mapp Services

mapp Services stellt die Infrastruktur für Maschinen her.

1 mapp AlarmX: Alarmverwaltung

mapp AlarmX sammelt und verwaltet mapp, sowie Benutzeralarme. Die Alarme werden über Automation Studio konfiguriert, in der Applikation vewaltet und dann über eine beliebige Visualisierung dargestellt, bzw. als Datei exportiert.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp AlarmX (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal sind viele Tutorials zum Thema zu finden.

1.1 Konzept

In diesem Abschnitt ist das Konzept von mapp AlarmX erklärt:

Allgemein:

- Als Maschinenbediener möchte ich informative und einfache Alarmmeldungen in meiner eigenen Sprache
- Als Applikationist möchte ich dasselbe Alarmsystem verwenden, welches mapp auch intern verwendet
- Als Applikationist möchte ich alle Funktionen, welche über die Visualisierung vorhanden sind auch über meine Applikation zur Verfügung haben. Zum Beispiel möchte ich in der Lage sein alle Alarme auch über meine Applikation zu quittieren
- Als Applikationist möchte ich in der Lage sein unterschiedlichste Informationen an meinen Alarm anzuhängen

Definieren des Alarmverhaltens:

- Als Maschinenbediener möchte ich bei der Anzeige von mehreren Alarmen immer eindeutig sehen, welche die wichtigsten bzw. kritischsten aktiven Alarme sind
- Als Servicetechniker möchte ich Zugang zu detailierteren Alarminformationen als der Maschinenbediener. Dazu zählt die Alarmquelle (z.B. Förderband, Heizstation, ...), mögliche Fehlernummern, Links zu weiteren Informationen, ...)

- Als Applikationist möchte ich, dass sich meine Alarme intelligent Verhalten. Zum Beispiel möchte ich, dass meine Alarme unterschiedliche Reaktionen auslösen, nach oben eskaliert werden können oder zusammengefasst werden können
- Als Applikationist möchte ich Alarm auf unterschiedlichen Levels darstellen können. Ich möchte es dem Maschinenbediener ermöglichen den Alarm für einen bestimmten Maschinenteil aufzulösen bevor der Alarm weitergereicht wird

Alarmhistorie:

- Als Maschinenbediener möchte ich die Alarmhistorie exportieren und weiterverarbeiten können
- Als Servicetechniker möchte ich Zugang zur Alarmhistorie, um besser zu verstehen wie die Maschine verwendet wurde

Alarme visualisieren:

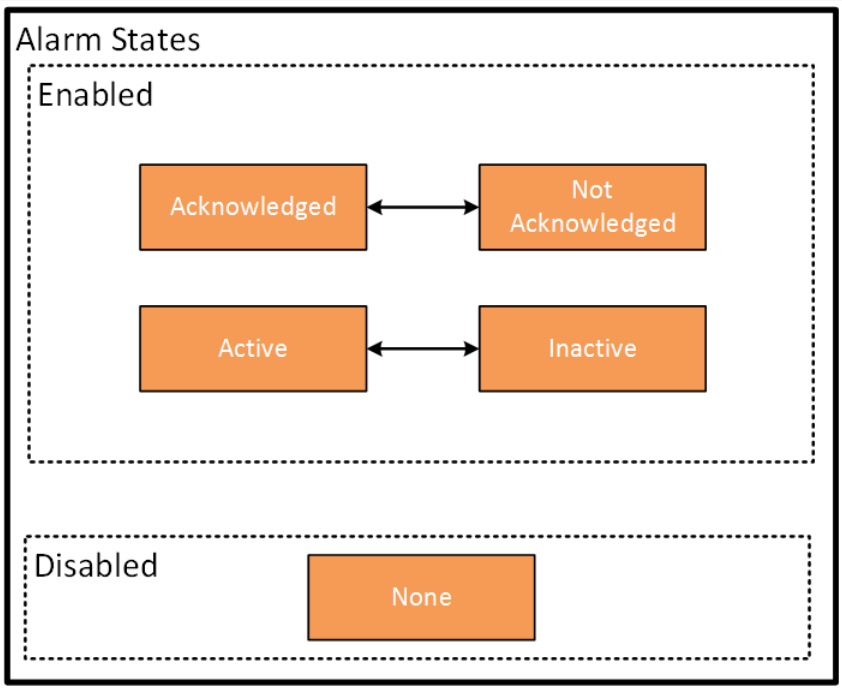
- Als Maschinenbediener oder Servicetechniker möchte ich die Alarme sowie detaillierte Alarminformation an meiner Maschinenvisualisierung einsehen. Das könnte über einen informativen Alarmtext, ein Video oder eine weiterführende Referenz (PDF, HTML,...) abgedeckt werden

1.1.1 Alarm Grundkonzept

In diesem Kapitel wird das Grundkonzept der Alarmverwaltung beleuchtet. Es wird erklärt welche Zustände ein Alarm annehmen kann und wie sich diese Zustände zueinander verhalten. Außerdem wird erklärt was der Unterschied zwischen Fehler und Alarm ist, sowie was der Unterschied zwischen einem mapp Alarm und einem benutzerdefinierten Alarm ist.

Grundzustand eines Alarms

Folgende Grundzustände kann ein Alarm annehmen:



Ein Alarm kann durch einzelne Funktionen wie MpAlarmXSet oder MpAlarmXReset kontrolliert werden oder durch den Funktionsbaustein MpAlarmXAlarmControl.

Je nachdem in welcher Reihenfolge der Alarm aktiviert, zurückgesetzt und quittiert wird, können verschiedene Kombinationen miteinander auftreten.

Alarm aktivieren -> Alarm zurücksetzen -> Alarm quittieren

- **Alarm aktivieren:** Wird ein Alarm aktiviert, So hat der Alarm die Zustände "Active" und "Not Acknowledged".
- **Alarm zurücksetzen:** Wird der Alarm danach zurückgesetzt, so hat der Alarm die Zustände "Inactive" und "Not Acknowledged".
- **Alarm quittieren:** Wurde anschließend der Alarm quittiert, so hat der Alarm den Zustand "None". Die Quittierung kann in der Applikation über MpAlarmXAcknowledge bzw. MpAlarmXAlarmControl erfolgen oder in der Visualisierung über MpAlarmXListUI.

Alarm aktivieren -> Alarm quittieren -> Alarm zurücksetzen

- **Alarm aktivieren:** Wird der Alarm aktiviert, So hat der Alarm die Zustände "Active" und "Not Acknowledged"
- **Alarm quittieren:** Wurde anschließend der Alarm quittiert, so hat der Alarm die Zustände "Active" und "Acknowledged". Die Quittierung kann in der Applikation über MpAlarmXAcknowledge bzw. MpAlarmXAlarmControl erfolgen oder in der Visualisierung über MpAlarmXListUI.
- **Alarm zurücksetzen:** Wird der Alarm danach zurückgesetzt, so hat der Alarm den Zustand "None".

Den Status eines Alarms kann man über die Funktion MpAlarmXCheckState herausfinden. Dabei wird einer der obigen fünf Status angegeben und man bekommt boolsches Feedback, ob der angegebene Status aktiv ist. Weiters kann über die Alarm Konfiguration festgelegt werden, ob ein Alarm quittiert werden, muss oder nicht und auch ob ein Alarm zurückgesetzt werden muss oder nicht. Wird ein Alarm über die Konfiguration deaktiviert ("Disable"), so hat der Alarm den Zustand "None".

Fehler, benutzerdefinierte Alarme und mapp Alarme

Grundsätzlich muss man im Umgang mit mapp Komponenten zwischen Fehlern und Alarmen unterscheiden. Fehler werden über die "StatusID" am Ausgang einer Funktion / eines Funktionsbausteins angezeigt, bzw. im Logger eingetragen. Die "StatusID" dient zum diagnostizieren, was beim Entwicklen der Applikation nicht funktioniert hat, bzw. später um eine detailierte Diagnose vornzunehmen.

Alarme geben dem Maschinenbediener Auskunft über den Zustand der Maschine. Alarme sollten auf eine Art und Weise verfasst sein, dass der Maschinenbediener damit etwas anfangen kann. Das Alarmhandling ist nicht dazu gedacht, eine detailierte Diagnose durchzuführen (z.B. "StatusID" auswerten, ACOPOS Fehlertexte anzuzeigen, ...).

Alarme und Fehler sind grundsätzlich getrennt voneinander zu betrachten. Alarme müssen nicht die Konsequenz eines Fehler sein sondern können auch eigenständig auftreten (z.B. *Benutzer wurde gesperrt wegen falscher Passworteingabe*, ein Alarm wird ausgelöst). Fehler geben Auskunft über die falschen Anwendung einer Hardware bzw. Software und können parallel zu einem Alarm auftreten, muss aber nicht so sein (z.B. *Laden eines Rezepts*, wird über die "StatusID" angezeigt, außerdem wird eine Alarm erzeugt). Folgende Tabelle zeigt einige Unterschiede auf:

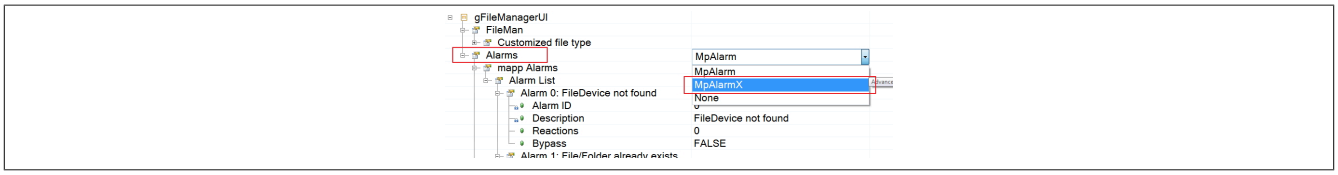
Fehler	
Alarme	mapp Alarm
	Benutzer Alarm

mapp Alarme

Für jede mapp Komponenten sind bereits Alarme angedacht. Dabei handelt es sich um Standardalarme, welchen dem Maschinenbediener helfen sollen den Status der Maschine zu ermitteln. Einige Beispiele:

- Rezept Laden fehlgeschlagen
- Encoder Fehler
- Benutzer wurde wegen mehrfach falscher Passworteingabe gesperrt
- Speichermedium wurde nicht gefunden

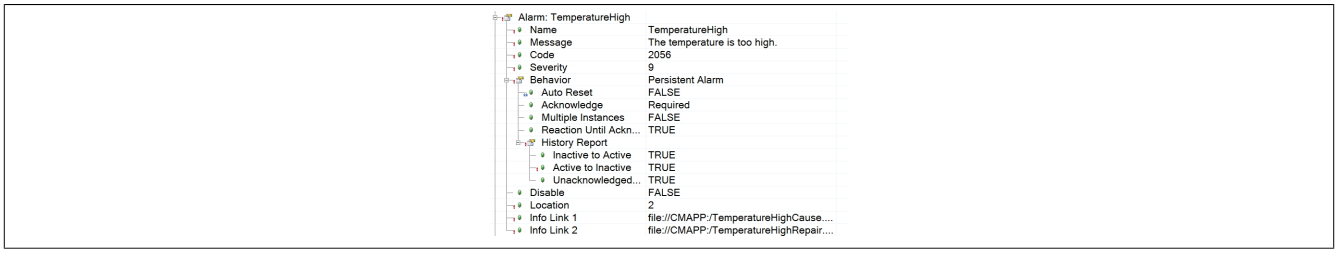
mapp Alarme können in der Konfiguration der jeweiligen Komponenten gefunden werden. In der Konfiguration muss der Benutzer festlegen, ob er die Alarme in Kombination mit MpAlarmX oder MpAlarm verwenden möchte:



Hier können die mapp Alarme auch weiter verändert werden, z.B. Textquelle anpassen, Reaction definieren oder ganz deaktivieren (disablen).

Benutzerdefinierte Alarme

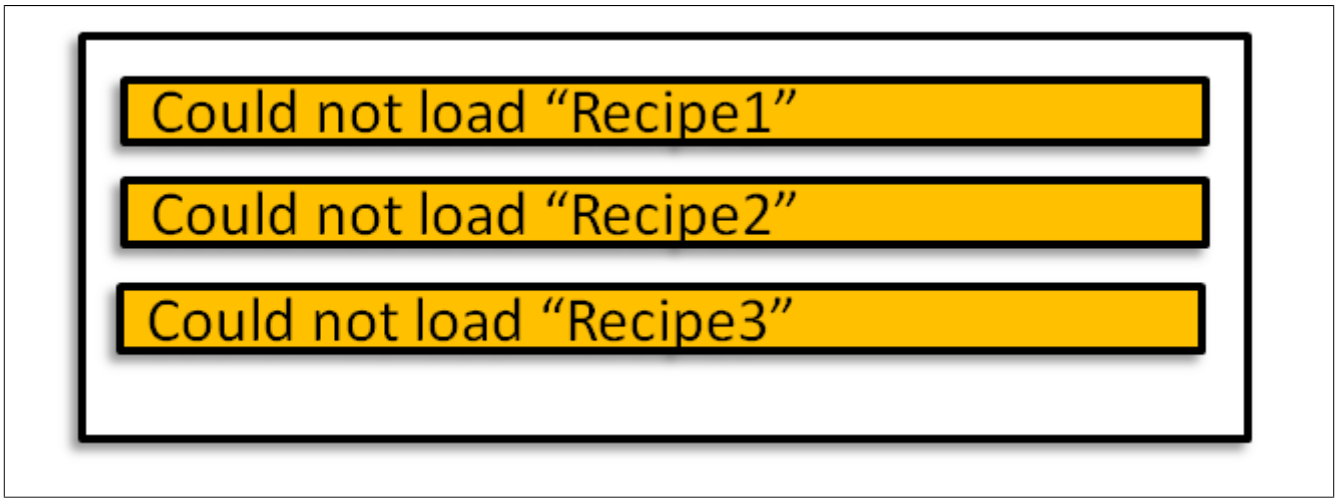
Eigene Alarme kann der Entwickler über die MpAlarmXCore Konfiguration festlegen. Dort kann dann für jeden Alarm das Verhalten angepasst werden, sowie unterschiedliche Einstellungen vorgenommen werden, z.B. Reaktion bestimmen, Links zu weiterführenden Daten oder den Level (Schweregrad) des Alarms angeben:



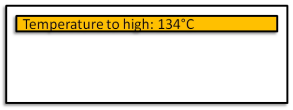
Weitere Details zu den möglichen Einstellungen, welche in der Alarm Konfiguration vorgenommen werden können, sind im Abschnitt Alarmliste zu finden.

1.1.2 Alarm Instanzen

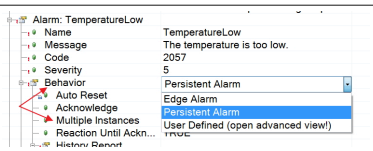
Zum Verhalten des Alarms gehört auch ob ein Alarm nur einfach aktiv sein kann, oder mehrere Instanzen desselben Alarms aktiv sein können. Gibt es beispielsweise einen Alarm, welcher anzeigt, dass ein Rezept nicht geöffnet werden konnte, so kann dieser Alarm potenziell mehrfach auftreten. Versucht man ein beschädigtes Rezept mit Namen "Recipe1" zu öffnen, dann soll eine Alarmeintrag erstellt werden, welcher aussagt "Folgendes Rezept konnte nicht geöffnet werden: Recipe1". Wird nun versucht ein weiteres Rezept, Recipe2, zu öffnen, welches wiederum nicht existiert, dann soll ein neuer Alarm in der Liste aufscheinen "Folgendes Rezept konnte nicht geöffnet werden: Recipe2". Dieser soll aber nicht den vorhergehenden ersetzen, da es sich um einen anderen Alarm handelt. In diesem Fall wird bei MpAlarmX von mehrfach instanzierten Alarmen gesprochen.



Ein Alarm, welcher eine zu hohe Temperatur anzeigt, kann dagegen nicht mehr als einmal aktiviert werden. Ist die Temperatur zu hoch, so bleibt dieser eine Alarm aktiv, bis die Temperatur wieder im Normalbereich liegt. Erst dann kann der Alarm erneut aktiviert werden.



Ob ein Alarm einfach oder mehrfach instanziert ist, wird grundsätzlich über das Verhalten festgelegt, kann jedoch selbst noch angepasst werden.



Einfach instanzierte Alarme

Einfach instanzierte Alarme sind gedacht für Alarme welche nach dem Setzen kein weiteres mal zur selben Zeit aktiv werden können. Ein solcher Alarm scheint in der aktuellen Alarmliste nur einmal auf. In der Alarmhistorie kann der einfach instanzierte Alarm, je nachdem wie oft er aktiviert wurde, mehrmals aufscheinen. Folgende Beispiele zeigen typisch einfach instanzierte Alarme:

- Wasserlevel in Tank zu hoch
- Temperatur im Extruder (154°C) liegt über dem maximalen Grenzwert (150°C)
- Not Aus gedrückt

Jeder "Persistent Alarm" ist als einfach instanzierter Alarm vorkonfiguriert.

Mehrfach instanzierte Alarme

Mehrfach instanzierte Alarme können öfter als einmal gesetzt werden. Diese Art Alarme können mehrfach in der Alarmliste auftauchen. Folgende Beispiele zeigen typisch mehrfach instanzierte Alarme:

- Folgendes Rezept konnte nicht geladen werden: "Recipe1"
- Senden des SMS an "Max Mustermann" ist fehlgeschlagen
- Modul nicht ok: "X20DI2653"

Jeder "Edge Alarm" ist als mehrfach instanzierter Alarm vorkonfiguriert. Mehrfach instanzierte Alarme können nicht mehr nur über deren Namen erkannt werden, da möglicherweise mehrere Alarme desselben Typs aktiv sind. Dafür gibt es die Intanz ID. Diese bekommt man als Rückgabewert von MpAlarmXSet zurück. Über diese Instanz ID wird eine instanzierter Alarm eindeutig erkannt.

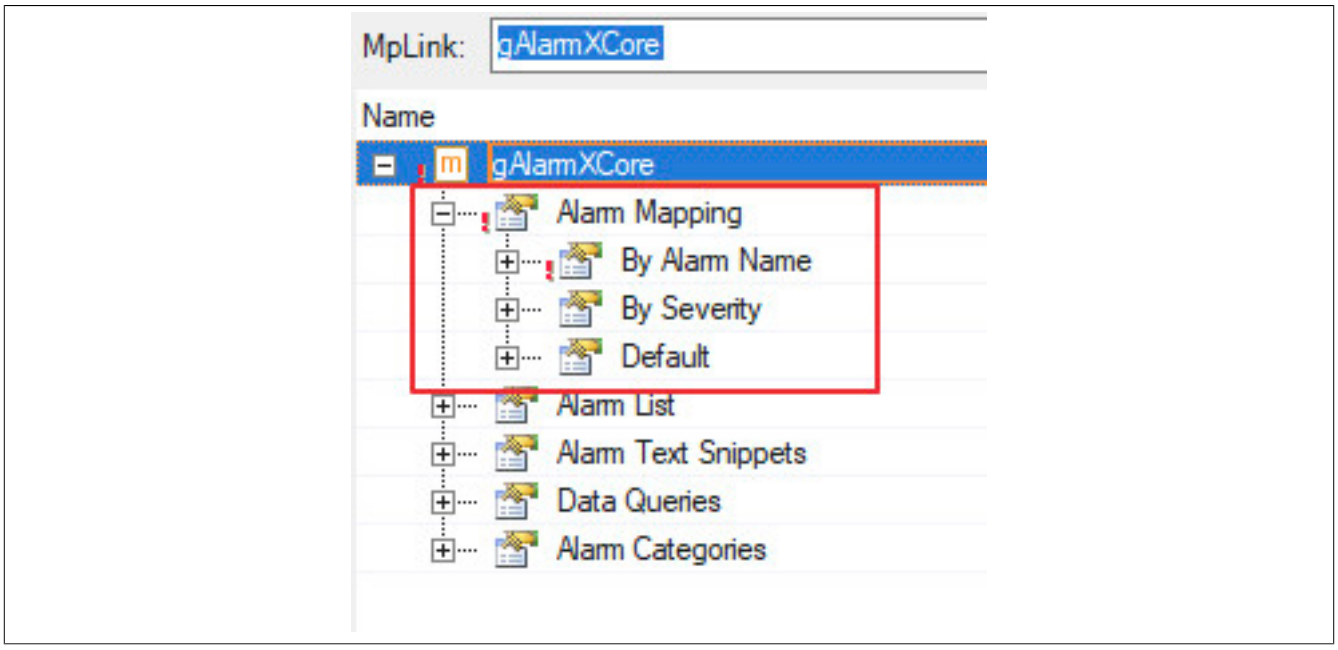
Verwaltung instanzierter Alarme

Um über die Applikation Zugriff auf eine bestimmte Instanz eines Alarms zu gewährleisten gibt es ID-Funktionen. Unter anderem sind MpAlarmXResetID, MpAlarmXAcknowledgeID und MpAlarmXCheckStateID vorhanden. Diese Funktionen funktionieren gleich wie dieselben Funktionen ohne das ID am Ende. Der einzige Unterschied ist, dass für diese Funktionen der Alarm über dessen Instanz ID angegeben wird, nicht über den Namen.

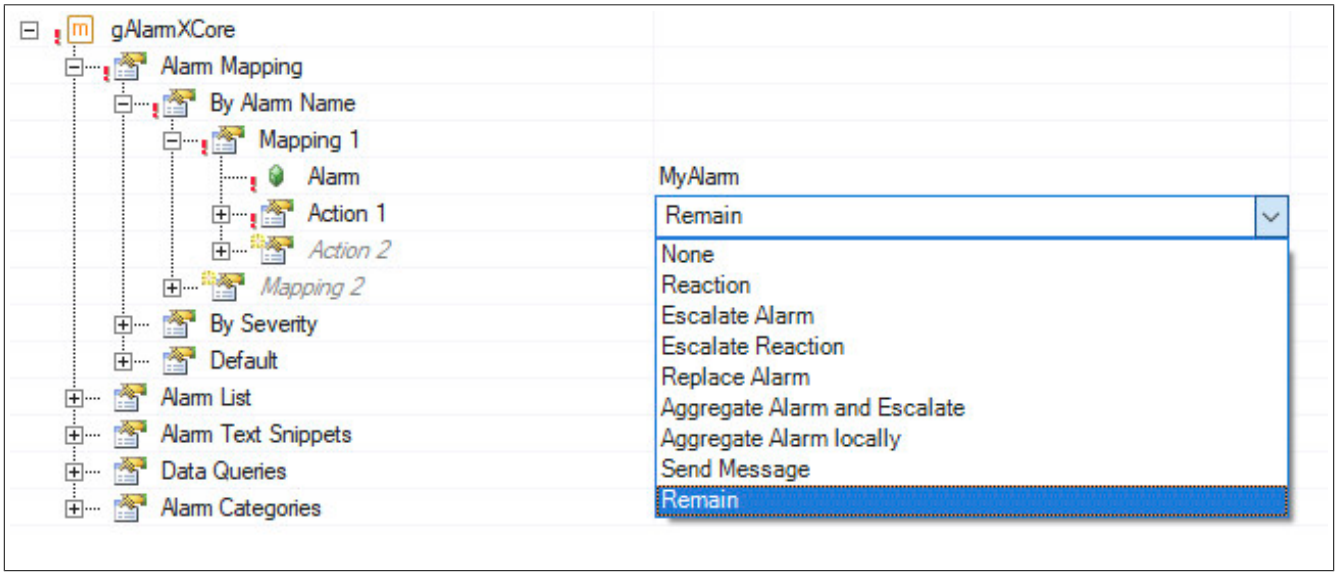
1.1.3 Alarm Mapping

Neben den Alarmeigenschaften kann der Benutzer auch definieren, wie sich der Alarm im System verhalten soll. Das bedeutet, es kann definiert werden welcher Alarm bzw. welcher Schweregrad eines Alarm welche Aktion auslösen soll. Dies wird im Abschnitt "Alarm Mapping" definiert.

Auf unserem B&R Tutorial Portal wird mithilfe der Tutorials mapp AlarmX Part 10 (GER/ENG) und mapp AlarmX Part 11 (GER/ENG) erklärt, wie das Alarm Mapping funktioniert. (Zugangsdaten)



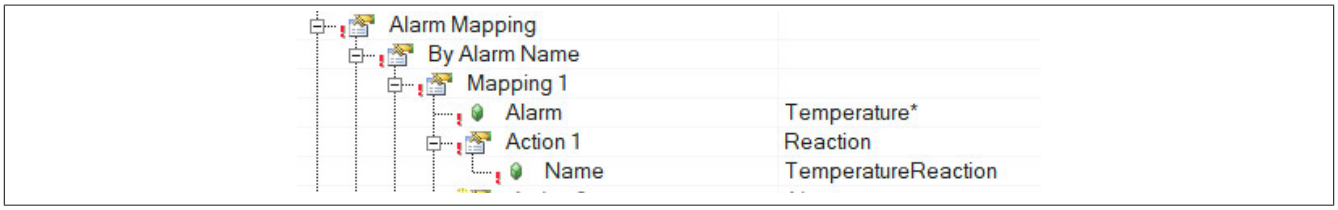
Dabei kann ein Alarm ("By Alarm Name") bzw. Schweregrad ("By Severity") verschiedene Aktionen auslösen:



By Alarm Name

Es werden ein oder mehrere Aktionen ausgeführt, wenn ein definierter Alarm aktiv wird.

Das Alarm Mapping unterstützt das Wildcard Prinzip. Soll auf mehrere Alarme gleich reagiert werden, so wird das Symbol "*" verwendet. Sollen beispielsweise alle Alarme, welche etwas mit der Temperatur zu tun haben ("TemperatureHigh", "TemperatureLow") die Reaktion "TemperatureReaction" auslösen, so kann dies folgendermaßen angegeben werden:

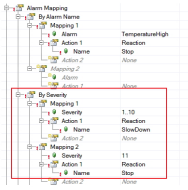


Auch auf die festgelegten Grenzwerte von Alarmen, welche über das Alarm Monitoring konfiguriert wurden, kann reagiert werden. Für mehr Information siehe Alarm Monitoring / Reaktion auf Grenzwerte.

Weiters kann auch auf Alarme anderer mapp Komponenten reagiert werden. Für mehr Information siehe mapp Alarme / Reaktion.

By Severity

Es werden eine oder mehrere Aktionen ausgeführt, wenn die Severity (eines oder mehrerer Alarme) einen bestimmten Wert annimmt, bzw. wenn die Severity in einem definierten Bereich liegt. Im Beispiel wird einmal eine Aktion als Antwort auf eine Severity in einem definierten Bereich festgelegt und beim zweiten mal eine Aktion als Antwort auf eine bestimmte Severity.



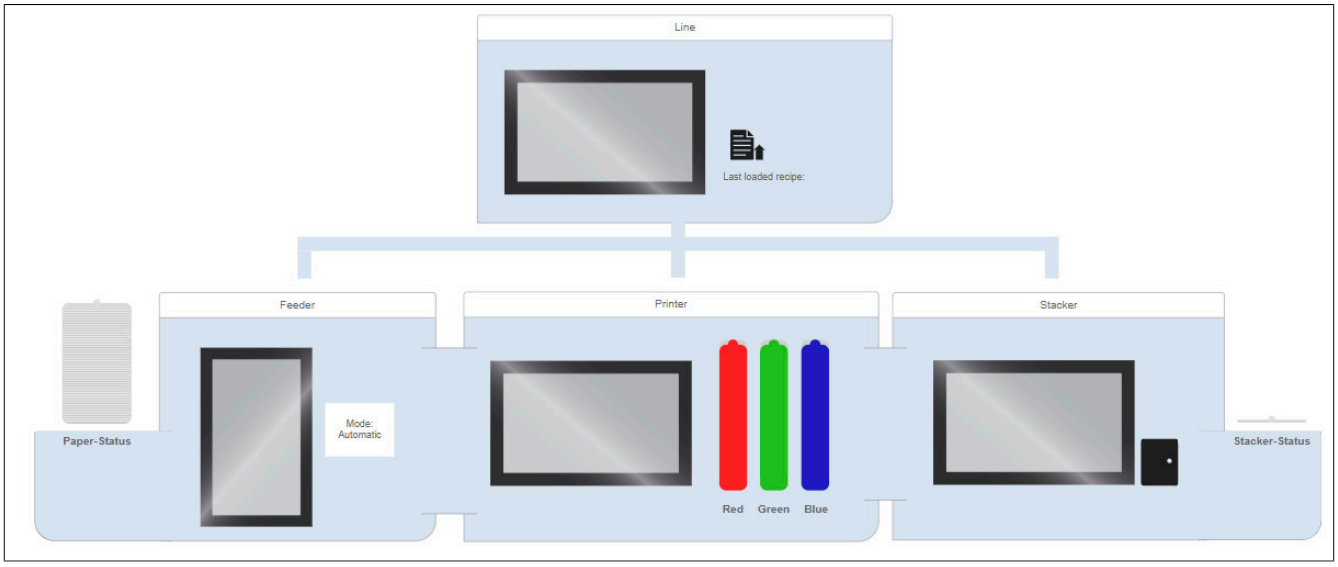
Die Severity wird folgendermaßen angegeben:

- **Für einen Wertebereich:** 1..10
- **Für einen bestimmten Wert:** 11

> **Es können mehrere Aktionen auf einen Alarm folgen. Wiedersprechen sich diese, so gewinnt im Mapping immer die erste Aktion, welche dem Alarm zugewiesen ist. Wird einem Alarm beispielsweise die Aktion Escalate Alarm und die Aktion Remain zugewiesen, dann wird die erste im Mapping vorhandene Aktion durchgeführt.**

Zur Erklärung der Aktionen wird folgende Maschinenlinie als Beipiel verwendet:

Wie diese Maschinenlinie aufgebaut ist wird im Anwendungsfall "Alarmverwaltung für Druckermaschinenlinie" erklärt.



In dem Beispiel handelt es sich um eine Druckermaschinenlinie, welche aus mehreren Stationen besteht. Die erste Station ist ein "Feeder", welche für die Papierzufuhr verantwortlich ist, danach folgt ein Printer, welcher für das Drucken zuständig ist und zum Schluss ein Stacker, welcher das bedruckte Papier lagert.

Die gesamte Linie wird über eine Steuerung betrieben. Pro Station wird zur Überwachung eine eigene Visualisierung verwendet. Weiters gibt es eine Visualisierung, welche die gesamte Linie überwacht.

In Folgendem werden die einzelnen Aktionen und Möglichkeiten anhand von verschiedenen Beispiel-Alarmen erklärt:

- None
- Reaction
- Escalate Alarm
- Escalate Reaction

- Replace Alarm
- Aggregate Alarm and Escalate
- Aggregate Alarm locally
- Send Message
- Remain
- Alarm Monitoring / Reaktion auf Grenzwerte
- mapp Alarme / Reaktion

Default Action

Die *Default Action* gibt an was standardmäßig mit allen Alarmen, welche nicht anders gemappt werden, passiert. Jede der beschriebenen Aktionen kann hier ausgewählt werden. Beispielsweise kann der Benutzer festlegen, dass, egal welcher Alarm aktiv wird, immer die Reaktion "YellowLamp" aktiv werden soll. Diese Reaktion wiederum lässt an der Maschine eine gelbe Warnleuchte blinken. Die Konfiguration würde folgendermaßen aussehen:

Die Standard *Default Action* ist *Escalate Alarm*. Für Systeme mit nur einer Visualisierung / einer Gruppe hat dies keine Bedeutung, da die Alarme nicht weiter eskaliert werden können. Für das obige Beispiel, unsere Maschinenlinie, würde das bedeuten, dass alle konfigurierten Alarme standardmäßig zur Linienüberwachung hin eskaliert werden.

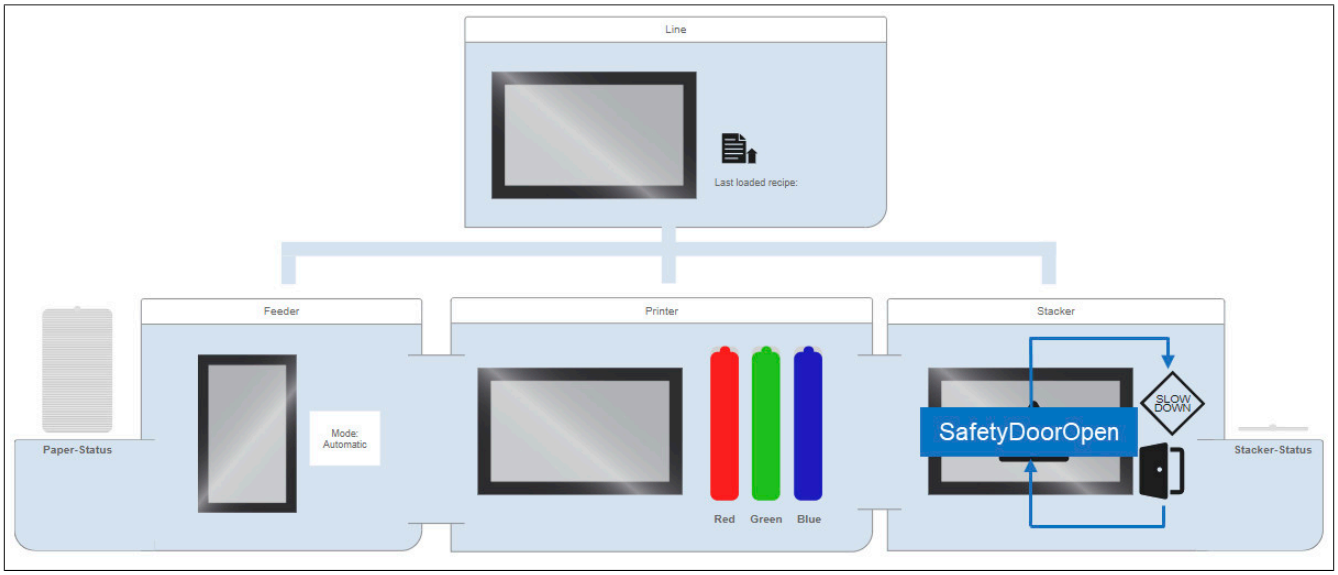
1.1.3.1 None

Bei der Angabe von "None" wird keine Aktion ausgeführt.

1.1.3.2 Reaction

Bei der Angabe von "Reaction" wird lokal eine Reaktion ausgeführt. Das bedeutet, sobald ein definierte Alarm im System auftritt wird eine Reaktion im System ausgelöst.

Beispiel:



Wird im "Stacker" die Sicherheitstür geöffnet, wird der Alarm "SafetyDoorOpen" ausgelöst. Daraufhin soll die Zufuhrgeschwindigkeit zum Stacker verringert werden. Dies wird durch die Reaktion "SlowDown" realisiert.

Dies kann in der Konfiguration folgendermaßen aussehen:

Damit der Alarm lokal behandelt werden kann, wird die zweite Aktion "Remain" definiert.

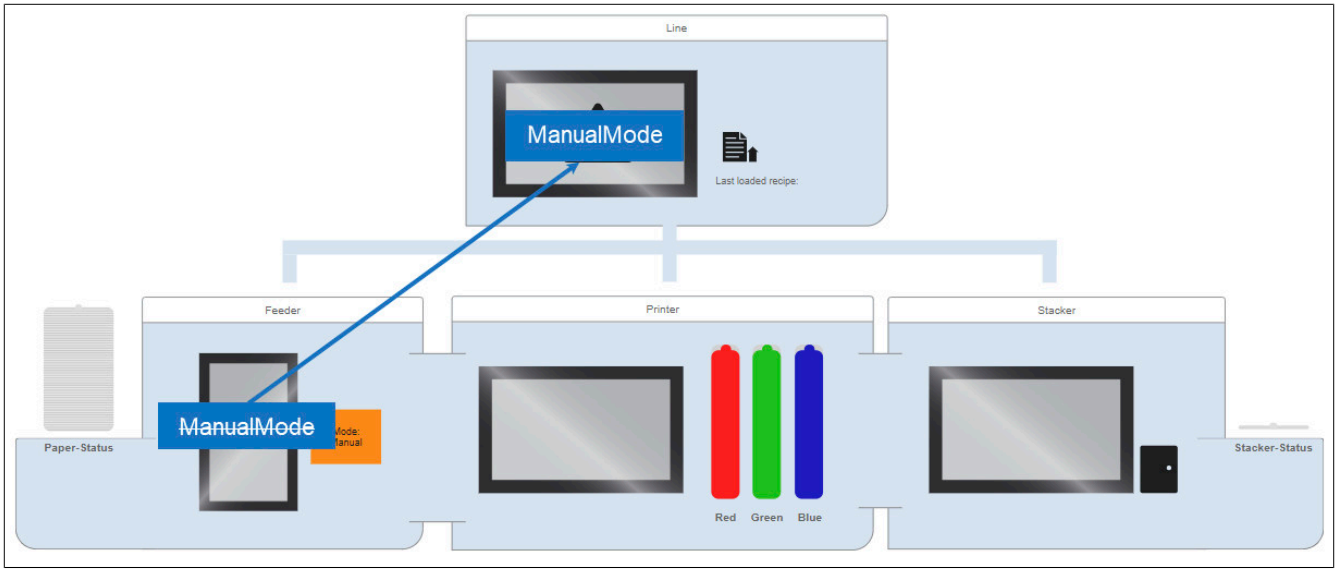
Um in der Applikation auf die Reaktion zu reagieren, wird die Funktion MpAlarmXCheckReaction verwendet:

```
IF MpAlarmXCheckReaction(gStackerAlarm, 'SlowDown') THEN
    SlowDown := TRUE;
ELSE
    SlowDown := FALSE;
END_IF
```

1.1.3.3 Escalate Alarm

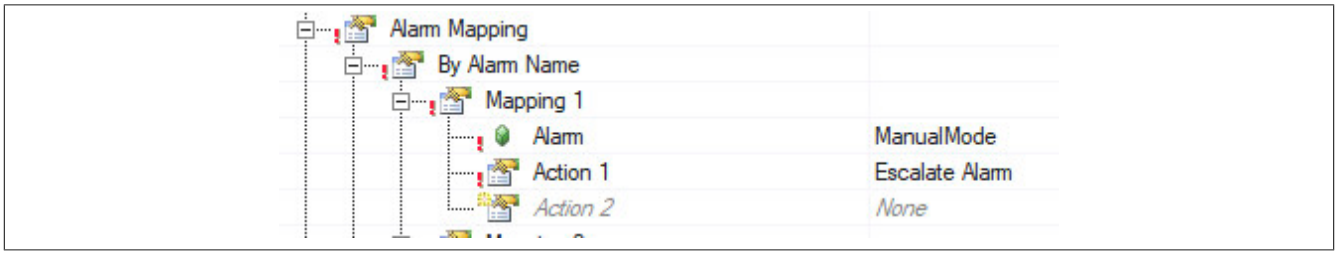
Mithilfe von "Escalate Alarm" wird ein Alarm an die übergeordnete Gruppe hin eskaliert.

Beispiel:



Wechselt der Maschinenbediener den Modus der Maschine von "Automatic" auf "Manual", so wird der Alarm "ManualMode" ausgelöst. Der Alarm wird in der Linienüberwachung behandelt und lokal im "Feeder" nicht mehr angezeigt. Die Quittierung des Alarms erfolgt über die Linienüberwachung.

Dies kann in der Konfiguration folgendermaßen aussehen:



Folgendes muss beachtet werden:

Quittieren:

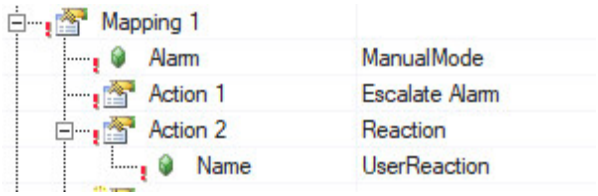
Soll ein Alarm quittiert werden, welcher durch die Aktion "Escalate Alarm" zu einer übergeordneten Gruppe hin eskaliert wurde, so muss zum quittieren des Alarms der mapp Link des übergeordneten Alarmsystems angegeben werden.

Zurücksetzen des Alarms:

Soll ein Alarm zurückgesetzt werden, welcher durch "Escalate Alarm" zur übergeordneten Gruppe eskaliert wurde, so muss zum deaktivieren des Alarms der mapp Link der Gruppe angegeben werden, von der der Alarm ursprünglich kommt.

Zusätzliche Verwendung von "Reaction":

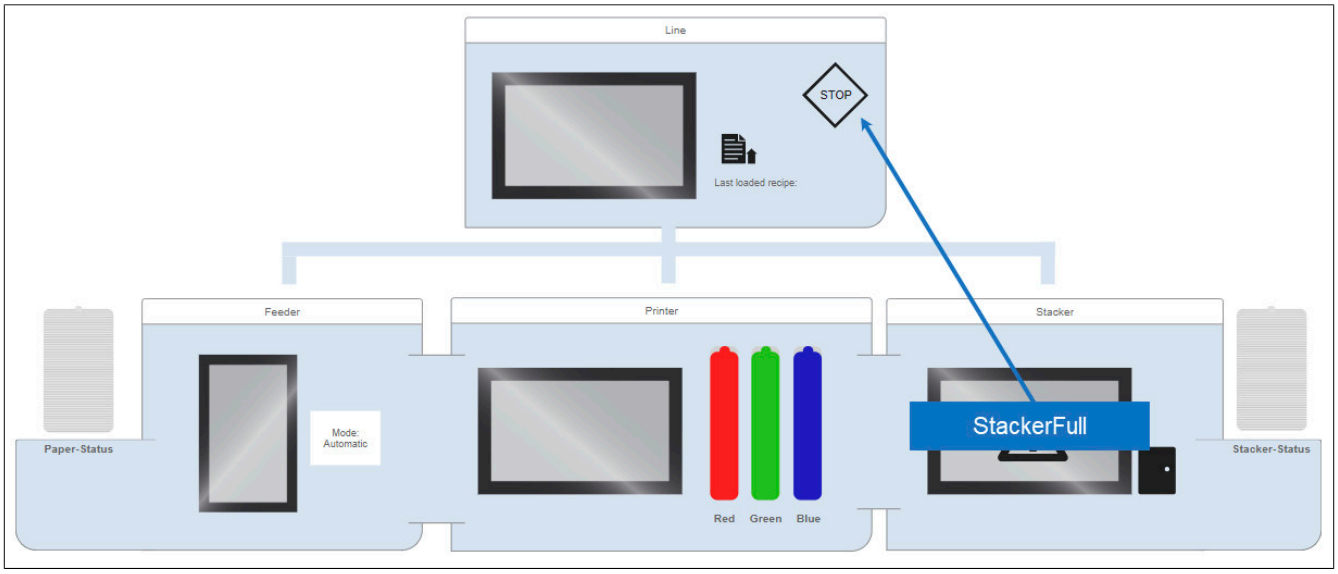
Wird zusätzlich zur Reaktion "Escalate Alarm" noch eine weitere "Reaction" Aktion hinzugefügt, so gilt es zu beachten, dass der Alarm durch "Escalate Alarm" in der übergeordneten Gruppe (Linienüberwachung) sowie durch "Reaction" lokal (Feeder) angezeigt wird.



1.1.3.4 Escalate Reaction

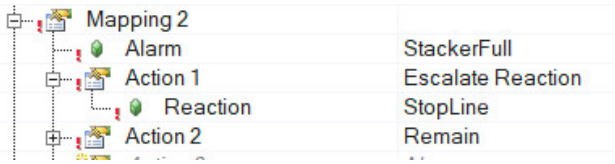
Durch "Escalate Reaction" wird, sobald ein definierter Alarm im System auftritt, die Reaktion darauf an die übergeordnete Gruppe weitergeleitet. Der Alarm wird weiterhin lokal angezeigt, nur die Reaktion wird nach oben hin weitergeleitet.

Beispiel:



Ist der "Stacker" voll, wird der Alarm "StackerFull" ausgelöst. Sobald dieser Alarm aktiv ist, wird die Reaktion "StopLine" an die Linenüberwachung weitergeleitet. Daraufhin wird die gesamte Maschinenlinie gestoppt.

Dies kann in der Konfiguration folgendermaßen aussehen:



Damit der Alarm lokal behandelt werden kann, wird die zweite Aktion "Remain" definiert.

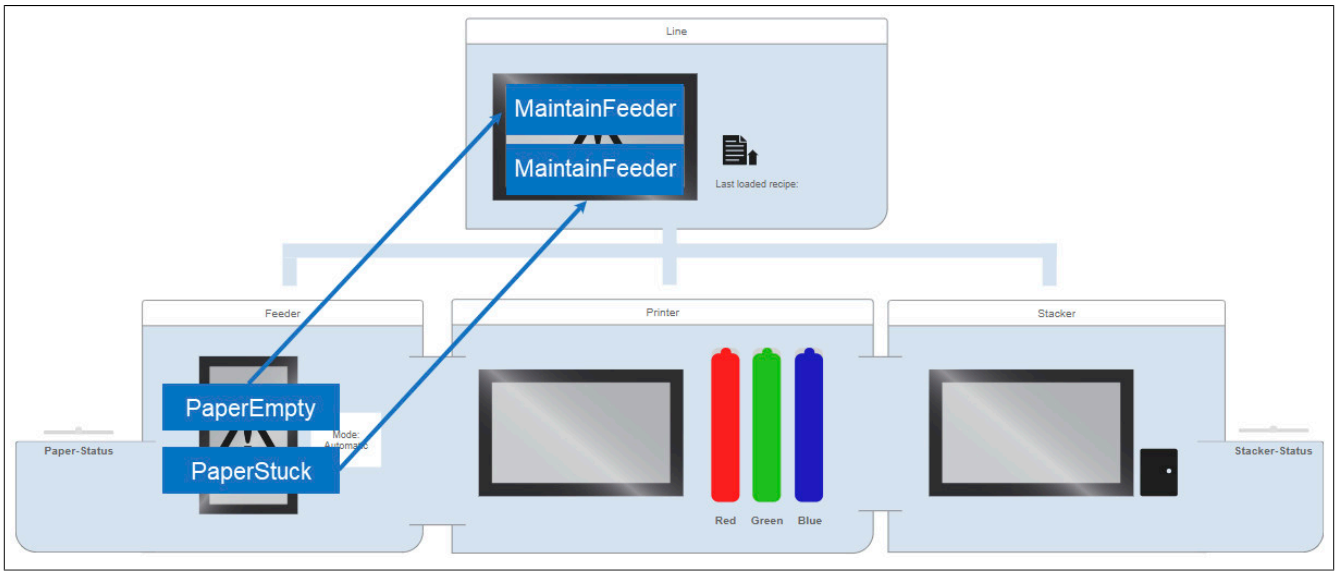
Um in der Applikation auf die Reaktion zu reagieren, wird die Funktion MpAlarmXCheckReaction verwendet:

```
IF MpAlarmXCheckReaction(gLineAlarm, 'StopLine') THEN
    StopLine := TRUE;
ELSE
    StopLine := FALSE;
END_IF
```

1.1.3.5 Replace Alarm

Mithilfe von "Replace Alarm" wird ein definierter Alarm lokal behandelt sowie durch einen anderen ersetzt welcher nach oben hin zur übergeordneten Gruppe eskaliert wird.

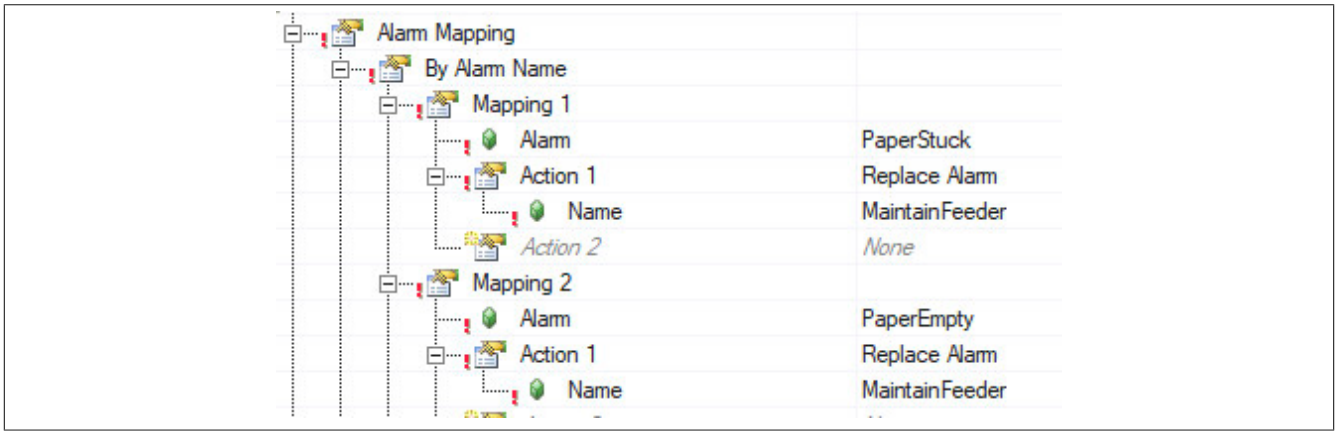
Beispiel:



Im Feeder wird der Füllstand des Papiers sowie die Zufuhr des Papiers zum Drucker hin überwacht.

Ist das Papier zu wenig bzw. staut sich das Papier zum Printer hin wird der Alarm "PaperEmpty" bzw. "PaperStuck" ausgelöst. Damit der Benutzer über das Problem im "Feeder" Bescheid weiß, soll ein Alarm zur Linienüberwachung geschickt werden ("MaintainFeeder"). Dieser Alarm soll den Benutzer über auszuführende Wartungsarbeiten im "Feeder" informieren. Lokal im "Feeder" ist dann die genau Ursache des Problems ersichtlich. Das bedeutet es kann mehrmals der Alarm "MaintainFeeder" in der Linienüberwachung auftreten, je nachdem ob der gemappte Alarm aktiv ist oder nicht.

In der Konfiguration kann dies folgendermaßen aussehen:



Folgendes muss beachtet werden:

Quittieren:

Wird der ersetzte Alarm, welcher nach oben hin eskaliert wurde, quittiert, so wird auch der Ursprünglich Alarm quittert. Wird der Ursprüngliche Alarm zuerst quittiert, so wird der eskalierte Alarm nicht quittiert.

Das heißt, wird der Alarm "MaintainFeeder" in der Linienüberwachung quittiert, so wird auch automatisch der Alarm "PaperEmpty" bzw. "PaperStuck" quittiert.

Wird zuerst "PaperEmpty" oder "PaperStuck" quittiert, bleibt der Alarm in der Linienüberwachung unquittiert.

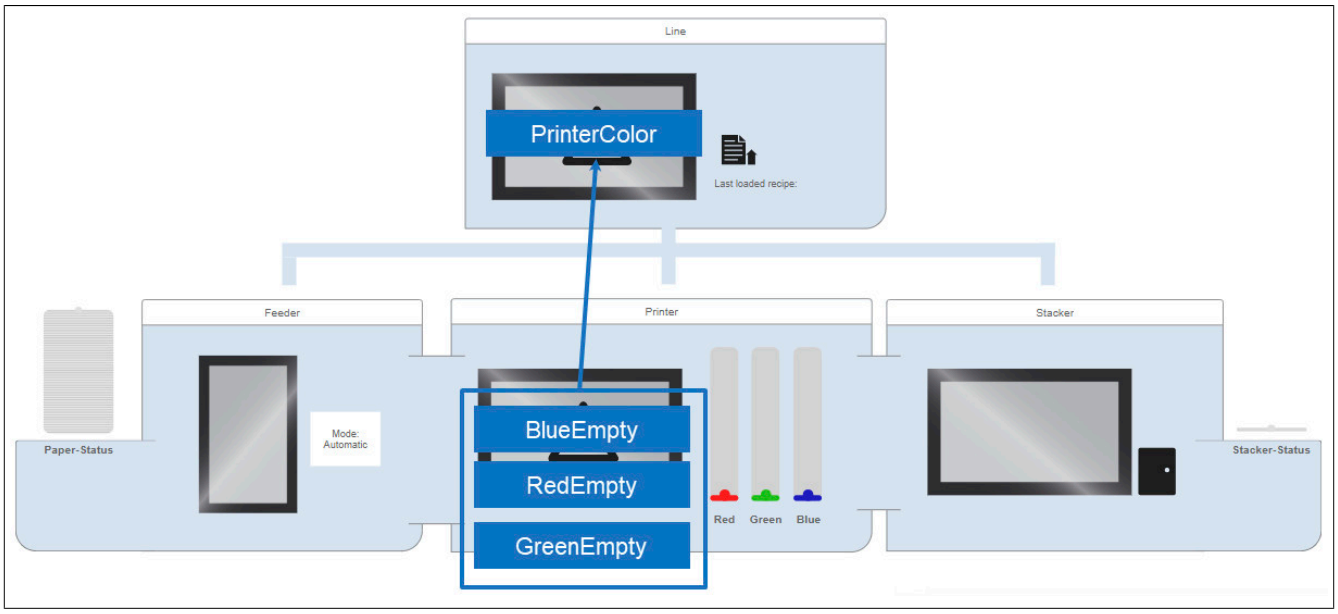
Zurücksetzten des Alarms:

Sobald der Ursprüngliche Alarm zurückgesetzt wird, wird automatisch der ersetzte und nach oben hin eskalierte Alarm zurückgesetzt.

1.1.3.6 Aggregate Alarm and Escalate

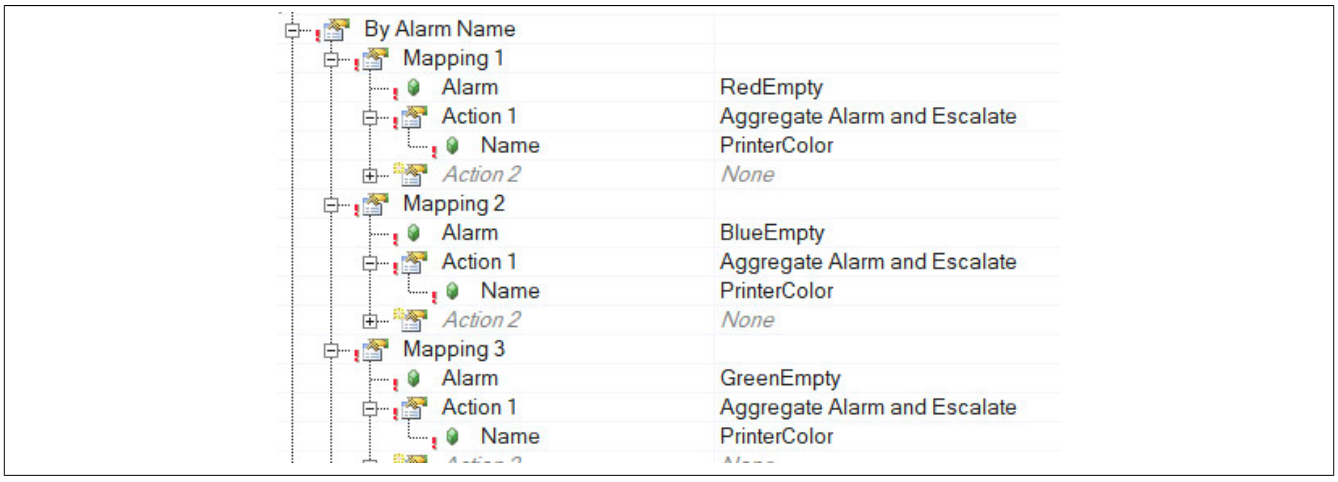
Mithilfe von "Aggregate Alarm and Escalate" können ein odere mehrere Alarme durch einen anderen Alarm ersetzt und direkt nach oben hin esklaliert werden.

Beispiel:



Sobald eine der Farben im "Printer" leer ist, wird der jeweilige Alarm "RedEmpty", "BlueEmpty" bzw. "GreenEmpty" ausgelöst und lokal im "Printer" angezeigt. Auf der Linienüberwachung wird nur der aggregierte Alarm "PrinterColor" angezeigt. Angenommen es sind mehrere Farben leer, so wird immer nur ein Alarm in der Linienüberwachung angezeigt. Dies verhindert ein "überfluten" der Alarmliste.

Dies kann in der Konfiguration folgendermaßen aussehen:

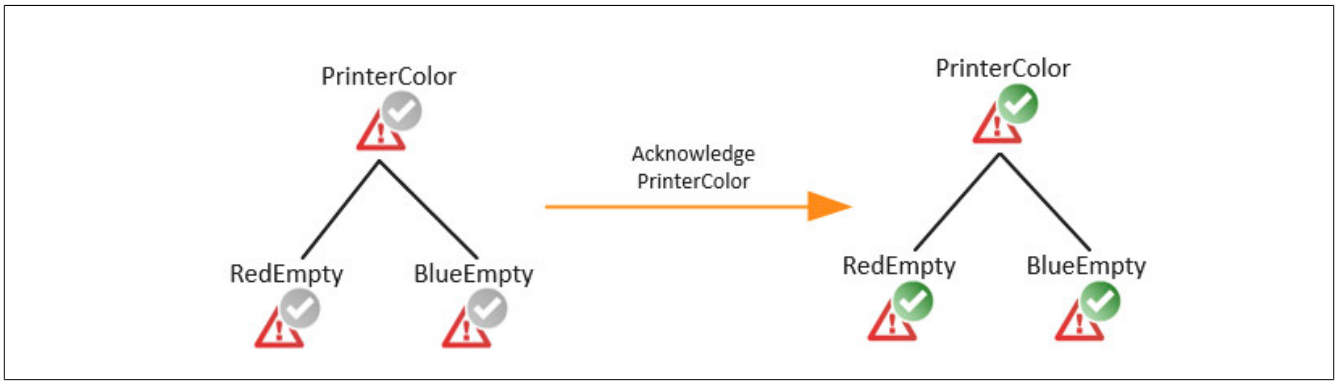


Der aggregierte Alarm wird nicht mehr in der Liste angezeigt, sobald alle gemappten Alarme deaktiviert und / oder quittiert wurden.

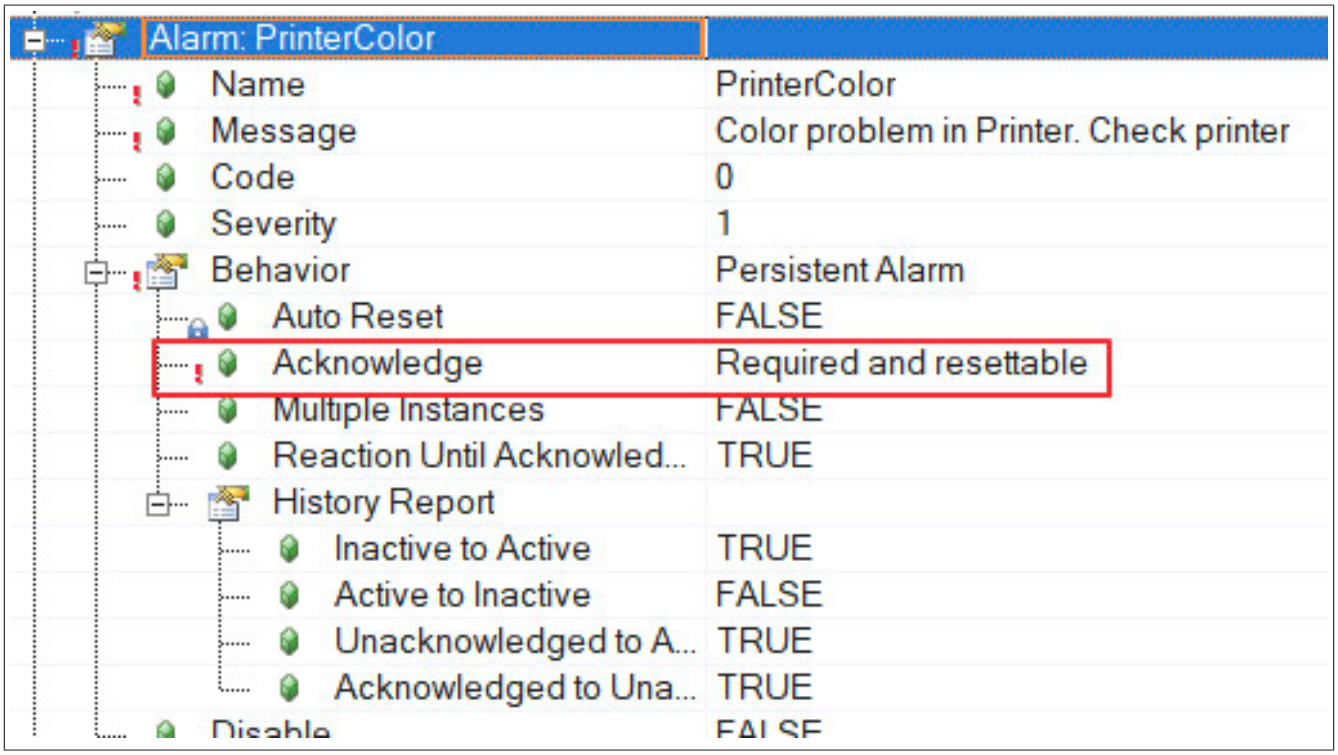
Folgendes gilt zu beachten:

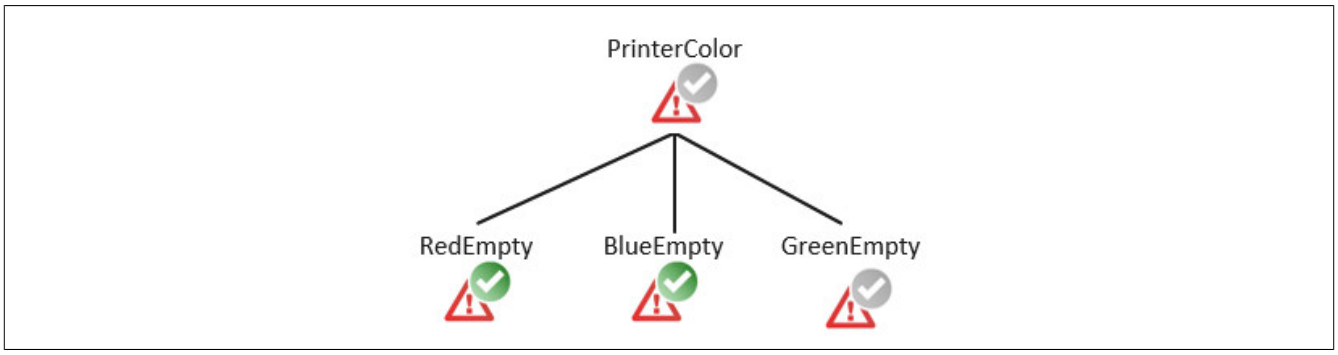
Quittieren:

Wird der Aggregierte Alarm ("PrinterColor") quittiert, so werden alle Alarme die auf diesen Alarm gemappt und aktiv sind automatisch auch quittiert.



Wurde "PrinterColor" quittiert und es tritt ein weiterer Farb-Alarm auf ("GreenEmpty"), so kann der Quittierstatus des aggregierten Alarm zurückgesetzt werden. Dies ist durch das Quittierverhalten "Acknowledge and resetable" möglich. Dadurch weiß der Benutzer, dass ein weiterer Farb-Alarm aktiviert, oder bereits ein bestehender nochmals aktiviert wurde.





Wird "PrinterColor" wieder quittiert, so wird "GreenEmpty" auch quittiert.

