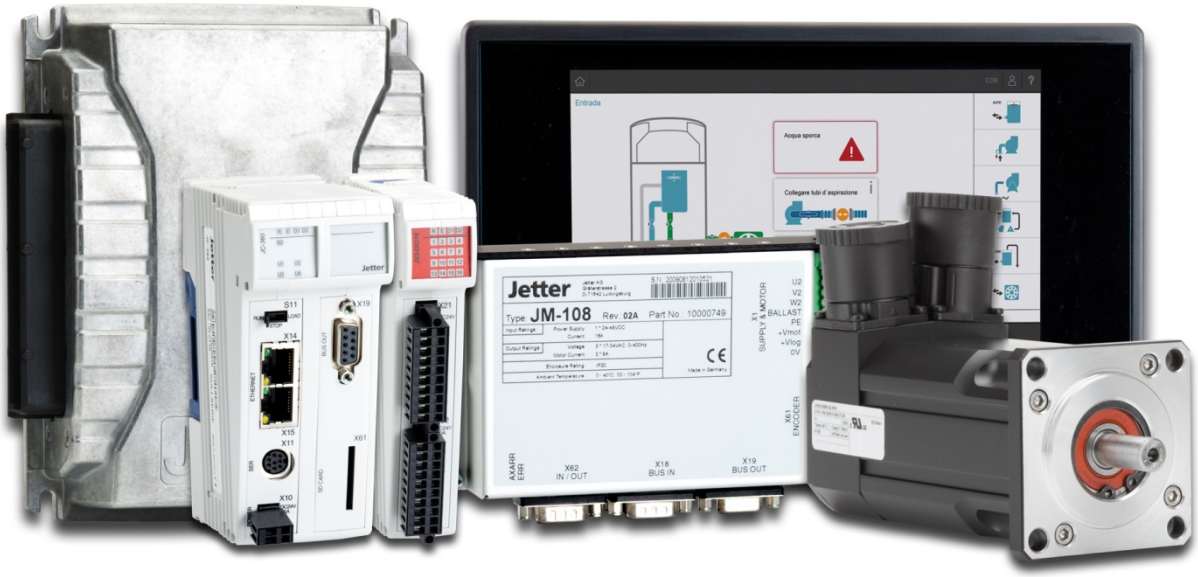
Jetter automation



MCX – Motion Control Basics

Application Note 054

608 860 66_00

We automate your success.

This document has been compiled by Jetter AG with due diligence based on the state of the art as known to them. Any revisions and technical advancements of our products are not automatically made available in a revised document. Jetter AG shall not be liable for any errors either in form or content, or for any missing updates, as well as for any damage or detriment resulting from such failure.

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg
Germany

www.jetter.de

Phone:

Switchboard	+49 7141 2550-0
Sales	+49 7141 2550-531
Technical hotline	+49 7141 2550-444

E-mail:	info@jetter.de
Technical hotline	hotline@jetter.de
Sales	vertrieb@jetter.de

Product name	MCX – Motion Control Basics
Document type	Application Note 054

Translation of the original German language document

Document revision	1.00
Item number	608 860 66_00
Date of issue	2021-05-26

Table of Contents

1 MCX vs. Motion Control

Motion Control is a software that calculates the movement of an axis.
In most cases the MC software runs on the servo amplifier. In single-axis mode, the positions of the axis movement are calculated.
MCX, on the other hand, is a centralized motion controller. The individual axes are connected via fieldbus and synchronized with each other. This ensures that all axes receive a new setpoint at the same time.
Furthermore, MCX is able to move individual axes in coordination with each other to form, for example, an electrical gearbox, cam disc, or path group.

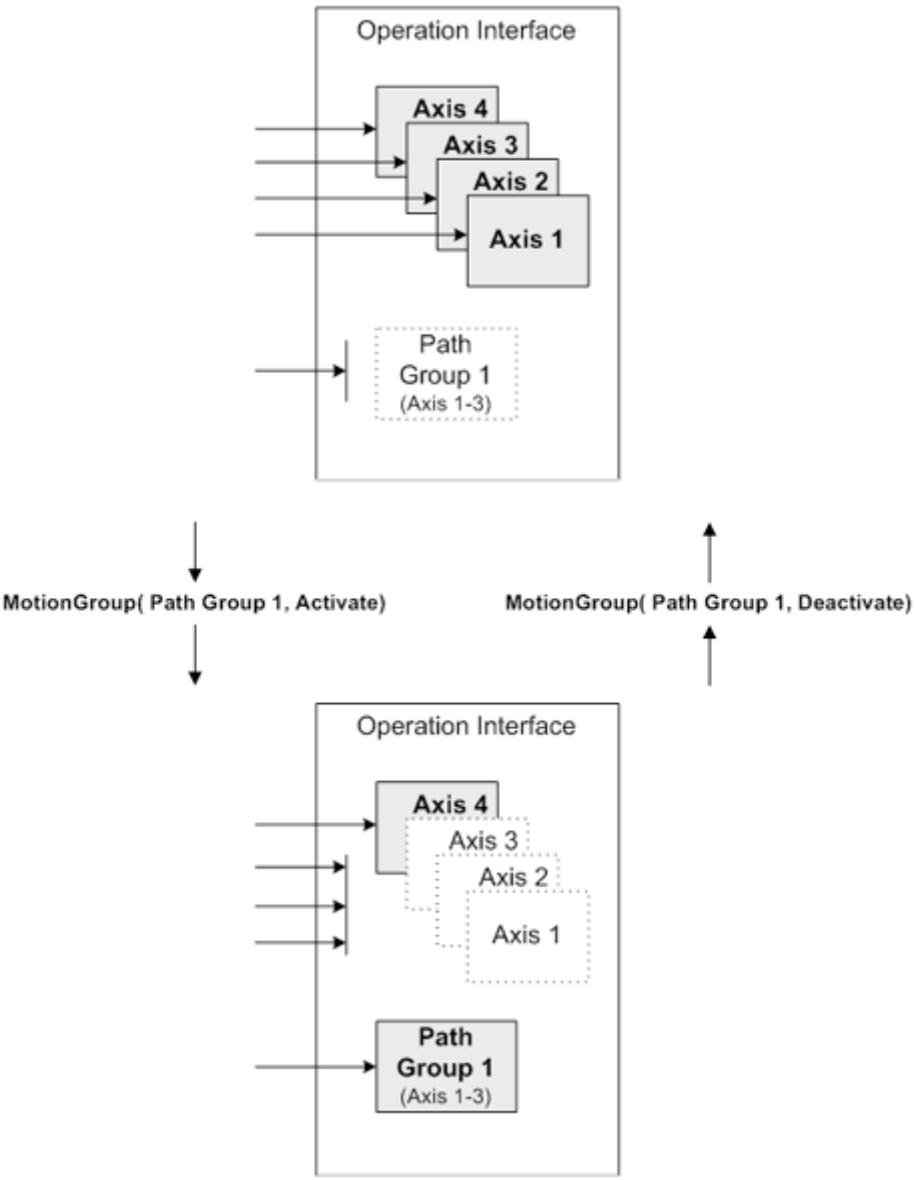
1.1 Object Model

In the Motion Control software, individual axes or axes groups are regarded as objects. At first, these objects are created in the Hardware Manager. The hardware configuration is set here, too. Possible objects are:

- **Axes -** are usually linear or rotary single axes driven by a JetMove. They are used as normal or modulo axes. They can also be designed as simulation axes if no hardware is connected yet. However, they can also be purely virtual axes.
- **Path Groups -** A path group is a combination of previously configured single axes performing a path motion. The involved axes, in most cases, form a kinematic chain with a tool at the end. Through this tool the axes are mechanically coupled with each other. Examples of path groups are 3D cartesian groups or SCARA robots, etc.
- **Technology Groups -** A technology group is a combination of previously configured single axes involved in a common process sequence. In most cases, there is no mechanical coupling between the axes and they can be split-up into one master axis and several follower axes. Examples of technology groups are electronic gearbox, electronic cam, etc.

