

# Motion Control - Technologieverbund

Application Note 049

608 847 04\_01

We automate your success.

Dieses Dokument wurde von der Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt. Änderungen und technische Weiterentwicklungen an unseren Produkten werden nicht automatisch in einem überarbeiteten Dokument zur Verfügung gestellt. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Jetter AG  
Gräterstraße 2  
71642 Ludwigsburg  
Deutschland

[www.jetter.de](http://www.jetter.de)

**Telefon:**

Zentrale	+49 7141 2550-0
Vertrieb	+49 7141 2550-531
Technische Hotline	+49 7141 2550-444

**E-Mail:**

Technische Hotline	info@jetter.de
Vertrieb	hotline@jetter.de
	vertrieb@jetter.de

Produktname	Motion Control - Technologieverbund
Dokumentart	Application Note 049
Originaldokument	
Dokumentenversion	1.01
Artikelnummer	608 847 04_01
Ausgabedatum	23.06.2021

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>1</b>
1.1	Voraussetzungen .....	1
1.2	Was ist ein Technologieverbund? .....	1
1.3	Arten der Kopplung .....	1
1.3.1	Elektrisches Getriebe .....	1
1.3.2	Kurvenscheibe .....	1
1.3.3	Ist-Sollwertkopplung .....	2
1.4	Arten von Leitachsen .....	2
1.5	Arten von Folgeachsen .....	3
<b>2</b>	<b>Anlegen eines Technologieverbunds.....</b>	<b>4</b>
2.1	Erzeugen eines Technologieverbunds .....	4
2.2	Konfigurieren des Technologieverbunds .....	11
2.3	Downloaden der MC-Dateien auf Steuerung und Neustart.....	14
<b>3</b>	<b>Anwenden des Technologieverbunds im MotionSetup .....</b>	<b>16</b>
3.1	Bedienen des Technologieverbunds .....	16
3.1.1	Fehlerübersicht .....	18
3.1.2	Statusübersicht .....	19
3.1.3	Weitere Ansichten .....	19
<b>4</b>	<b>Anwenden des Technologieverbunds im Anwendungsprogramm .....</b>	<b>20</b>
4.1	Einbinden der Motion API-Bibliothek in ein Projekt .....	20
4.2	Programm neu starten .....	21
4.2.1	Rebooten der Steuerung .....	21
4.2.2	Neustart der Applikation .....	22
4.3	Aktivieren des Technologieverbunds .....	24
4.4	Deaktivieren des Technologieverbunds .....	25
4.5	Positionierung .....	26
4.5.1	MovePtp .....	27
4.5.2	MoveVelocity .....	28
4.5.3	MoveHalt .....	29
4.5.4	MoveHome .....	30
4.6	Diagnose .....	31

# 1 Einleitung

## 1.1 Voraussetzungen

Für die Code- und Projektbeispiele sowie Screenshots wurden folgende Versionen verwendet:

- JetSym 5.60

<ul style="list-style-type: none"><li>• JC-440MC, OS 1.08.0.00</li><li>• Motion API 2.0.0.4</li></ul>	<ul style="list-style-type: none"><li>• JC-365MC, OS 1.33.0.00</li><li>• Motion API 1.0.0.16</li></ul>
---	--

Für die Application Note werden exemplarisch diese beiden Steuerungen als Vertreter der unterschiedlichen Motion API 1 + 2 verwendet.

Soweit Unterschiede in der Verwendung auftreten, werden diese in dem Dokument dargestellt.

## 1.2 Was ist ein Technologieverbund?

- In einem Technologieverbund können Einzelachsen zu einem Verbund zusammengefasst werden, um die Bewegung der Einzelachsen aufeinander abzustimmen. Die Verbundart ist dabei eine Leit-Folgeachs-Kopplung. Die Folgeachsen können dynamisch ein- und ausgekoppelt werden.
- Er besteht immer aus einer Leitachse und 1 ... n Folgeachsen.

## 1.3 Arten der Kopplung

### 1.3.1 Elektrisches Getriebe

Der Begriff "elektrisches Getriebe" wird im übertragenen Sinn für eine elektronische Schaltung verwendet, die wie ein mechanisches Getriebe zu einer Drehzahl- und Positionsänderung führt. Im Gegensatz zum mechanischen Getriebe ist bei einem elektrischen Getriebe eine stetige Änderung des Übersetzungsverhältnisses möglich. Dynamische Ein- und Auskoppelvorgänge, sowie Korrekturen von Positionsverschiebungen lassen sich mit dem elektrischen Getriebe leichter verwirklichen.



- Ein elektrisches Getriebe wird bei der Definition eines Technologieverbundes angelegt. Das Getriebeverhältnis zwischen der Leitachse und einer Folgeachse kann bei aktiviertem Technologieverbund mit einem Quotienten dynamisch individuell vorgegeben werden.
- Für die Folgeachse sind überlagerte Bewegungen möglich.

### 1.3.2 Kurvenscheibe

Kurvenscheibenantriebe werden eingesetzt, wenn die Position einer oder mehrerer Folgeachsen von der Position einer Leitachse abhängt und komplexe Bewegungsabläufe in zyklisch laufenden Maschinen aufeinander abzustimmen sind. Konventionell werden dabei mechanische Kurvenge triebe verwendet.

- Elektrische Kurvenscheiben bieten eine weit größere Flexibilität, da sehr schnell auf andere Kurvenverläufe umgeschaltet werden kann und die mitbewegte Masse kleiner ist.

- Es wird eine höhere Dynamik und höhere Taktzahl erreicht, die Antriebe sind kleiner und der Verschleiß ist reduziert.
- Eine elektronische Kurvenscheibe lässt sich flexibler ein- und ausschalten und über einen beliebigen Offset verschieben. Bei einer mechanischen Kurvenscheibe ist ein Offset nur mit höherem Aufwand durch eine Verdrehung des Kurvenkörpers auf der Antriebswelle möglich.

### 1.3.3 Ist-Sollwertkopplung

Ein Achsobjekt hat in der Regel einen Positionssollwert und einen Positionsiswert.

- Der Positionssollwert wird zyklisch durch die MCX berechnet und dem Lageregler der Achse zur Regelung übergeben.
- Der Positionsiswert beschreibt die Position, an der sich die Achse im Moment befindet.

Bei der **Istwertkopplung** werden die Folgeachsen an den Positionsiswert der Leitachse gekoppelt.

Ist die Leitachse eine reale Achse oder ein externer Geber, ist selbst im Stillstand die Istposition nicht exakt konstant. Diese Schwankungen erhalten die Folgeachsen als Leitwert, was zu einem unruhigen Verhalten führen kann. Um diesen Effekt zu reduzieren, kann ein entsprechender Filter zur Glättung eingestellt werden. Allerdings führt ein Filter auch zur Verzögerung der Folgeachsreaktionen.

Des Weiteren entsteht durch die Übertragung des Istwerts zum MC-Kernel eine Verzögerung, welches auch die Folgeachsreaktion verzögert.

Diese Verzögerungen sind hauptsächlich bei Geschwindigkeitsänderungen der Leitachse bemerkbar.

Istwertkopplung wird verwendet, um eine Folgeachse z.B. mit Hilfe eines Reibrads (Leitachse) mit einem vorbeifahrenden Produkt oder Schlupf-behafteten Transport zu synchronisieren.

Bei der **Sollwertkopplung** werden die Folgeachsen an den Positionssollwert der Leitachse gekoppelt.

Aufgrund der Berechnung des Positionssollwerts durch die MCX ist dieser Wert genau und z.B. im Stillstand immer konstant. Ein Filter zur Glättung des Positionssollwerts ist nicht nötig.

Der neue Positionssollwert steht den Folgeachsen im gleichen Berechnungszyklus der MCX zur Verfügung, so dass keine Verzögerungen auftreten. Geschwindigkeitsänderungen werden damit sofort an die Folgeachsen weitergegeben.

Deshalb ist eine Sollwertkopplung in den meisten Fällen zu bevorzugen. In Fällen mit einem externen Geber, wie z.B. Reibrad, kann aber nur die Istwertkopplung verwendet werden.

## 1.4 Arten von Leitachsen

- Reale Achse
  - Hat einen Servoverstärker, außer als Simulationsachse
  - Die Sollwerte werden durch die MC berechnet
  - Die Istwerte werden vom Servoverstärker zurückgemeldet
  - Kann alle MCX-Betriebszustände einnehmen, wie z. B. "Enabled" oder "Running".  
Siehe dazu auch die Übersicht der MCX-Betriebszustände
- Virtuelle Achse
  - Hat keinen Servoverstärker
  - Sollwerte werden durch die MCX berechnet
  - Istwert = Sollwert
  - Kann alle MC-Betriebszustände einnehmen, wie z.B. "Enabled" oder "Running".  
Siehe dazu auch die Übersicht der MCX-Betriebszustände
- Externe Achse
  - Hat ein Zählermodul, also reine Geberauswertung
    - Zählermodul bei JC-440MC, JC-945MC oder JC-975MC mit EtherCAT® ist der zusätzliche Gebereingang auf einem Drive, z.B. JM3000.

- Zählermodul bei JC-647MC, JC-365MC, JC-940MC oder JC-970MC ist ein zusätzlicher Antriebsverstärker, z.B. JM105, JM 108, JM2xx.
  - Sollwerte werden nicht berechnet
  - Istwerte werden vom Zählermodul zurückgemeldet
  - MC-Betriebszustand ist immer "Inaktiv"
- Mitlaufachse/Shadow axis
  - Hat keinen Servoverstärker
  - Ist- und Sollwerte werden von der Quellachse übernommen
  - MC-Betriebszustand ist immer "Inaktiv"

## 1.5 Arten von Folgeachsen

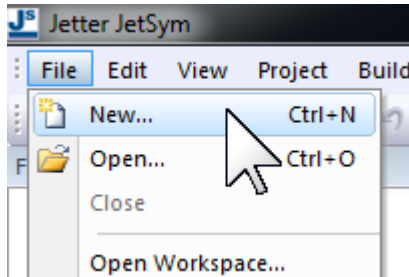
- Reale Achse
- Virtuelle Achse

## 2 Anlegen eines Technologieverbunds

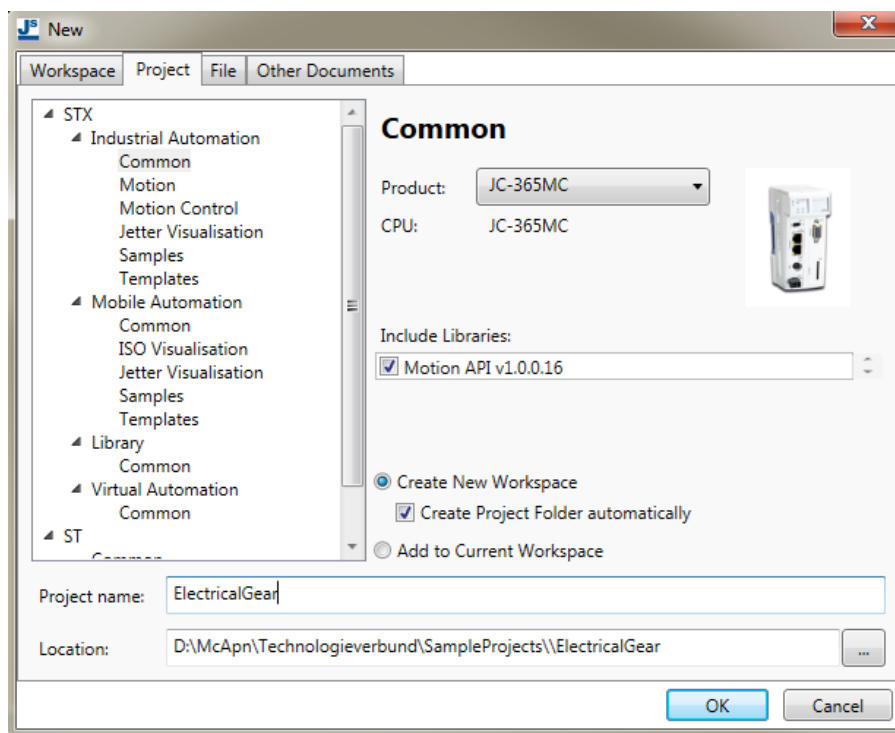
### 2.1 Erzeugen eines Technologieverbunds

Am Beispiel eines neuen Projekts sehen Sie, wie ein Technologieverbund angelegt wird.

- Öffnen Sie JetSym
- Öffnen Sie das Menü "*Datei*" und wählen "*Neu*"

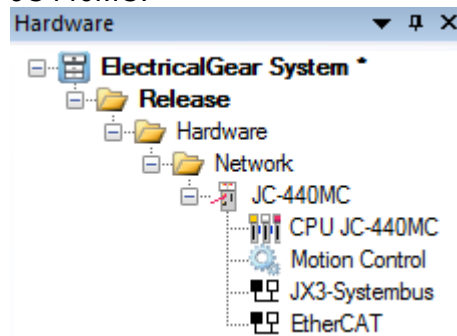


- Es öffnet sich der Dialog zum Anlegen eines Projektes. Wählen Sie hierbei z.B. die MotionControl-Steuerung "JC-440MC" oder „JC-365MC" aus, binden die vorgeschlagene Motion API-Bibliothek ein. Nach Eingabe eines Projektnamens und Auswahl des passenden Ordners in dem das Projekt gespeichert werden soll, bestätigen Sie den Dialog mit "OK"

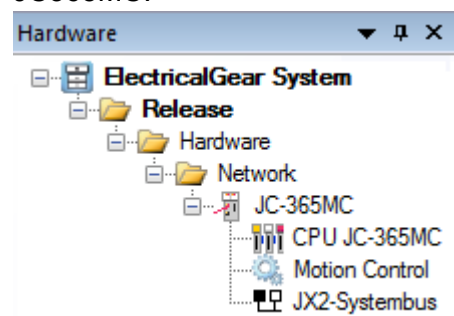


- Es wird eine Grundstruktur im Hardwarebaum erstellt.