Alarm Instanzen:

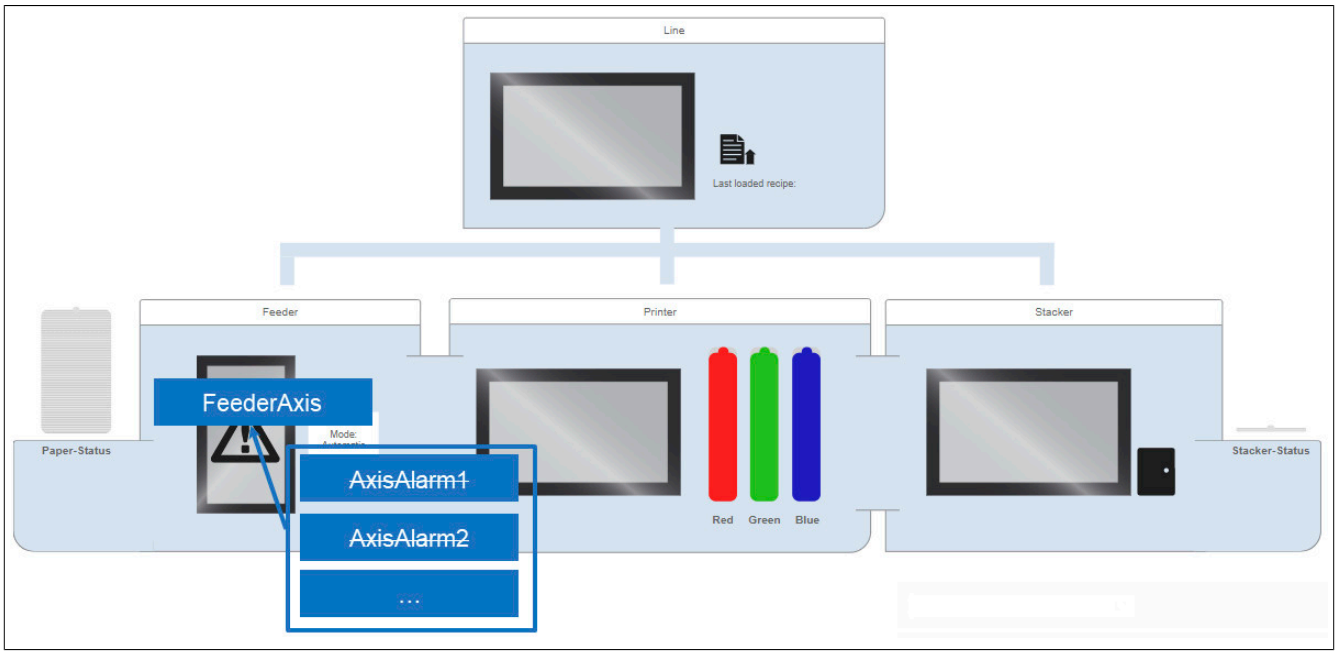
Ein aggregierter Alarm kann immer nur eine Instanz und nicht mehrere haben. Das beudeutet, die Option "Multiple Instances =TRUE" ist bei einem aggregierten Alarm nicht wirksam.

1.1.3.7 Aggregate Alarm locally

Zur Aktion "Aggregate Alarm locally" gibt es auch einen Anwendungsfall "Aggregieren von Alarmen"

Durch die Aktion "Aggregate Alarm locally" ist es möglich einen oder mehrere Alarme durch einen anderen Alarm zu ersetzen. Der Alarm wird lokal behandelt.

Beispiel:



Das Förderband welches für die Papierzufuhr im "Feeder" verantwortlich ist, wird mithilfe von mapp Axis angesteuert. Worauf bei der Verwendung von mapp Alarmen geachtet werden muss, ist hier zu finden. Sobald ein Alarm in der Achse auftritt wird dieser automatisch an das Alarmmanagement weitergeleitet. Aber anstelle mehrerer Achs-Alarme wird nur der Alarm "FeederAxis" angezeigt. Dadurch wird ein "überfluten" der Alarmliste verhindert.

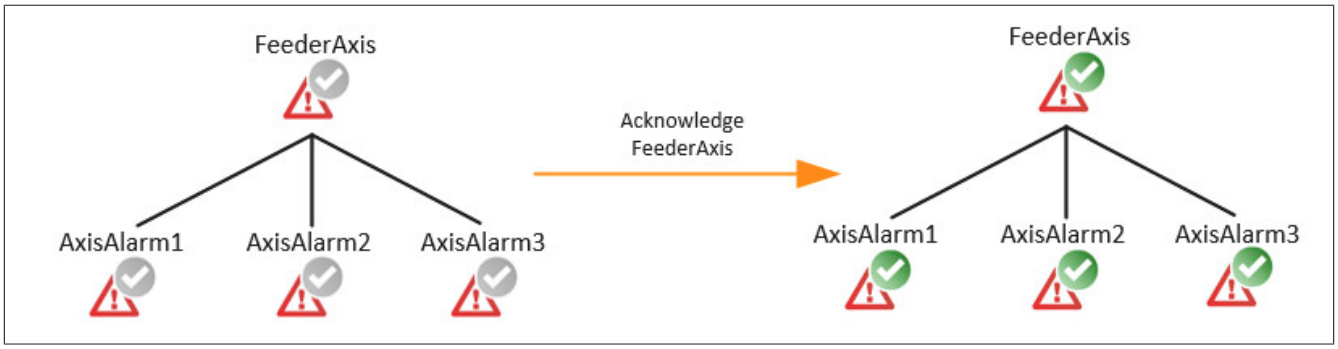
Dies kann in der Konfiguration folgendermaßen aussehen:

Der aggregierte Alarm wird nicht mehr in der Liste angezeigt, sobald alle gemappten Alarme deaktiviert und / oder quittiert wurden.

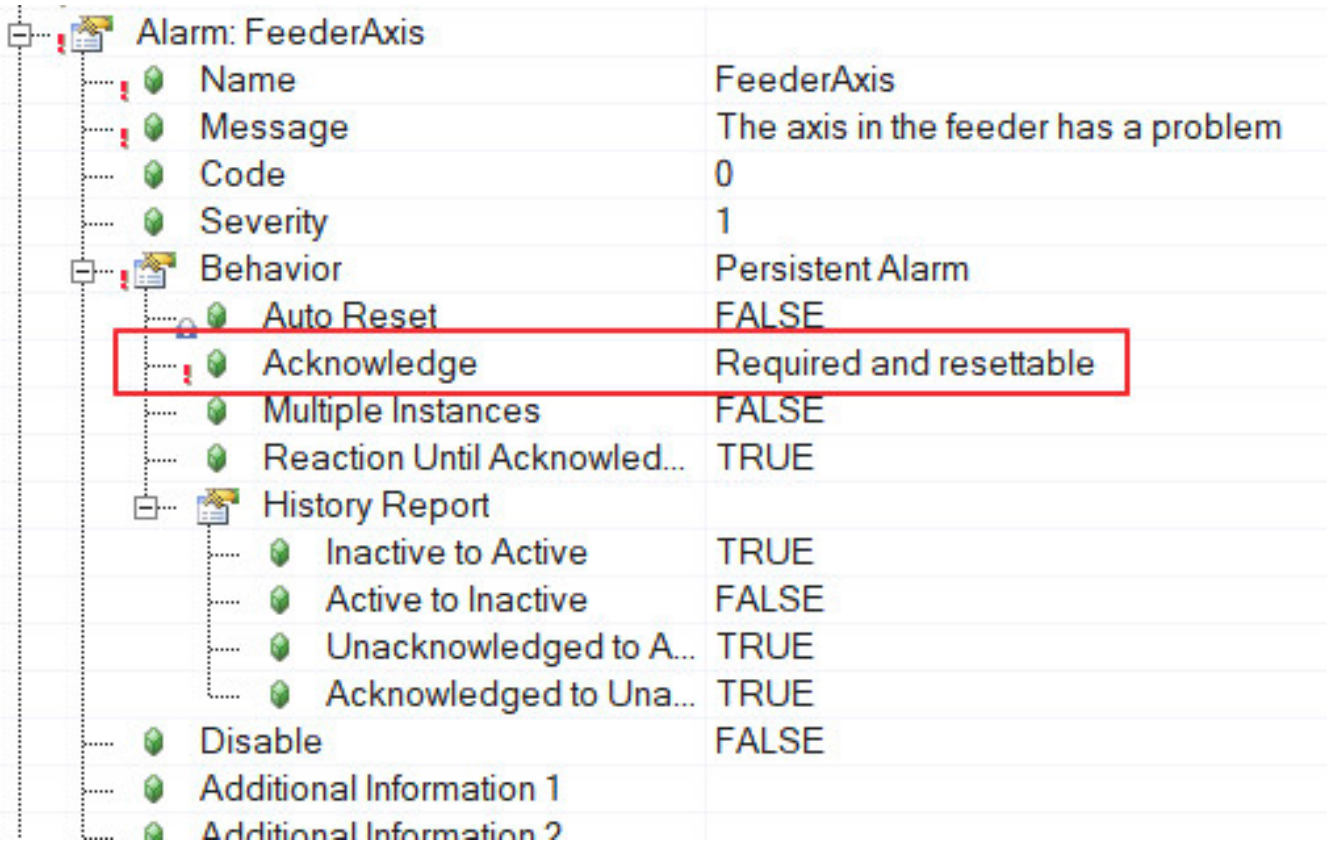
Folgendes gilt es zu beachten:

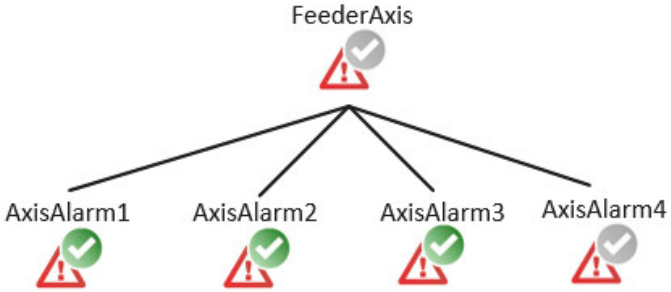
Quittieren:

Wird der Aggregierte Alarm ("FeederAxis") quittiert, so werden alle Alarme die auf diesen Alarm gemappt und aktiv sind automatisch auch quittiert.



Wurde "FeederAxis" quittiert und es tritt ein weitere Achs-Alarm auf, so kann der Quittierstatus des aggregierten Alarm zurückgesetzt werden. Dies ist durch das Quittierverhalten "Acknowledge and resetable" möglich. Dadurch weiß der Benutzer, dass ein weiterer Achs-Alarm aktiviert, oder bereits ein bestehender nochmals aktiviert wurde.





Wird "FeederAxis" wieder quittiert, so wird "AxisAlarm4" auch quittiert.

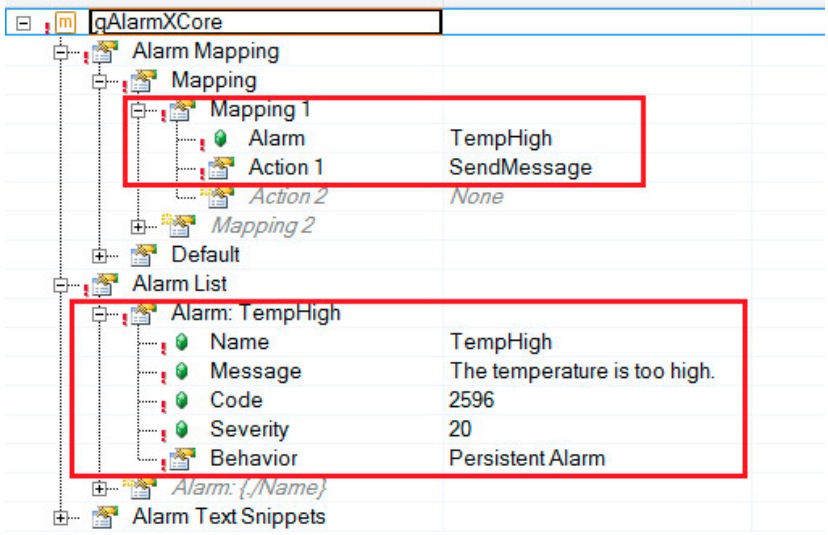
Alarm Instanzen:

Ein aggregierter Alarm kann immer nur eine Instanz und nicht mehrere haben. Das beudeutet, die Option "Multiple Instances =TRUE" ist bei einem aggregierten Alarm nicht wirksam.

1.1.3.8 Send Message

Durch die Aktion "SendMessage" ist es möglich im Alarmfall eine Benachrichtigung in Form einer SMS an einen angegebenen Benutzer zu senden. Um diese Option zu verwenden muss die Komponente mapp Tweet eingefügt werden.

Die Konfiguration kann dabei folgendermaßen aussehen:

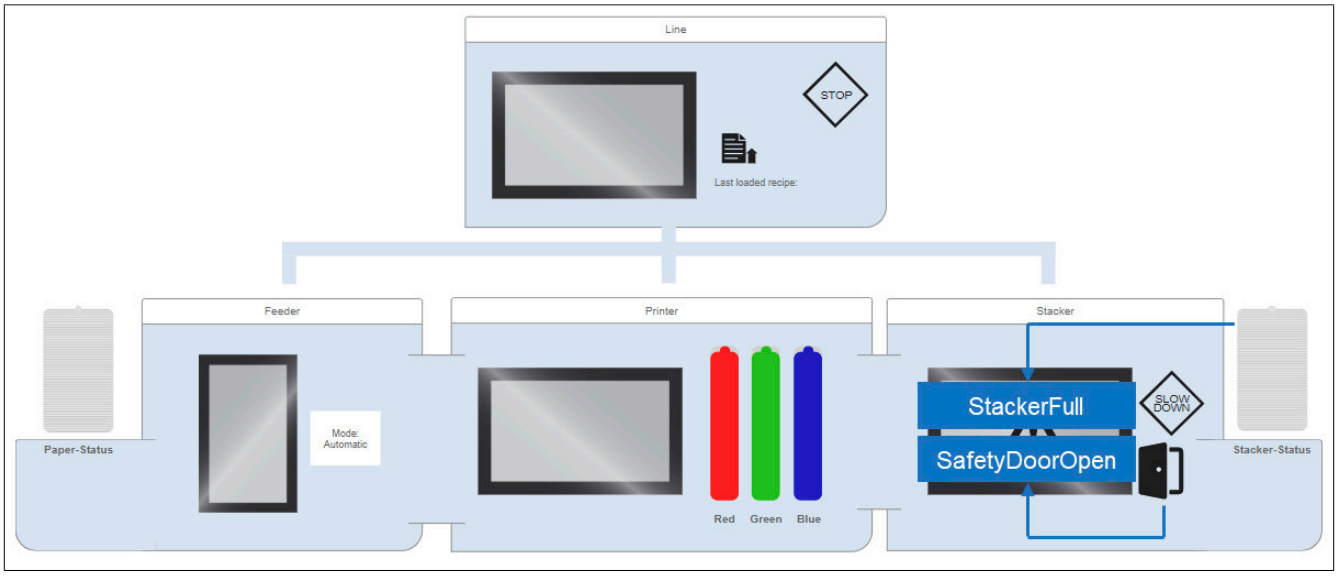


Sobald der Alarm "TempHigh" aktiv wird, wird eine Nachricht an einen Benutzer gesendet. Der Benutzer wird in der MpTweetCore Konfiguration festgelegt. Für mehr Information siehe Verbindung zur Alarmverwaltung.

1.1.3.9 Remain

Durch "Remain" wird der definierte Alarm nur auf dem Level angezeigt in dem er konfiguriert wurde.

Beispiel:



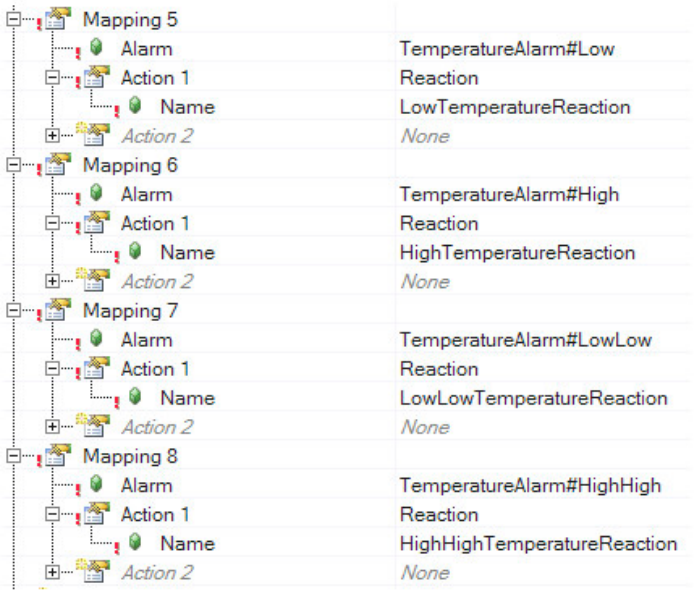
Sobald der "Stacker" voll ist oder die Sicherheitstür offen ist, wird der Alarm "StackerFull" bzw. "SafetyDoorOpen" ausgelöst. Beide Alarme lösen jeweils eine Reaktion aus die lokal behandelt oder nach oben hin eskaliert wird (Siehe Abschnitt "Reaction" und "Escalate Reaction"). Die zwei Alarme werden aber jeweils lokal behandelt.

Die Konfiguration könnte folgendermaßen aussehen:

1.1.3.10 Alarm Monitoring / Reaktion auf Grenzwerte

Auch auf die festgelegten Grenzwerte von Alarmen, welche über das Alarm Monitoring konfiguriert wurden, kann reagiert werden.

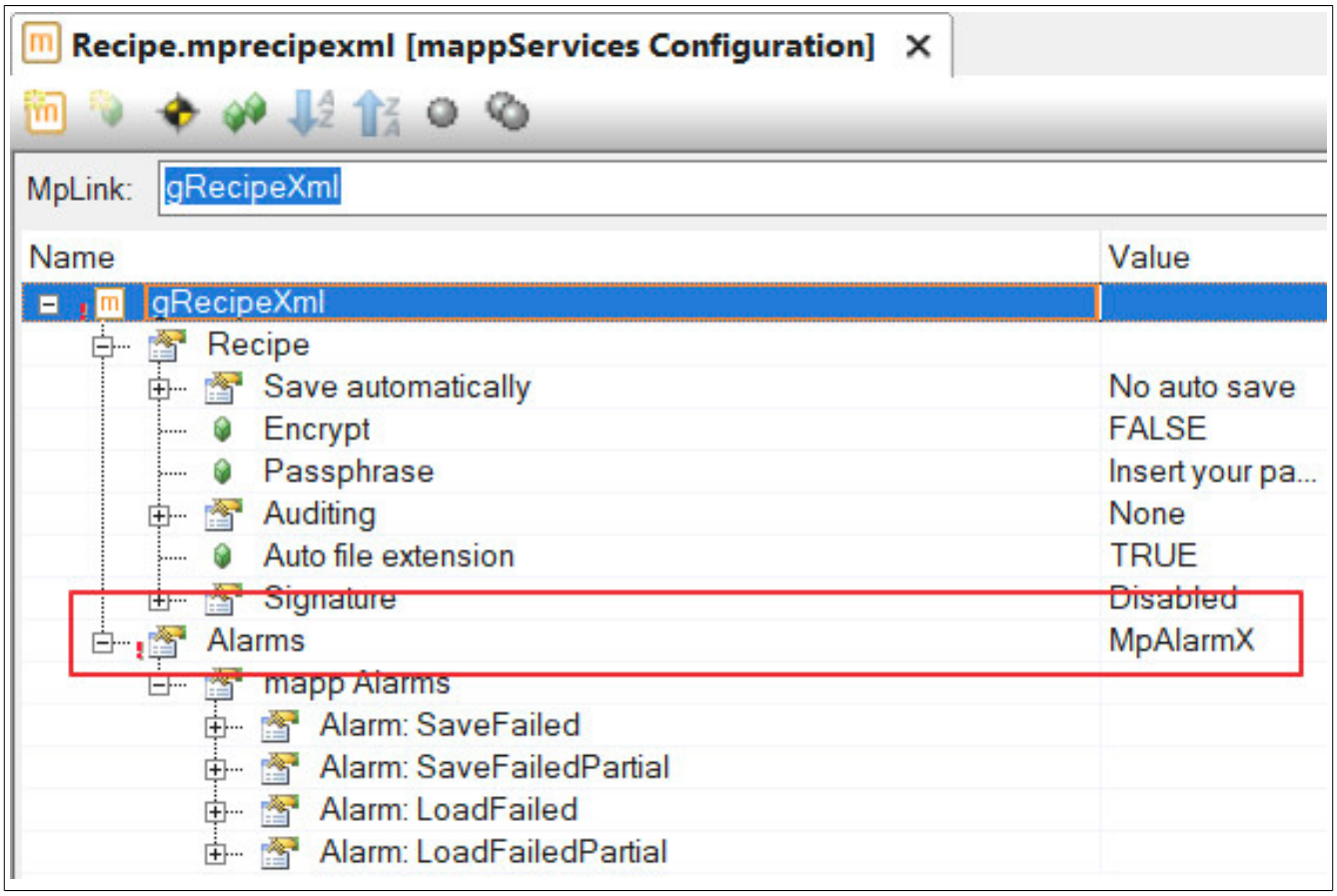
Dabei muss nur das gewünschte Grenzwert-Schlüsselwort den Alarmnamen hinzugefügt werden. Je nachdem auf welchem Grenzwert reagiert werden soll, können die Schlüsselwörter "#Low", "#LowLow", "#High" oder "#HighHigh" verwendet werden. Das kann folgendermaßen aussehen: "AlarmName#LowLow"



Im obigem Beispiel gibt es den Alarm "TemperatureAlarm" welcher über das Alarm Monitoring (Level Monitoring) festgelegt wurde. Für den Alarm wurden 2 obere ("High", "HighHigh") und zwei untere Grenzwerte ("Low", "LowLow") festgelegt.

1.1.3.11 mapp Alarme / Reaktion

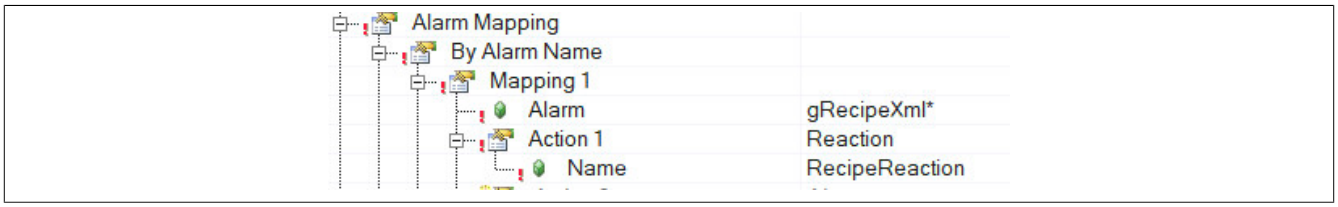
Es kann auch auf die Alarme anderer mapp Komponenten reagiert werden. Dabei muss der mapp Link der Komponente unter "Alarm" definiert werden. Wichtig hierbei ist, dass das Alarmmanagement "MpAlarmX" in der jeweilig verwendeten mapp Komponente verwendet werden muss. Dies wird in der Konfiugration der jeweiligen Komponente definiert:



Es kann jede beliebige Aktion ausgewählt werden. Um auf mapp Alarme zu reagieren gibt es zwei verschiedene Möglichkeiten:

1. Möglichkeit

Es wird automatisch auf alle Alarme der Komponente reagiert. Dabei muss der mapp Link der gewünschten Komponente sowie das Wildcard-Symbol "*" verwendet werden.



2. Möglichkeit

Es werden auf bestimmte Alarme der mapp Komponente reagiert. Dabei wird der mapp Link sowie der Name des Alarms angegeben. Dies kann folgendermaßen aussehen: "mappLink.Alarmname"

1.1.4 Alarm Texte

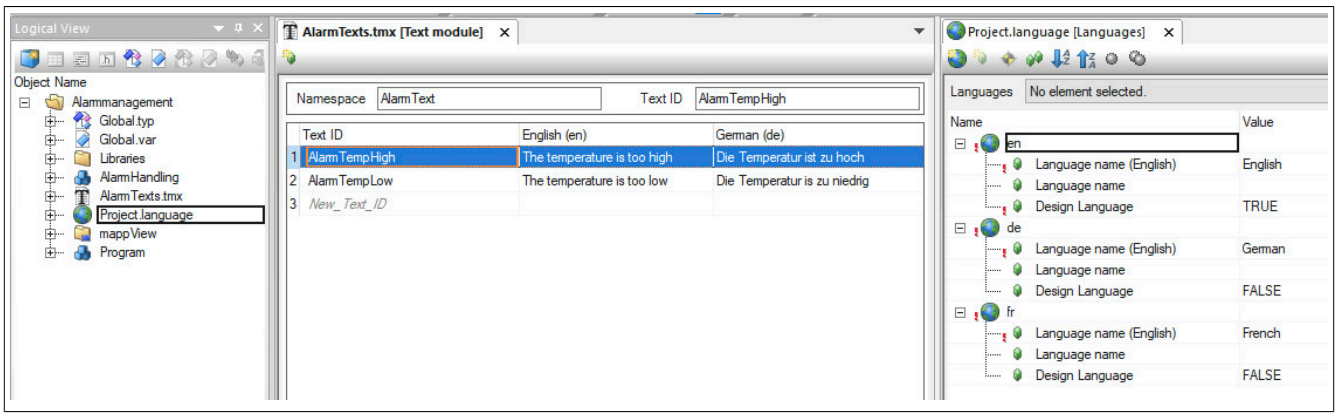
In diesem Abschnitt wird erklärt, welche Schritte notwendig sind um Texte in mapp AlarmX lokalisiert über das Textsystem anzugeben.

Auf unserem B&R Tutorial Portal wird über das Tutorial "Lokalisierbare Texte in mapp" (GER/ENG) erklärt, wie das Textsystem in Kombination mit mapp Services verwendet wird. (Zugangsdaten)

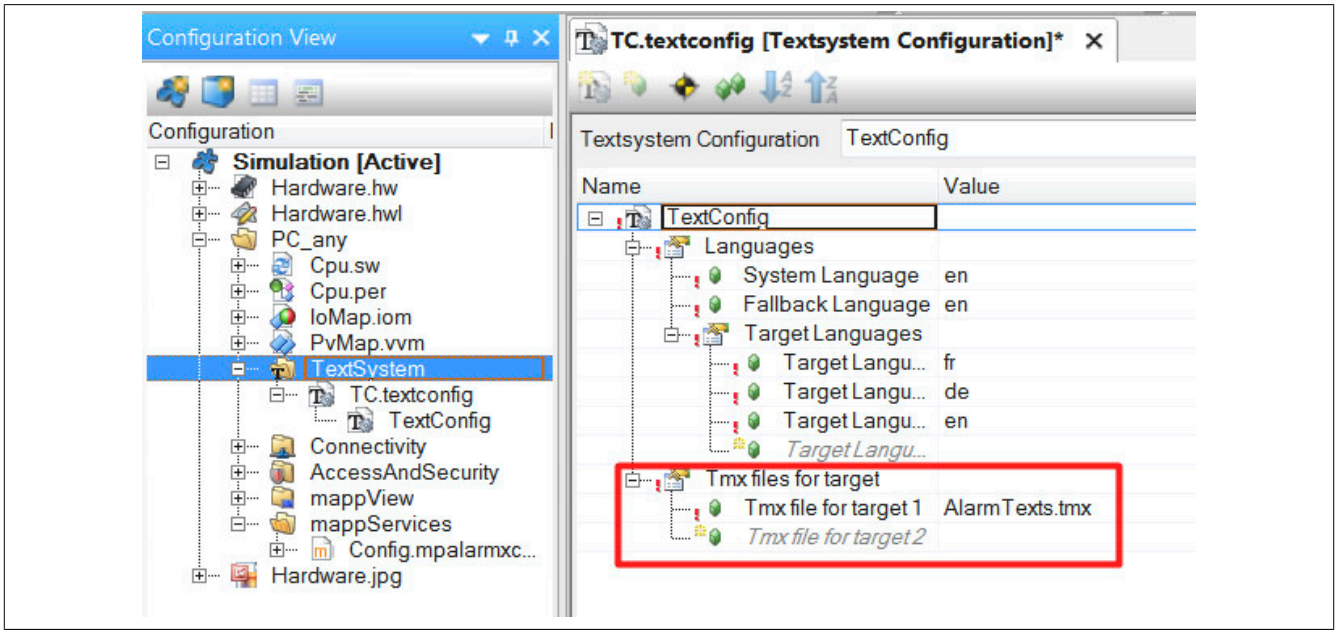
Dateien für Textsystem einfügen

Um Texte im Alarmsystem zu lokalisieren wird das Automation Studio Textsystem verwendet. Um zu definieren, welche Sprachen verfügbar sein sollen, müssen die gewünschten Projektsprachen ("Project.language") in Automation Studio konfiguriert werden. Die Alarm Texte selbst werden in einem Textmodul definiert. Für das Textmodul muss ein Namespace definiert werden. Über diesen Namespace kann auf das Textmodul zugegriffen werden. Zusätzlich wird für jeden Text im Textmodul eine ID definiert.

Damit mapp AlarmX auf einen der definierten Texte zugreifen kann, wird auf den Namespace des Textmoduls sowie auf die gewünschte Text ID referenziert.



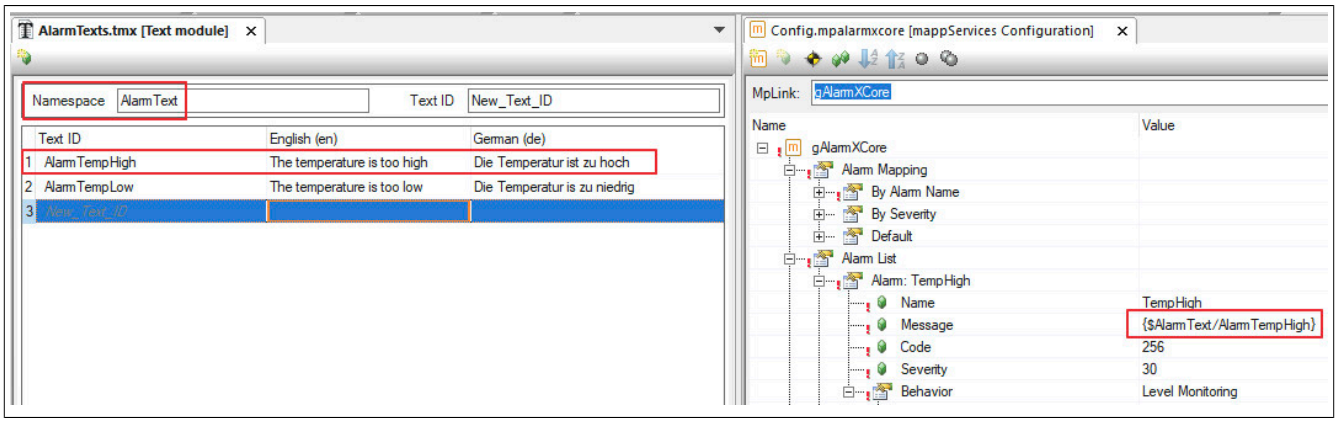
Damit die Alarm Texte zur Laufzeit auch verwendet werden, muss das Textmodul in der Textkonfiguration angegeben werden:



Alarmnachricht lokalisieren

Um eine Alarmnachricht ("Message") zu lokalisieren muss auf die gewünschte Text ID eines Textmoduls verlinkt werden. Dabei wird folgendes Format verwendet:

- {$Namespace/TextID}



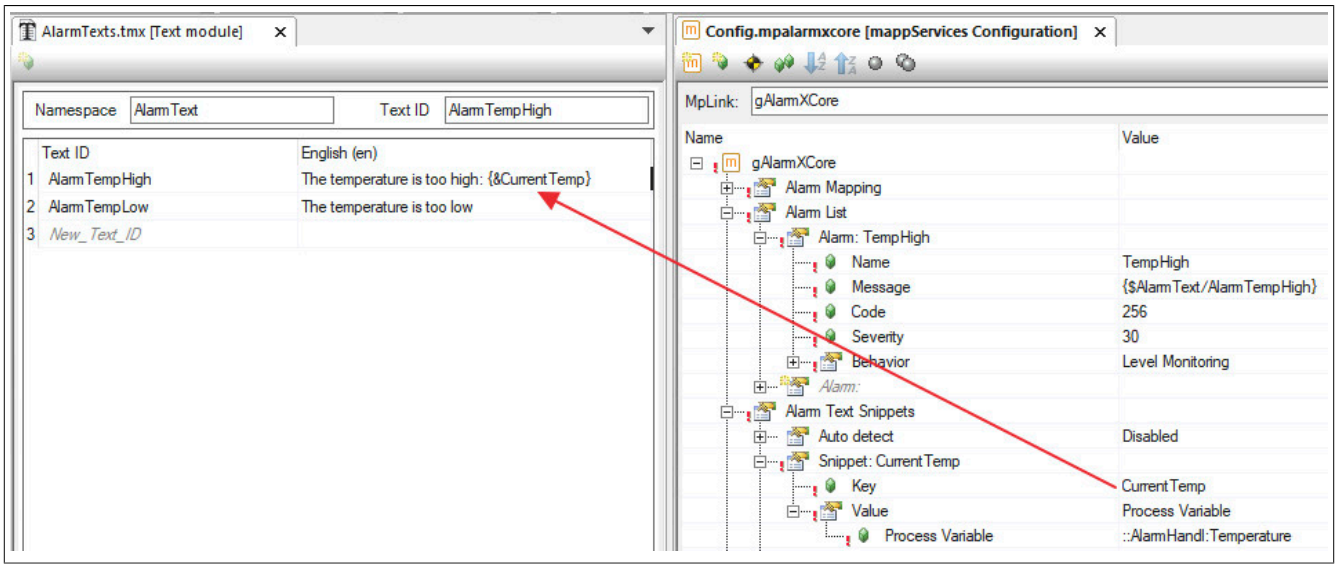
Dadurch kann die Alarmnachricht sprachabhängig angezeigt werden

Alarm Textsnippets lokalisieren

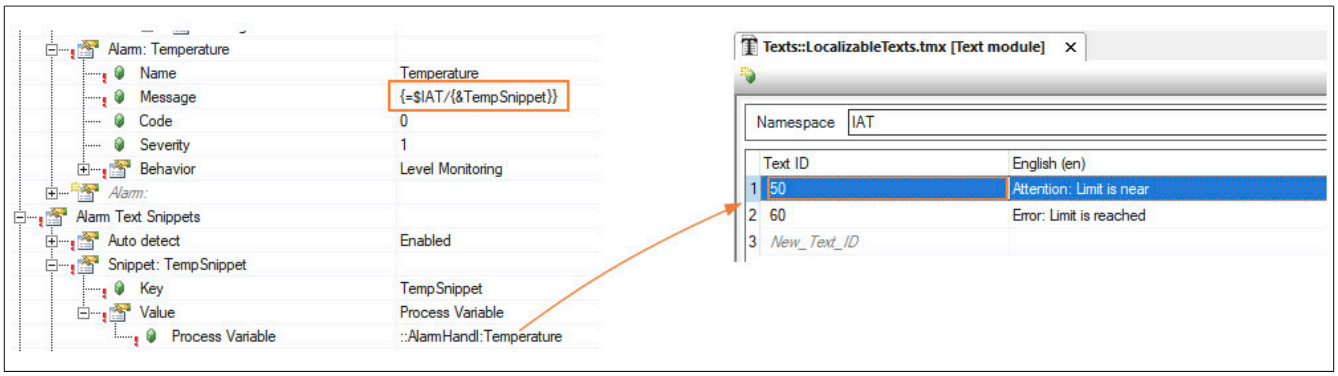
Im Konfigurationsabschnitt Alarm Textsnippets kann für einen Alarm zusätzliche Applikationsinformation angegeben werden.

Es gilt zu beachten, dass dort erstellte Alarm Textsnippets nur für Alarme verwendet werden können, welche im Abschnitt Alarmlist definiert wurden. Alarm Textsnippets können nicht in den Alarmen anderen mapp Komponenten verwendet werden.

Wurde die Alarmnachricht lokalisiert über das Textsystem definiert (siehe vorheriger Abschnitt), so kann das Alarm Textsnippet auch im Textmodul verwendet werden:



Je nachdem welchen Wert das Snippet besitzt kann auch ein anderer Alarmtext angzeigt werden. Dazu wird das Snippet in Kombination mit dem Textsystem verwendet:



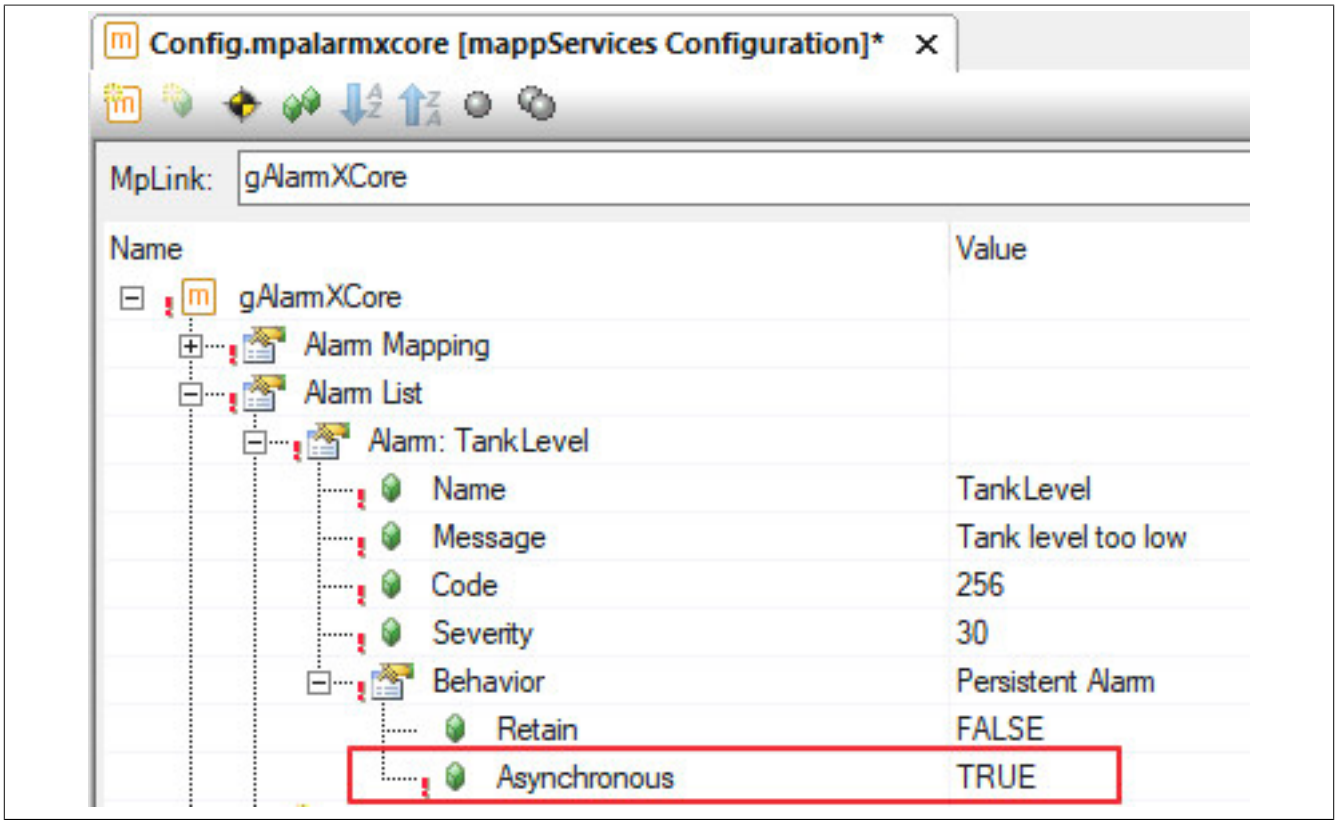
Wichtig hierbei ist, dass in diesem Fall in der Alarmnachricht das Format "{=$Namespace/{&TextSnippetID}}" verwendet wird.

Besitzt in diesem Beispiel das Snippet den Wert 50, so wird als Alarmtext jener Text angezeigt, welcher als TextID 50 verwendet.

1.1.5 Synchrone und asynchrone Alarme

In diesem Abschnitt wird erklärt, wie sich synchrone und asynchrone Alarme im System verhalten.

Standardmäßig verhält sich ein Alarm synchron. Durch die Konfigurations-Eigenschaft "Asynchronous = `TRUE`" wird ein Alarm asynchron.



Synchrone Alarme

Synchrone Alarme werden im gleichen Kontext aktiviert, zurückgesetzt oder quittiert wo sie aufgerufen werden. Es ist nicht relevant in welcher Taskklasse sich MpAlarmXCore befindet. Ist MpAlarmXCore in einer Taskklasse mit 100ms und ein Alarm wir in einer 10ms Taskklasse aktiviert, so wird der Alarm direkt nach dem Setzen in der 10ms Taskklasse im System aktiv. Auch eine mögliche Reaktion ist sofort aktiv und kann in der Code-Zeile danach bereits mithilfe von MpAlarmXCheckReaction überprüft werden.

Weiters werden Snippet Informationen sofort abgegriffen, wenn ein synchroner Alarm aktiviert wird. Das Synchrone Alarmverhalten sollte für Alarme gewählt werden, welche zeitkritisch sind. Wird beispielsweise die Reaktion sofort benötigt oder muss der Wert des Alarmsnippets direkt abgefragt werden, so empfiehlt sich ein synchrones Verhalten. Alle anderen Alarme können asynchron definiert werden.

Asynchrone Alarme

Die Konfigurations-Eigenschaft "Asynchronous" ist nur für benutzerdefinierte, edge sowie persistente Alarme möglich. Auch bei Alarmen anderer mapp Komponenten kann die Einstellung definiert werden.

Asynchrone Alarme werden nicht im gleichen Kontext aktiviert, zurückgesetzt und quittiert wo sie aufgerufen wurden, sondern werden an MpAlarmXCore weitergeleitet und dort verarbeitet. Bei dieser Weiterleitung wird lediglich der Zeitstempel des Alarms mitgeliefert. Weitere Informationen wie Alarmsnippets oder in der Konfiguration definierte Alarm Mappings werden im Kontext von MpAlarmXCore aufgelöst. Das heißt die Information von Alarmsnippets sowie auch durch Alarm Mappings definierte Aktionen werden zeitverzögert, abhängig davon in welcher Taskklasse sich MpAlarmXCore befindet, aufgelöst.

Zeitstempel, sowie Reihenfolge der Alarme wird unabhängig davon ausgewertet, ob der Alarm als synchron oder asynchron konfiguriert ist.

Ein wesentlicher Vorteile von asynchron konfigurierten Alarmen ist die verbesserte mapp AlarmX Performance, da weniger Daten in kuzer Zeit ausgewertet werden müssen. Das Kommando um Alarme zu aktivieren, zurückzusetzen oder zu quittieren kann in einer höher prioren Taskklasse stattfinden. Die Zykluszeit sowie Taskklassen-Toleranz ist dabei standardmäßig niedriger und wird durch das Setzen von Alarmen nicht negativ beeinflusst. Die Alarmverwaltung über MpAlarmXCore kann in einer höheren Taskklasse verwaltet werden, wo die Zyklus-, sowie Toleranz-Zeit standardmäßig höher ist.

1.1.6 Auditing der Alarmverwaltung

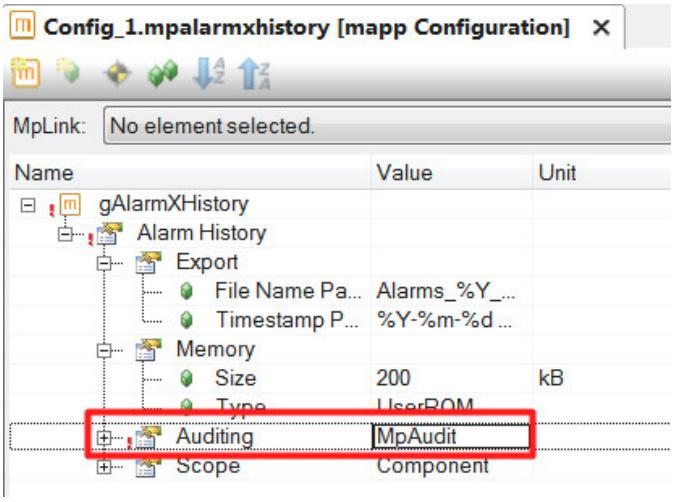
Für diese Komponente können Ereignisse über MpAudit aufgezeichnet werden. Für mehr Information wie ein Eintrag erstellt werden kann, siehe Abschnitt Definieren des Ereignistexts.

Folgende Ereignisse können für diese mapp Komponenten auftreten:

Alarmsystem Ereignis

Mithilfe dieses Ereignistyps können Ereignisse rund um das Alarmsystem, wie zum Beispiel wann ein Alarm aufgetreten ist, aufgezeichnet werden.

Damit die Ereignisse aufgezeichnet werden können, muss in der MpAlarmXHistory Konfiguration die Option "Auditing" freigeschalten werden.



Ereignis ID 33	Komponenten MpAlarmX Komponenten
Ereignisse	
ev	Ereignis ID
idx	Liefert den Index des aktuell generierten Ereignisses. Der Wert wird bei jedem neuen Ereignis inkrementiert. Das erste generierte Ereignis in der Ereignisliste hat somit den Wert 1, dass fünfzigste Ereignis hat den Index 50, das hundertste den Index 100 und so weiter.
name	Alarm Name
msg	Alarmnachricht
code	Alarm Code
stold	Letzter Zustand des Alarms. Kann die Zustände "Inactive", "Active", "Acknowledged" und "Unacknowledged" annehmen
stnew	Aktueller Zustand des Alarms. Kann die Zustände "Inactive", "Active", "Acknowledged" und "Unacknowledged" annehmen
sev	Alarm Level (Schweregrad)
time	Zeit, zu welcher sich der Alarm Zustand verändert hat. Kann abweichen von der Zeit, an welcher das Ereignis geloggt wurde

Tabelle 1: MpAlarmX Ereignis

Wie der Ereignistext angegeben werden kann ist in Abschnitt Verwendung des neuen Formats erklärt.

Beispiel neues Format:

- **'Alarm: {&name} Time: {&time[TIME=%Y-%m-%d %H:%M]}, old State: {&stold}, new State:{&stnew}'** ausgewertet: 'Alarm: TempHigh Time: 2018-07-18 07:26 old State: Inactive, new State: Active' , was bedeutet der Alarm "TempHigh" wurde zur angegebenen Zeit aktiv.

Beispiel altes Format:

- **'Alarm: [%name] Time: [%time$[TIME=%Y-%m-%d %H:%M]], old State: [%stold], new State:[%stnew]'** ausgewertet: 'Alarm: TempHigh Time: 2018-07-18 07:26 old State: Inactive, new State: Active' , was bedeutet der Alarm "TempHigh" wurde zur angegebenen Zeit aktiv.

1.1.7 Einheiten von Variablen anzeigen

Um festzulegen welche Variable welche Einheit besitzen soll, siehe Einheitenverwaltung.

Über den "Language" Parameter an MpAlarmXHistory, MpAlarmXHistoryUIConnectType und MpAlarmXListUI-ConnectType kann definiert werden in welcher Sprache bzw. Einheit die Alarme angezeigt werden.

Zu diesem Thema gibt es den Anwendungsfall Einheiten im Alarmtext anzeigen.

Auf unserem B&R Tutorial Portal wird im Tutorial "Einheitenverwaltung in mapp" (GER/ENG) erklärt, wie die Einheitenverwaltung in mapp Komponenten funktioniert. (Zugangsdaten)

Definition von Sprache und Einheit für Anzeige und Export

Die Darstellung der Einheit wird im Punkt "Darstellung der Einheit innerhalb eines Textes" definiert. Beim Export bzw. bei der Anzeige ist es allerdings notwendig die Sprache, sowie das Einheitensystem anzugeben. Das erfolgt bei den oben genannten Komponenten über den Eingang, bzw. Parameter "Language". Um beim Export bzw. der Anzeige Sprache und Einheitensystem festzulegen ist der Eingang "Language" auf folgende Weise zu verwenden:

'Sprache|Einheitensystem'

Einige Beispiele:

- **'de'** : Sprache ist Deutsch, Werte in Engineering-Einheit
- **'en|imperial-us'** : Sprache Englisch, Einheitensystem ist Imperial-US
- **'de|metric'** : Sprache Deutsch, Einheitensystem ist Metrisch
- **'|imperial'** : Default-Sprache, Einheitensystem ist Imperial

Details zur Definition der Anzeige-Einheiten sind im Abschnitt MpComUnit Konfiguration zu finden.

Darstellung der Einheit innerhalb eines Textes

Neben dem Darstellen und Umrechnen der Prozesswerte, welche Teil eines Textes sind (z.B. Alarme, Audit-Ereignisse), kann definiert werden in welcher Form die Einheit dargestellt werden soll:

- **Symbol %s:** Hiermit wird definiert, dass das Symbol der aktuellen Anzeige-Einheit angezeigt wird (z.B. "mm").
- **Kurztext %n:** Hiermit wird definiert, dass der vollständige Name der aktuellen Anzeige-Einheit angezeigt wird (z.B. "Millimeter").
- **Beschreibung %d:** Hiermit wird definiert, dass die Beschreibung der aktuellen Anzeige-Einheit angezeigt wird (z.B. "Abstand gemessen in Millimeter"). Nicht für jede Einheit existiert auch eine Beschreibung.

Die Umrechnung erfolgt automatisch, die Darstellung der Einheit wird über folgende Syntax definiert:

{&key}{&key[UNIT= xxx]}

- {&key}: Definiert die Prozessvariable, von welcher der Wert angezeigt werden soll. Der Wert wird automatisch auf die aktuelle Anzeige-Einheit umgerechnet. Wurde keine Anzeige-Einheit definiert, wird der Wert in seiner Engineering-Unit angegeben.
- {&key[UNIT=xx]}: Druckt den Einheiten-Text für die Einheit, die dem Wert zugeordnet ist.
  Unter "xx" können die vorher erwähnten Parameter (%s, %n, %d) angegeben werden.

Einige Beispiele:

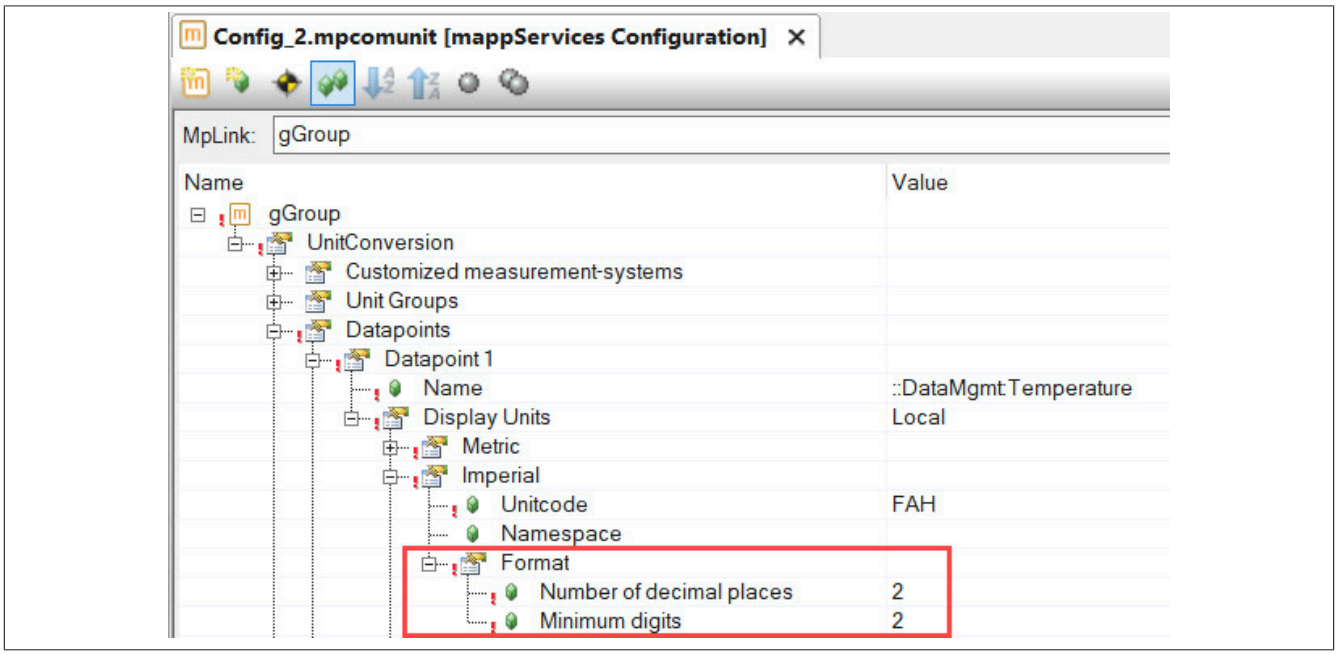
- "Die aktuelle Temperatur {&actualTemperature}{&actualTemperature[UNIT=%s]} hat den kritischen Wert erreicht!"
  Die aktuelle Temperatur 120°C hat den kritischen Wert erreicht!
- "Die Achse steht an ihrer Endposition: {&actualPosition}{&actualPosition[UNIT= %n]}!"
  Die Achse steht an ihrer Endposition: 500.0 Millimeter!
- "Der aktuelle Druck {&currentPressure}{&currentPressure[UNIT= %s]} überschreitet den empfohlenen Maximalwert!"
  Der aktuelle Druck 50 bar überschreitet den empfohlenen Maximalwert!

Definieren des Formats

Für eine Prozessvariable kann definiert werden in welchem Format der Wert angezeigt werden soll. Das bedeutet es kann festgelegt werden, wie viele Nachkommastellen und welche Mindestanzahl an Gesamtstellen verwendet werden sollen. Dazu gibt es zwei Möglichkeiten dies festzulegen:

MpComUnit Konfiguration

Über die MpComUnit Konfiguration wird bereits festgelegt, welche Einheit pro Einheitengruppe für die gewünschte Prozessvariable verwendet werden soll. Zusätzlich kann dabei noch angegeben werden in welchem Format die Prozessvariable angezeigt werden soll. Es wird die Anzahl der Nachkommastellen bzw. die Mindestanzahl an Gesamtstellen über die Konfiguration festgelegt:



Für mehr Information siehe Abschnitt Festlegen eines Formats.

Textsystem

Wird die Prozessvariable in der Alarmnachricht lokalisiert über das Textsystem angegeben, so kann das Format auch über das Textsystem bestimmt werden. Dies kann folgendermaßen aussehen:

- "Die Temperatur ist zu hoch: {&TempSnippet|.2f}{&TempSnippet[UNIT=%s]}"
  Die Temperatur ist zu hoch: 65.85 °C

Durch |.2f wird bestimmt wie viele Nachkommastellen angezeigt werden. Werden 3 Nachkommastellen benötigt, so wird |.3f angegeben.

Für mehr Information dazu siehe Abschnitt Einbinden von Applikationsdaten in der Konfiguration .

> **Formatierungsinformationen, die in der MpComUnit Konfiguration angegeben werden, können mit Formatierungsinformationen, die direkt im Textsystem übergeben werden, in Konflikt geraten. Aus diesem Grund darf nur eine der beiden Varianten verwendet werden.**
>
> **Werden beispielsweise im Textsystem 2 Nachkommastellen angegeben und es wurde in der MpComUnit Konfiguration eine Mindestanzahl von Gesamtstellen von 2 definiert, so werden die Nachkommastellen im Textsystem von der MpComUnit Konfiguration als Gesamtstellen interpretiert. Wurde beispielsweise der Wert 120 aufgezeichnet, so wird der Wert durch die doppelte Formatierung auf 12 gekürzt. Aus diesem Grund sollte nur eine der beiden Formatierungs-Varianten verwendet werden.**

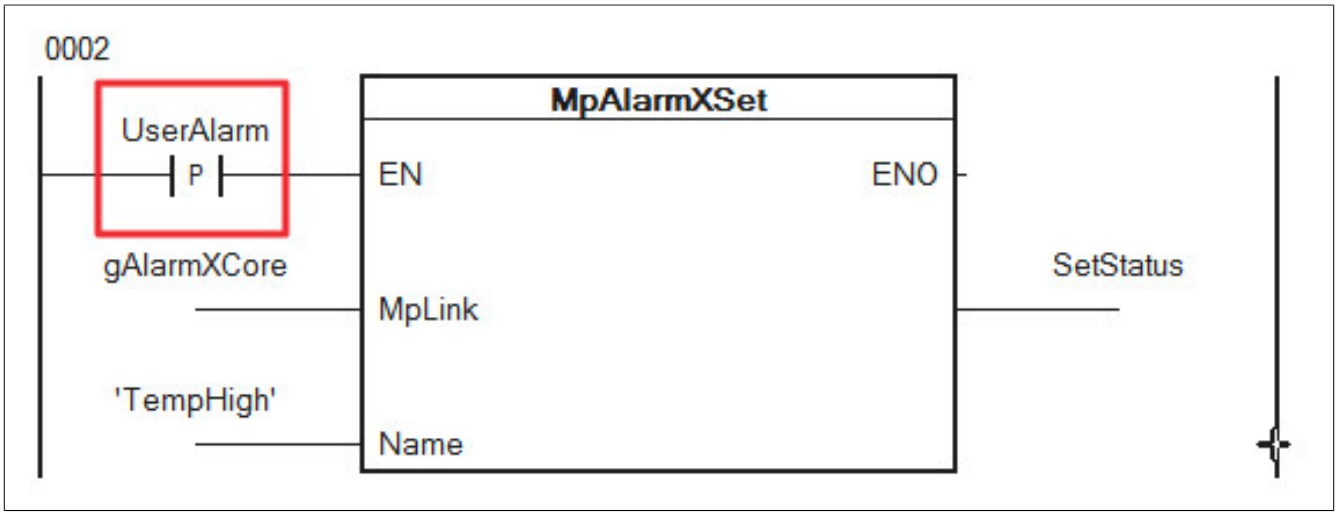
1.1.8 Best Practise

In diesem Abschnitt wird erklärt worauf man bei der Implementierung von mapp AlarmX achten sollte.

Aktivieren von Alarmen

Es muss darauf geachtet werden wie ein Alarm, welcher als Edge Alarm konfiguriert wurde, mit MpAlarmXSet aktiviert wird. Ein Edge-Alarm setzt sich automatisch selbst zurück.

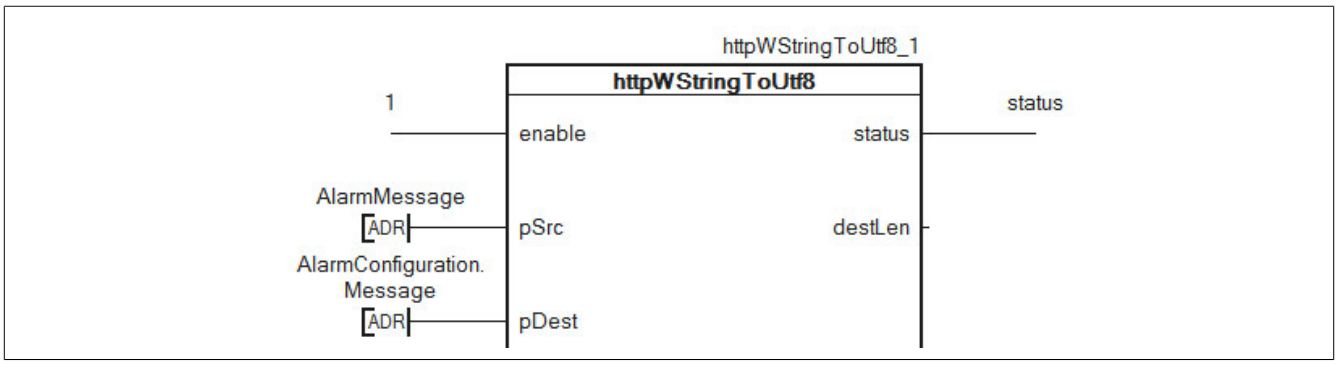
Wird die Funktion zyklisch aufgerufen, so wird auch der Alarm mit jedem Zyklus aktiviert. Das heißt pro Zyklus wird ein weiterer Alarm an MpAlarmXCore angezeigt. Als Abhilfe kann zum Beispiel der Alarm nur bei einer positiven Flanke eines Triggers aktiviert werden. Dadurch wird der Alarm nur einmal aktiviert.



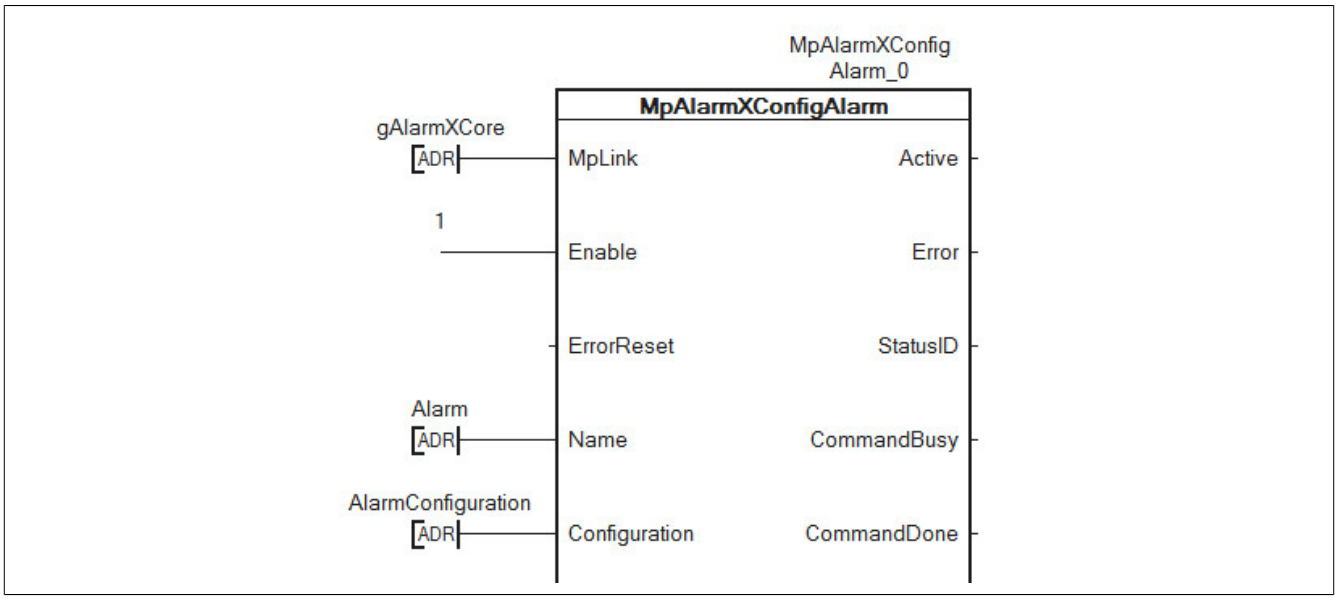
Alarmtexte mit MpAlarmXConfigAlarm ändern

Soll, zum Beispiel, der Alarmtext ("Message") oder die zusätzliche Information ("AdditionalInformation") eines Alarms mithilfe von MpAlarmXConfigAlarm angepasst werden, so kann eine Konvertierung des Strings durchgeführt werden. Dies kann mithilfe der Funktion httpWStringToUtf8 realisiert werden. Somit kann bei der Alarmnachricht zum Beispiel auch Umlaute wie "ä" oder "ü" verwendet werden.