Once these objects are **activated**, they can be addressed through the functions integrated into JetSym ST/STX (MotionControl instructions and setup). First, axis groups shall explicitly be activated using the command MotionGroup. Individual axes are implicitly activated if they are not involved in an active axis group. Thus, once an axis group has been activated, the involved axes can no longer be addressed as individual axes.

The following illustration shows how axes and axis groups can be addressed before and after their activation:



Note: For an inactive object, only the commands for activation and diagnostics are allowed. The command to clearing errors is also allowed for an inactive object, but it does not necessarily need to be issued.

Thanks to the approach to treat axis groups as individual objects programming of coordinated motions has been made easier and more systematic. Programming of a path group is carried out independent of the actual axis configuration in a user-specific coordinate system that is fixed in space. If, for example, a technology group is stopped by means of the corresponding function, all axes involved automatically come to a stop.

If, due to technological reasons, individual axes within a group are to show a special behavior (e.g. decoupling a flying saw, changing controller parameters during movement), a function is provided for this within the axis group.

In the **state-oriented object model** used in the MCX software, active objects are always in a defined operating state (see also State Management). Depending on the operating

state, objects can be programmed, parameterized and diagnosed via several operating channels.

The state-oriented approach has the following benefits:

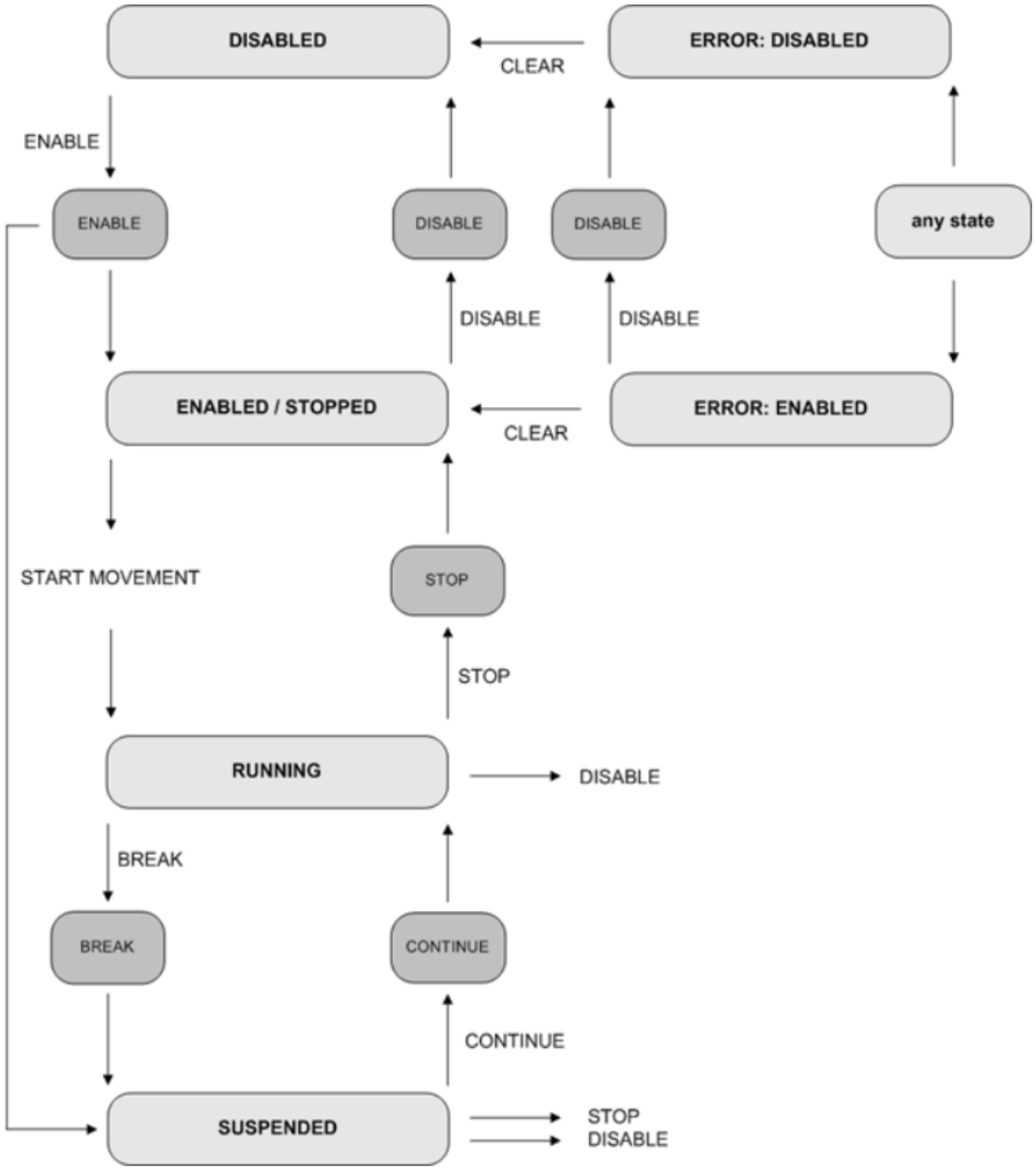
- Senseless function calls or programming errors will be detected and ignored by the system.
- Illegal function calls are reported to the user.
- The system is less prone to errors.
- The structured procedure assists programming and commissioning of axes/axis groups.

1.2 State Management

Usually, objects used in the Motion Control software are directly instructed via Motion Control commands integrated into JetSym ST/STX. This way, operating state transitions are automatically triggered.

For direct state management - e.g. for responding in case of an error - the Motion API command <Object>.State.Transitions is provided in JetSym STX. Explicit operating state transitions can be triggered using this command, as well as by the command <Object>.Diagnostics.ClearErrors().

The following illustration shows the possible operating state transitions of an object:



Notes:

An object (axis groups only after activation) can always assume only **one** defined operating state. If, for example, the object is in the operating state *Running*, the operating state *Enabled* is not active (although the axes have been enabled, of course).

State.Transitions commands always take effect **immediately** even in buffered mode. All look-ahead buffers are emptied.

In detail, the operating states are characterized as follows:

DISABLED (State.IsDisabled):

Axis: The control loops of the axis drive are open.

Technology/path group: The control loops of all drives belonging to the group are open.

ENABLED/STOPPED (State.IsEnabled):

Axis: The control loops of the axis drive are closed. The axis interpolator is standing still. All look-ahead buffers of the function channel are empty.

Technology group: The control loops of all drives belonging to the group are closed. The axis interpolators of all axes belonging to the group are standing still.

Path group: The control loops of all drives belonging to the group are closed. The path interpolator is standing. All look-ahead buffers of the function channel are empty.

RUNNING (State.IsRunning):

Axis: The setpoint generator is running. In the buffered operating mode, the look-ahead buffers of the function channel are processed and can continue to be filled.

Technology group: The axis interpolator of at least one axis belonging to the group is running.

Path group: The path interpolator is running. The look-ahead buffers of the function channel are processed and can continue to be filled.

