity Offset 401
  - Drive Power Attributes 404
    - Bus Regulator ID 404
    - Power Supply ID 404
- E**
- Editing 1784-PM16SE Module Properties 67
  - Editing Controller Properties 17
    - Advanced Tab 32
      - Controller Fault Handler 33
      - Memory Total 32
      - Memory Unused 32
      - Memory Used 32
      - Power-Up Handler 33
      - System Overhead Time Slice 33
    - Date/Time Tab 30
      - Coordinated System Time master 31
      - Date 31
      - Set 31
      - Status 31
      - Time 31
    - File Tab 35
      - Created 35
      - Edited 35
      - Name 35
      - Path 35
    - General Tab 17
      - Change Type 18
      - Chassis Type 18
      - Description 18
      - Name 18
      - Revision 18
      - Slot 18
      - Type 17
      - Vendor 17
    - Major Faults Tab 28
      - Clear Majors 29
      - Number of Major Faults Since Last Cleared 29
      - Recent Faults 29
    - Minor Faults Tab 29
      - Clear Minors 30
      - Fault Bits 30
      - Number of Minor Faults Since Last Cleared 29
      - Recent Faults 30
  - Nonvolatile Memory Tab 39
    - Image Note 42
    - Load Image 42
    - Load Mode 42
    - Load/Store 43
    - Name 41
    - Revision 41
    - Stored 43
    - Type 41
  - Redundancy Tab 36
    - Advanced Button 39
      - Memory Usage 39
      - Retain Test Edits on Switchover 39
    - Partner Minor Faults 38
    - Partner Status 38
    - Redundancy Enabled 36
    - Redundancy Status 36
  - Serial Port Tab 20
    - Baud Rate 20
    - Continuous Carrier 21
    - Control Line 21
    - Data Bits 20
    - Mode 20
    - Parity 20
    - RTS Off Delay 21
    - RTS Send Delay 21
    - Stop Bits 21
  - SFC Execution Tab 33
    - Execution Control 34
    - Last Scan of Active Steps 34
    - Restart Position 34
  - System Protocol Tab 22
    - Active Station Tag 25
    - Common Parameters 22
      - ACK Timeout 23
      - Enable Duplicate Detection 23
      - Error Detection 23
      - Protocol 22
      - Station Address 22
    - DF1 Master Parameters 24
      - Master Transmit 25
      - Polling Mode 24
      - Reply Message Wait 24
      - Transmit Retries 24
    - DF1 Point to Point Parameters 23
      - Embedded Responses 23
      - ENQ Transmit Limit 23
      - NAK Receive Limit 23
    - DF1 Slave Parameters 24
      - EOT Suppression 24
      - Slave Poll Timeout 24

- Transmit Retries 24
  - DH485 Parameters 25
    - Max Station Address 25
    - Token Hold Factor 26
  - Normal Poll Group Size 25
  - Normal Poll Node Tag 25
  - Priority Poll Node Tag 25
- User Protocol Tab 26
  - Append Character 1 and 2 27
  - Buffer Size 26
  - Delete Mode 28
  - Echo Mode 27
  - Protocol 26
  - Termination Character 1 and 2 27
  - XON/OFF 27
- Editing the Motion Group Properties 82
  - Attribute Tab 83
    - Auto Tag Update 84
    - Base Tag 86
    - Coarse Update Period 84
    - Data Type 86
    - Description 85
    - General Fault Type 84
    - Name 85
    - Produce 86
    - Reset Max 84
    - Scan Times 84
    - Scope 86
    - Style 86
    - Tag Type 85
  - Axis Assignment Tab 83
    - Add 83
    - Assigned 83
    - Remove 83
    - Unassigned 83
  - Tag Tab 85
- Editing the Ultra Drive Properties 211
  - Associated Axes Tab (Ultra3000 Drives) 217
    - Ellipsis (...) 217
    - New Axis 217
    - Node 217
  - Connection Tab 214
    - Inhibit Module 215, 230
    - Major Fault 216
    - Module Fault 216
      - Connection Request Error 216
      - Electronic Keying Mismatch 216
      - Module Configuration Invalid 216
      - Service Request Error 216
    - Requested Packet Interval 215, 244
  - General Tab 211
    - Description 212
    - Electronic Keying 213
    - Compatible Modules 213
    - Disable Keying 213
    - Exact Match 213
    - Name 212
    - Node 212
    - Revision 212
    - Slot 212
    - Status 214
    - Type 211
    - Vendor 212
  - Module Info 219
    - Configured 221
    - Identification 220
    - Internal State Status 220
    - Major/Minor Fault Status 220
    - Module Identity 221
    - Owned 221
    - Refresh 222
    - Reset Module 222
  - Power Tab - Ultra Drive 218
    - Bus Regulator ID 218
- Editing Your1784-PM02AE Motion Module Settings 51