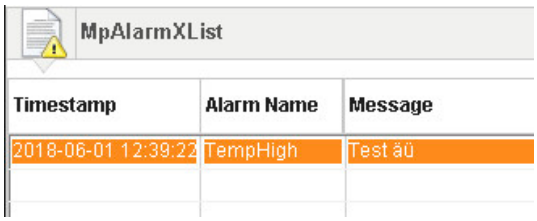
In diesem Beispiel wird die Alarmnachricht über die Variable "AlarmMessage" (Datentyp WSTRING) an die Variable "AlarmConfiguration.Message" (Datentyp STRING) übergeben.



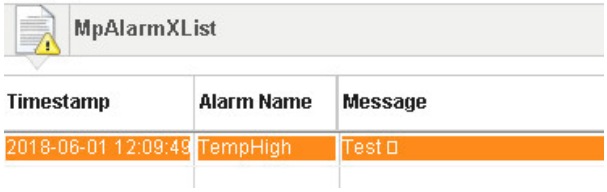
"AlarmConfiguration" wird am Eingangsparameter "Configuration" von MpAlarmXConfigAlarm angegeben:



Dadurch wird die Alarmnachricht in der Visualisierung (bei Verwendung von Umlauten) richtig angezeigt:



Ansonsten kann es bei der Anzeige zu Problemen kommen:



Alarme zur Laufzeit erstellen

Alarme können zur Laufzeit mithilfe von MpAlarmXConfigAlarm erstellt werden.

Eine weitere Möglichkeit bietet der Funktionsbautein MpComConfigManager. Mithilfe dieses Funktionsbausteins können Benutzeralarme schnell und einfach importiert werden.

Wie Benutzeralarme importiert werden können ist in diesem Anwendungsfall erklärt.

1.1.8.1 Zeitliches Verhalten beim Aktivieren/Deaktivieren von Alarmen im System

In diesem Abschnitt wird erklärt, wann Alarme im System genau aktiviert bzw. deaktiviert werden.

Um sicherzustellen, dass MpAlarmXCore Alarme sofort beim Hochstart erkennt, wird dazu geraten diesen Funktionsbaustein bereits im INIT Programm über den "Enable" Eingang zu aktivieren. Damit wird die Initialisierung aller Alarme bereits im INIT Programm durchgeführt.

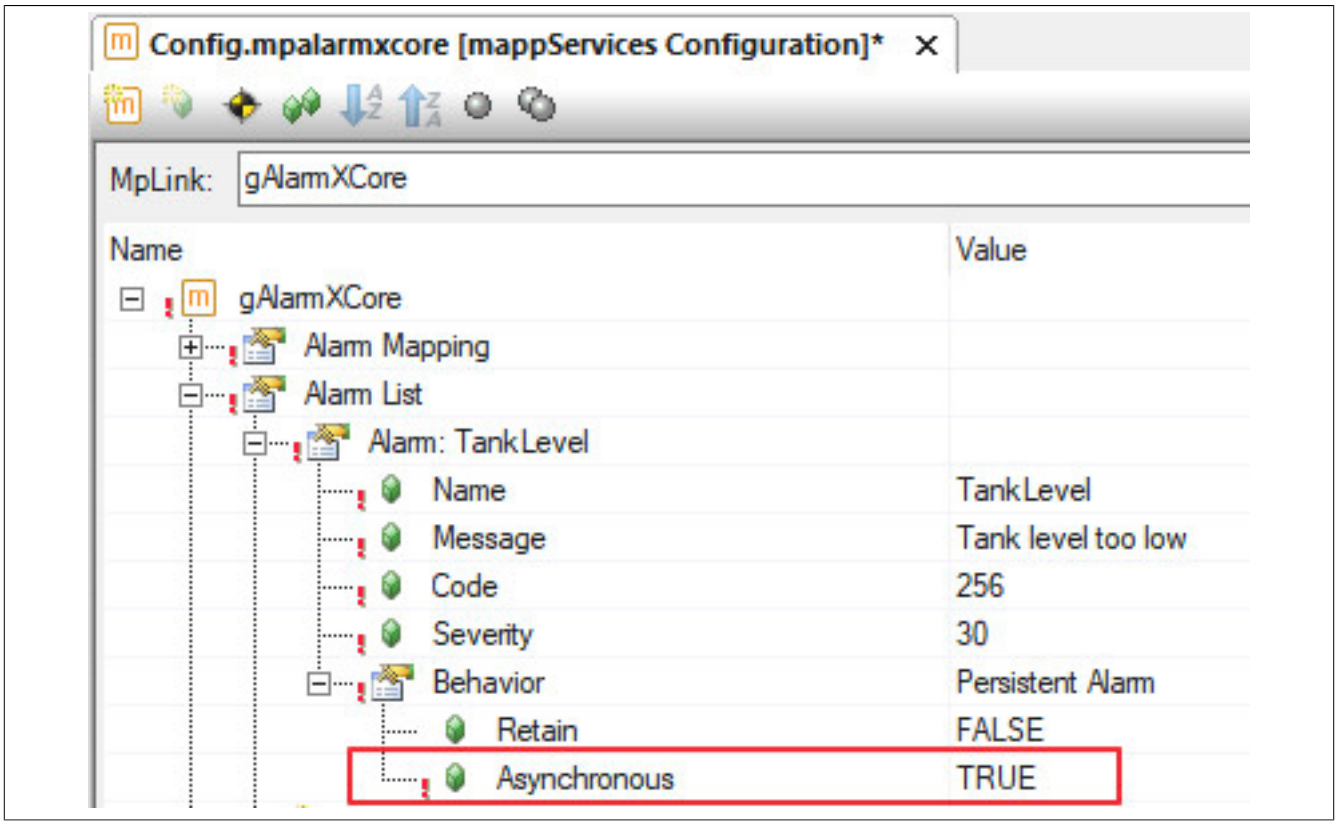
Grundsätzlich muss zwischen 3 verschiedenen Alarm-Typen differenziert werden.

Alarme welche in der Applikation aktiviert werden

Alarme welche in der Applikation aktiviert bzw. deaktiviert werden, haben als Alarmverhalten "Edge Alarm", "Persistent Alarm" oder "User defined". Das Alarmverhalten wird in der MpAlarmXCore Konfiguration festgelegt. Für mehr Information siehe Abschnitt vordefiniertes Alarmverhalten. Das zeitliche Verhalten hängt von 2 verschiedenen Faktoren ab. Erstens, ob ein Alarm synchron oder asynchron behandelt wird und zweitens, ob der Alarm durch Funktionen wie MpAlarmXSet oder durch den Funktionsbaustein MpAlarmXAlarmControl kontrolliert wird.

Synchrone vs. Asynchrone Alarme

Standardmäßig ist ein Alarm synchron. Dies kann durch die Konfigurations-Eigenschaft "Asynchronous" geändert werden.



Synchrone Alarme werden im gleichen Kontext aktiviert, zurückgesetzt oder quittiert wo sie aufgerufen werden. Es ist nicht relevant in welcher Taskklasse sich MpAlarmXCore befindet.

Ist MpAlarmXCore in einer Taskklasse mit 100ms und ein Alarm wir in einer 10ms Taskklasse aufgerufen, so wird der Alarm direkt nach dem Aufruf in der 10ms Taskklasse im System aktiv. Auch eine mögliche Reaktion ist sofort aktiv und kann in der Code-Zeile danach mithilfe von MpAlarmXCheckReaction überprüft werden.

Asynchrone Alarme werden nicht im gleichen Kontext aktiviert, zurückgesetzt und quittiert wo sie aufgerufen wurden, sondern werden an MpAlarmXCore weitergeleitet und dort verarbeitet. Bei dieser Weiterleitung wird lediglich der Zeitstempel des Alarms mitgeliefert. Weitere Informationen wie Alarmsnippets oder in der Konfiguration definierte Alarm Mappings werden im Kontext von MpAlarmXCore aufgelöst.

Für mehr Information siehe Abschnitt Synchrone und asynchrone Alarme.

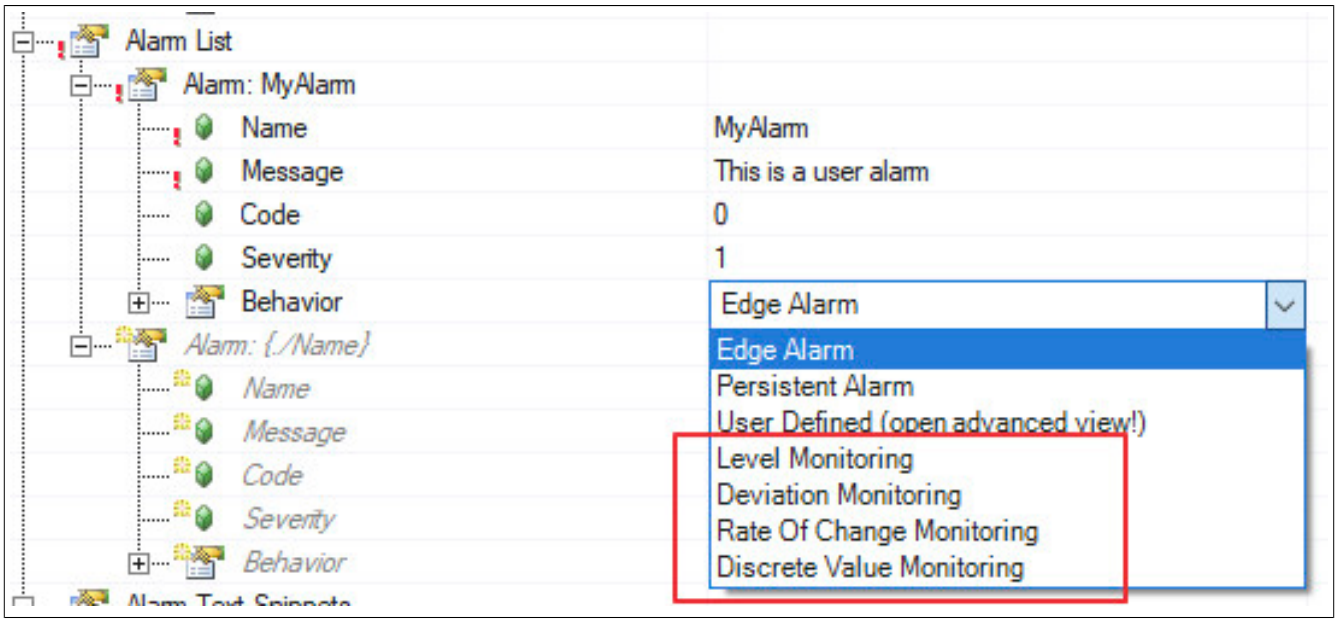
Verwenden von Funktionen vs. Funktionsbaustein

Wird die Funktion MpAlarmXSet verwendet um einen Alarm zu aktivieren, so wird, sobald die Funktion aufgerufen wird, die Alarmkonfiguration gelesen. Es werden definierte Alarmeigenschaften, Alarmsnippets, Alarm Mappings usw. aufgelöst. Dies geschieht bei jedem Funktionsaufruf. Hiervon ausgenommen sind asynchrone Alarme. In diesem Fall werden Alarmsnippets, Alarm Mappings usw. von MpAlarmXCore aufgelöst.

Beim Verwenden des Funktionsbausteins MpAlarmXAlarmControl wird die Alarmkonfiguration einmalig ausgelesen. Dies geschieht während des Aktivier-Vorgangs ("Enable = `TRUE`") des Funktionsbausteins. Dadurch ist das Aktivieren, Zurücksetzen sowie Quittieren des Alarms weniger zeitintensiv als beim Verwenden von Funktionen. Wird die Alarmkonfiguration zur Laufzeit geändert, so wird die Änderung automatisch an MpAlarmXAlarmControl weitergeleitet.

Alarme welche über die Konfiguration geregelt werden

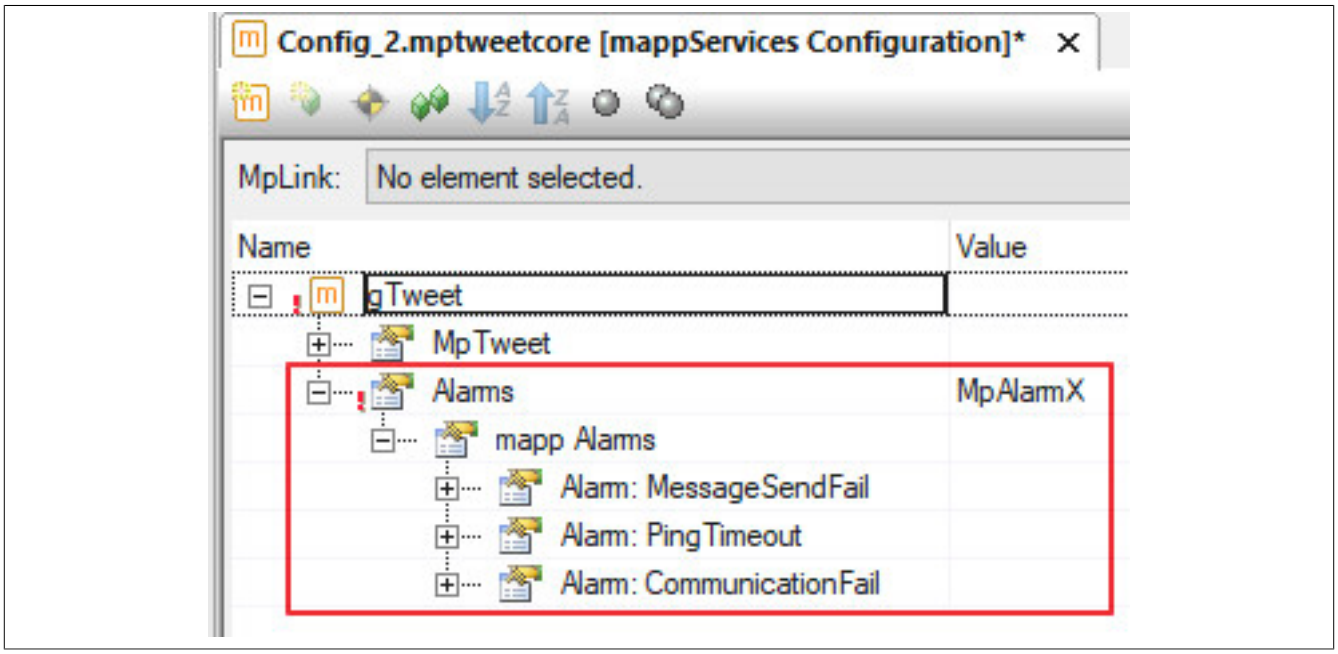
Alarme welche über die Konfiguration geregelt werden, besitzen als Alarmverhalten einer der Monitoring-Typen. Das Alarmverhalten wird in der MpAlarmXCore Konfiguration festgelegt. Für mehr Information zu diesen Alarmen siehe Alarm Monitoring.



Diese Art von Alarme werden mithilfe von MpAlarmXCore aktiviert. Bei jedem Funktionsbaustein-Aufruf wird geprüft, ob einer der konfigurierten Alarme aktiviert werden muss oder nicht. Ist dies der Fall, so wird der Alarm im System aktiviert. Befindet sich also MpAlarmXCore in einer 100ms Taskklasse, so wird alle 100ms geprüft ob ein Alarm aktiviert bzw. deaktiviert werden muss oder nicht.

mapp Alarme

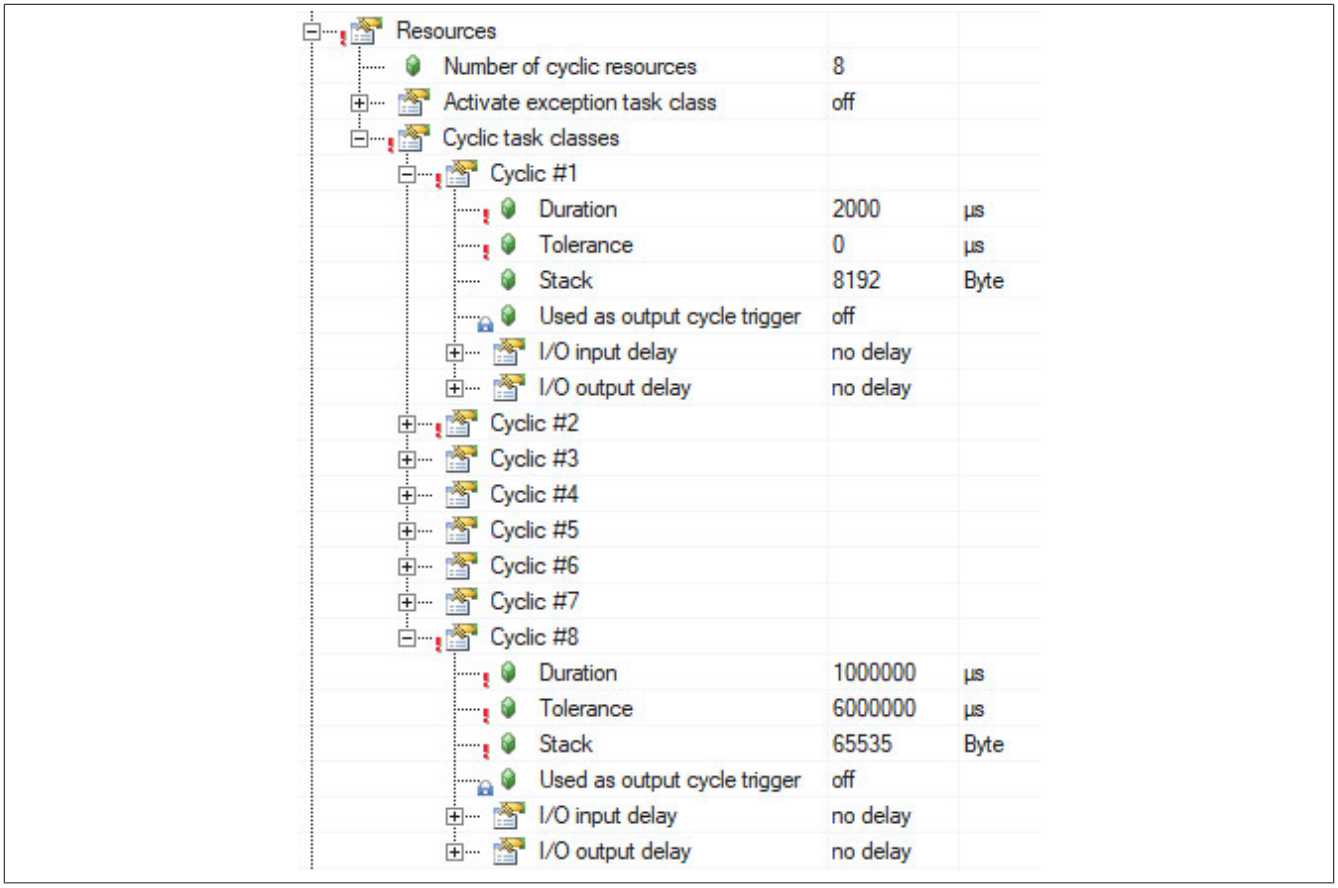
Alarme von mapp Komponenten werden in der jeweiligen Konfiguration verwaltet. Dafür muss als Alarmsystem "MpAlarmX" verwendet werden. Dadurch werden zum Beispiel die Alarme von mapp Tweet automatisch an mapp AlarmX weitergeleitet.



Tritt in einer mapp Komponente ein Alarm auf, so wird dieser direkt an mapp AlarmX weitergeleitet. Abhängig davon in welcher Taskklasse sich die mapp Komponente befindet wird der Alarm aktiviert. Befindet sich also mapp Tweet in einer 100ms Taskklasse, so wird alle 100ms geprüft ob ein mapp Alarm aktiviert bzw. deaktiviert werden muss oder nicht.

Verbessern der Alarmperformance

Treten beim Setzen von vielen Alarmen zur selben Zeit Probleme auf, beispielsweise eine Zykluszeitverletzung, dann kann dem mithilfe von asynchronen Alarmen entgegengewirkt werden. Dafür werden alle nicht-zeit-kritischen Alarme als asnychron deklariert. Das bedeutet ein Großteil der Aktivitäten die beim Setzen passieren werden in den Kontext von MpAlarmXCore verlagert. In folgendem Beispiel wird gezeigt, wie sich diese Verlagerung auswirkt. Verwendet wurde eine X20CP1586 mit folgenden Einstellungen für Taskklasse 1 und Taskklasse 8:



Funktion / Funktionsbaustein	Alarmverhalten	Alarme
MpAlarmXSet	synchron	< 10
MpAlarmXAlarmControl	synchron	< 10
MpAlarmXSet	asynchron	< 130
MpAlarmXAlarmControl	asynchron	< 1000

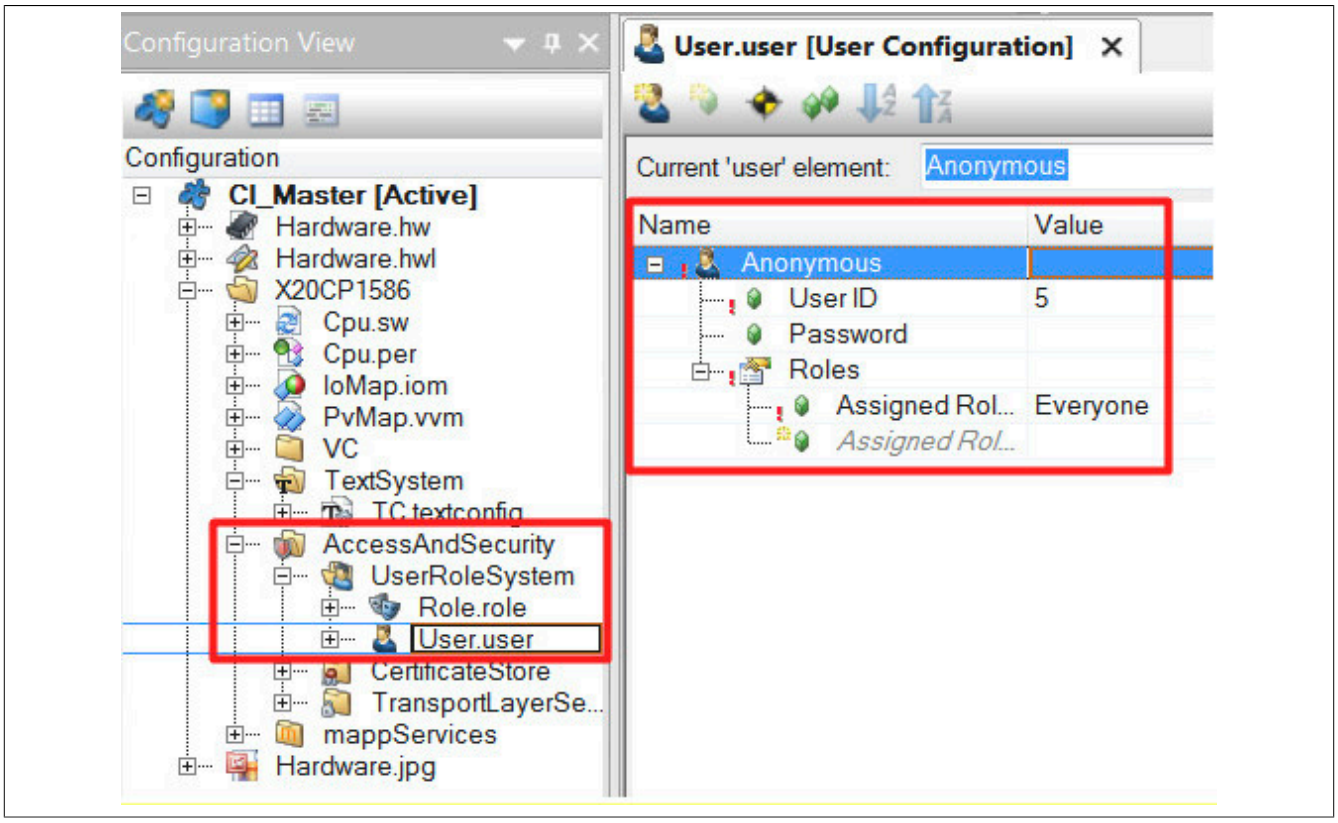
Entscheidend ob eine Verbesserung erreicht wird, ist dass MpAlarmXCore in einer Taskklasse mit niedrigerer Priorität und vor allem hoher Toleranz ausgeführt wird. Nur dann wird die Steuerung durch eine Verlagerung der Aktivitäten während dem Setzen eines Alarms auch wirklich entlastet.

1.1.9 Anbindung an mapp View

Bei der Verwendung von mapp AlarmX in Kombination mit dem AlarmList-Widget, kann die Alarmliste schnell und einfach in einer mapp View Visualisierung dargestellt werden. Das Widget stellt mehrere Aktionen und Ereignisse zur Verfügung um mit der Alarmliste zu interagieren.

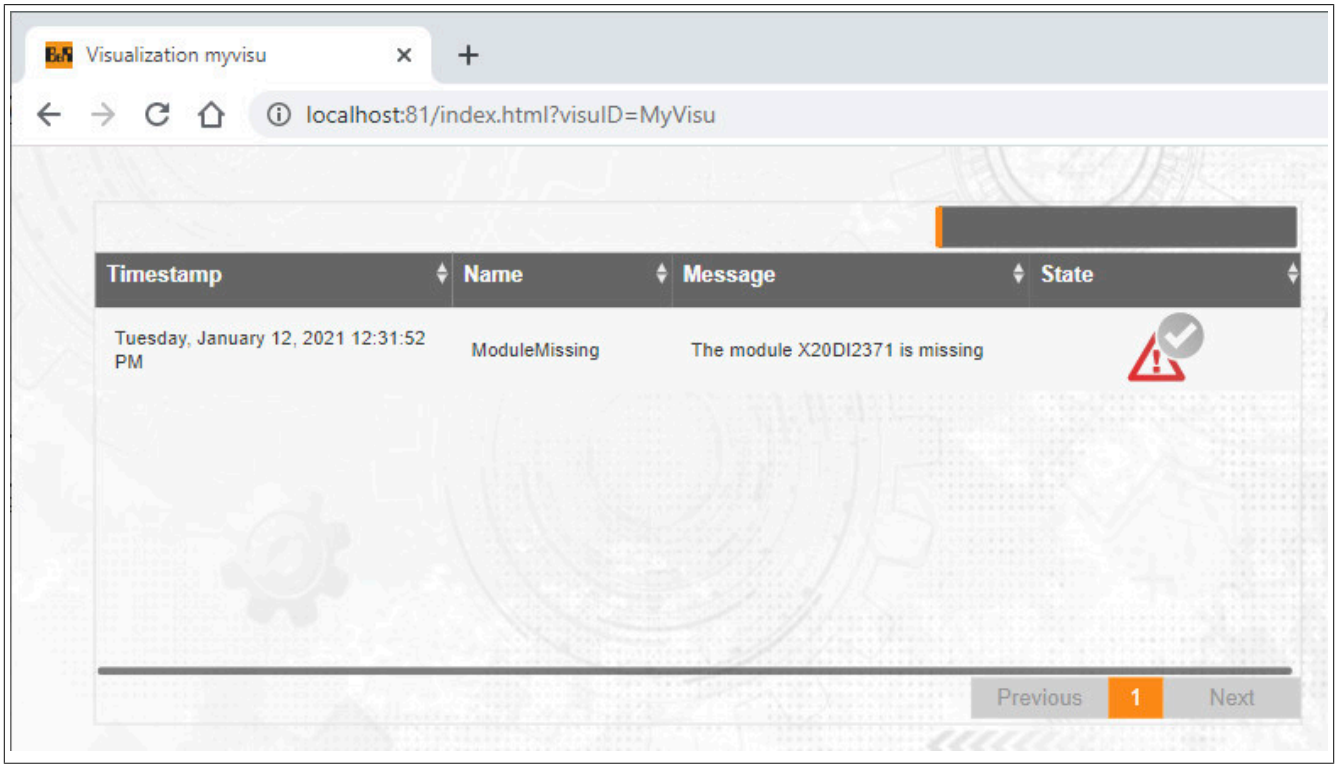
Um die Alarmhistorie anzeigen zu lassen kann das AlarmHistory-Widget verwendet werden.

Es gilt zu beachten, dass für die Funktion der Benutzer "Anonymous" mit der Rolle "Everyone" benötigt wird. Diese sind standardmäßig im Benutzer-Rollen-System von Automation Studio enthalten und sollten nicht verändert werden.



Wie mapp AlarmX in Kombination mit dem AlarmList-Widget verwendet werden kann, ist im Getting Started Benutzeralarme mit mapp View Visualisierung erklärt.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp AlarmX und dem AlarmList-Widget (GER/ENG) zu finden. (Zugangsdaten)



1.1.10 Konfiguration während der Laufzeit

Die Konfigurations Parameter können zur Laufzeit über MpAlarmXAlarmConfigType und MpAlarmXMappingConfigType verändert werden. Dabei ist für die unterschiedlichen Parameter jedoch zu beachten in welchem Zustand sich die Komponente befinden muss um eine Änderung zuzulassen. Folgende Zustände sind möglich:

- : Dieses Symbol bedeutet, dass dieser Parameter mit einer neue positiven Flanke am "Enable" Eingang von MpAlarmXCore übernommen wird.

Jeder Parameter der Konfigurations Struktur (MpAlarmXAlarmConfigType, MpAlarmXMappingConfigType) ist mit einem dieser Symbole ausgestattet.

1.1.11 Rahmenbedingungen

In diesem Abschnitt werden Rahmenbedingungen von mapp AlarmX aufgelistet:

Verwendung des Alarmsystems

In einem Projekt darf man entweder mit mapp Alarm oder mit mapp AlarmX arbeiten. Nicht mit Beidem parallel. mapp AlarmX ist ein eigenständiges Alarmsystem, unabhängig von einer Visualisierung. Weiters bietet mapp AlarmX einen großen Funktionsumfang, welcher die Konfiguration des Alarmsystems, sowie erweiterte Funktionen zur Laufzeit anbietet. mapp Alarm ist als Erweiterung für das Alarmsystem in VC4 gedacht.

Alarm Monitoring

Folgende Datentypen werden für das Alarm-Monitoring unterstützt:

- BOOL
- SINT
- INT
- DINT
- USINT
- UINT
- UDINT
- REAL
- LREAL
- Abgeleitete Datentypen können nur mit AR D4.26 oder AR4.34 registriert werden
- Es können einzelne Variablen aber auch Strukturen registriert werden

Batterie gepufferter Speicher (SRAM)

Für manche Funktionen innerhalb mapp AlarmX kann ein Batterie gepufferter Speicher (SRAM) verwendet werden. Es gilt zu beachten, dass dafür ein SRAM-Speicher am Zielsystem vorhanden sein muss. Folgende Funktionen können einen SRAM-Speicher verwenden:

- Retain Alarme
- Alarmhistorie

Alarm Textsnippets

Alarm Textsnippets, welche in der MpAlarmXCore Konfiguration definiert werden, können nur für Alarme verwendet werden, welche im Abschnitt Alarmlist definiert wurden. Alarm Textsnippets können nicht in den Alarmen anderer mapp Komponenten verwendet werden.

Umkonfigurieren/ neu erstellen von Alarmen durch MpAlarmXConfigAlarm

Mithilfe des Funktionsbausteins MpAlarmXConfigAlarm können bestehende Alarme bzw. neue Alarme konfiguriert werden. Das Alarmverhalten wird dabei über die Konfigurationsstruktur bestimmt. Es gilt zu beachten, dass bestehende Alarme bzw. neue Alarme immer mit dem Alarmverhalten "User defined" abgespeichert werden. Es können keine Monitored-Alarme durch MpAlarmXConfigAlarm erstellt werden.

1.2 Getting Started

Hier sind Getting Started Kapitel zu finden.

1.2.1 Benutzeralarme erstellen

In diesem Abschnitt wird Schritt für Schritt beschrieben wie man mithilfe von MpAlarmXCore, MpAlarmXHistory, MpAlarmXSet und MpAlarmXReset Benutzeralarme erstellt.

1.2.1.1 Projekt erstellen

Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

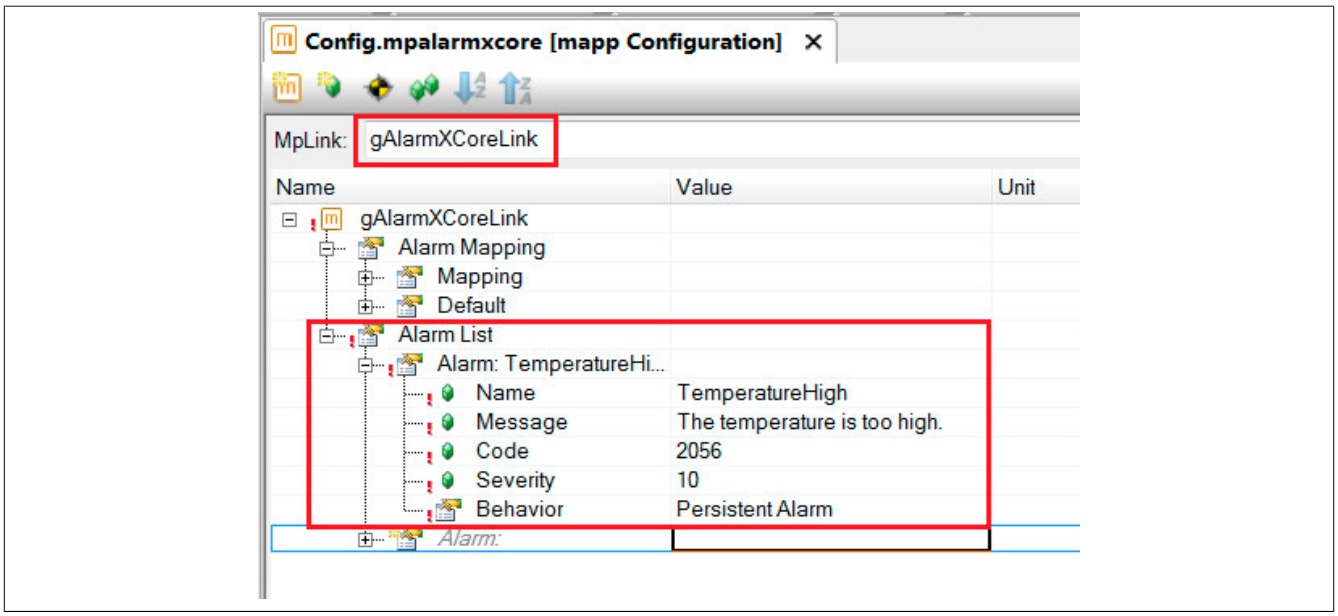
1.2.1.2 mapp Komponente einfügen

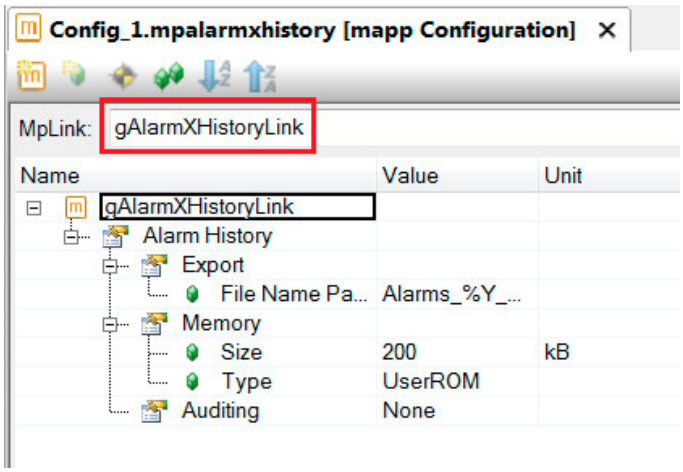
Konfiguration MpAlarmXCore und MpAlarmXHistory einfügen

Beim nächsten Schritt wird über die Toolbox die gewünschte mapp Komponente ausgewählt und die Konfiguration dann per Drag&Drop im "mapp Services" Ordner eingefügt.

MpLink benennen

Hier befindet sich der mapp Link, welcher die mapp Komponente in der "world of mapp" eindeutig identifiziert. Der mapp Link stellt die Verbindung zwischen Programmierung und Konfiguration her. Unter "Name" wird der Name des Benutzeralarms angegeben, welcher später zur Identifikation in der Applikation dient, wenn es um das Setzen, Rücksetzen des Alarms geht. Der Alarmtext, welcher später auch in der Visualisierung angezeigt wird, wird unter "Message" eingefügt. "Code" und "Severity" bestimmen den Alarmcode und die Schwere des Alarms. Das Verhalten des Alarms wird unter "Behaviour" festgelegt. In unserem Fall handelt es sich um einen anhaltenden Alarm (Persistent), da der Alarm so lange aktiv sein soll, bis sich die Temperatur wieder im Normalbereich befindet.





1.2.1.3 Programm hinzufügen

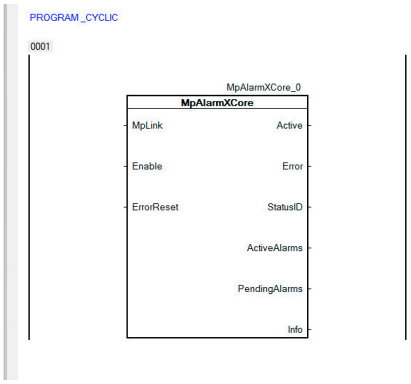
Der nächste Schritt ist das Einfügen eines Kontaktplan-Programms. Für mehr Information siehe hier.

1.2.1.4 Funktionsbausteine einfügen und parametrieren

In diesem Abschnitt wird erklärt, welche Funktionsbausteine eingefügt und parametriert werden.

1.2.1.4.1 MpAlarmXCore einfügen

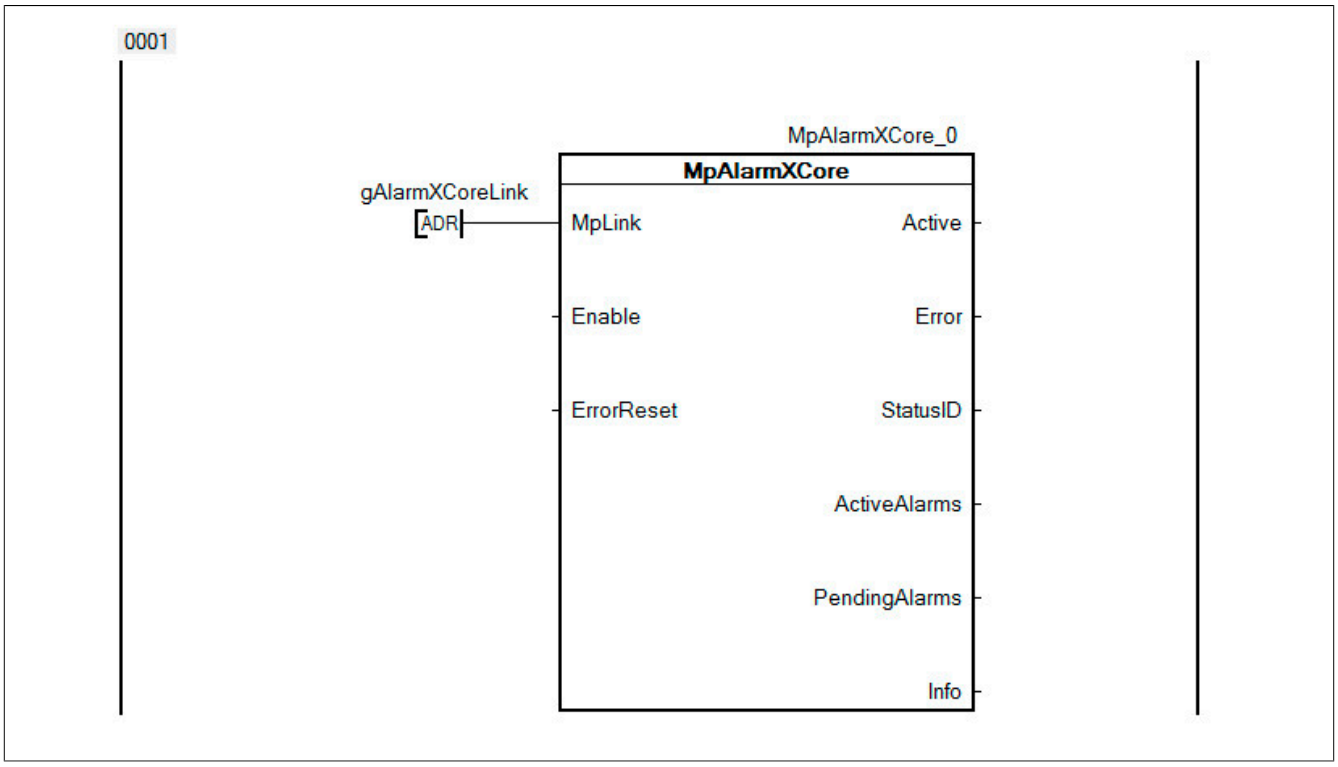
Es wird der Funktionsbaustein MpAlarmXCore eingefügt.



1.2.1.4.2 Parametrieren von MpAlarmXCore

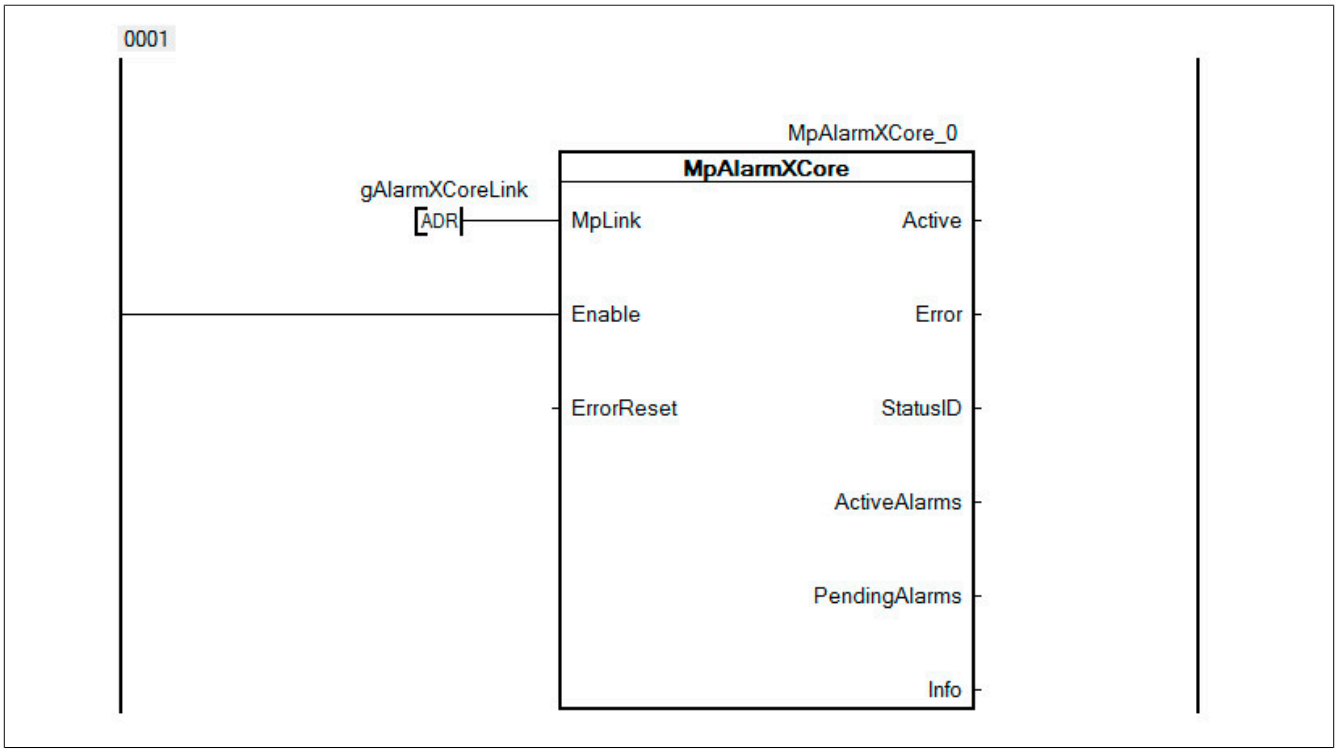
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gAlarmX-CoreLink erstellt haben.



Aktivieren des Funktionsbausteins

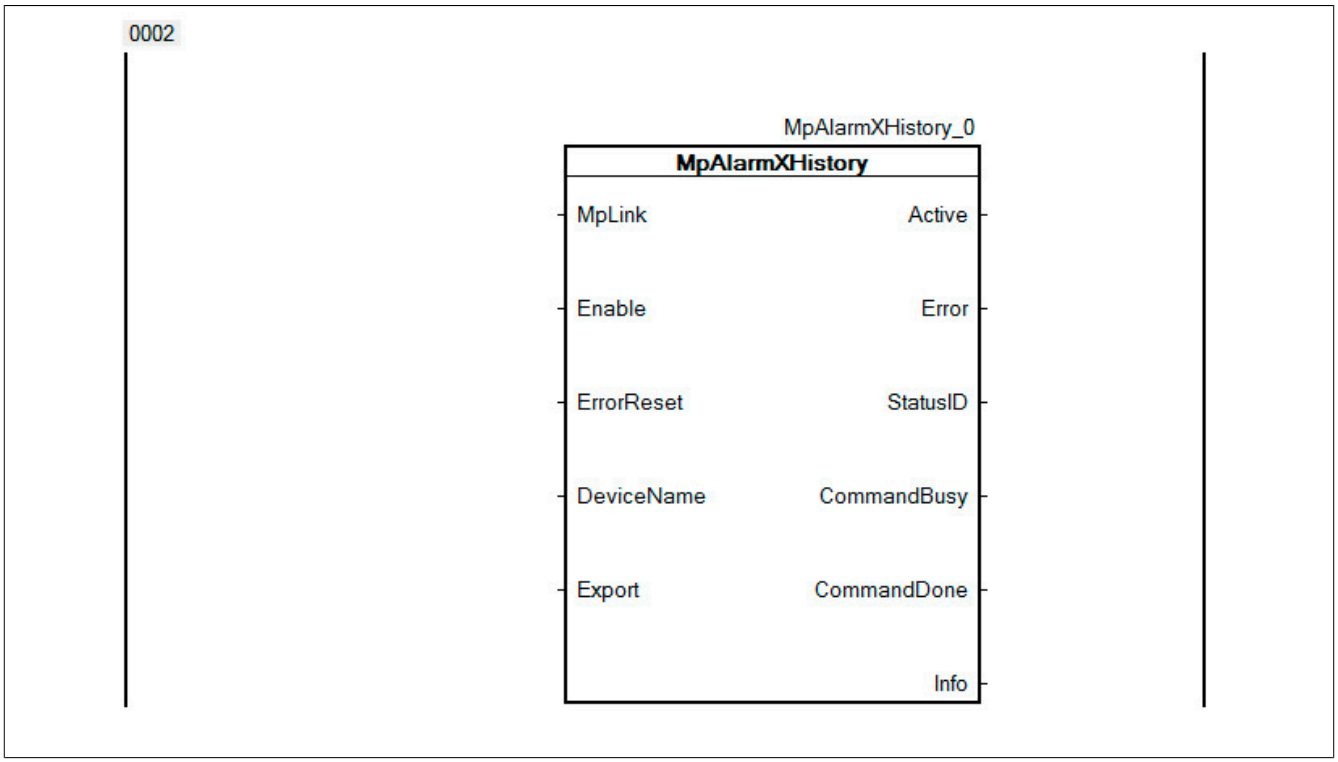
Der Funktionsbaustein wird durch den Eingang "Enable" aktiviert.



1.2.1.4.3 MpAlarmXHistory einfügen

Funktionsbaustein einfügen

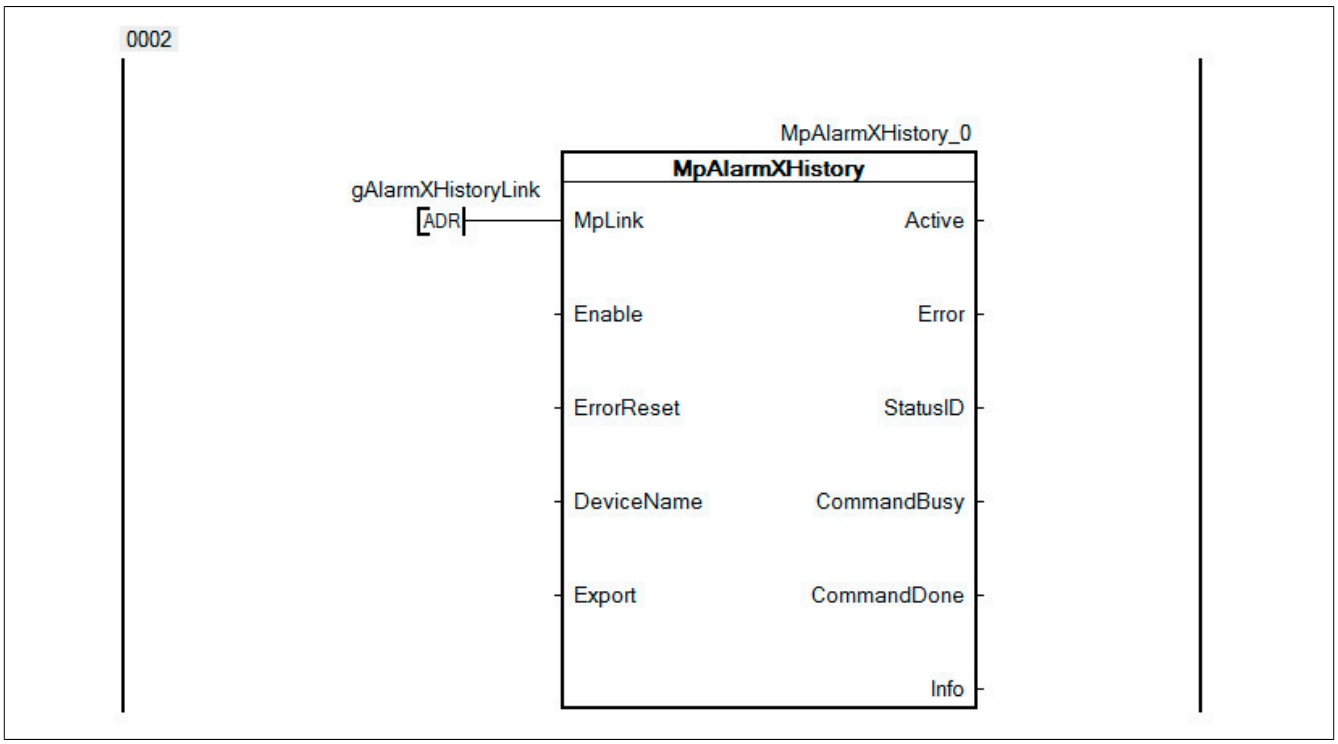
Es wird der Funktionsbaustein MpAlarmXHistory eingefügt.



1.2.1.4.4 Parametrieren von MpAlarmXHistory

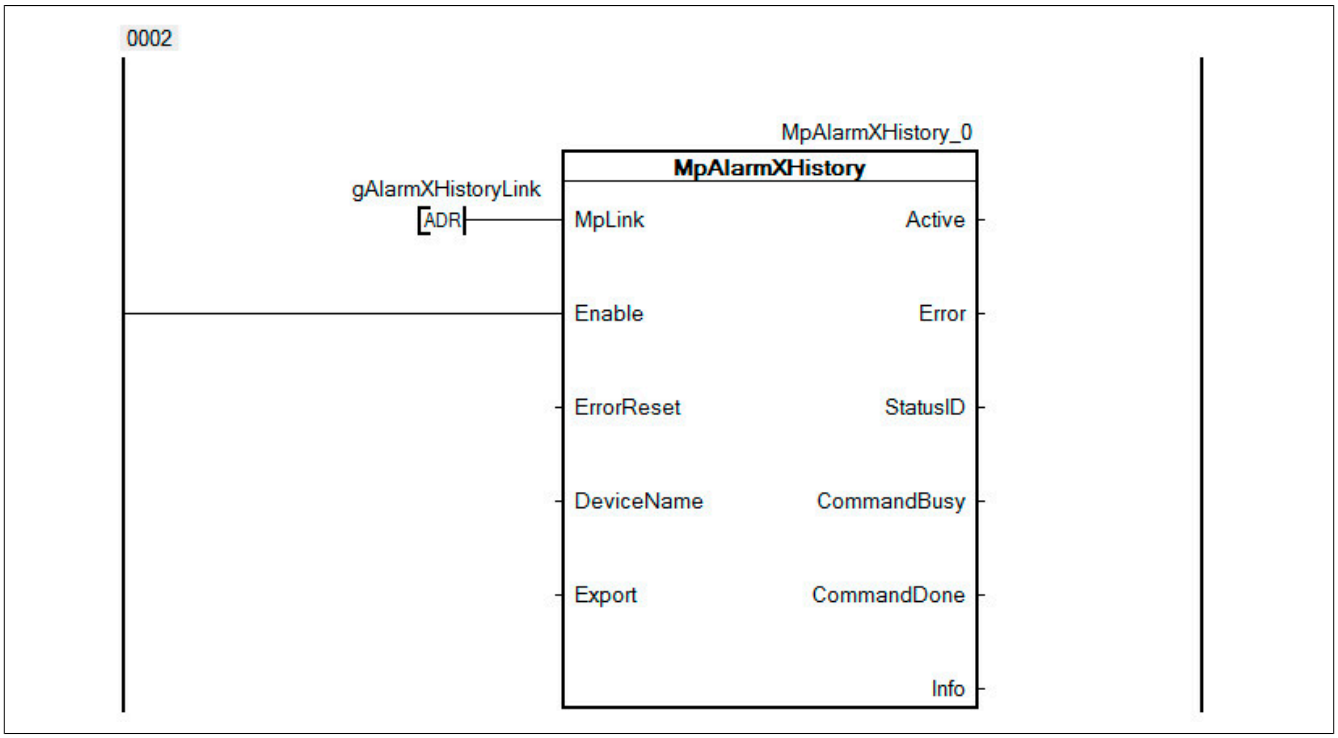
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gAlarmX-CoreLink erstellt haben.



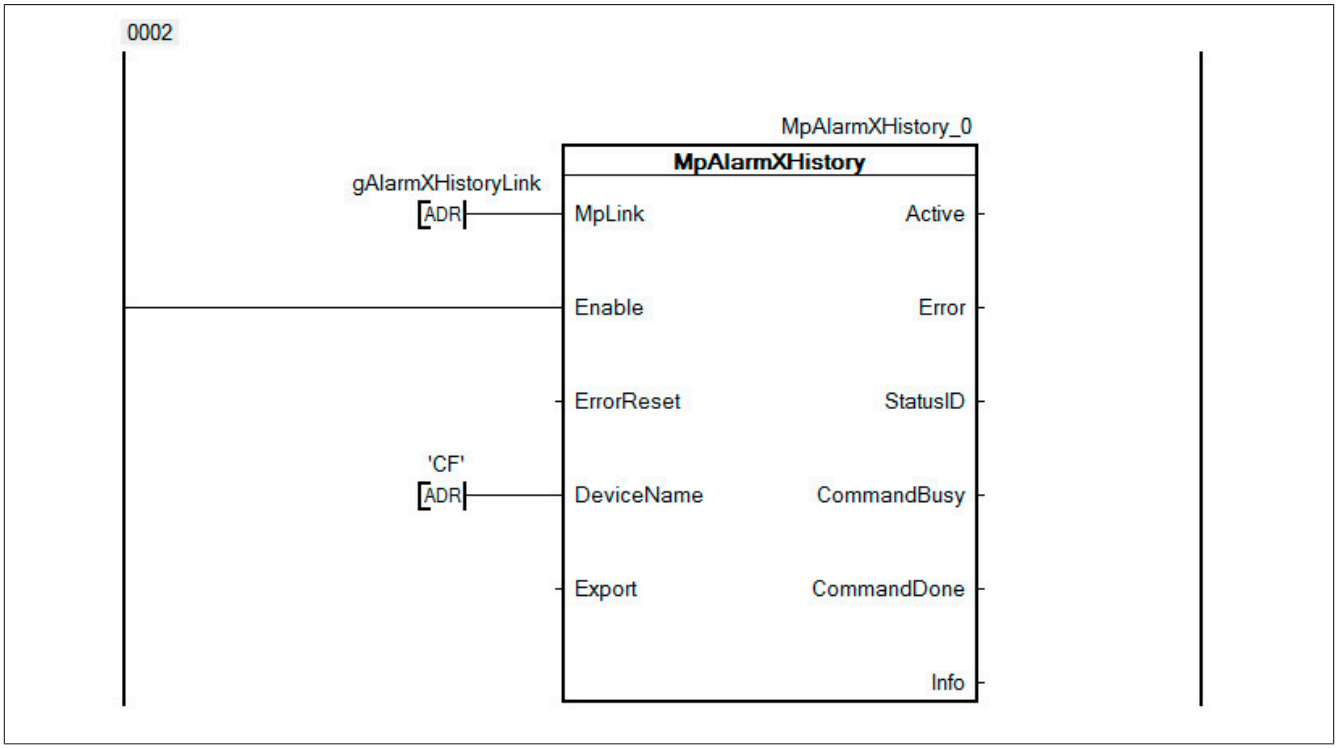
Aktivieren des Funktionsbausteins

Durch Setzen des "Enable" Eingangs wird der Funktionsbaustein aktiviert. Das wird über den Ausgang "Active" angezeigt.

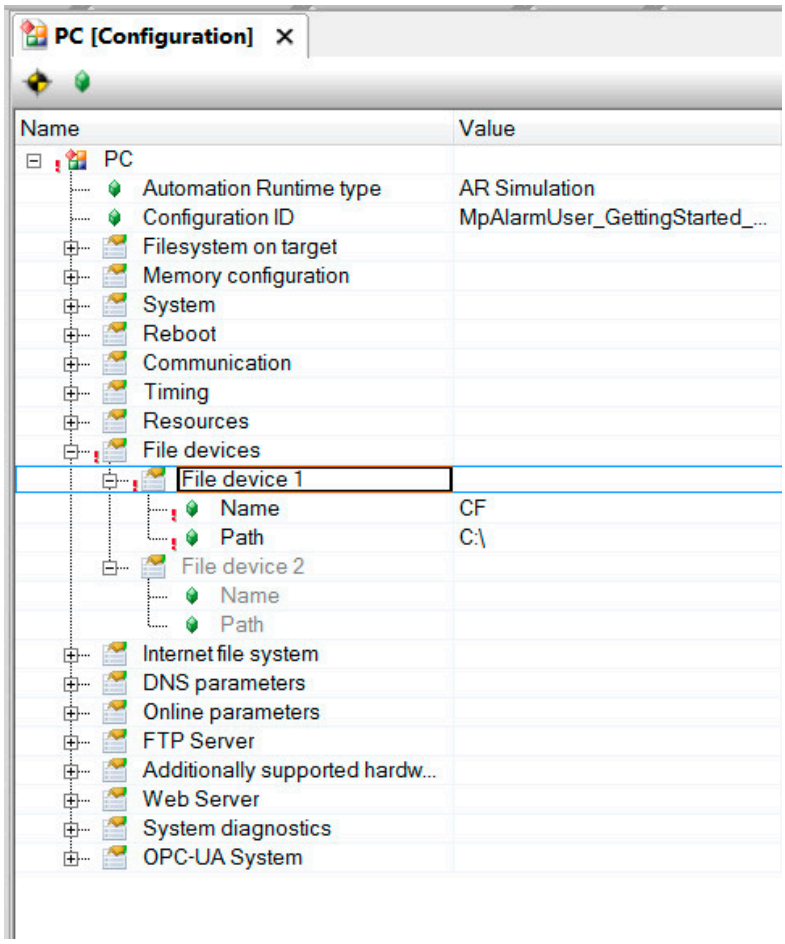


Speichermedium angeben

Mit MpAlarmXHistory kann die gesamte Alarmhistorie auf ein angegebenes Speichermedium exportiert werden. Aus diesem Grund wird beim Parameter "DeviceName" das Speichermedium "CF" angegeben.

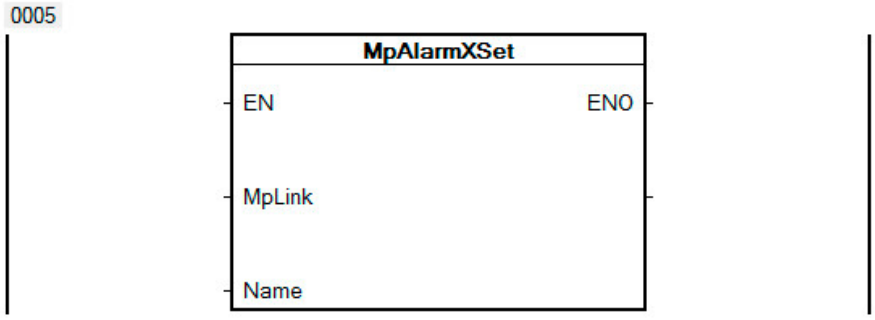


Das Speichermedium wird in der Hardwarekonfiguration unter "FileDevice" erstellt.



1.2.1.4.5 MpAlarmXSet einfügen

Das setzten der Benutzeralarme wird mit der Funktion MpAlarmXSet realisiert. Über Drag & Drop wird diese Funktion in das Programm gezogen. Damit die Funktion nicht zyklisch aufgerufen wird, fügen wir über einen Rechtsklick auf die Funktion "EN/ENO" hinzu.