SUSPENDED (State.IsSuspended):

This is the same operating state as **ENABLED**, except that all look-ahead buffers remain filled, but are locked. A suspended motion can be resumed using the State.Transition command "Resume".

ERROR: DISABLED/ERROR: ENABLED:

Error conditions resulting from internal response to errors.

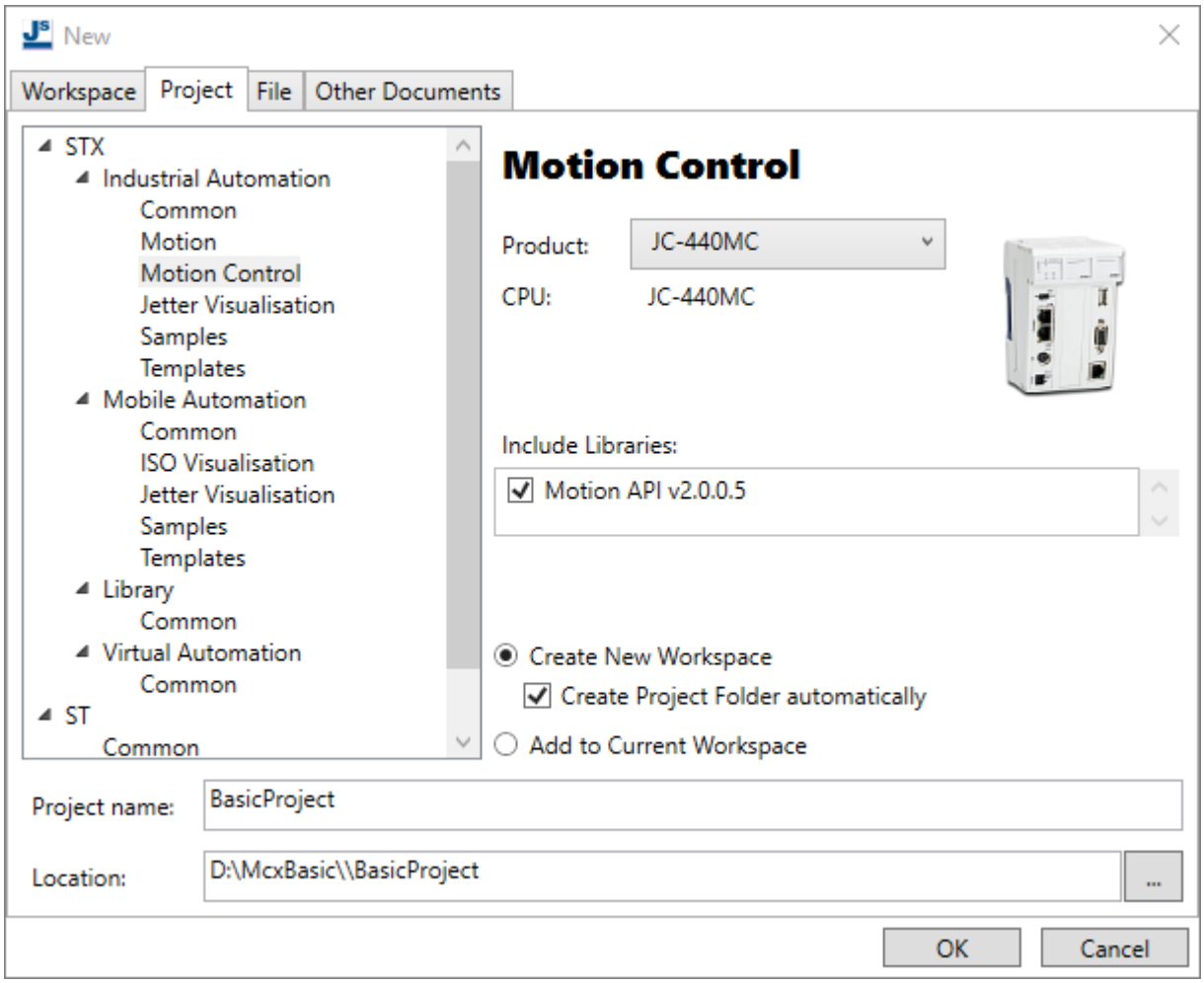
2 Configuring an MCX controller

Prerequisites:

- JetSym 5.6.1
- Right-handed mouse:
  - Open the submenu: Right-click
  - Double-click: Two quick clicks with the left mouse button
  - Selection: One click on the left mouse button

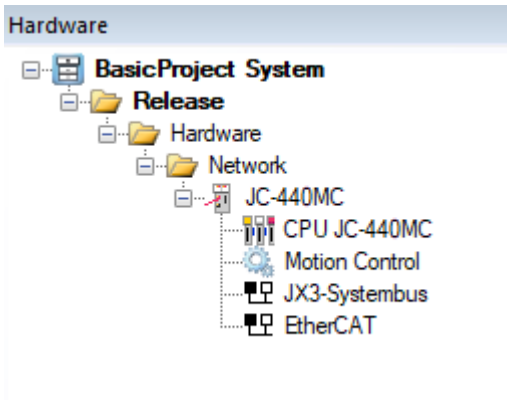
2.1 Creating an MCX project

Open the project wizard in JetSym via "File->New".



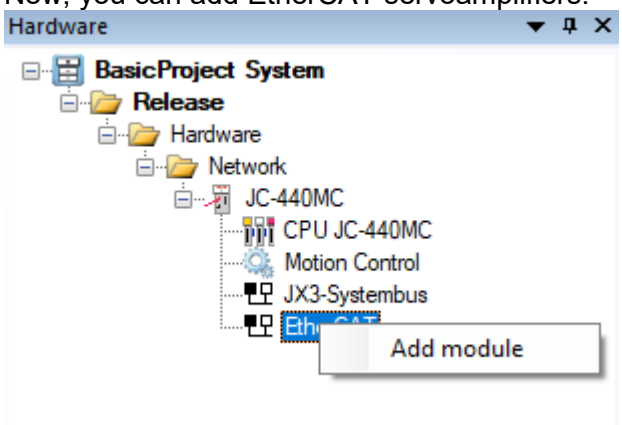
Select an MCX controller and include the Motion API. JetSym always suggests the version currently installed and suitable for the selected controller.
When you click "OK", the project tree will be created.

In the Hardware Manager, the controller and the basic nodes are now created for further configuration.