## F

- Fault handling
  - Errors 12
  - Minor/major faults 12
- Faults
  - Types 12
- Feedback Tab (AXIS\_SERVO) 104
  - Feedback Type 104
- Following Error 392

## G

- General Tab - AXIS\_VIRTUAL 97

## H

- Handling Motion Faults 12

## I

- Inputs:Drive Fault 321

## K

- Kinetix 6000 Drive
  - Configuring 223
- Kinetix Drive
  - Properties 225
    - Associated Axes Tab 231

- Ellipsis (...) 232
- New Axis 232
- Node 232
- Connection Tab 229
  - Major Fault 230
  - Module Fault 231
    - Connection Request Error 231
    - Electronic Keying Mismatch 231
    - Module Configuration Invalid 231
    - Service Request Error 231
  - Requested Packet Interval 229
- General Tab 226
  - Type 226
  - Description 227
  - Electronic Keying 227
    - Compatible Modules 227
    - Disable Keying 227
    - Exact Match 227
  - Name 226
  - Node 227
  - Revision 227
  - Status 228
  - Vendor 226
- Module Info Tab 233
  - Configured 235
  - Identification 234
  - Internal State Status 234
  - Major/Minor Fault Status 234
  - Module Identity 235
  - Owned 235
  - Refresh 236
  - Reset Module 235
- Power Tab 232
  - Bus Regulator Catalog Number 232

## M

- Modifying Motion Configuration Parameters 11
- Motion Apply Axis Tuning 256
- Motion Apply Hookup Diagnostic 256
- Motion Arm Output Cam 255
- Motion Arm Registration 255
- Motion Arm Watch Position 255
- Motion attributes
  - Changing configuration parameters 11
- Motion Axis Fault Reset 254
- Motion Axis Gear 254
- Motion Axis Home 254
- Motion Axis Jog 254
- Motion Axis Move 254
- Motion Axis Position Cam 254
- Motion Axis Shutdown 253

- Motion Axis Shutdown Reset 254
- Motion Axis Stop 254
- Motion Axis Time Cam 254
- Motion Calculate Cam Profile 254
- Motion Change Dynamics 254
- Motion Configuration Instructions 255
- Motion Coordinated Change Dynamics 256
- Motion Coordinated Circular Move 256
- Motion Coordinated Linear Move 256
- Motion Coordinated Move Instructions 256
- Motion Coordinated Shutdown 256
- Motion Coordinated Shutdown Reset 256
- Motion Coordinated Stop 256
- Motion Direct Commands 257
- Motion Direct Drive Off 254
- Motion Direct Drive On 254
- Motion Disarm Output Cam 255
- Motion Disarm Registration 255
- Motion Disarm Watch Position 255
- Motion Dynamics Configuration 299
  - Maximum Acceleration/Deceleration 299
  - Maximum Speed 299
  - Programmed Stop Mode 300
    - Fast Disable 300
    - Fast Shutdown 300
    - Fast Stop 300
    - Hard Disable 300
    - Hard Shutdown 301
- Motion Event Instructions 255
- Motion Group 79
- Motion Group Instructions 255
- Motion Group Shutdown 255
- Motion Group Shutdown Reset 255
- Motion Group Stop 255
- Motion Group Strobe Position 255
- Motion Instructions 253
  - Motion Configuration Instructions 255
    - Motion Apply Axis Tuning (MAAT) 256
    - Motion Apply Hookup Diagnostic (MAHD) 256
    - Motion Run Axis Tuning (MRAT) 256
    - Motion Run Hookup Diagnostic (MRHD) 256
  - Motion Coordinated Move Instructions 256
    - Motion Coordinated Change Dynamics MCCD 256
    - Motion Coordinated Circular Move (MCCM) 256
    - Motion Coordinated Linear Move (MCLM) 256
    - Motion Coordinated Shutdown (MCS) 256
    - Motion Coordinated Shutdown Reset (MCSR) 256
    - Motion Coordinated Stop (MCS) 256
  - Motion Direct Commands 257
  - Motion Event Instructions 255
    - Motion Arm Output Cam (MAOC) 255

