

# CONEX-AGAP

## *Agilis-D Controller with Strain Gages Feedback*



**Newport®**  
Experience | Solutions

## Command Library API Manual

V2.0.0

*For Motion, Think Newport™*

# Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	PURPOSE .....	1
1.2	OVERVIEW .....	1
<b>2</b>	<b>COMMAND INTERFACE .....</b>	<b>1</b>
2.1	CONSTRUCTOR.....	1
2.2	FUNCTIONS .....	1
2.2.1	<i>General</i> .....	1
2.2.1.1	OpenInstrument.....	1
2.2.1.2	CloseInstrument.....	2
2.2.1.3	GetDevices .....	2
2.2.1.4	WriteToInstrument .....	2
2.2.2	<i>Commands</i> .....	2
2.2.2.1	DB_Get.....	2
2.2.2.2	DB_Set.....	3
2.2.2.3	DD_Get.....	3
2.2.2.4	DD_Set .....	3
2.2.2.5	ID_Get .....	3
2.2.2.6	ID_Set.....	4
2.2.2.7	JA_Get .....	4
2.2.2.8	JA_Set.....	4
2.2.2.9	KI_Get.....	4
2.2.2.10	KI_Set.....	5
2.2.2.11	KI_Set.....	5
2.2.2.12	KP_Get .....	5
2.2.2.13	KP_Set.....	5
2.2.2.14	KY_Get .....	6
2.2.2.15	KY_Set.....	6
2.2.2.16	KZ_Get .....	6
2.2.2.17	KZ_Set.....	6
2.2.2.18	LF_Get.....	7
2.2.2.19	LF_Set .....	7
2.2.2.20	MM_Get .....	7
2.2.2.21	MM_Set.....	8
2.2.2.22	PA_Get.....	8
2.2.2.23	PA_Set.....	8
2.2.2.24	PR_Get .....	8
2.2.2.25	PR_Set.....	9
2.2.2.26	PW_Get.....	9
2.2.2.27	PW_Set .....	9
2.2.2.28	RS.....	10
2.2.2.29	RS485 .....	10
2.2.2.30	SA_Get .....	10
2.2.2.31	SA_Set.....	10
2.2.2.32	SL_Get.....	11
2.2.2.33	SL_Set .....	11
2.2.2.34	SR_Get .....	11
2.2.2.35	SR_Set.....	11
2.2.2.36	ST .....	12
2.2.2.37	SU_Get.....	12

2.2.2.38 SU\_Set ..... 12

2.2.2.39 TB..... 12

2.2.2.40 TE ..... 13

2.2.2.41 TH..... 13

2.2.2.42 TP ..... 13

2.2.2.43 TS ..... 13

2.2.2.44 VE..... 14

2.2.2.45 XR\_Get ..... 14

2.2.2.46 XR\_Set..... 14

2.2.2.47 XU\_Get..... 14

2.2.2.48 XU\_Set ..... 15

2.2.2.49 ZT ..... 15

SERVICE FORM.....16



# CONEX-AGAP

## Agilis-D Controller with Strain Gages Feedback

### 1 Introduction

---

#### 1.1 Purpose

The purpose of this document is to describe the application programming interface (API) of the command library (ConexAGAPCmdLib.dll) that is used to communicate with the CONEX-AGAP device.

#### 1.2 Overview

The command library provides public methods to communicate with any CONEX-AGAP device and these methods work in both synchronous and asynchronous mode. Many of the ASCII commands that can be programmatically sent to the instrument have a corresponding method that can be called in the command library. For example, the ASCII “VE” command can be sent to the instrument to get the controller version, and the command library has a corresponding public method “VE” that returns the controller version and error information. For more information on a particular ASCII command see the manual for the controller.

### 2 Command Interface

---

#### 2.1 Constructor

ConexAGAPCmds ()

The constructor is used to create an instance of the command library.

#### 2.2 Functions

##### 2.2.1 General

##### 2.2.1.1 OpenInstrument

###### Syntax

int OpenInstrument(string strDeviceKey)

string strDeviceKey: the device key is a serial COM port

return: 0 = successful or -1 = failure

###### Description

This function allows opening communication with the selected device. If the opening failed, the returned code is -1.

