

MCX – 运动控制基础

应用笔记 054

608 860 66_00

We automate your success.

本文件是由 Jetter AG 基于已知的技术慎重地编制而成。产品的变更和技术开发未全部包含在修订文件中。Jetter AG 对内容和形式上的错误、缺少更新以及由此产生的损害或不利情况不承担任何法律责任。

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

目录

1	MCX 与运动控制.....	1
1.1	对象模型	1
1.2	状态管理	3
2	配置 MCX 控制器.....	6
2.1	创建 MCX 项目	6
2.1.1	添加伺服驱动器.....	7
2.1.2	创建额外的 MC 对象	9
2.1.3	错误的硬件配置.....	10
2.1.4	配置轴对象.....	10
2.1.5	配置轴组	11
2.1.6	传输配置	12
3	MCX 编程	14
3.1	示例程序	15
3.2	选择 Motion API 版本	15
3.2.1	添加 Motion API	18

1 MCX 与运动控制

Motion Control 是一种计算轴运动的软件。

在大多数情况下，MC 软件在伺服驱动器上运行。在单轴模式下，MC 软件计算轴运动的位置。

另一方面，MCX 是一个集中式运动控制器。各个轴通过现场总线连接并相互同步。这可确保所有轴同时接收新的设定值。

此外，MCX 能够相互协调地移动各个轴，以形成电子齿轮、凸轮盘或路径组等。

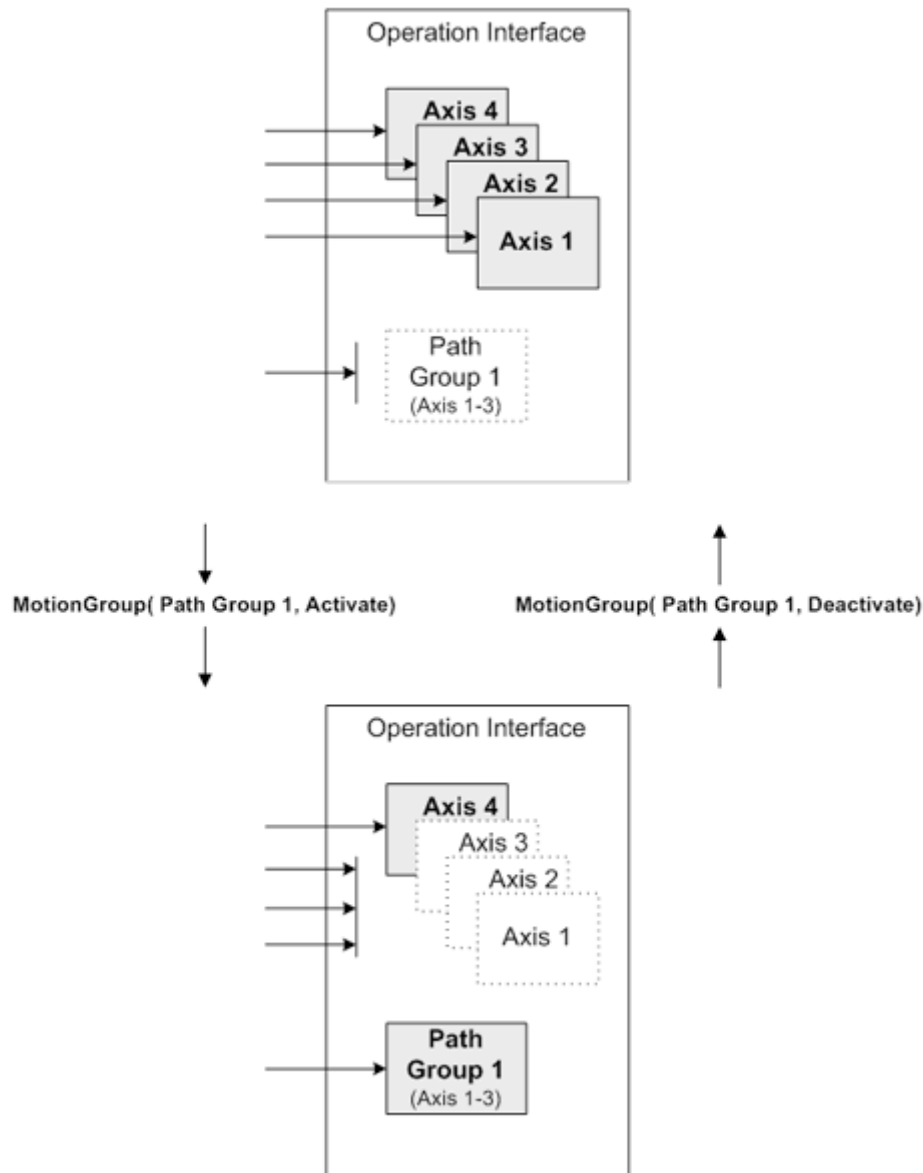
1.1 对象模型

在运动控制软件中，单个轴或轴组被视为对象。首先，这些对象是在 Hardware Manager 中创建的。硬件配置也在这里设置。可用的对象是：

- **轴** - 通常是由 JetMove 驱动的线性或旋转单轴。它们用作一般轴或模态轴。如果尚未连接硬件，也可以将它们配置为仿真轴。然而，它们也可以是纯虚拟轴。
- **路径组** - 路径组是预先配置的单轴的组合，用以执行路径运动。在大多数情况下，所涉及的轴组合成一个末端带有工具的运动链。通过该工具，轴彼此机械耦合。路径组的典型应用是 3D 笛卡尔运动或 SCARA 机器人等。
- **工艺组** - 工艺组是预先配置的单个轴的组合，这些轴用作执行通用工艺顺序。在大多数情况下，轴之间没有机械耦合，它们可以分成一个主轴和多个从轴。工艺组的例子有电子齿轮、电子凸轮等。

一旦这些对象被激活，它们就可以通过集成在 JetSym ST/STX 中的函数（运动控制指令和设置）进行寻址。首先，应使用命令 MotionGroup 明确激活轴组。如果单个轴不包含于任何激活的轴组中，则它们是隐式激活的。因此，一旦激活了一个轴组，就不能再将相关轴作为单独的轴进行寻址。

下图显示了如何在激活前后对轴和轴组进行寻址：



注意：对于非活动对象，仅允许激活和诊断命令。对非活动对象也允许清除错误的命令，但不一定需要发出。

由于将轴组视为单个对象的方法，协调运动的编程变得更容易和更系统。路径组的编程与空间固定的用户特定坐标系中的实际轴配置无关。例如，如果一个工艺组通过相应的功能停止，所有涉及的轴自动停止。