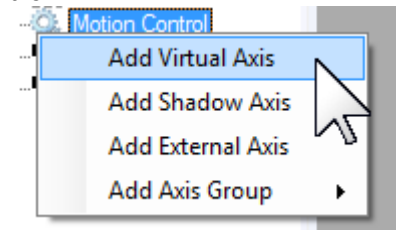
JC440MC:



JC365MC:

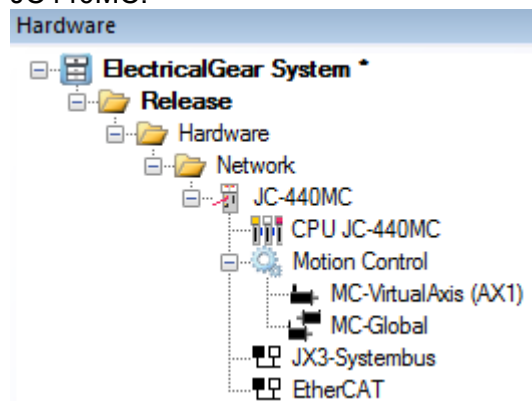


- Als Erstes legen Sie eine virtuelle Achse als Leitachse an. Über das Kontextmenü "*Virtuelle Achse hinzufügen*" – erreichbar per Rechtsklick mit der Maus auf den Knoten "*MotionControl*" - wird die virtuelle Achse angelegt.

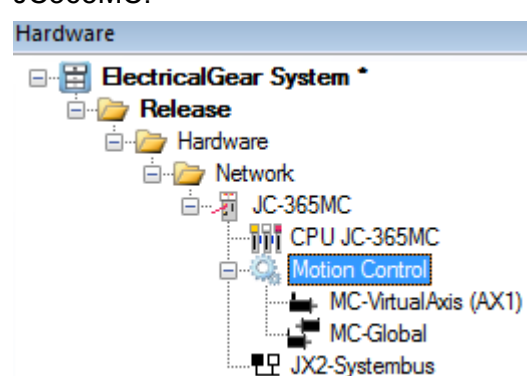


- Diese wird nun im Hardwarebaum angezeigt:

JC440MC:

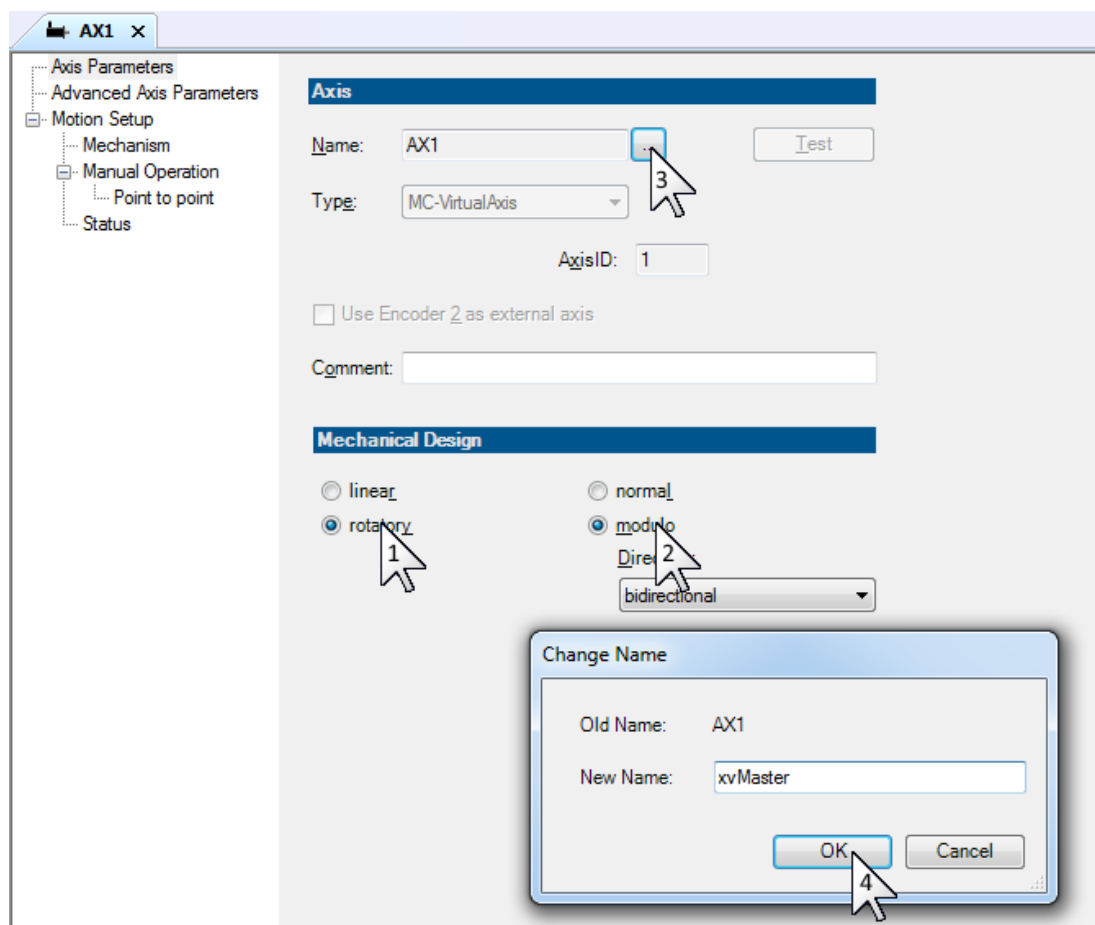


JC365MC:





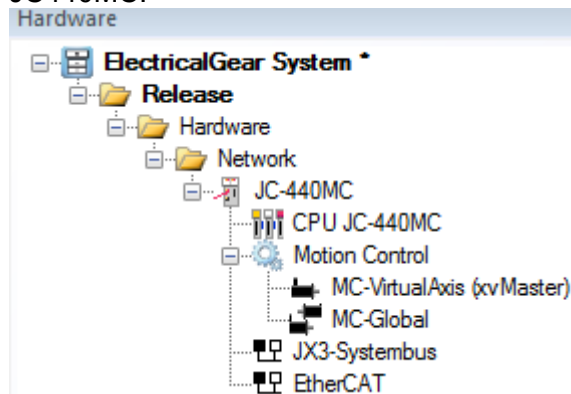
- Durch Doppelklick auf den Knoten "MC-VirtualAxis (AX1)" öffnen Sie das MotionSetup für diese virtuelle Achse.



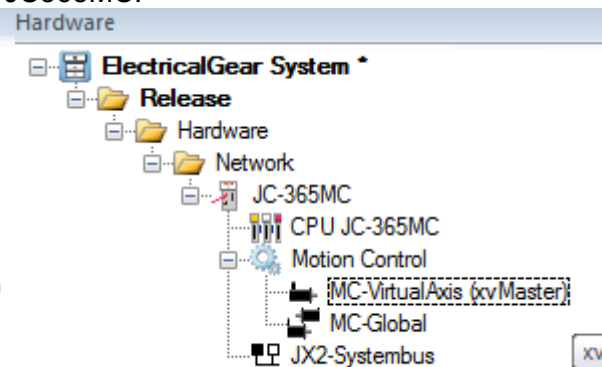
Setzen Sie die "Mechanische Ausführung" auf "Rotatorisch"(1) und "Modulo"(2). Durch Klicken auf den Button "..."(3) öffnen Sie den Dialog zum Ändern des Achsnamens. In diesem Beispiel benennen Sie die Achse "xvLeitachse" und bestätigen dies mit "OK"(4). Bestätigen Sie die weiteren Nachfragen mit "Ja". Innerhalb des STX-Projekts wird ein MotionAPI-Objekt erstellt, über das die virtuelle Achse angesprochen wird.

- Im Hardwarebaum erscheint nun ebenfalls der neue Name der Achse.

JC440MC:

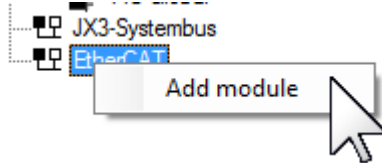


JC365MC:

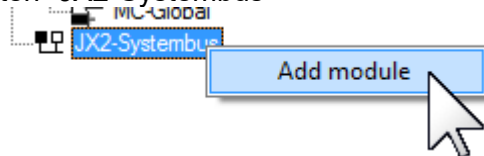


- Nun fügen Sie eine reale Achse hinzu.

JC440MC: Öffnen Sie dazu das Kontextmenü durch Rechtsklick mit der Maus auf den Knoten "EtherCAT"

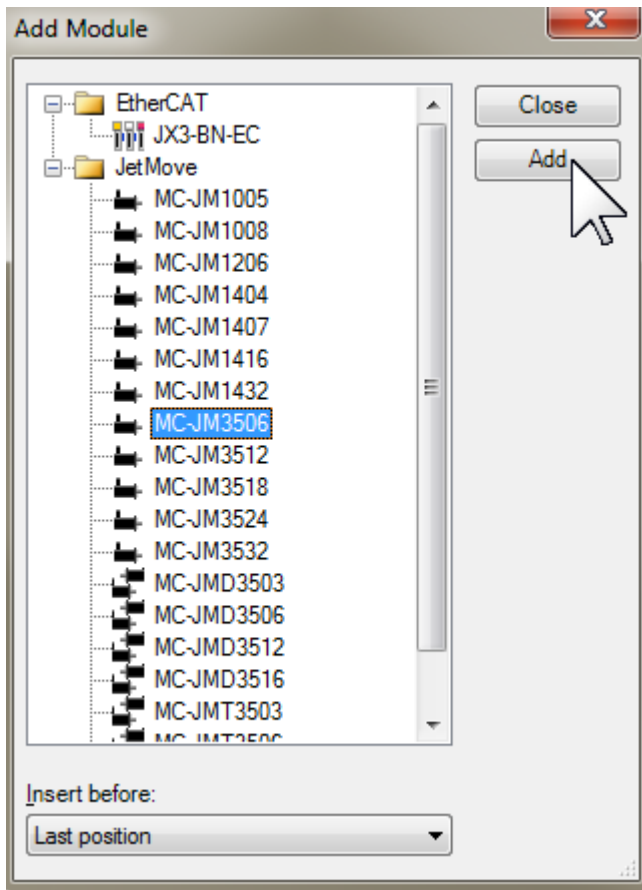


JC365MC: Öffnen Sie dazu das Kontextmenü durch Rechtsklick mit der Maus auf den Knoten "JX2-Systembus"

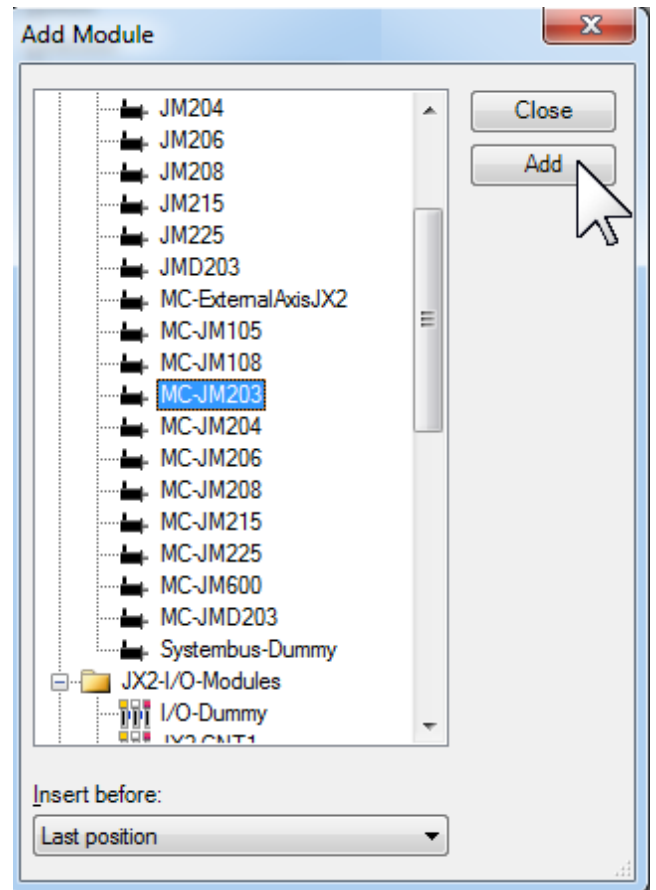


- Fügen Sie nun die benötigte Anzahl an Achsen hinzu. Bitte wählen Sie hierfür Achsen mit einem führenden "MC" wie z.B. "MC-JM203" aus. Für dieses Beispiel genügt eine Achse. Schließen Sie danach das Dialogfeld.

JC440MC:

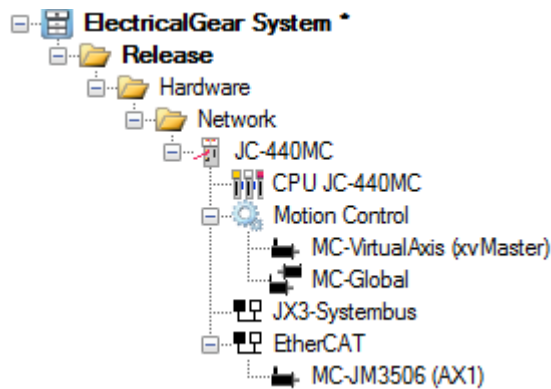


JC365MC:

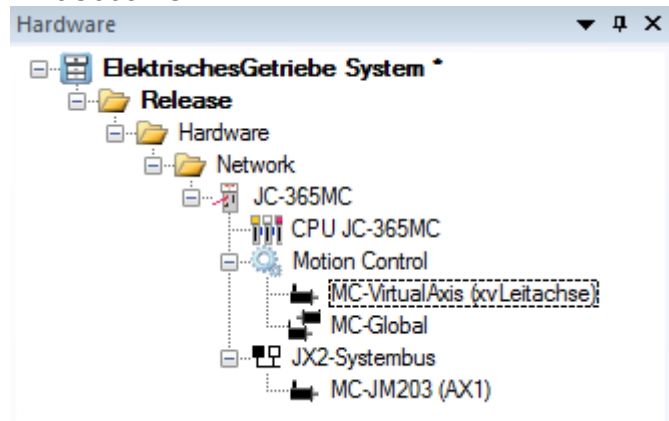


Der Hardwarebaum zeigt nun die angelegten Achsen als Unterknoten zu dem gewählten Bus an.