- Motion Arm Registration (MAR) 255
- Motion Arm Watch Position (MAW) 255
- Motion Disarm Output Cam (MDOC) 255
- Motion Disarm Registration (MDR) 255
- Motion Disarm Watch Position (MDW) 255
- Motion Group Instructions 255
  - Motion Group Shutdown (MGSD) 255
  - Motion Group Shutdown Reset (MGSR) 255
  - Motion Group Stop (MGS) 255
  - Motion Group Strobe Position (MGSP) 255
- Motion Move Instructions 254
  - Motion Axis Gear (MAG) 254
  - Motion Axis Home (MAH) 254
  - Motion Axis Jog (MAJ) 254
  - Motion Axis Move (MAM) 254
  - Motion Axis Position Cam (MAPC) 254
  - Motion Axis Stop (MAS) 254
  - Motion Axis Time Cam (MATC) 254
  - Motion Calculate Cam Profile (MCCP) 254
  - Motion Change Dynamics (MCD) 254
  - Motion Redefine Position (MRP) 254
- Motion State Instructions 253
  - Motion Axis Fault Reset (MAFR) 254
  - Motion Axis Shutdown (MASD) 253
  - Motion Axis Shutdown Reset (MASR) 254
  - Motion Direct Drive Off (MDF) 254
  - Motion Direct Drive On (MDO) 254
  - Motion Servo Off (MSF) 253
  - Motion Servo On (MSO) 253
- Motion Move Instructions 254
- Motion Object Attributes 267
  - Object Support Attributes 267
    - Axis Configuration State T 270
    - Axis Data Type 269
    - Axis Instance 267
    - Axis State 270
    - C2C Connection Instance 269
    - C2C Map Instance 268
    - Group Instance 268
    - Map Instance 268
    - Memory Usage 269
    - Memory Use 269
    - Module Channel 268
    - Module Class Code 268
    - Output Cam Execution Targets 270
  - Status Attributes 270
- Motion Object Configuration Attributes 287
  - Conversion Configuration 288
    - Conversion Constant 288
    - Position Unwind I 289
    - Rotary Axis 289
  - Home Configuration
    - Bits 297
    - Home Offset 298
    - Home Position 297
  - Home Configuration Bits
    - Home Switch Normally Closed 297
  - Homing Configuration 289
    - Home Mode 289
      - Absolute 290
      - Active 290
      - Passive 290
    - Home Return Speed 298
    - Home Sequence & Direction 290
      - Active Bi-directional Home with Marker 293
      - Active Bi-directional Home with Switch 291
      - Active Bi-directional Home with Switch then Marker 294
      - Active Homing 291
      - Active Immediate Home 291
      - Active Uni-directional Home with Marker 295
      - Active Uni-directional Home with Switch 294
      - Active Uni-directional Home with Switch then Marker 295
      - Passive Home with Marker 296
      - Passive Home with Switch 296
      - Passive Home with Switch then Marker 297
      - Passive Homing 296
      - Passive Immediate Home 296
    - Home Speed 298
  - Unit Configuration 287
    - Average Velocity Timebase 287
- Motion Redefine Position 254
- Motion Run Axis Tuning 256
- Motion Run Hookup Diagnostic 256
- Motion Servo Off 253
- Motion Servo On 253
- Motion Status and Configuration Parameters 11
- Motion Status Attributes 271
  - Actual Acceleration 275
  - Actual Position 271
  - Actual Velocity 274
  - Average Velocity 273
  - Command Acceleration 275
  - Command Position 272
  - Command Velocity 274
  - Interpolated Actual Position 277
  - Interpolated Command Position 277
  - Interpolation Time 277
  - Master Offset 277
  - Registration Position 276
  - Registration Time 277
  - Start Master Offset 278
  - Start Position 273