如果由于技术原因，组内的单个轴会表现出特殊的行为（例如，飞锯解耦，运动过程中改变控制器参数），轴组内会为此专门提供一个函数。

在 MCX 软件中使用的面向状态的对象模型中，活动对象始终处于预定义的操作状态（另请参见[状态管理](#)）。根据操作状态，可以通过多个操作通道对对象进行编程、参数化和诊断。

面向状态的方法有以下好处：

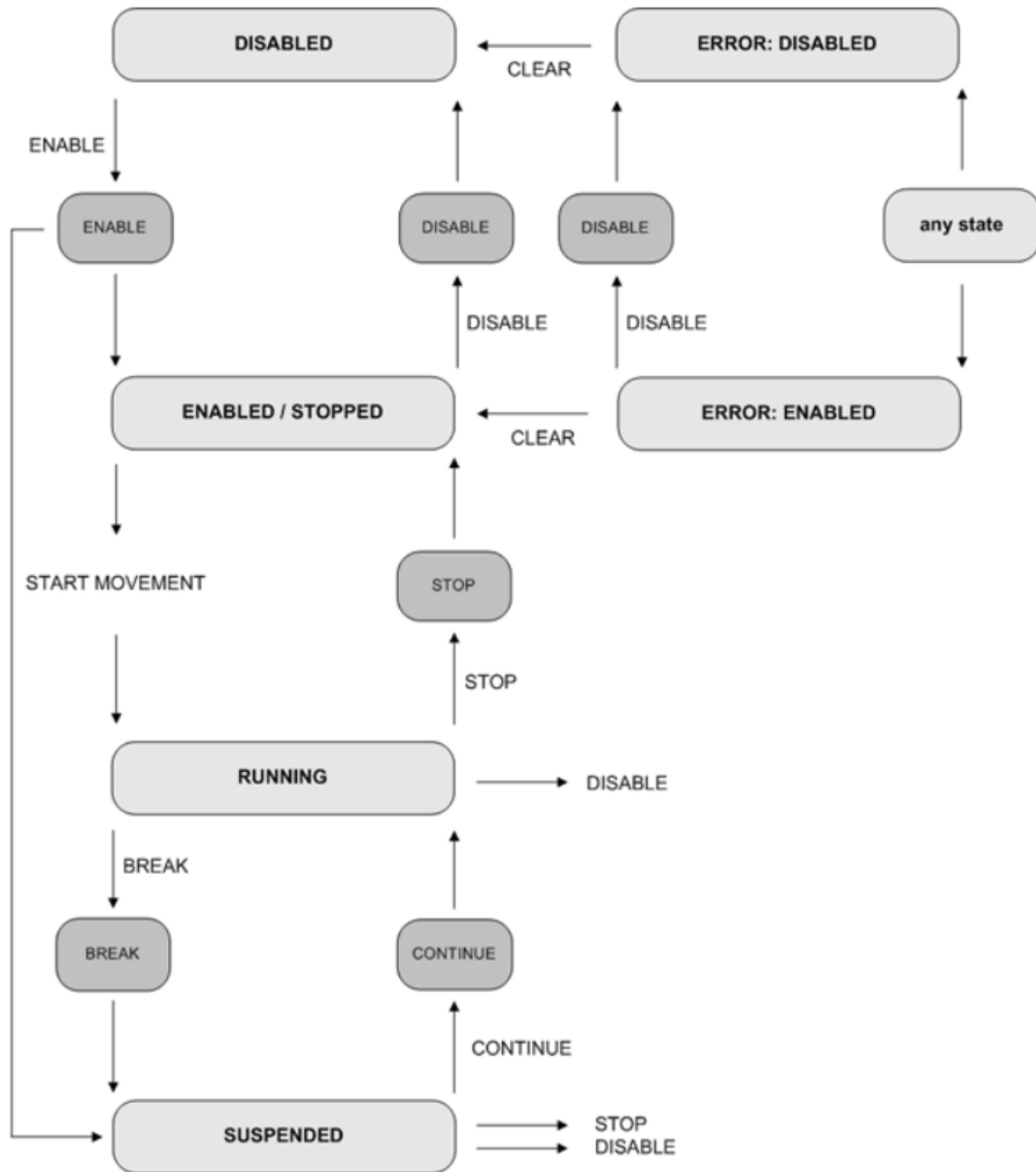
- 系统会检测并忽略无意义的函数调用或编程错误。
- 向用户报告非法函数调用。
- 系统不易出错。
- 结构化程序有助于轴/轴组的编程和调试。

1.2 状态管理

通常，运动控制软件中使用的对象是通过集成在 JetSym ST/STX 中的运动控制命令直接指示的。这样，操作状态转换会自动触发。

对于直接状态管理 - 例如用于在发生错误时进行响应 - JetSym STX 中提供了 Motion API 命令 `<Object>.State.Transitions`。可以使用此命令以及命令 `<Object>.Diagnostics.ClearErrors()` 触发显式操作状态转换。

下图显示了对象可能的操作状态转换：



提示：

一个对象（仅指激活后的轴组）始终只能采用一种定义的操作状态。例如，如果对象处于 *Running* 状态，那么就不处于 *Enabled* 状态（虽然轴事实上已经使能了）。

即使在缓冲模式下，`State.Transitions` 命令也总是立即生效。所有前瞻缓冲区都被清空。

具体来说，操作状态的特点如下：

DISABLED (State.IsDisabled):

轴：轴驱动器的控制回路开路。

工艺/路径组：属于该组的所有驱动器的控制回路开路。

ENABLED/STOPPED (State.IsEnabled):

轴：轴驱动器的控制回路闭合。轴插补器处于静止不动。功能通道的所有前瞻缓冲区都是空的。

工艺组：属于该组的所有驱动器的控制回路都是闭合的。属于该组的所有轴的轴插补器都静止不动。

路径组：属于该组的所有驱动器的控制回路都是闭合的。路径插补器都静止不动。功能通道的所有前瞻缓冲区都是空的。

RUNNING (State.IsRunning):

轴：设定值发生器正在运行。在缓冲操作模式下，功能通道的前瞻缓冲区被处理并可继续填充。

工艺组：至少一个属于该组的轴的轴插补器正在运行。

路径组：路径插补器正在运行。功能通道的前瞻缓冲区被处理并且可以继续填充。

SUSPENDED (State.IsSuspended):

这是与 ENABLED 相同的操作状态，除了所有前瞻缓冲区保持填充状态，但被锁定。可以使用 State.Transition 命令“Resume”恢复暂停的运动。

ERROR: DISABLED/ERROR: ENABLED:

对错误的内部响应产生的错误条件。

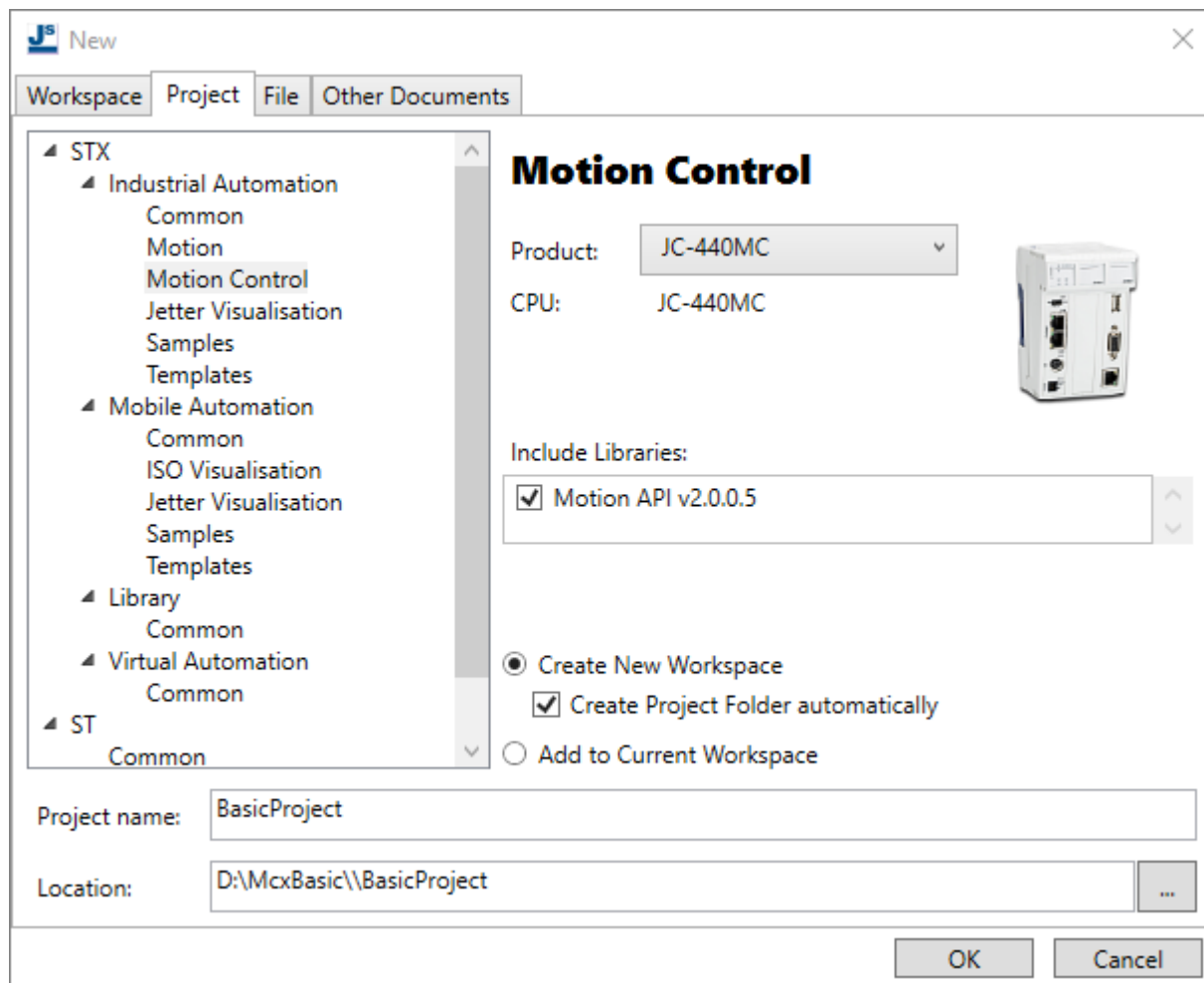
2 配置 MCX 控制器

前提条件:

- JetSymb 5.6.1
- 右手鼠标:
 - 打开子菜单: 右键单击
 - 双击: 用鼠标左键快速单击两次
 - 选择: 单击鼠标左键

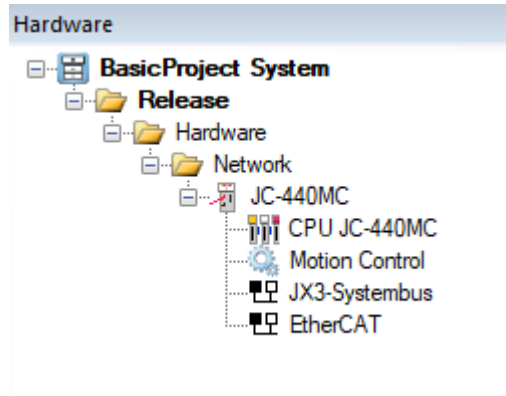
2.1 创建 MCX 项目

通过"File->New"在 JetSymb 中打开项目向导。



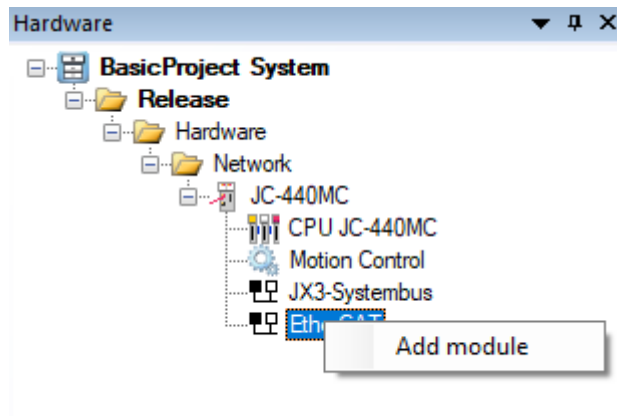
选择一个 MCX 控制器并包含 Motion API。JetSymb 始终建议当前安装并适合所选控制器的 API 版本。单击“确定”后，将创建项目树。