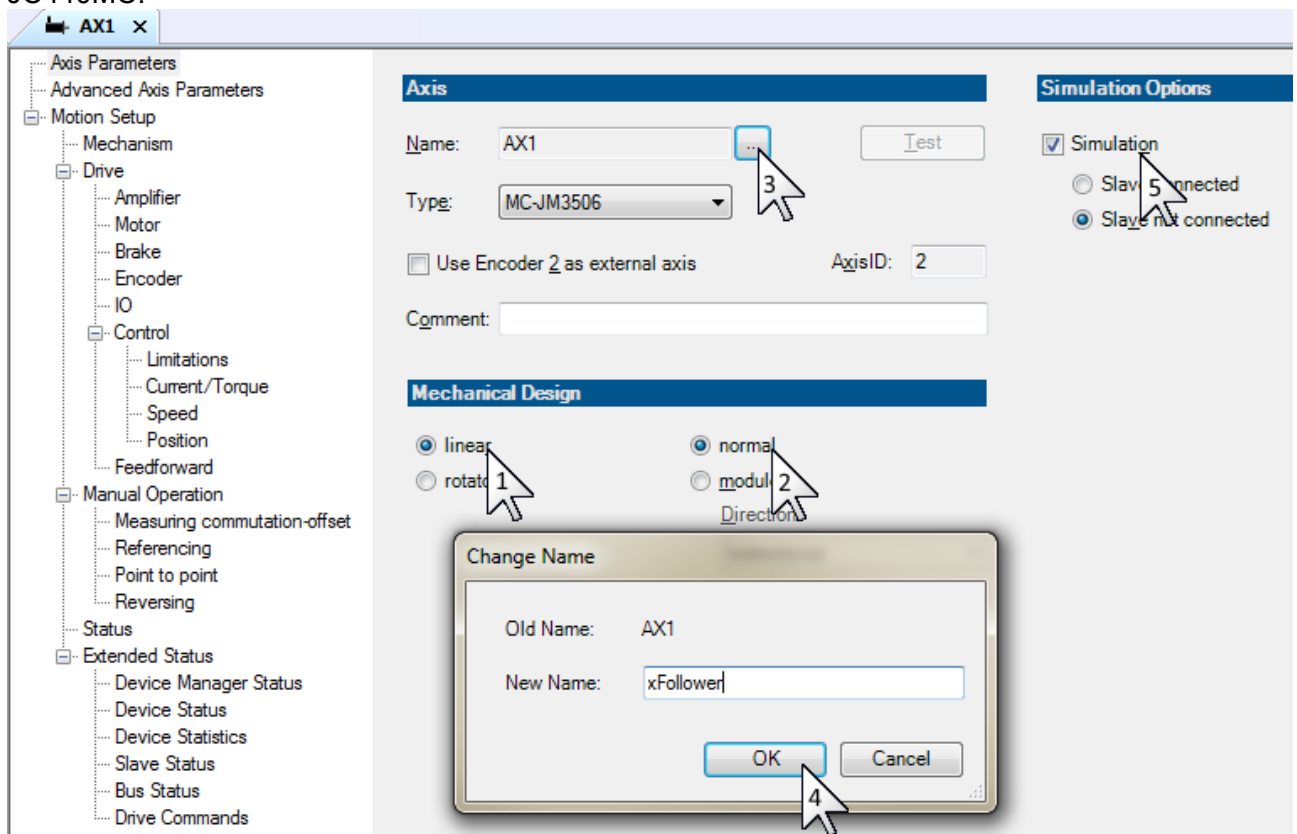
JC440MC:



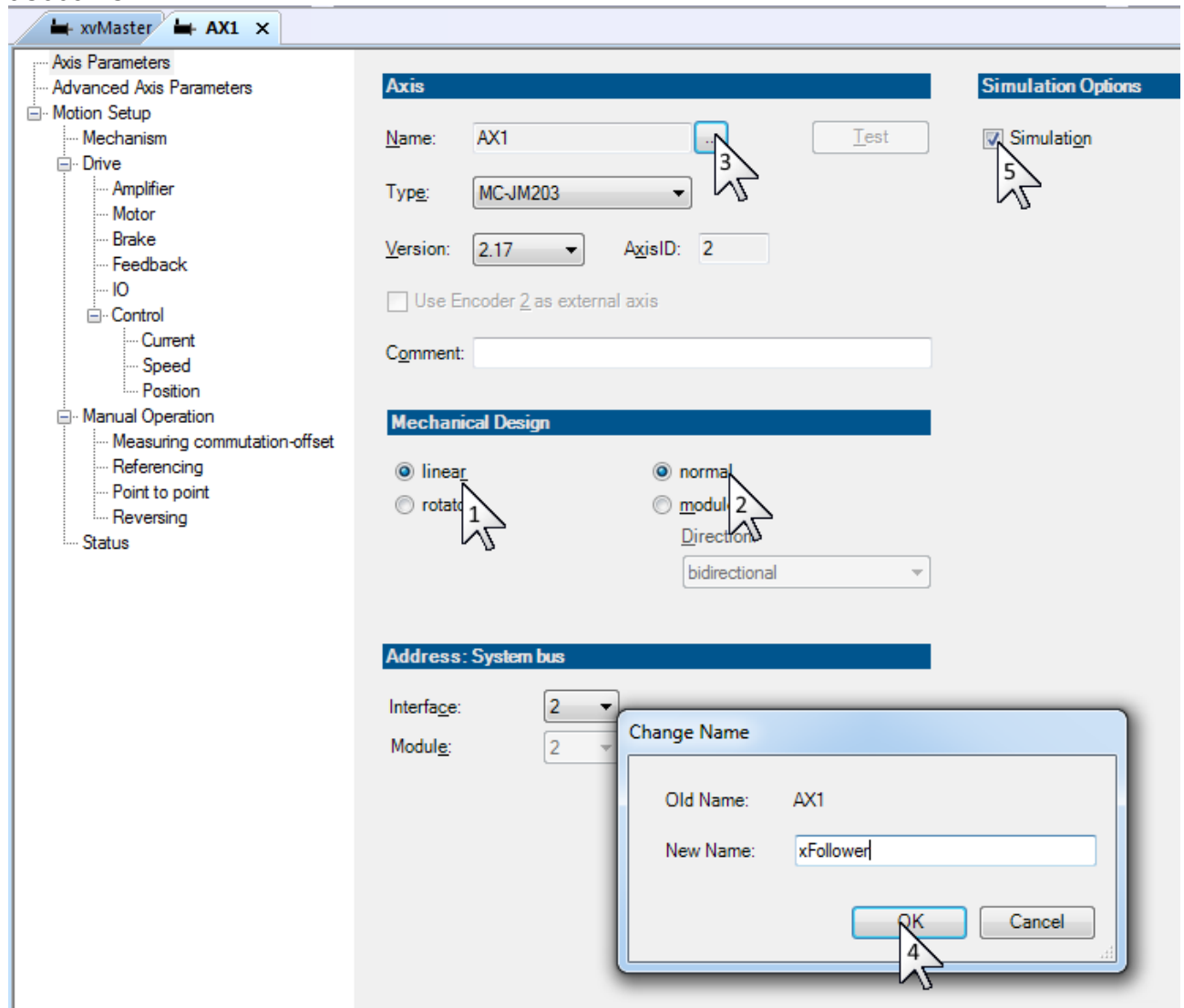
JC365MC:



- Zum Umbenennen und Konfigurieren öffnen Sie das MotionSetup der Achse "AX1" durch einen Doppelklick auf den Knoten.
- JC440MC:



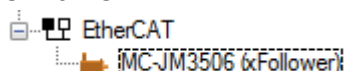
## - JC365MC:



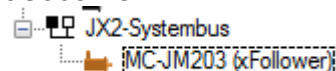
Wählen Sie auch hier die passende "Mechanische Ausführung" entsprechend Ihrer Anwendung "*Linear*" oder "*Rotatorisch*" (1) und "*Normal*" oder "*Modulo*" (2). Durch Klicken auf den Button "..."(3) öffnen Sie den Dialog zum Ändern des Achsnamens. In diesem Beispiel benennen Sie die Achse "*xFolgeachse*" und bestätigen dies mit "OK"(4). Bestätigen Sie die weiteren Nachfragen mit "Ja". Wählen Sie zudem durch Anhängen von "*Simulation*" (5) aus, ob es sich hier um eine Simulationsachse handelt. Innerhalb des STX-Projekts wird ein MotionAPI-Objekt erstellt, über das die Achse angesprochen wird.

- Der geänderte Name wird nun auch im Hardwarebaum angezeigt.

## JC440MC:

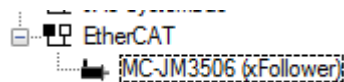


## JC365MC:

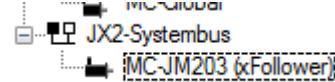


- Am Achsensymbol können Sie schon im Hardwarebaum erkennen, ob es sich um eine Simulationsachse (orange) oder reale Achse (schwarz) handelt.

JC440MC:

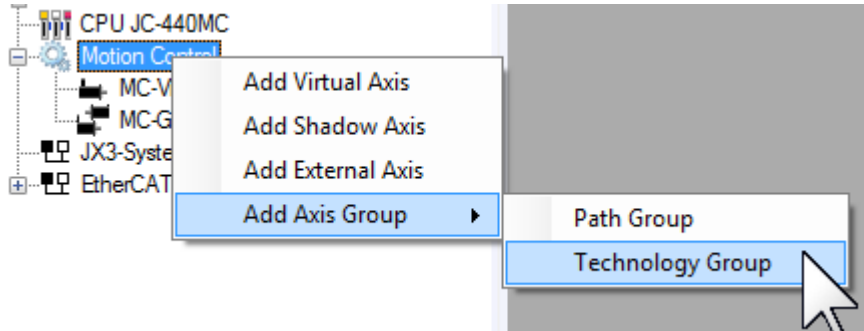


JC365MC:

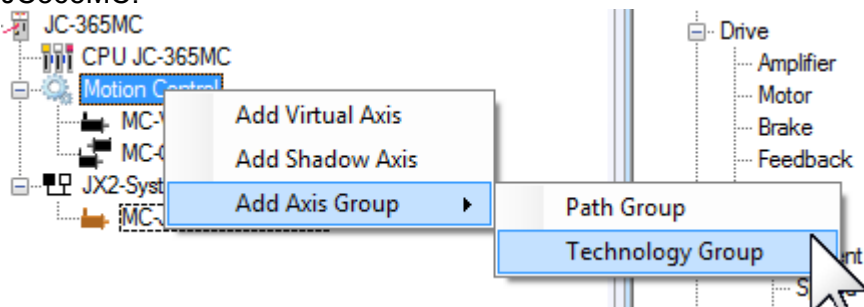


- Öffnen Sie durch Rechtsklick auf den Knoten "MotionControl" das Kontextmenü und wählen Sie den Menüpunkt "Achsverbund hinzufügen". Klicken Sie anschließend auf "Technologieverbund".

JC440MC:

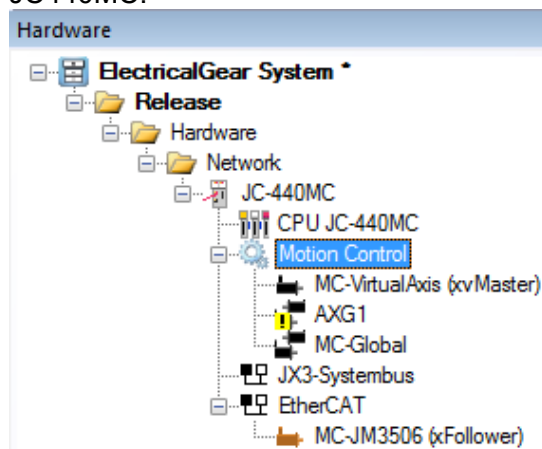


JC365MC:

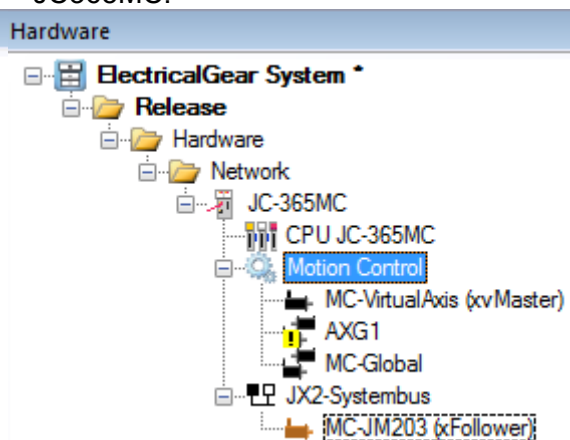


- Der neue Technologieverbund wird nun im Hardwarebaum angezeigt. Da der Technologieverbund noch nicht konfiguriert ist, wird er im Hardwarebaum mit einem Warnsymbol markiert. Sobald eine gültige Konfiguration vorliegt, verschwindet dieses Symbol wieder.

JC440MC:

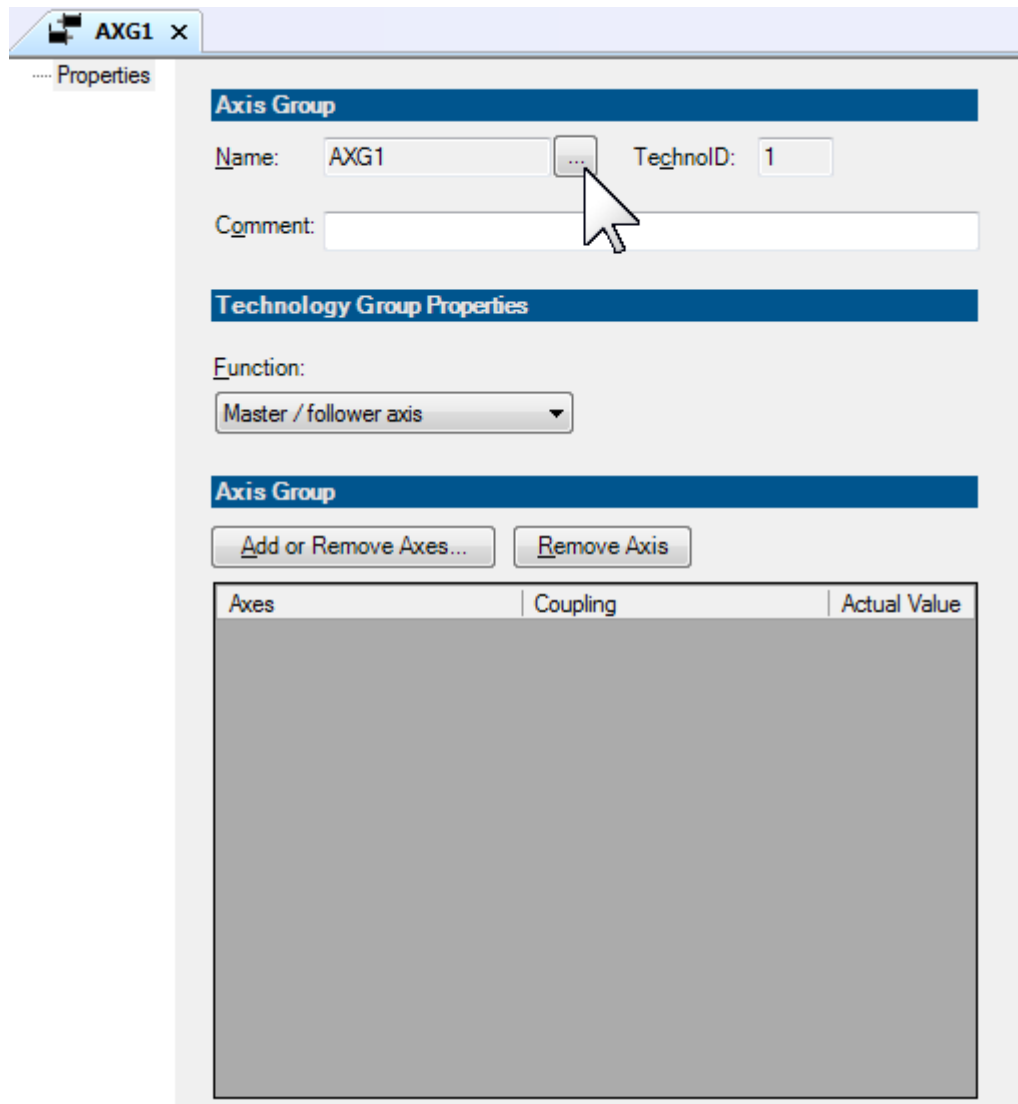


JC365MC:

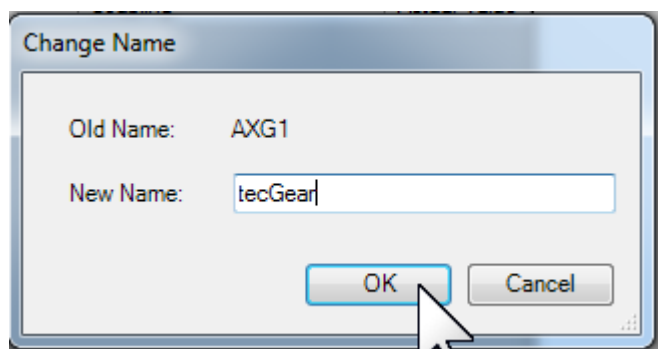


## 2.2 Konfigurieren des Technologieverbunds

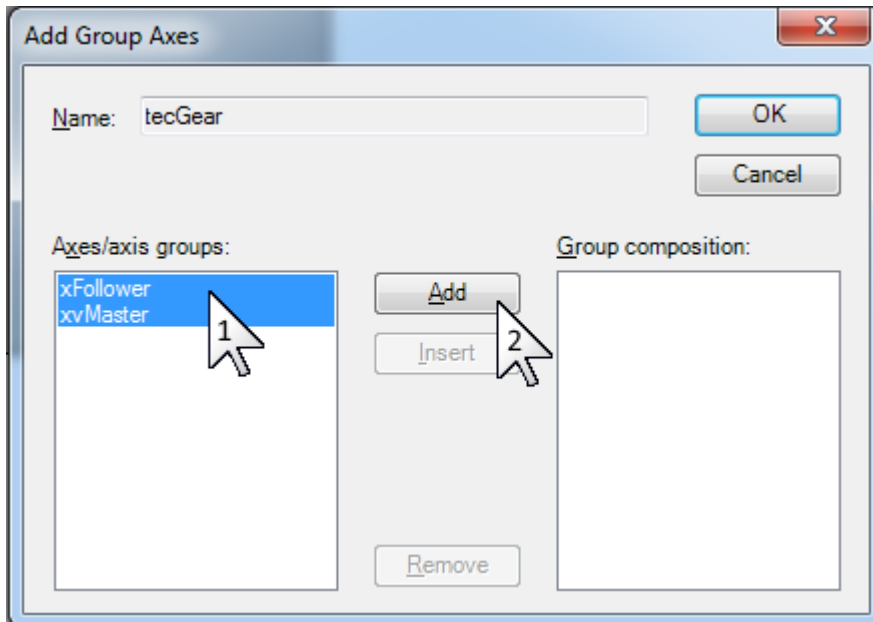
Durch einen Doppelklick auf den neu angelegten Verbund öffnen Sie die Einstellungen für den Technologieverbund.



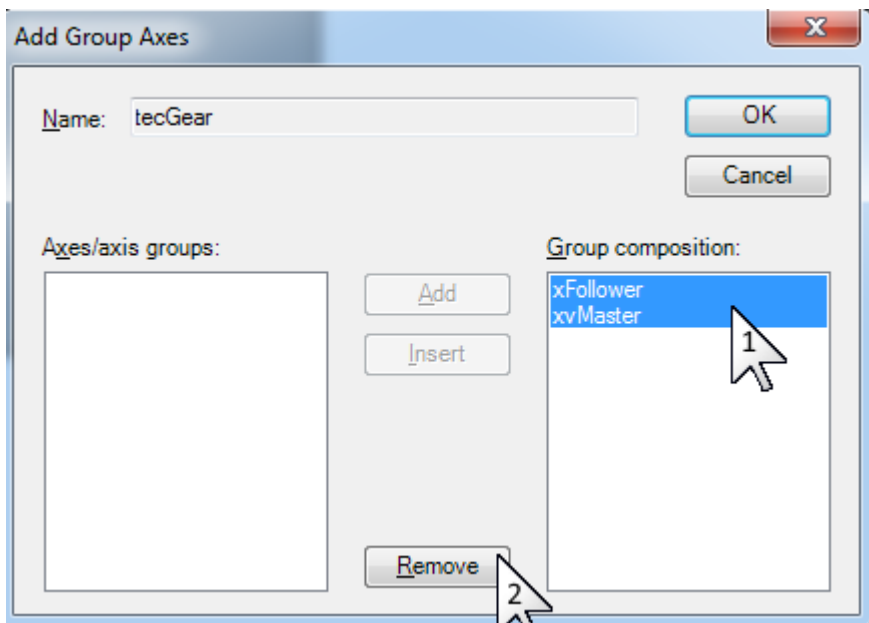
Durch Klick auf den Button "..." können Sie nun dem Technologieverbund einen verständlichen Namen vergeben. Innerhalb des STX-Projekts wird ein MotionAPI-Objekt mit diesem Namen erzeugt. Bestätigen Sie die Umbenennung durch JetSym, damit der neue Name auch im STX-Projekt übernommen wird.



Per "*Achse hinzufügen*" bestimmen Sie nun die Verbundachsen. Es öffnet sich folgender Dialog:



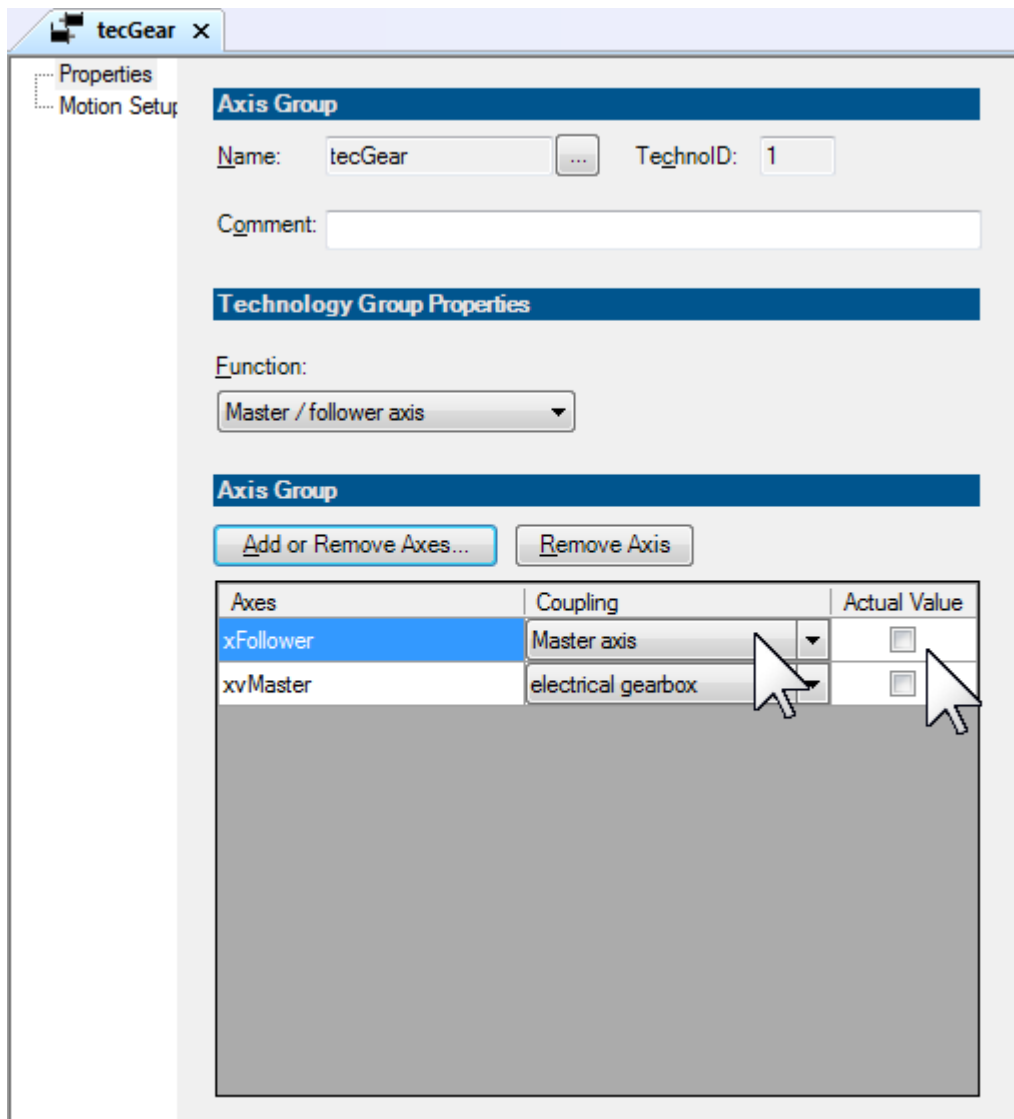
Markieren Sie die Achsen (1) und verschieben Sie diese per "*Hinzufügen (Add)*" oder per Doppelklick auf die jeweilige Achse auf die rechte Seite zu "*Verbundelemente (Group composition)*". Hier sehen Sie nur die Auflistung der Mitglieder im Verbund. Die Zuweisung der jeweiligen Aufgaben erfolgt in der übergeordneten Ansicht.



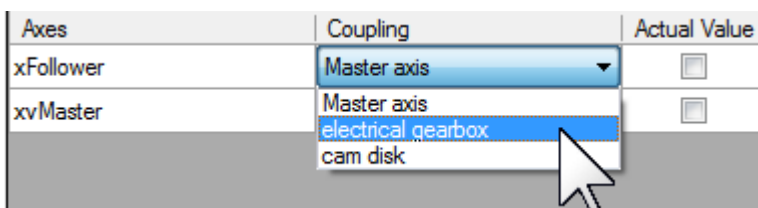
Sollten Sie wieder Achsen aus der „Verbundelemente (Group composition)“-Liste entfernen wollen, so können Sie diese markieren (1) und per "*Entfernen (Remove)*" (2) oder per Doppelklick aus der Liste löschen.

Bei der ersten Konfiguration eines Technologieverbunds wird die erste Achse in der Liste der Verbundachsen als Leitachse vorgeschlagen.

Bestätigen Sie die Auswahl mit "OK" (3).



Nun bestimmen Sie die Kopplungsart und ob eine Istwertkopplung verwendet werden soll. Im aktuellen Bild ändern Sie die Kopplung von "xFollower" auf "*Elektrisches Getriebe (electrical gearbox)*" und von "xvMaster" auf "*Masterachse (Master axis)*".



Damit ergibt sich folgende Konfiguration:

Axes	Coupling	Actual Value
xFollower	electrical gearbox	<input type="checkbox"/>
xvMaster	Master axis	<input type="checkbox"/>

### Beachten Sie:

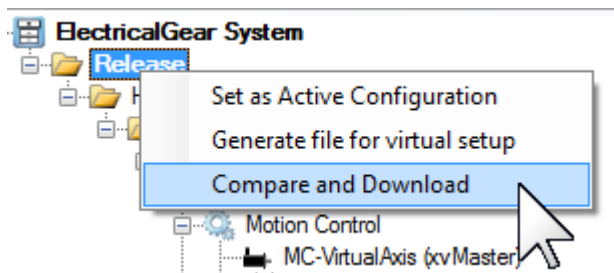
Die Leitachse muss nicht zwangsläufig an erster Position stehen. Die Reihenfolge ist beliebig.

Es darf nur eine Leitachse konfiguriert werden. JetSym wird Sie im Hardware-Baum durch ein Warnsymbol und einen Hinweis im Tooltip auf einen solchen Konfigurationsfehler hinweisen. Des Weiteren erscheint dieser Hinweis im Dialog "Vergleichen und Downloaden".