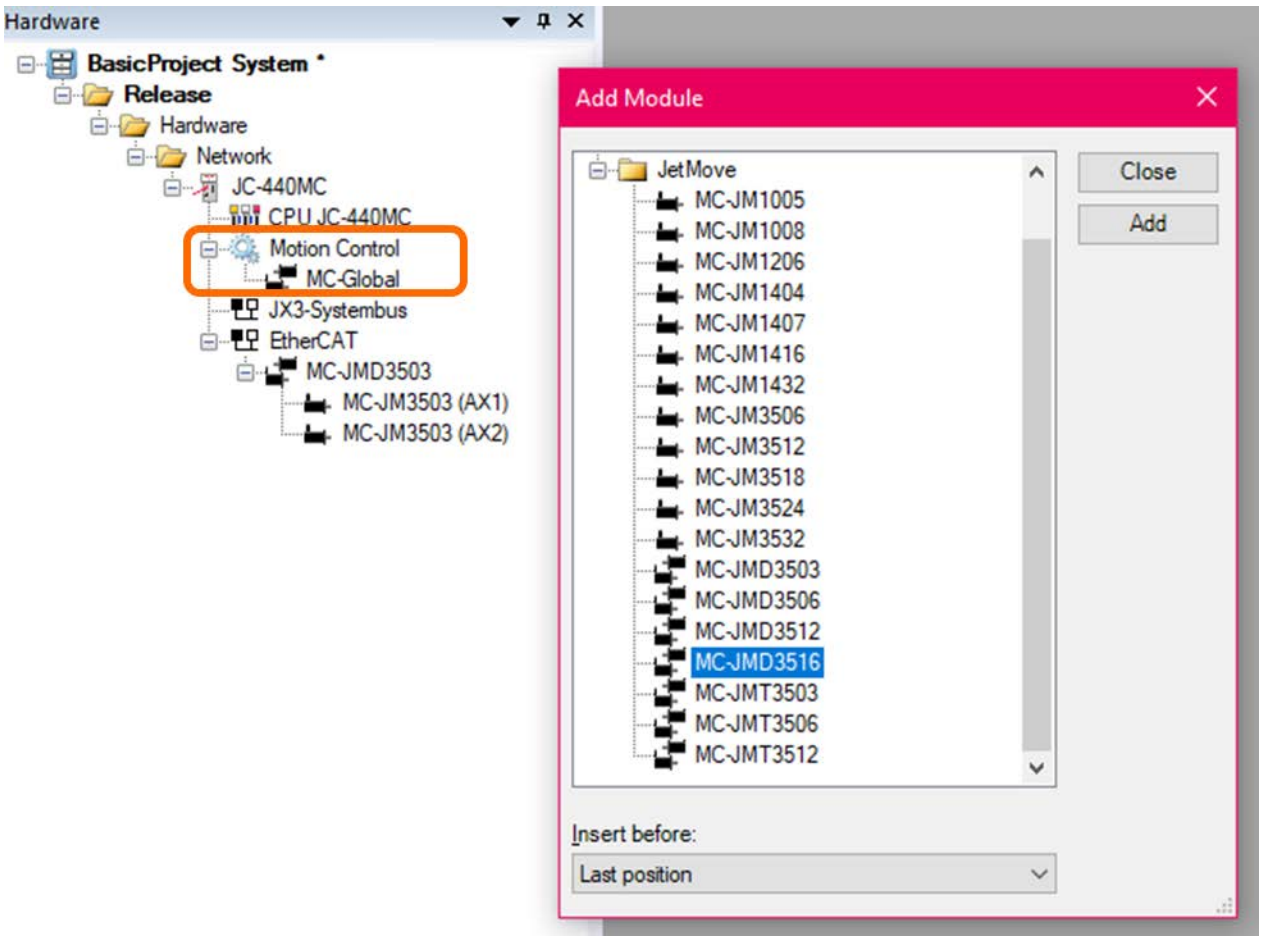


2.1.1 Adding servoamplifiers

Now, you can add EtherCAT servoamplifiers.



The submenu on the "EtherCAT" node lets you select the modules to be added to this node.

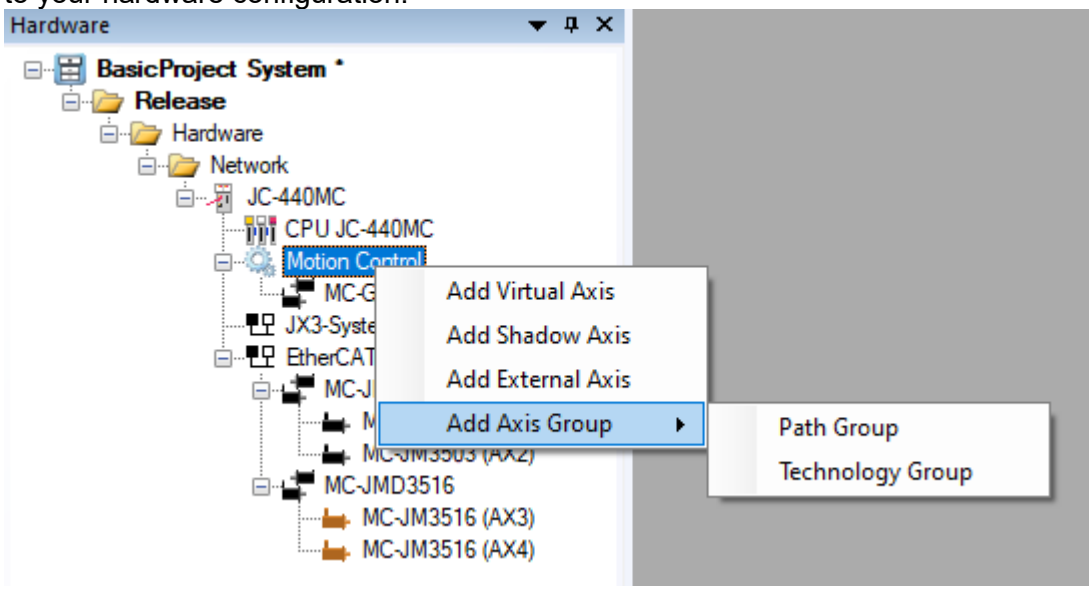


Now select the required servoamplifiers and add them to the hardware configuration by double-clicking or by clicking on "Add".

When an MC axis is created, the "MC-Global" node is also created.

2.1.2 Creating additional MC objects

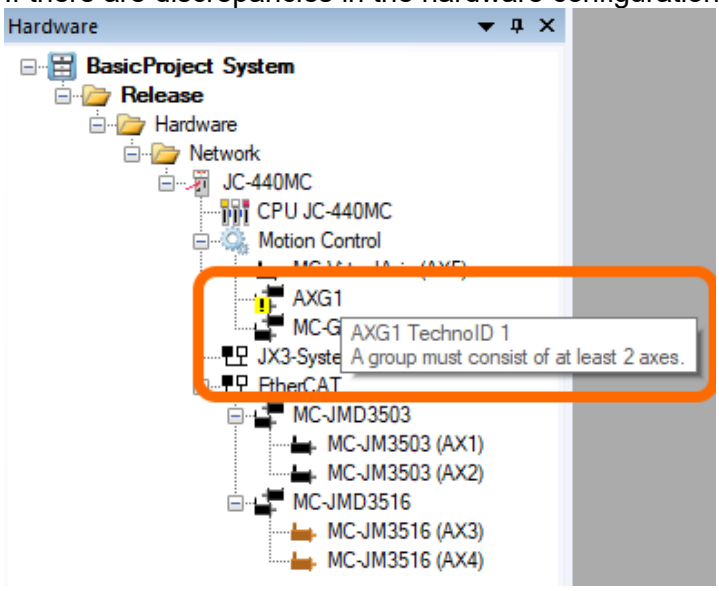
Open the submenu on "Motion Control". You can now select which other objects you want to add to your hardware configuration.



Virtual axis: This axis does not have its own hardware, but the setpoints are calculated as for a real axis.
Shadow axis: This axis inherits the setpoints and actual values from a created axis object.
External axis: Here, the actual values come from a counter module, such as the second encoder input on a JM-3000. This allows friction wheels to be connected for position detection, for example. This axis cannot be actively controlled by the MC.
Path group: This creates a path group for Cartesian or SCARA kinematics.
Technology group: Lets you link follower axes to a master axis to form an electrical gearbox or cam disc.