在 Hardware Manager 中，控制器和基本节点已经创建用以进行进一步配置。

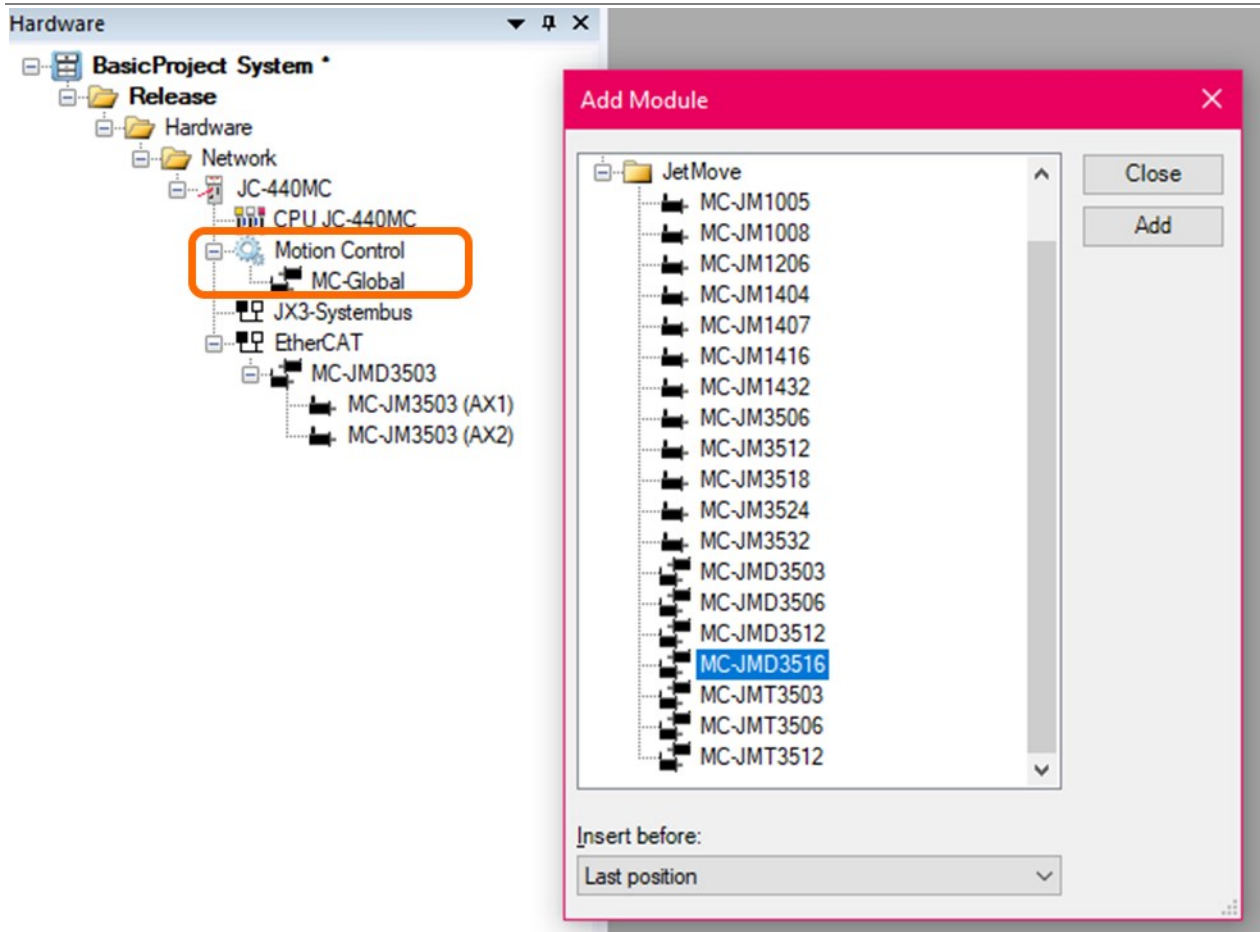


2.1.1 添加伺服驱动器

现在，您可以添加 EtherCAT 伺服驱动器。



"EtherCAT"节点上的子菜单允许您选择要添加到该节点的模块。

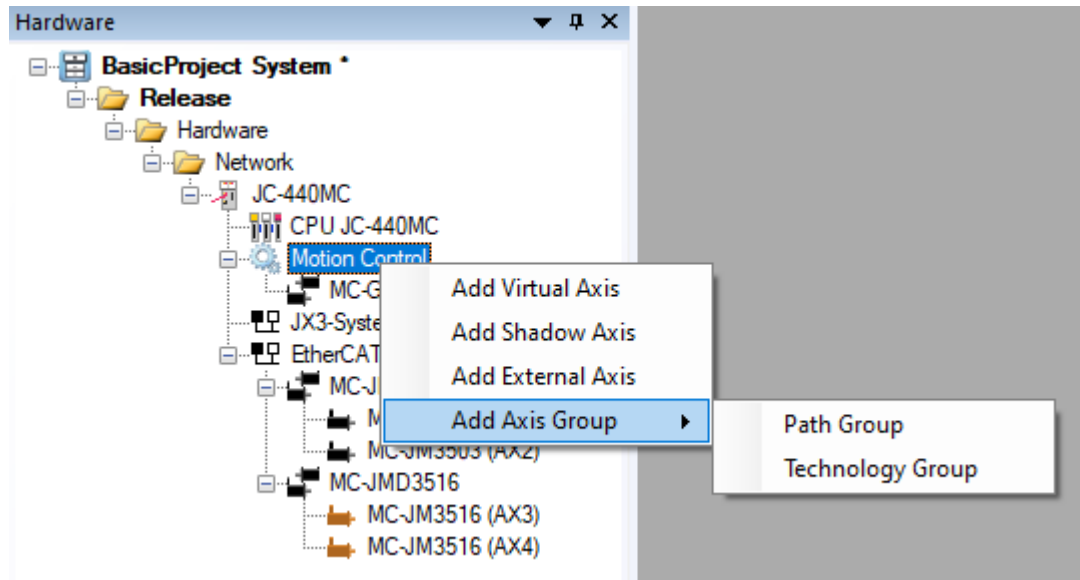


现在选择所需的伺服驱动器并通过双击或单击"Add"将它们添加到硬件配置中。

创建 MC 轴时，也会创建"MC-Global"节点。

2.1.2 创建额外的 MC 对象

打开"Motion Control"子菜单。您现在可以选择要添加到硬件配置中的其他对象。



虚拟轴：该轴没有自己的硬件，但设定值的计算方式与实际轴相同。

影子轴：此轴从创建的轴对象继承设定值和实际值。

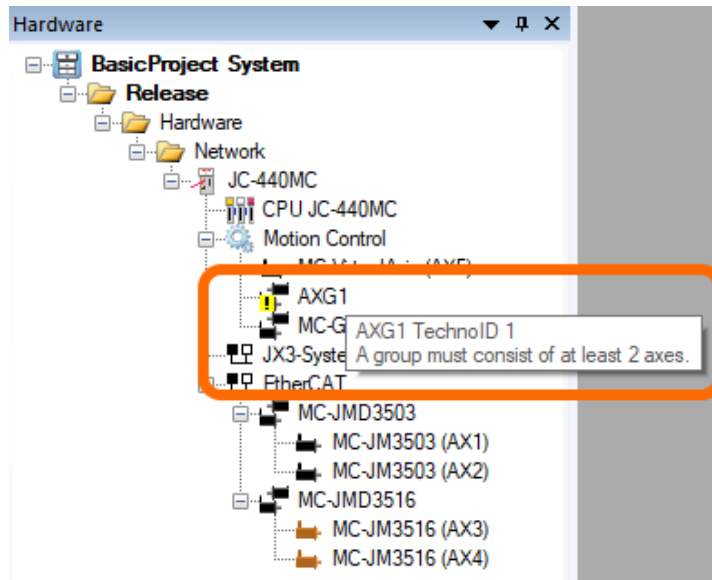
外部轴：此处，实际值来自计数器模块，例如 JM-3000 上的第二个编码器输入。例如，这允许连接摩擦轮以进行位置检测。该轴实际上不能由 MC 主动控制。