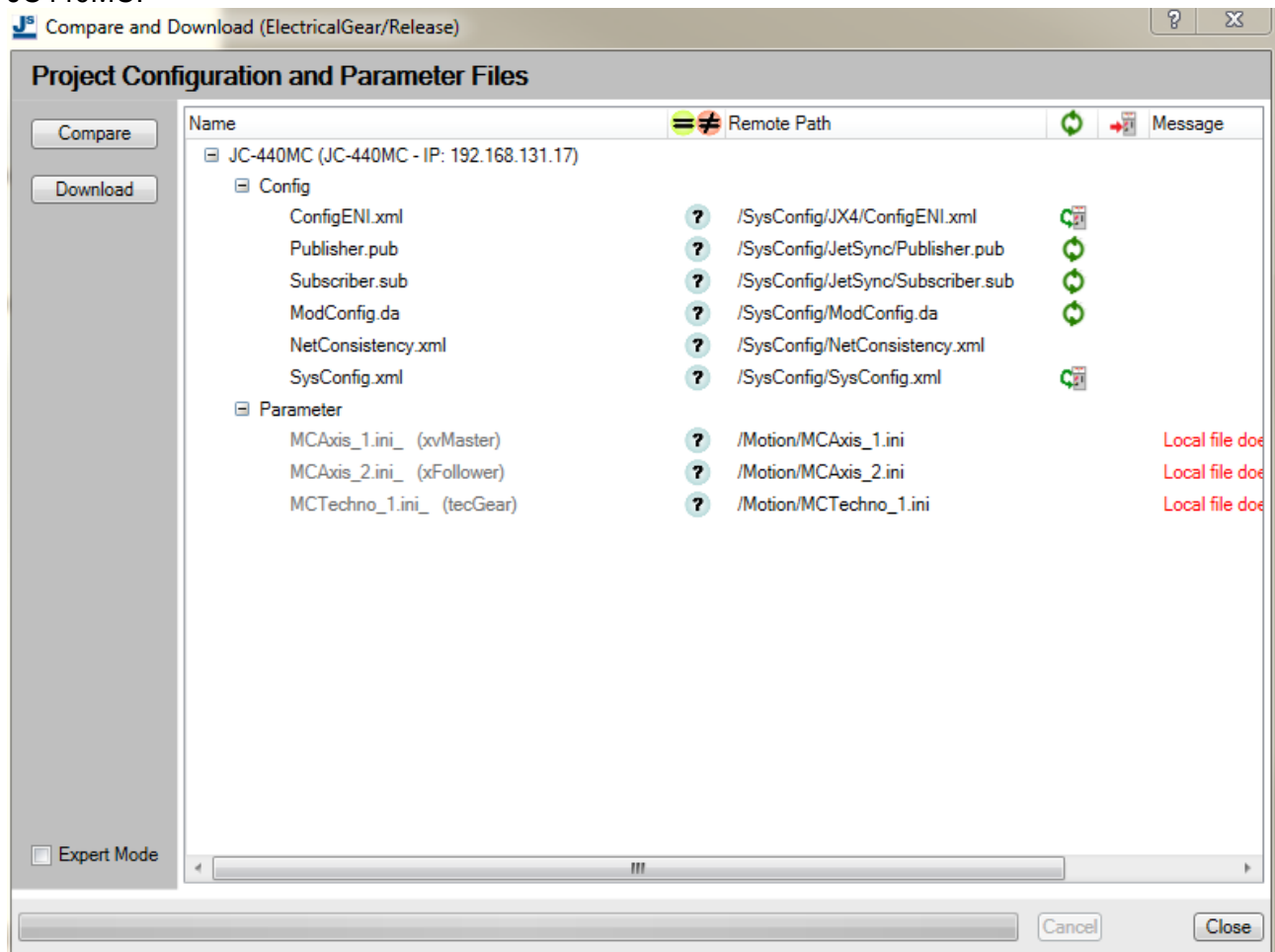
## 2.3 Downloaden der MC-Dateien auf Steuerung und Neustart

Durch das Erzeugen und Download der notwendigen MotionControl-Dateien wird nun der Steuerung die neue MotionControl-Konfiguration bekannt gemacht. Nach einem Reboot kann diese Konfiguration durch die Steuerung verwendet werden.

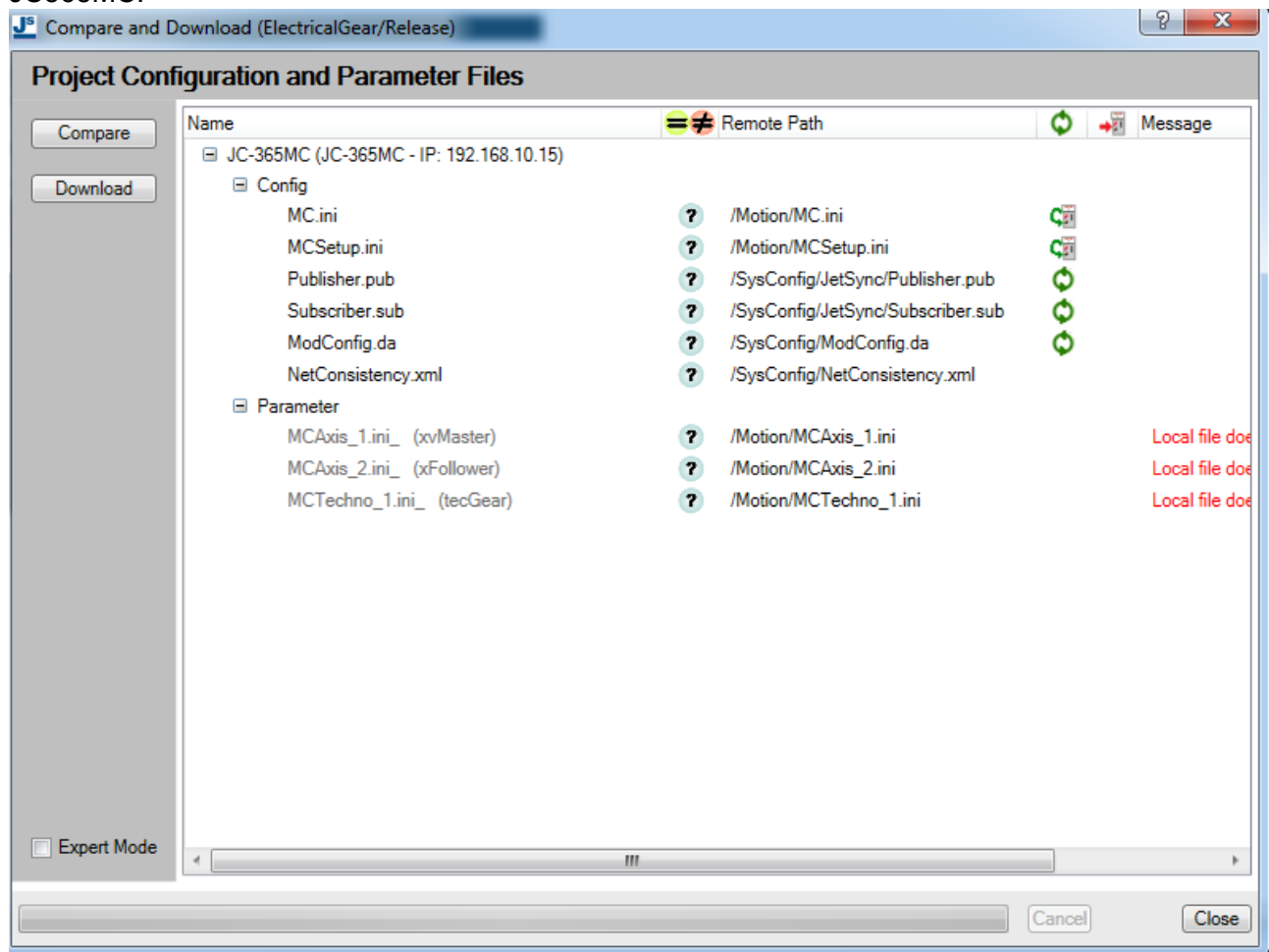


Wählen Sie im Kontextmenü des Konfigurationsordner (hier z.B. „Release“) *„Vergleichen und Downloaden (Compare and Download)“* aus.

JC440MC:



## JC365MC:



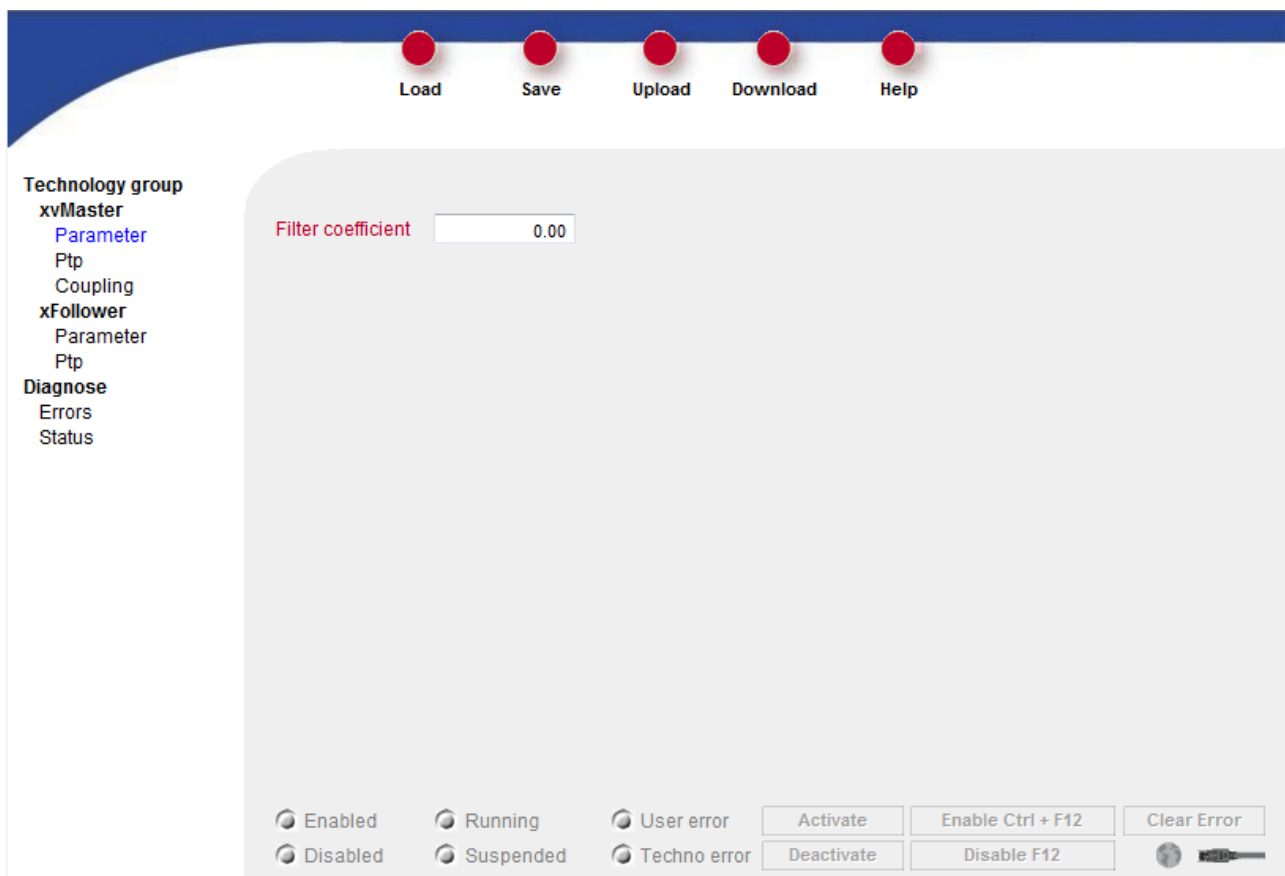
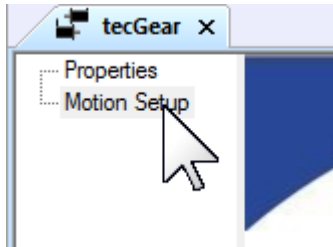
Im Dialog "Vergleichen und Downloaden" übertragen Sie die Hardware-Konfiguration per "Downloaden" auf die konfigurierte Steuerung.

Ist ein "Neustart" der Steuerung nötig, wird ein entsprechender Dialog angezeigt, in dem Sie diesen Neustart durchführen können.

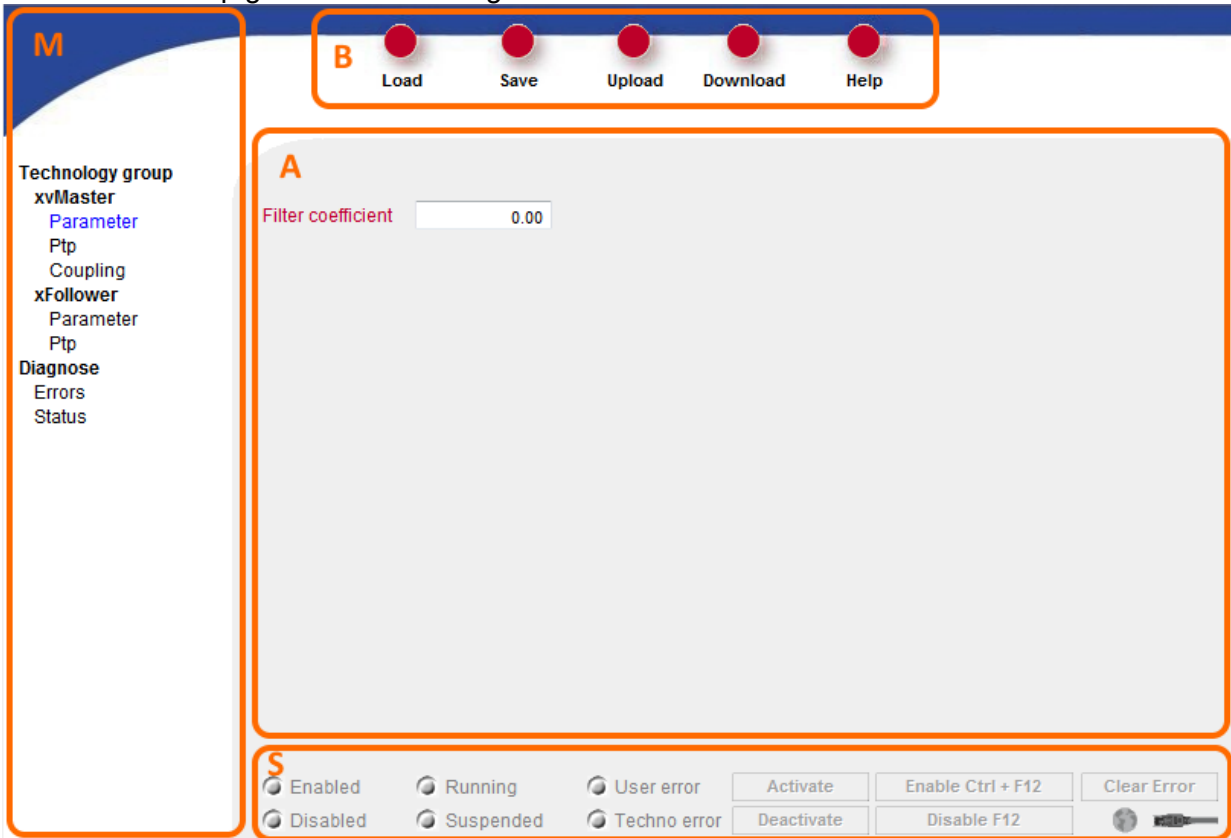
## 3 Anwenden des Technologieverbunds im MotionSetup

### 3.1 Bedienen des Technologieverbunds

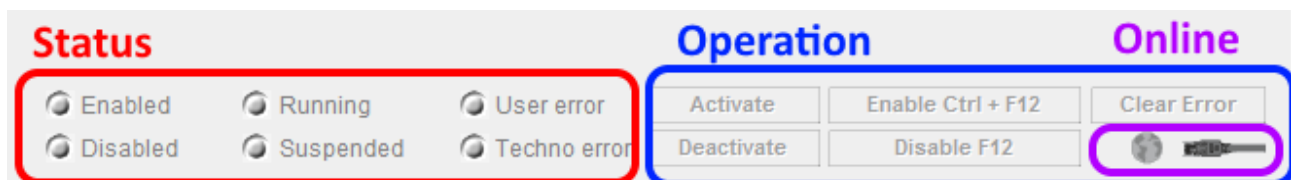
Öffnen Sie das Motion-Setup per Doppelklick auf den entsprechenden Technologieverbund. Wechseln Sie dann im Konfigurationsmenü auf "*Motion-Setup*".