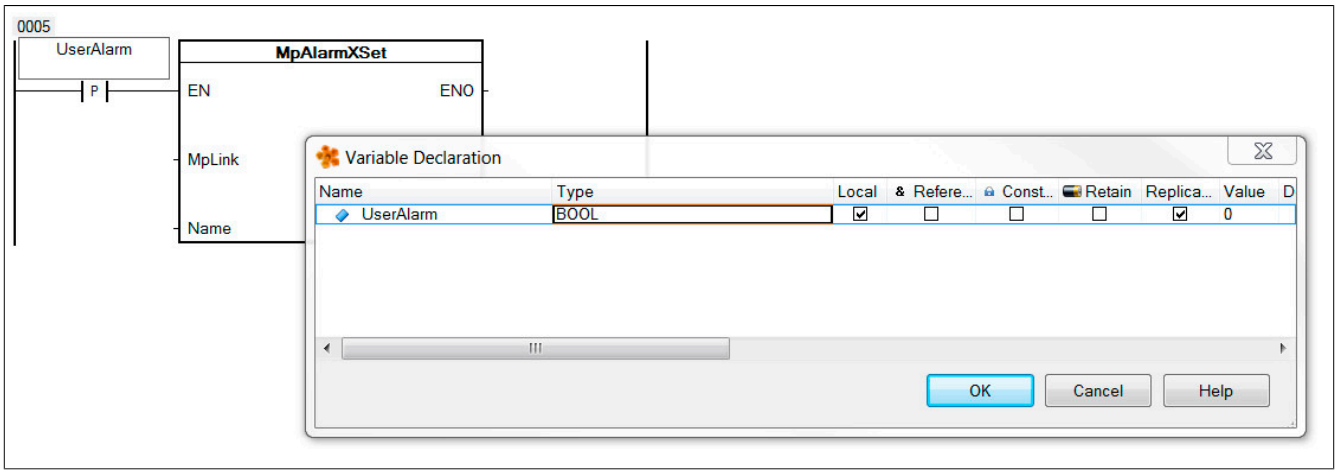


1.2.1.4.6 Parametrieren von MpAlarmXSet

Aktivieren der Funktion

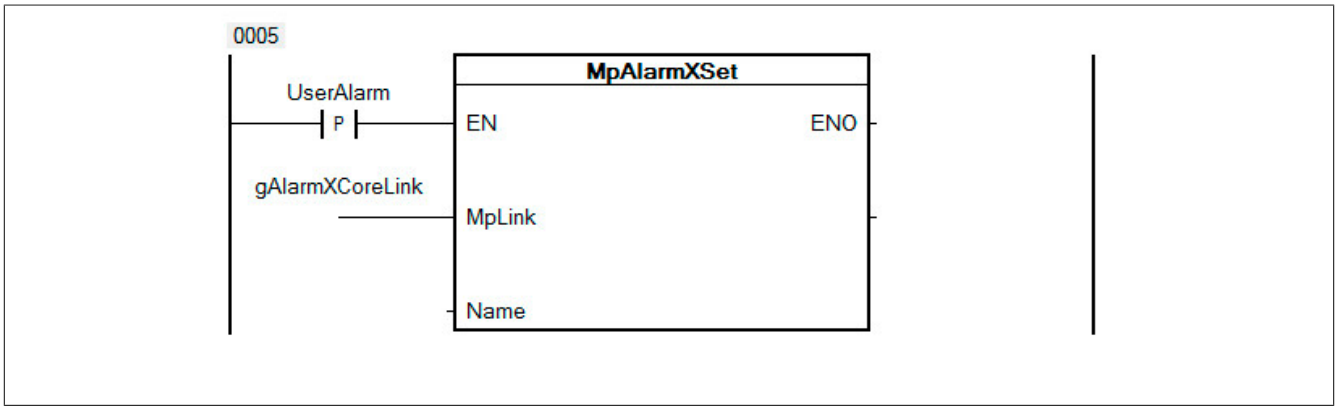
Damit die Funktion nur einmal aufgerufen wird und nicht zyklisch, wird ein Schalter verwendet der nur auf die positive Flanke von "UserAlarm" reagiert. Dieser wird vor dem Eingang "EN" positioniert.

Die Funktion bzw. der Alarm ist also nur dann aktiv, wenn die Variable "UserAlarm" zu *TRUE* wird.



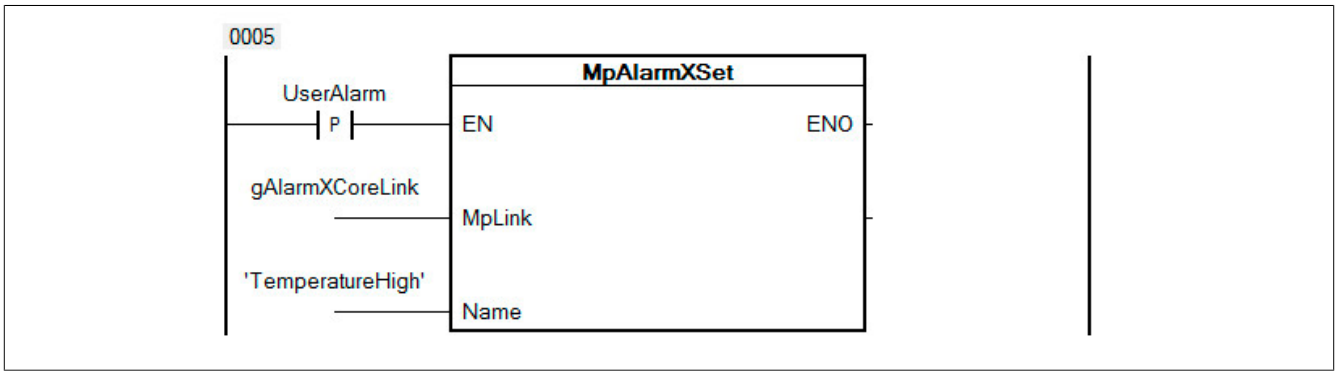
Verbinden des MpLinks

Danach wird der Eingang "MpLink" mit dem in der Configuration View bereits erstellten MpLink verbunden.



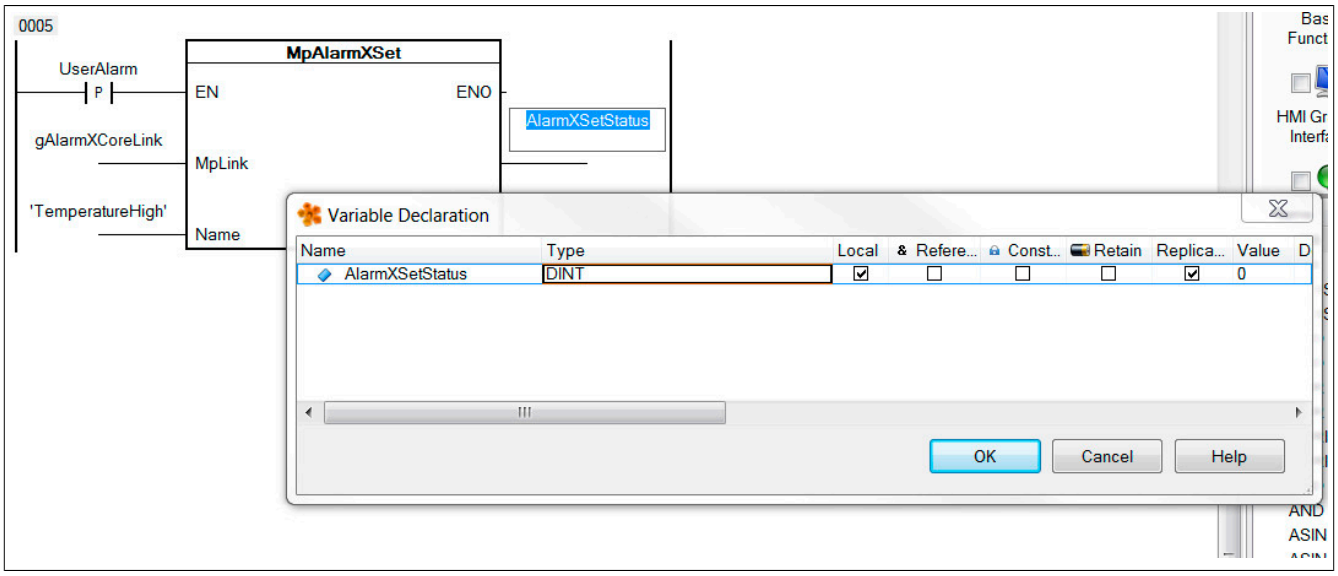
Alarmnamen angeben

Über den Eingang "Name" wird der Name des Benutzeralarms, welcher in der Konfiguration von MpAlarmXCore erstellt wurde, angegeben. Bei Aktivierung der Funktion wird der Alarm mit dem angegebenen Namen gesetzt.



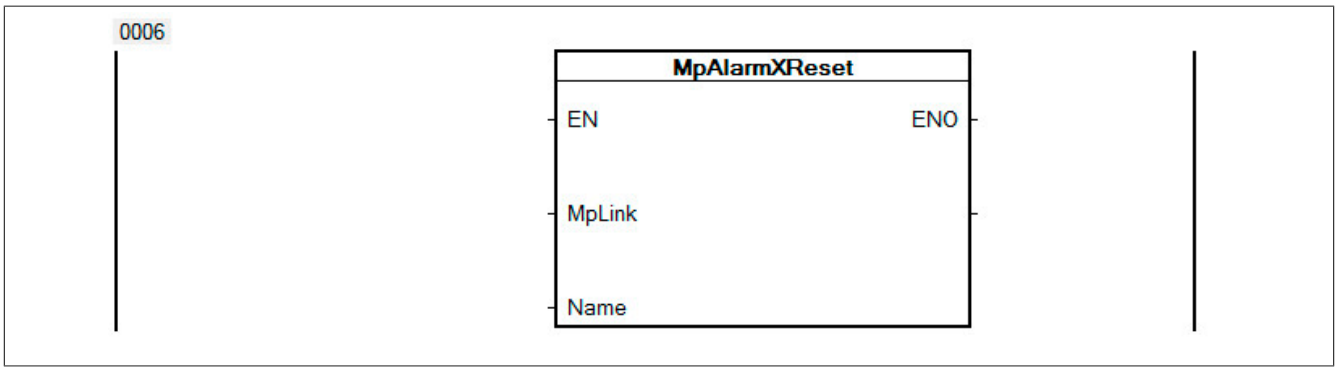
Status der Funktion MpAlarmXSet

Um den aktuellen Status der Funktion zu sehen wird die Variable "AlarmXSetStatus" erstellt.



1.2.1.4.7 MpAlarmXReset einfügen

Um Benutzeralarme wieder zurücksetzten zu können, wird MpAlarmXReset verwendet. Über Drag & Drop wird die Funktion in das Programm gezogen. Damit die Funktion nicht zyklisch aufgerufen wird, fügen wird über einen Rechtsklick auf die Funktion "EN/ENO" hinzu.

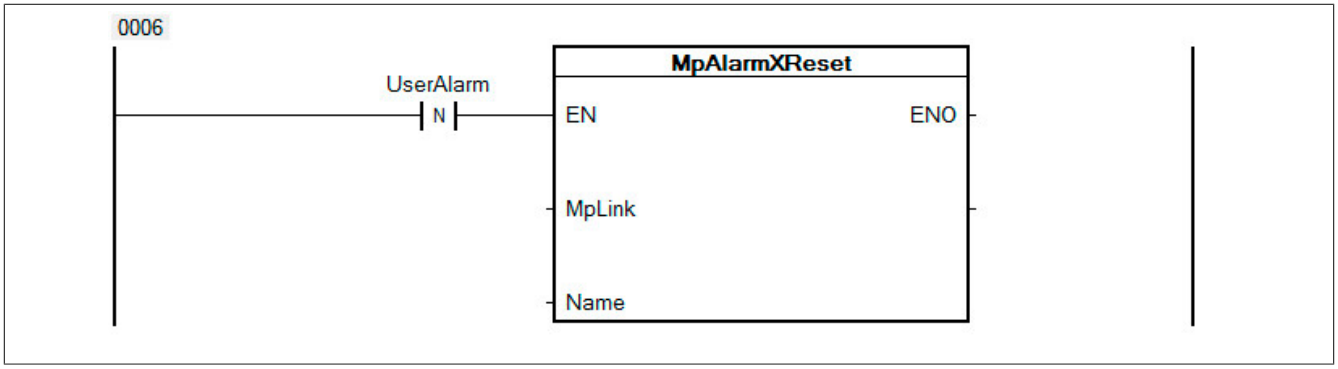


1.2.1.4.8 Parametrieren von MpAlarmXReset

Aktivieren der Funktion

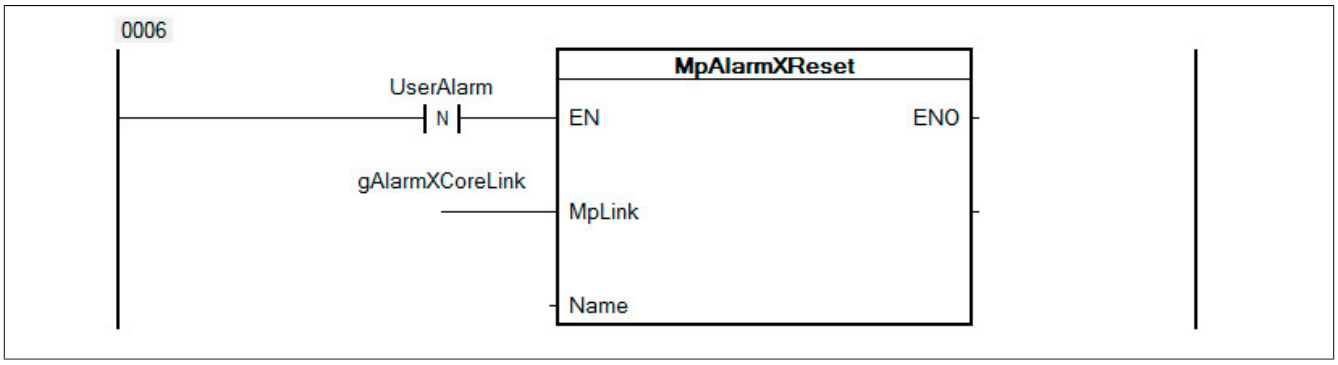
Damit die Funktion nur einmal aufgerufen wird und nicht zyklisch, wird ein Schalter verwendet der nur auf die negative Flanke von "UserAlarm" reagiert. Dieser wird vor dem Eingang "EN" positioniert.

Die Funktion ist also nur dann aktiv, wenn die Variable "UserAlarm" zu *FALSE* wird. Ist die Funktion aktiv wird der Alarm zurückgesetzt.



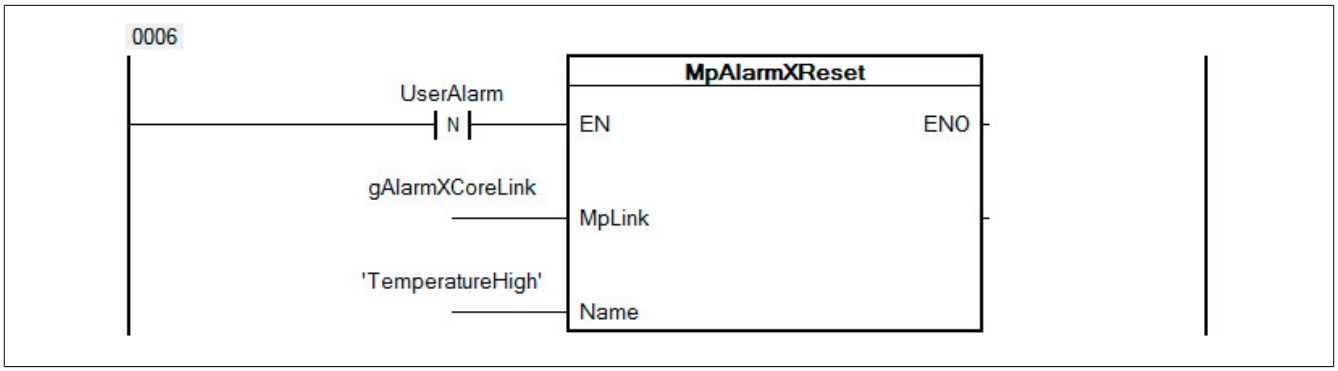
Verbinden des MpLinks

Der Eingang "MpLink" wird mit dem in der Configuration View bereits erstellten MpLink verbunden.



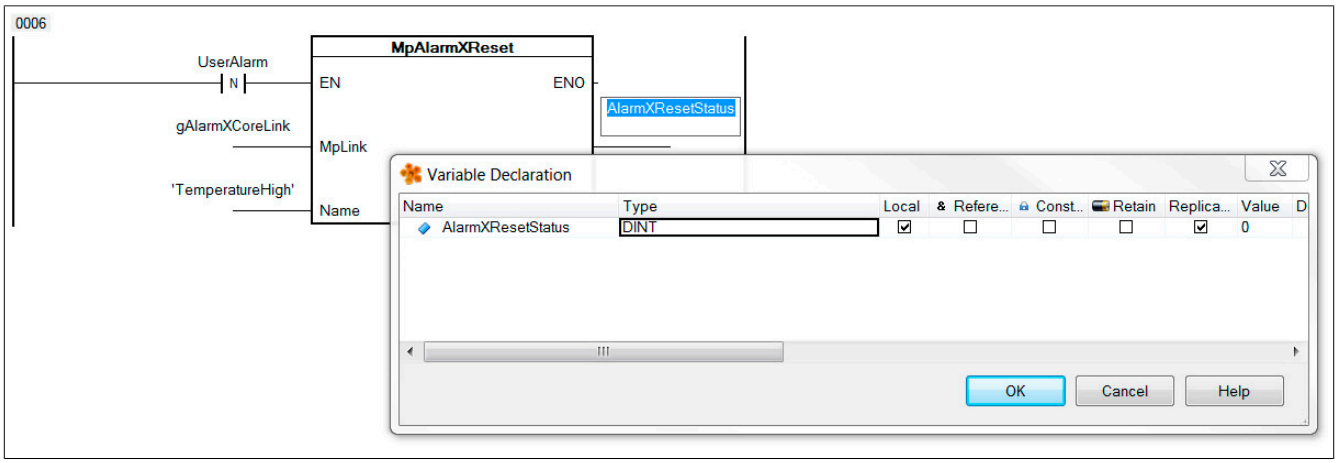
Alarmnamen angeben

Über den Eingang "Name" wird der Name des Benutzeralarms, welcher in der Konfiguration von MpAlarmXCore erstellt wurde, angegeben. Bei Aktivierung der Funktion, wird der Alarm mit den angegebenen Namen zurückgesetzt.



MpAlarmXReset Status

Um den aktuellen Status der Funktion zu sehen, wird die Variable AlarmXResetStatus erstellt.



1.2.1.5 Dateistruktur erzeugen

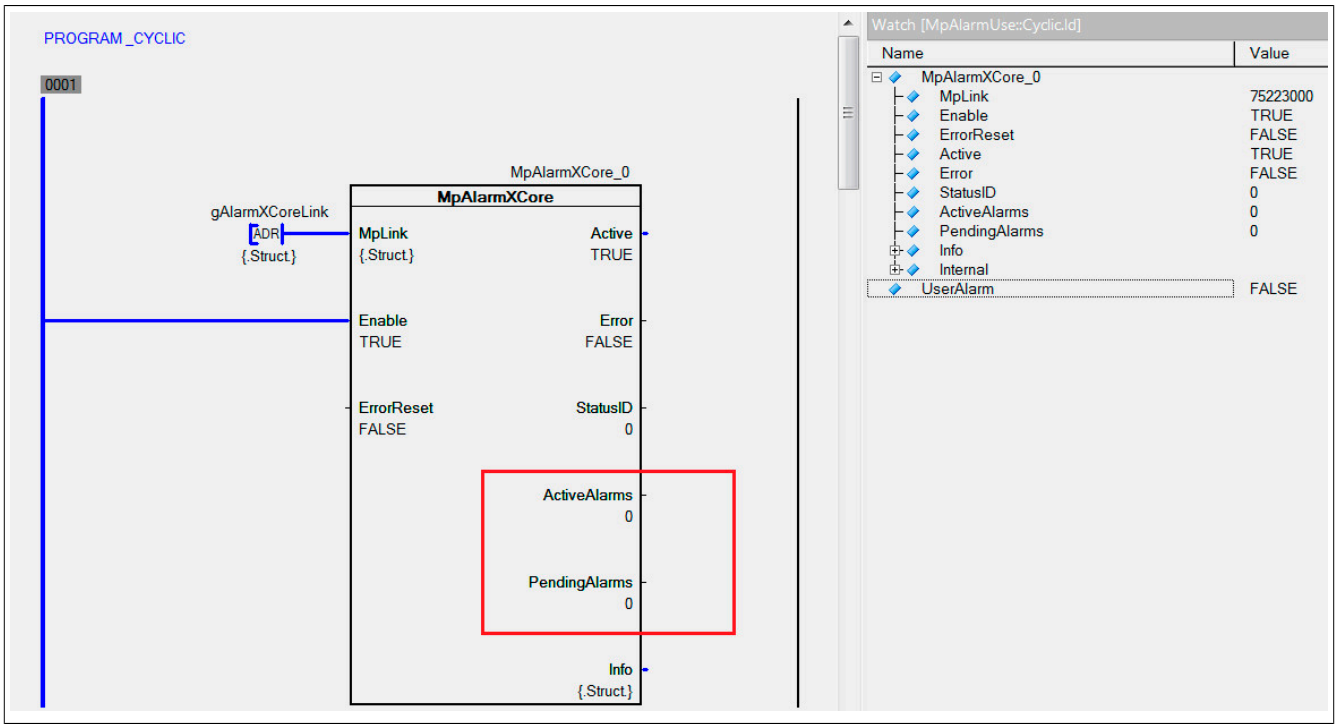
Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

1.2.1.6 Testen des Programms

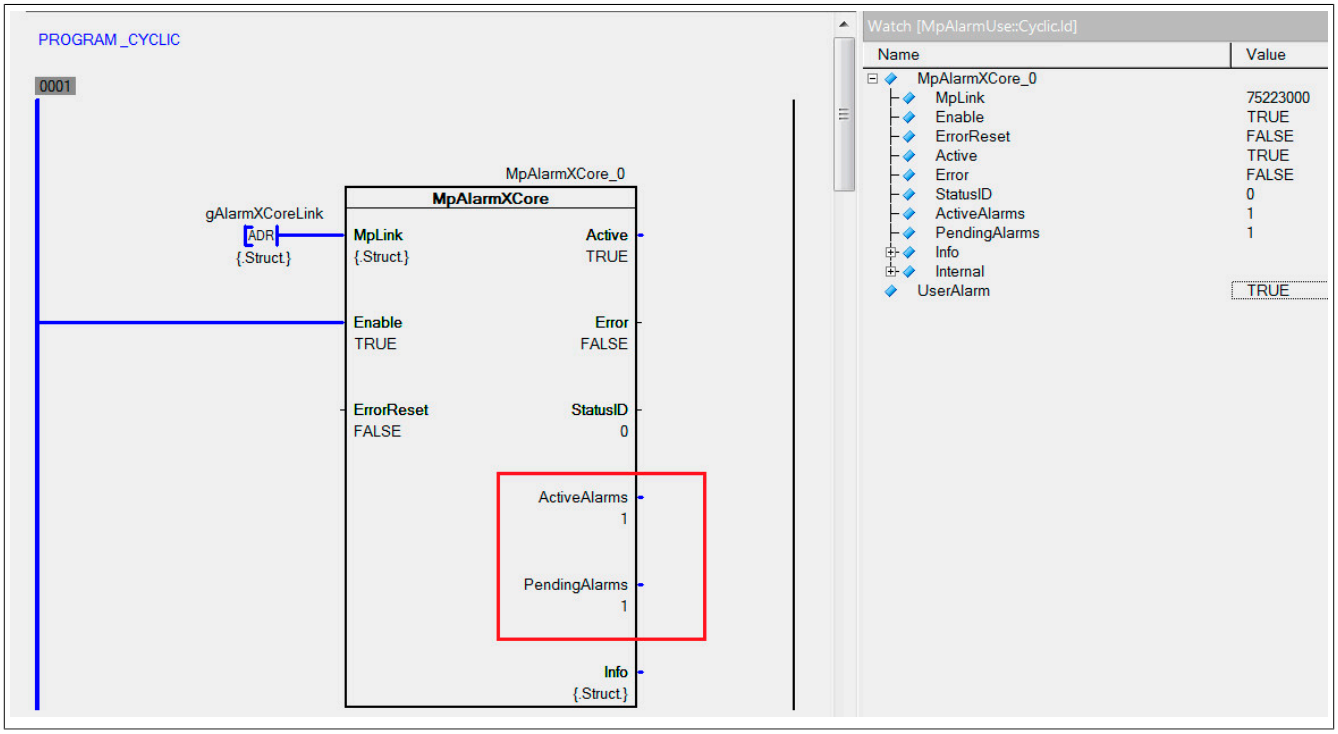
Aktivieren des Benutzeralarms

Nach einem Download der Änderungen kann das Programm getestet werden. Der Monitor-Modus wird aktiviert.

Zu Beginn ist die Variable "UserAlarm" noch auf *FALSE*. Die Ausgänge "ActiveAlarms" und "PendingAlarms" sind beide noch null.



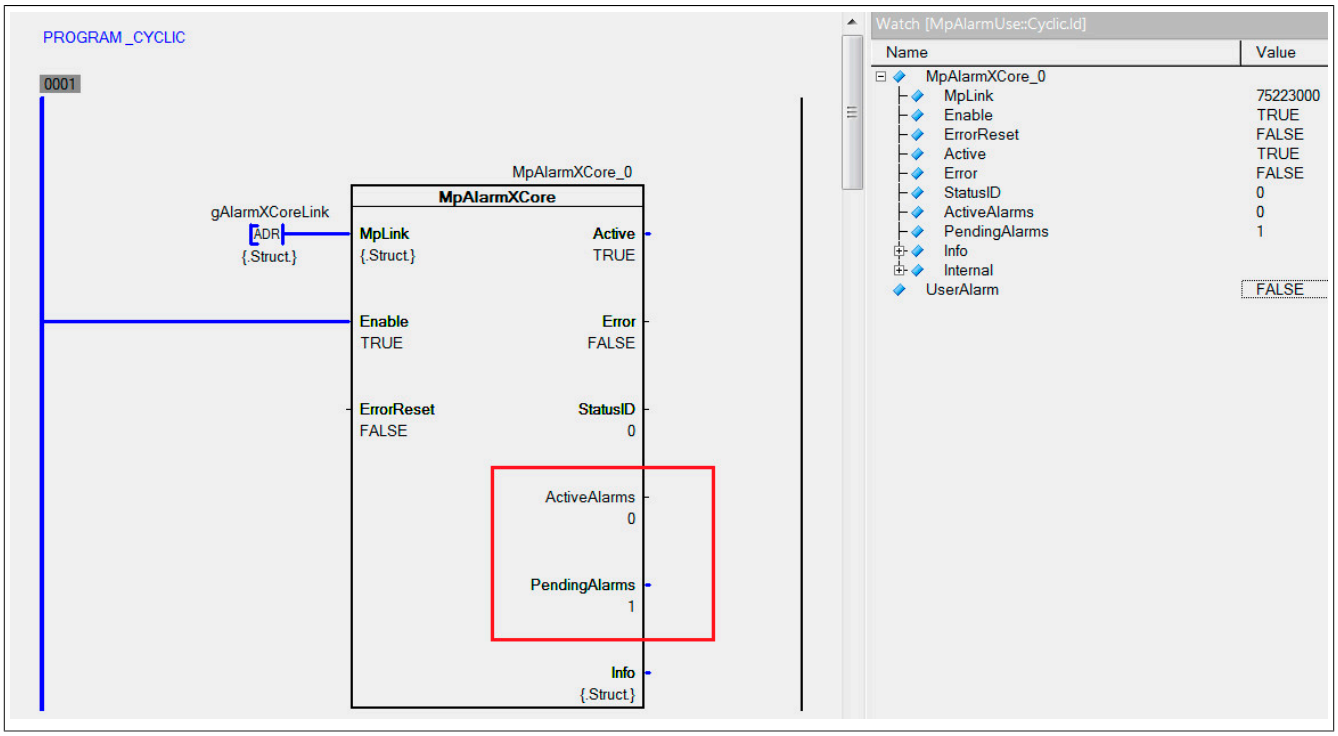
Um nun einen Alarm anzeigen lassen zu können, muss die Variable "UserAlarms" auf *TRUE* gesetzt werden. Dadurch wird die Funktion MpAlarmXSet aktiviert und der Benutzeralarm ist gesetzt. Die beiden Ausgänge von MpAlarmXCore signalisieren, dass ein Alarm aktiv ist.



Zurücksetzen des Benutzeralarms

Um den Alarm wieder zu deaktivieren, wird die Variable "UserAlarm" wieder auf FALSE gesetzt. Die Funktion MpAlarmXReset wird somit aktiviert und der Alarm wird zurückgesetzt.

Da der Alarm in der Visualisierung noch nicht quittiert wurde, zeigt der Ausgang "PendingAlarm" von MpAlarmX-Core noch immer 1 an.



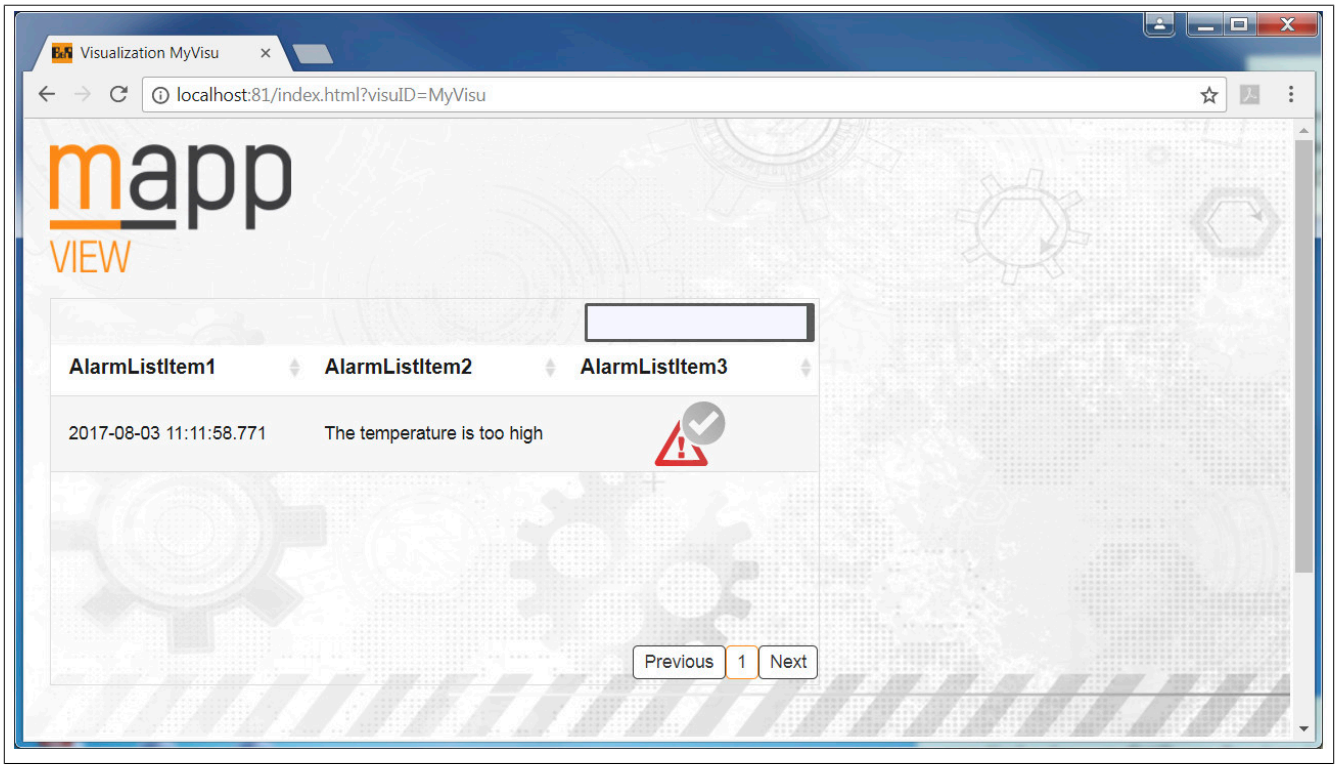
Um Benutzeralarme in einer VC4-Visualisierung darzustellen, siehe Benutzeralarme mit VC4 mapp Vorlage.

Um Benutzeralarme in einer mapp View Visualisierung darzustellen, siehe Benutzeralarme in mapp View Visualisieren.

1.2.2 Benutzeralarme mit mapp View Visualisierung

In diesem Abschnitt wird erklärt wie Benutzeralarme in einer mapp View Visualisierung dargestellt werden können. Wie ein Alarmsystem erstellt werden kann, ist im Getting Started Benutzeralarme erstellen erklärt.

Damit Alarme in einer mapp View Visualisierung dargestellt werden können, wird das AlarmList-Widget verwendet.



1.2.2.1 mapp View Visualisierung einfügen

Damit die Alarme visualisiert werden können, wird eine mapp View Visualisierung im Projekt eingefügt.

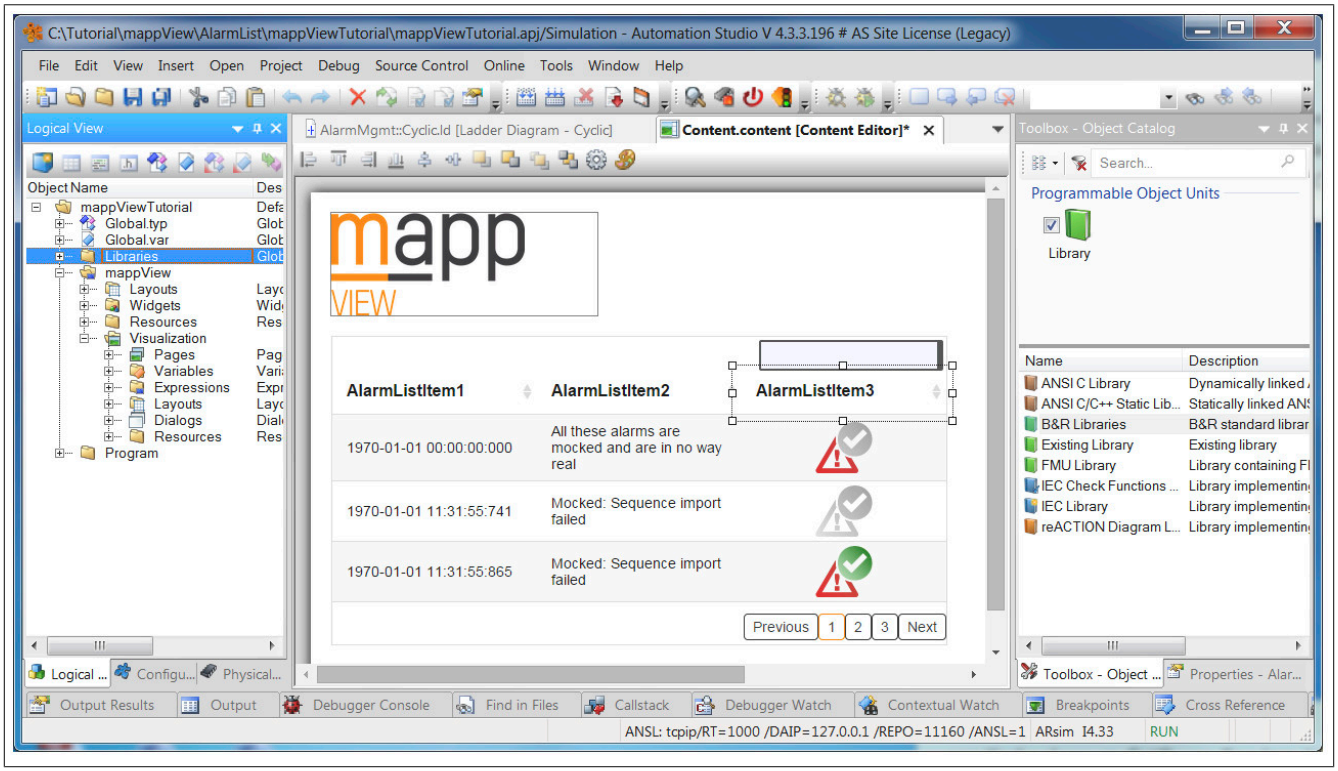
Wie eine mapp View Visualisierung erstellt wird, ist im Getting Started hier erklärt.

1.2.2.2 Widgets einfügen und parametrieren

Die Alarme werden mithilfe des AlarmList-, und AlarmListItem-Widget visualisiert.

Wie diese Widgets konfiguriert werden müssen, wird hier erklärt.

Es werden 3 Spalten definiert. Eine für den Zeitstempel, eine für die Alarmnachricht und eine für den aktuellen Alarmzustand.



Einfügen der MpServer Bibliothek

Zuletzt muss die Bibliothek MpServer eingefügt werden. Somit sind alle notwendigen Schritte, um die Alarme über die Visualisierung darzustellen, abgeschlossen. Für mehr Information zur MpServer Bibliothek siehe hier.

1.2.2.3 Dateistruktur erzeugen

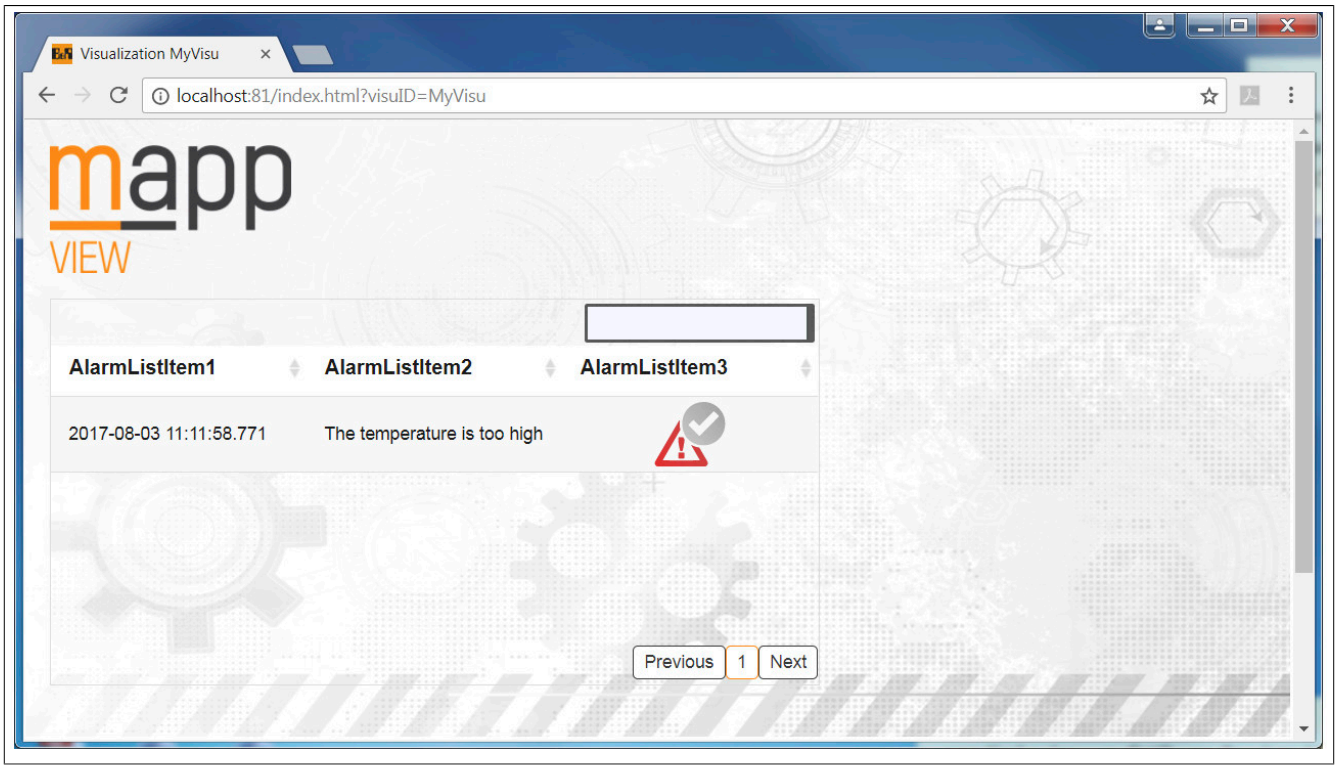
Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

1.2.2.4 Visualisierung testen

Browser öffnen und aktivieren eines Benutzeralarms

Die Visualisierung wird mit dem Google Chrome Browser getestet.

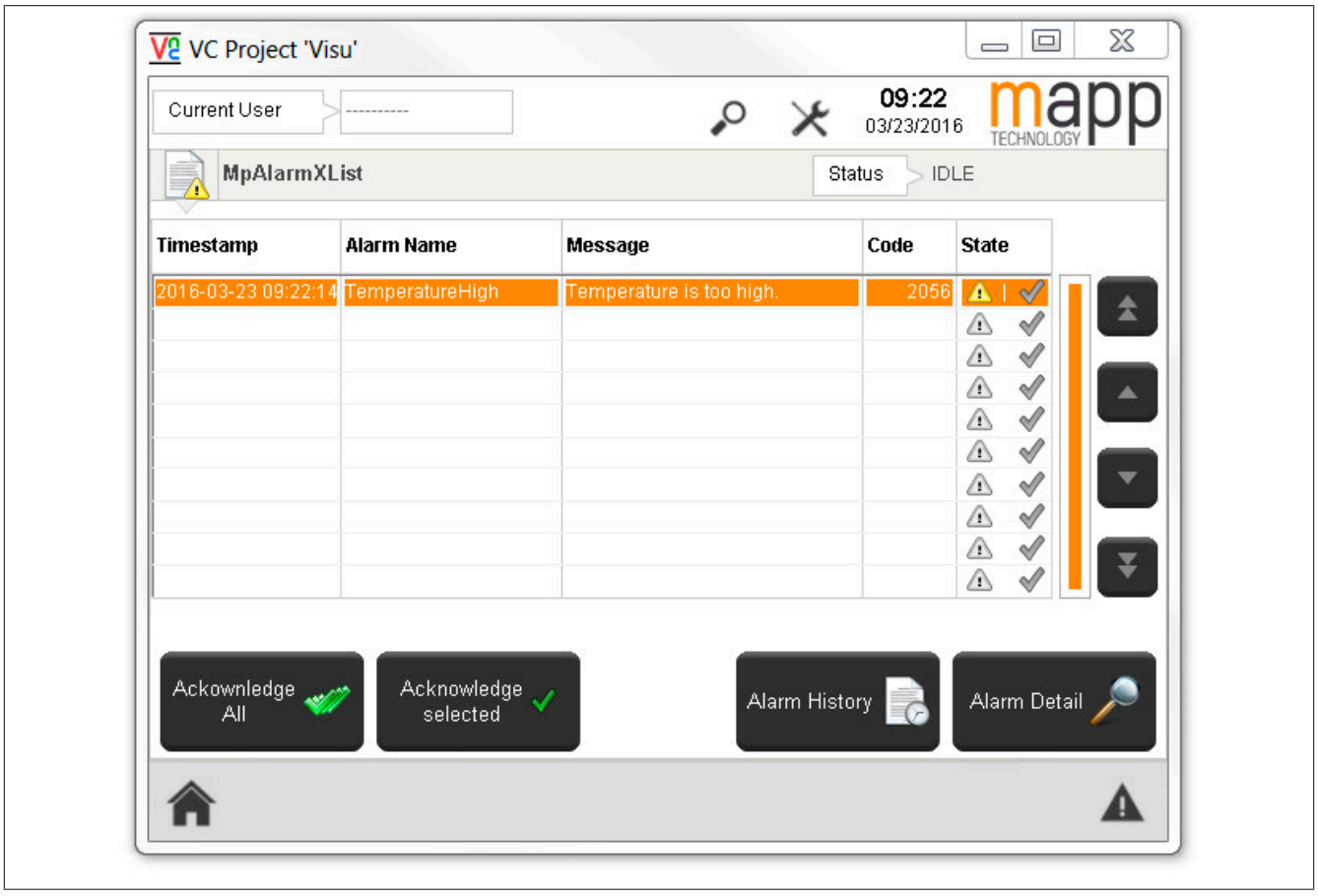
In der mapp View Visualisierung wird nun ein Alarm angezeigt, wenn wie im Abschnitt Testen des Programms, die Variable "UserAlarm" auf `TRUE` gesetzt wird.



1.2.3 Benutzeralarme mit VC4 mapp Vorlage

In diesem Abschnitt wird Schritt für Schritt beschrieben wie man mithilfe von MpAlarmXListUI und MpAlarmXHistoryUI Benutzeralarme und die Alarmhistorie auf einer VC4 Visualisierung anzeigen lassen kann.

Wie ein Alarmsystem erstellt werden kann, siehe Benutzeralarme erstellen.



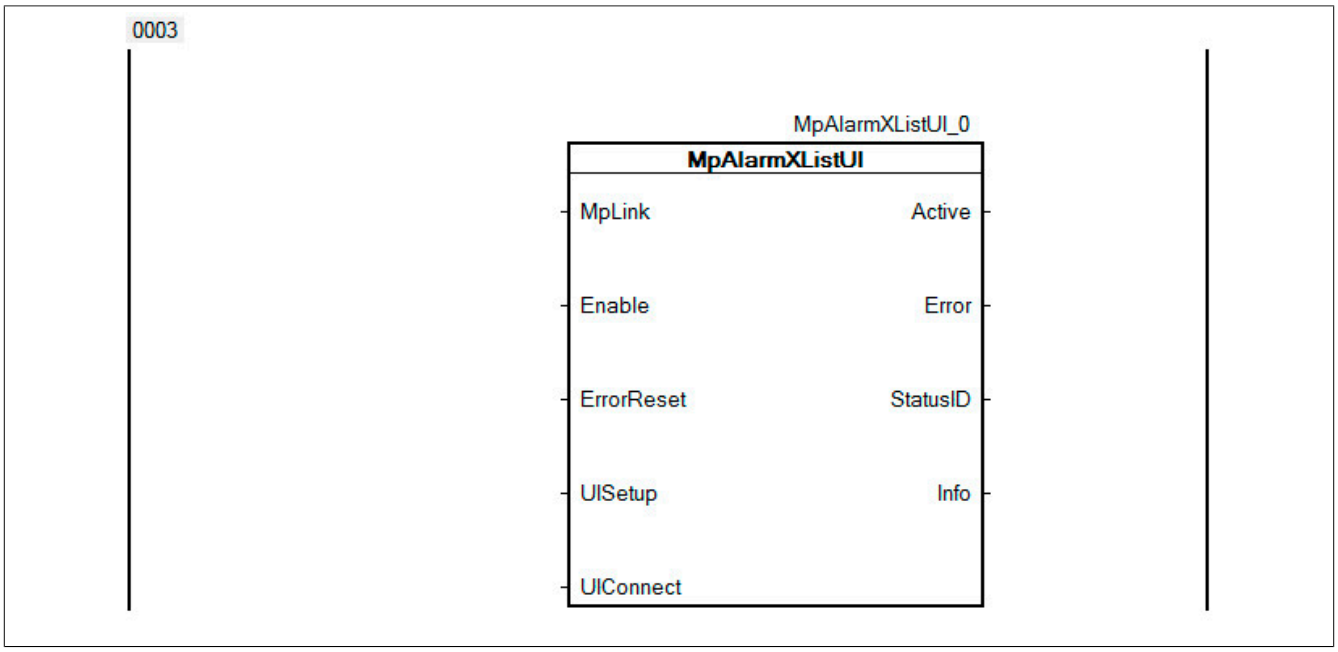
1.2.3.1 Funktionsbausteine einfügen und parametrieren

In diesem Abschnitt wird erklärt, welche Funktionsbausteine eingefügt und parametriert werden. Die Funktionsbausteine können in ein bereits bestehendes Programm oder in ein neues Programm eingefügt werden.

1.2.3.1.1 MpAlarmXListUI im Programm einfügen

Funktionsbaustein einfügen

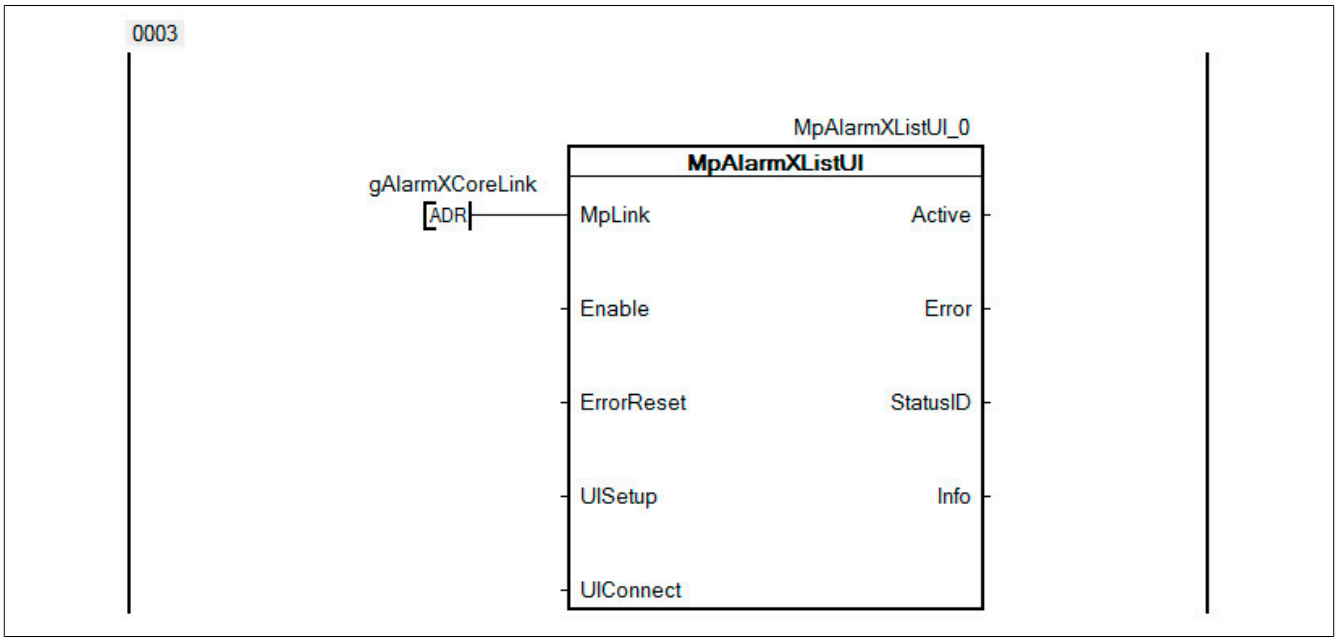
Es wird der Funktionsbaustein MpAlarmXListUI eingefügt.



1.2.3.1.2 Parametrieren von MpAlarmXListUI

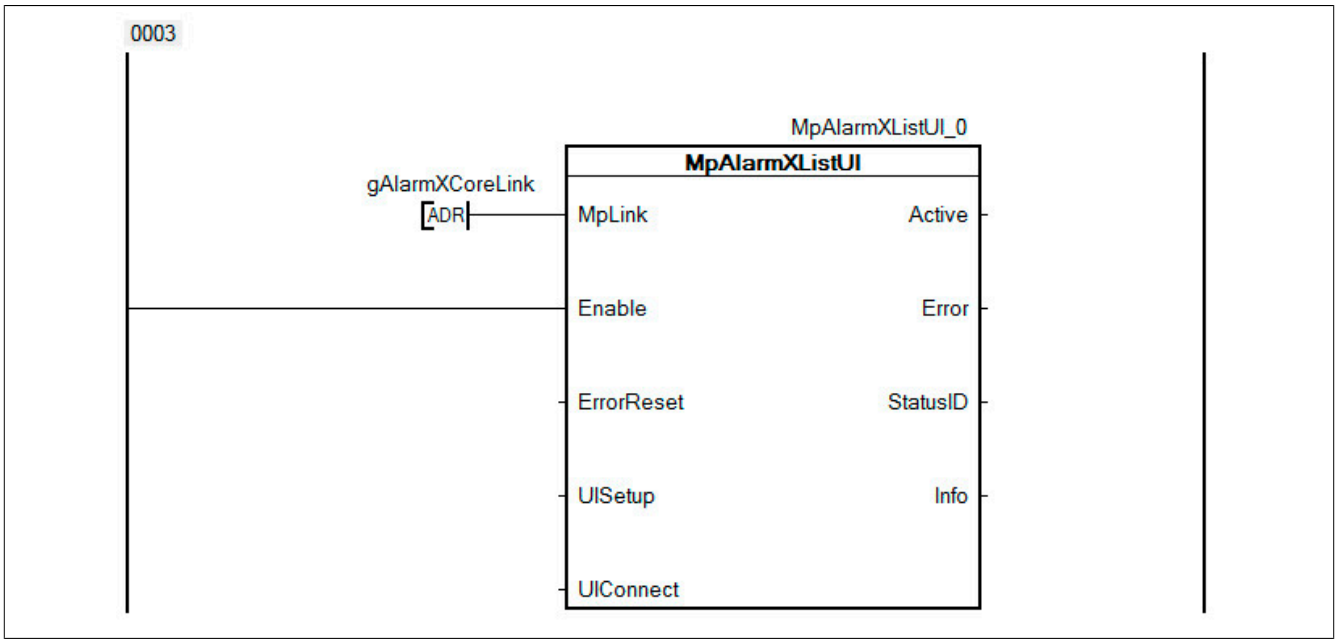
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gAlarmX-CoreLink erstellt haben.



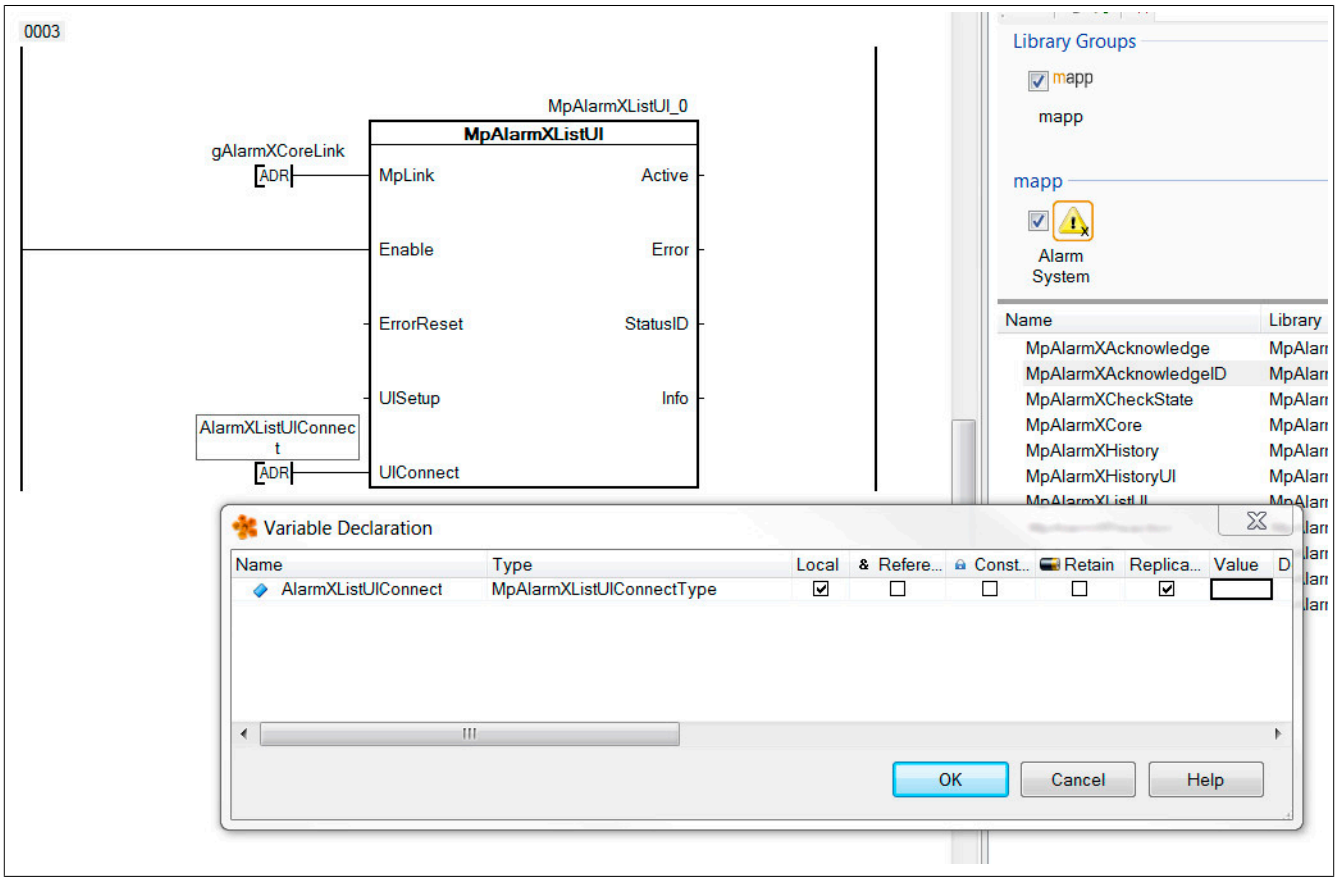
Aktivieren des Funktionsbausteins

Der Funktionsbaustein wird durch den Eingang "Enable" aktiviert.



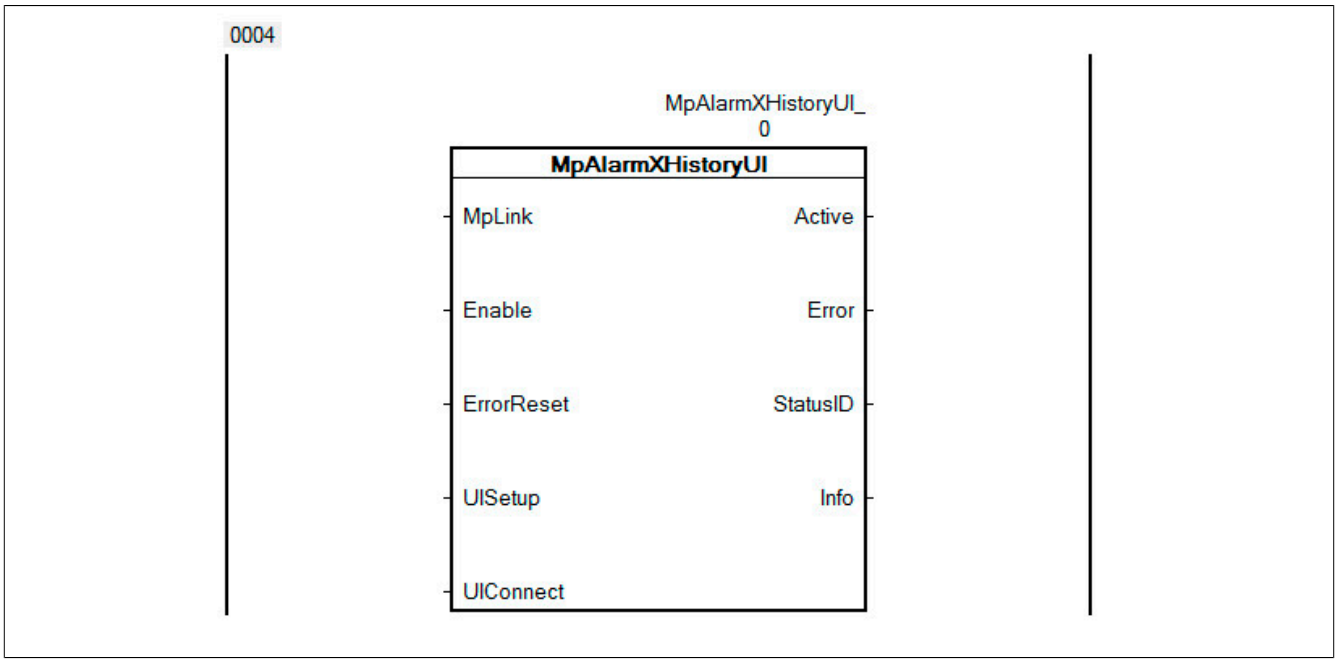
UIConnect bestimmen

Um später das Programm und die Visualisierung zu verbinden, muss eine Variable vom Typ *MpAlarmXListUIConnectType* erstellt werden.



1.2.3.1.3 MpAlarmXHistoryUI im Programm einfügen

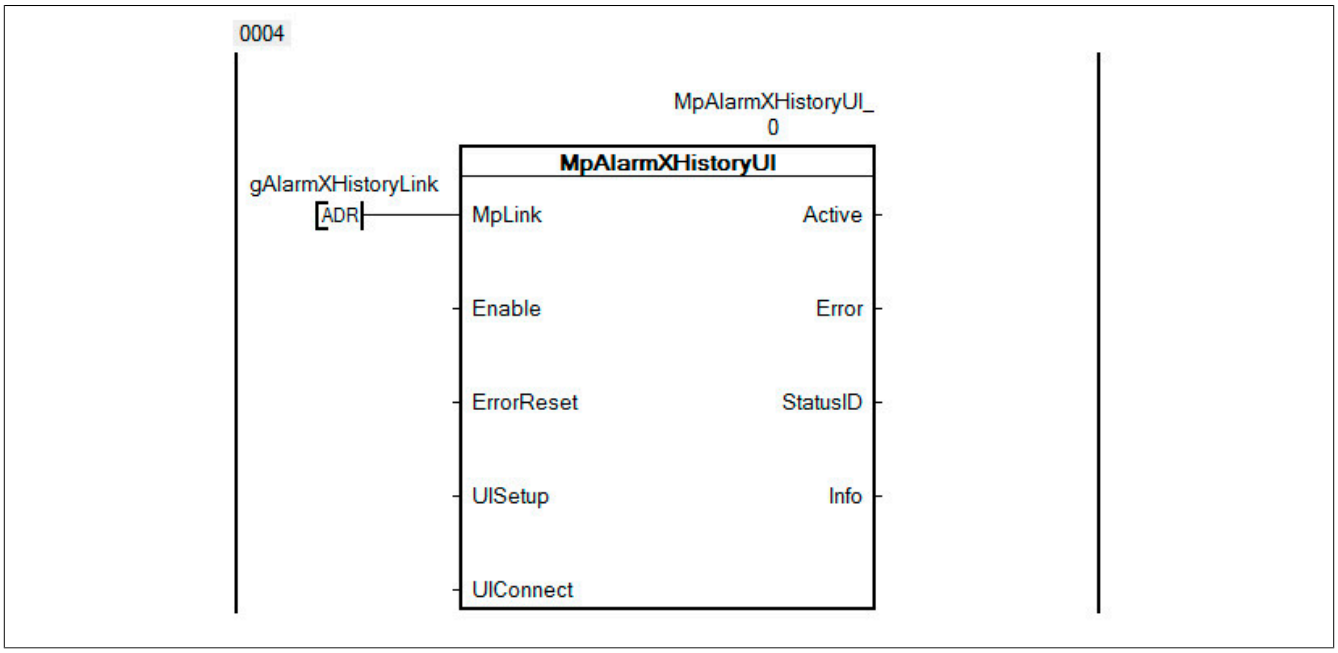
Es wird der Funktionsbaustein MpAlarmXHistoryUI eingefügt.



1.2.3.1.4 Parametrieren von MpAlarmXHistoryUI

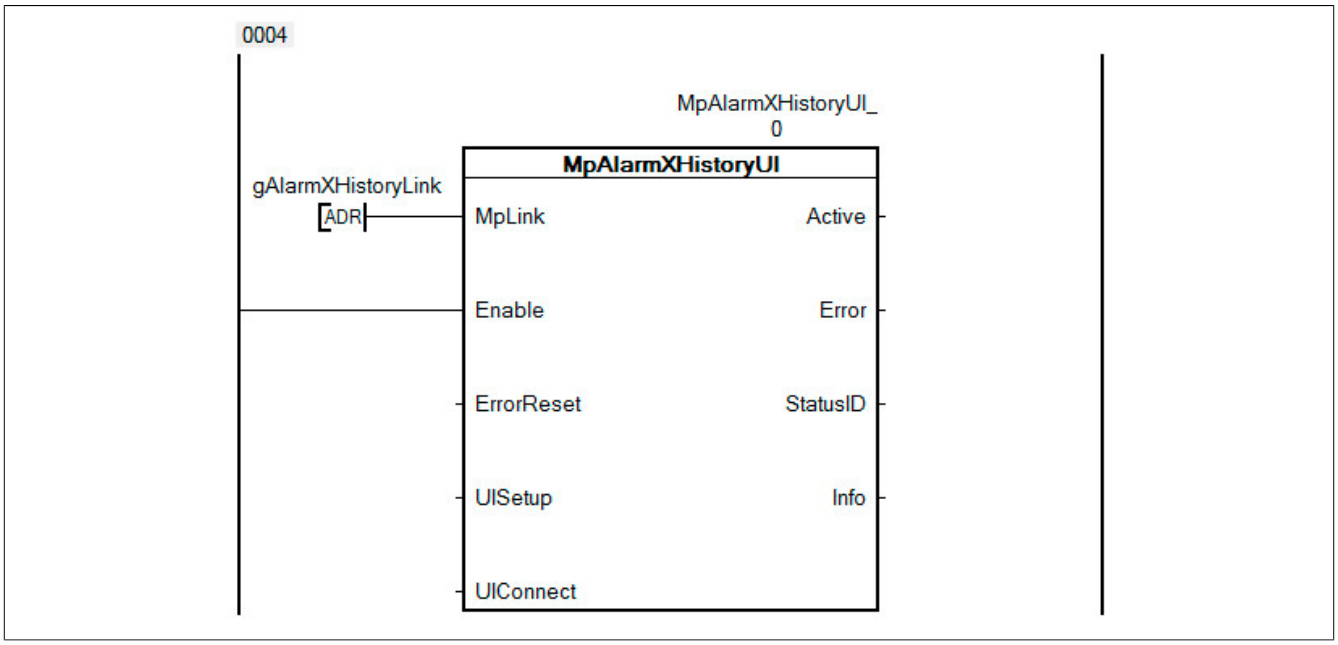
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gAlarmX-HistoryLink erstellt haben.