路径组：这会为笛卡尔或 SCARA 运动创建路径组。

工艺组：允许您将从动轴链接到主轴以组成电子齿轮或电子凸轮。

2.1.3 错误的硬件配置

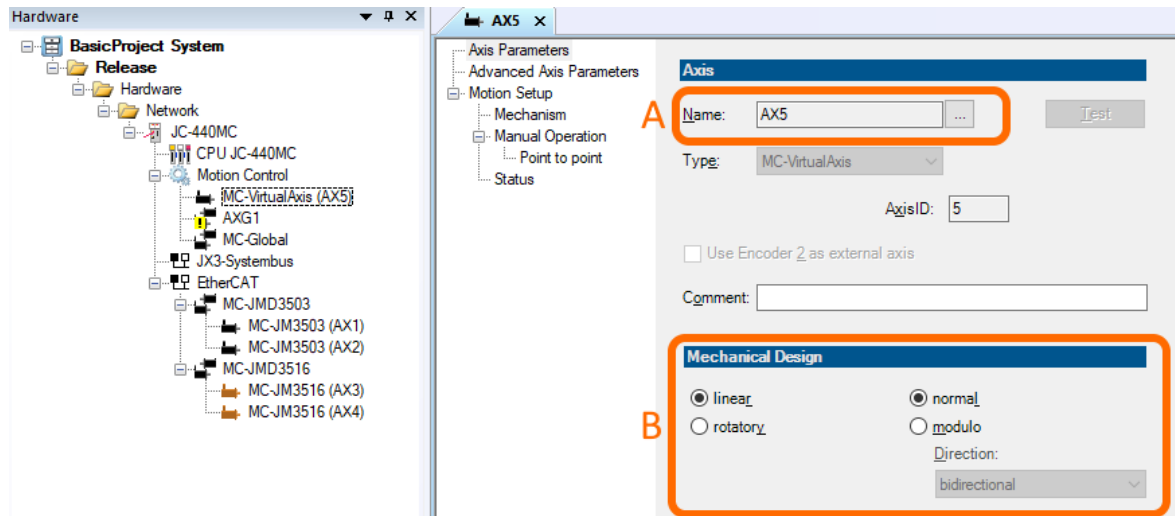
如果硬件配置存在错误，会有一个黄色感叹号指示出来。



在此示例中，已创建工艺组"AXG1"，但仍缺少将哪些轴分配给该组的组态。当您光标移到所选对象上时，JetSim 会显示一个简要说明。

2.1.4 配置轴对象

双击轴对象打开 Motion Setup - 一个用以配置和调试的窗口。



在本例中，虚拟轴"AX5"的 Motion Setup 处于打开状态。

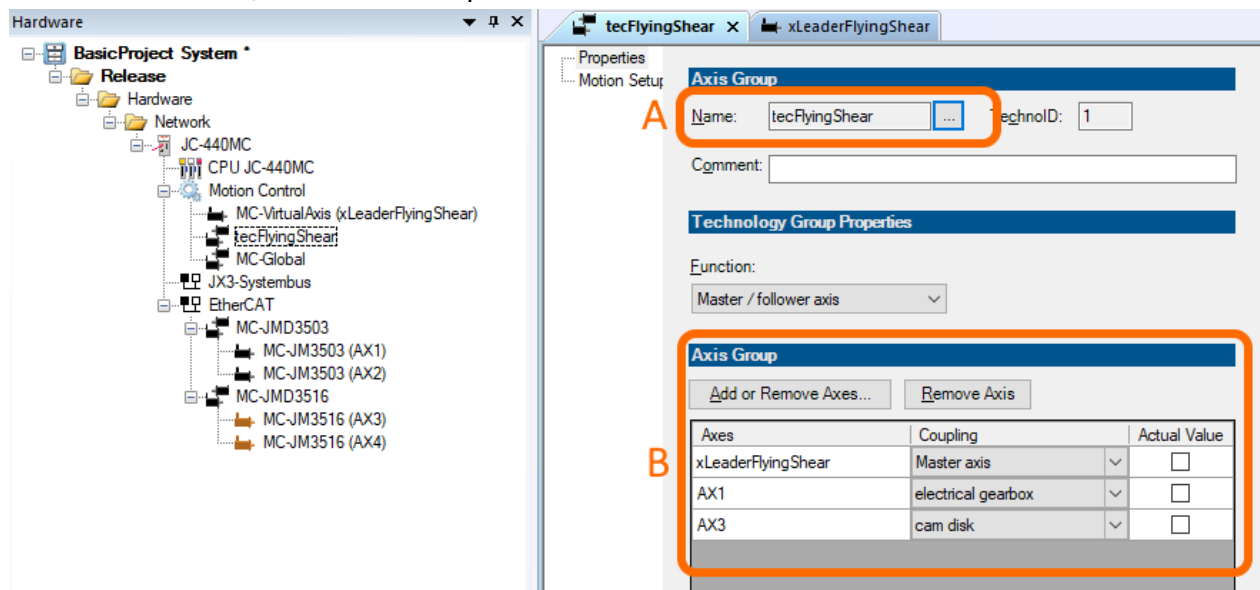
在 Motion Setup 开始页面上，可以定义轴对象的基本设置。

- A. 建议指定一个对应用有意义的描述性名称，而不是建议的轴名称"AX5"。该名称还用于在 STX 程序中寻址相应的轴对象。
- B. 在"Mechanical Design"中，这里设定旋转运动还是直线运动，以及轴是模态轴还是一般轴。
 - 一般轴：该轴具有固定的无法超出的行程范围。
 - 模数轴：当超出模数范围时，轴的位置跳回到起始位置。