2.1.3 Incorrect hardware configuration

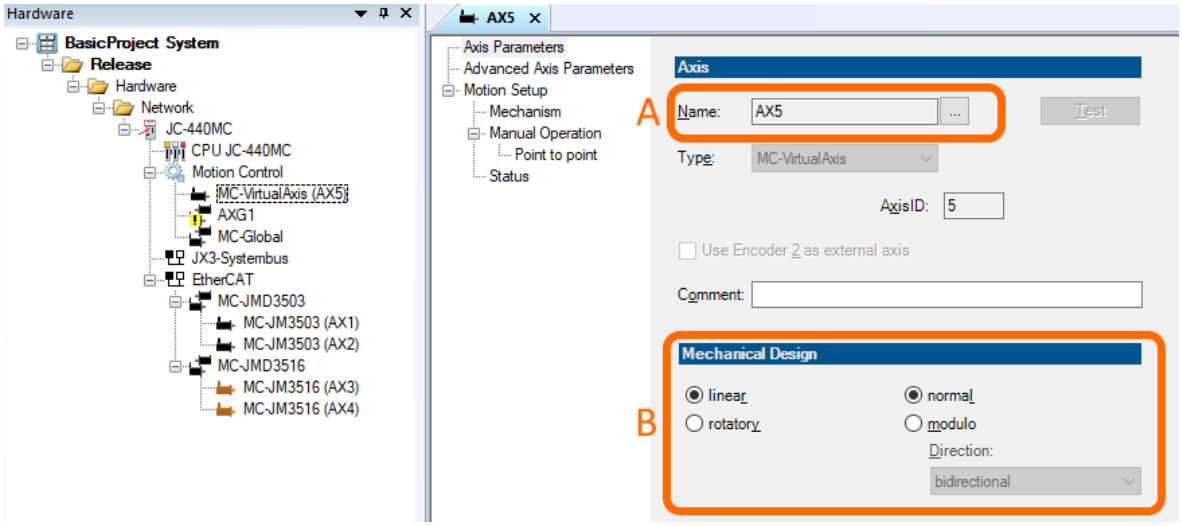
If there are discrepancies in the hardware configuration, an exclamation mark indicates this.



In this example, the technology group "AXG1" has been created, but the configuration of which axes are assigned to the group is still missing.
When you move the cursor over the selected object, JetSym displays a brief explanation.

2.1.4 Configuring axis objects

Double-clicking on an axis object opens the Motion Setup - a configuration and commissioning window.

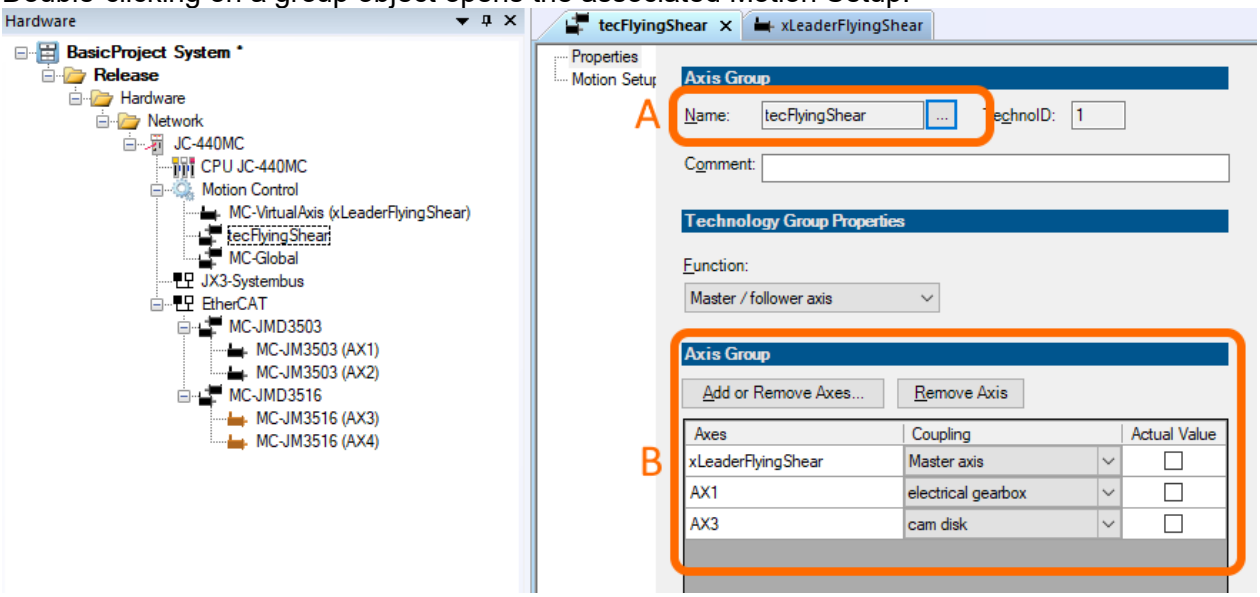


In this example, the Motion Setup of the virtual axis "AX5" is open.
On the Motion Setup start page, the basic settings of the axis object can be defined.

A. It is recommended to assign a descriptive name that makes sense for the application instead of the suggested axis name "AX5". This name is also used to address the respective axis object in the STX program.
B. In the "Mechanical Design", it is determined whether we have a rotary or linear motion here and whether the axis is a modulo axis or a normal axis.
   - Normal axis: The axis has a fixed travel range that cannot be exited.
   - Modulo axis: The position of the axis jumps back to the beginning when the modulo range is exceeded.
   Here, the movement at the working point is usually taken into account.

2.1.5 Configuring axis groups

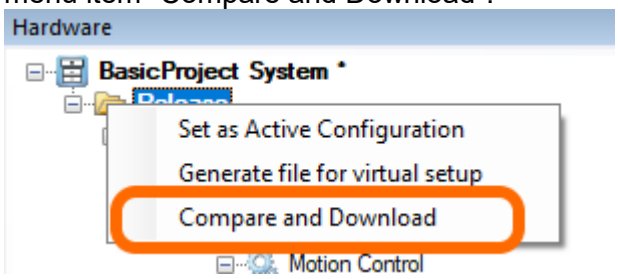
Double-clicking on a group object opens the associated Motion Setup.