Das Motion-Setup gliedert sich in folgende Bereiche:

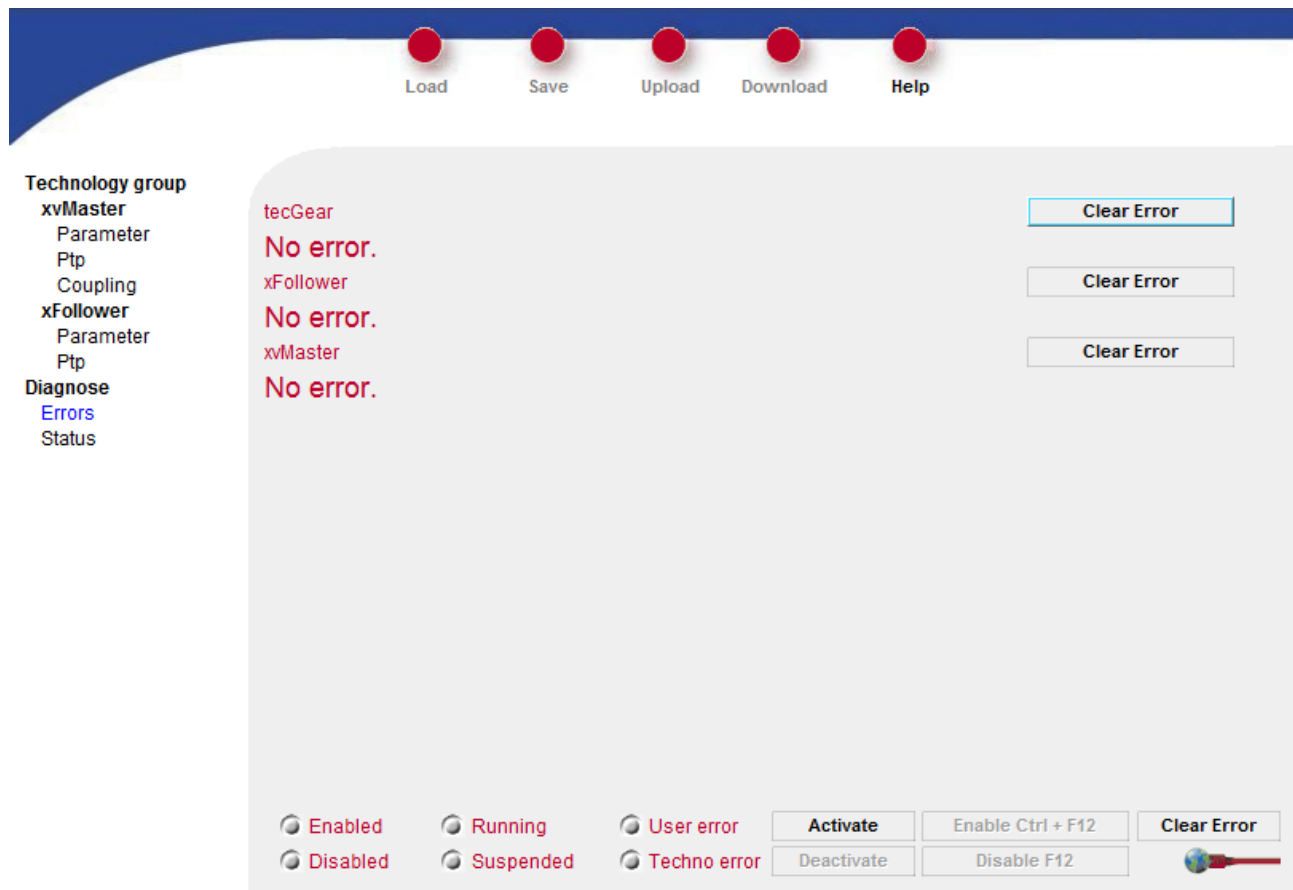


- M: Menü – Auswahl der einzelnen Inbetriebnahme-Seiten  
 B: Bedienleiste  
 A: Anzeigebereich  
 S: Status



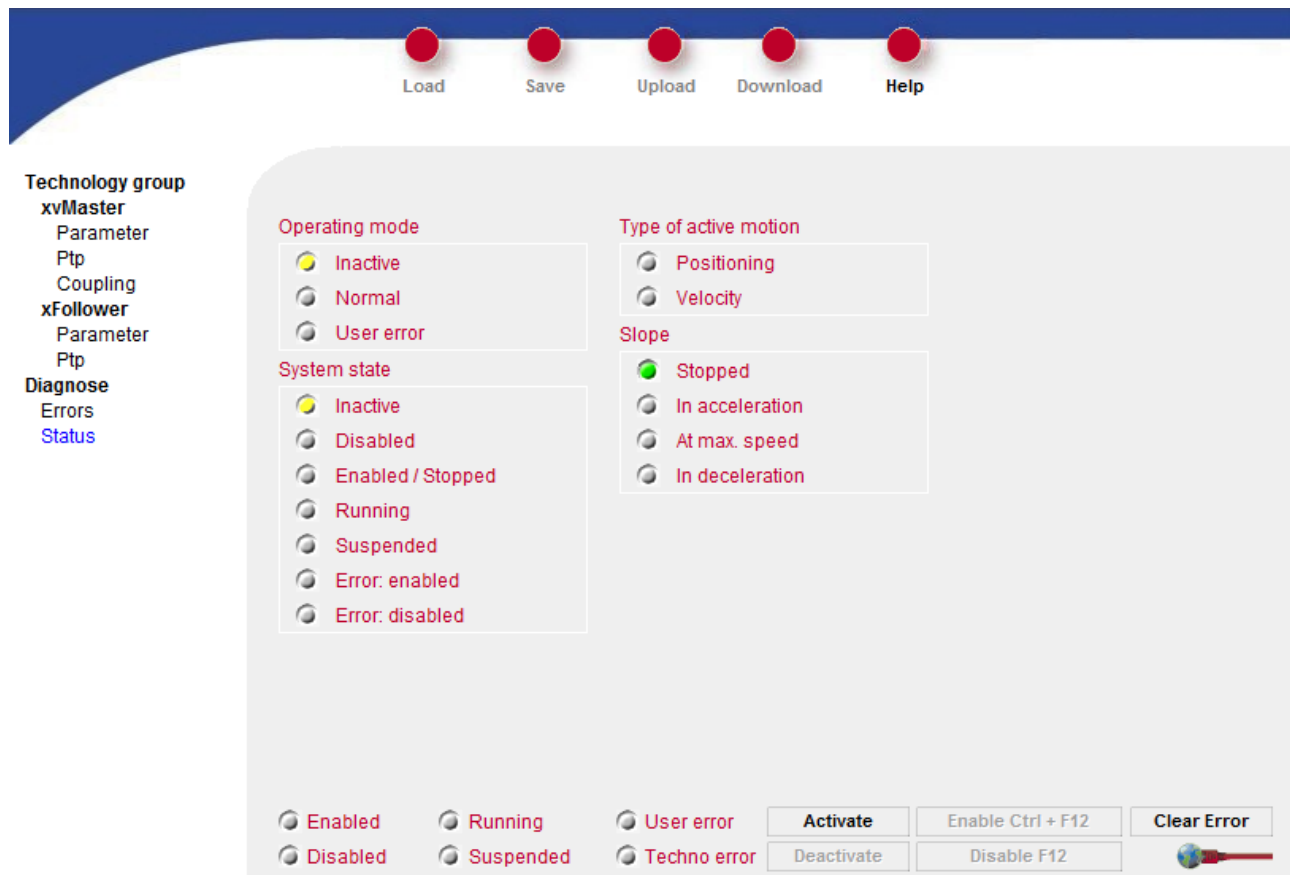
- Die Elemente erlauben die übergeordnete Bedienung des Technologieverbundes
  - Aktivieren/Deaktivieren (Activate/Deactivate) des Technologieverbundes
  - Freigabe/Sperren (Enable/Disable) (nur wenn Verbund aktiv und fehlerfrei)
  - Fehler löschen (Clear error): Löscht Fehler des Verbunds und der Mitgliedsachsen im aktiven Verbund
- Statusanzeige
  - Freigegeben (Enabled)
  - Gesperrt (Disabled)
  - Läuft (Running)
  - Unterbrochen (Suspended)
  - Ist der Verbund inaktiv, so sind die Status ausgegraut
- Fehler
  - Bedienfehler (User error)
  - Technologieverbundfehler (Techno error)
- Online
  - Anhand des Online-Symbols ist erkennbar, ob eine Verbindung mit der Steuerung besteht.
  - Offline:
  - Online:

### 3.1.1 Fehlerübersicht



Hier werden die anliegenden Fehler des Verbunds und der Mitgliedsachsen angezeigt. Zusätzlich zu der "Fehler löschen"-Taste in der Bedienleiste können hier die Fehler einzeln gelöscht werden.

### 3.1.2 Statusübersicht



Der Technologieverbund zeigt den Status für Betriebsart, Betriebszustand, Bewegungsart und Rampenstatus an.

Bewegungsart und Rampenstatus beziehen sich auf die Bewegung der Leitachse.

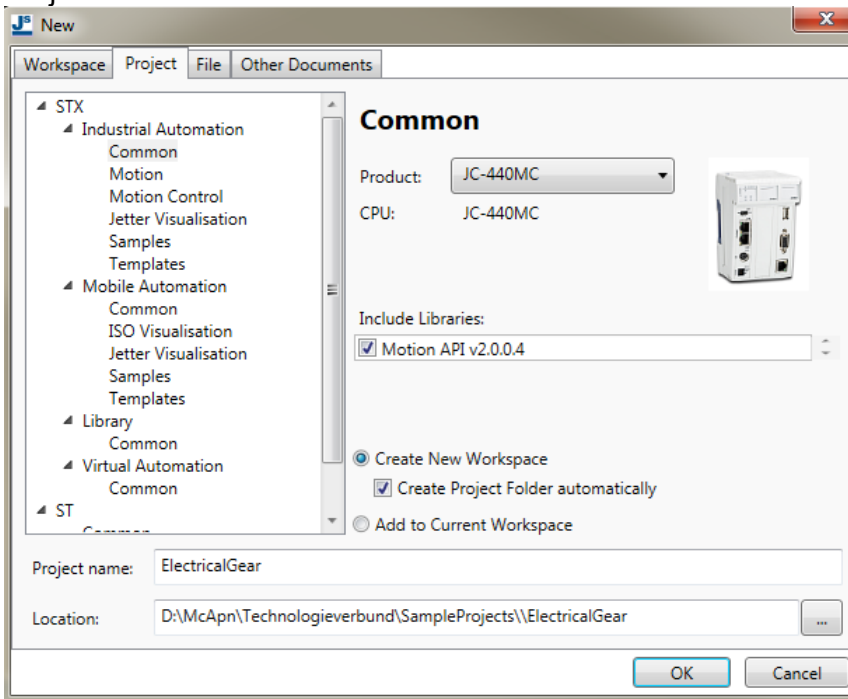
### 3.1.3 Weitere Ansichten

Weitere Seiten zum Einstellen und Testen des Technologieverbundes werden in den thematisch zugehörigen Application Notes erklärt.

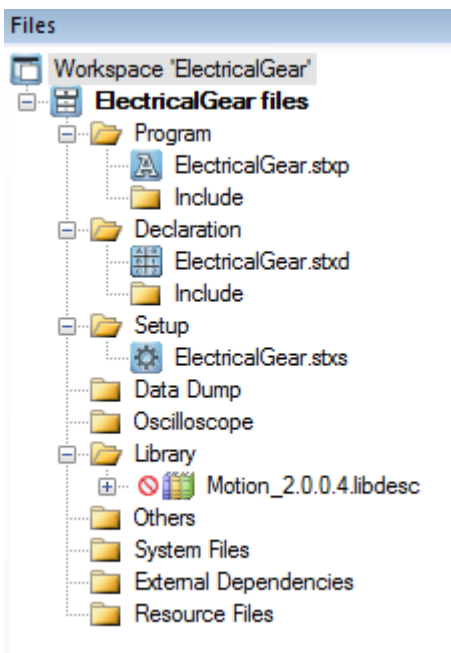
## 4 Anwenden des Technologieverbunds im Anwendungsprogramm

### 4.1 Einbinden der Motion API-Bibliothek in ein Projekt

Die erste Möglichkeit ist die direkte Auswahl der Motion API-Bibliothek beim Erstellen eines neuen Projekts.

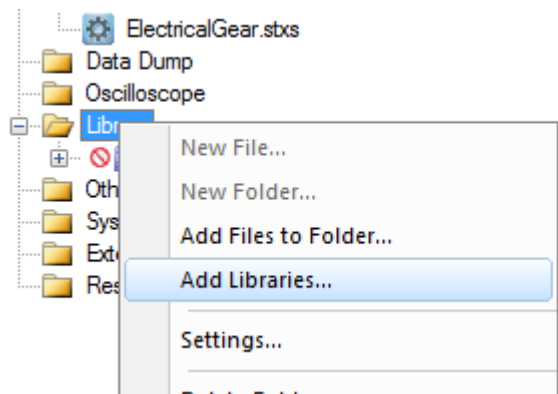


Die Bibliothek wird dann direkt im Projektbaum eingebunden und in das STX-Projekt inkludiert.



Durch das Anlegen der Hardwarekonfiguration und der Verwendung der Motion API wird automatisch eine "HardwareConfig.stxp" erstellt, welche durch die "Include"-Einstellungen in den Projekteinstellungen auch dem aktuellen Projekt zur Verfügung steht.

In dieser Programmdatei werden die Motion-Objekte instanziiert und die Listen des MC-Managers erstellt.



In dieser Ansicht können über das Kontextmenü des Ordners "Bibliothek" (Library) ebenfalls Bibliotheken geändert und eingebunden werden.

Zu einer Zeit kann nur eine MotionAPI-Bibliothek eingebunden sein. Soll diese geändert werden, so muss die bestehende Verbindung zuerst gelöscht und dann die gewünschte Bibliothek hinzugefügt werden. Das Löschen der Bibliothek in dieser Ansicht löscht nicht die Dateien oder Bibliotheken im Dateisystem!