在这里，通常会考虑工作点的运动。

2.1.5 配置轴组

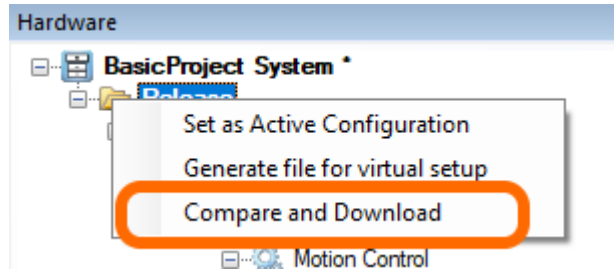
双击组对象打开关联的 Motion Setup。



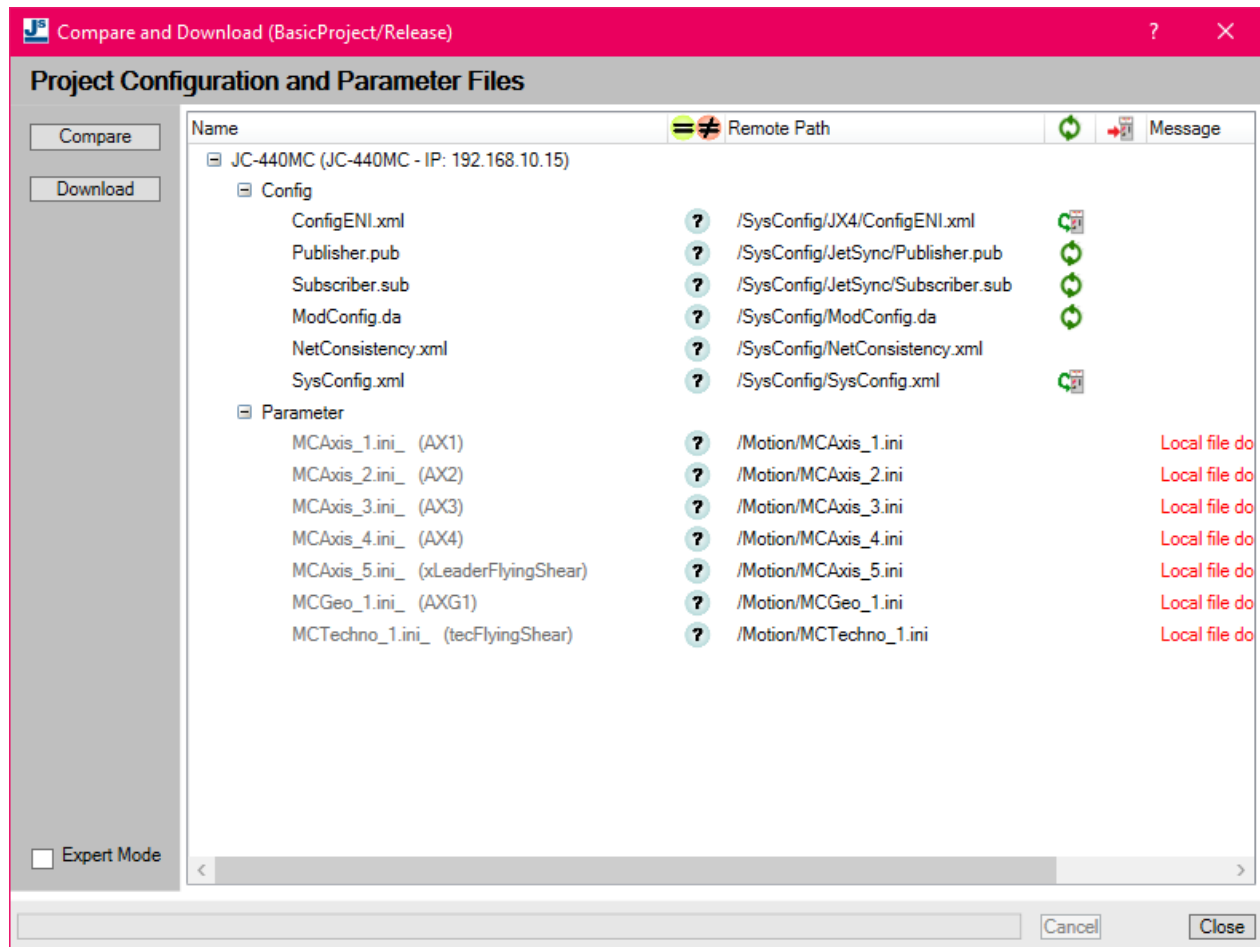
- A. 建议分配一个对应用程序有意义的描述性名称，而不是建议的组名称"AXG1"。此名称还用于寻址 STX 程序中的相应组对象。
- B. 在这里，您可以添加组成员并定义他们的角色。
 - a. 在工艺组中，它们是：
 - i. 主轴
 - ii. 通过电子齿轮耦合的从轴
 - iii. 通过电子凸轮耦合的从轴
 - b. 在路径组中，它们是：
 - i. 笛卡尔坐标：XYZ 轴
 - ii. SCARA：肩、肘、腕、Z 轴

2.1.6 传输配置

在相应 JetSym 配置节点的子菜单中 - 此处为"Release" - 选择菜单项"Compare and Download"。



将会打开一个对话框用以将硬件配置文件下载到控制器。



如果该轴对象的配置文件可从轴调试中获取，也可以使用该文件将配置传输到控制器。

如果选择"Compare"，将本地文件与控制器上的文件进行比较。

当您单击下载按钮时，传输将开始。如果硬件配置已更改，软件会提供控制器重启选项。

在执行重新启动之前，不会使用新的硬件配置。

在 MC-Global Motion Setup (Motion Setup -> Status) 中，您可以在重启后检查 MCX 内核是否能够使用此硬件配置成功启动。

CPU MC-Global X

Properties

Motion Setup

Load

Save

Upload

Download

Help

Diagnose

Errors

Status

Files

Time

Axis / Axis group

Message

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

Start file-logging

Configuration

Open Log-File

More messages

No error.

3 MCX 编程

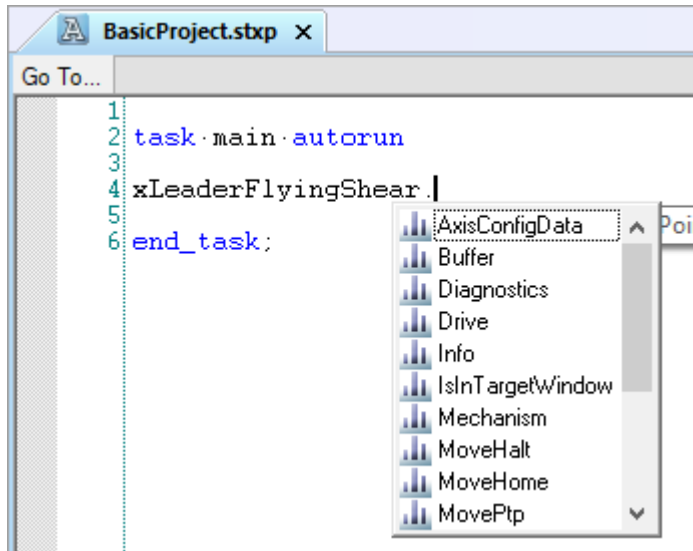
Motion API 允许您在 STX 中对 MCX 进行编程。这是一个面向对象的库，其中可用的函数和属性是按逻辑组织的。