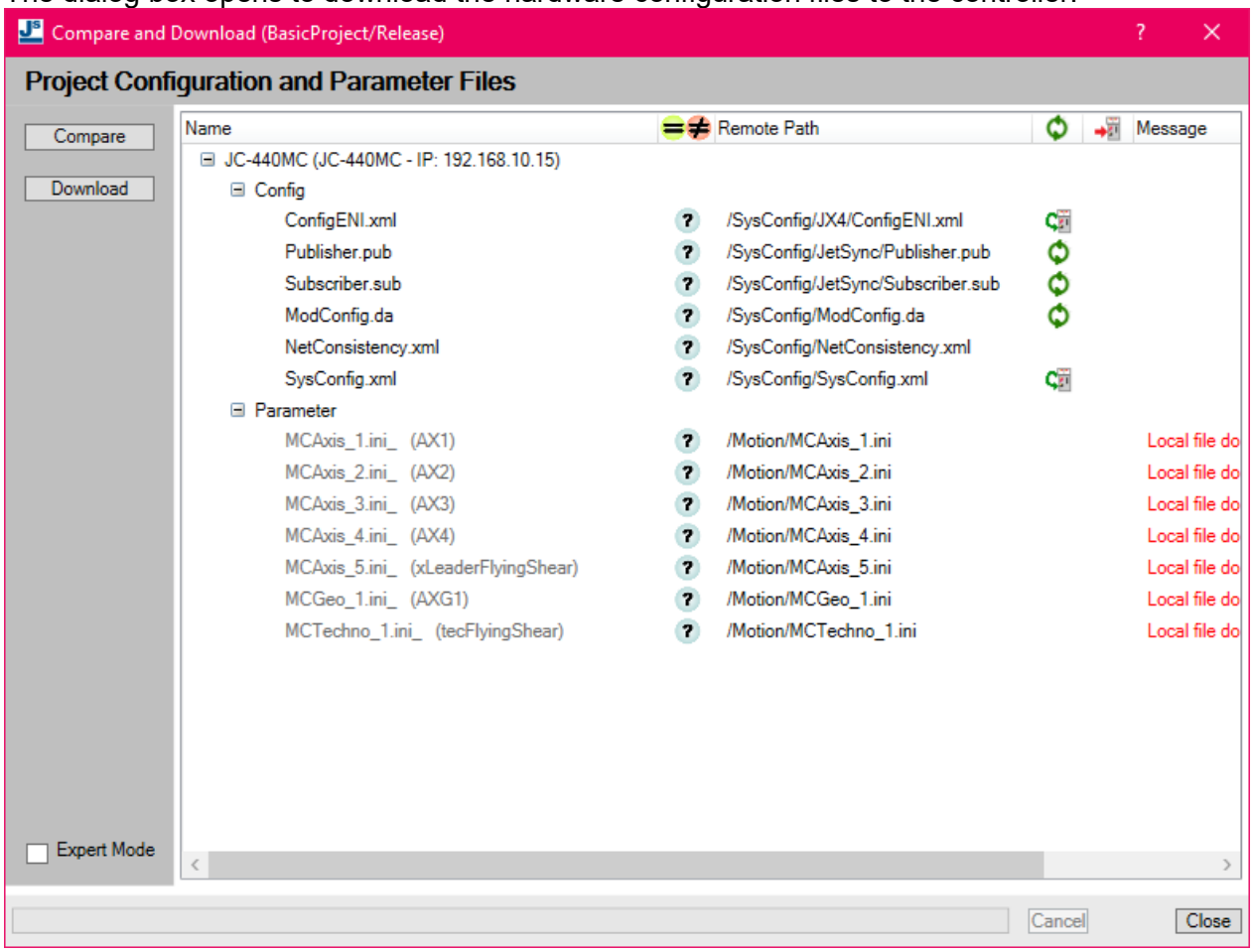
A. It is recommended to assign a descriptive name that makes sense for the application , instead of the suggested group name "AXG1". This name is also used to address the respective group object in the STX program.
B. Here, you can add group members and define their role.
    a. In a technology group, these are:
        i. Master axis
        ii. Coupling of the follower axis via electrical gearbox
        iii. Coupling of the follower axis via cam disc
    b. In a path group, these are:
        i. Cartesian: Axes XYZ
        ii. SCARA: Shoulder, elbow, wrist, Z-axis

2.1.6 Transferring the configuration

In the submenu on the node of the respective JetSym configuration - here "Release" - select the menu item "Compare and Download".

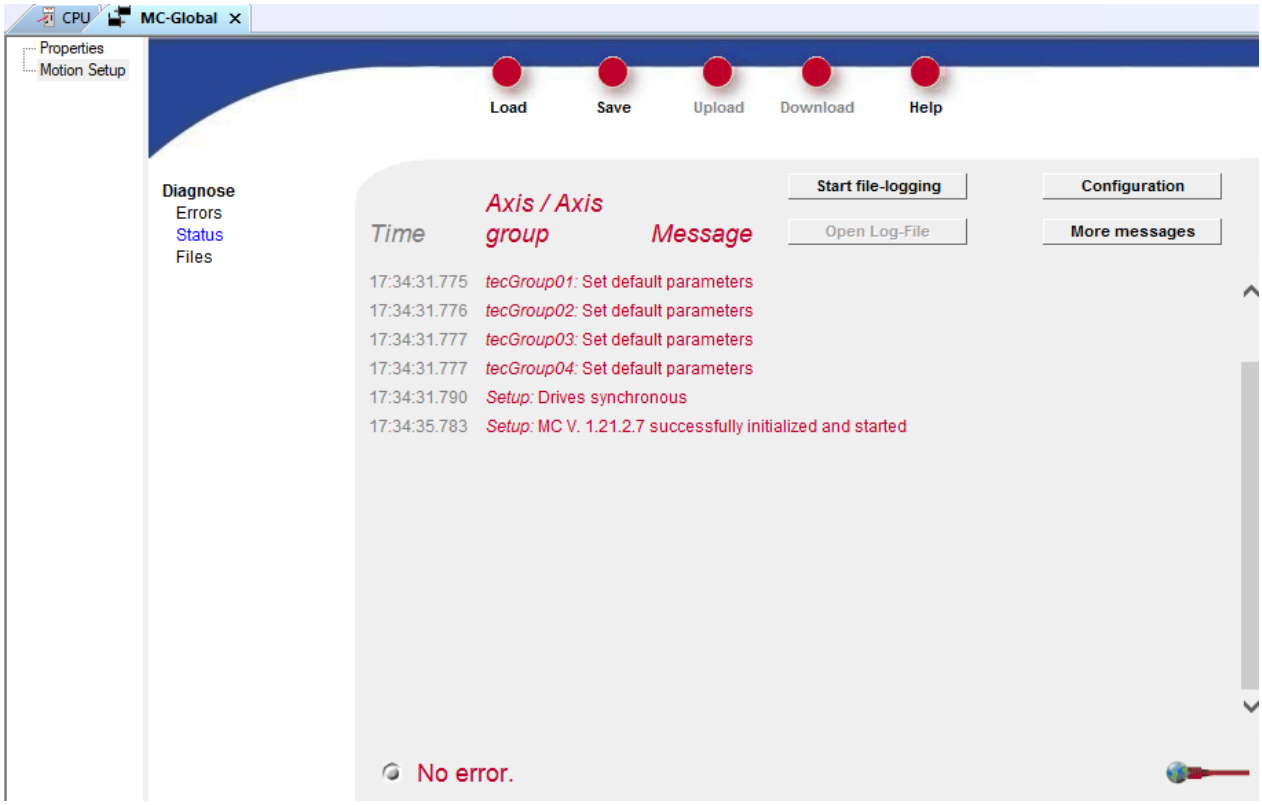


The dialog box opens to download the hardware configuration files to the controller.



If the configuration file for this axis object is available from an axis commissioning, the configuration can also be transferred to the controller using this file.
If you select "Compare", the local files are compared with those on the controller.
When you click the download button, the transfer begins. If the hardware configuration has changed, a restart of the controller is offered.
A new hardware configuration is not activated until a restart is performed.

In the MC-Global Motion Setup (Motion Setup -> Status), you can check after the restart whether the MCX kernel was able to boot successfully with this hardware configuration.

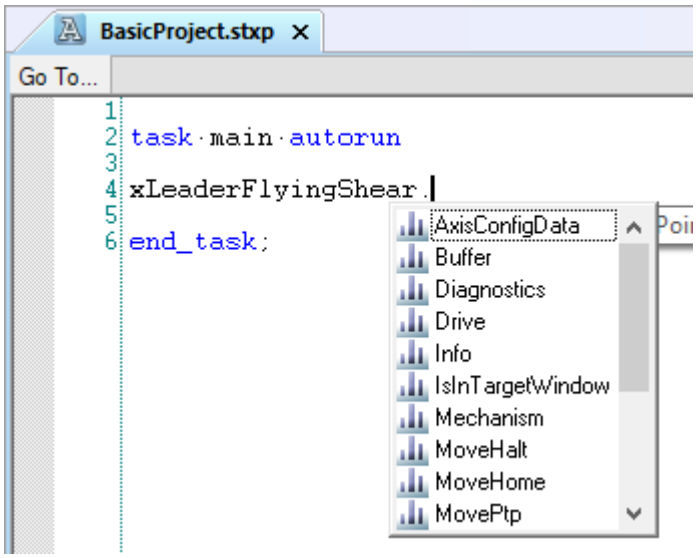

CPU
MC-Global
Properties
Motion Setup
Load
Save
Upload
Download
Help
Diagnose
Errors
Status
Files
Start file-logging
Configuration
Time
Axis / Axis group
Message
Open Log-File
More messages
17:34:31.775 tecGroup01: Set default parameters
17:34:31.776 tecGroup02: Set default parameters
17:34:31.777 tecGroup03: Set default parameters
17:34:31.777 tecGroup04: Set default parameters
17:34:31.790 Setup: Drives synchronous
17:34:35.783 Setup: MC V. 1.21.2.7 successfully initialized and started
No error.