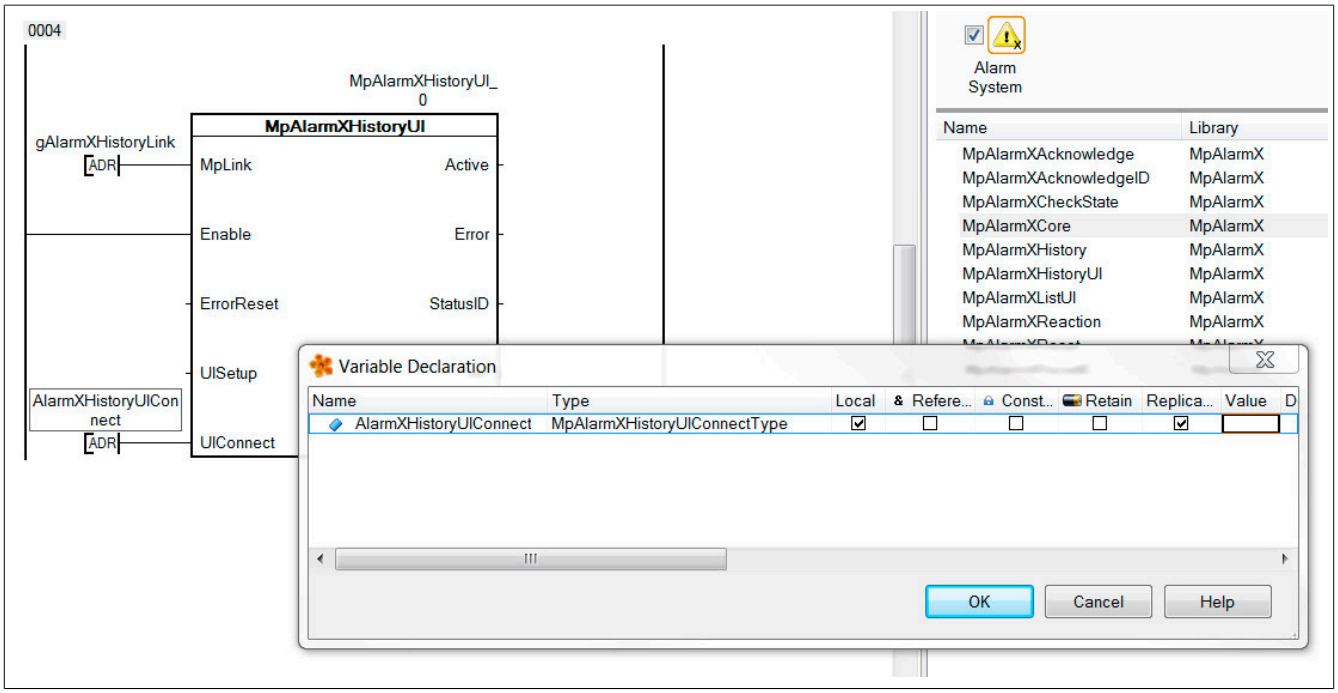
Aktivieren des Funktionsbausteins

Der Funktionsbaustein wird durch den Eingang "Enable" aktiviert.



UIConnect bestimmen

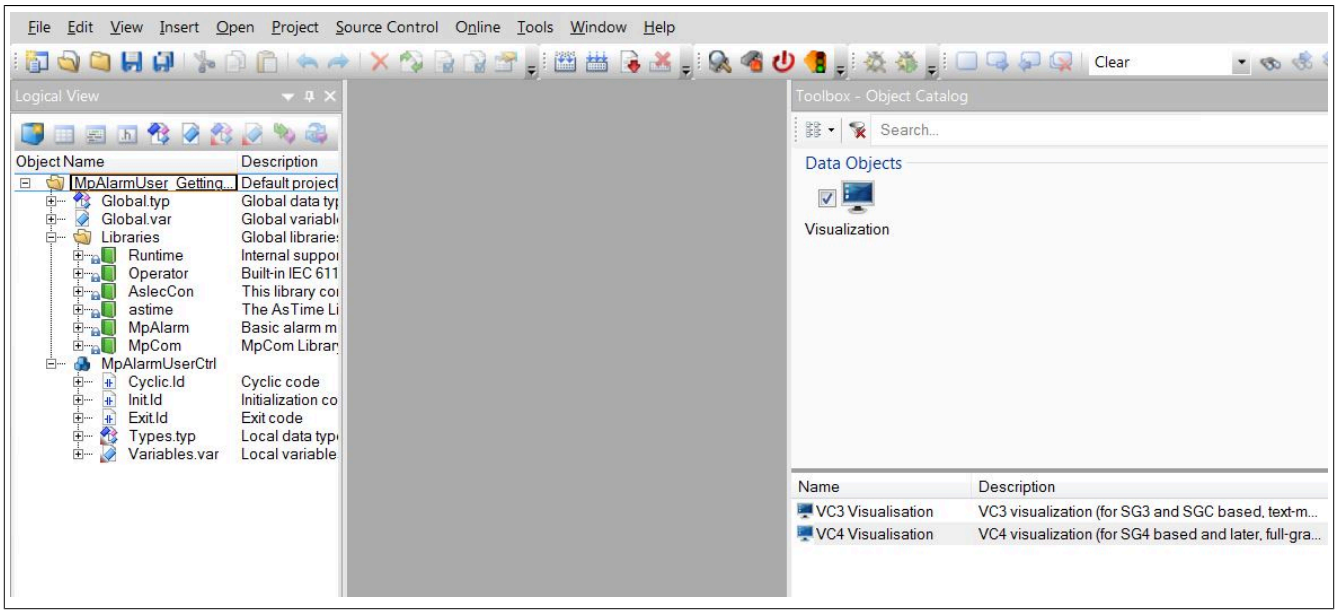
Um später das Programm und die Visualisierung zu verbinden, muss eine Variable vom Typ *MpAlarmXHistoryUI-ConnectType* erstellt werden.



1.2.3.2 Visualisierung erstellen

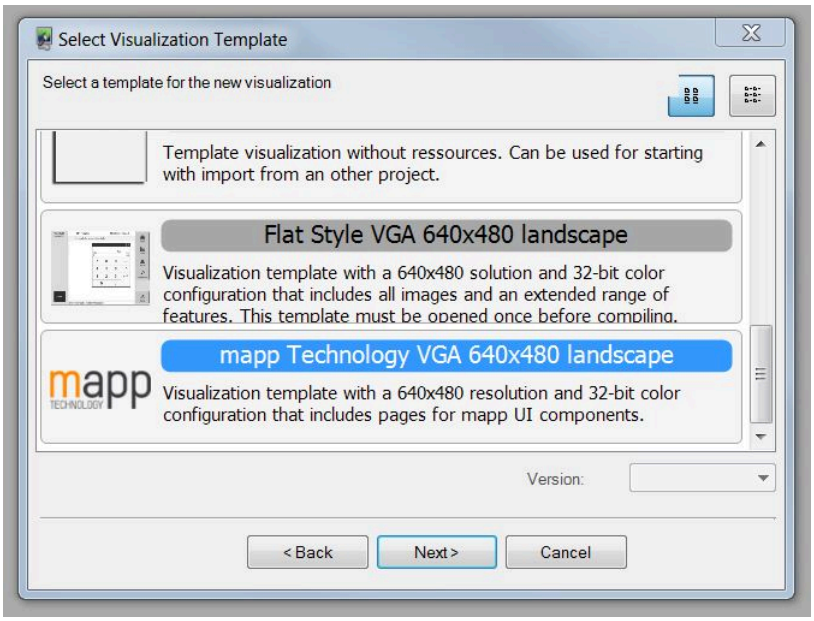
Visualisierung einfügen

Um den Benutzeralarm zu testen wird eine VC4 Visualisierung erstellt.



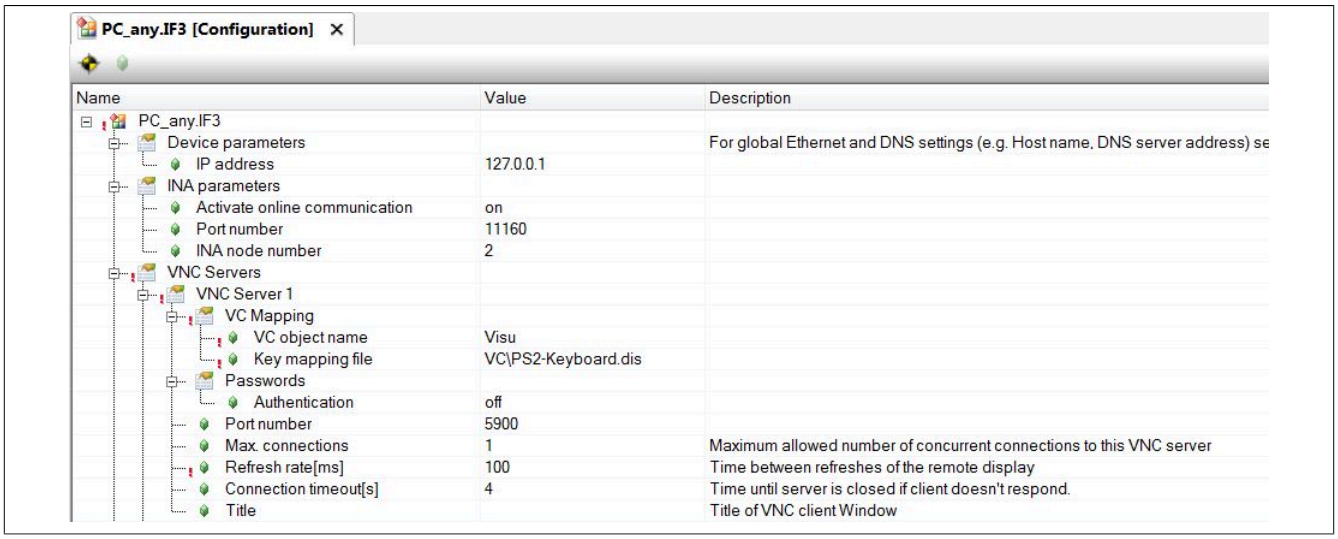
Vorlage auswählen

Damit die mapp Visualisierung angezeigt werden kann, muss die Vorlage "mapp Technology VGA 640x480 landscape" ausgewählt werden.



VNC Server für die Visualisierung konfigurieren

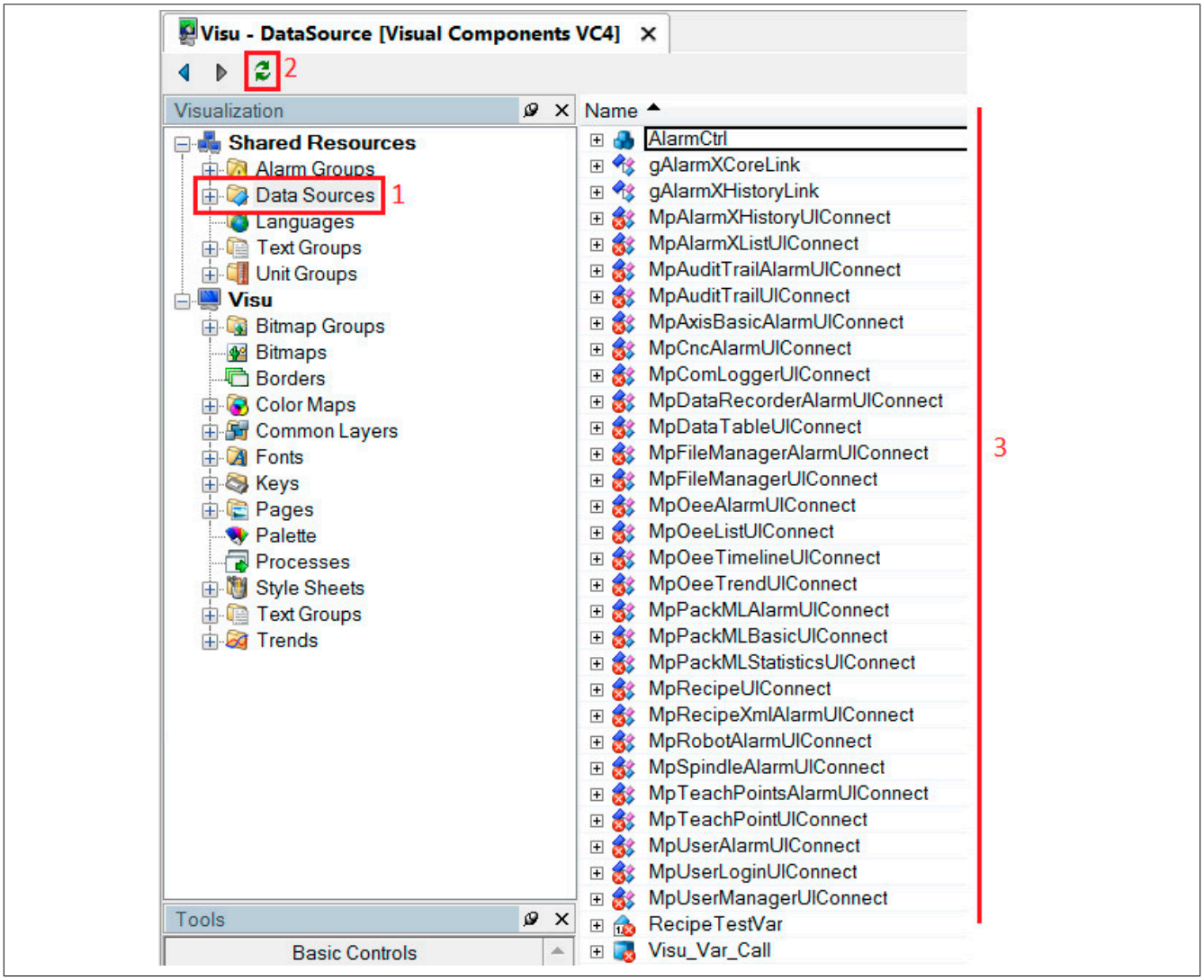
Danach wird die Visualisierung mit einem VNC Server verbunden.



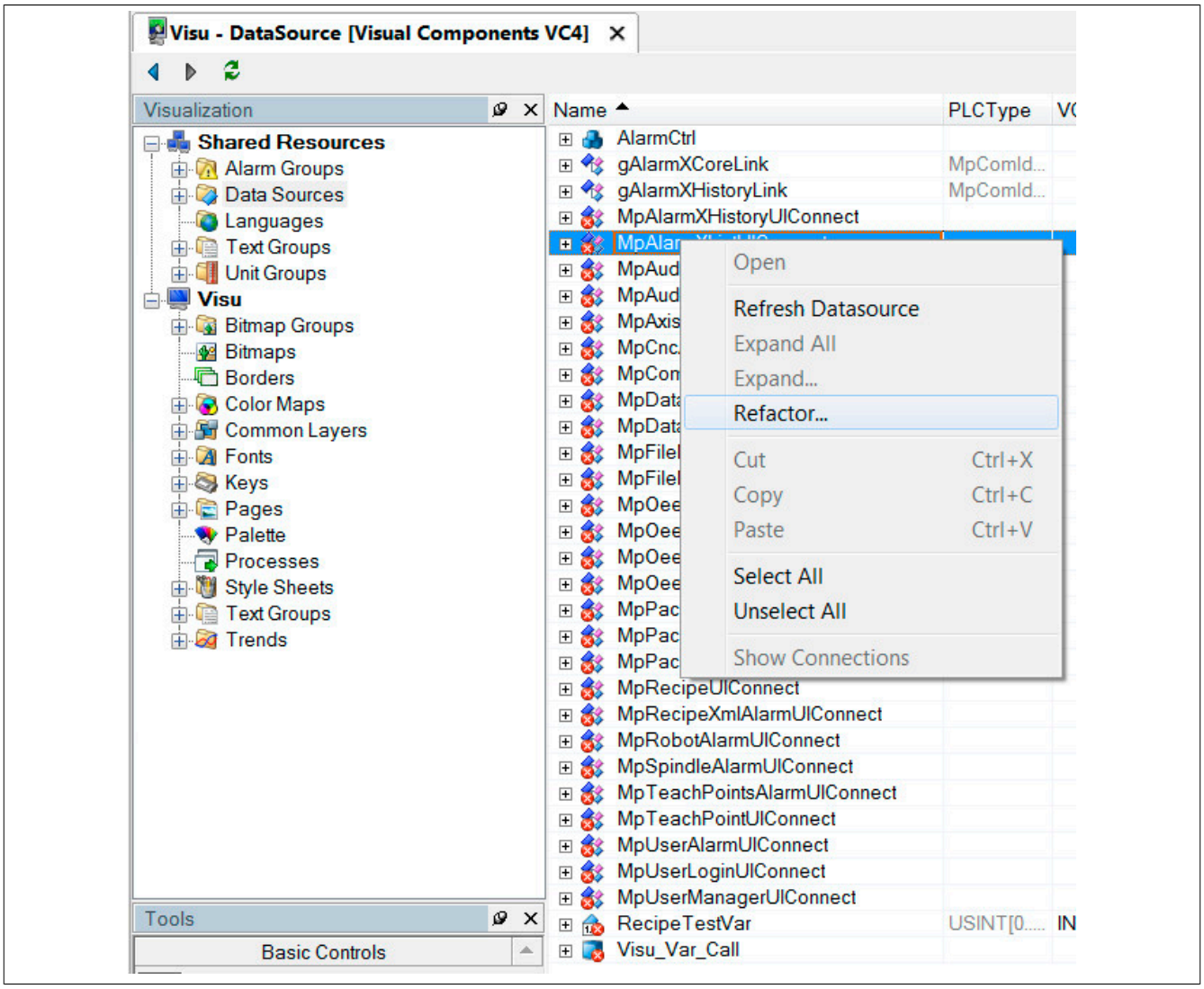
1.2.3.3 Anbinden der Vorlage an das eigene Projekt

Erstellen der Verbindung zwischen Task und Visualisierung

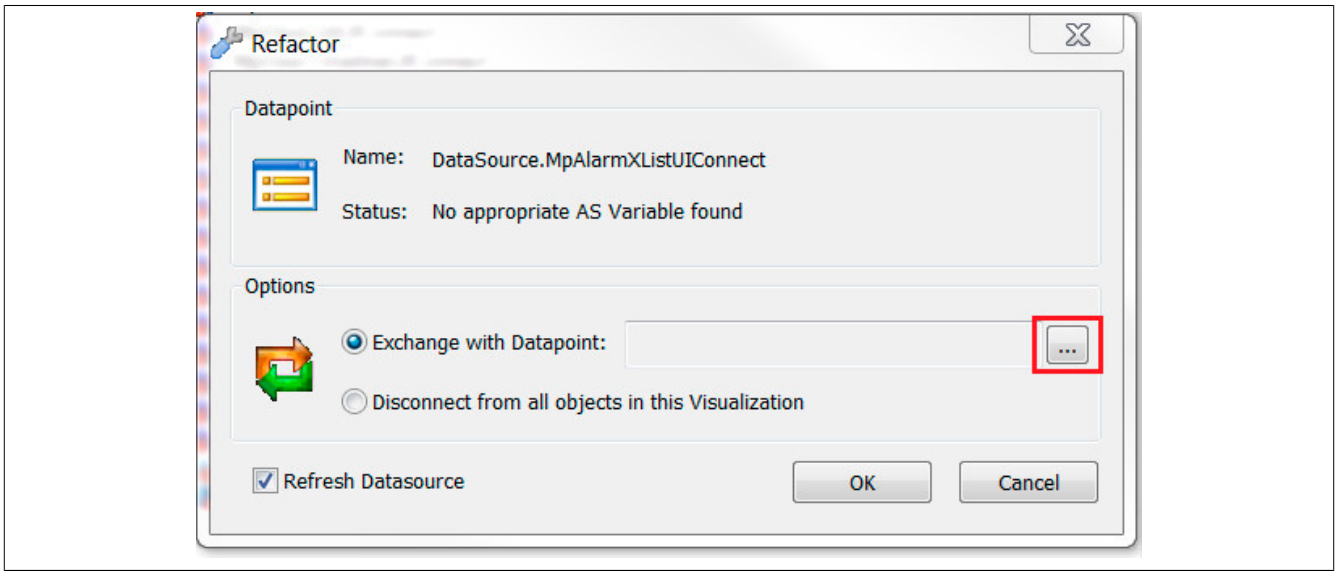
Nach dem Öffnen der Visualisierung wird zunächst die Datenquelle ausgewählt[1] und aktualisiert[2]. Nach dem aktualisieren wird sichtbar welche Variablen in der Visualisierung verbunden aber im Projekt nicht vorhanden sind[3].

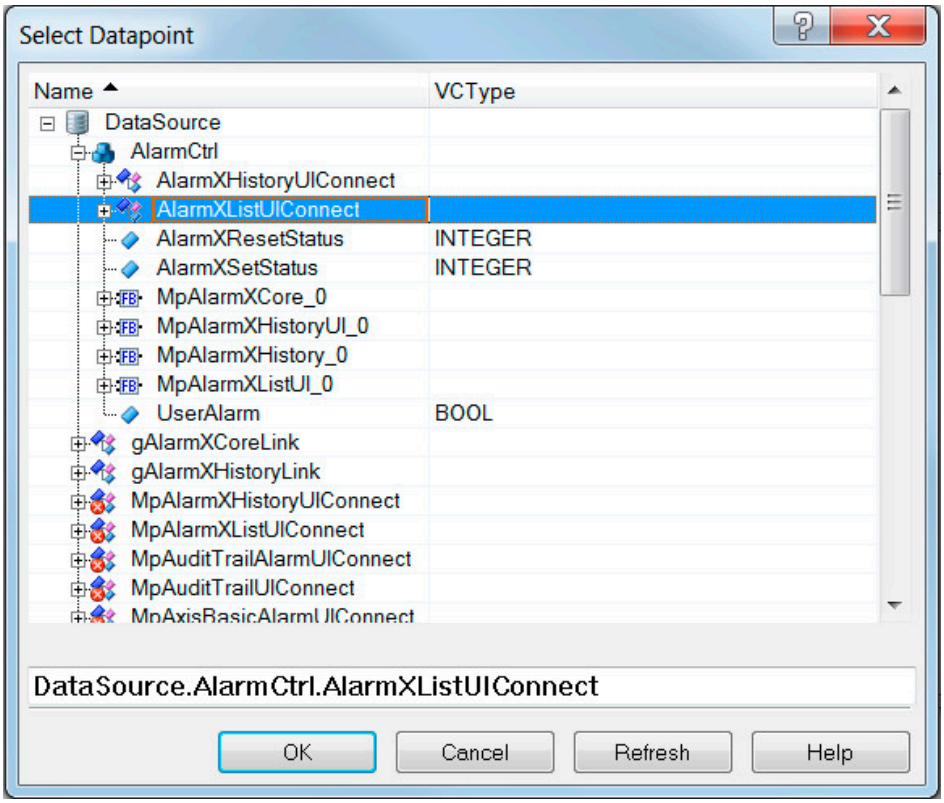


Aus dieser Liste wird die Variable ausgewählt welche mit der eigenen ersetzt werden soll. Mit einem Rechtsklick auf *MpAlarmXListUIConnect* wählt man den Eintrag *Refactor* aus.

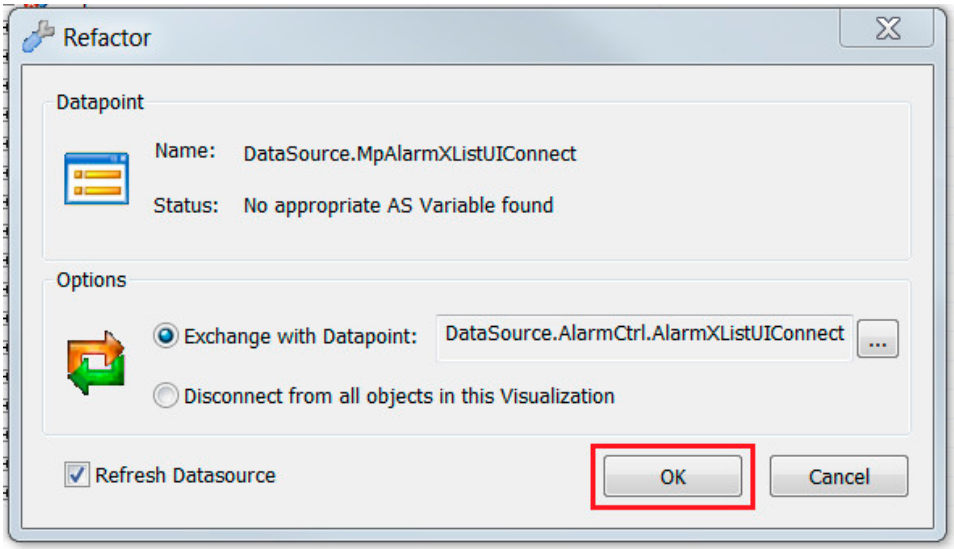


Jetzt wird über das Dialogfenster, welches sich öffnet, die eigene "UIConnect" Struktur ausgewählt.

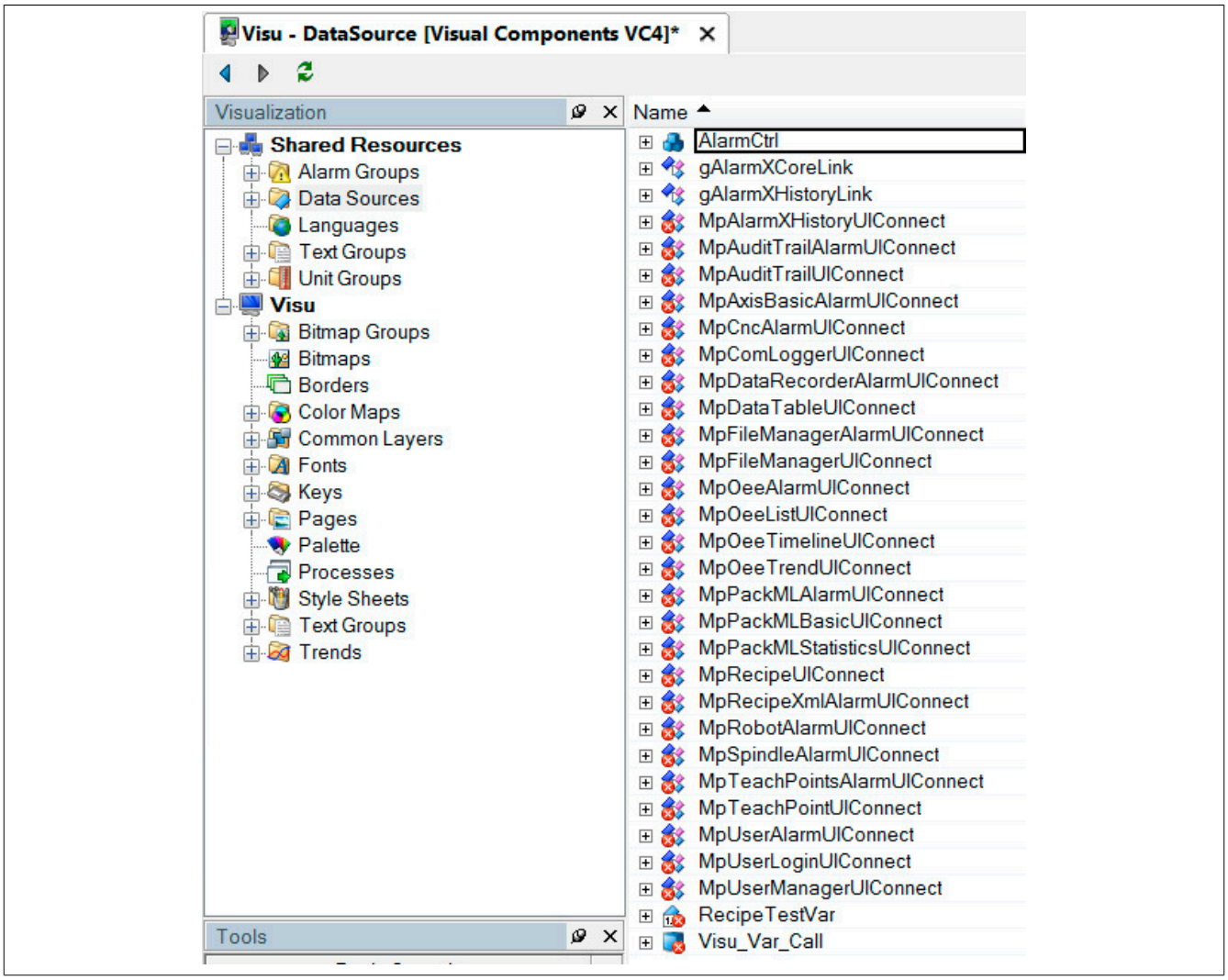




Danach wird der Austausch bestätigt. Alle Verbindungen werden jetzt automatisch erstellt.

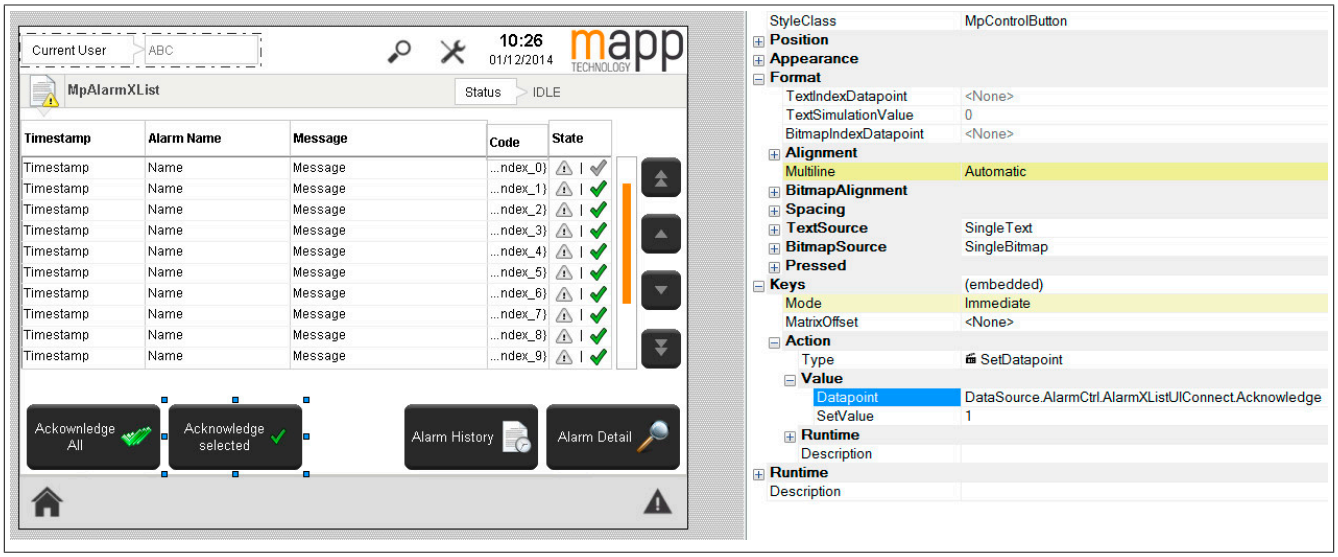


Nach dem *Refactoring* wird die Datenquelle wieder aktualisiert. Jetzt sollte die vorhergehende Verbindung *MpAlarmXListUIConnect* nicht mehr vorhanden sein. Dieser Vorgang wird für die Struktur *MpAlarmXHistoryUIConnect* wiederholt.



Alarmseite

Alle vorhandenen Datenpunkte der Alarmseite wurden durch die eigene Struktur ersetzt.



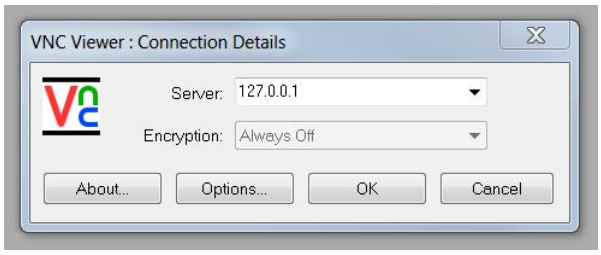
1.2.3.4 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

1.2.3.5 Visualisierung testen

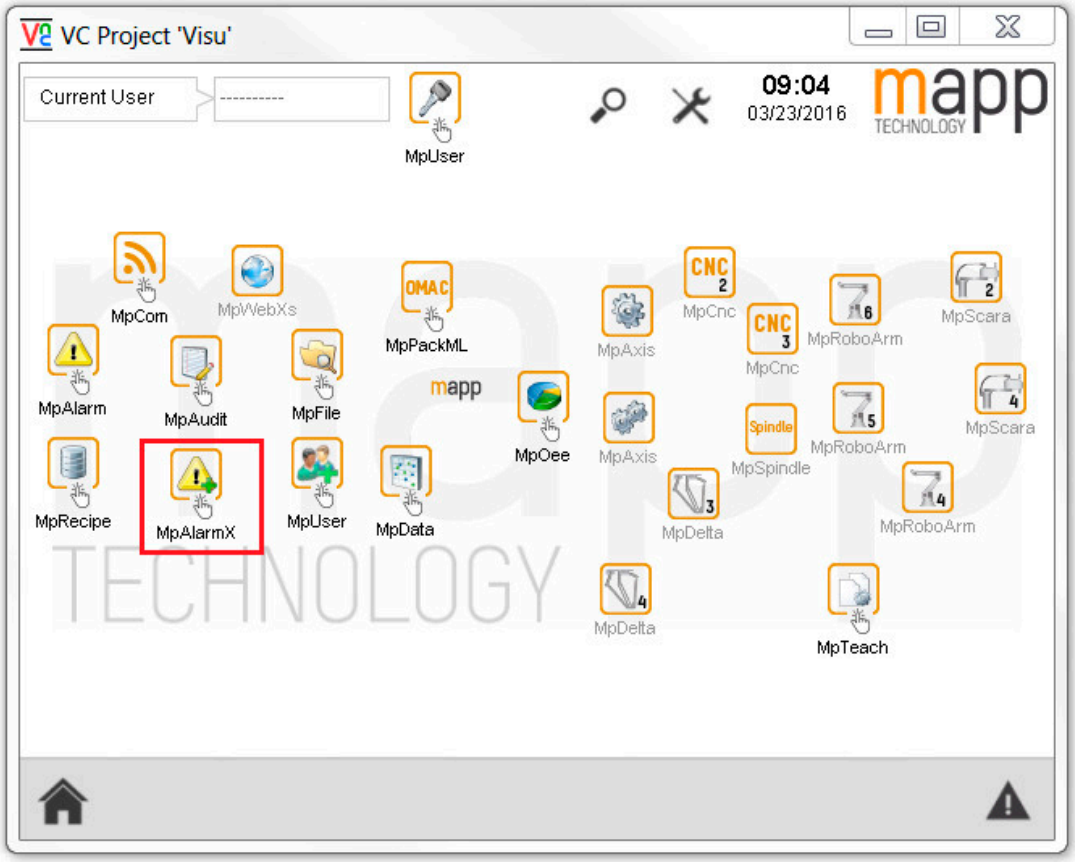
VNC Viewer öffnen

Die Visualisierung kann über den VNC Viewer geöffnet werden.



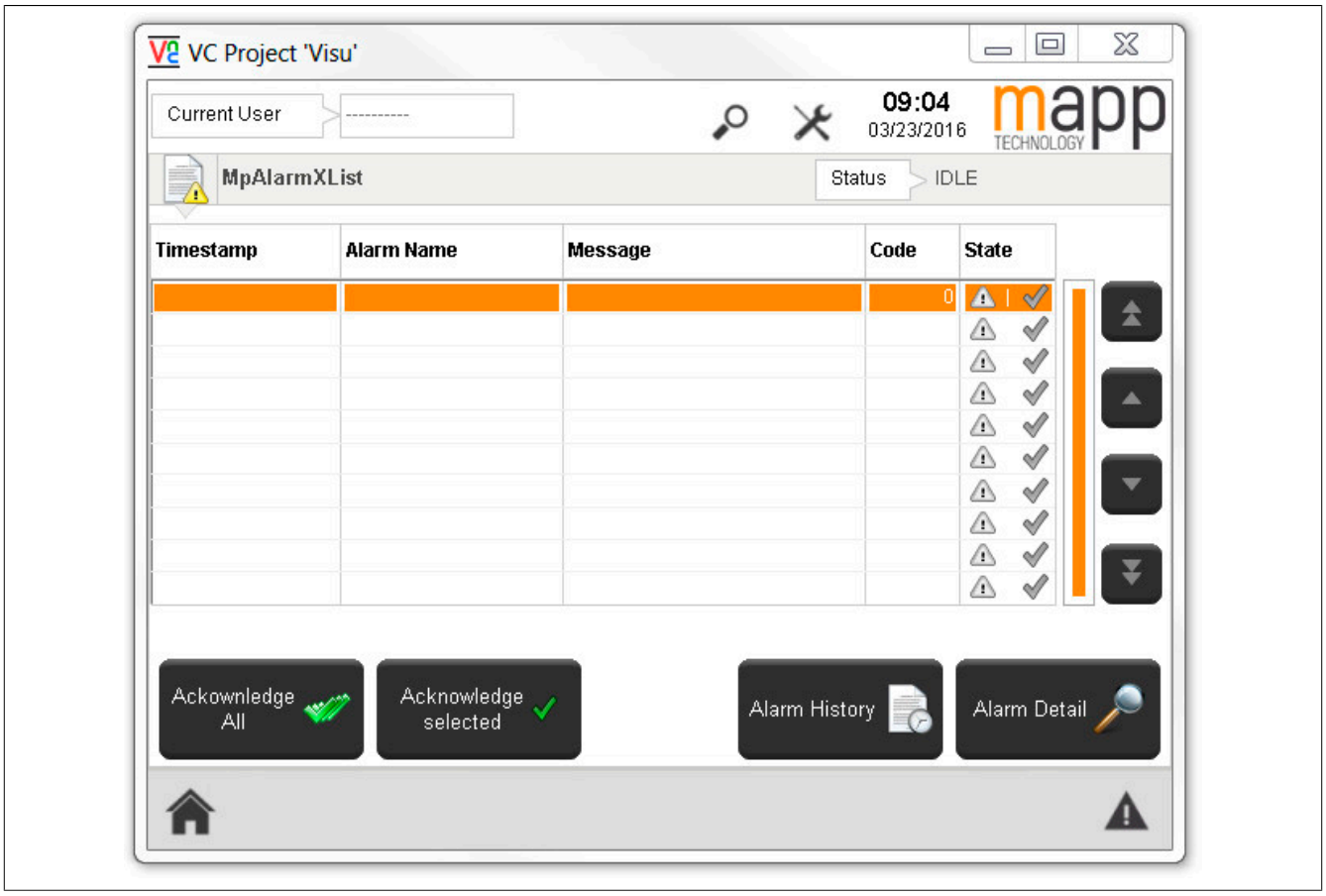
Startseite der Visualisierung

Hat man sich erfolgreich mit der Visualisierung durch den VNC Viewer verbunden, wird die Startseite angezeigt. Durch klicken des Buttons "MpAlarmX" gelangt man zu der Alarm Seite.



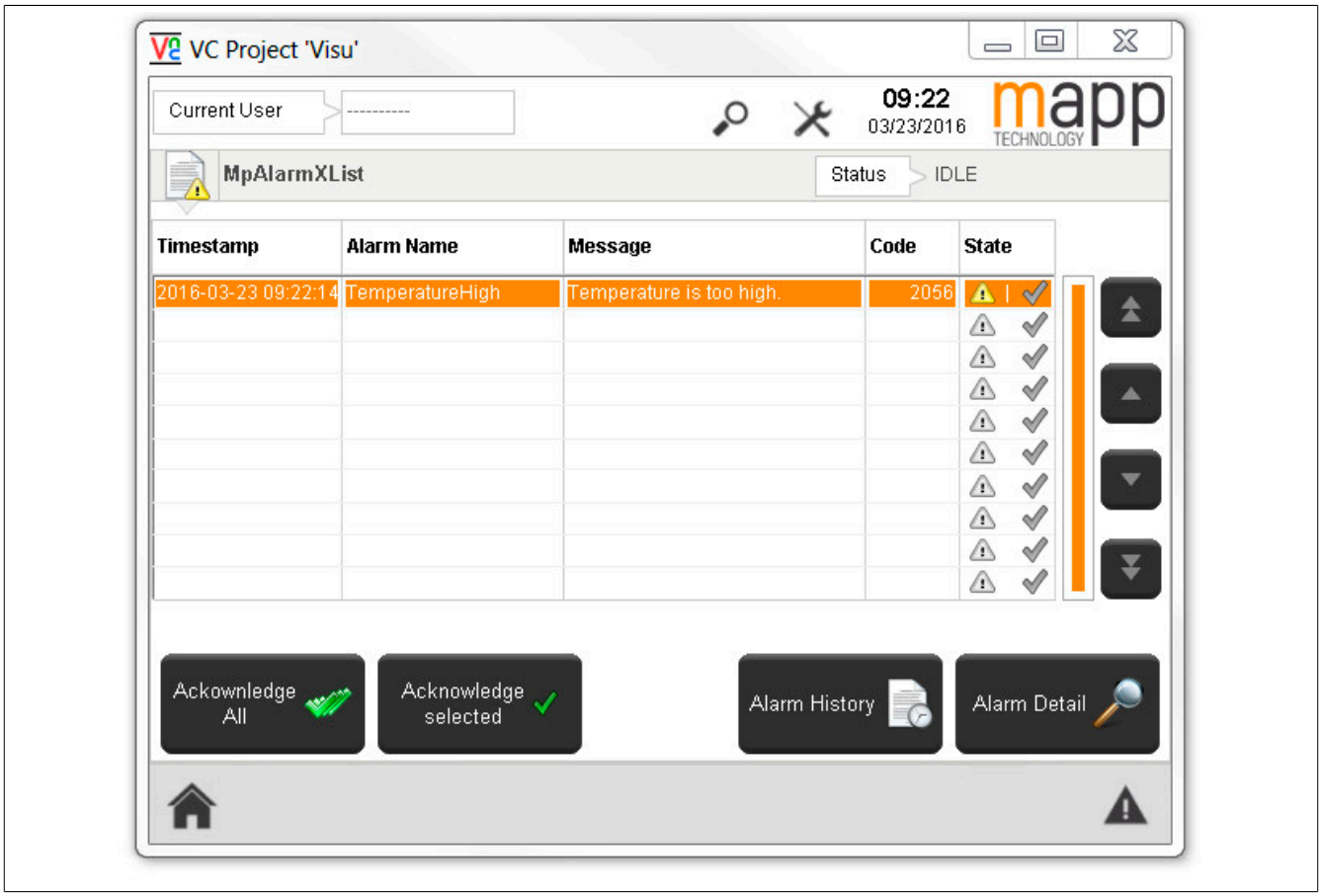
Alarm Seite

Auf der Alarm Seite können nun die aktiven Alarme angezeigt werden. Dazu muss aber erst ein Alarm gesetzt werden.



Aktivieren von Alarmen

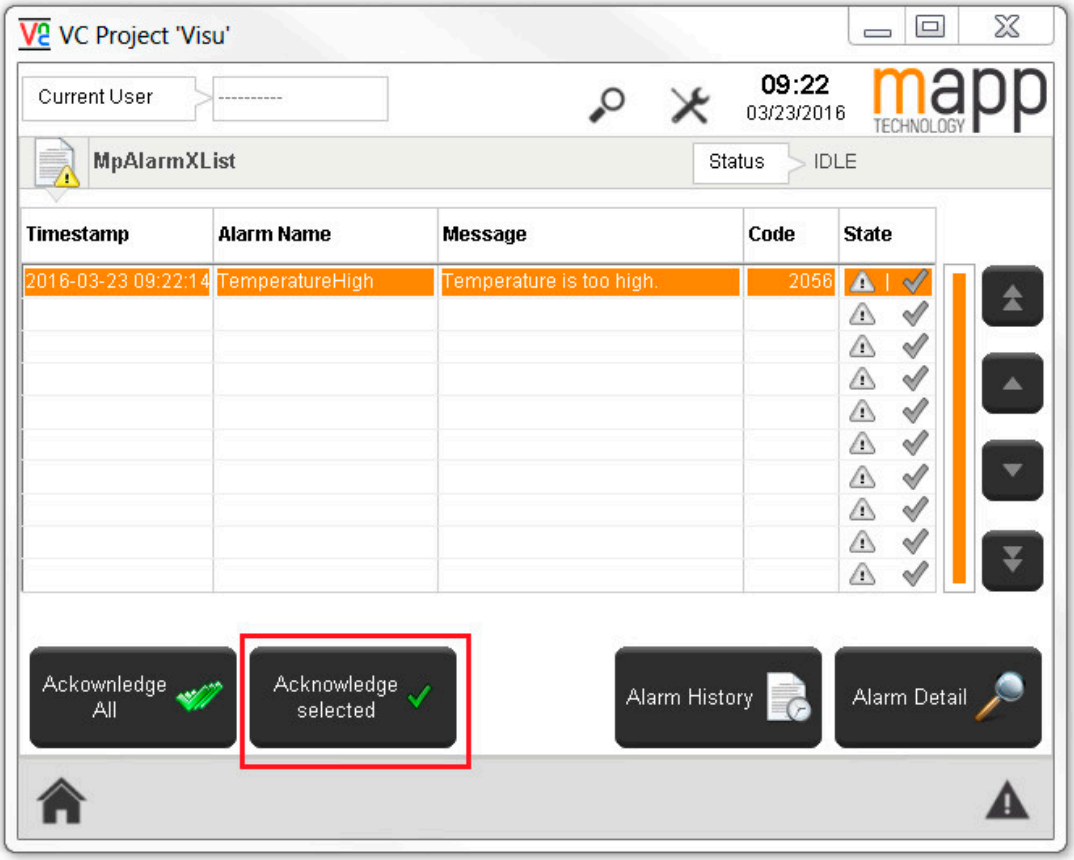
Im VNC Viewer wird nun ein Alarm angezeigt, wenn wie in Abschnitt Testen des Programms , die Variable "UserAlarm" auf *TRUE* gesetzt wird.



Quittieren von Alarmen

Wird im Programm die Variable "UserAlarm" wieder auf *FALSE* gesetzt, kann der Alarm mithilfe des Buttons "Acknowledge selected" quittiert werden.

Durch den Button "Alarm History" wird man zur Alarmseite weitergeleitet, inder die Alarm Historie angezeigt wird.



1.3 Konfiguration

1.3.1 MpAlarmXCore Konfiguration

Die Konfiguration kann mithilfe der Funktionsbausteine MpAlarmXConfigMapping und MpAlarmXConfigAlarm zur Laufzeit angepasst werden.

Auf unserem Youtube Kanal ist zum Thema Alarmkonfiguration ein Tutorial zu finden!

In diesem Abschnitt werden die einzelnen Teile der Konfiguration beschrieben.

Mapping von Alarmen

Ist ein Alarm Aktiv kann eine Aktion ausgelöst werden. Dies wird im Abschnitt "Alarm Mapping" definiert. Welche Art von Aktionen ausgewählt werden können, kann im Abschnitt Alarm Mapping gefunden werden.

Alarmliste

Im Abschnitt "Alarm List" können in der Konfiguration Benutzeralarme erstellt werden. Mehr Information kann im Abschnitt Alarmliste gefunden werden.

Retain Alarme

Werden die erweiterten Parameter der Konfiguration aktiviert, so können die Retain-Alarme konfiguriert werden. Über diese Funktion kann die gesamte Information des Alarms (Alarmzustand, zusätzliche Informationen, Alarmsnippets, konfigurierte Reaktionen usw.) nach einem Neustart der Steuerung wiederhergestellt werden. Für mehr Information siehe Abschnitt Retain Alarme.

Alarmsnippets

Um einem Alarm zusätzliche Information hinzuzufügen, können Text Snippets verwendet werden. Für mehr Information siehe Abschnitt Alarm Textsnippets.

Abfrage

Zu diesem Thema gibt es auch den Anwendungsfall "In der Alarmliste nach Alarmen suchen"

Unter "Data Queries" können Abfragen erstellt werden. Diese Abfragen können verwendet werden, um nach bestimmten Alarmen in der Alarmliste zu filtern bzw. zu suchen. Für mehr Information siehe Abschnitt Abfragen.

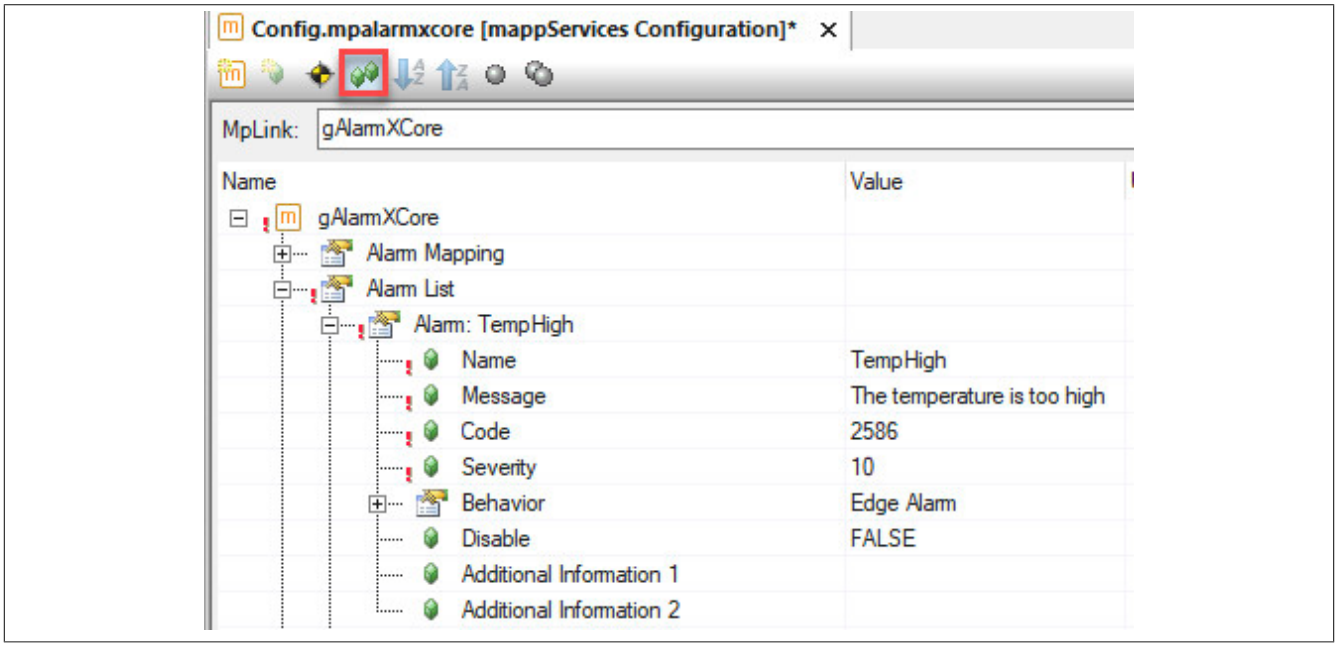
Alarmkategorie

Im Abschnitt "Alarm Categories" können Alarme kategorisiert werden. Für mehr Information siehe Abschnitt Alarmkategorien.

1.3.1.1 Alarmliste

Alarmeigenschaften

Im Abschnitt "Alarm list" werden alle Benutzeralarme definiert. Dabei können folgende Eingenschaften definiert werden:



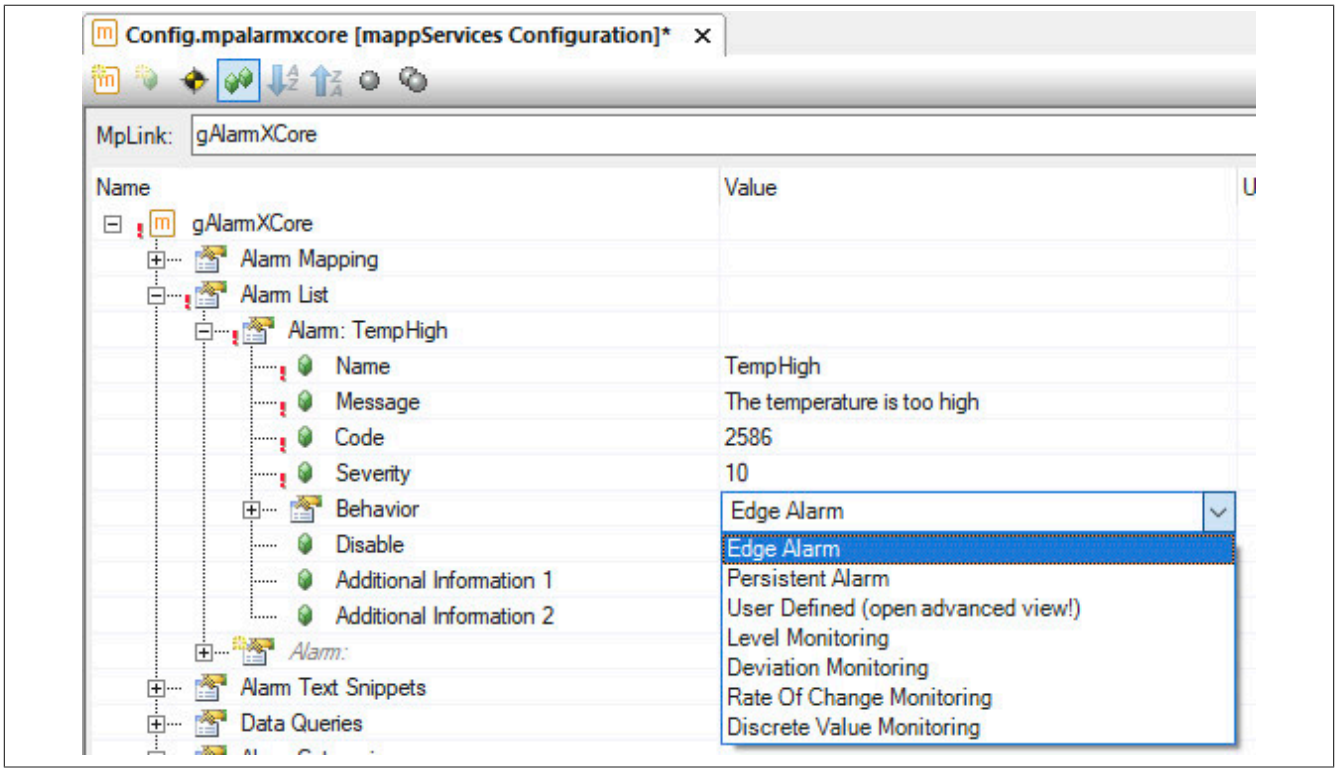
Um alle Eigenschaften anzuzeigen, müssen die erweiterten Parameter der Konfiguration aktiviert werden.

- **Name:** Der Name des Alarms. Über diesen Namen wird der Alarm identifiziert. Wird der Alarm gesetzt (MpAlarmXSet), zurückgesetzt (MpAlarmXReset) oder quittiert (MpAlarmXAcknowledge, MpAlarmXListUI). Weiter kann für jeden Alarm der aktuelle Grundzustand (MpAlarmXCheckState) ermittelt werden. Der Name eines Alarms kann keine Spezialzeichen beinhalten (z.B. ü, ö, ä, ...). Auch das Zeichen "#" ist nicht erlaubt.
- **Message:** Hier wird der Alarmtext eingetragen. Dies kann entweder statisch oder lokalisiert über das Textsystem angegeben werden.
- **Code:** Hier kann für jeden Alarm eine Nummer vergeben werden. Dieser Parameter kann genützt werden, falls mit Alarmnummern gearbeitet werden soll
- **Severity:** Die Severity gibt den Level (Schweregrad) des Alarms an. Erwartet wird hier eine Zahl (UDINT), welche frei wählbar und definierbar ist. Beispielweise kann man unkritische Alarme mit der Severity 10 versehen, während kritische Alarme die Severity 100 bekommen
- **Behavior:** Das Verhalten des Alarms wird über diesen Parameter genauer bestimmt. Details siehe weiter unten.

- **Disable:** Über diese Eigenschaft kann der Alarm ausgeschaltet (disabled) werden, siehe Grundzustand eines Alarms. Solange diese Eigenschaft aktiv ist, kann dieser Alarm nicht gesetzt werden, bzw. in einer anderen Form aufscheinen
- **AdditionalInformation1 & AdditionalInformation2:** Über diese Parameter können zusätzliche Informationen zu einem Alarm verlinkt werden, wie beispielsweise ein PDF, ein Video oder eine HTML Datei. Diese Dateien können verwendet werden, um die genaue Ursache des Alarms zu erklären oder Lösungsvorschläge zu liefern. Die Angabe erfolgt dabei statisch oder lokalisiert über das Textsystem. Zusätzlich können auch Alarm Textsnippets im Text angehängt werden.

1.3.1.1.1 Alarmverhalten

Im Abschnitt "Behavior" kann das Verhalten eines Alarms festgelegt werden.



Dabei wird zwischen 3 verschiedenen Verhaltenstypen unterschieden:

Vordefiniertes Alarmverhalten

- Persistent Alarm
- Edge Alarm

Alarm Monitoring

- Level Monitoring
- Deviation Monitoring
- Rate of Change Monitoring
- Discrete Value Monitoring

Benutzerdefiniertes Alarmverhalten

- User defined

Vordefiniertes Alarmverhalten

Es gibt zwei vordefinierte Verhaltensweisen. Dabei handelt es sich zum einen um "Edge Alarme", welche kurzlebige Alarme abbilden - Ereignis.

Zum anderen "Persistente Alarme", welche solange aktiv bleiben bis eine bestimmte Bedingung erfüllt ist - Zustand.

Für mehr Information zu den einzelnen Alarmtypen siehe Abschnitt Vordefiniertes Alarmverhalten.

Alarm Monitoring

Das Alarm Monitoring ermöglicht es das Verhalten des Alarms inklusive dem Setzen und Rücksetzen, über die Konfiguration zu definieren. Dabei werden unterschiedliche Bedingungen überwacht und der Alarmzustand wird entsprechend angepasst. Die zyklische Überwachung der Alarmbedingungen erfolgt dabei im Kontext von MpAlarmXCore. Folgende Arten des Alarm Monitoring sind möglich:

- **Level Monitoring:** Es wird eine PV überwacht. Unter- bzw. überschreitet der Wert ein bestimmtes Level, so wird ein Alarm ausgelöst
- **Deviation Monitoring:** Es wird eine PV überwacht. Erreicht der Wert eine bestimmte Abweichung vom Sollwert, so wird ein Alarm ausgelöst
- **Rate of Change Monitoring:** Es wird eine PV überwacht. Unter- bzw. überschreitet der Wert eine bestimmtes Änderungsrate, so wird ein Alarm ausgelöst
- **Discrete Value Monitoring:** Es wird eine PV überwacht. Hat diese PV einen bestimmten Wert, so wird ein Alarm ausgelöst

Für mehr Information siehe Abschnitt Alarm Monitoring.

Benutzerdefiniertes Alarmverhalten

Passt keines der beiden Standardverhalten, hat der Benutzer die Möglichkeit das Verhalten selbst festzulegen. Dazu wird als Verhalten *User defined* verwendet. Dabei ist zu beachten, dass die erweiterten Eigenschaften der Konfiguration aktiviert werden müssen, um die verschiedenen Einstellmöglichkeiten zu sehen.

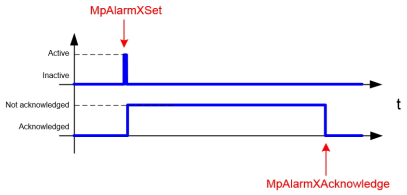
Im Folgenden werden die einzelnen Einstellungen im Detail beleuchtet:

- **Auto Reset:** Hier kann festgelegt werden, ob der Alarm direkt nachdem Setzen automatisch wieder zurückgesetzt werden soll (`TRUE`), oder ob das Zurücksetzen manuell über MpAlarmXReset erfolgen muss (`FALSE`)
- **Acknowledge:** Legt fest ob eine Quittierung notwendig ist. Dabei gibt es drei Varianten:
  - Disabled: Es ist keine Quittierung notwendig
  - Required: Es ist eine Quittierung notwendig
  - Required after Active: Es ist eine Quittierung notwendig, aber das Quittieren ist erst möglich, nachdem der Alarm wieder inaktiv ist
  - Required and Resetable: Es ist eine Quittierung notwendig, der Quittierstatus des Alarms wird zurückgesetzt sobald der Alarm nocheinmal aktiviert wurde
- **Multiple Instances:** Definiert ob ein Alarm einfach oder mehrfach instanziert werden kann, also ob derselbe Alarm mehr als einmal aktiviert werden und in der Alarmliste aufscheinen kann
- **Reaction Until Acknowledge:** Legt fest wie lange die Reaktion anhält. Entweder bis der(die) Alarm(e) quittiert wurde(n), oder bis der(die) Alarm(e) zurückgesetzt wurde(n)
- **Retain:** Die Eigenschaft "Retain" legt fest, ob der aktuelle Zustand des Alarms nach einem Neustart des Zielsystems vorhanden sein soll oder nicht. Bei "Retain = `TRUE`" gilt es die Retain-Parameter in der Konfiguration zu beachten. Für mehr Information siehe Abschnitt "Retain Alarme" in der MpAlarmXCore Konfiguration.
- **History Report:** Über diesen Abschnitt kann man festlegen welche Zustandsänderungen in der Alarm Historie auscheinen sollen.
  - Inactive to Active: Definiert ob in der Alarm Historie das Setzen des Alarms aufgezeichnet werden soll
  - Active to Inactive: Definiert ob in der Alarm Historie das Zurücksetzen des Alarms aufgezeichnet werden soll
  - Unacknowledged to Acknowledged: Definiert ob in der Alarm Historie das Quittieren des Alarms aufgezeichnet werden soll
  - Acknowledged to Unacknowledged: Definiert ob in der Alarm Historie der Zustandswechsel des Alarms von Quittiert auf Unquittiert aufgezeichnet werden soll

1.3.1.1.2 Vordefiniertes Alarmverhalten

Edge Alarm

Edge Alarme sind kurzlebige Alarme. Man kann sie mit einem Ereignis gleichsetzen. Sofort nach dem Setzen dieser Alarme werden sie automatisch zurückgesetzt. Wie in dem Zeitdiagramm ersichtlich wird ein Edge Alarm über MpAlarmXSet gesetzt. Das Zurücksetzen erfolgt automatisch und muss nicht nochmal explizit vom Benutzer ausgeführt werden. Der Benutzer muss den Edge Alarm nur noch quittieren über MpAlarmXAcknowledge bzw. über die Visualisierung.