- Strobe Master Offset 278
- Strobe Position 272
- Watch Position 275
- Motion Status Bit Attributes 279
  - Axis Event Bit Attributes 284
    - Home Event Armed Status 285
    - Home Event Status 286
    - Registration 1 Event Armed Status 285
    - Registration 1 Event Status 285
    - Registration 2 Event Armed Status 285
    - Registration 2 Event Status 285
    - Watch Event Armed Status 285
    - Watch Event Status 285
  - Axis Fault Bit Attributes 283
    - Configuration Fault 283
    - Module Fault 283
    - Physical Axis Fault 283
  - Axis Status Bit Attributes 282
    - Configuration Update in Process 282
    - Drive Enable Status 282
    - Servo Action Status 282
    - Shutdown Status 282
  - Module Fault Bit Attribute 283
    - Control Sync Fault 284
- Motion Status Bits 279
  - Acceleration/Deceleration Status 279
  - Gearing Lock Status 281
  - Gearing Status 280
  - Homed Status 280
  - Homing Status 280
  - Jog Status 280
  - Master Offset Move Status 281
  - Move Status 279
  - Position Cam Lock Status 281
  - Position Cam Pending Status 281
  - Position Cam Status 280
  - Stopping Status 280
  - Time Cam Pending Status 281
  - Time Cam Status 280
- Output Cam Lock Status 286
- Output Cam Pending Status 286
- Output Cam Status 286
- Output Cam Transition Status 287
- MOTION\_INSTRUCTION control structure
  - Motion Instruction tag 10
- Motor and Feedback Configuration 380
  - Advanced Feedback Attributes 382
  - Aux Feedback Ratio 381
  - Feedback Resolution 381
  - Feedback Type 381
  - Motor Data 381
  - Motor ID 380

## N

- Naming & Configuring Your Motion Axis 87
- Naming a Coordinate System
  - Entering Tag Information
    - Parameters
      - Tag Type
        - Alias 176
- Naming an Axis 87
  - Entering Tag Information 89
    - Common Parameters 90
      - Data Type 91
      - Description 90
      - Name 90
      - Tag Type 90
        - Alias 90
        - Base 90
        - Consumed 90
        - Produced 90
- New Controller Dialog
  - Redundancy Enabled 15

## P

- Performance guidelines 417
- product support, local
  - telephone number 3

## Q

- Quadrature 363

## R

- Related Documentation 2
- Right Mouse Click Properties 188
  - Copy/Paste 189
  - Cut, Copy, Paste, and Delete Behavior 189
  - Cut/Paste 190
  - Delete 190
- Rockwell Automation Support 2
  - Local Product Support 3
  - On the Web 3
- Rockwell Automation support
  - Technical product assistance 3
- RSLogix 5000 programming software
  - Motion Instructions 253

## S

- Select Module Type window 46
- Servo Amplifiers 333, 335, 395
- Servo Configuration Attributes 319