### 2.2.1.2 CloseInstrument

#### Syntax

int CloseInstrument()

return: 0 = successful or -1 = failure

#### Decription

This function allows closing communication with the selected device. If the closing failed, the returned code is -1.

### 2.2.1.3 GetDevices

#### Syntax

string[] GetDevices()

return: list of strings that contains the accessible COM ports.

#### Decription

This function returns the list of connected devices available to communicate.

### 2.2.1.4 WriteToInstrument

#### Syntax

int WriteToInstrument(string command, ref string resp, int stage)

command: Instrument command

resp: Response of the command

stage: Instrument Stage

return: function error

#### Decription

This overridden function Queries or writes the command given by the user to the instrument.

## 2.2.2 Commands

### 2.2.2.1 DB\_Get

#### Syntax

int DB\_Get (int controllerAddress, string Axis, out double CorrectorDeadband, out string errstring)

controllerAddress: Address of Controller

CorrectorDeadband: CorrectorDeadband

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous DB Get command which is used to Get corrector deadband.

#### 2.2.2.2 DB\_Set

##### **Syntax**

int DB\_Get (int controllerAddress, string Axis, double CorrectorDeadband, out string errstring)

controllerAddress: Address of Controller

CorrectorDeadband: CorrectorDeadband

errString: The failure reason

Return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous DB Set command which is used to Set corrector deadband.

#### 2.2.2.3 DD\_Get

##### **Syntax**

int DD\_Get (int controllerAddress, string Axis, out int DeadbandSettlingTime, out string errstring)

controllerAddress: Address of Controller

DeadbandSettlingTime: DeadbandSettlingTime

errString: The failure reason

Return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous DD Get command which is used to Get deadband settling time.

#### 2.2.2.4 DD\_Set

##### **Syntax**

int DD\_Set (int controllerAddress, string Axis, int DeadbandSettlingTime, out string errstring)

controllerAddress: Address of Controller

DeadbandSettlingTime: DeadbandSettlingTime

errString: The failure reason

Return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous DD Set command which is used to Set deadband settling time.

#### 2.2.2.5 ID\_Get

##### **Syntax**

int ID\_Get(int controllerAddress, out string StageIdentifier, out string errString)

controllerAddress: Address of Controller

StageIdentifier: StageIdentifier

errString: The failure reason

Return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous ID Get command which is used to Get stage identifier.

### 2.2.2.6 ID\_Set

#### **Syntax**

int ID\_Set(int controllerAddress, string StageIdentifier, out string errString)

controllerAddress: Address of Controller

StageIdentifier: StageIdentifier

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous ID Set command which is used to Set stage identifier.

### 2.2.2.7 JA\_Get

#### **Syntax**

int JA\_Get(int controllerAddress, string Axis, out double JogVelocity, out string errString)

controllerAddress: Address of Controller

JogVelocity: JogVelocity

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous JA Get command which is used to Move jogging.

### 2.2.2.8 JA\_Set

#### **Syntax**

int JA\_Set(int controllerAddress, string Axis, double JogVelocity, out string errString)

controllerAddress: Address of Controller

JogVelocity: JogVelocity

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous JA Set command which is used to Move jogging.

### 2.2.2.9 KI\_Get

#### **Syntax**

int KI\_Get(int controllerAddress, string Axis, out int IntegralGain, out string errstring)

controllerAddress: Address of Controller

IntegralGain: IntegralGain

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous KI Get command which is used to Get integral gain.



### 2.2.2.10 KI\_Set

#### **Syntax**

int KI\_Set(int controllerAddress, string Axis, out int IntegralGain, out string errstring)

controllerAddress: Address of Controller

IntegralGain: IntegralGain

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous KI Set command which is used to Set integral gain.

### 2.2.2.11 KI\_Set

#### **Syntax**

int int KI\_Set (int controllerAddress, string Axis, int IntegralGain, out string errstring)

controllerAddress: Address of Controller

IntegralGain: IntegralGain

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous KI Set command which is used to Set integral gain.