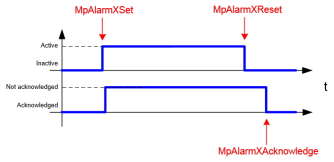
Edge Alarme können mehrfach instanziert werden, heißt Edge Alarme können mehr als einmal gesetzt werden. Wird ein Edge Alarm zweimal über MpAlarmXSet gesetzt, so werden in der Alarmliste zwei Einträge angezeigt. Quittieren ist für Edge Alarme notwendig.

Einige Beispiele für typische Edge Alarme:

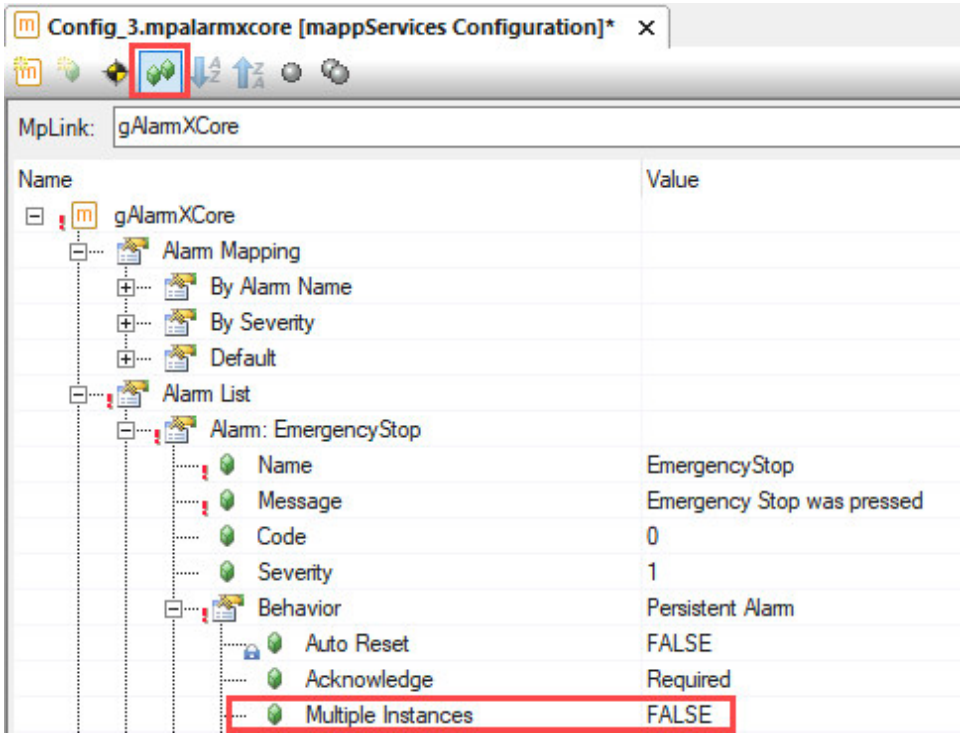
- Rezept konnte nicht geladen werden: "Recipe1"
- SMS konnte nicht an "Schichtleiter" gesendet werden
- Speichermedium nicht vorhanden
- Diese Operation ist nicht erlaubt

Persistent Alarm

Persistente Alarme bleiben aktiv bis eine bestimmte Bedingung erfüllt ist, bis sich der Zustand des Alarms verändert. Diese Alarme werden manuell gesetzt und auch zurückgesetzt. Wie in dem Zeitdiagramm ersichtlich wird ein Persistenter Alarm über MpAlarmXSet gesetzt und nachdem eine bestimmte Bedingung erfüllt ist (z.B. Temperatur ist wieder im Normalbereich) werden sie wieder zurückgesetzt über MpAlarmXReset. Persistente Alarme können nach ihrer Aktivierung jederzeit über MpAlarmXAcknowledge bzw. über die Visualisierung quittiert werden. Die Quittierung ist unabhängig davon möglich, ob der Alarm noch aktiv ist, oder bereits zurückgesetzt wurde. Es gilt zu beachten, dass Persistente Alarme welche einen Punkt im Alarmnamen beinhalten, wie zum Beispiel "Temp.High", nicht mit MpAlarmXAcknowledge quittiert werden können.



Standardmäßig werden persistene Alarme einfach instanziert (Multiple Instances = `FALSE`), heißt ein Persistenter Alarm scheint maximal einmal in der Alarmliste auf. Die Option kann durch aktivieren der erweiterten Eigenschaften geändert werden.



Bei "Multiple Instances = `FALSE`", wird ein Persistenter Alarm nicht quittiert und bleibt in der Alarmliste. Wird er nun wieder aktiv, so wird *kein* neuer Eintrag in der Alarmliste erstellt, sondern der bereits existierende Alarm wird wieder aktiv. Standardmäßig wird dabei der Zeitstempel nicht aktualisiert. Bei der Verwendung von Alarmsnippets wird das Snippet ebenfalls nicht aktualisiert.

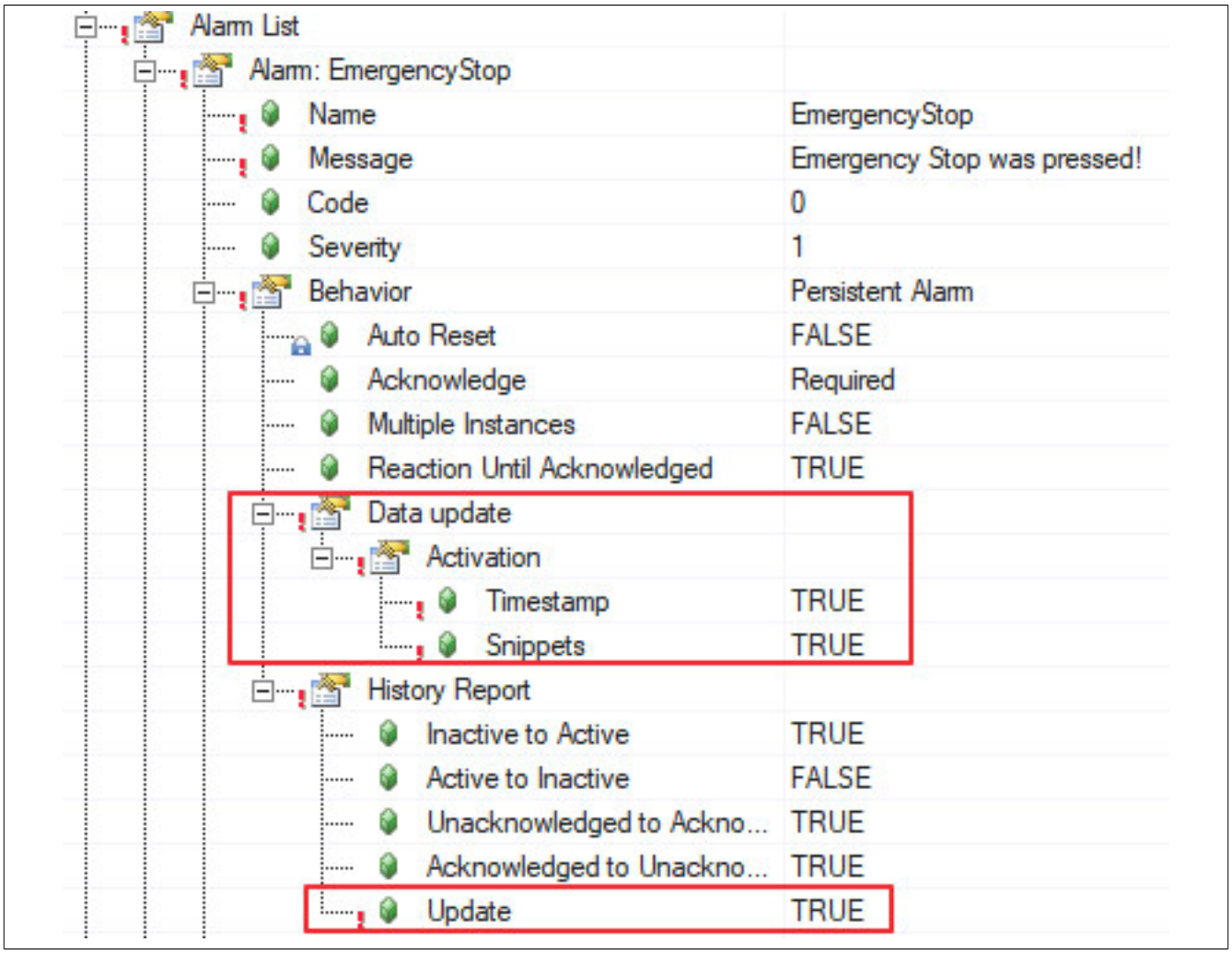
Quittieren ist für Persistente Alarme notwendig. Einige Beispiele für typische Persistente Alarme:

- Die Temperatur (123°C) liegt nicht im Normalbereich (100°C - 120°C)
- Wasserstand in Tank zu hoch
- Not Aus gedrückt
- X20DI8371 nicht gesteckt

Aktualisierung des Zeitstempels

Die Einstellung kann bei edge Alarmen sowie persistenten Alarmen durchgeführt werden.

Wird eine Aktualisierung des Zeitstempels oder Alarmsnippet gewünscht, so kann dies unter "Data update" aktiviert werden:



Damit die Aktualisierung in der Alarmhistorie ebenfalls aufgezeichnet wird, kann unter "History Report" der Parameter "Update" auf `TRUE` gesetzt werden.

Das aktualisieren des Zeitstempels bzw. Snippets ist auch für bereits aktive Alarme möglich. Durch ein weiteres Aufrufen des Alarms mithilfe von MpAlarmXSet kann dadurch der Zeitstempel bzw. das Snippet aktualisiert werden.

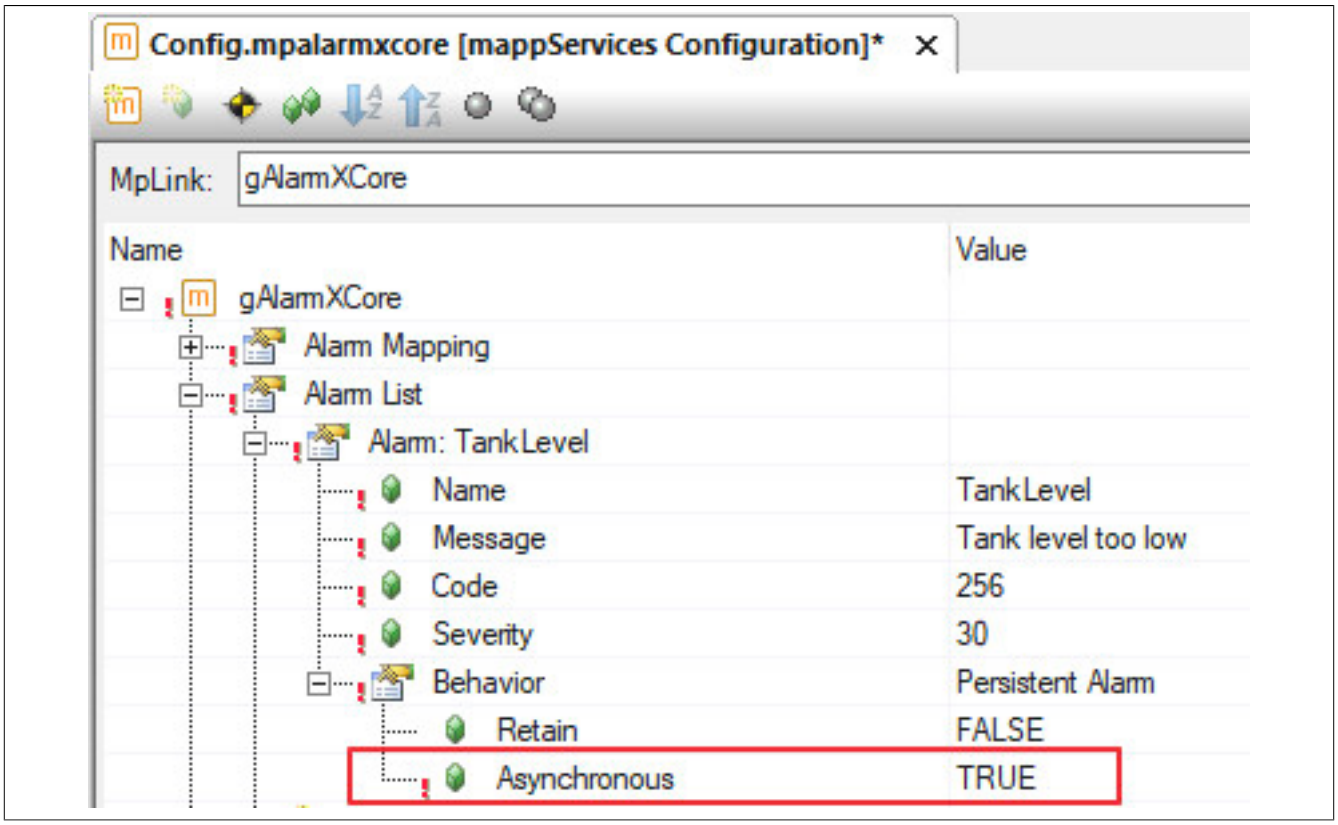
Zu diesem Thema gibt es den Anwendungsfall Aktualisieren der Alarm-Information.

Asynchrone Alarme

Die Einstellung kann bei edge Alarmen sowie persistenten Alarmen durchgeführt werden.

Durch "Asynchronous = `TRUE`" kann ein Alarm asynchron behandelt werden.

Mehr Information zu asynchrone Alarme kann im Abschnitt Synchrone und asynchrone Alarme gefunden werden.



1.3.1.1.3 Alarm Monitoring

Wie die Monitoring-Funktion verwendet werden kann, ist im Anwendungsfall Alarmverwaltung in der Konfiguration erklärt.

Auf unserem B&R Tutorial Portal ist ein Tutorial zum Thema Alarm Monitoring (GER/ENG) zu finden. (Zugangsdaten)

Folgende Datentypen werden für das Alarm-Monitoring unterstützt:

- BOOL
- SINT
- INT
- DINT
- USINT
- UINT
- UDINT
- REAL
- LREAL
- Abgeleitete Datentypen können nur mit AR D4.26 oder AR4.34 registriert werden
- Es können einzelne Variablen aber auch Strukturen registriert werden

Im Folgenden werden die Arten des Alarm Monitoring im Detail beschrieben.

Level Monitoring

(= Überwachung des Levels)

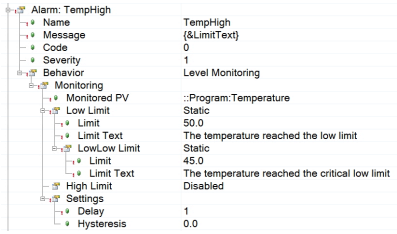
Es wird das Level einer PV überwacht. Die PV wird in der Konfiguration festgelegt. Dabei besteht die Möglichkeit zwei untere, sowie zwei obere Grenzwerte zu definieren. Wird ein Grenzwert überschritten, so wird der Alarm aktiviert. Die Grenzwerte sind standardmäßig deaktiviert (= *Disabled*). Wird ein oberer oder unterer Grenzwert aktiviert, kann festgelegt werden, ob der Grenzwert statisch oder dynamisch erfolgt.

- **Statisch:** Der Grenzwert wird direkt in der Konfiguration eingegeben. Im Bild unten sind statische Grenzwerte eingestellt
- **Dynamisch:** Der Grenzwert wird über eine weitere PV definiert. Diese Grenzwert-PV wird mit der Konfiguration verbunden

Weiterhin kann für jeden Grenzwert, über den Parameter *Limit Text,* ein eigener Text als Text Snippet eingegeben werden. Dieses Text Snippet kann dann beim Alarmtext verwendet werden, siehe Bild unten. Abhängig vom überschrittenen Grenzwert ändert sich der Text im *Limit Text* Snippet:

- Temperatur <= 50°C: The temperature reached the low limit
- Temperatur <= 45°C: The temperature reached the critical low limit

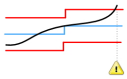
Wird der Grenzwert wieder unterschritten, so wird auch der Alarm wieder zurückgesetzt.



Level Monitoring Alarme sind standardmäßig exklusive Alarme, heißt es wird immer nur eine Grenzüberschreitung zu einem Zeitpunkt angezeigt in der Alarmliste. Im obigen Beispiel hieße das, dass entweder ein Alarm existiert, welcher die 50°C unterschritten hat oder ein Alarm existiert, welcher 45°C unterschritten hat. Aber es werden nicht beide zum selben Zeitpunkt angezeigt. Mehr Details zur Alarm Exklusivität weiter unten.

Deviation Monitoring

(= Überwachung der Abweichung)



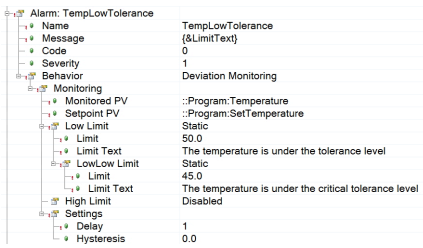
Es wird die Abweichung von einem definierten Level überwacht. Dazu wird zum einen eine PV mit dem aktuellen Wert angegeben und weiterhin eine zweite PV, welche den Sollwert vorgibt. Weicht nun der aktuelle Wert um eine definierte Toleranz vom Sollwert ab, so wird ein Alarm generiert. Dabei besteht die Möglichkeit zwei untere, sowie zwei obere Grenzwerte zu definieren. Wird ein Grenzwert erreicht/überschritten, so wird der Alarm aktiviert. Die Grenzwerte sind standardmäßig deaktiviert (= *Disabled*). Wird ein oberer oder unterer Grenzwert aktiviert, kann festgelegt werden, ob der Grenzwert statisch oder dynamisch erfolgt.

- **Statisch:** Die Toleranz wird direkt in der Konfiguration eingegeben. Im Bild unten sind statische Toleranzwerte eingestellt
- **Dynamisch:** Der Toleranzwert wird über eine weitere PV definiert. Diese Toleranz-PV wird mit der Konfiguration verbunden

Weiterhin kann für jedes über- bzw. unterschreiten einer Toleranz, über den Parameter *Limit Text,* ein eigener Text als Text Snippet eingegeben werden. Dieses Text Snippet kann dann beim Alarmtext verwendet werden, siehe Bild unten. Abhängig von der unter- bzw. überschrittenen Toleranz ändert sich der Text im *Limit Text* Snippet:

- The temperature is under the tolerance level
- The temperature is under the critical tolerance level

Liegt der aktuelle Wert wieder unter/über dem Toleranzbereich, so wird der Alarm zurückgesetzt. Im Beispiel muss die Variable "Temperature" größer als 50 sein, damit der Alarm zurückgesetzt wird.



Deviation Monitoring Alarme sind standardmäßig exklusive Alarme, heißt es wird immer nur eine Toleranzüberschreitung zu einem Zeitpunkt angezeigt in der Alarmliste. Im obigen Beispiel hieße das, dass entweder ein Alarm existiert, welcher das untere Toleranz Level unterschritten hat, oder ein Alarm existiert, welcher das kritische Toleranz Level unterschritten hat. Aber es werden nicht beide zum selben Zeitpunkt angezeigt. Mehr Details zur Alarm Exklusivität weiter unten.

Rate of Change Monitoring

(= Überwachung der Änderungsrate)

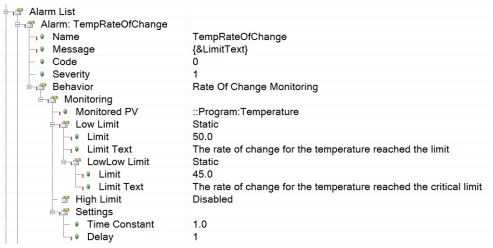
Es wird das die Änderungsrate einer PV überwacht. Die PV wird in der Konfiguration festgelegt. Dabei besteht die Möglichkeit zwei untere, sowie zwei obere Grenzwerte zu definieren. Wird ein Grenzwert überschritten, so wird der Alarm aktiviert. Die Grenzwerte sind standardmäßig deaktiviert (= *Disabled*). Wird ein oberer oder unterer Grenzwert aktiviert, kann festgelegt werden, ob der Grenzwert statisch oder dynamisch erfolgt.

- **Statisch:** Der Grenzwert wird direkt in der Konfiguration eingegeben. Im Bild unten sind statische Grenzwerte eingestellt
- **Dynamisch:** Der Grenzwert wird über eine weitere PV definiert. Diese Grenzwert-PV wird mit der Konfiguration verbunden

Weiterhin kann für jeden Grenzwert, über den Parameter *Limit Text,* ein eigener Text als Text Snippet eingegeben werden. Dieses Text Snippet kann dann beim Alarmtext verwendet werden, siehe Bild unten. Abhängig vom überschrittenen Grenzwert ändert sich der Text im *Limit Text* Snippet:

- The rate of change for the temperature reached the limit
- The rate of change for the temperature reached the critical limit

Wird der Grenzwert wieder überschritten, so wird auch der Alarm wieder zurückgesetzt.

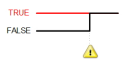


Rate of Change Monitoring Alarme sind standardmäßig exklusive Alarme, heißt es wird immer nur eine Grenzüberschreitung zu einem Zeitpunkt angezeigt in der Alarmliste. Im obigen Beispiel hieße das, dass entweder ein Alarm existiert, welcher den Grenzwert der Änderungsrate erreicht hat, oder ein Alarm existiert, welcher den kritischen Grenzwert der Änderungsrate erreicht hat. Aber es werden nicht beide zum selben Zeitpunkt angezeigt. Mehr Details zur Alarm Exklusivität weiter unten.

Der Parameter "Time Constant" ist die Zeitkonstante eines PT1 Filters. Der Wert kann statisch oder dynamisch über eine Prozessvariable vom Datentypen LREAL festgelegt werden.

Discrete Value Monitoring

(= Überwachung bestimmter Werte)

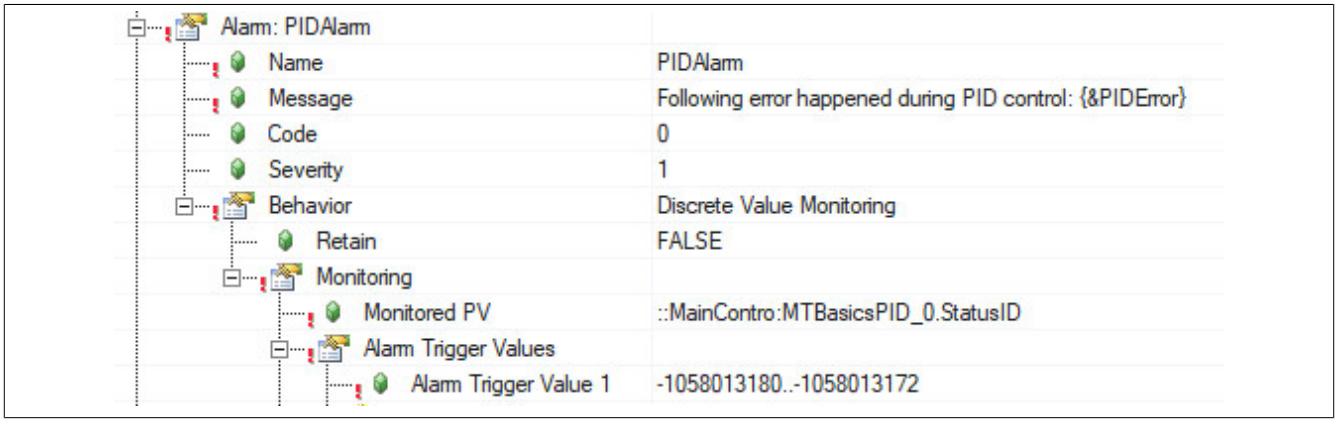


Es werden bestimmte PV Werte überwacht. Die PV wird in der Konfiguration festgelegt. Nimmt die PV einen speziellen Wert an, dann wird ein Alarm aktiviert. Es können ein oder mehrere Werte definiert werden. Die Definition erfolgt über die *Alarm Trigger Values*. Über die Zeichen ".." kann ein Wertebereich definiert werden:

- **1..5:** Ein Alarm wird gesetzt für den Fall, dass der Wert der PV 1, 2, 3, 4 oder 5 annimmt
- **3:** Ein Alarm wird gesetzt für den Fall, dass der Wert der PV 3 ist
- **TRUE:** Ein Alarm wird gesetzt für den Fall, dass der Wert der PV `TRUE` ist.

Speziell in Verbindung mit dem Verarbeiten von Funktionsbaustein Fehlern sind Discrete Value Monitoring Alarme von Interesse. Es kann überwacht werden, ob ein bestimmter Fehler am Funktionsbaustein angezeigt wird und auf diesen mit einem Alarm reagiert werden. Im Beispiel unten wird die "StatusID" eines MTBasicsPID Funktionsbausteins überwacht. Wird nun ein Fehler am Ausgang des Funktionsbaustein zurückgegeben, so wird ein Alarm generiert, welcher die Fehlernummer enthält.

Liegt der Wert wieder außerhalb des definierten Bereichs, so wird der Alarm zurückgesetzt.



Die PVs, welche in der Konfiguration definiert werden für das Discrete Value Monitoring, müssen BOOL, INT, DINT, SINT, UDINT, UINT oder USINT sein.

Folgende Eigenschaften gelten für alle Monitoring-Typen:

Exklusive und nicht exklusive Alarme

Monitoring Alarme sind exklusive Alarme:

- **Exklusiv:** Sind für einen Alarm mehrere Grenzwerte auf einmal unter- oder überschritten, so wird trotzdem nur ein aktiver Alarm mit dem höchsten (bzw. niedrigsten) Grenzwert in der Alarmliste angezeigt und mehrere inaktive aber nicht quittierte. Das bedeutet, an der Komponente MpAlarmXCore wird am Ausgang "ActiveAlarms" ein aktiver Alarm angezeigt. Am Ausgang "PendingAlarms" werden mehrere Alarme angezeigt, da mehrere Grenzwerte überschritten bzw. unterschritten wurden.
- **Nicht exklusiv:** Sind für einen Alarm mehrere Grenzwerte auf einmal unter- oder überschritten, so wird jeweils ein Alarm pro unter- bzw. überschrittenen Grenzwert in der Alarmliste angezeigt

Blockieren

Durch den Parameter *Inhibit-PV* ist es möglich die Monitoring Alarme zu blockieren. Dadurch ist es möglich die Aktivierung von Alarmen unter definierten Bedinungen zu verhindern, beispielsweise sollen keine Temperatur-Alarme aktiviert werden, solange die Heizelemente einer Maschine gewartet werden.

Dazu wird über die Konfiguration eine Prozessvariable mit Datentyp BOOL ausgewählt. Solange diese Variable den Wert TRUE hat, wird der Alarm nicht ausgelöst. Bereits aktive Alarme dieses Typs werden zurückgesetzt. Wird die *Inhibit-PV* von TRUE auf FALSE gesetzt, so wird überprüft ob die Alarmbedingung aktiv ist und unter Berücksichtigung von Verzögerung und Hysterese wird der Alarm gesetzt.

Verzögerung

Über den Parameter *Delay* kann eine Verzögerungszeit für das Aktivieren des Alarms festgelegt werden. Diese kann dynamisch oder statisch festgelegt werden. Wird beispielsweise eine statische Verzögerungszeit von 3 Sekunden eingestellt, wird der Alarm erst 3 Sekunden nach Erfüllen der Setzbedingung gesetzt. Die Setzbedingung muss während dem gesamten *Delay* Zeit gegeben sein. Bei einer dynamischen Angabe muss eine Prozessvariable vom Datentypen REAL angegeben werden.

Hysterese

Über den Parameter Hysterersis kann eine Hystere definiert werden, damit ein mögliches Flickern verhindert wird. Der Hysteresewert gilt in positive und negative Richtung vom Sollwert. Die Hysterese kann statisch oder dynamisch angegeben werden. Wird beispielsweise eine statische Hysterese von 1°C eingestellt bei einem Sollwert von 80°C, so wird der Alarm bei 81°C gesetzt und bei 79°C zurückgesetzt. Bei der Angabe einer dynamischen Hysterese muss eine PV vom Datentypen LREAL angegeben werden.

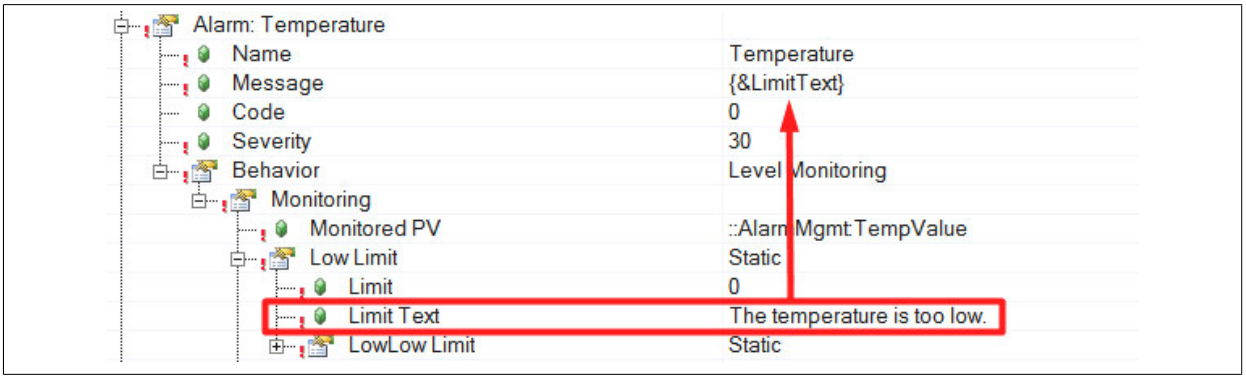
Mithilfe von Textschnipsel kann dem Alarm zusätzliche Information übergeben werden.

Einfügen von Zusätzlicher Information über Text Snippets

Mithilfe von Text Snippets kann dem Alarm zusätzliche Information übergeben werden.

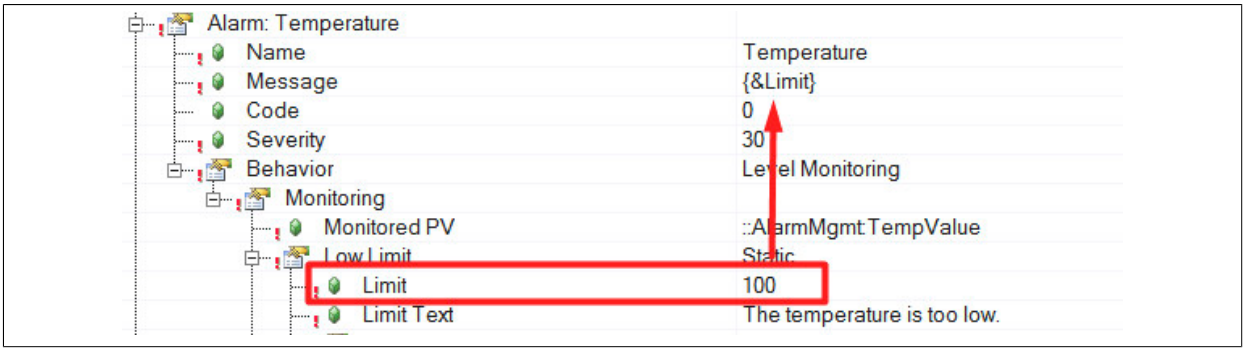
Folgende Text Snippets können im Alarmtext miteingebunden werden:

- **LimitText:** Über das Text Snippet "LimitText" kann der Text, welcher über den Parameter "Limit Text" definiert wird, der Alarmnachricht übergeben werden. Das Text Snippet wird am Parameter "Message" folgendermaßen angegeben: {&LimitText}



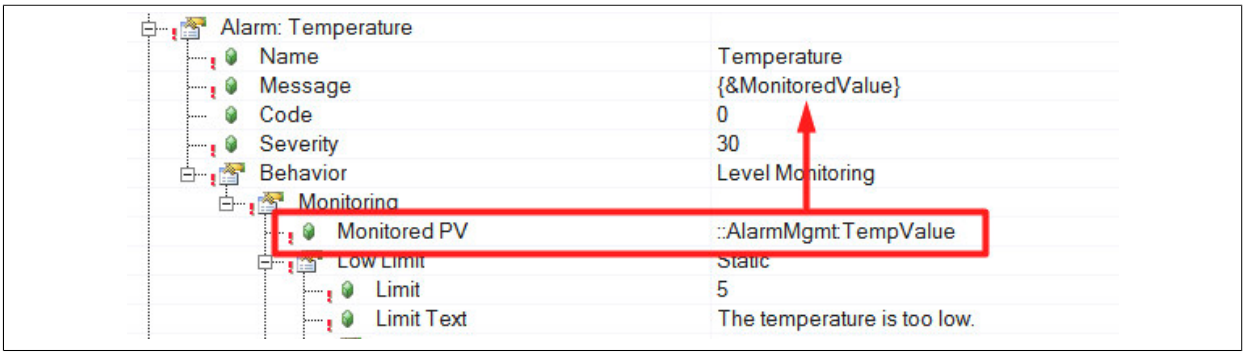
Das Text Snippet kann bei den den Alarmverhalten Level Monitoring, Deviation Monitoring und Rate Of Change Monitoring verwendet werden.

- **Limit:** Über das Text Snippet "Limit" kann das in der Konfiguration definierte Limit dem Alarmtext übergeben werden. Wurde zum Beispiel ein Limit von 100 definiert, so wird dieser Wert in der Alarmnachricht mitangezeigt. Das Text Snippet wird am Parameter "Message" folgendermaßen angegeben: {&Limit}



Das Text Snippet kann bei den den Alarmverhalten Level Monitoring, Deviation Monitoring und Rate Of Change Monitoring verwendet werden.

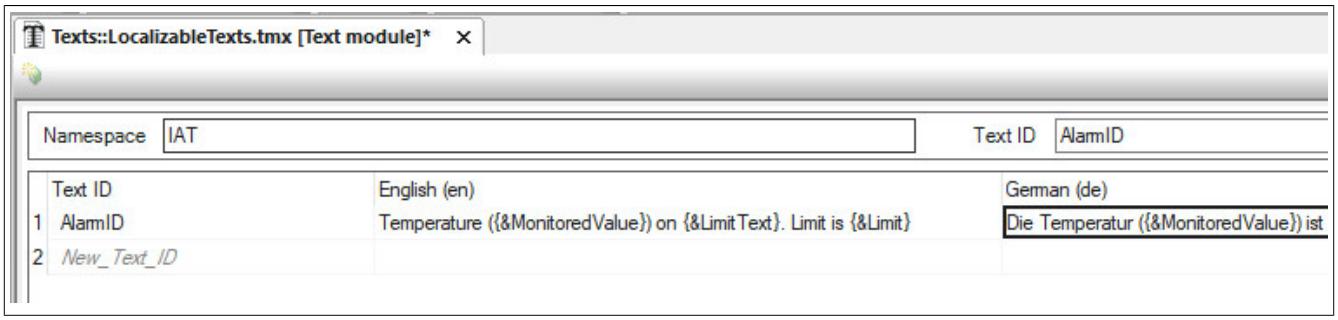
- **MonitoredValue:** Über das Text Snippet "MonitoredValue" kann der Wert der Prozessvariable übergeben werden, welche am Parameter "Monitored PV" angegeben wurde. Hat die Variable bei der Aktivierung des Alarms den Wert 120, so wird in der Alarmnachricht der Wert 120 angezeigt.
  Das Text Snippet wird am Parameter "Message" folgendermaßen angegeben: {&MonitoredValue}



Das Text Snippet kann bei den den Alarmverhalten Level Monitoring, Deviation Monitoring, Rate Of Change Monitoring und Discrete Value Monitoring verwendet werden.

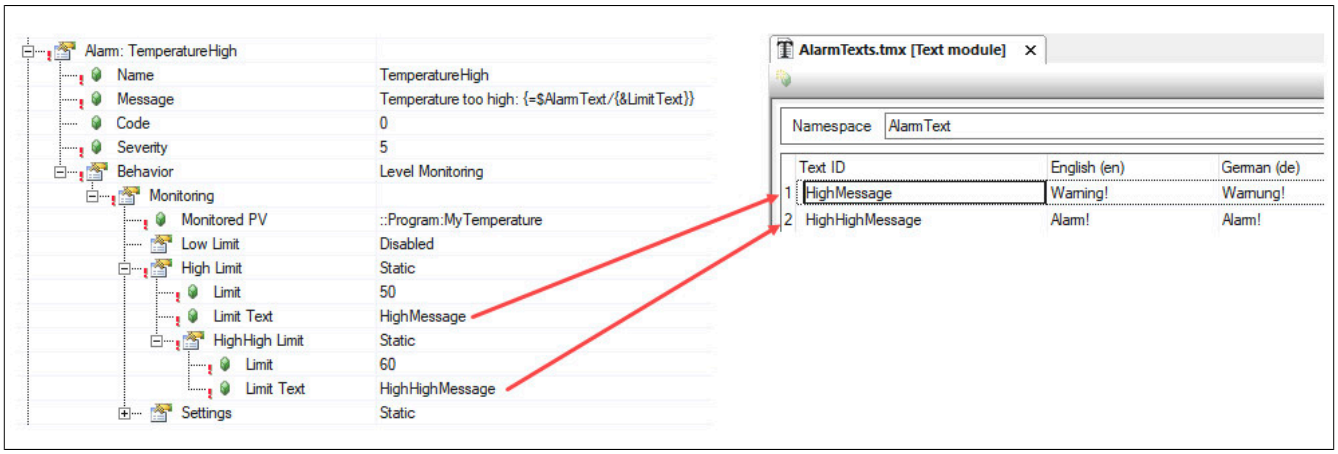
Text Snippets lokalisiert über das Textsystem

Wird die Alarmnachricht lokalisiert über das Textsystem definiert, so können die Text Snippets auch im Textsystem verwendet werden:



Für mehr Information siehe Alarm Texte.

Damit die Text Snippets lokalisiert angegeben werden können, wird zusätzlich das "=" Symbol in der Syntax verwendet "{=$Namespace/{&LimitText}}":



In diesem Fall ist die Alarmnachricht "Temperature too high: Warning!" oder "Temperature too high: Alarm!"

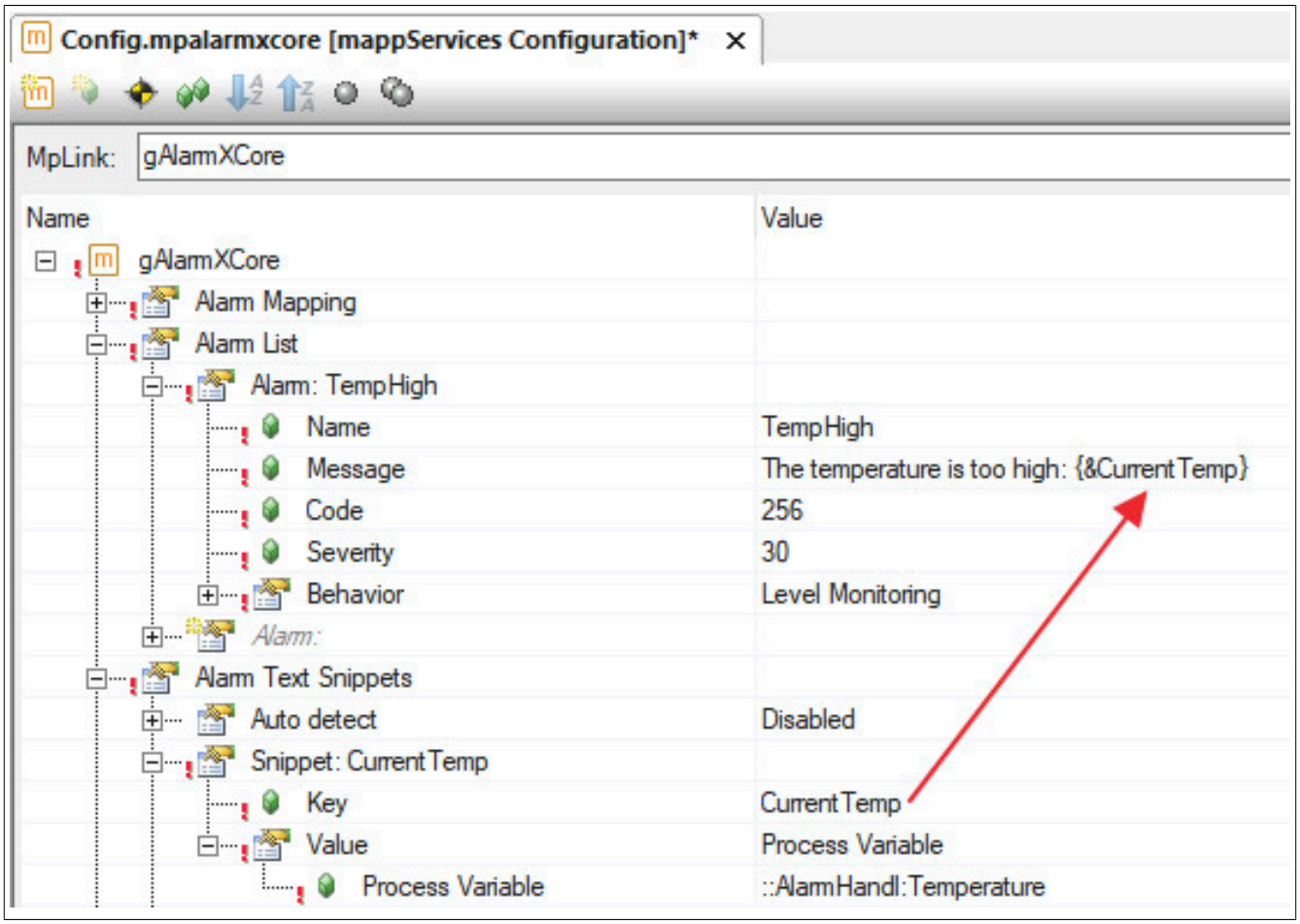
1.3.1.2 Alarm Mapping

Ist ein Alarm Aktiv kann eine Aktion ausgelöst werden. Dies wird im Abschnitt "Alarm Mapping" definiert. Welche Art von Aktionen ausgewählt werden können, kann im Abschnitt Alarm Mapping gefunden werden.

1.3.1.3 Alarm Textsnippets

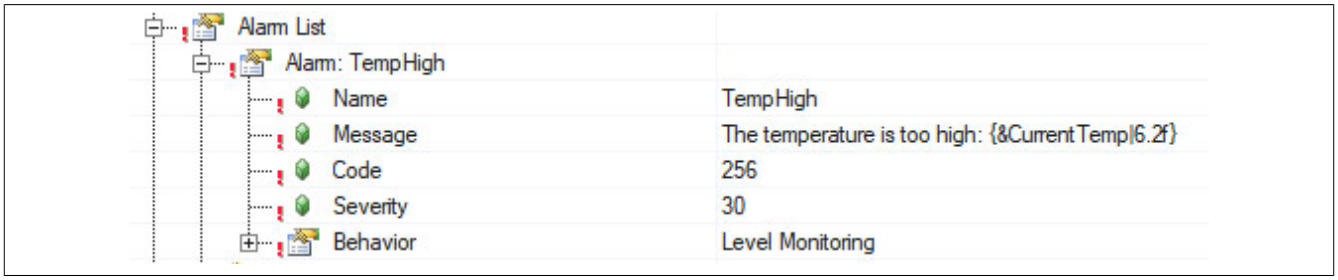
Um einem Alarm zusätzliche Information hinzuzufügen, können Textsnippets verwendet werden. Diese werden im Abschnitt "Alarm Text Snippets" angegeben. Es gilt zu beachten, dass hier erstellte Alarm Textsnippets nur für Alarme verwendet werden können, welche im Abschnitt Alarmlist definiert wurden. Alarm Textsnippets können nicht in den Alarmen anderen mapp Komponenten verwendet werden.

In diesem Abschnitt werden alle Textsnippets definiert, welche dann für unterschiedliche Alarmtexte verwendet werden können. Ein- und dasselbe Textsnippet kann dabei für einen oder für mehrere Alarm verwendet werden. Für jedes Textsnippet wird eine ID ("Key") definert, über welchen das Textsnippet im Alarmtext identifiziert wird. Weiter wird die PV definiert, welche die Applikationsdaten liefert. Zum Schluss muss das Textsnippet in den Alarmtext eingebunden werden. Dazu wird die Syntax {&Key} verwendet:



In den geschwungenen Klammern wird über das & - Symbol angezeigt, dass es sich um ein Textsnippet handelt. Nach dem & - Symbol folgt dann der entsprechende Key.

Die Daten der angegebenen Prozessvariable können nach Belieben formatiert werden. Beispielsweise kann festgelegt werden, wieviele Nachkommastellen angezeigt werden sollen. Dazu muss neben dem Textsnippet der entsprechende Formatstring angefügt werden:



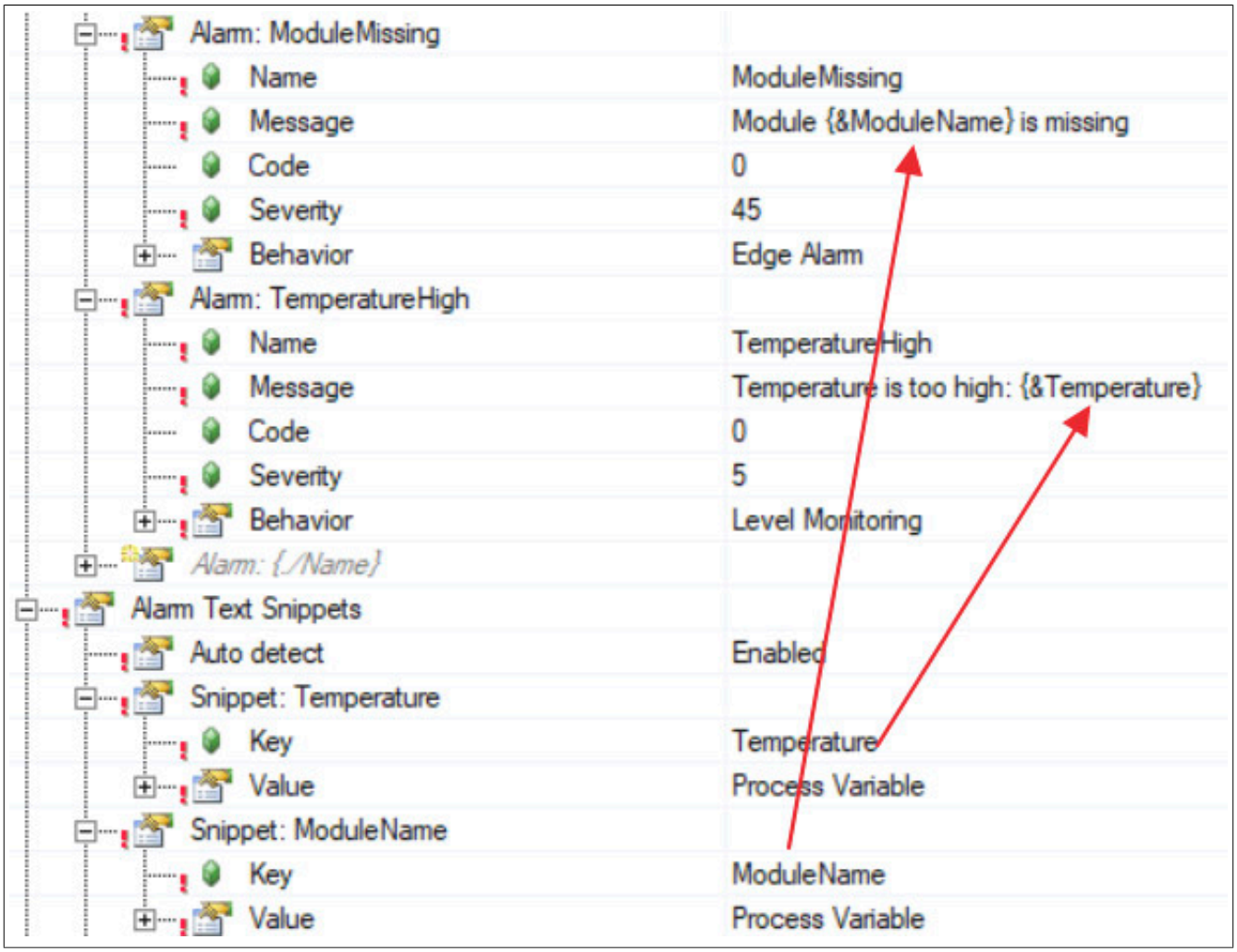
Mehr Details zu den damit einhergehenden Möglichkeiten sind im Abschnitt Formatstrings zu finden.

Folgende Datentypen können über ein Snippet in den Text eingebunden werden:

- **BOOL**
- **SINT**
- **INT**
- **DINT**
- **USINT**
- **UINT**
- **UDINT**
- **REAL**
- **LREAL**
- **STRING**
- **WSTRING**

Über die Eigenschaft "Auto detect" wird geprüft welche Alarmsnippets für welchen Alarm benötigt werden. Diese Option verbessert, vor allem bei vielen Alarmen und Alarmsnippets, Performance sowie Speicherverbrauch.

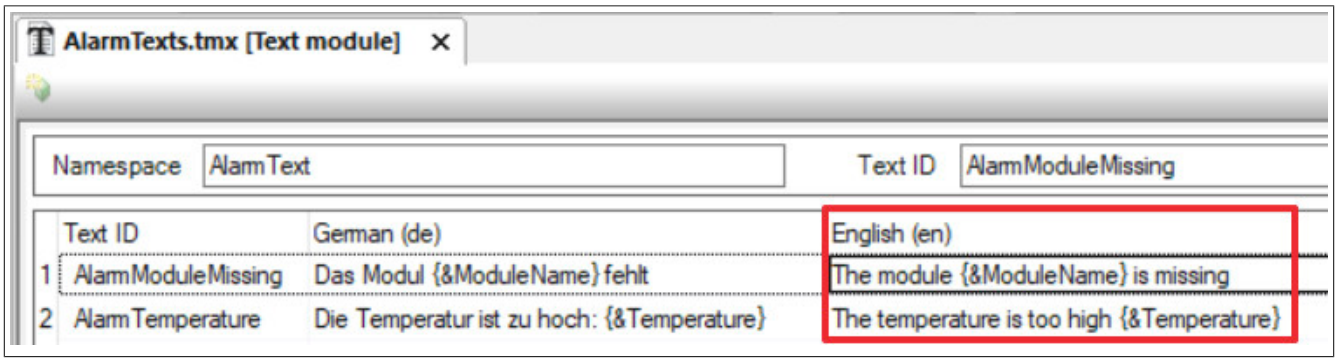
Bei "Auto detect = Enabled" wird automatisch geprüft welcher Alarm welche Snippets verwendet. Dadurch werden die Snippets nur an jene Alarme angehängt welche das Snippet auch wirklich brauchen.



Wird die Alarmnachricht lokalisiert über das Textsystem definiert, muss folgendes beachtet werden:

Werden die erweiterten Parameter in der Konfiguration aktiviert, kann "Auto detect" abhängig von der gewünschten Sprache genutzt werden. Diese Funktion kann nur genutzt werden wenn die Alarmnachricht lokalisiert angegeben wird. (Siehe Alarm Texte). Wird keine Sprache angegeben, so wird standardmäßig die in der Textkonfiguration angegebene Fallback-Sprache verwendet.

Das bedeutet, es wird in der gewünschten Sprache geprüft welche Alarme welche Snippets verwenden. Dadurch werden die Snippets wieder nur jenen Alarmen angehängt, welche diese auch wirklich benötigen.



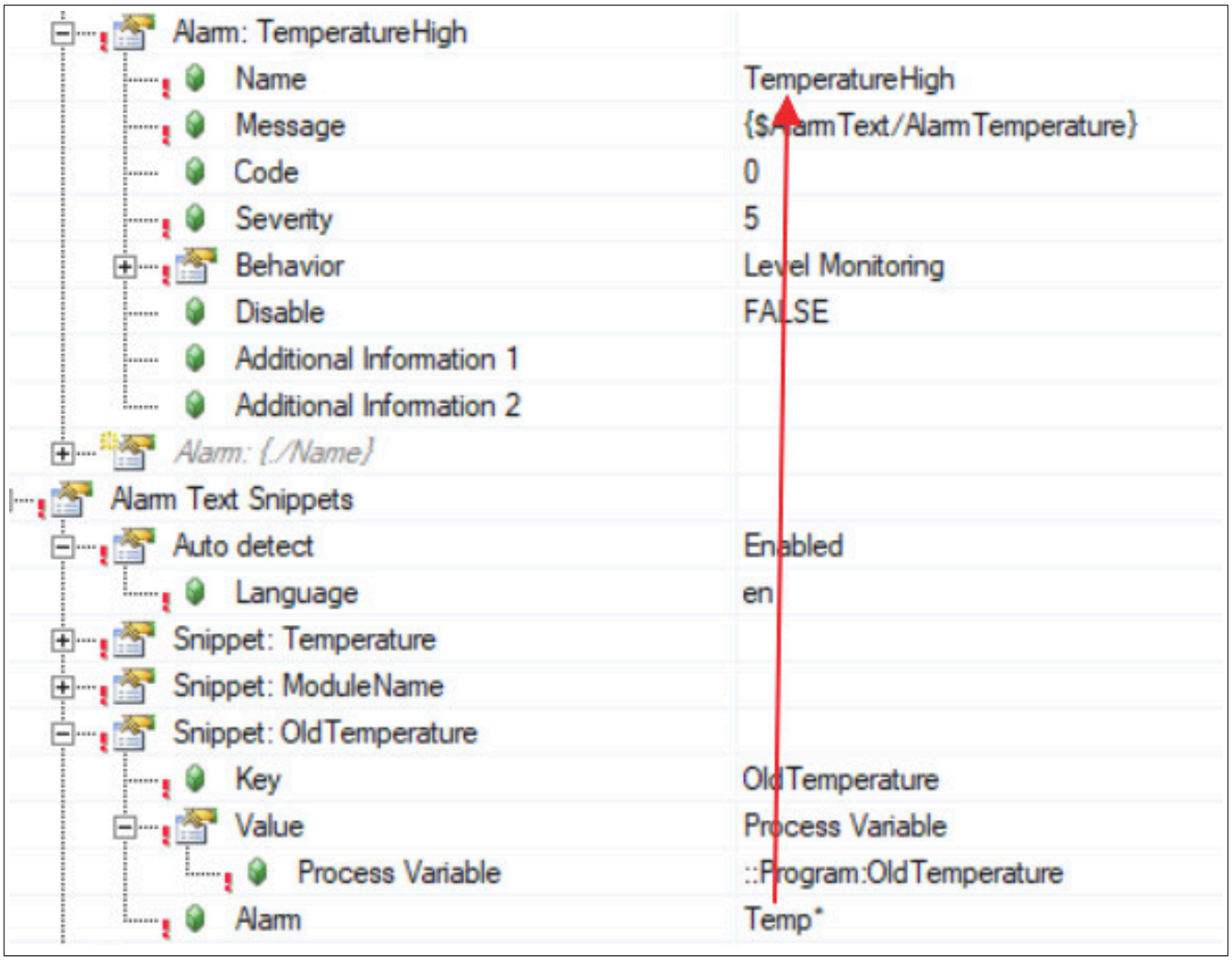
Bei der Verwendung von lokalisierten Alarmnachrichten kann es sein, dass für eine Sprache ein oder mehrere weitere Snippets hinzugefügt werden. Wird jetzt als zu prüfende Sprache Englisch verwendet, wo ein Snippet angegeben ist und in Deutsch wird noch ein weiteres Snippet verwendet, so wird das zweite Snippet nicht an den Alarm angehängt und nicht angezeigt:

AlarmTexts.tmx [Text module]*

Namespace: AlarmText | Text ID: New_Text_ID_1

	Text ID	German (de)	English (en)
1	AlarmModuleMissing	Das Modul {&ModuleName} fehlt	The module {&ModuleName} is missing
2	AlarmTemperature	Die Temperatur ist zu hoch: {&Temperature}, alte Temperatur: {&OldTemperature}	The temperature is too high {&Temperature}

Um solche Spezialfälle abzudecken kann bei den Alarmsnippets einzeln definiert werden zu welchen Alarmen das Snippet hinzugefügt werden soll. Das kann über die Eigenschaft "Alarm" erreicht werden. Bei der Angabe des Alarmnamens kann die Wildcard-Funktion verwendet werden. Das heißt mithilfe von "Temp*" wird das Snippet bei allen Alarmen angehängt welche mit "Temp" beginnen.



1.3.1.4 Abfragen

Zu diesem Thema gibt es auch den Anwendungsfall "In der Alarmliste nach Alarmen suchen"

Unter "Data Queries" können Abfragen erstellt werden. Diese Abfragen können verwendet werden, um nach bestimmten Alarmen in der Alarmliste zu filtern bzw. zu suchen. Dabei muss unter "Name" ein eindeutiger Name für die Abfrage angegeben werden. Soll die Anfrage über MpAlarmXQuery gestartet werden, so muss am Eingangsparameter "Name" der Name angegeben werden, welcher hier definiert wird.

Der Wert der Variable welche an "Update count" angegeben wird, wird automatisch um 1 erhöht sobald neue Information verfügbar ist. Das bedeutet sobald neue Daten, die den angegebenen Filter-Kriterien (WHERE) entsprechend, vorhanden sind, ändert sich der Zählerstand der Variable an "Update count". Dies kann als Trigger

verwendet werden, um den Befehl "Execute" an MpAlarmXQuery zu starten. Werden die erweiterten Parameter aktiviert, kann der "Update mode" definiert werden. Dabei kann bestimmt werden, ob nach jeder Änderung eines Alarms ("On every change") der Wert an "Update count" erhöht werden soll, oder nur wenn ein Alarm aktiviert oder deaktiviert ("On new/removed alarms only") wird.

Als Abfrage-Quelle ("Query source") kann bestimmt werden ob nach aktuellen Alarmen ("Pending alarms") oder nach allen konfigurierten Alarmen ("Configured alarms") gesucht werden soll. Welche Information die Abfrage enthalten soll, wird über "Select" und "Where" definiert:

SELECT

Unter "Select" wird bestimmt, welche Information erfragt werden sollen und wo diese hinkopiert werden sollen. An "Process variable" muss eine Variable angegeben werden. Wird die Abfrage über MpAlarmXQuery gestartet, so wird die Information an dieser Prozessvariable angezeigt. Es kann auch eine Array-Variable angegeben werden. Die Array-Variable muss folgendermaßen angegeben werden: "::MyVar[]"
"Column" bestimmt welche Information abgefragt werden soll. Für mehr Information zu den einzelnen Typen siehe Alarmeigenschaften.

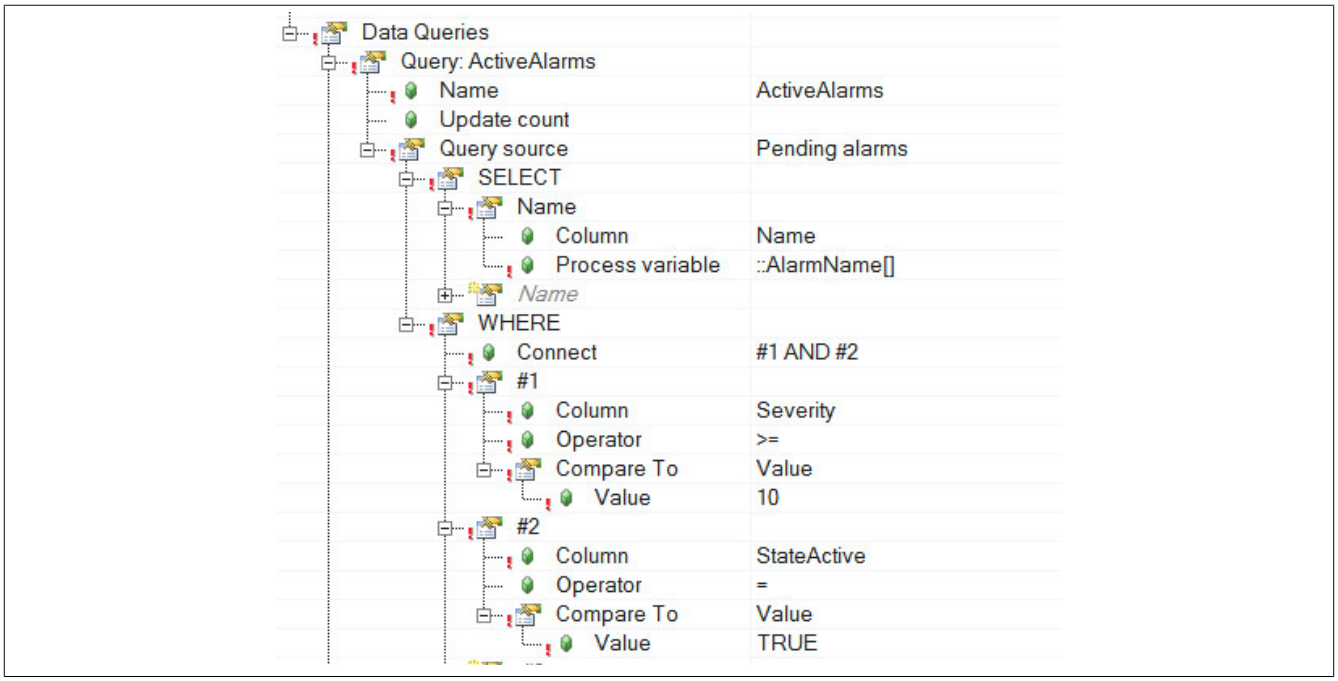
WHERE

Unter "Where" können die Filtereigenschaften weiter konfiguriert werden. Unter "Column" kann bestimmt werden nach welcher Information gefiltert werden soll. Für mehr Information zu den einzelnen Typen siehe Alarmeigenschaften.
Über "Operator" und "Compare To" kann die Anfrage mit zusätzlichen Filtern versehen werden, beispielsweise Alarme mit einem bestimmten Namen oder Alarme ab einem bestimten Zeitpunkt. Das bedeutet es kann zum Beispiel nach bestimmten Werten abgefragt werden. "Compare To" kann dabei ein fixer Wert ("Value") oder eine Prozessvariable ("PV") sein.
Wird als Vergleich ("Operator") "LIKE" verwendet, so kann nach ähnlichen Einträgen gesucht werden. Dabei muss das Platzhaltersymbol "*" verwendet werden. Wird als Wert ("Value"), zum Beispiel "*Test*" verwendet, können zum Beispiel nach den Einträgen "Test01", "MyTest" oder "MyTest01" gesucht werden.

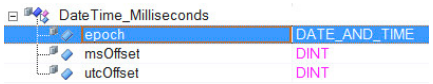
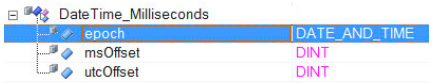
Beispiel

In der Alarmliste soll nach allen aktuell aktiven Alarmen gesucht werden. Das bedeutet als "Query source" wird "Pending alarms" ausgewählt. Es soll nach dem Alarmnamen gefiltert werden. Die Information soll an einer Variable angezeigt werden welche unter "Process Variable" angegeben wurde. Es wurde die Variable "AlarmName" angegeben welche vom Datentypen STRING[200][100] ist. Das heißt die Variable ist eine Array-Variable mit 100 Elementen. Da es sich um eine Array-Variable handelt, muss die Variable mit *::AlarmName[]* angegeben werden.