3 Programming the MCX

The Motion API lets you program the MCX in STX. This is an object-oriented library where the available functions and properties are logically organized.
Based on the hardware configuration, the HardwareConfig.stxp is generated in the background. In doing so, the MC objects such as axes, groups, and also an MC Manager for the Motion API are created.
In order for the objects to be available in the program, an error-free compilation process is necessary. Therefore, a default task can be helpful to compile without errors.

```
task main autorun
end_task;
```

Now that all objects are created for the Motion API, the IntelliSense assists you in entering commands.



All available functions of the respective object are mapped in the Motion API.

3.1 Sample program

Here is a short example program to show what the programming can look like and what elements are available.

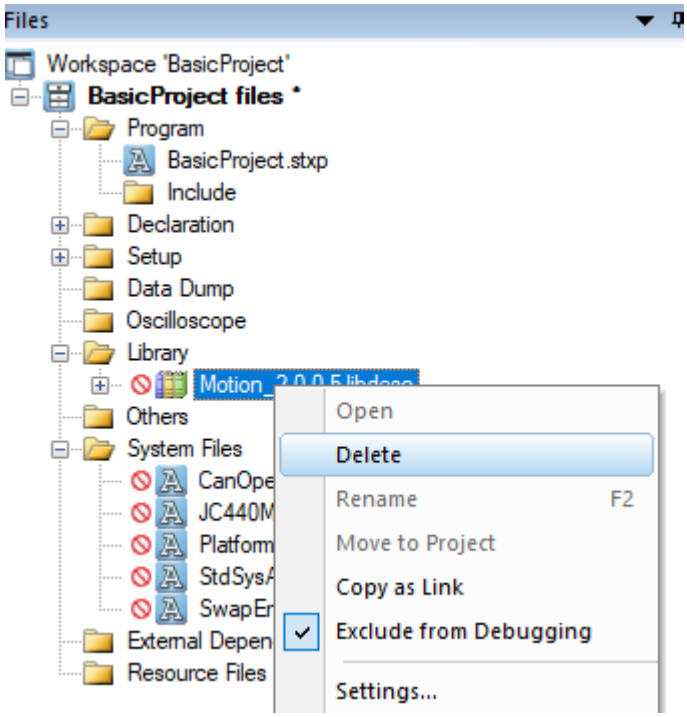
```
task main autorun
        xLeaderFlyingShear.MoveHome.SetReference(0.0);                          // A
        xLeaderFlyingShear.Power.Enable();                                      // A
        when xLeaderFlyingShear.State.IsEnabled continue;                       // B
        tecFlyingShear.Activate();                                              // C
        when tecFlyingShear.State.IsEnabled continue;                   // B
        tecFlyingShear.MoveVelocity.Start(xLeaderFlyingShear);          // D
        when tecFlyingShear.State.IsRunning continue;                   // B
        when tecFlyingShear.Mechanism.Slope.IsAtConstantSpeed continue; // B1
        tecFlyingShear.MoveHalt.Start(xLeaderFlyingShear, MCTechnoHaltModes.Normal);// D
        when tecFlyingShear.State.IsEnabled continue;                   // B
        tecFlyingShear.Deactivate();                                            // C
        when tecFlyingShear.State.IsInactive continue;                  // B
        xLeaderFlyingShear.Power.Disable(MCPowerDisableModes.Forced); // A
        when xLeaderFlyingShear.State.IsDisabled continue;                      // B
end_task;
```

A. Single-axis commands, such as setting the home position, enabling, disabling
B. Status queries of the MC objects: Either the status of the single axis or the group
   B1: Queries for special statuses, such as ramp status
C. Activation and deactivation of a group
D. Group commands, such as positioning and stopping group members

3.2 Selecting the Motion API version

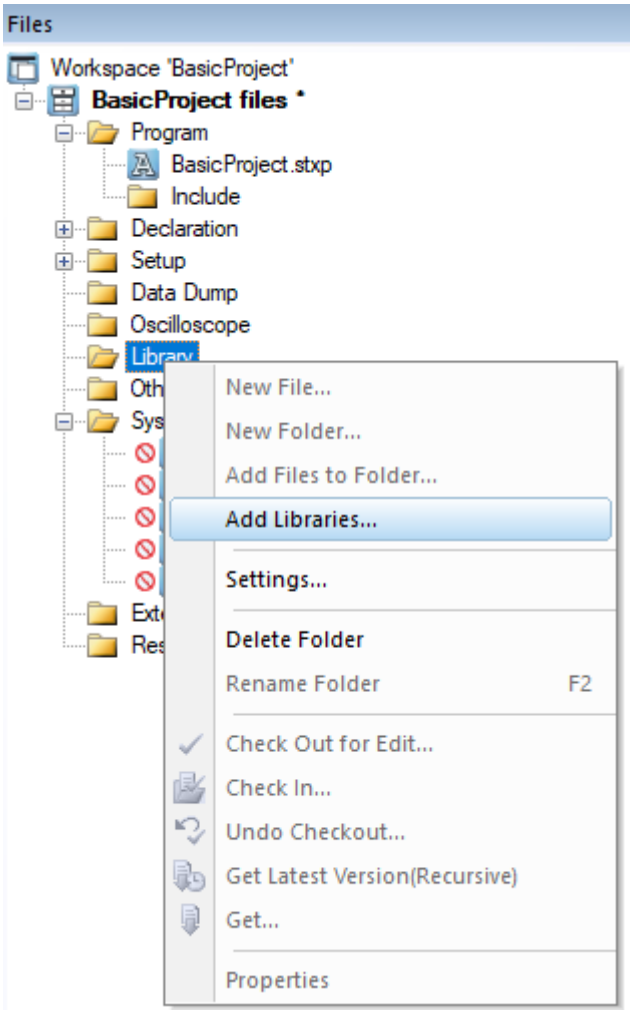
When you create a new project, JetSym always suggests the currently installed version. However, if a specific version is to be used or if a newer version is to be integrated in an existing project, the following steps are necessary.