### 2.2.2.12 KP\_Get

#### **Syntax**

int KP\_Get (int controllerAddress, string Axis, out double ProportionalGain, out string errstring)

controllerAddress: Address of Controller

ProportionalGain: ProportionalGain

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous KP Get command which is used to Get proportional gain.

### 2.2.2.13 KP\_Set

#### **Syntax**

int KP\_Set (int controllerAddress, string Axis, double ProportionalGain, out string errstring)

controllerAddress: Address of Controller

ProportionalGain: ProportionalGain

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous KP Set command which is used to Set proportional gain.

#### 2.2.2.14 KY\_Get

##### Syntax

int KY\_Get (int controllerAddress, string CoeffReference, out double CalibrationCoefficients, out string errstring)

controllerAddress: Address of Controller

CoeffReference: CoeffReference

CalibrationCoefficients: CalibrationCoefficients

errString: The failure reason

Return: 0 in success and -1 on failure

##### Description

This function is used to process synchronous KY Get command which is used to Get calibration coefficients.

#### 2.2.2.15 KY\_Set

##### Syntax

int KY\_Set (int controllerAddress, string CoeffReference, double CalibrationCoefficients, out string errstring)

controllerAddress: Address of Controller

CoeffReference: CoeffReference

CalibrationCoefficients: CalibrationCoefficients

errString: The failure reason

Return: 0 in success and -1 on failure

##### Description

This function is used to process synchronous KY Set command which is used to Set calibration coefficients.

#### 2.2.2.16 KZ\_Get

##### Syntax

int KZ\_Get (int controllerAddress, string CoeffReference, out double ConfigurationCoefficients, out string errstring)

controllerAddress: Address of Controller

CoeffReference: CoeffReference

ConfigurationCoefficients: ConfigurationCoefficients

errString: The failure reason

Return: 0 in success and -1 on failure

##### Description

This function is used to process synchronous KZ Get command which is used to Get configuration coefficients.

#### 2.2.2.17 KZ\_Set

##### Syntax

int KZ\_Set (int controllerAddress, string CoeffReference, double ConfigurationCoefficients, out string errstring)

controllerAddress: Address of Controller

CoeffReference: CoeffReference

ConfigurationCoefficients: ConfigurationCoefficients

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous KZ Set command which is used to Set configuration coefficients.

#### **2.2.2.18 LF\_Get**

##### **Syntax**

int LF\_Get (int controllerAddress, out double Frequency, out string errstring))

controllerAddress: Address of Controller

Frequency: Frequency

errString: The failure reason

Return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous LF Get command which is used to Get low pass filter frequency.

#### **2.2.2.19 LF\_Set**

##### **Syntax**

int LF\_Set (int controllerAddress, double Frequency, out string errstring))

controllerAddress: Address of Controller

Frequency: Frequency

errString: The failure reason

Return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous LF Set command which is used to Set low pass filter frequency.

#### **2.2.2.20 MM\_Get**

##### **Syntax**

int MM\_Get(int controllerAddress, out string State, out string errString)

controllerAddress: Address of Controller

State: State

errString: The failure reason

Return: 0 in success and -1 on failure

##### **Description**

This function is used to process synchronous MM Get command which is used to Leave DISABLE state.

### 2.2.2.21 MM\_Set

#### **Syntax**

int MM\_Set(int controllerAddress, int State, out string errString)

controllerAddress: Address of Controller

State: State

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous MM Set command which is used to Leave DISABLE state.

### 2.2.2.22 PA\_Get

#### **Syntax**

int PA\_Get(int controllerAddress, string Axis, out double Target, out string errString)

controllerAddress: Address of Controller

Target: Target

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous PA Get command which is used to Move absolute.

### 2.2.2.23 PA\_Set

#### **Syntax**

int PA\_Set(int controllerAddress, string Axis, double Target, out string errString)

controllerAddress: Address of Controller

Target: Target

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous PA Set command which is used to Move absolute.

### 2.2.2.24 PR\_Get

#### **Syntax**

int PR\_Get(int controllerAddress, string Axis, out double Step, out string errString)

controllerAddress: Address of Controller

Step: Step

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous PR Get command which is used to Move relative.