Als nächstes wurden weitere Filterkriterien unter "Where" definiert. Einmal soll der Schweregrad des Alarms >= 10 sein und der Alarm soll aktiv sein. Die Filteroptionen wurden unter "Connect" angegeben, dass heißt es wird nach Filterkriterium 1 und 2 gefiltert.



Wird für einen Spalten-Typen eine PV angegeben, so muss folgender Datentyp für die PV verwendet werden:

Spalten-Typ	Datentyp
Name	STRING
Message	WSTRING
InstanceID	STRING, WSTRING, UDINT
Code	STRING, WSTRING, UDINT
Severity	STRING, WSTRING, UDINT
Scope	STRING, WSTRING
StateActive	BOOL, STRING, WSTRING, UDINT, UINT, DINT, REAL
StateAcknowledged	BOOL, STRING, WSTRING, UDINT, UINT, DINT, REAL
TimeStamp	Um den Zeitstempel mit einer Sekunden-Genauigkeit zu erhalten, können folgende Datentypen verwendet werden: DATE, DATE_AND_TIME, UDINT Um den Zeitstempel mit einer Millisekunden-Genauigkeite zu erhalten, muss folgende Struktur verwendet werden:  • epoch: Liefert das Datum und die Zeit (in Sekunden-Genaugikeit) • msOffset: Liefert Millisekunden • utcOffset: Minuten zur UTC-Zeit. Wird in mapp AlarmX nicht verwendet und ist immer 0.
AdditionalInformation1	WSTRING
AdditionalInformation2	WSTRING
TimeStamp Acknowledge	Um den Zeitstempel der Quittierung mit einer Sekunden-Genauigkeit zu erhalten, können folgende Datentypen verwendet werden: DATE, DATE_AND_TIME, UDINT Um den Zeitstempel mit einer Millisekunden-Genauigkeite zu erhalten, muss folgende Struktur verwendet werden:  • epoch: Liefert das Datum und die Zeit (in Sekunden-Genaugikeit) • msOffset: Liefert Millisekunden • utcOffset: Minuten zur UTC-Zeit. Wird in mapp AlarmX nicht verwendet und ist immer 0.

1.3.1.5 Alarmkategorien

Im Abschnitt "Alarm Categories" können Alarme kategorisiert werden. Dabei wird die gewünschte Kategorie unter "Name" angegeben. Der Kategorie-Name darf nur einmal in der Konfiguration vorkommen. Es kann zum Beispiel die Kategorie "Information" oder "Warnung" verwendet werden. Die Kategorie wird über den Schweregrad ("Severity") oder den Code des Alarms definiert.

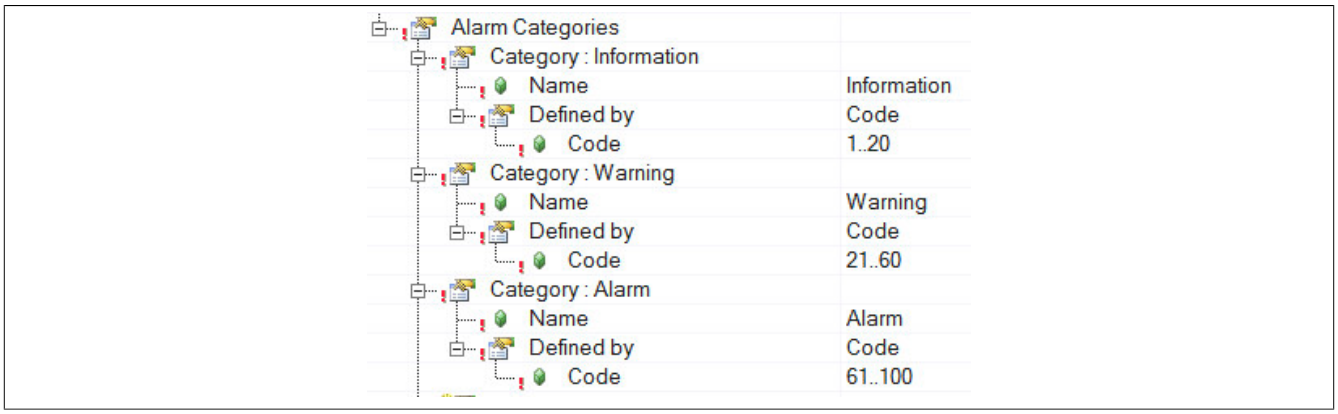
Die Kategorien werden über das AlarmList-Widget in einer mapp View Visualisierung grafisch dargestellt. Für mehr Information siehe AlarmList-Widget Konzept im Abschnitt "Kategorie anzeigen lassen". Die Angabe der Kategorie dient ausschließlich der visuellen Repräsentation der Alarme.

> **Kann ein Alarm mehreren Kategorien zugewiesen werden, so wird immer die Kategorie verwendet die als erstes zutrifft.**

Beispiel:

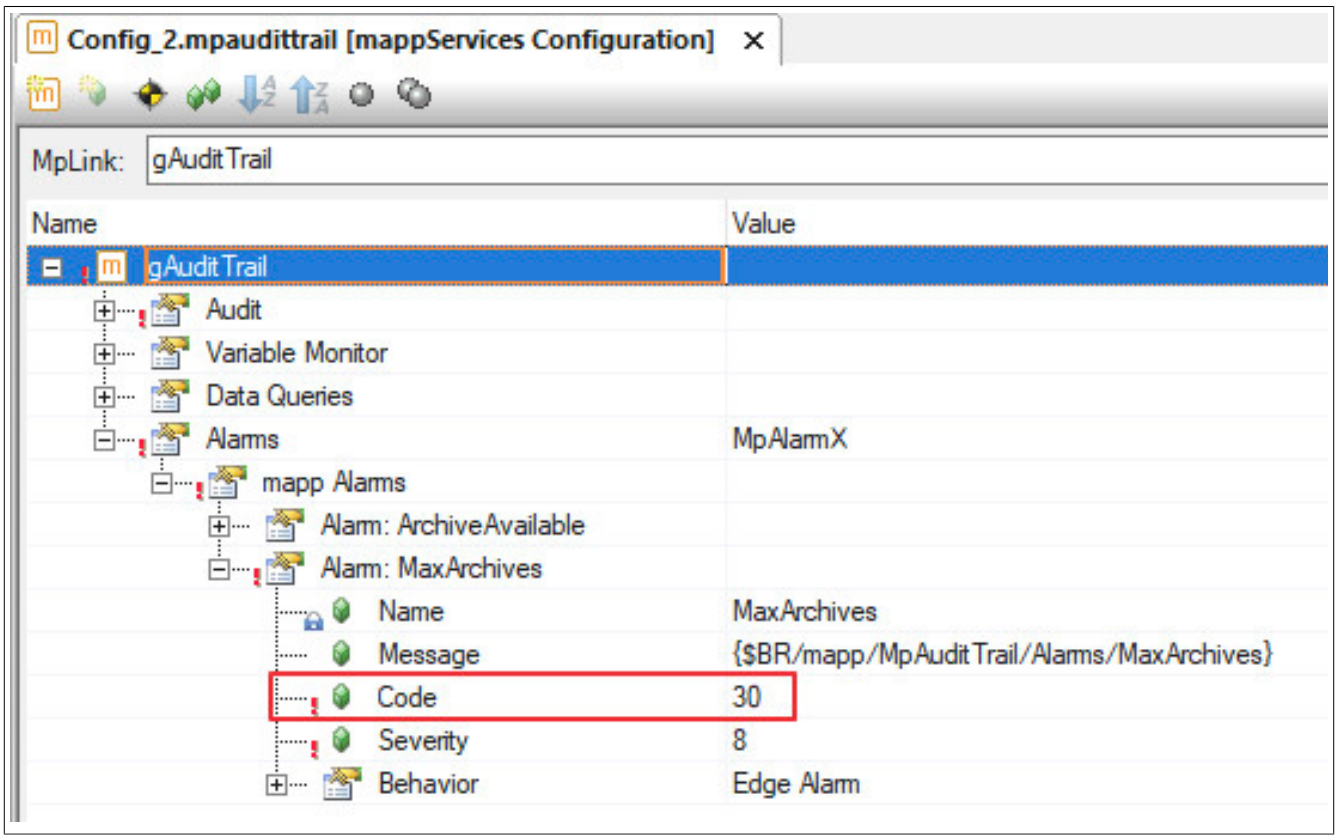
Es gibt 100 Benutzeralarme. Die Codes der Alarme gehen dabei von 1 bis 100. Die Codes von 1 bis 20 sollen als Information für den Benutzer dienen. Die Codes von 21 bis 60 sollen als Warnung angezeigt werden und die Codes von 61 bis 100 als Alarme.

Dafür werden 3 Alarmkategorien angegeben: "Information", "Warning" und "Alarm"



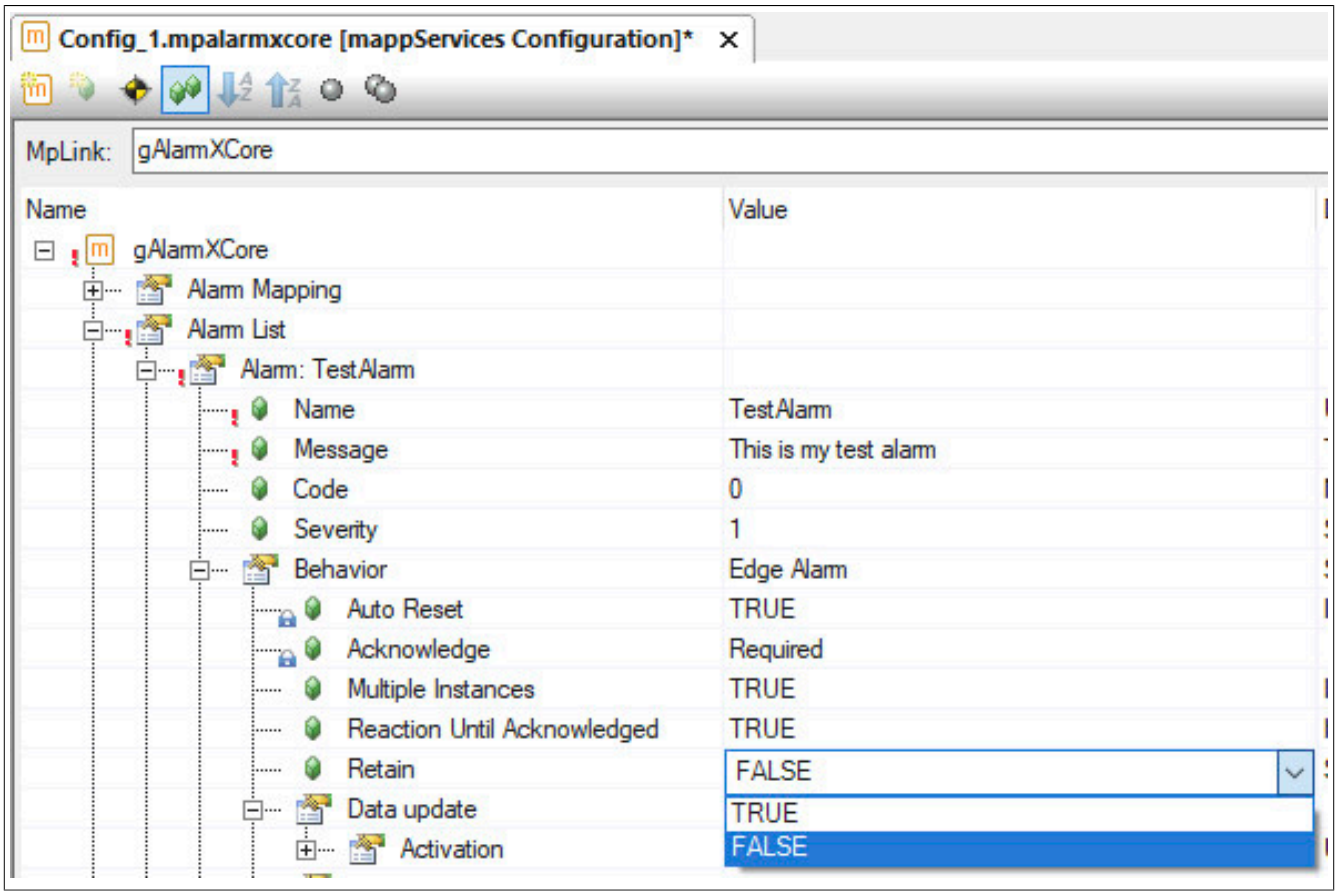
mapp Alarme

Auch Alarme von anderen mapp Komponenten können in der Alarmkategorie von mapp AlarmX berücksichtigt werden. Dabei muss nur der Schweregrad bzw. der Code der gewünschten mapp Komponente angepasst werden, damit dieser kategorisiert werden kann. In diesem Beispiel wurde in mapp Audit der Alarm so angepasst, dass dieser in der Kategorie "Warnung" angezeigt wird.

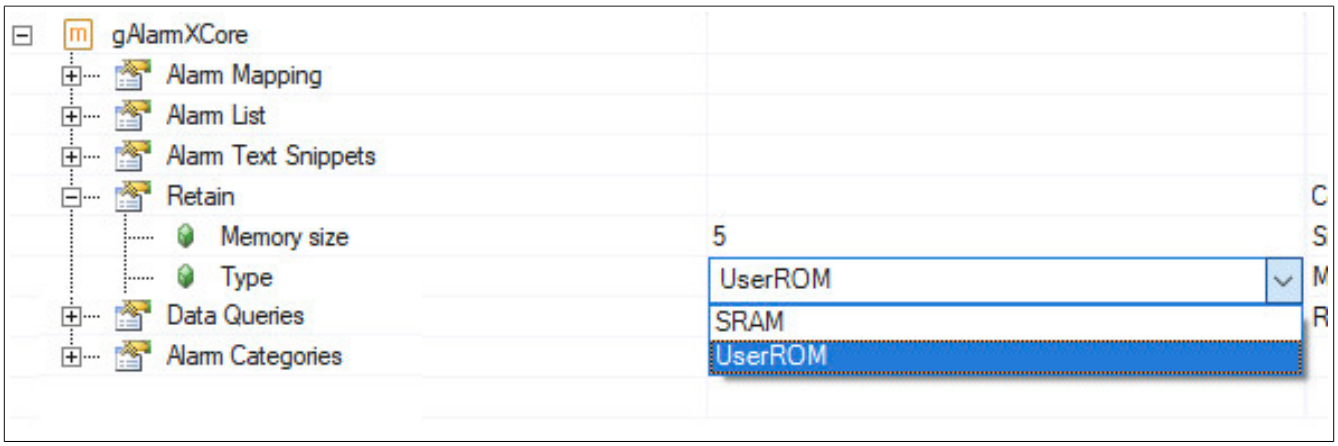


1.3.1.6 Retain Alarme

Werden die erweiterten Parameter der Konfiguration aktiviert, so können die Retain-Alarme konfiguriert werden. Über diese Funktion kann die gesamte Information des Alarms (Alarmzustand, zusätzliche Informationen, Alarmsnippets, konfigurierte Reaktionen usw.) nach einem Neustart der Steuerung wiederhergestellt werden. Damit ein Alarm als Retain-Alarm behandelt wird, muss in der Alarmliste der Parameter "Retain" auf `TRUE` gesetzt werden.



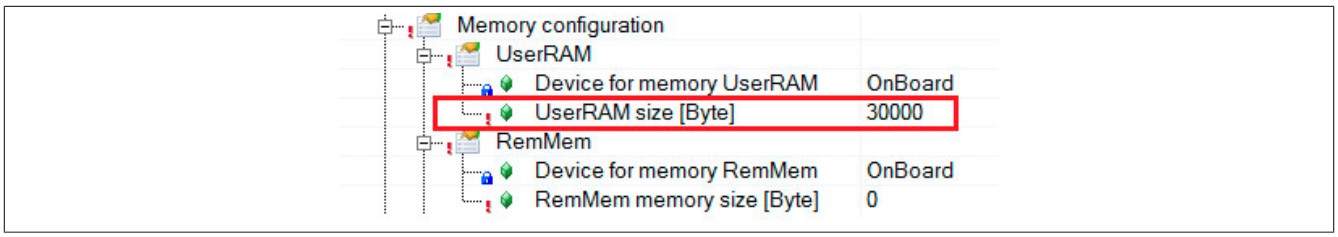
Die Information des Retain-Alarms wird zeitgleich mit der Historien-Information des Alarms in einem Datenobjekt abgespeichert. Über "Memory-Size" kann die Größe des Retain-Datenobjekts festgelegt werden. Unter "Type" kann bestimmt werden, ob die Alarminformationen im UserROM oder im SRAM des Zielsystems abgespeichert werden sollen.



Dabei gilt es folgendes zu beachten:

UserROM: Die gesammelten Daten werden zuerst in einem Datenobjekten im RAM zwischengespeichert. Anschließend werden diese stromausfall-sicher im UserROM gesichert.

SRAM: Bei der Verwendung von Batterie gepuffertem Speicher, muss in der Konfiguration der Steuerung genügend Speicher reserviert werden. Die gesammelten Daten werden in einem Datenobjekt zwischengespeichert und anschließend im SRAM gesichert. Kann nur verwendet werden, wenn benützte Hardware einen SRAM Speicher besitzt.



Pro Alarm werden ca. 26 Byte benötigt. Die Gößе setzt sich dabei aus dem Alarmnamen, Alarmzustand, konfigurierten Aktionen, zusätzlichen Information usw. zusammen. Je länger der Alarmname und beispielsweise die verwendeten Alarmsnippets sind, desto größer wird auch der benötigte Speicher für einen Alarm. Ist der konfigurierte Speicher zu klein, so wird der Fehler "Speichern fehlgeschlagen" ausgelöst. Dieser Fehler wird im Logger angezeigt.

Nach einem Neustart der Steuerung und Aktivierung des Funktionsbausteins MpAlarmXCore werden die abgespeicherten Alarminformationen wiederhergestellt. Das heißt definierte Aktionen (siehe Abschnitt "Mapping von Alarmen" weiter oben) werden, gegebenfall wieder aktiviert. Es gilt die Hinweise im unteren Abschnitt "Retain-Alarme in Kombination mit gemappten Alarmen" zu beachten. Diese Widerherstellung des Alarms wird nicht in der Alarmhistorie vermerkt.

Retain-Alarme in Kombination mit gemappten Alarmen

Abhängig vom ausgewählten Mapping-Typen verhalten sich Retain-Alarme folgendermaßen:

- **Reaction, Send Message, Remain, Escalate Reaction:** Wurde der Alarm als Retain-Alarm definiert, so werden auch die Mapping-Typen "Reaction", "Send Message", "Remain" und "Escalate Reaction" wieder ausgeführt. Das bedeutet es wird beispielsweise wieder eine Reaktion ausgelöst sowie eine Alarmnachricht versendet, sobald der betroffene Alarm wiederhergestellt wurde und vor dem Neustart aktiv war.
- **Escalate Alarm, Replace Alarm, Aggregate Alarm and Escalate, Aggregate Alarm locally:** Bei der Verwendung dieser Typen gibt es immer einen Ursprungs-Alarm (Parent-Alarm) sowie einen Alarm der beispielsweise weitergeleitet wird oder ersetzt wird (Child-Alarm). Abhängig davon ob der Parent-Alarm oder der Child-Alarm ein Retain-Alarm ist, gilt es folgendes zu beachten:

Parent-Alarm	Child-Alarm	Verhalten
Retain-Alarm	Retain-Alarm	Parent-, sowie Child-Alarm wird wiederhergestellt und definierte Aktion, wie beispielsweise "Replace Alarm" oder "Aggregate Alarm locally" wird wieder ausgeführt.
Retain-Alarm	Kein Retain-Alarm	In diesem Fall wird nur der Parent-Alarm nach einem Neustart der Steuerung wiederhergestellt. Der Child-Alarm wird nicht wiederhergestellt.
Kein Retain-Alarm	Retain-Alarm	Der Child-Alarm wird mit dem gleichen Status wie vor dem Neustart der Steuerung wiederhergestellt. Dadurch wird automatisch der Parent-Alarm auch wiederhergestellt, da der Child-Alarm nur in Verbindung mit dem Parent-Alarm existieren kann.
Kein Retain-Alarm	Kein Retain-Alarm	Alarm sowie definierte Aktion wird nicht wiederhergestellt.

1.3.2 MpAlarmXHistory Konfiguration

> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein de-, und wieder aktiviert werden.

Die geänderten Konfigurationsparameter werden sofort von der Komponente übernommen.

Export

Im Abschnitt "Export" können Einstellungen für den Export definiert werden. Unter "File Name Pattern" kann der Name der abzuspeichernden Datei bestimmt werden. "Timestamp Pattern" definiert den Zeitstempel für die Alarmhistorie.

Für mehr Information siehe "Einfügen der Zeit im Eintrag" in Definieren des Ereignistexts.

Durch das aktivieren der erweiterten Parameter kann die Option "Overwrite file" verwendet werden. Bei "Overwrite file = `TRUE`" wird, wenn eine Datei den gleichen Namen hat, die Datei überschrieben. Bei `FALSE` wird eine neue Datei erstellt welche die Endung "_n", wie zum Beispiel "xxx_2", hat.

Speicherort

Der Parameter "Memory" bestimmt, auf welche Weise die aufgezeichneten Daten abgespeichert werden. Ein Eintrag in der Alarmhistorie braucht ca. 350 Byte. Die Größe ist abhängig davon wie viel Alarminformation gespeichert werden muss. Wird beispielsweise ein Alarmsnippet verwendet erhöht dies den Bedarf. Standardmäßig beträgt die "Recording size" 200kB. Bei einem Alarm-Eintrag von ca. 350 Byte könnten ungefähr 570 Einträge erstellt werden, bis das der angegebene Buffer voll ist.

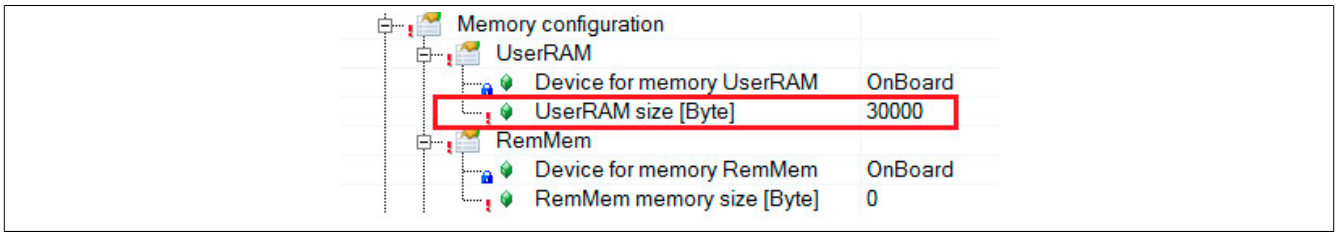
Folgende Modi können ausgewählt werden:

Temporary: Die Daten werden auf dem DRAM gespeichert. Es gilt zu beachten, dass die Daten bei einem Stromausfall oder bei einem Neustart nicht gesichert sind!

UserROM: Die gesammelten Daten werden zuerst in einem Datenobjekten im RAM zwischengespeichert. Anschließend werden diese stromausfall-sicher im UserROM gesichert.

> **Bei der Verwendung von UserROM wird bei jedem Schreibvorgang die Compact Flash Karte belastet. Treten innerhalb kürzester Zeit sehr viele Ereignisse auf, so kann die Karte davon beeinträchtigt werden. Um dem entgegenzuwirken kann alternativ der Modi "Buffered UserRom" verwendet werden. In diesem Fall wird der Inhalt zuerst zwischengespeichert und im Anschluss übertragen.**

SRAM: Bei der Verwendung von Batterie gepuffertem Speicher, muss in der Konfiguration der Steuerung genügend Speicher reserviert werden. Die gesammelten Daten werden in einem Datenobjekt zwischengespeichert und anschließend im SRAM gesichert. Kann nur verwendet werden, wenn benützte Hardware einen SRAM Speicher besitzt.



Buffered UserROM: Wird die Speicherart Buffered UserROM verwendet so werden die Daten zuerst im SRAM in einem Datenobjekt zwischengespeichert, danach werden die Daten auf das UserROM übertragen.

Auditing

Unter "Auditing" kann festgelegt werden, ob Ereignisse welche im Alarmmanagement ausgelöst wurden, von MpAudit aufgezeichnet werden sollen. Für mehr Information siehe Auditing der Alarmverwaltung.

Scope

Mithilfe von "Scope" wird der Umfang der Alarmhistorie bestimmt. Wurde eine mapp Hierarchie erstellt, so kann mit "Component" die Alarmhistorie für Komponenten auf gleicher Ebene und mit "Branch" die Alarme innerhalb eines Branches gesammelt werden.

Abfrage

Zu diesem Thema gibt es auch den Anwendungsfall "In der Alarmliste nach Alarmen suchen".

Unter "Data Queries" können Abfragen erstellt werden. Diese Abfragen können verwendet werden, um nach bestimmten Alarmen in der Alarmhistorie zu filtern bzw. zu suchen. Dabei muss unter "Name" ein eindeutiger Name für die Abfrage angegeben werden. Soll die Anfrage über MpAlarmXQuery gestartet werden, so muss am Eingangsparameter "Name" der Name angegeben werden, welcher hier definiert wird.

Der Wert der Variable welche an "Update count" angegeben wird, wird automatisch um 1 erhöht sobald neue Information verfügbar ist. Das bedeutet sobald neue Daten, die den angegebenen Filter-Kriterien (WHERE) entsprechend, vorhanden sind, ändert sich der Zählerstand der Variable an "Update count". Dies kann als Trigger verwendet werden, um den Befehl "Execute" an MpAlarmXQuery zu starten.

Welche Information die Abfrage enthalten soll, wird über "Select" und "Where" definiert:

SELECT

Unter "Select" wird bestimmt, welche Information erfragt werden sollen und wo diese hinkopiert werden sollen. An "Process variable" muss eine Variable angegeben werden. Wird die Abfrage über MpAlarmXQuery gestartet, so wird die Information an dieser Prozessvariable angezeigt. Es kann auch eine Array-Variable angegeben werden. Die Array-Variable muss folgendermaßen angegeben werden: "::MyVar[]"
"Column" bestimmt welche Information abgefragt werden soll. Für mehr Information zu den einzelnen Typen siehe Alarmeigenschaften. "StateChange" gibt die Zustands-Änderung an. Diese kann "Set", "Reset" oder "Acknowledged" sein. Es muss beachtet werden, dass die Zustands-Änderung in Form einer Zahl zurückgegeben wird. Dabei werden die Zahlen 2("Set"), 3 ("Reset") oder 4("Acknowledge") zurückgegeben.

WHERE

Unter "Where" können die Filtereigenschaften weiter konfiguriert werden. Unter "Column" kann bestimmt werden nach welcher Information gefiltert werden soll. Für mehr Information zu den einzelnen Typen siehe Alarmeigenschaften. Wird nach "StateChange" gefiltert, so muss beachtet werden, dass die Zustands-Änderung in Form einer Zahl angegeben wird. Wie im oberen Punkt erwähnt, kann die Zustands-Änderung "Set", "Reset", "Acknowledged" oder "Unacknowledged" sein. Dabei werden die Zahlen 2 ("Set"), 3 ("Reset"), 4 ("Acknowledged") und 5 ("Unack-

nowledged") verwendet.
Über "Operator" und "Compare To" kann die Anfrage mit zusätzlichen Filtern versehen werden, beispielsweise Alarme mit einem bestimmten Namen oder Alarme ab einem bestimten Zeitpunkt. Das bedeutet es kann zum Beispiel nach bestimmten Werten abgefragt werden. "Compare To" kann dabei ein fixer Wert ("Value") oder eine Prozessvariable ("PV") sein.
Wird als Vergleich ("Operator") "LIKE" verwendet, so kann nach ähnlichen Einträgen gesucht werden. Dabei muss das Platzhaltersymbol "*" verwendet werden. Wird als Wert ("Value"), zum Beispiel "*Test*" verwendet, können zum Beispiel nach den Einträgen "Test01", "MyTest" oder "MyTest01" gesucht werden.

Beispiel

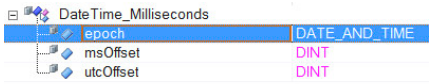
In der Alarmhistorie soll überprüft werden, ob der Alarm "TemperatureHigh" aktiviert ("Set") oder deaktiviert ("Reset") wurde. Es soll nach der Instanz-ID gefiltert werden. Die Information soll an einer Variable angezeigt werden, welche unter "Process Variable" definiert wurde. Es wurde die Variable "ID" angegeben, welche vom Datentypen STRING[200][10] ist. Das heißt die Variable ist eine Array-Variable mit 10 Elementen. Da es sich um eine Array-Variable handelt, muss die Variable mit ::ID[] angegeben werden.

Als nächstes wurden weitere Filterkriterien unter "Where" definiert. Einmal wird nach dem Alarm "TemperatureHigh" gesucht. Als nächstes wird nach der Zustands-Änderung "Set" (2) und "Reset" (3) gesucht.

Die Filteroptionen wurden unter "Connect" miteinander verbunden. Das heißt, es wird fix nach Filterkriterium 1 und Kriterium 2 oder 3 gefiltert.

Wird für einen Spalten-Typen eine PV angegeben, so muss folgender Datentyp für die PV verwendet werden:

Spalten-Typ	Datentyp
Name	STRING
Message	WSTRING
InstanceID	STRING, WSTRING, UDINT
Code	STRING, WSTRING, UDINT
Severity	STRING, WSTRING, UDINT
Scope	STRING, WSTRING

Spalten-Typ	Datentyp
StateChange	STRING, WSTRING, UDINT
TimeStamp	Um den Zeitstempel mit einer Sekunden-Genauigkeit zu erhalten, können folgende Datentypen verwendet werden: DATE, DATE_AND_TIME, UDINT Um den Zeitstempel mit einer Millisekunden-Genauigkeite zu erhalten, muss folgende Struktur verwendet werden:  • epoch: Liefert das Datum und die Zeit (in Sekunden-Genaugikeit) • msOffset: Liefert Millisekunden • utcOffset: Minuten zur UTC-Zeit. Wird in mapp AlarmX nicht verwendet und ist immer 0.
AdditionalInformation1	WSTRING
AdditionalInformation2	WSTRING

1.4 Anwendungsfälle

In diesem Abschnitt werden verschiedene Anwedungsmöglichkeiten der MpAlarmX Komponenten ausgeführt.

1.4.1 Alarmverwaltung mit Visualisierung

Anforderung

Ein Förderband einer Maschine liefert Produkte zu einer Verpackungsstation. Die Maschine hat eine Alarmverwaltung welche alle Alarme des Förderbandes sammelt und diese auf einer Visualisierung darstellt.

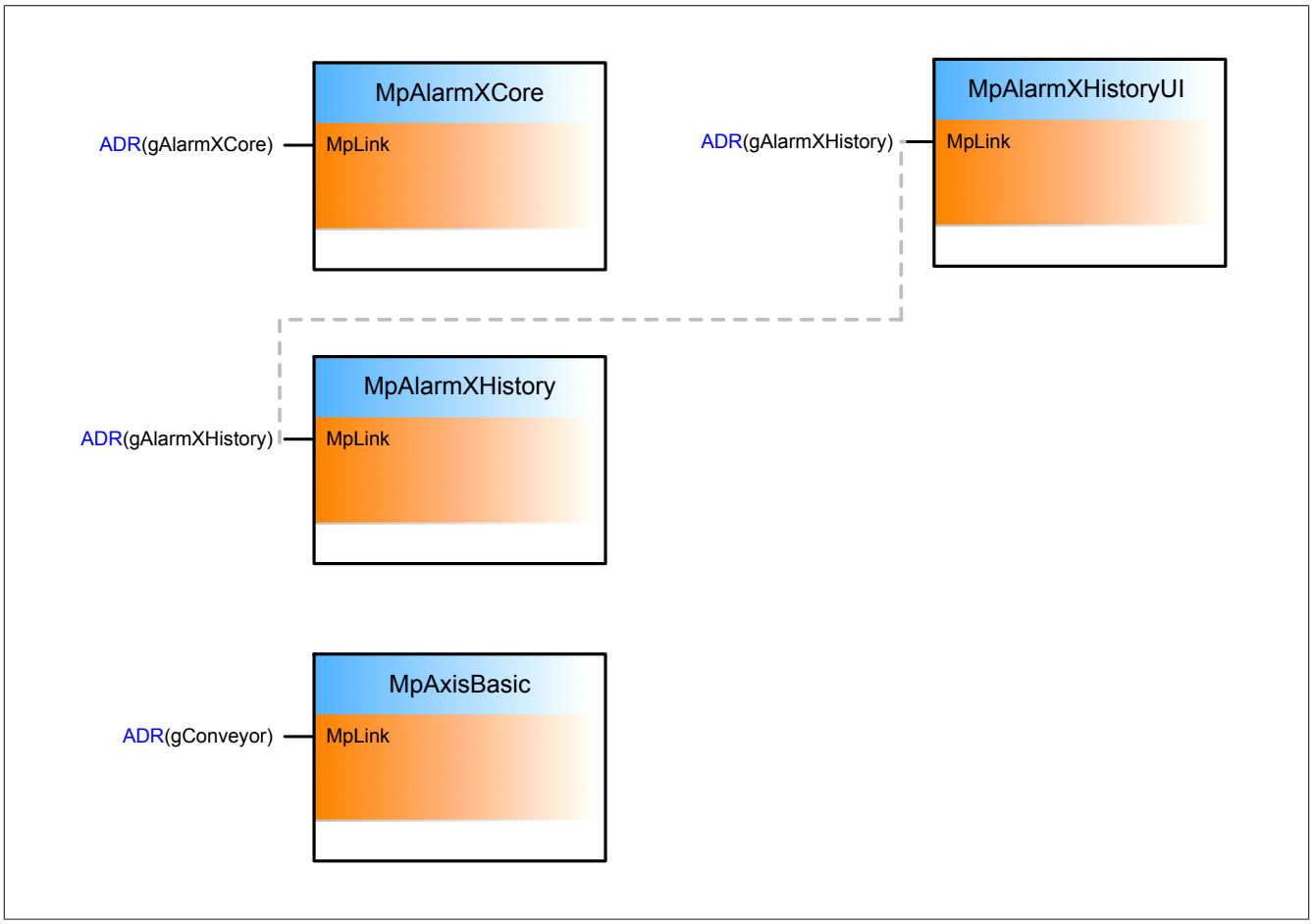
Es soll möglich sein die Alarme, welche auf der Visualisierung dargestellt werden, auf einen USB Stick abzuspeichern.

Lösung

Komponentenliste

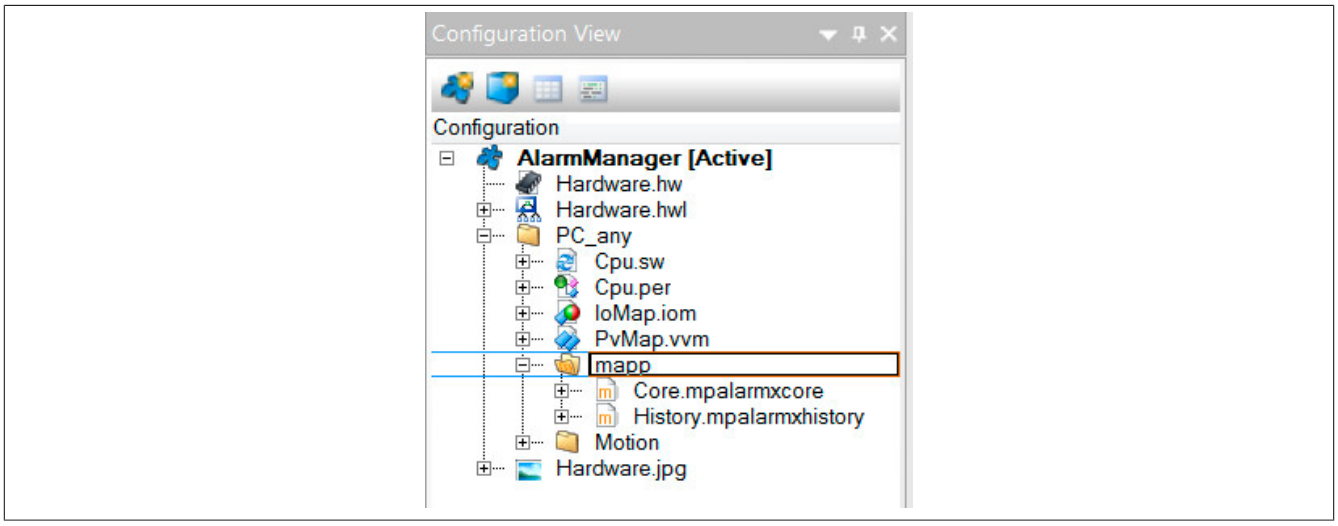
- ***MpAlarmXCore*** (Eigener MpLink): Sammelt alle Alarme der mapp Komponenten
- ***MpAlarmXHistory*** (Eigener MpLink): Speichert alle Alarme und deren Zustandsänderung
- ***MpAlarmXHistoryUI*** (MpLink von MpAlarmXHistory): Beinhaltet alle nötigen Parameter und Srukturen um eine Verbindung zu einer Visualisierung herzustellen und alle gespeicherten Alarme von MpAlarmXHistory anzuzeigen
- ***MpAxisBasic*** (Eigener MpLink): Achsensteuerung des Förderbandes

Verbindungsschema



Alarmkonfiguration

Für die Alarmverwaltung werden die Konfigurationen MpAlarmXCore und MpAlarmxHistory eingefügt.



Verwendung der mapp Komponenten

Es werden die Komponenten **MpAlarmXCore**, **MpAlarmXHistory**, **MpAlarmXHistoryUI** für die Alarmverwaltung und **MpAxisBasic** für die Steuerung des Förderbandes eingefügt.

Die Komponenten werden wie im Punkt Verbindungschema miteinander verbunden. **MpAxisBasic** muss sich in der gleichen Gruppe wie die Alarmfunktionsbausteine befinden.

Anschließend werden alle Komponenten parametriert. Um durch **MpAlarmXHistoryUI** eine Verbindung zu einer Visualisierung herzustellen siehe Abschnitt VC4 Vorlagen.

Für jede Achse gibt es Standardalarme, welcher der Benutzer nicht selbst konfigurieren muss, wie zum Beispiel Schleppfehler, Encoderfehler und viele mehr. Wird nun ein Fehler ausgelöst wird dieser auf der Visualisierung angezeigt. Durch das Kommando "MpAlarmXHistory.Export = *TRUE*" wird eine CSV Datei, welche die gesamte Alarm Historie beinhaltet, auf das angegebene Speichermedium exportiert. Das Speichermedium kann zum Beispiel ein USB Stick sein.

1.4.2 Benutzeralarmverwaltung mit Visualisierung

Anforderung

Ein Förderband einer Maschine transportiert PET. Das Förderband muss dabei durch eine Heizzone. Die Temperaturüberwachung der Heizzone ist in der Applikation implementiert. Es wird ein Alarm ausgelöst sobald die Temperatur in der Heizzone den oberen Grenzwert überschreitet.

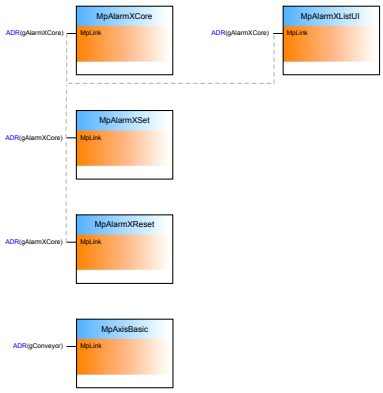
Alle Alarme sollen auf einer Visualisierung sichtbar sein.

Lösung

Komponentenliste

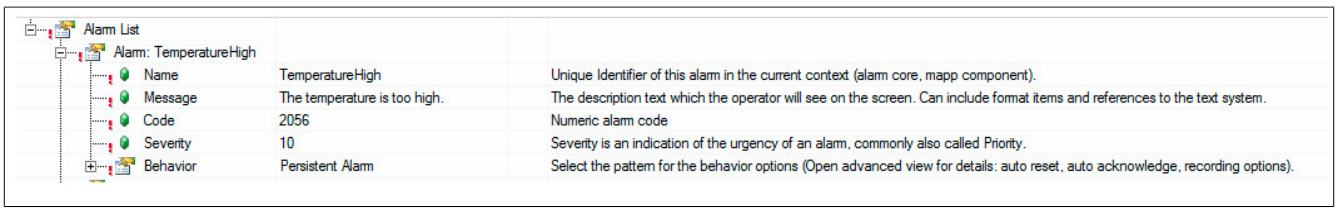
- ***MpAlarmXCore*** (Eigener MpLink): Sammelt alle Alarme der mapp Komponenten
- ***MpAlarmXSet*** (MpLink von MpAlarmXCore): Setzen eines Benutzeralarms
- ***MpAlarmXReset*** (MpLink von MpAlarmXCore): Rücksetzen eines Benutzeralarms
- ***MpAlarmXListUI*** (MpLink von MpAlarmXCore): Beinhaltet alle nötigen Parameter und Srukturen um eine Verbindung zu einer Visualisierung herzustellen und alle Alarme von MpAlarmXCore anzuzeigen
- ***MpAxisBasic*** (eigener MpLink): Achsensteuerung des Förderbandes

Verbindungsschema



Alarmkonfiguration

In der Konfiguration MpAlarmXCore können die Alarmeigenschaften editiert werden. Unter "Name" wird nun der Name des Benutzeralarms angegeben. Über diesen Namen wird der Alarm später in der Applikation identifiziert, wenn es um das Setzen, Zurücksetzen und quittieren geht. Der Alarmtext, welcher später auch in der Visualisierung angezeigt wird, wird unter "Message" eingefügt. "Code" und "Severity" bestimmen den Alarmcode und die Schwere des Alarms. Das Verhalten des Alarms wird unter "Behaviour" festgelegt. In unserem Fall handelt es sich um einen anhaltenden Alarm (Persistent), da der Alarm so lange aktiv sein soll, bis sich die Temperatur wieder im Normalbereich befindet.



Alarm List		
Alarm: TemperatureHigh		
Name	TemperatureHigh	Unique Identifier of this alarm in the current context (alarm core, mapp component).
Message	The temperature is too high.	The description text which the operator will see on the screen. Can include format items and references to the text system.
Code	2056	Numeric alarm code
Severity	10	Severity is an indication of the urgency of an alarm, commonly also called Priority.
Behavior	Persistent Alarm	Select the pattern for the behavior options (Open advanced view for details: auto reset, auto acknowledge, recording options).

Verwendung der mapp Komponenten

Es werden die Komponenten **MpAlarmXCore**, **MpAlarmXSet**, **MpAlarmXReset** und **MpAlarmXListUI** für die Alarmverwaltung und **MpAxisBasic** für die Steuerung des Förderbandes eingefügt.

Die Komponenten werden wie im Punkt Verbindungschema miteinander verbunden.

Anschließend werden alle Komponenten parametriert. Um durch **MpAlarmXListUI** eine Verbindung zu einer Visualisierung herzustellen siehe Abschnitt VC4 Vorlagen.

An den Komponenten **MpAlarmXSet** und **MpAlarmXReset** wird am Eingang "Name" der Name des Benutzeralarms angegeben (TemperatureHigh). Wird nun in der Heizzone der Alarm ausgelöst muss die Komponente **MpAlarmXSet** aufgerufen werden. Dadurch ist der Benutzeralarm aktiv und wird in der Visualisierung angezeigt.

Befindet sich die Temperatur der Heizzone wieder im Normalbereich, wird durch aufrufen der Komponente **MpAlarmXReset** der Alarm zurückgesetzt und ist somit nicht mehr aktiv.

1.4.3 Einheiten im Alarmtext anzeigen

Anforderung

In einer Heizschnecke wird, sobald der Temperaturwert zu hoch ansteigt, ein Alarm mit dem Text "Die Temperatur is zu hoch: xxx°C" ausgelöst. Es soll möglich sein, dass die Alarmnachricht in Deutsch oder Englisch angezeigt wird. Wird der Alarm in Deutsch angezeigt, soll das Metrische Einheitensystem verwendet werden (°C). Wir die Alarmnachricht in Englisch angezeigt, so soll das Imperiale Einheitensystem verwendet werden (°F).

Je nach Einheit soll die Temperatur automatisch umgerechnet werden.

Das Einheitensystem, sowie die Sprache werden über die Visualisierung umgeschalten. Es kann zwischen dem Metrischen, dem Imperial und dem Imperial-US System gewählt werden. Bei Verwendung des Metrischen Systems soll der Temperaturwert in Grad Celsius, beim Imperial bzw. Imperial-US soll Grad Fahrenheit angezeigt werden.

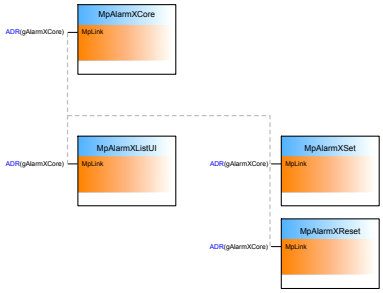
In der Software wird in Grad Celsius gearbeitet.

Lösung

Komponentenliste

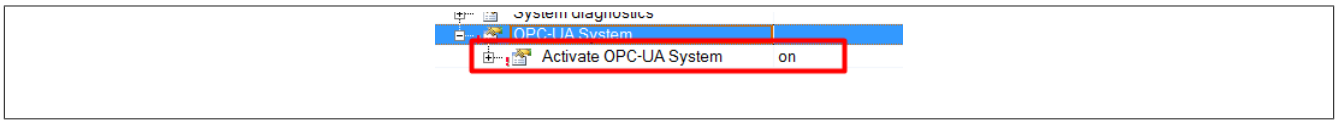
- ***MpAlarmXCore*** (Eigener MpLink): Sammelt alle Alarme der mapp Komponenten
- ***MpAlarmXSet*** (MpLink von MpAlarmXCore): Setzen eines Benutzeralarms
- ***MpAlarmXReset*** (MpLink von MpAlarmXCore): Rücksetzen eines Benutzeralarms
- ***MpAlarmXListUI*** (MpLink von MpAlarmXCore): Beinhaltet alle nötigen Parameter und Srukturen um eine Verbindung zu einer Visualisierung herzustellen und alle Alarme von MpAlarmXCore anzuzeigen

Verbindungsschema



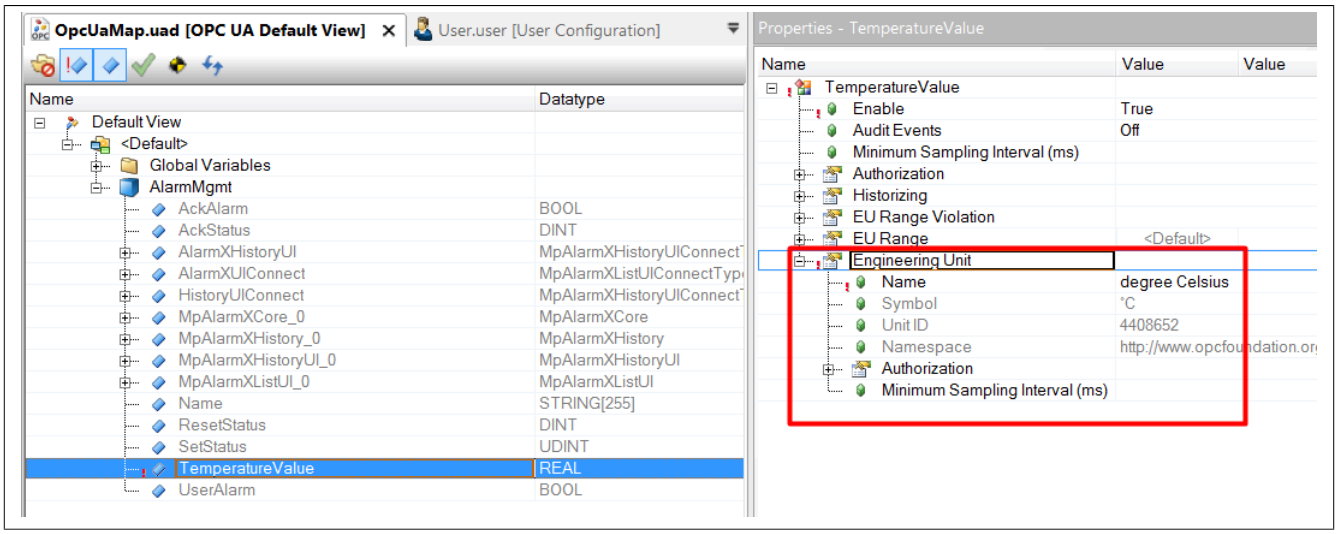
Vorbereitung zur Verwendung des Einheitensystems

Damit das Einheitensystem verwendet werden kann, muss der OPC-UA Server in der Konfiguration des Zielsystems aktiviert werden:

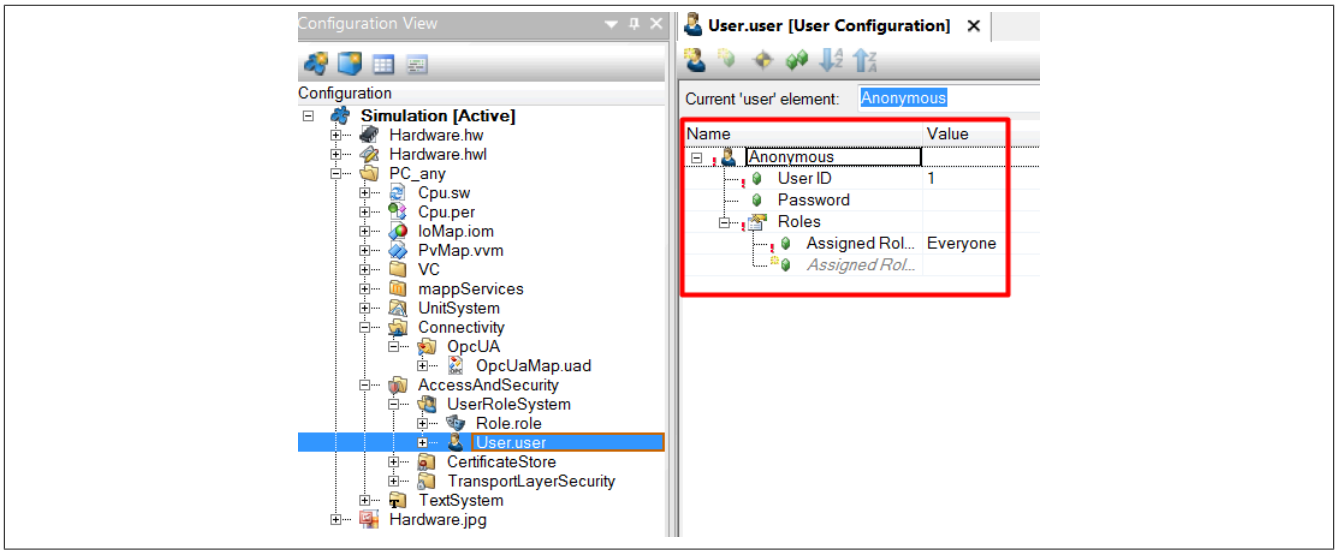


Als nächstes muss die OPC UA Default View Konfiguration eingefügt werden. In der Datei können die Variablen, welche im Projekt verwendet werden, als OPC-UA Variable freigeschaltet werden. Für mehr Information siehe Ansicht.

Die Variable *TemperatureValue* wird freigeschaltet und als Einheit wird degree Celsius verwendet, da in der Applikation mit **Grad Celsius** gearbeitet wird.



Um das Einheitensystem verwenden zu können, muss das Benutzer-Rollen-System eingefügt werden. Dem Benutzer Anonymous muss dabei die Rolle Everyone zugewiesen werden:

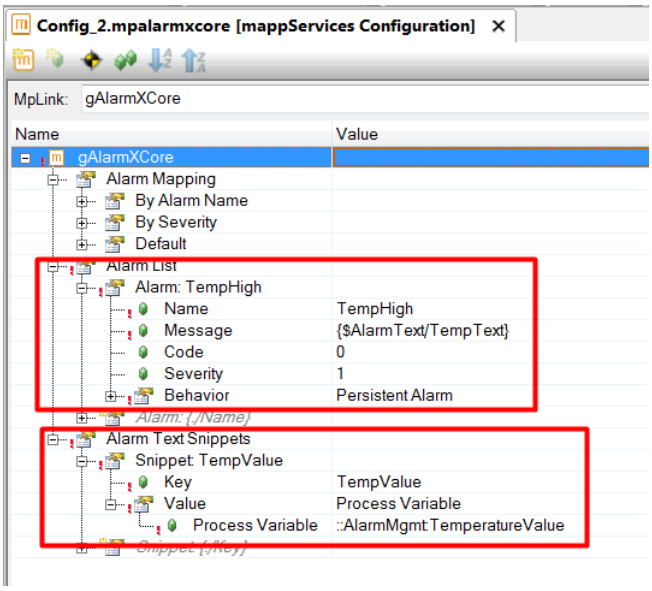


MpAlarmXCore Konfiguration

Um das Alarmsystem zu verwenden muss die Konfiguration MpAlarmXCore eingefügt werden. Es wird der Benutzeralarm "TempHigh" erstellt. Als Alarmverhalten wird "Persistent Alarm" ausgewählt.

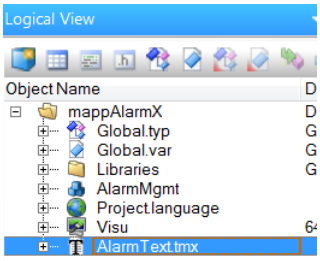
In der Alarmnachricht soll der Wert der Temperatur sowie die Einheit der Temperatur angezeigt werden. Um die Alarmnachricht sprachunabhängig zu konfigurieren, wird die Nachricht über das Textsystem angegeben. Für mehr Information siehe Abschnitt "Alarmtext lokalisiert über das Textsystem" in Alarm Texte.

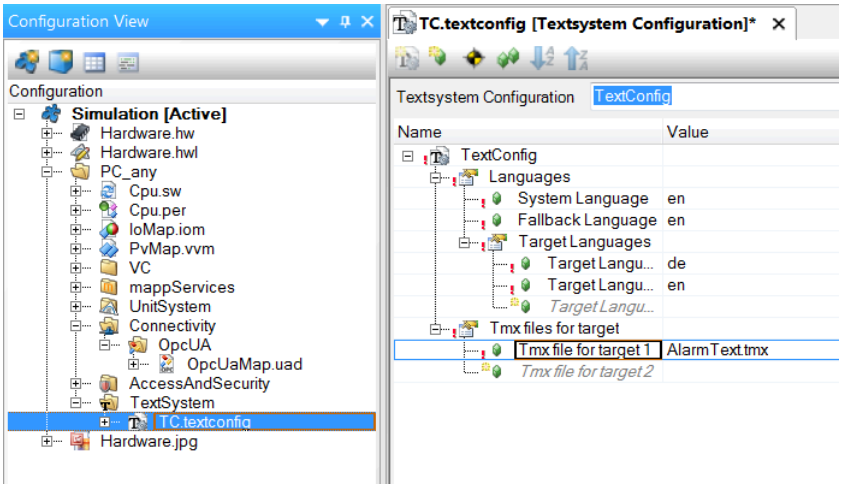
Um den Wert der Temperatur zu erhalten wird das Text Snippet "TempValue" erstellt. Damit der Wert in der Alarmnachricht angezeigt werden kann, wird dieser anschließen im Textsystem angegeben. Für mehr Information siehe Abschnitt "Einbinden von Applikationsdaten in den Alarmtext" in Alarm Texte.



Textsystem Konfiguration

Der Text, welcher für den Alarm ausgegeben wird, wird über das Textsystem definiert. Dazu wird eine .tmx Datei eingefügt, die zu verwendenden Projektsprachen, sowie die Konfiguration des Textsystems. In der Konfiguration muss die .tmx Datei unter "Tmx files for target" angegeben werden. Für mehr Information zum Textsystem siehe Textsystem.



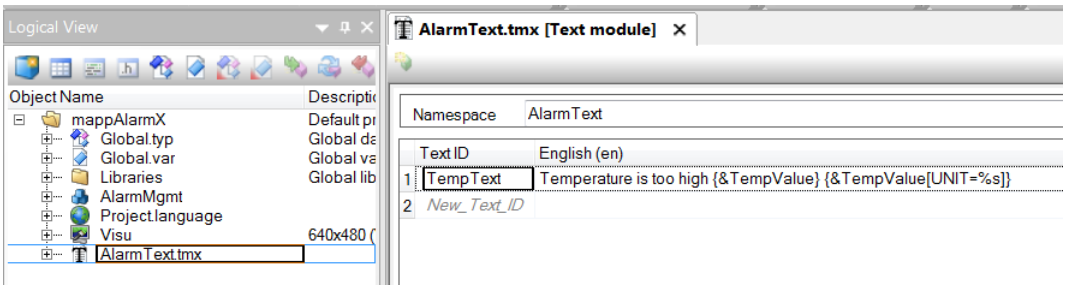


In der Textdatei muss der Namespace sowie die TextID definiert werden. Es wird, wie in der MpAlarmXCore Konfiguration definiert, der Namespace "AlarmText" und die Text ID "TempText" verwendet. Für mehr Information siehe Abschnitt "Alarmtext lokalisiert über das Textsystem" in Alarm Texte

Sobald der Alarm "TempHigh" aktiv ist, wird als Alarmnachricht der Text verwendet, welcher unter der TextID "TempText" angegeben ist.

Durch {&TempValue} wird der Wert der Temperaturvariable angezeigt. Mithilfe von [UNIT=%s] wird die Einheit (z.B. °C) der Variable angezeigt.

Für mehr Information zur Angabe der Einheit siehe Einheiten von Variablen anzeigen.



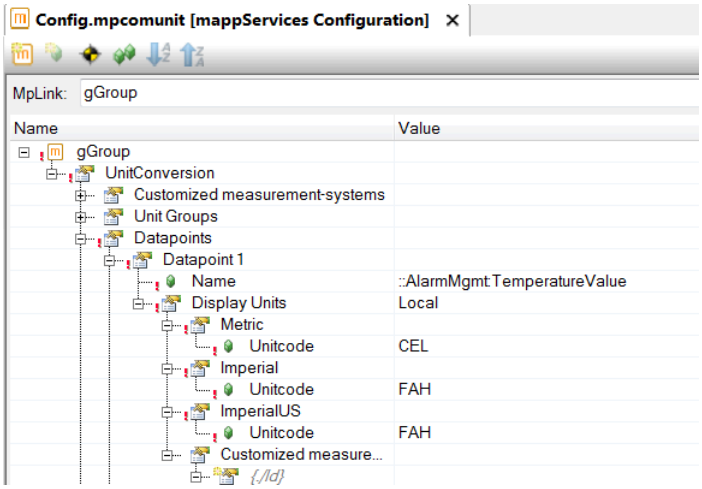
MpComUnit Konfiguration

Die Definition, welche Einheit in welchem Einheitensystem angezeigt werden soll, findet in der MpComUnit Konfiguration statt.

Unter "Datapoints" wird die aufzuzeichnende Variable *TemperatureValue* angegeben. Im Abschnitt "DisplayUnits" wird festgelegt welche Anzeigeeinheit pro Einheitensystem verwendet werden soll. Für mehr Information siehe Einheitenverwaltung.

Im Metrischen System soll Grad Celsius verwendet werden. Im Imperial und Imperial-US soll Grad Fahrenheit verwendet werden. Aus diesem Grund werden die Einheiten-Abkürzungen CEL bzw. FAH verwendet.

Welches Kürzel welche Einheit darstellt kann hier gefunden werden.



Verwendung der mapp Komponenten

Es wird die Komponente **MpAlarmXCore** für die Alarmverwaltung eingefügt.

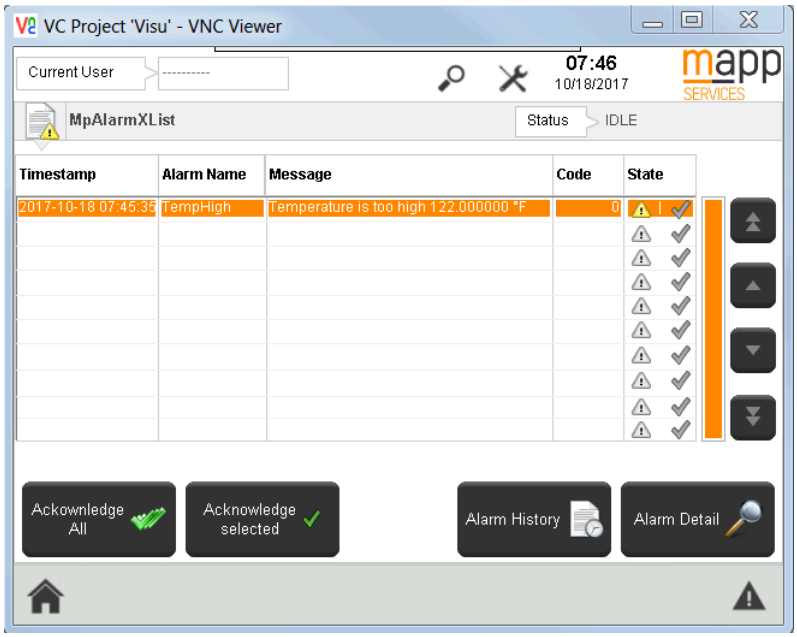
Für das Setzen und Rücksetzen des Benutzeralarms werden die Komponenten **MpAlarmXSet** und **MpAlarmXReset** verwendet.

Die Komponenten werden, wie im Punkt Verbindungschema, miteinander verbunden und anschließend parametriert.

Um durch **MpAlarmXListUI** eine Verbindung zu einer Visualisierung herzustellen siehe Abschnitt VC4 Vorlagen.

Um den Alarm in der Visualisierung in Deutsch oder in Englisch anzuzeigen, wird der Parameter "Language" welcher in der MpAlarmXListUIConnectType Struktur ist, verwendet.

Soll zum Beispiel die Alarmnachricht in Englisch und somit das Imperiale Einheitensystem verwendet werden, so muss an " Language='en|imperial' " angegeben werden. Für mehr Inforamtion siehe Einheiten von Variablen anzeigen.



1.4.4 Alarmliste nach Schweregrad filtern

Anforderung

In einer Maschine existieren mehrere Alarme welche nach Schweregrad unterteilt sind. Je nach Schweregrad handelt es sich um eine Information (10) eine Warnung (20) oder einen kritischen Alarm (30).

Für eine bessere Übersicht soll es möglich sein, in der Alarmliste nach aktiven Alarmen zu suchen, welche einen bestimmten Schweregrad besitzen. So kann schnell und einfach nach Informationen, Warnungen oder kritischen Alarmen gesucht werden.

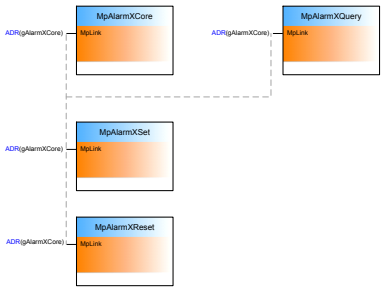
Die Alarme sollen applikativ gesucht werden, da die Alarminformation an einer 3rd party Visualisierungen angezeigt wird.