1. Delete the existing Motion API from the project tree.

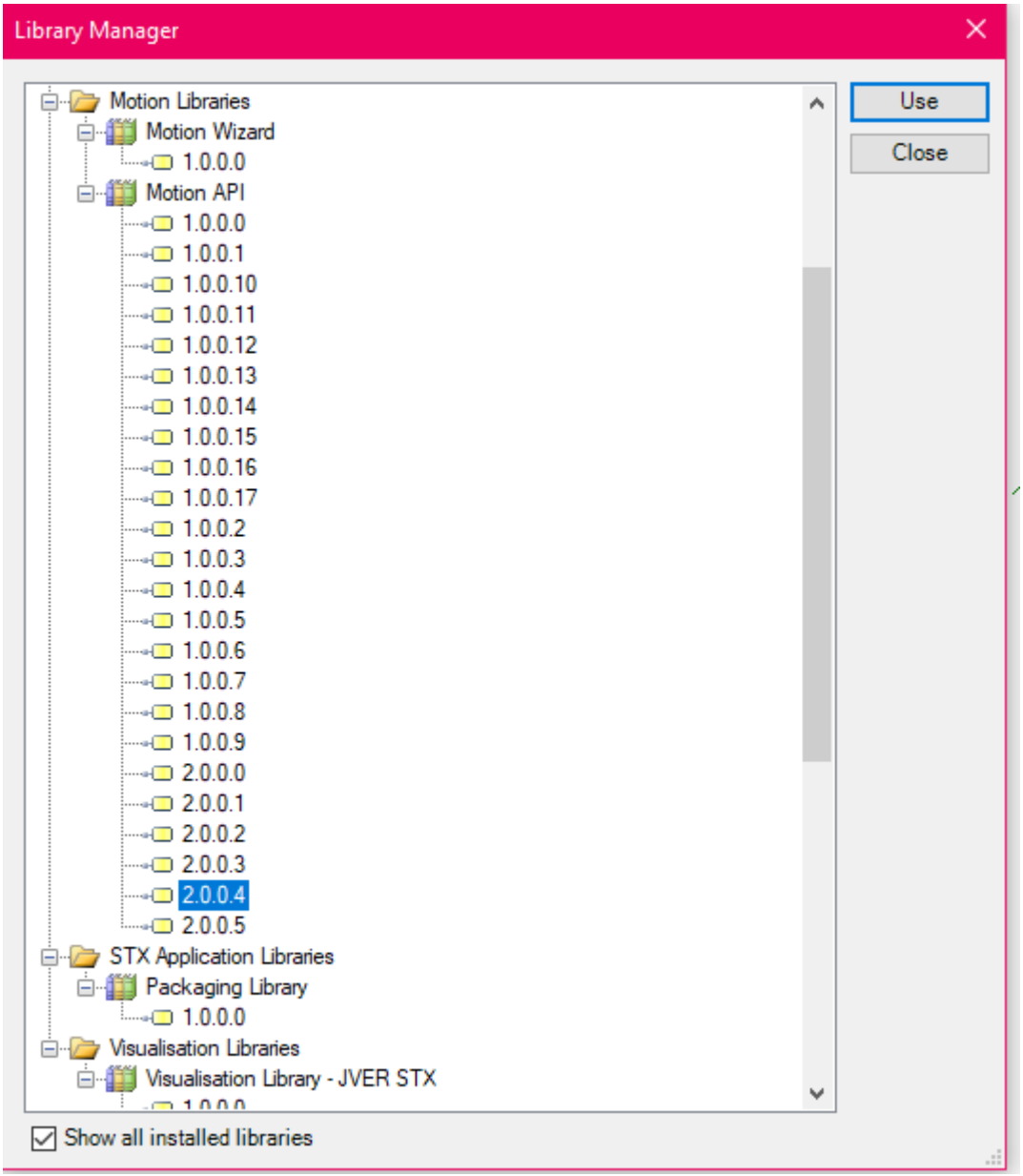


Only one Motion API can be integrated in a project at a time. Therefore, if you do not remove the included version, no other version can be included.

2. Open the library selection via the "Library" sub-menu in the project tree.



3. Select the desired Motion API version.

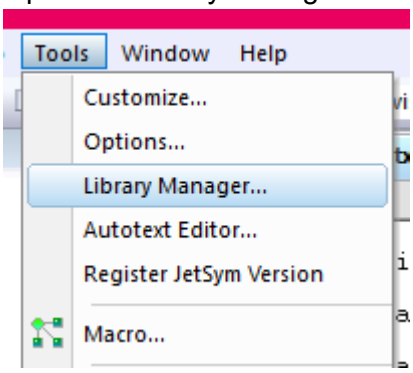


If necessary, "Show all installed libraries" must be activated.
By clicking on "Use", the selected Motion API is now integrated.

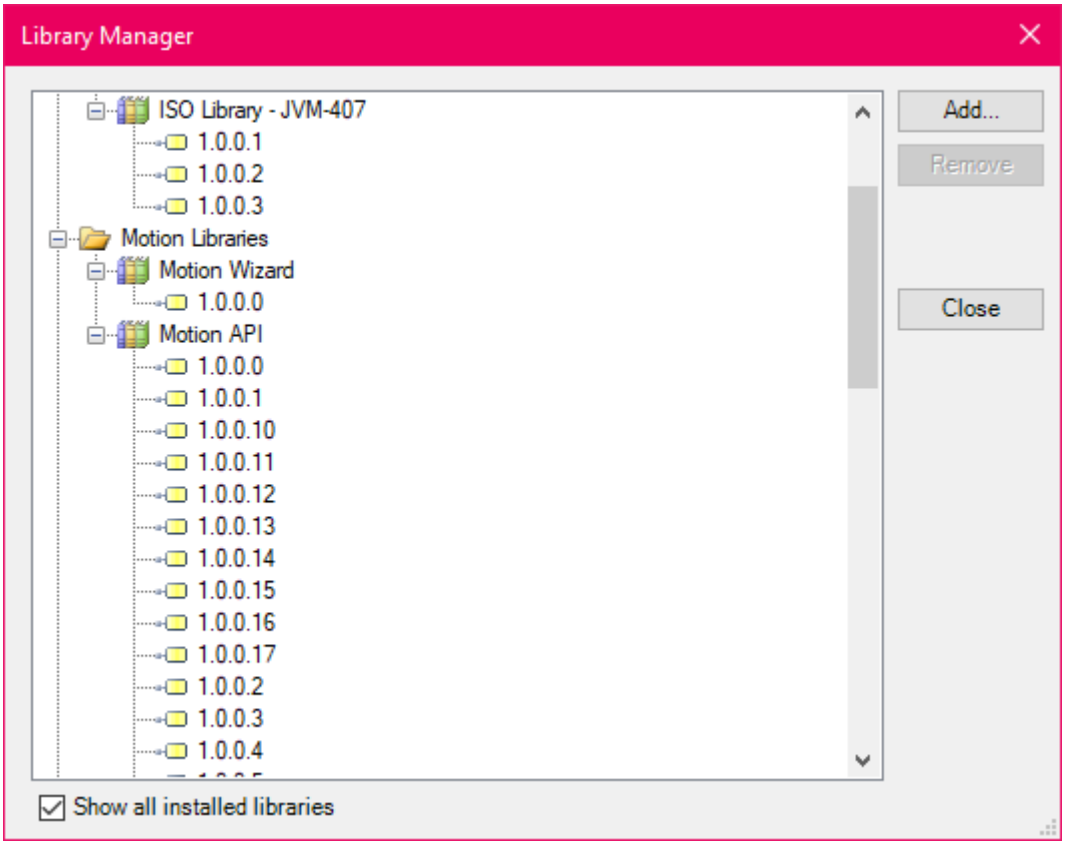
3.2.1 Adding Motion APIs

Only Motion API libraries that were previously added to the Library Manager can be included in a project.

1. Open the Library Manager.



2. The Library Manager opens and shows the installed libraries.



3. Click on "Add" to open the "Open file" dialog. Select the "libpackage" which should be added.
4. If necessary, the option "Show all installed libraries" must be activated so that the libraries are displayed.
   Now you can include the additional library in your project as described under "Selecting the Motion API version".

Jetter automation

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg
Germany
www.jetter.de

E-mail info@jetter.de
Phone +49 7141 2550-0

We automate your success.