### 2.2.2.25 PR\_Set

#### Syntax

int PR\_Set(int controllerAddress, string Axis, double Step, out string errString)

controllerAddress: Address of Controller

Step: Step

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous PR Set command which is used to Move relative.

### 2.2.2.26 PW\_Get

#### Syntax

int PW\_Get(int controllerAddress, out int State, out string errString)

controllerAddress: Address of Controller

State: State

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous PW Get command which is used to Enter/Leave CONFIGURATION state.

### 2.2.2.27 PW\_Set

#### Syntax

int PW\_Set(int controllerAddress, int State, out string errString)

controllerAddress: Address of Controller

State: State

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous PW Set command which is used to Enter/Leave CONFIGURATION state.

---

#### NOTE

**The PW command is limited to 100 writes. Unit failure due to excessive use of the PW command is not covered by warranty.**

**The PW command is used to change the configuration parameters that are stored in memory, and not parameters that are needed to be changed on the fly.**

---

### 2.2.2.28 RS

#### Syntax

int RS(int controllerAddress, out string errString)

controllerAddress: controllerAddress identifying the Address of Controller

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous RS Set command which is used to Reset controller.

### 2.2.2.29 RS485

#### Syntax

int RS485(int controllerAddress, out string errString)

controllerAddress: controllerAddress identifying the Address of Controller

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous RS## Set command which is used to Reset controller's address to 1.

### 2.2.2.30 SA\_Get

#### Syntax

int SA\_Get(int controllerAddress, out int Address, out string errString)

controllerAddress: Address of Controller

Address: Address

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous SA Get command which is used to Get controller's RS-485 address.

### 2.2.2.31 SA\_Set

#### Syntax

int SA\_Set(int controllerAddress, int Address, out string errString)

controllerAddress: Address of Controller

Address: Address

errString: The failure reason

Return: 0 in success and -1 on failure

#### Description

This function is used to process synchronous SA Set command which is used to Set controller's RS-485 address.

### 2.2.2.32 SL\_Get

**Syntax**

int SL\_Get(int controllerAddress, string Axis, out double Limit, out string errString)

controllerAddress: Address of Controller

Limit: Limit

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous SL Get command which is used to Get negative software limit.

### 2.2.2.33 SL\_Set

**Syntax**

int SL\_Set(int controllerAddress, string Axis, double Limit, out string errString)

controllerAddress: Address of Controller

Limit: Limit

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous SL Set command which is used to Set negative software limit.

### 2.2.2.34 SR\_Get

**Syntax**

int SR\_Get(int controllerAddress, string Axis, out double Limit, out string errString)

controllerAddress: Address of Controller

Limit: Limit

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous SR Get command which is used to Get positive software limit.

### 2.2.2.35 SR\_Set

**Syntax**

int SR\_Set(int controllerAddress, string Axis, double Limit, out string errString)

controllerAddress: Address of Controller

Limit: Limit

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous SR Set command which is used to Set positive software limit.

### 2.2.2.36 ST

#### **Syntax**

int ST(int controllerAddress, out string errString)

controllerAddress: controllerAddress identifying the Address of Controller

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous ST Set command which is used to Stop motion.

### 2.2.2.37 SU\_Get

#### **Syntax**

int SU\_Get (int controllerAddress, out double Resolution, out string errstring)

controllerAddress: controllerAddress identifying the Address of Controller

Resolution: Resolution

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous SU Get command which is used to Get encoder resolution.

### 2.2.2.38 SU\_Set

#### **Syntax**

int SU\_Set (int controllerAddress, double Resolution, out string errstring)

controllerAddress: controllerAddress identifying the Address of Controller

Resolution: Resolution

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous SU Set command which is used to Set encoder resolution.

### 2.2.2.39 TB

#### **Syntax**

int TB(int controllerAddress, string inErrorCode, out string outErrorCode, out string errString)

controllerAddress: Address of Controller

inErrorCode: inErrorCode.

outErrorCode: outErrorCode

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous TB Get command which is used to Get command error string.



**2.2.2.40 TE****Syntax**

int TE(int controllerAddress, out string LastCommandError, out string errString)

controllerAddress: Address of Controller

LastCommandError: LastCommandError

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous TE Get command which is used to Get last command error.