根据硬件配置，在后台生成 HardwareConfig.stxp。在此过程中，会创建 MC 对象，例如轴、组以及用于 Motion API 的 MC 管理器。

为了使对象在程序中可用，需要提前无错误的编译一次。因此，空白任务有助于无错误地编译。

```
task main autorun
end_task;
```

现在系统已经为 Motion API 创建了所有对象，IntelliSense 将帮助您输入指令。



相应对象的所有可用功能都映射在 Motion API 中。

3.1 示例程序

这是一个简短的示例程序，用于显示具体如何编程以及可用的元素。

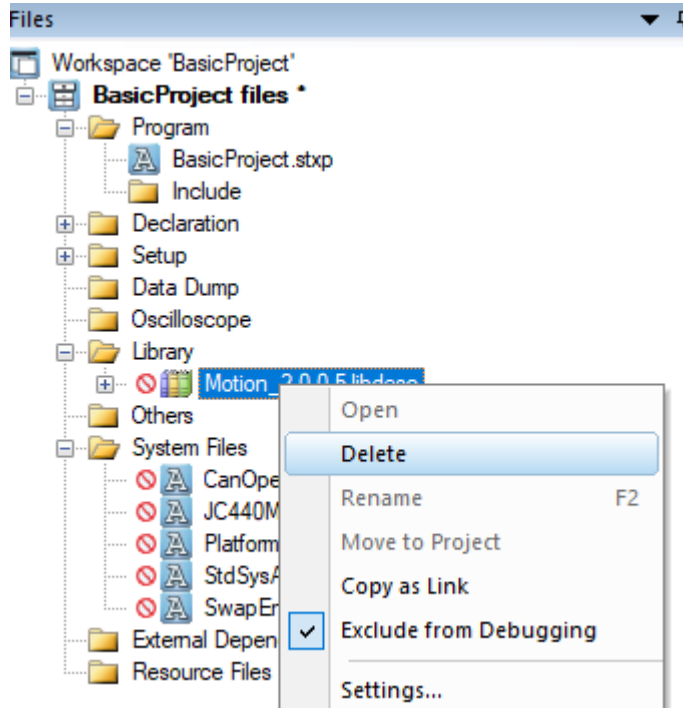
```
task main autorun
    xLeaderFlyingShear.MoveHome.SetReference(0.0);           // A
    xLeaderFlyingShear.Power.Enable();                       // A
    when xLeaderFlyingShear.State.IsEnabled continue;       // B
    tecFlyingShear.Activate();                                // C
    when tecFlyingShear.State.IsEnabled 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. 单轴命令，如设置原点、启用、禁用
- B. MC 对象的状态查询：单轴或轴组的状态
 - B1：特殊状态查询，如斜坡状态
- C. 轴组的激活和停用
- D. 轴组命令，例如定位和停止群轴组成员

3.2 选择 Motion API 版本

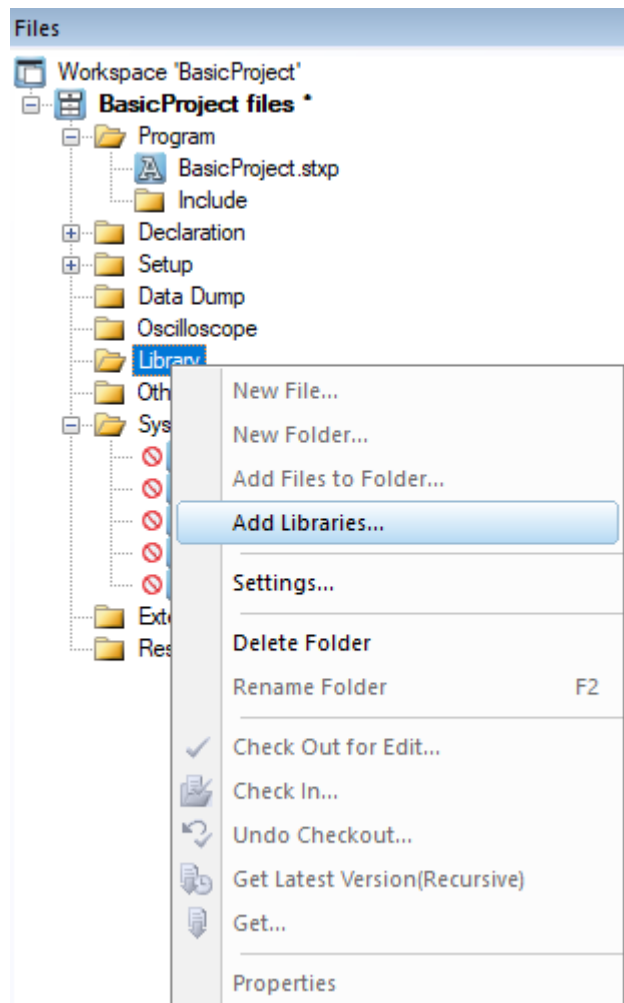