Diese Datei wird beim Kompilieren erzeugt. Damit die Motion-Objekte im Programm mit Hilfe der IntelliSense im Programm verwendet werden können, muss das Programm einmalig durchkompiliert werden.

Damit fehlerfrei kompiliert werden kann, sollte bei einem neuen Projekt ohne vorhandenen Code dazu die folgende kurze Routine erstellt werden:

```
Task tMain autorun
End_Task;
```

Danach sind die Motion-Objekte in der IntelliSense eingetragen und Sie werden bei der Programmerstellung durch diese unterstützt.

## 4.2 Programm neu starten

Bei einem Programmstart unterscheiden wir zwei Szenarien:

- Booten der Steuerung
- Neustart der Applikation durch Download oder direkt durch JetSym

### 4.2.1 Rebooten der Steuerung

Beim Rebooten der Steuerung werden alle Bestandteile des Betriebssystems der Steuerung neu gestartet. Dies beinhaltet auch den MotionControl-Kern. Dieser erstellt aufgrund der Konfigurationsdateien MC.ini und MCSetup.ini die Motion-Objekte des MotionControl-Kerns. Des Weiteren stellt er die Verbindung zu den angeschlossenen Servoverstärkern her und synchronisiert diese. Anschließend werden die zu den MotionControl-Objekten zugehörigen Konfigurationsdateien geladen, so dass u.a. Regelparameter, Übersetzungsverhältnisse usw. beim Anwendungsstart zu Verfügung stehen.

Im Normalfall sind nach Abschluss der Bootphase alle MotionControl-Objekte initialisiert, synchronisiert und fehlerfrei. Achsverbunde sind inaktiv und reale, virtuelle Achsen sowie Simulationsachsen im Betriebszustand "Gesperrt". Dies ist die Standardausgangssituation von der das Programm ausgeht.

Besonderheiten:

- Boot-Fehler des MotionControl-Kerns
  - o Hardware-Konfiguration im MCSetup.ini entspricht nicht der Realität, z.B. wenn mehr Achsen konfiguriert wurden als angegeben oder die IP-Konfiguration des Servoverstärkers nicht korrekt ist.



- Synchronisation ist fehlgeschlagen z.B. durch falsches Betriebssystem der Schnittstellenkarten der Servoverstärker.
- Fehler der Servoverstärker
  - Häufige Fehler:
    - Geberfehler bei JM-200: Wird ein anderes Gebersystem als Resolver oder Hi-perface verwendet, so wird dies nicht automatisch durch den JM-200 erkannt und er zeigt initial einen Geberfehler. Dieser Fehler kann durch ClearError gelöscht werden, da in der MCAXisXX.ini schon der korrekte Geber eingetragen sein sollte.
    - Synchronisationsfehler F43 bei Ethernet-Synchronisation: Meistens erzeugt durch einen alleinigen Reboot der Steuerung ohne gleichzeitiges Neustarten der JM-200. Beim Reboot der Steuerung erkennen die JM-200 einen Verlust der Synchronisation, wodurch F43 erzeugt wird. Dieser Fehler kann durch ClearError gelöscht werden.

Bei Programmstart sollte diesen Besonderheiten Rechnung getragen werden, um einerseits in die Standardausgangssituation zu kommen und andererseits bei Boot-Fehlern den Benutzer zu informieren, damit dieser entsprechende Aktionen vornehmen kann.

#### 4.2.2 Neustart der Applikation

Beim Neustart der Applikation kann es eine Vorgeschichte der MotionControl-Objekte geben. Sei es durch ein vormals gelaufenes Applikationsprogramm oder durch die Verwendung der MotionControl-Objekte durch das MotionSetup.

Dies sollte in Standardapplikationen berücksichtigt werden. Da die eigentliche Anwendung von einer Standardausgangssituation starten soll, sollten die MotionControl-Objekte in diesen Zustand überführt werden.

Hierzu zählen Fehler löschen, Antriebe sperren und Verbunde deaktivieren.

##### *Rücksetzen der MotionControl-Objekte*

Eine Vorgehensweise zum Zurücksetzen der MotionControl-Objekte funktioniert sowohl nach dem Booten der Steuerung als auch beim Programmneustart.

Durch die Struktur der MotionControl ergibt sich eine grobe Reihenfolge, in der dies passieren kann.

- Rücksetzen der Verbunde:
  - Sperren( Disable)
  - Fehler löschen (ClearErrors)
  - Deaktivierung (Deactivate)
- Rücksetzen der Achsen
  - Sperren (Disable)
  - Fehler löschen (ClearErrors)

Es empfiehlt sich, Programmneustarts oder einen Reboot bei Maschinenstillstand vorzunehmen. In den nachfolgenden Vorschlägen wird ein Sperren mit Notstopprampe durchgeführt, welches zu einem harten Anhalten führt. Sollen laufende Achsen gezielt angehalten werden, muss das Prozedere entsprechend angepasst werden. Im einfachsten Fall reicht vor dem Sperren ein gezieltes "MoveHalt", erreicht aber aufgrund unterschiedlichster Anforderungen der Anwendungen schnell eine gewisse Komplexität, welche hier nicht allgemeingültig abgebildet werden kann.

Oft besteht auch der Wunsch, dass Parameter wieder zurückgesetzt werden sollen, welche zur Laufzeit oder durch MotionSetup verändert wurden. Hierzu ist nach dem "Fehler löschen" ein Laden der Parameter einzufügen.

#### 4.2.2.1 Neustart der Applikation - Vorschlag 1:

```
// Alle Technoverbunde zurücksetzen
if not tecGear.State.IsInactive then
  if not tecGear.State.IsDisabled or not tecGear.State.IsDisabledError then
    tecGear.State.Transitions.Disable(MCStateTransitionDisableModes.VelocityControlSlope);
    when tecGear.State.IsDisabled or tecGear.State.IsDisabledError continue;
  end_if;

  tecGear.Diagnostics.ClearErrors();
  when tecGear.State.IsDisabled continue;

  tecGear.Deactivate();
  when tecGear.State.IsInactive continue;
end_if;

// Alle Achsen zurücksetzen
if not xvMaster.State.IsDisabled or not xvMaster.State.IsDisabledError then
  xvMaster.Power.Quickstop();
when xvMaster.State.IsDisabled or tecGear.State.IsDisabledError continue;
end_if;

xvMaster.Diagnostics.ClearErrors();
when xvMaster.State.IsDisabled continue;

//Optional
xvMaster.Settings.LoadFromFile();
when xvMaster.State.IsDisabled continue;

if not xFollower.State.IsDisabled or not xFollower.State.IsDisabledError then
  xFollower.Power.Quickstop();
when xFollower.State.IsDisabled or xFollower.State.IsDisabledError continue;
end_if;

xFollower.Diagnostics.ClearErrors();
when xFollower.State.IsDisabled continue;

// Optional
xFollower.Settings.LoadFromFile();
when xFollower.State.IsDisabled continue;
```

In diesem Beispiel werden die Rücksetz-Routinen für jedes angelegte MotionControl-Objekt separat ausgeführt.

**Siehe auch Beispielprojekt:**

- [TechnologieStartUp1](#)

#### 4.2.2.2 Neustart der Applikation - Vorschlag 2:

```
function TechnoResetAll()
var
  nListIndex: int;
end_var;

for nListIndex := 0 to mcMgr.TechnoCount - 1 by 1 do
  if not mcMgr.TechnoList[nListIndex].State.IsInactive then
```

```

    if not mcMgr.TechnoList[nListIndex].State.IsDisabled or
       not mcMgr.TechnoList[nListIndex].State.IsDisabledError
    then
        mcMgr.TechnoList[nListIndex].State.Transitions.Disable(
            MCStateTransitionDisableModes.VelocityControlSlope);
        when mcMgr.TechnoList[nListIndex].State.IsDisabled or
            mcMgr.TechnoList[nListIndex].State.IsDisabledError continue;
    end_if;
    mcMgr.TechnoList[nListIndex].Diagnostics.ClearErrors();
    when mcMgr.TechnoList[nListIndex].State.IsDisabled continue;

    mcMgr.TechnoList[nListIndex].Deactivate();
    when mcMgr.TechnoList[nListIndex].State.IsInactive continue;
end_if;
end_for;
end_function;

function AxisResetAll()
var
    nListIndex: int;
end_var;

for nListIndex := 0 to mcMgr.AxisCount - 1 by 1 do
    mcMgr.AxisList[nListIndex].State.Transitions.Disable(
        MCStateTransitionDisableModes.VelocityControlSlope);
    when mcMgr.AxisList[nListIndex].State.IsDisabled or
        mcMgr.AxisList[nListIndex].State.IsDisabledError continue;
    mcMgr.AxisList[nListIndex].Diagnostics.ClearErrors();
    when mcMgr.AxisList[nListIndex].State.IsDisabled continue;
//Optional
    mcMgr.AxisList[nListIndex].Settings.LoadFromFile();
    when mcMgr.AxisList[nListIndex].State.IsDisabled continue;

end_for;
end_function;

```

Dieses Beispiel verwendet den MC-Manager, wodurch unabhängig von Namen und Menge der in der Hardware-Konfiguration angelegten MotionControl-Objekte die Rücksetz-Routinen ausgeführt werden. Diese Funktionen können Sie in weiteren Projekten als Standard-Funktionen einbinden.

**Siehe auch Beispielprojekt:**

- [TechnologieStartUp2](#)

## 4.3 Aktivieren des Technologieverbunds

Im Standardausgangszustand sind Verbunde inaktiv. Der Betriebszustand ist somit "Inactive", welcher mit "tecGroup.State.IsInactive" abgefragt werden kann.

Das Aktivieren des Technologieverbundes erfolgt mit "tecGroup.Activate". Die erfolgreiche Aktivierung kann je nach vorherigem Zustand der Mitgliedsachsen vor der Aktivierung mit "Disabled" oder "Enabled" als entsprechenden Zustandswechsel abgefragt werden.

Der Verbund erhält den Betriebszustand „Enabled“ oder „Disabled“ entsprechend der Zustände der Mitgliedsachsen vor der Aktivierung. Haben nicht alle zustandsbehafteten Mitgliedsachsen den gleichen Zustand, so wird bei der Aktivierung ein Fehler angezeigt.

Zustandsbehaftete Achsen sind solche, die einen anderen Zustand als „Inactive“ annehmen können. Dies sind virtuelle und reale sowie Simulationsachsen.

Externe Achsen oder Mitlaufachsen haben immer den Zustand "Inactive" und bleiben unberücksichtigt.

#### Beachte:



Ein Verbund lässt sich nur aktivieren, falls die zustandsbehafteten Mitgliedsachsen im Zustand "Enabled" oder "Disabled" sind. Befindet sich z.B. mindestens eine zustandsbehaftete Mitgliedsachse im Zustand "Running" ist eine Aktivierung nicht möglich.

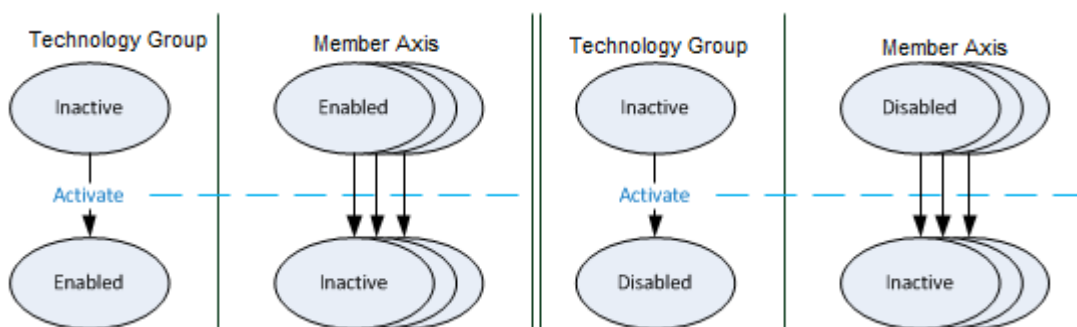
#### Beispiel:

```
xvMaster.Power.Enable();
xFollower.Power.Enable();
when xvMaster.State.IsEnabled continue;
when xFollower.State.IsEnabled continue;

// Aktivieren des Verbunds
tecGroup.Activate();
// Warten, dass Aktivierung erfolgreich
when tecGroup.State.IsEnabled continue;
```

Zur Vereinfachung wird im Prozess eine erfolgreiche Aktivierung vorausgesetzt. Die Handhabung von Fehlern kann in einem separaten Task durchgeführt werden. Typischer Fehler ist die Aktivierung des Verbunds bei unterschiedlichen Betriebszuständen der zustandsbehafteten Mitgliedsachsen.

Nach der Aktivierung des Verbunds wechselt der Betriebszustand der zustandsbehafteten Mitgliedsachsen auf "Inactive".



*Zustandsübergänge bei Aktivierung des Technologieverbundes*

## 4.4 Deaktivieren des Technologieverbunds

Das Deaktivieren des Technologieverbunds erfolgt mit "tecGroup.Deactivate". Die erfolgreiche Deaktivierung kann überprüft werden, indem auf "Inactive" abgefragt wird. Alternativ reicht es aus, den Betriebszustand der zustandsbehafteten Mitgliedsachsen auf "Enabled" oder "Disabled" abzufragen.

Die zustandsbehafteten Mitgliedsachsen wechseln von "Inactive" auf "Enabled", falls sich der Verbund vor der Deaktivierung im Zustand "Enabled" befindet.

Die zustandsbehafteten Mitgliedsachsen wechseln von "Inactive" auf "Disabled", falls sich der Verbund vor der Deaktivierung im Zustand "Disabled" befindet.

Beachte: Ein Verbund lässt sich nur deaktivieren, wenn er sich im Zustand "Enabled" oder "Disabled" befindet.

Beispiel "Disabled":

```
when tecGroup.State.IsDisabled continue;
// Deaktivieren des Verbunds
tecGroup.Deactivate();
// Warten, dass Deaktivierung erfolgreich
when tecGroup.State.IsInactive continue;
```

Beispiel "Enabled":

```
when tecGroup.State.IsEnabled continue;
// Deaktivieren des Verbunds
tecGroup.Deactivate();
// Warten, dass Deaktivierung erfolgreich
when xvMaster.State.IsEnabled continue;
when xFollower.State.IsEnabled continue;
```

Nach dem Deaktivieren des Verbunds wechselt der Betriebszustand des Verbunds auf "Inactive".