**2.2.2.41 TH****Syntax**

int TH(int controllerAddress, string Axis, out double Position, out string errString)

controllerAddress: Address of Controller

Position: Position

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous TH Get command which is used to Get target position.

**2.2.2.42 TP****Syntax**

int TP(int controllerAddress, string Axis, out double Position, out string errString)

controllerAddress: Address of Controller

Position: Position

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous TP Get command which is used to Get current position.

**2.2.2.43 TS****Syntax**

int TS(int controllerAddress, out string errorCode, out string controllerState, out string errString)

controllerAddress: Address of Controller

errorCode: errorCode

controllerState: controllerState

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous TS Get command which is used to Get positioner error and controller state.

**2.2.2.44 VE****Syntax**

int VE(int controllerAddress, out string Information, out string errString)

controllerAddress: Address of Controller

Information: Information

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous VE Get command which is used to Get controller revision information.

**2.2.2.45 XR\_Get****Syntax**

int XR\_Get(int controllerAddress, string Axis, out double Step, out string errString)

controllerAddress: Address of Controller

Step: Step

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous XR Get command which is used to Move stepping.

**2.2.2.46 XR\_Set****Syntax**

int XR\_Set(int controllerAddress, string Axis, double Step, out string errString)

controllerAddress: Address of Controller

Step: Step

errString: The failure reason

Return: 0 in success and -1 on failure

**Description**

This function is used to process synchronous XR Set command which is used to Move stepping.

**2.2.2.47 XU\_Get****Syntax**

int XU\_Get(int controllerAddress, string Axis, out double StepSize, out string errString)

controllerAddress: Address of Controller

StepSize: StepSize

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous XU Get command which is used to Get step size for STEPPING OL state.

#### **2.2.2.48 XU\_Set**

##### **Syntax**

int XU\_Set(int controllerAddress, string Axis, double StepSize, out string errString)

controllerAddress: Address of Controller

StepSize: StepSize

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous XU Set command which is used to Set step size for STEPPING OL state.

#### **2.2.2.49 ZT**

##### **Syntax**

int ZT(int controllerAddress, out List<string> Parameters, out string errString)

controllerAddress: Address of Controller

Parameters: Parameters

errString: The failure reason

Return: 0 in success and -1 on failure

#### **Description**

This function is used to process synchronous ZT Get command which is used to Get all controller parameters.

# Service Form

## Your Local Representative

Tel.: \_\_\_\_\_

Fax: \_\_\_\_\_

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

Country: \_\_\_\_\_

P.O. Number: \_\_\_\_\_

Item(s) Being Returned: \_\_\_\_\_

Model#: \_\_\_\_\_

Return authorization #: \_\_\_\_\_

*(Please obtain prior to return of item)*

Date: \_\_\_\_\_

Phone Number: \_\_\_\_\_

Fax Number: \_\_\_\_\_

Serial #: \_\_\_\_\_

Description: \_\_\_\_\_

Reasons of return of goods (please list any specific problems): \_\_\_\_\_

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



**Newport®**  
Experience | Solutions

Visit Newport Online at:  
**[www.newport.com](http://www.newport.com)**

**North America & Asia**

Newport Corporation  
1791 Deere Ave.  
Irvine, CA 92606, USA

**Sales**

Tel.: (800) 222-6440  
e-mail: [sales@newport.com](mailto:sales@newport.com)

**Technical Support**

Tel.: (800) 222-6440  
e-mail: [tech@newport.com](mailto:tech@newport.com)

**Service, RMAs & Returns**

Tel.: (800) 222-6440  
e-mail: [rma.service@newport.com](mailto:rma.service@newport.com)

**Europe**

MICRO-CONTROLE Spectra-Physics S.A.S  
9, rue du Bois Sauvage  
91055 Évry CEDEX  
France

**Sales**

Tel.: +33 (0)1.60.91.68.68  
e-mail: [france@newport-fr.com](mailto:france@newport-fr.com)

**Technical Support**

e-mail: [tech\\_europe@newport.com](mailto:tech_europe@newport.com)

**Service & Returns**

Tel.: +33 (0)2.38.40.51.55