- Acceleration Feedforward Gain 330
- Axis Info Select 322
- Axis Type 319
- Bandwidth Method 331
- External Drive Type 320
- Fault Configuration Bits 321
  - Drive Fault Checking 321
  - Drive Fault Normally Closed 322
  - Hard Overtravel Checking 321
  - Soft Overtravel Checking 321
- Integrator Hold Enable 337
- Output LP Filter Bandwidth 337
- Position Integral Gain 332
- Position Proportional Gain 331
  - Bandwidth Method 331
  - Loop Gain Method 331
  - Maximum Bandwidth 332
- Position Servo with Torque Servo Drive 324
- Position Servo with Velocity Servo Drive 326
- Servo Configuration 319
- Servo Gains 328
- Servo Loop Block Diagrams 323
- Servo Loop Configuration 320
- Servo Polarity Bits 322
  - Feedback Polarity Negative 323
  - Servo Polarity Negative 323
- Torque Scaling 336
- Velocity Feedforward Gain S 329
- Velocity Integral Gain 335
- Velocity Proportional Gain 333
  - Maximum Bandwidth 334
- Velocity Scaling 335
- Velocity Servo with Torque Servo Drive 325
- Velocity Servo with Velocity Servo Drive 327
- Servo Drive Configuration Attributes 372
  - Advanced Polarity Attributes 379
  - Advanced Scaling Attributes 377
    - Rotational Position Resolution 377
  - Advanced Servo Configuration Attributes 374
  - Axis Info Select 380
  - Axis Type 373
  - Drive Axis ID 373
  - Drive Configuration 373
  - Drive Polarity 378
    - Custom Polarity 378
    - Positive/Negative Polarity 378
  - Drive Scaling Bits 376
    - Custom Scaling 376
  - Fault Configuration Bits 375
    - Hard Overtravel Checking 376
    - Soft Overtravel Checking 376
  - Servo Loop Configuration 374
  - Servo Drive Status Attributes 348
    - Acceleration Command 352
    - Acceleration Feedback 352
    - Attribute Error Code 368
    - Attribute Error ID 368
    - Aux Position Feedback 350
    - Axis Control Bit Attributes 358
      - Abort Event Acknowledge 359
      - Abort Event Request 358
      - Abort Home Acknowledge 359
      - Abort Home Request 358
      - Abort Process 358
      - Abort Process Acknowledge 359
      - Change Cmd Reference 359
      - Change Pos Reference 360
      - Shutdown Request 358
      - Shutdown Request Acknowledge 359
    - Axis Response Bit Attributes 359
    - Bus Regulator Capacity 353
    - DC Bus Voltage 354
    - Drive Capacity 353
    - Drive Fault Bit Attributes 360
      - Commutation Fault 364
      - Drive Control Voltage Fault 364
      - Drive Cooling Fault 364
      - Drive Hardware Fault 363
      - Drive Overcurrent Fault 364
      - Drive Overtemperature Fault 364
      - Drive Overvoltage Fault 364
      - Drive Undervoltage Fault 364
      - Feedback 1 or Feedback 2 Fault 362
      - Feedback 1 or Feedback 2 Noise Fault 363
      - Feedback Fault 364
      - Ground Short Fault 363
      - Motor Overtemperature Fault 364
      - Overload Fault 363
      - Overspeed Fault 363
      - Position Error Fault 362
      - Positive/Negative Hardware Overtravel Faults 362
      - Positive/Negative Software Overtravel Faults 362
      - Power Phase Loss Fault 365
      - SERCOS Fault 365
    - Drive Status Attributes 348
    - Drive Status Bit Attributes 355
      - Absolute Reference Status 357
      - Acceleration Limit Status 357
      - Drive Enable Status 356
      - Enable Input Status 357
      - Home Input Status 356
      - Negative Overtravel Input Status 356
      - Position Lock Status 358

- Positive Overtravel Input Status 356
- Power Limit Status 358
- Process Status 356
- Registration 1/2 Input Status 356
- Servo Action Status 356
- Shutdown Status 356
- Torque Limit Status 357
- Velocity Limit Status 357
- Velocity Lock Status 357
- Velocity Standstill Status 357
- Velocity Threshold 357
- Drive Warning Bit Attributes 367
  - Cooling Error Warning 367
  - Drive Overtemperature Warning 367
  - Motor Overtemperature Warning 367
  - Overload Warning 367
- Marker Distance 352
- Module Fault Bit Attributes 365
  - Control Sync Fault 366
  - Module Hardware Fault 366
  - Module Sync Fault 366
  - SERCOS Ring Fault 366
  - Timer Event Fault 366
- Motor Capacity 353
- Motor Electrical Degrees 353
- Pos./Neg. Dynamic Torque Limit 353
- Position Command 350
- Position Error 350
- Position Feedback 350
- Position Integrator Error 350
- Power Capacity 353
- SERCOS Error Code 368
- Torque Command 352
- Torque Feedback 352
- Torque Limit Source 354
- Velocity Command 351
- Velocity Error 351
- Velocity Feedback 351
- Velocity Integrator Error 351
- Servo Fault Configuration 342
  - Servo Fault Actions 342
    - Disable Drive 343
    - Shutdown 342
    - Status Only 343
    - Stop Command 343
- Servo Limits 337
  - Maximum Positive/Negative Travel 337
  - Output Limit 339
  - Position Error Tolerance 338
  - Position Lock Tolerance 338
- Servo Loop Block Diagrams 382
  - Acceleration Feedforward Gain 393
  - Advanced Drive Gain Attributes 398
  - Auxiliary Dual Command Servo 387
  - Auxiliary Position Servo 384
  - Drive Gains 389
  - Dual Position Servo 385
  - Integrator Hold Enable 397
  - Motor Dual Command Servo 386
  - Motor Position Servo 383
  - Output LP Filter Bandwidth 396
  - Output Notch Filter Frequency 396
  - Position Integral Gain 391
  - Position Proportional Gain 390
    - Bandwidth Method 390
    - Loop Gain Method 390
    - Maximum Bandwidth 391
  - Torque Scaling 397
  - Torque Servo 389
  - Velocity Feedforward Gain 392
  - Velocity Integral Gain 395
  - Velocity Proportional Gain 394
    - Maximum Bandwidth 395
  - Velocity Servo 388
- Servo Offsets 340
  - Friction Compensation 340
  - Output Offset 341
  - Torque Offset 341
  - Velocity Offset 340
- Servo Status Attributes 301
  - Acceleration Command 304
  - Acceleration Feedback 305
  - Aux Position Feedback 303
  - Marker Distance 305
  - Position Command 302
  - Position Error 303
  - Position Feedback 302
  - Position Integrator Error 303
  - Servo Output Level 305
  - Servo Status Attributes 301
  - Velocity Command 303
  - Velocity Error 304
  - Velocity Feedback 304
  - Velocity Integrator Error 304
- Servo Status Bit Attributes 306
  - Attribute Error Code 314
  - Attribute Error ID 315
  - Axis Control Bit Attributes 308
    - Abort Event Request 308
    - Abort Home Request 308
    - Abort Process Request 308



