

# MotionObjects ActiveX Control

The MotionObjects ActiveX Control provides software for developing motion control applications using Visual Basic. The MotionObjects ActiveX Control encapsulates a single Soft Motion Axis, and incorporates much of the MotionObjects Class Library for use in an ActiveX container with ORMEC's Soft Motion drives.

### Why MotionObjects for Visual Basic?

Developing motion control applications in Visual Basic is easier than ever using the MotionObjects ActiveX Control.

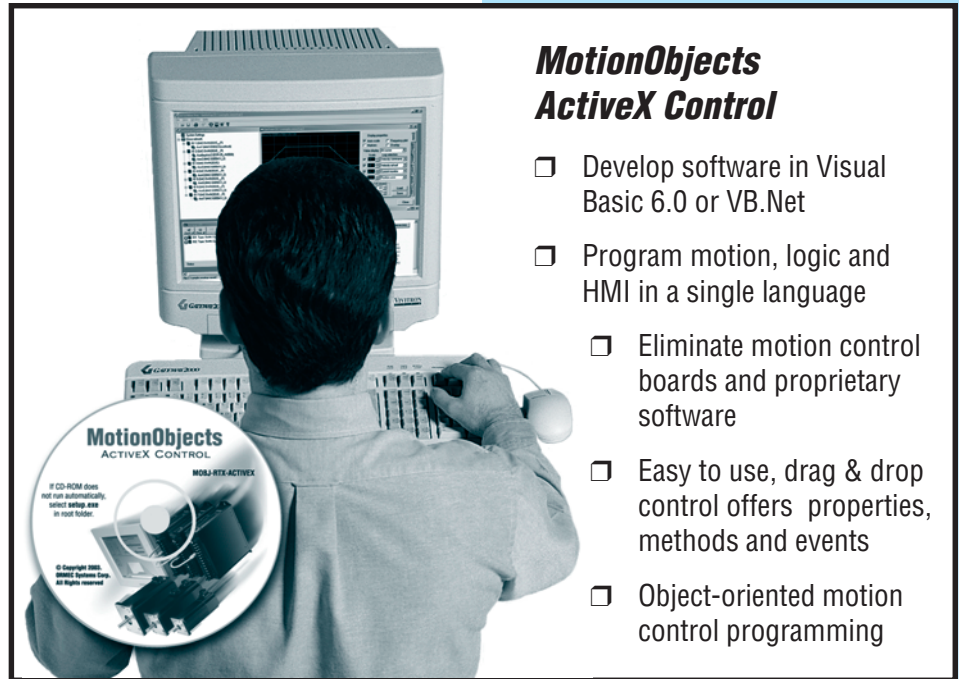
- Develop software in Visual Basic 6.0 or VB.Net using Microsoft Visual Studio, *while preserving the base of application programming you've already developed.*
- Utilize object-oriented motion programming methods to increase software effectiveness and speed up the development process.
- Use a motion control software suite that runs on standard Windows PCs, but doesn't require per-machine motion software royalties.
- Eliminate the need to utilize motion control boards and learn to program in a proprietary motion control language.

### MotionObjects ActiveX Control

The MotionObjects ActiveX Control is added as a component and becomes an extension of your Visual Basic Toolbox. The ActiveX control enables the software developer to easily add axes to their application program by dragging an axis control onto the form for each axis.

ServoWire® Pro is an invaluable tool for generating drive setup and configuration information. ServoWire® Pro provides a graphical user interface that leads the developer through all the appropriate selections required to fully specify axis objects.

Once axes are created, pre-



### MotionObjects ActiveX Control

- Develop software in Visual Basic 6.0 or VB.Net
- Program motion, logic and HMI in a single language
- Eliminate motion control boards and proprietary software
- Easy to use, drag & drop control offers properties, methods and events
- Object-oriented motion control programming

programmed methods in the ActiveX control can be used to generate motions, collect feedback and system information, configure system parameters and handle exceptions.

### MotionObjects Features

**MotionAxis Class:** The MotionAxis Class creates objects capable of generating motion, and provides *methods* to create motion, *properties* to retrieve and/or set axis parameters and *events* to conveniently indicate state changes.

**Move Classes:** The hierarchy of Move Classes provides member functions that are common to all time-based trapezoidal motions, and can be utilized by all of the Axis objects. A listing of the available Move classes is given below:

- MoveAtVel:** Move the axis at a velocity.
- MoveForDist:** Move the axis for a distance at maximum axis speed.
- MoveForDistInTime:** Move the axis for a distance in the given time.
- MoveForDistAtSpeed:** Move the axis for a distance using the given

speed.

- MoveToPos:** Move axis to a position at the maximum axis speed.
- MoveToPosInTime:** Move axis to a position in the given time.
- MoveToPosAtSpeed:** Move axis to position using the given speed.

Each member of the Move class offers a number of syntaxes for building move commands and pre-programmed software functions that make it simple to control motion performance and gather system information.

**Gear Classes:** The Gear Classes provide functions for electronic gearing of multiple servos and/or pacer axes. MotionObjects allows you to configure your machine drive system to gear multiple servomotors together -- allowing servo axes to be linked to common position information. Each electronic lineshaft (or sub-lineshaft) can be driven by commanded position (software motion) or actual position (hardware feedback) from its pacer axis.