创建新项目时，JetSym 始终建议当前安装的版本。但是，要使用特定版本或要在现有项目中集成更新版本 Motion API，则需要执行以下步骤。

1. 从项目树中删除现有的 Motion API。

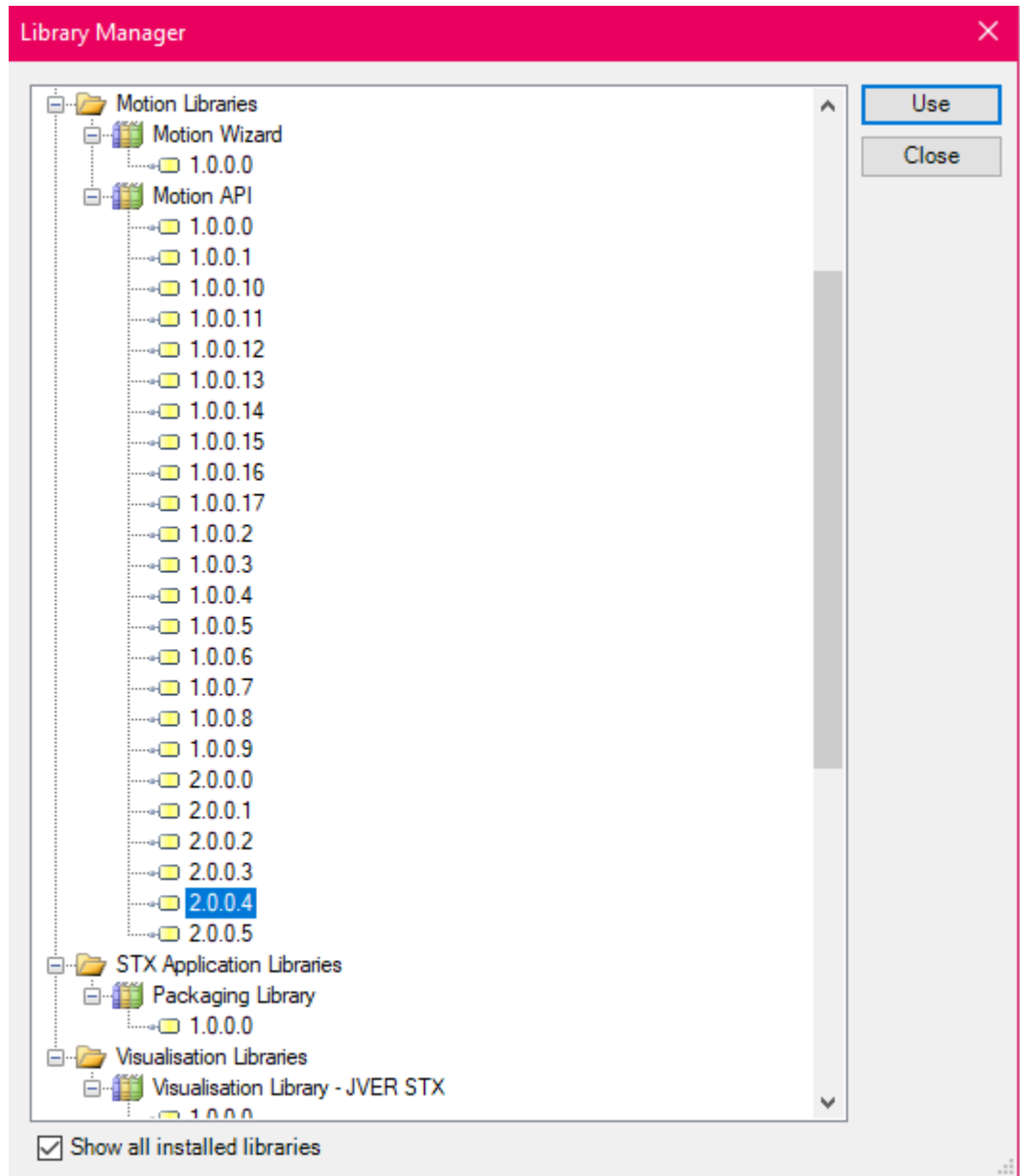


一个项目中一次只能集成一个 Motion API。如果不删除已包含的版本，则无法包含其他版本。

2. 通过项目树中的"Library"子菜单打开库选择。



3. 选择所需的 Motion API 版本。



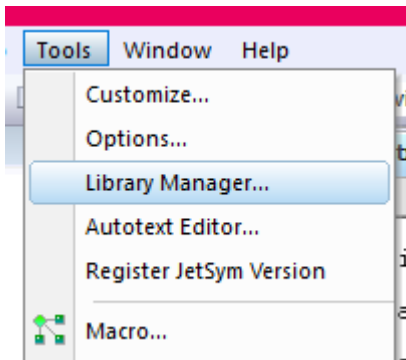
如果必须的话，必须选中"Show all installed libraries"。

通过单击"Use"，所选的 Motion API 即可集成。

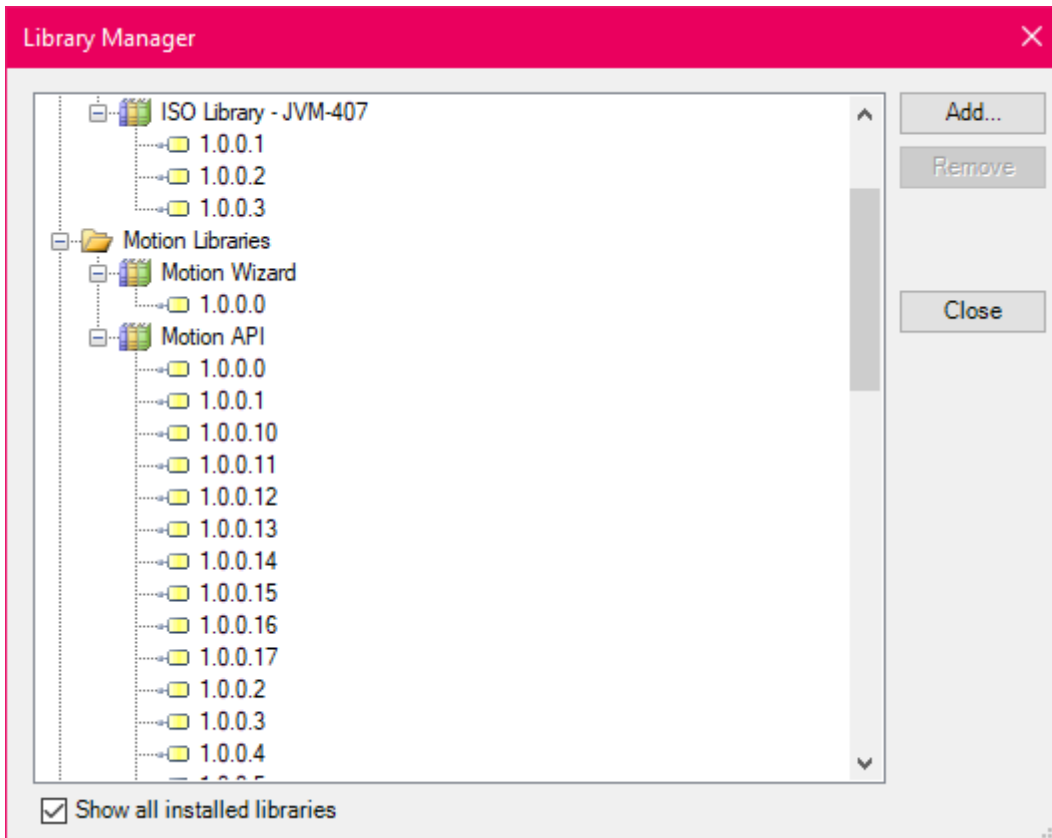
3.2.1 添加 Motion API

只有先前添加到库管理器的 Motion API 库才能包含在项目中。

1. 打开库管理器。



2. 库管理器打开并显示已安装的库。



3. 单击"Add"以打开"Open file"对话框。选择应该添加的"libpackage"。
4. 如果必须的话，必须选中"Show all installed libraries"，以便显示库。

现在，您可以按照“选择 Motion API 版本”中的说明将附加库包含在您的项目中。

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

坚德自动化技术（上海）有限公司.
上海市浦东新区康桥路 787 号 6 号楼 105 室
邮编: 201315
www.jetterat.cn

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

contact@jetterat.cn
+86 21 5869 1233