- Change Cmd Reference 308
- Shutdown Request 308
- Zero DAC Request 308
- Axis Response Bit Attributes 309
  - Abort Event Acknowledge 309
  - Abort Home Acknowledge 309
  - Abort Process Acknowledge 309
  - Change Position Reference 309
  - Shutdown Request Acknowledge 309
  - Zero DAC Request Acknowledge 309
- Drive Enable Status 306
- Home Input Status 307
- Module Fault Bit Attributes 312
  - Control Sync Fault 313
  - Module Hardware Fault 314
  - Module Sync Fault 313
  - Timer Event Fault 314
- Negative Overtravel Input Status 307
- Output Limit Status 307
- Position Lock Status 307
- Positive Overtravel Input Status 307
- Process Status 307
- Registration 1/2 Input Status 307
- Servo Action Status 306
- Servo Fault Bit Attributes 310
  - Drive Fault 312
  - Feedback Loss Fault 311
  - Feedback Noise Fault 311
  - Position Error Fault 312
  - Positive/Negative Hardware Overtravel Faults 311
  - Positive/Negative Soft Overtravel Status 310
- Shutdown Status 306
- SoftLogix Controller 13
  - Introduction 13
  - New Controller Dialog 14
    - Browse 16
    - Chassis Type 16
    - Create In 16
    - Description 15
    - Name 15
    - Revision 16
    - Slot Number 16
    - Type 15
    - Vendor 15
- SoftLogix Motion Control 5
  - 1784-PM02AE servo module 5
  - 1784-PM16SE SERCOS interface module 5

- Chassis Monitor 5
  - RSLogix 5000 programming software 5
  - SoftLogix5860 controller 5
- SoftLogix motion control
  - Components 6
    - 16 Axis SERCOS interface Card 9
    - 1784-PM02AE servo module 9
    - Controller 8
    - RSLogix 5000 programming software 9
  - Componets
    - Chassis Monitor 7
- Specifications 417
- SSV instruction
  - Changing configuration parameters 11
- Stop Motion 402

## T

- Termination 27
- Troubleshooting 411
  - 1756-M16SE LED
    - SERCOS interface LED Indicators 412
  - 1784-PM02AE LED 411
    - Status LED 411
  - 1784-PM16SE LED
    - Communication Phase LED 415
    - OK LED 413
    - SERCOS interface LED 412
      - Checking the LED Indicators 412
    - SERCOS Ring Status LED 414

## U

- Ultra 3000 Drive 205
- Using This Manual 1

## W

- Windows
  - New module
    - New Module window 49
  - Select module type 46
- Wiring Diagrams 421
  - Block 421
    - Torque Servo Drive 422
    - Velocity Servo Drive 423
- World Wide Web site 3

For more information refer to our web site: **[www.ab.com/motion](http://www.ab.com/motion)**

For Allen-Bradley Technical Support information refer to: [www.ab.com/support](http://www.ab.com/support) or Tel: (1) 440.646.5800

**[www.rockwellautomation.com](http://www.rockwellautomation.com)**

---

**Power, Control and Information Solutions Headquarters**

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1784-UM003A-EN-P - June 2003

PN 957400-34

Copyright © 2003 Rockwell Automation. All rights reserved. Printed in USA.

*Rockwell Automation*

*SoftLogix PCI Motion Card*

*Setup and Configuration Manual*