# MotionObjects ActiveX Control

## Sample Program

This introductory program, and the accompanying **flowchart**, illustrates the essential stages in a MotionObjects application.

```

Private Sub Check1_Click()
  If Check1.Value = vbChecked Then
    ' Enable the drive in position mode.
    MyAxis.LoopMode = OrmLoopModePosition
  Else
    ' Disable the drive (no torque).
    MyAxis.LoopMode = OrmLoopModeDisabled
  End If
End Sub

Private Sub Command1_Click()
  ' Clear faults
  MyAxis.Fault = 0
End Sub

Private Sub Command2_Click()
  'Command the Motion "Move for 360
  user position units in 500 milliseconds.
  MyAxis.MoveForDistInTime 360, 500
End Sub

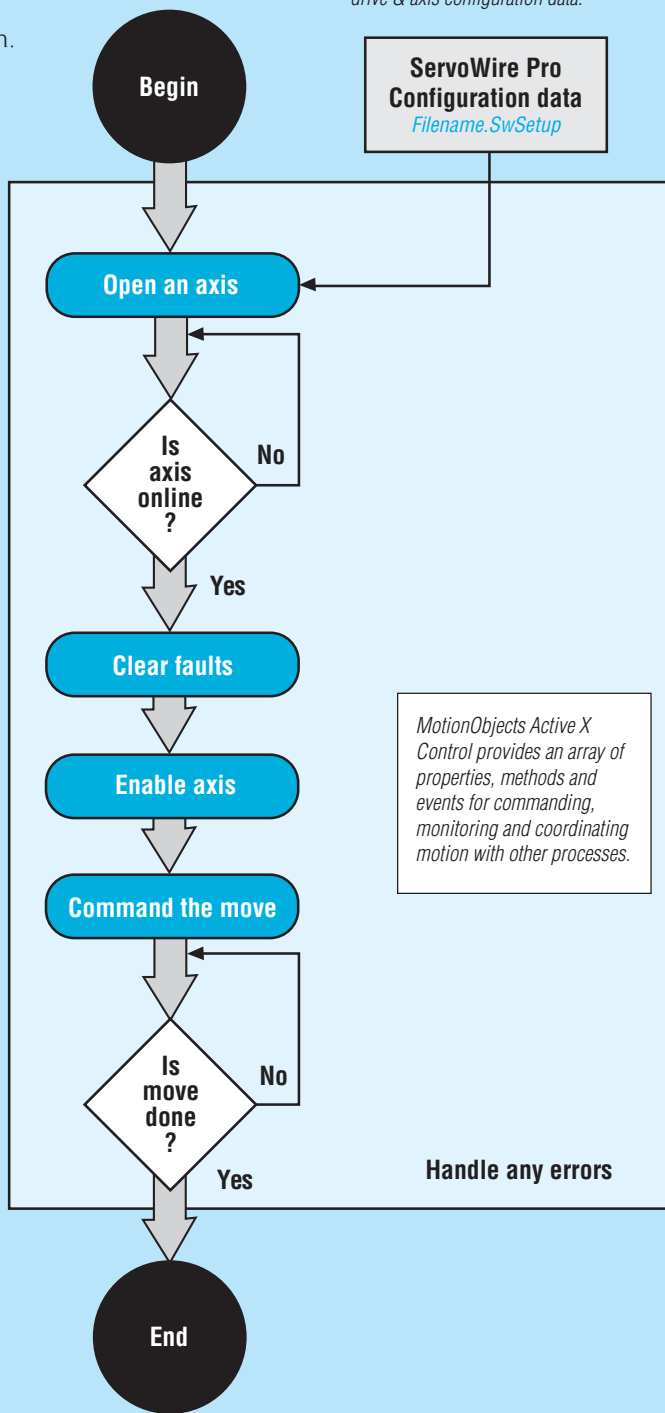
Private Sub Form_Load()
  On error goto errLoad
  'disable the controls until the axis comes online
  Command1.Enabled = False
  Command2.Enabled = False
  Check1.Enabled = False
  ' Open "Axis1" from the setup file
  MyAxis.Open "c:\Sample.SwSetup", "Axis1"
  Do
    DoEvents
  Loop Until MyAxis.IsOnline
  Exit Sub
errLoad:
  MsgBox "Error in Form_Load: " + MyAxis.ErrorDescription
End Sub

Private Sub Form_Unload(Cancel As Integer)
  'Always Close any Open axes before stopping Visual Basic
  MyAxis.Close
End Sub

Private Sub MyAxis_Online()
  'when the axis comes online then enable the controls
  Command1.Enabled = True
  Command2.Enabled = True
  Check1.Enabled = True
End Sub

```

ServoWire Pro provides an effective, convenient way to automatically generate drive & axis configuration data.



**ServoWire Pro Configuration data**  
Filename.SwSetup

MotionObjects Active X Control provides an array of properties, methods and events for commanding, monitoring and coordinating motion with other processes.

Handle any errors

- GearAtRatio:** Gear follower to pacer axis.
- GearForDistAtRatio:** Gear follower for specific distance in synchronism with pacer by

- specifying a target gear ratio.
- GearForDistInPDist:** Gear follower a specific distance from current position in synchronism with pacer.