*Zustandsübergänge bei Deaktivierung des Technologieverbundes:*

Siehe auch Beispielprojekt:

- [TechnologyActivateDeactivate](#)

## 4.5 Positionierung

Beim Positionieren einer Achse ist zu beachten, ob sie als Einzelachse oder als Mitgliedsachse im Verbund aktiv ist.

Bei einer Einzelachse werden die Positionierfunktionen direkt am Achs-Objekt aufgerufen.

Beispiel:

```
xvMaster.MovePtp.Start(AxisPositioningModes.AbsModuloAuto, 100.0, 200.0, 1000.0, 1000.0, 1.0);
xvMaster.MoveVelocity.Start(Directions.Positive, 100.0, 200.0, 300.0);
xvMaster.MoveHalt.Start(MCAxisHaltModes.Normal, 1000.0);
xvMaster.MoveHome.Start();
xvMaster.MoveHome.SetReference();
```

Ist der Verbund aktiv, werden die Positionierfunktionen über den Verbund aufgerufen:

Beispiel:

```
tecGear.MovePtp.Start(xvMaster, MCTechnoPositioningModes.AbsModuloAuto, 100.0, 200.0, 1000.0, 1000.0, 1.0);
tecGear.MoveVelocity.Start(xvMaster, Directions.Positive, 100.0, 200.0, 300.0);
tecGear.MoveHalt.Start(xvMaster, MCTechnoHaltModes.Normal, 1000.0);
```

Werden Positionierfunktionen an die Einzelachse abgesetzt, obwohl diese in einem Verbund aktiv ist, so wird der Fehler "10 - Objekt inaktiv" generiert, der darauf hinweist, dass hier auf ein inaktives Objekt zugegriffen wurde.

**Siehe auch Beispielprojekte:**

- [TechnologyMoveInGroup](#)
- [TechnologyMoveNoGroup](#)

#### 4.5.1 MovePtp

*Einzelachse:*

Bei der Abfrage, ob eine Positionierung beendet ist, gibt es zwei Möglichkeiten:

- Abfrage des Betriebszustands: `Achse.State.IsEnabled`
  - Bei Ausführung der Positionierung wechselt der Betriebszustand der Achse auf "Running", welcher bei Beendigung wieder auf "Enabled" zurückwechselt.
- Abfrage des Rampenstatus
  - Bei der Ausführung der Positionierung wechseln die Rampenstatus von "Accelerating", "ConstantSpeed", "Decelerating" auf "Stopped", sobald die Achse gestoppt ist.
  - Beschleunigen: `Achse.Mechanism.Slope.IsAccelerating`
  - Konstante Geschwindigkeit: `Achse.Mechanism.Slope.IsAtConstantSpeed`
  - Verzögern: `Achse.Mechanism.Slope.IsDecelerating`
  - Angehalten: `Achse.Mechanism.Slope.IsStopped`

Diese Abfragen beziehen sich alle auf den Sollwert der Achse.

Ob auch die Istposition das Ziel erreicht hat, sollte noch zusätzlich über das Zielfenster mit `Achse.IsInTargetWindow` abgefragt werden.

Eine Achse, die z.B. durch ein `MoveHalt` vorzeitig gestoppt wird, wechselt in den Betriebszustand "Enabled" und den Rampenstatus "Stopped", hat aber u.U. das Ziel nicht erreicht!

*Verbundachse:*

Bei der Abfrage, ob eine Positionierung beendet ist, sollte nur der Rampenstatus und ggf. das Zielfenster der jeweiligen Achse berücksichtigt werden. Der Betriebszustand des Verbundes ist eine Summe der Betriebszustände der Einzelachsen. Sobald sich ein Mitglied bewegt, wechselt der Betriebszustand auf "Running" und nur wenn alle Mitglieder stehen, wird der Zustand "Enabled" erreicht. Es kann deshalb je nach Anwendung mehrdeutig sein.

*Rampenstatus der Leitachse:*

Der Verbund selber besitzt auch einen Rampenstatus. Dies ist der Rampenstatus der Leitachse! Deshalb muss hier bei der Rampenstatusabfrage auch dieser berücksichtigt werden, wenn die Leitachse positioniert wird. Der Rampenstatus der Leitachse als Einzelachse bleibt während des aktiven Verbunds immer auf "Stopped" und kann somit nicht genutzt werden.

**Beispiel:**

```
// Leitachse
tecGear.MovePtp.Start(xvMaster, MCTechnoPositioningModes.AbsModuloAuto, 100.0, 200.0, 1000.0,
1000.0, 1.0);
when tecGear.Mechanism.Slope.IsStopped and xvMaster.IsInTargetWindow continue;

// Folgeachse
tecGear.MovePtp.Start(xFollower, MCTechnoPositioningModes.AbsModuloAuto, 100.0, 200.0, 1000.0,
1000.0, 1.0);
when xFollower.Mechanism.Slope.IsStopped and xFollower.IsInTargetWindow continue;
```

## 4.5.2 MoveVelocity

### Einzelachse:

Zur Abfrage, dass die Endlospositionierung gestartet wird, kann auf den Betriebszustand "Running" geprüft werden. In diesen Zustand wechselt die Achse, sobald sie mit der Beschleunigung beginnt.

Die Abfrage der Phasen einer Endlospositionierung wird mit dem Rampenstatus geprüft:

- Achse.Mechanism.Slope.IsAccelerating
- Achse.Mechanism.Slope.IsAtConstantSpeed

### Beispiel:

```
xvMaster.MoveVelocity.Start(Directions.Positive, 100.0, 200.0, 300.0);
when xvMaster.Mechanism.Slope.IsAtConstantSpeed continue;
```

### Verbundachse:

Wie bei MovePtp sollte hier der Rampenstatus wie bei der Einzelachse verwendet werden.

Der Betriebszustand des Verbunds ist eine Summe der Betriebszustände der Einzelachsen. Sobald sich ein Mitglied bewegt, wechselt der Betriebszustand auf "Running" und nur wenn alle Mitglieder stehen, wird der Zustand "Enabled" erreicht. Der Zustand kann deshalb je nach Anwendung mehrdeutig sein.

### Rampenstatus der Leitachse:

Der Verbund selber besitzt auch einen Rampenstatus. Dies ist der Rampenstatus der Leitachse! Deshalb muss hier bei der Rampenstatusabfrage auch dieser berücksichtigt werden, wenn die Leitachse positioniert wird. Der Rampenstatus der Leitachse als Einzelachse bleibt während des aktiven Verbunds immer auf "Stopped" und kann somit nicht genutzt werden.

### Beispiel:

```
// Leitachse
tecGear.MoveVelocity.Start(xvMaster, Directions.Positive, 100.0, 200.0, 300.0);
when tecGear.Mechanism.Slope.IsAtConstantSpeed continue;

// Folgeachse
tecGear.MoveVelocity.Start(xFollower, Directions.Positive, 100.0, 200.0, 300.0);
when xFollower.Mechanism.Slope.IsAtConstantSpeed continue;
```

### 4.5.3 MoveHalt

#### Einzelachse:

Bei der Abfrage, ob das Stoppen beendet ist, gibt es zwei Möglichkeiten:

- Abfrage des Betriebszustands: `Achse.State.IsEnabled`
  - o Bei Ausführung der Positionierung wechselt der Betriebszustand der Achse auf "Running", welcher bei Beendigung wieder auf "Enabled" zurückwechselt.
- Abfrage des Rampenstatus
  - o Beim Stoppen wechseln die Rampenstatus von "Accelerating", "ConstantSpeed", "Decelerating" auf "Stopped", sobald die Achse gestoppt ist.
  - o Beschleunigen: `Achse.Mechanism.Slope.IsAccelerating`
  - o Konstante Geschwindigkeit: `Achse.Mechanism.Slope.IsAtConstantSpeed`
  - o Verzögern: `Achse.Mechanism.Slope.IsDecelerating`
  - o Angehalten: `Achse.Mechanism.Slope.IsStopped`

#### Beispiel:

```
xvMaster.MoveHalt.Start(MCAxisHaltModes.Normal, 1000.0);
when xvMaster.Mechanism.Slope.IsStopped continue;
```

#### Verbundachse:

Bei der Abfrage, ob das Stoppen beendet ist, sollte nur der Rampenstatus berücksichtigt werden.

Der Betriebszustand des Verbunds ist eine Summe der Betriebszustände der Einzelachsen. Sobald sich ein Mitglied bewegt, wechselt der Betriebszustand auf "Running" und nur wenn alle Mitglieder stehen, wird der Zustand "Enabled" erreicht. Der Zustand kann deshalb je nach Anwendung mehrdeutig sein.

#### Rampenstatus der Leitachse:

Der Verbund selber besitzt auch einen Rampenstatus. Dies ist der Rampenstatus der Leitachse! Deshalb muss hier bei der Rampenstatusabfrage auch dieser berücksichtigt werden, wenn die Leitachse positioniert wird. Der Rampenstatus der Leitachse als Einzelachse bleibt während des aktiven Verbunds immer auf "Stopped" und kann somit nicht genutzt werden.

#### Beispiel:

```
// Leitachse
tecGear.MoveHalt.Start(xvMaster, MCTechnoHaltModes.Normal, 1000.0);
when tecGear.Mechanism.Slope.IsStopped continue;

// Folgeachse
tecGear.MoveHalt.Start(xFollower, MCTechnoHaltModes.Normal, 1000.0);
when xFollower.Mechanism.Slope.IsStopped continue;
```



#### 4.5.4 MoveHome

*Einzelachse:*

*MoveHome.Start*

Die Referenzfahrt wird wie gewohnt mit MoveHome.Start gestartet.

Bei der Abfrage, ob die Referenzfahrt beendet ist, sollte der Betriebszustand und der Referenzstatus verwendet werden

- Abfrage des Betriebszustands: Achse.State.IsEnabled
  - o Bei Ausführung der Positionierung wechselt der Betriebszustand der Achse zu "Running", welcher bei Beendigung wieder zu "Enabled" zurückwechselt.
- Abfrage des Referenzfahrtstatus
  - o Bei der Referenzfahrt wechseln die Status von "ReferenceRunStarted", "ReferenceFound", "SwitchFound" zu "ReferenceSet".
  - o Achse.MoveHome.IsReferenceRunStarted
  - o Achse.MoveHome.IsSwitchFound: Schalter gefunden
  - o Achse.MoveHome.IsReferenceFound: Referenz gefunden
  - o Achse.MoveHome.IsReferenceSet: Ende der Referenzfahrt

Beispiel:

```
xFollower.MoveHome.Start();  
when xFollower.State.IsEnabled and xFollower.MoveHome.IsReferenceSet continue;
```

*MoveHome.SetReference()*

Wird nur die Referenz gesetzt, reicht es aus, den Referenzstatus "ReferenceSet" abzufragen.

Beispiel:

```
xvMaster.MoveHome.SetReference();  
when xvMaster.MoveHome.IsReferenceSet continue;
```

Hinweis:

Eine ausführliche Beschreibung der Referenzfahrt finden Sie in der Dokumentation der JM-200 Servoverstärker.

*Verbundachse:*

Im Verbund sind die Referenzfahrt und das Referenzsetzen nicht verfügbar.

Deshalb sollten alle Achsen, die Mitglied in einem Verbund sind, vor Aktivierung des Verbunds referenziert werden, wenn dies erforderlich ist.

## 4.6 Diagnose

### Hinweis:

Weitere Informationen über die Diagnose von Fehlern sowie Warnungen und Meldungen finden Sie in der Application Note 054 "Motion-Control Basics".

### Einzelachse

Der Zugriff auf die Informationen für die Diagnose einer Einzelachse erfolgt auch über das Achsobjekt mit dem entsprechenden Namen. Fehlerabfrage und Fehlerlöschen erfolgen über die jeweils verfügbaren Funktionen und Eigenschaften.

### Beispiel:

```
// Liegt ein Fehler an der Achse an?
if xFollower.Diagnostics.IsErrorPending then
    // Lösche den Fehler der Achse
    xFollower.Diagnostics.ClearErrors();
    // Warten, bis Fehler gelöscht
    when xFollower.State.IsDisabled continue;
end_if;
```

### Technologieverbund

Im Technologieverbund erfolgen die Abfrage und das Fehlerlöschen analog zur Einzelachse. Hat eine Verbundachse im aktiven Verbund einen Fehler, z.B. Schleppfehler, so führt dies auch zu einem Fehlerzustand des Technologieverbunds. Wird nun der Fehler des Technologieverbunds gelöscht, so werden auch die Fehler der Verbundachsen gelöscht.

### Beispiel:

```
// Liegt ein Fehler am Technologieverbundes an?
if tecGear.Diagnostics.IsErrorPending then
    // Lösche den Fehler des Technologieverbundes
    tecGear.Diagnostics.ClearErrors();
    // Warten, bis Fehler gelöscht
    when tecGear.State.IsDisabled continue;
end_if;
```

Ist der Fehlerzustand der jeweiligen Verbundachse von Interesse, so kann über das Einzelachsobjekt gegangen werden. Ein Fehlerlöschen führt zum Zurücksetzen des Fehlers der jeweiligen Einzelachse. Der Fehler des Technologieverbunds muss aber weiterhin explizit gelöscht werden.

### Beispiel:

```
// Liegt ein Fehler des Technologieverbunds an?
if tecGear.Diagnostics.IsErrorPending then
    // Liegt ein Fehler der Folgeachse an?
    if xFollower.Diagnostics.IsErrorPending then
        // Lösche den Fehler der Folgeachse
        xFollower.Diagnostics.ClearErrors();
        // Warten, bis Fehler gelöscht
        when not xFollower.Diagnostics.IsErrorPending continue;
    end_if;

    // Lösche den Fehler des Technologieverbunds
    tecGear.Diagnostics.ClearErrors();
end_if;
```

```
// Warten, bis Fehler gelöscht  
when tecGear.State.IsDisabled continue;  
end_if;
```



### ***Hinweis***

In den Beispielen wird davon ausgegangen, dass die Fehler löschar sind. Es ist aber in der Praxis auch möglich, dass Fehler weiterhin anstehen und das Löschen der Fehler scheitert. Ist z.B. ein realer Geber einer Achse defekt, so bleibt er auch nach dem Fehlerlöschen defekt. Obige Abfragen würden nun an der Wartebedingung hängen bleiben.

Jetter AG  
Gräterstraße 2  
71642 Ludwigsburg  
[www.jetter.de](http://www.jetter.de)

E-Mail [info@jetter.de](mailto:info@jetter.de)  
Telefon +49 7141 2550-0

We automate your success.