Lösung

Komponentenliste

- ***MpAlarmXCore*** (Eigener MpLink): Sammelt alle Alarme der mapp Komponenten
- ***MpAlarmXSet*** (MpLink von MpAlarmXCore): Setzen eines Benutzeralarms
- ***MpAlarmXReset*** (MpLink von MpAlarmXCore): Rücksetzen eines Benutzeralarms
- ***MpAlarmXQuery*** (MpLink von MpAlarmXCore): Startet Abfragen. Dadurch kann nach bestimmten Alarmen in der Alarmliste gesucht werden.

Verbindungsschema



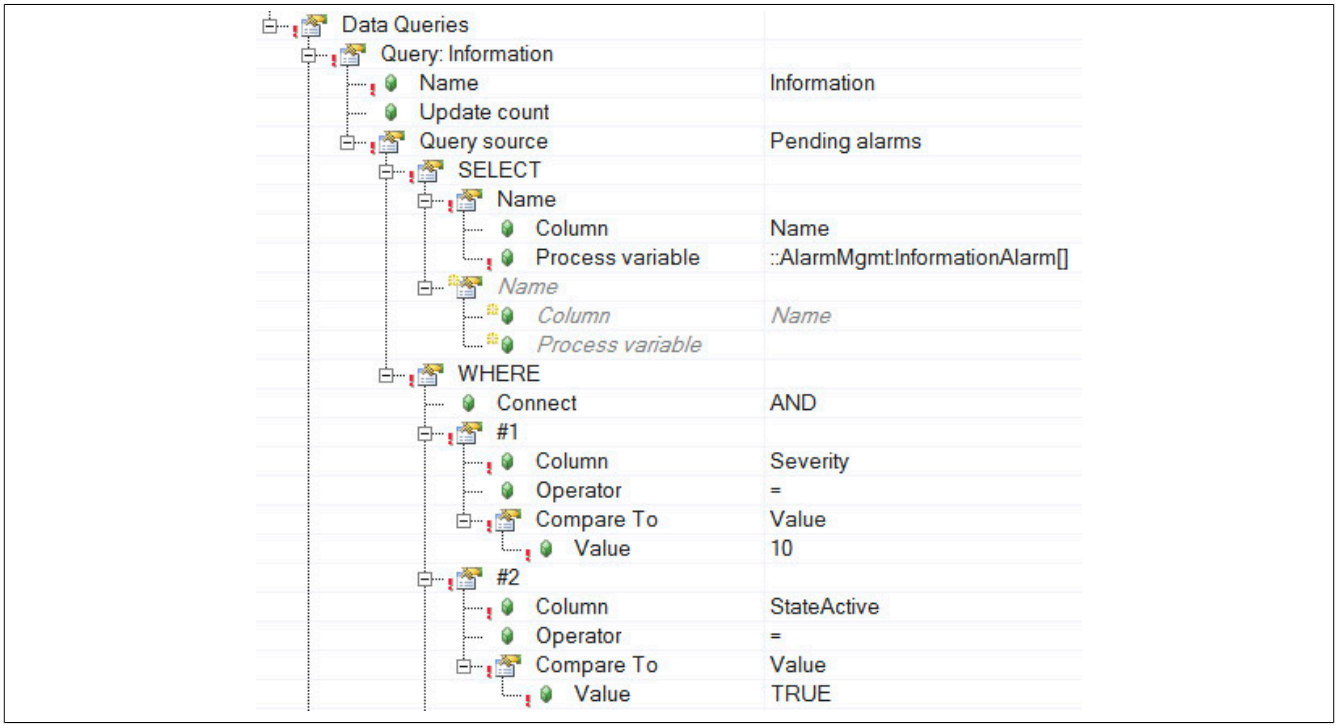
MpAlarmXCore Konfiguration

Um das Alarmsystem zu verwenden muss die MpAlarmXCore Konfiguration eingefügt werden. Es werden alle benötigten Benutzeralarme erstellt. Je nachdem um welche Art von Alarm es sich handeln soll (Information, Warnung, Alarm) wird den Alarmen der Schweregrad 10, 20 oder 30 zugewiesen.

Um später die Alarmliste nach Alarmen eines bestimmten Schweregrads zu suchen, werden Abfragen, sogenannte Queries erstellt.

Um die Alarmliste nach dem Informations-Alarmtypen abzusuchen, wird die Abfrage "Information" erstellt. Unter "Select" wird bestimmt, welche Information erfragt werden soll und wo diese hinkopiert werden soll. Es wird nach den Alarmnamen gesucht. Die ausgewählte Information wird beim Ausführen der Anfrage auf die Variable an "Process variable" kopiert. Es wird die Array-Variable "InformationAlarm" angegeben. Arrays müssen immer mit "[]" angegeben werden.

Unter "Where" können die Filtereigenschaften konfiguriert werden. Es soll nach aktiven Alarmen gesucht werden, deren Schweregrad 10 ist.



Die gleiche Art von Abfrage wird für den Alarmtypen "Warnung" und "Alarm" erstellt. Dabei wird jeweils der Schweregrad angepasst.

Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpAlarmXCore** für die Alarmverwaltung eingefügt.

Für das Setzen und Rücksetzen des Benutzeralarms werden die Funktionen **MpAlarmXSet** und **MpAlarmXReset** verwendet.

Die Funktionsbausteine werden, wie im Punkt Verbindungschema, miteinander verbunden und anschließend parametriert.

Um die in der MpAlarmXCore Konfiguration definierten Abfragen starten zu können, wird der Funktionsbaustein **MpAlarmXQuery** eingefügt. Als Modus wird "mpALARMX_QUERY_MODE_ALL" verwendet. Das bedeutet, es wird die gesamte Alarmliste durchsucht. Der Abfrage-Name wird über "Name" bestimmt. Es wird der in der MpAlarmXCore Konfiguration definierte Abfrage-Name, wie zum Beispiel "Infomation" angegeben.

Mit "Execute = `TRUE`" an **MpAlarmXQuery** wird die Abfrage gestartet. Das Ergebnis wird in der Variable angzeigt, welche für die Abfrage definiert wurde.

1.4.5 Alarmverwaltung in der Konfiguration

Anforderung

Das Kühlwasser für eine Anlage wird in einem Tank gelagert. Je nachdem ob der Wasserstand zu niedrig oder zu hoch ist, soll der Alarm "WaterTankLevel" ausgelöst werden.

Weiters kommuniziert die Anlage über TCP mit anderen Anlagen in einer Fabrik. Um sicherzustellen, dass die Kommunikation einwandfrei läuft, soll, sobald ein Fehler in der Kommunikation auftritt, der Alarm "CommunicationError" ausgelöst werden.

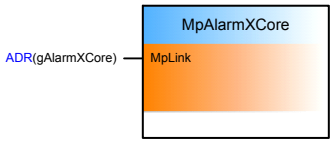
Um den applikativen Aufwand der Alarmbehandlung so gering wie möglich zu halten, sollen die Alarme über die MpAlarmXCore Konfiguration verwaltet werden.

Lösung

Komponentenliste

- ***MpAlarmXCore*** (Eigener MpLink): Sammelt alle Alarme der mapp Komponenten

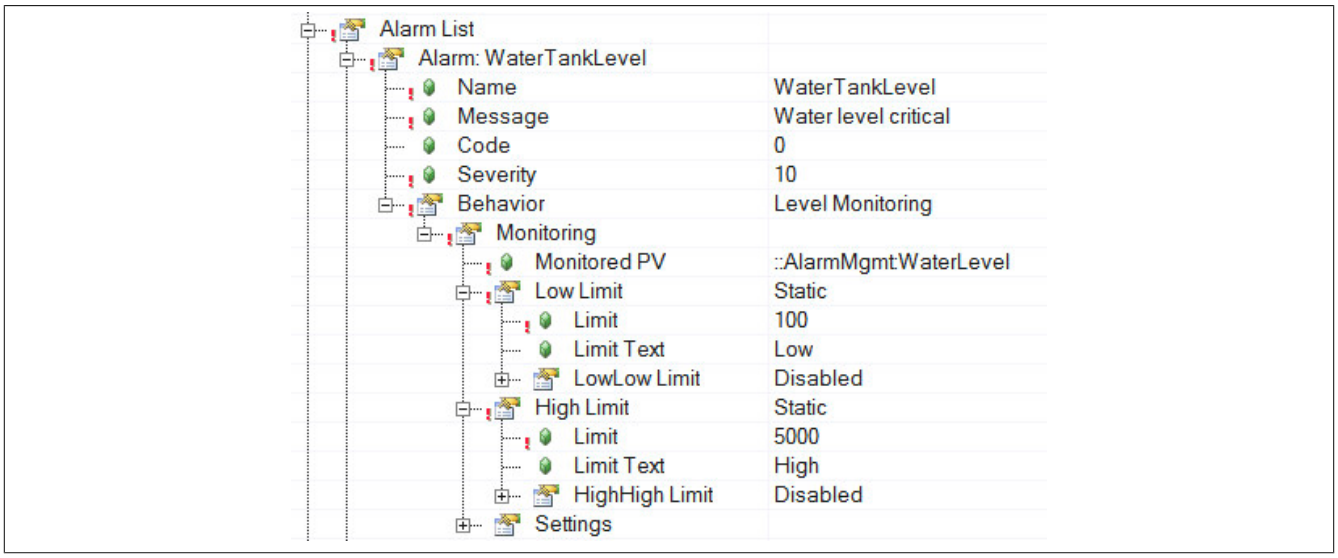
Verbindungsschema



MpAlarmXCore Konfiguration

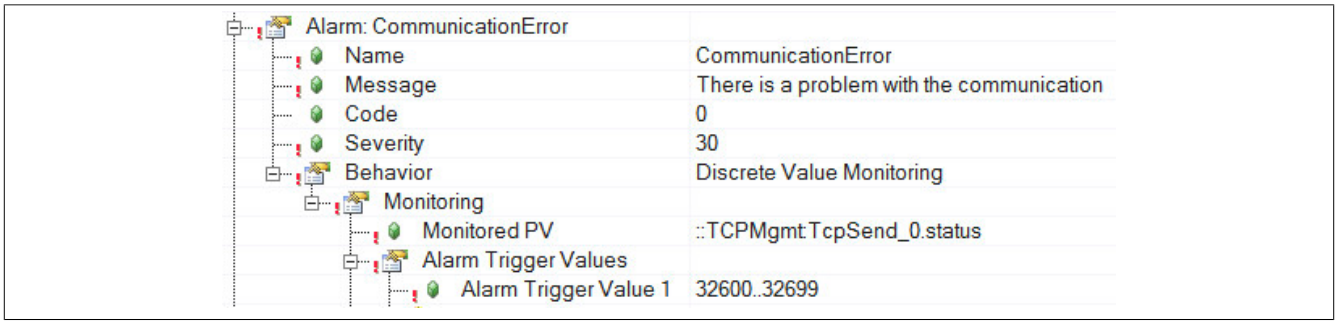
Um das Alarmsystem zu verwenden muss die MpAlarmXCore Konfiguration eingefügt werden.

Mithilfe des Alarm Monitoring können Benutzeralarme über die Konfiguration verwaltet werden. Für den Alarm "WaterTankLevel" wird als Monitoring-Typ "LevelMonitoring" verwendet. Dabei kann eine Prozessvariable angegeben werden, deren Limits automatisch kontrolliert werden. Der Füllstand des Tanks soll nicht unter 100l und nicht über 5000l gelangen. Aus diesem Grund wird "LowLimit" mit 100 und "HighLimit" mit 5000 festgelegt. Der Alarm wird automatisch ausgelöst, sobald eines der Limits unter-, bzw. überschritten wird.



Alarm List	
Alarm: WaterTankLevel	
Name	WaterTankLevel
Message	Water level critical
Code	0
Severity	10
Behavior	Level Monitoring
Monitoring	
Monitored PV	::AlarmMgmt:WaterLevel
Low Limit	Static
Limit	100
Limit Text	Low
LowLow Limit	Disabled
High Limit	Static
Limit	5000
Limit Text	High
HighHigh Limit	Disabled
Settings	

Um Daten über TCP an andere Stationen zu senden, wird die Funktion **TcpSend** verwendet. Je nachdem welchen Status die Funktion hat, ist die Kommunikation aktiv oder fehlerhaft. Um Alarme auszulösen sobald die Kommunikation fehlschlägt, wird der Status der Funktion über den Monitoring-Typ "DiscreteValueMonitoring" überwacht. Der Status der Funktion liefert Fehlernummern von 32600 bis 32699. Das beudeutet, sobald ein Fehler auftritt wird der Alarm "CommunicationError" automatisch ausgelöst.



Alarm: CommunicationError	
Name	CommunicationError
Message	There is a problem with the communication
Code	0
Severity	30
Behavior	Discrete Value Monitoring
Monitoring	
Monitored PV	::TCPMgmt:TcpSend_0.status
Alarm Trigger Values	
Alarm Trigger Value 1	32600..32699

Für mehr Information zum Alarm Monitoring siehe hier.

Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpAlarmXCore** für die Alarmverwaltung eingefügt.

Sobald das Wasserlevel unter 100l oder über 5000l ist, wird automatisch der Alarm "WaterTankLevel" angezeigt.

Tritt ein Fehler in der TCP-Kommunikation auf, so wird automatisch der Alarm "CommunicationError" angezeigt.

1.4.6 Zurücksetzen des Quittierstatus

Anforderung

An einer Maschine gibt es insgesamt 2 verschiedene Not-Aus Schalter. Sobald einer der Schalter betätigt wird, soll in der Maschinenvisualisierung der Alarm "EmergencyStop" angzeigt werden.

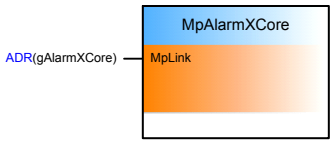
Wird der Alarm quittiert und der zweite Not-Aus Schalter wird betätigt, so soll der Quittierstatus von "EmergencyStop" zurückgesetzt werden. Dadurch weiß der Maschinenbediener, dass ein weiterer Not-Aus Schlater betätigt wurde. Der Alarm muss erneut quittiert und behoben werden, damit der Alarm nicht mehr in der Maschinenvisualisierung angzeigt wird.

Lösung

Komponentenliste

- ***MpAlarmXCore*** (Eigener MpLink): Sammelt alle Alarme der mapp Komponenten

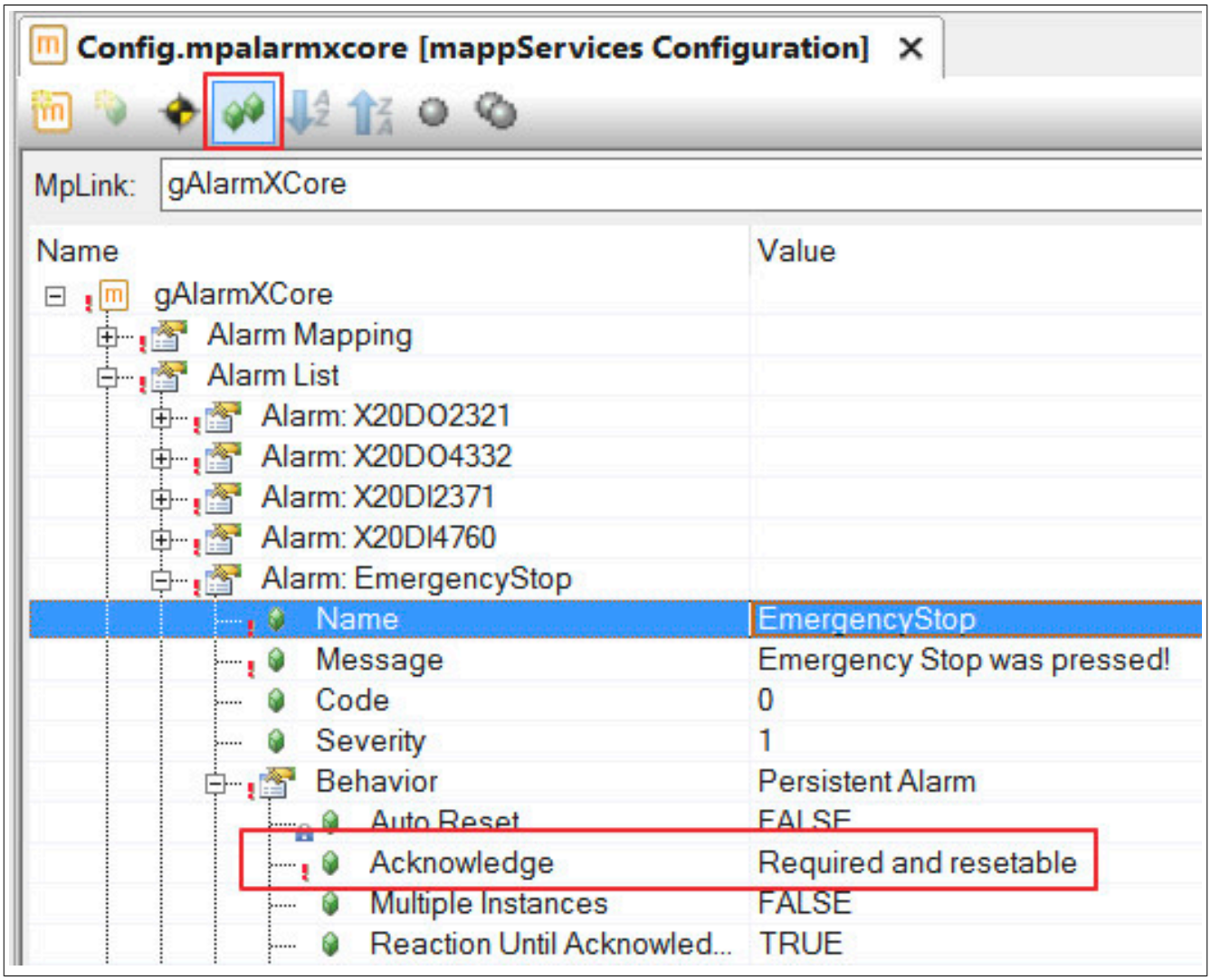
Verbindungsschema



MpAlarmXCore Konfiguration

Der Alarm "Emergencystop" wurde in der MpAlarmXCore Konfiguration definiert. Als Alarmverhalten wurde "Persistent Alarm" gewählt.

Damit der Quittierstatus von "EmergencyStop" zurückgesetzt werden kann, muss das Quittierverhalten auf "Required and resetable" gesetzt werden. Diese Option kann nur definiert werden, wenn die erweiterten Parameter der Konfiguration aktiviert wurden.



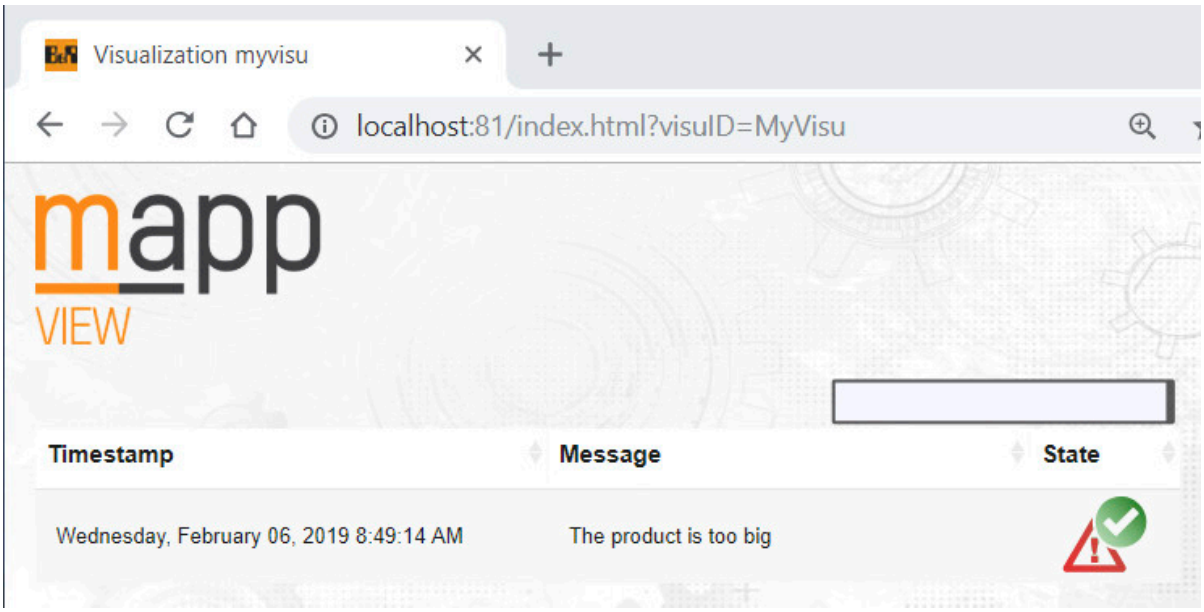
Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpAlarmXCore** für die Alarmverwaltung eingefügt.

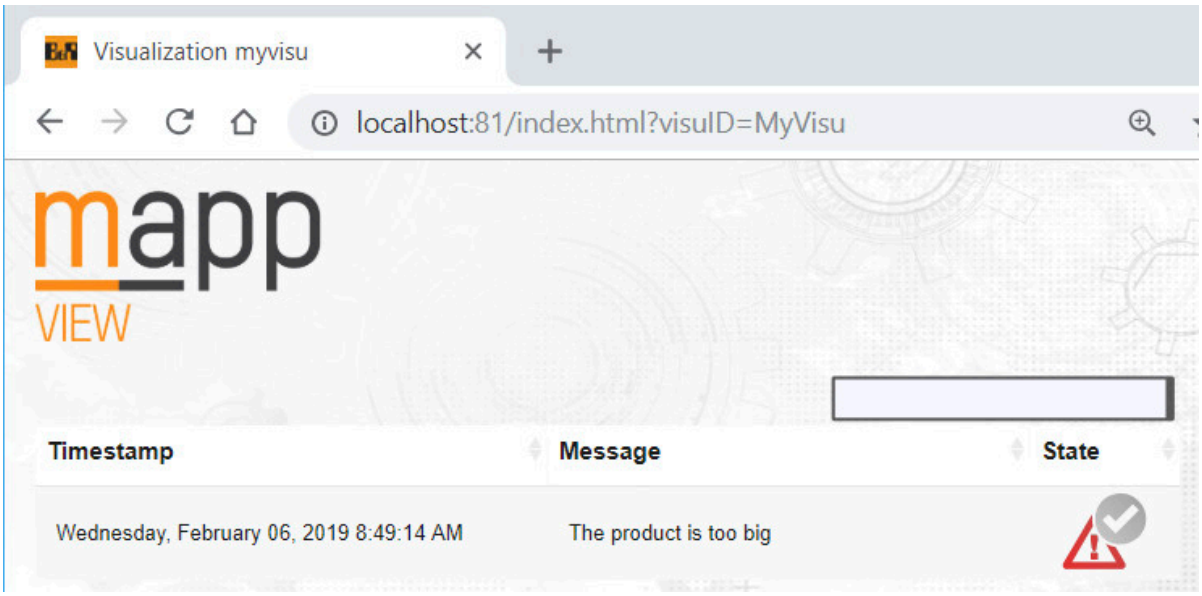
Mithilfe von **MpAlarmXSet** und **MpAlarmxReset** kann der Alarm aktiviert und deaktiviert werden.

Die Alarme werden über eine mapp View Visualisierung mithilfe des AlarmList-Widgets angezeigt. Wie das Widget parametriert werden muss, ist im Abschnitt Anzeigen von Alarmen erklärt.

Tritt nun der Alarm in der Maschine auf, kann der Maschinenanwender den Alarm über die Visualisierung quittieren. Der Alarm wird in der Visualisierung quittiert angezeigt:



Bei erneuten Auftreten des Alarms wird der Quittierstatus wieder zurückgesetzt:



1.4.7 Aggregieren von Alarmen

Anforderung

In einer Maschine befördern 10 verschiedene Achsen verschiedene Produkte.

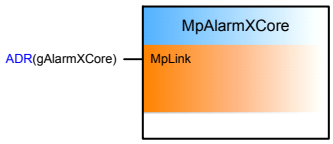
Treten bei manchen der Achsen Probleme auf, so soll in der Maschinenvisualisierung nur ein Achs-Alarm angezeigt werden, welcher stellvertretend für alle Achs-Alarme steht. Dadurch soll ein "überfluten" der Alarmliste durch verschiedene Achs-Alarme verhindert werden. Das bedeutet, alle Achs-Alarme aggregieren zu einen Alarm. Der Maschinenbediener weiß daher sofort durch einen Alarm bescheid und kann dementsprechend handeln. Sobald der Alarm quittiert wird, werden alle vorhandenen Achs-Alarme automatisch quittiert. Tritt nach der Quittierung des aggregierten Alarms ein weitere Achs-Alarm auf, so soll sich der Quittierstatus des Alarms ändern. Dadurch wird der Maschinenbediener über neue Achs-Alarme informiert.

Lösung

Komponentenliste

- ***MpAlarmXCore*** (Eigener MpLink): Sammelt alle Alarme der mapp Komponenten

Verbindungsschema



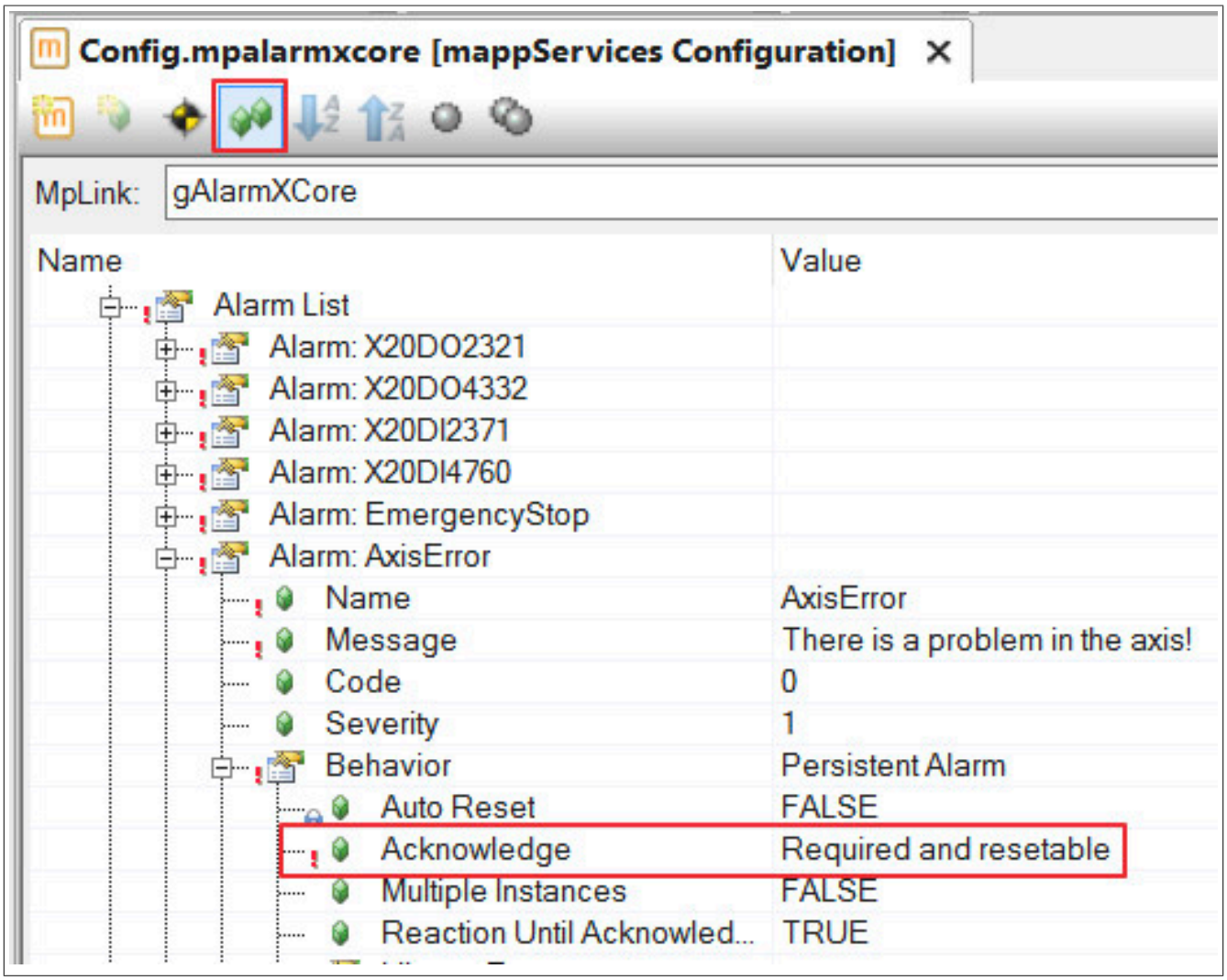
Achskonfiguration

Die Achsen werden über mapp Axis gesteuert. Das bedeutet, alle Achs-Alarme werden automatisch an mapp AlarmX geleitet. Dafür muss in der Konfiguration von mapp Axis MpAlarmX unter "Alarms" ausgewählt werden.

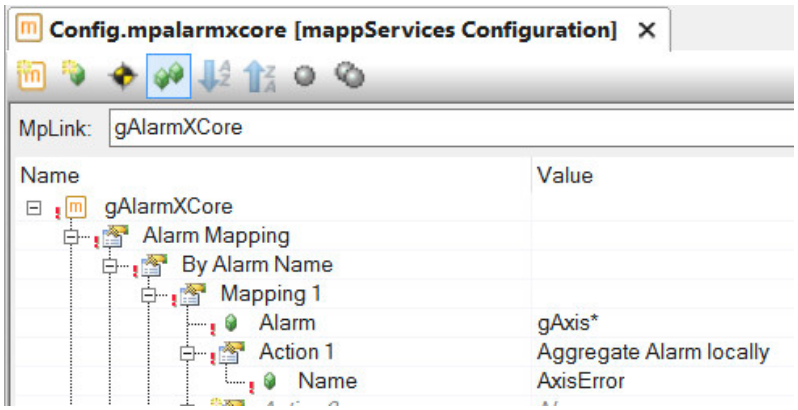
MpAlarmXCore Konfiguration

Um das Alarmsystem verwenden zu können, wird die MpAlarmXCore Konfiguration eingefügt.

Es wird der Alarm "AxisError" erstellt, welcher stellvertretend für alle Achs-Alarme in der Maschinenvisualisierung angzeigt werden soll. Damit sich der Quittierstatus des Alarms ändern kann, wird als Quittierverhalten "Required and resetable" ausgewählt. Um diese Option festzulegen, müssen die erweiterten Parameter der Konfiguration aktiviert werden.



Als nächstes müssen alle Achs-Alarme auf den Alarm "AxisError" gemappt werden. Damit die Alarme agreggiert werden können, wird als Aktion "Aggregate Alarm locally" ausgewählt. Damit alle Achs-Alarme automatisch auf "AxisError" aggregieren, wird als Name der mapp Link der Achs-Komponenten "gAxis*" definiert.

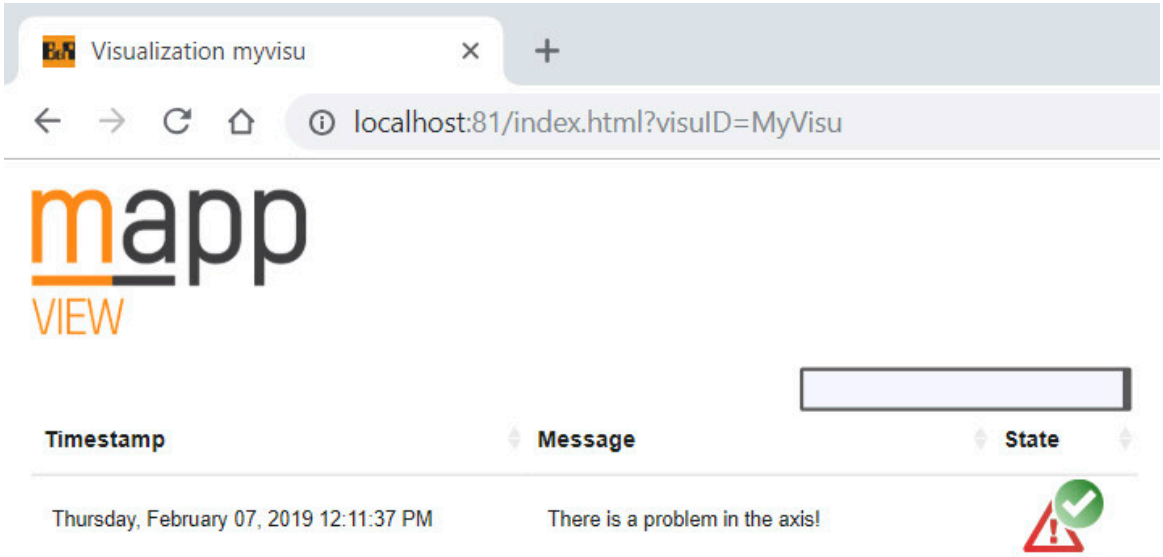


Verwendung der mapp Komponenten

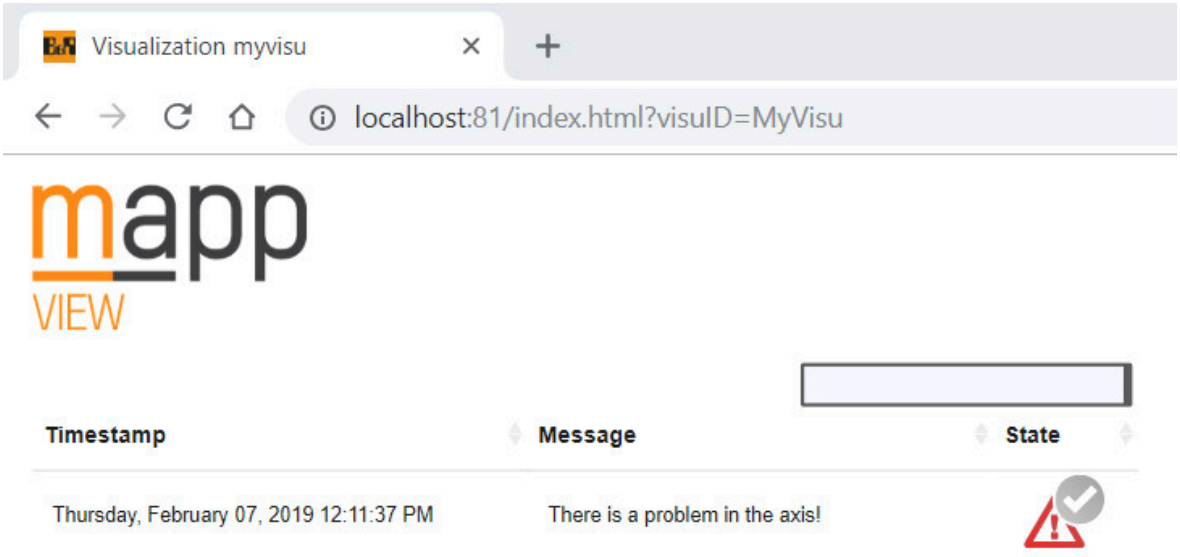
Es wird der Funktionsbaustein **MpAlarmXCore** für die Alarmverwaltung eingefügt.

Die Alarme werden über eine mapp View Visualisierung mithilfe des AlarmList-Widgets angezeigt. Wie das Widget parametriert werden muss, ist im Abschnitt Anzeigen von Alarmen erklärt.

Tritt nun ein Achs-Alarm an der Maschine auf, so wird der Alarm "AxisError" angzeigt. Der Maschinenbediener kann den Alarm quittieren.



Bei erneuten Auftreten eines Achs-Alarms wird der Quittierstatus wieder zurückgesetzt:

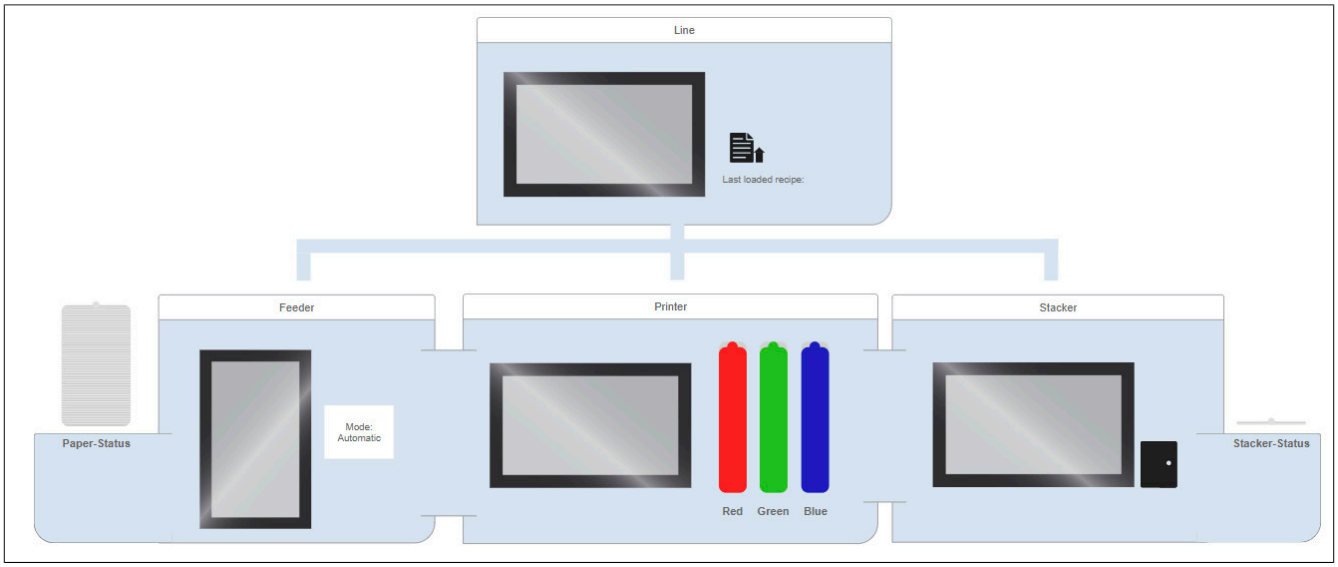


1.4.8 Alarmverwaltung für Druckermaschinenlinie

Anforderung

Eine Druckermaschinenlinie besteht aus mehreren Stationen. Einem Feeder, welche für die Papierzufuhr verantwortlich ist, dem Printer, welcher für das Drucken zuständig ist und einem Stacker welcher das bedruckte Papier lagert. Die gesamte Linie wird über eine Steuerung betrieben.

Pro Station wird zur Überwachung eine eigene Visualisierung verwendet. Weiters gibt es eine Visualisierung, welche die gesamte Linie überwacht.



Ziel ist es nun eine erweiterte Alarmverwaltung zu implementieren. Dabei sollen manche Alarme lokal und andere wiederum an der Linienüberwachung angezeigt werden. Folgende Alarme sollen in der Druckermaschinenlinie existieren:

Alarm	Beschreibung
Rezept-Alarm	Über die Linienüberwachung soll es möglich sein zu bestimmen was aktuell gedruckt wird. Dabei soll sichergestellt sein, dass nur gültige Druckaufträge bearbeitet werden. Bei Laden eines ungültigen Auftrags, soll dies über die Visualisierung sichtbar sein. Nur wenn das Rezept gültig ist, werden die Parameter an die einzelnen Module weitergegeben.
FeederAxis	Die Papierzufuhr im "Feeder" soll über ein Förderband geregelt werden. Sobald die Achse im Fehlerzustand ist, soll der Benutzer darüber informiert werden. Auftretende Achsfehler sollen durch einen einzigen Alarm auf der Visualisierung dargestellt werden. Dadurch wird sichergestellt, dass die Alarmliste nicht von Achs-Meldungen "überflutet" wird.
PaperEmpty	Der Papier-Füllstand soll überwacht werden. Ist nur noch wenig Papier übrig muss die Linienüberwachung darüber in Kenntnis gesetzt werden, dass der Feeder gewartet ("MaintainFeeder") werden muss. Lokal im Feeder sollen mehr Detail in der Alarmnachricht angezeigt werden.
PaperStuck	Die Papier-Zufuhr zum Printer soll überwacht werden. Bleibt das Papier stecken muss die Linienüberwachung darüber in Kenntniss gesetzt werden. Dabei soll der Alarm "MaintainFeeder" angezeigt werden. Dadurch weiß der Benutzer, dass der Feeder gewartet werden muss. Lokal im Feeder sollen mehr Detail in der Alarmnachricht angezeigt werden.
ManualMode	Sobald der Benutzer über den "Feeder" die Maschine in den manuellen Modus schaltet, muss die Linienüberwachung in Form eines Alarms darüber in Kentniss gesetzt werden.
PrinterColor	Im "Printer" werden die Füllstände der Patronen überwacht. Sind eine oder mehrere der Patronen leer, soll die Linie über die anstehende Wartungsarbeit in Kenntnis gesetzt werden. Was genaue gewartet werden muss ist nur lokal sichtbar. Welche Farbe genau ausgetauscht werden muss, soll über den "Printer" durch die Alarme "RedEmpty", "GreenEmpty" und "BlueEmpty" dargestellt werden.
SafetyDoorOpen	Beim Öffnen der Sicherheitstür muss sichergestllelt sein, dass sich die Geschwindigkeit des Stackers vermindert. Zusätzlich soll im Stacker angezeigt werden, dass die Sicherheitstür geöffnet wurde.
StackerFull	Sobald der Stacker voll ist muss die gesamte Maschinenlinie gestoppt werden. Zusätzlich soll im Stacker angezeigt werden, dass der Stacker voll ist.

Umsetzung

Um diese Produktionslinie mit den einzelnen Stationen zu realisieren wird eine mapp Hierarchie aufgebaut. Somit ist der Benutzer in der Lage die einzelnen Stationen unabhängig voneinander zu verwalten.

In den einzelnen Stationen werden auch andere mapp Komponenten verwendet. Das Förderband, welches für die Papierzufuhr sorgt wird mithilfe der mapp Komponente mapp Axis gesteuert. Um zu bestimmen welcher Auftrag gedruckt werden soll, wird ein Rezeptmanagement verwendet. Dieses wird mithilfe von mapp Recipe implementiert.

Erstellen der mapp Hierarchie

Aufbauen der Produktionslinie anhand einer mapp Hierarchie

Pro Station wird die Konfiguration "MpComGroupDefault" eingefügt. Unter "Child components" wird nun der mapp Link der Komponente angegeben, welche sich in der jeweiligen Maschinengruppe befinden soll.

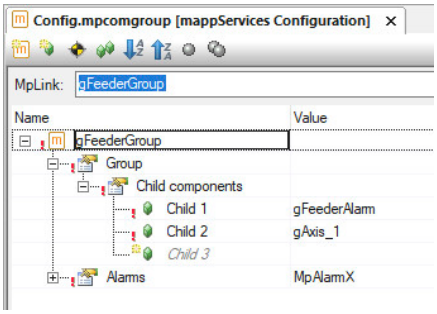


Abbildung 1: Station: "Feeder"

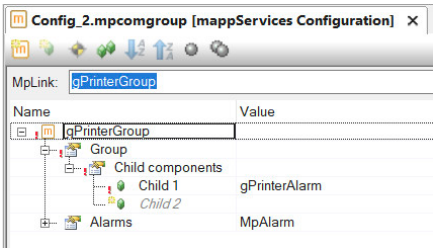


Abbildung 2: Station: "Printer"

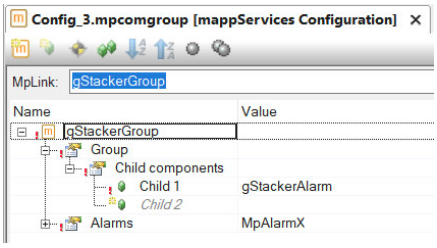


Abbildung 3: Station: "Stacker"

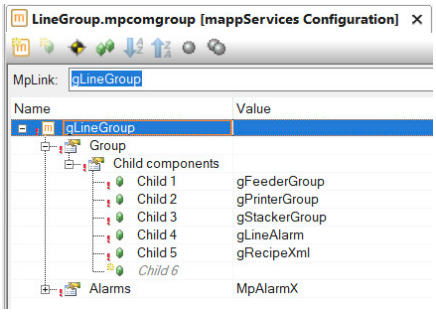


Abbildung 4: Station: "Line"

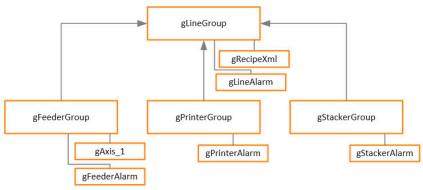


Abbildung 5: Überblick Hierarchie

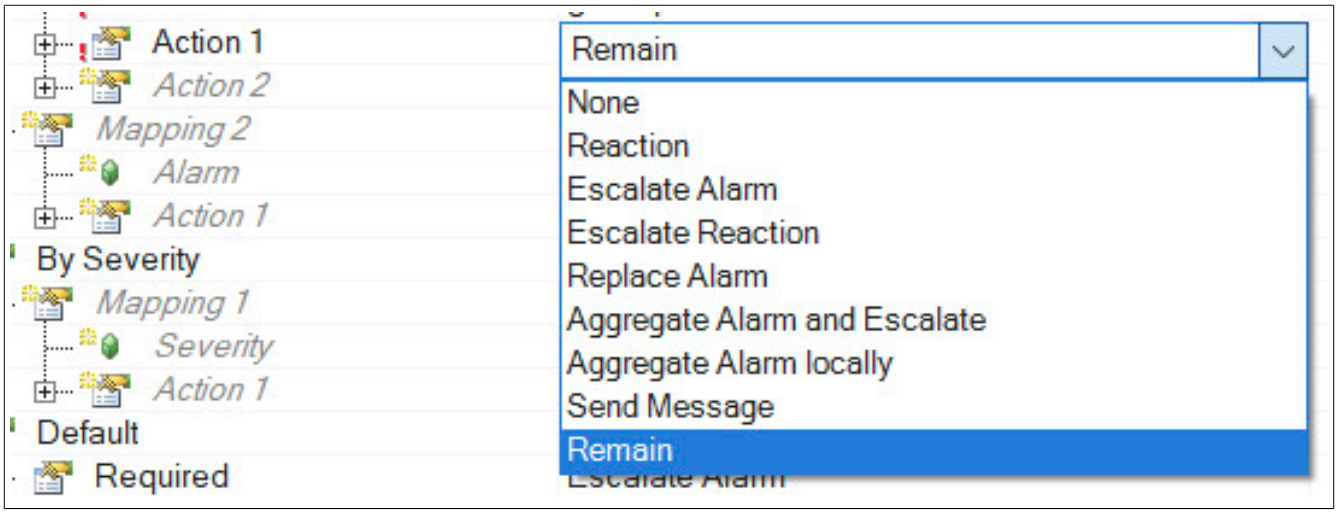
Alarmkonfiguration

Für die gesamte Linienüberwachung und pro Station wird die Konfiguration MpAlarmXCore eingefügt. In dieser Konfiguration werden die einzelnen Alarme und deren Alarmeigenschaften definiert. Dazu gehört das Alarmverhalten. Folgende Alarme werden pro Station erstellt:

Station	Alarm	Beschreibung	Alarm-Verhalten
Feeder	FeederAxis	Dieser Alarm soll ausgelöst werden sobald einer der Alarme von mapp Axis aktiv ist. Der Alarm soll nur einmal in der Alarmliste auftauchen und lokal quittiert werden.	Persistent Alarm
	Achs-Alarme	Alle Achs-Alarme werden über die mapp Axis Konfiguration verwaltet. Für mehr Information siehe Abschnitt "Achskonfiguration" weiter unten.	-
	PaperEmpty	Das Papier ist leer und muss nachgefüllt werden. Der Alarm soll lokal angezeigt werden sowie durch den Alarm "MaintainFeeder" ersetzt und nach oben hin zur Linienüberwachung eskaliert werden.	Persistent Alarm
	PaperStuck	Das Papier steckt fest. Der Alarm soll lokal angezeigt werden sowie durch den Alarm "MaintainFeeder" ersetzt und nach oben hin zur Linienüberwachung hin eskaliert werden.	Persistent Alarm
	MaintainFeeder	Der Alarm wird in der Linienüberwachung angezeigt. "MaintainFeeder" soll den Benutzer auf Wartungsarbeiten im "Feeder" aufmerksam machen. Die genaue Ursache wird lokal im Feeder angezeigt.	Persistent Alarm
	ManualMode	Wechselt der Benutzer vom automatischen Modus in den manuellen muss die Linienüberachung in Kentniss gesetzt werden. Der Alarm soll über die Linienüberwachung quittiert werden.	Edge Alarm
Printer	RedEmpty	Im Drucker ist die rote Farbe leer. Alarm soll lokal quittiert werden.	Level Monitoring
	BlueEmpty	Im Drucker ist die blaue Farbe leer. Alarm soll lokal quittiert werden.	Level Monitoring
	GreenEmpty	Im Drucker ist die grüne Farbe leer. Alarm soll lokal quittiert werden.	Level Monitoring
	PrinterColor	Alarm wird ausgelöst sobald einer der Farb-Alarme aktiv ist. PrinterColor soll in der Linienüberwachung quittiert werden.	Persistent Alarm
Stacker	StackerFull	Der Stacker ist voll. Als Reaktion auf diesen Alarm soll die gesamte Maschinenlinie gestoppt werden. Der Alarm wird lokal quittiert.	Level Monitoring
	SafetyDoorOpen	Die Sicherheitstür im Stacker ist offen. Als Reaktion auf diesen Alarm soll die Zufuhrgeschwindigkeit im Stacker vermindert werden. Der Alarm wird lokal quittiert.	Discrete Value Monitoring
Line	Rezept-Alarme	Alle Alarme der Rezeptverwaltung werden über die mapp Recipe Konfiguration verwaltet. Für mehr Information siehe Abschnitt "Rezeptverwaltung" weiter unten	-

Außerdem wird in der Konfiguration festgelegt, welche Alarme lokal verwaltet werden und welche nach oben zur Linienüberwachung eskaliert werden sollen. Zusätzlich wird definiert, welcher Alarm welche Reaktion auslösen soll. Dies wird im Abschnitt "Alarm Mapping" über "Action" konfiguriert.

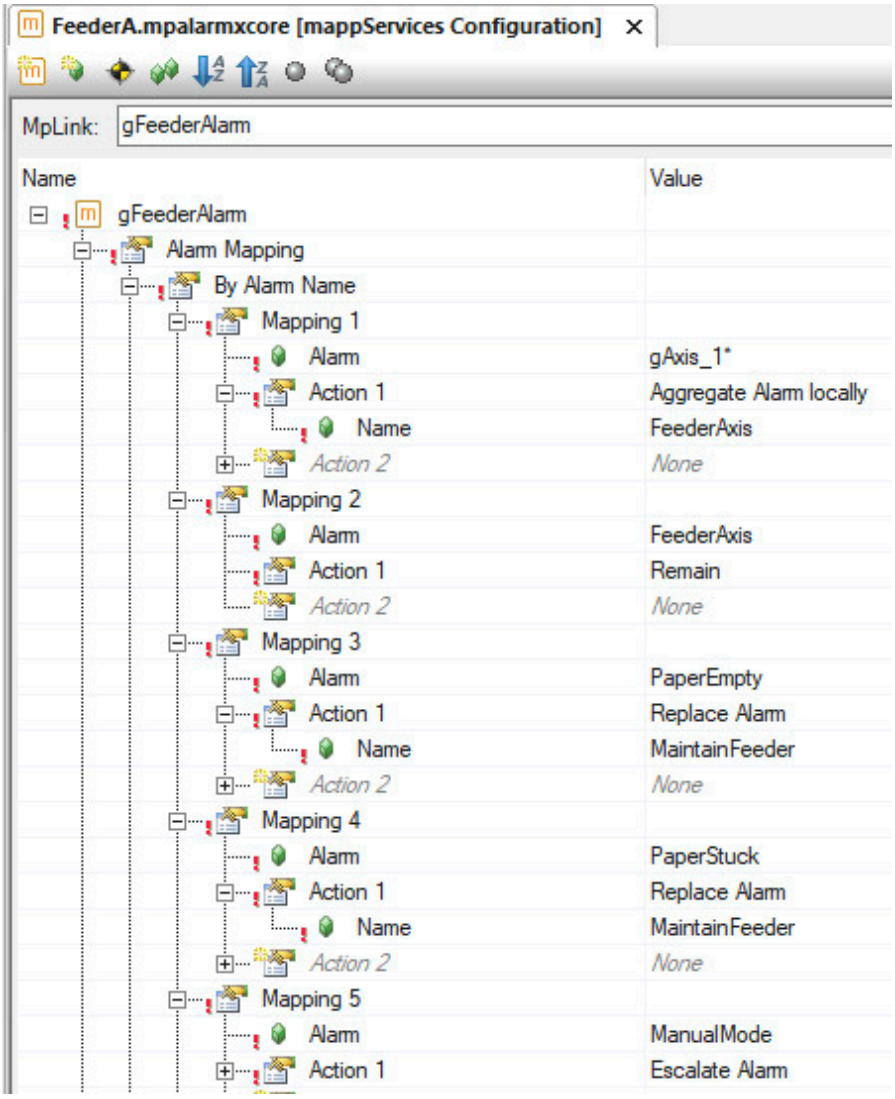
Welche Auswirkung die einzelnen Auswahlmöglichkeiten haben wird im Abschnitt Alarm Mapping behandelt.



Konfiguration "Feeder"

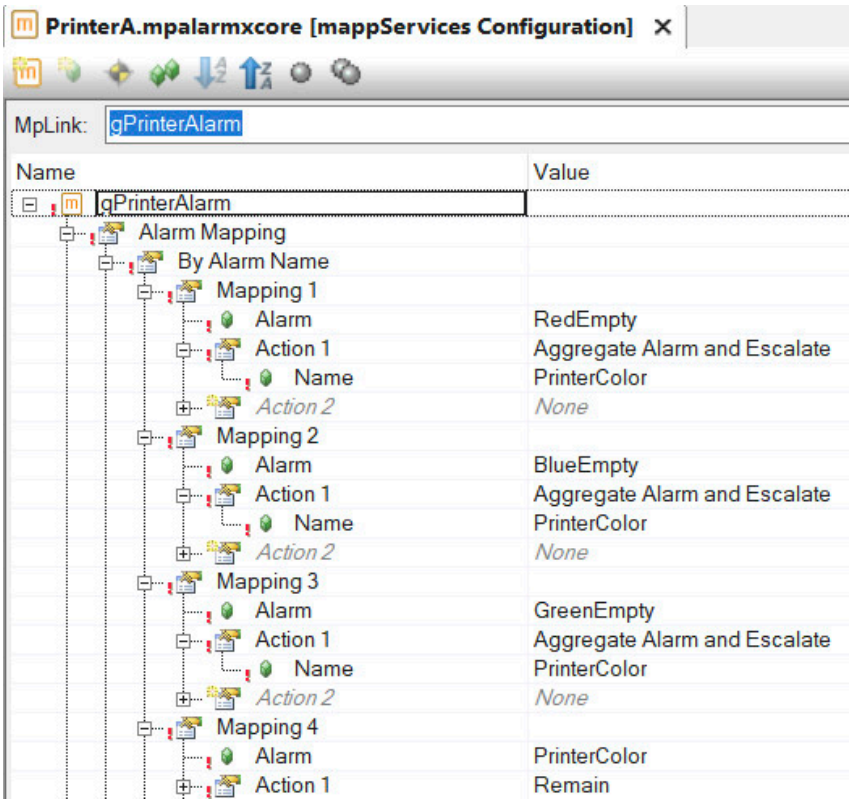
In der MpAlarmXCore Konfiguration für den "Feeder" werden die Alarme "PaperEmpty", "FeederAxis", "UserLocked" von mapp UserX sowie alle Alarme welche in mapp Axis auftreten gemappt.

- **gAxis_1*:** Alle Alarme welche durch mapp Axis ausgelöst werden, sollen lokal aggregiert werden. Anstelle einzelner Achs-Alarme soll ein gemeinsamer Alarm "FeederAxis" angezeigt werden.
- **FeederAxis:** Der Alarm "FeederAxis" tritt auf, sobald ein Achs-Alarm ausgelöst wird. Der Alarm "FeederAxis" soll dabei lokal ("Remain") behandelt werden.
- **PaperEmpty:** Der Alarm soll lokal angzeigt sowie durch den Alarm "MaintainFeeder" ersetzt und nach oben hin eskaliert werden. Dafür wird "Replace Alarm" ausgewählt.
- **PaperStuck:** Der Alarm soll lokal angzeigt sowie durch den Alarm "MaintainFeeder" ersetzt und nach oben hin eskaliert werden. Dafür wird "Replace Alarm" ausgewählt.
- **ManualMode:** Wechselt der Benutzer vom automatischen Modus in den manuellen, wird der Alarm "ManualMode" ausgelöst. Dieser soll nur in der Linienüberwachung angzeigt werden. Aus diesem Grund wird "Escalate Alarm" ausgewählt. Der Alarm wird dabei lokal nicht mehr angezeigt.



Konfiguration "Printer"

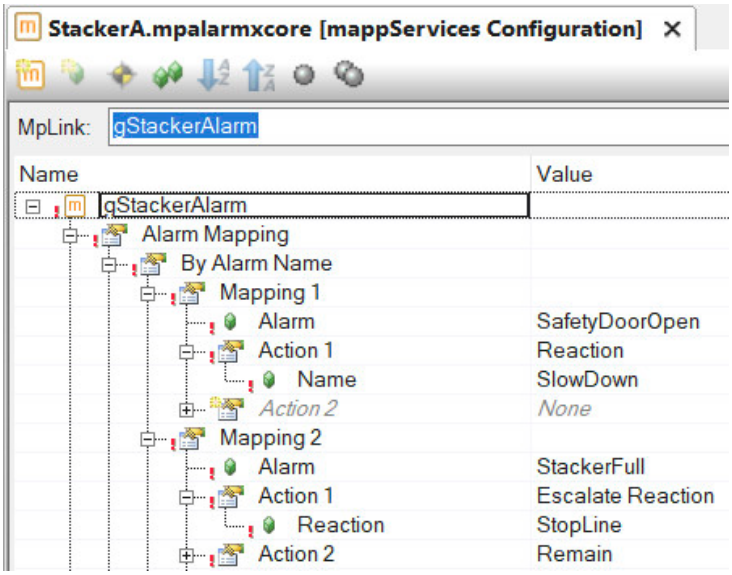
Für die Konfiguration des "Printer" werden die Alarme "RedEmpty", "BlueEmpty" sowie "GreenEmpty" gemappt. Sobald ein oder mehrere Farb-Alarme auftreten sollen diese aggregiert und nach oben hin zur Linienüberwachung eskaliert ("Aggregate Alarm and Escalate") werden. Das bedeutet, sobald ein oder mehrere Farb-Alarme ausgelöst werden, wird einmalig der Alarm "PrinterColor" aktiviert und nach oben hin eskaliert. Der Alarm "PrinterColor" kann dadurch nur einmal in der Linienüberwachung angezeigt werden.



Konfiguration "Stacker"

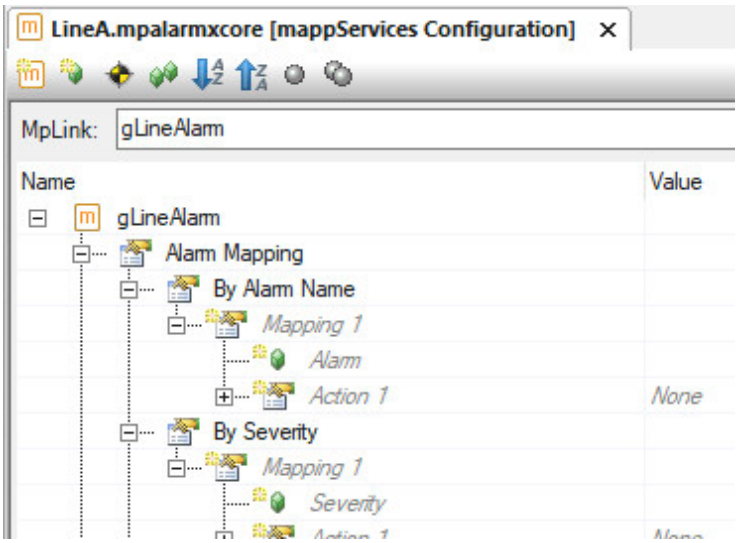
In der MpAlarmXCore Konfiguration des Stackers werden die Alarme "SafetyDoorOpen" und "StackerFull" gemappt. Sobald die Sicherheitstür geöffnet wird, ist der Alarm "SafetyDoorOpen" aktiv. Dieser Alarm soll eine Reaktion ("Reaction") auslösen, welche die Zufuhrgeschwindigkeit zum Stacker ("SlowDown") vermindert.

Sobald der Stacker voll ist, wird der Alarm "StackerFull" ausgelöst. Tritt der Alarm auf, soll als Reaktion darauf die gesamte Druckerlinie ("Escalate Reaction") gestoppt werden ("StopLine"). Die eskalierte Reaktion wird bei der Linienüberwachung überprüft und alle Module können dann über die nötige Aktion informiert werden.



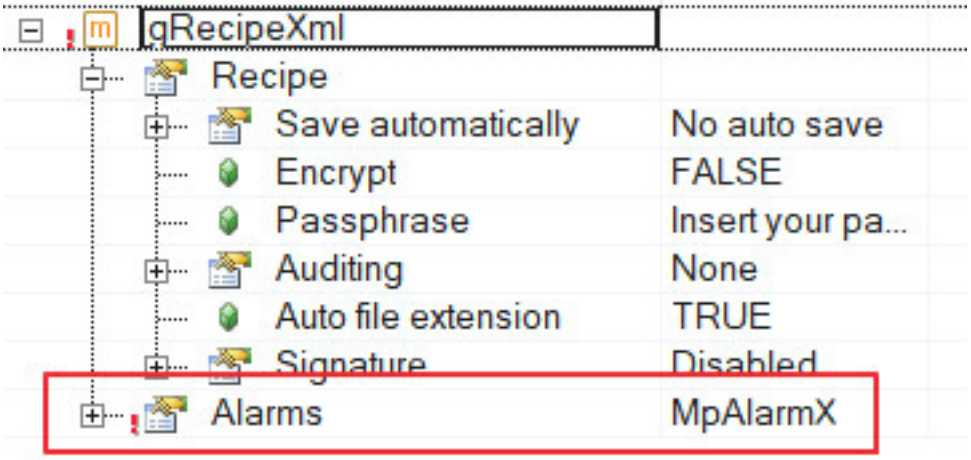
Konfiguration "Line"

In der Gruppe für die Linienüberwachung befindet sich die Rezeptverwaltung. Aus diesem Grund werden die Alarme in der Rezeptverwaltung automatisch in der Linienüberwachung angezeigt. Es ist kein weiteres Mapping nötig.



Rezeptverwaltung

Damit über die Linienüberwachung die verschiedenen Rezepte geladen werden können, wird die Konfiguration MpRecipeXml eingefügt. Damit die Alarme der Rezeptverwaltung automatisch an das Alarmmanagement weitergeleitet werden, wird "MpAlarmX" unter "Alarms" definiert.

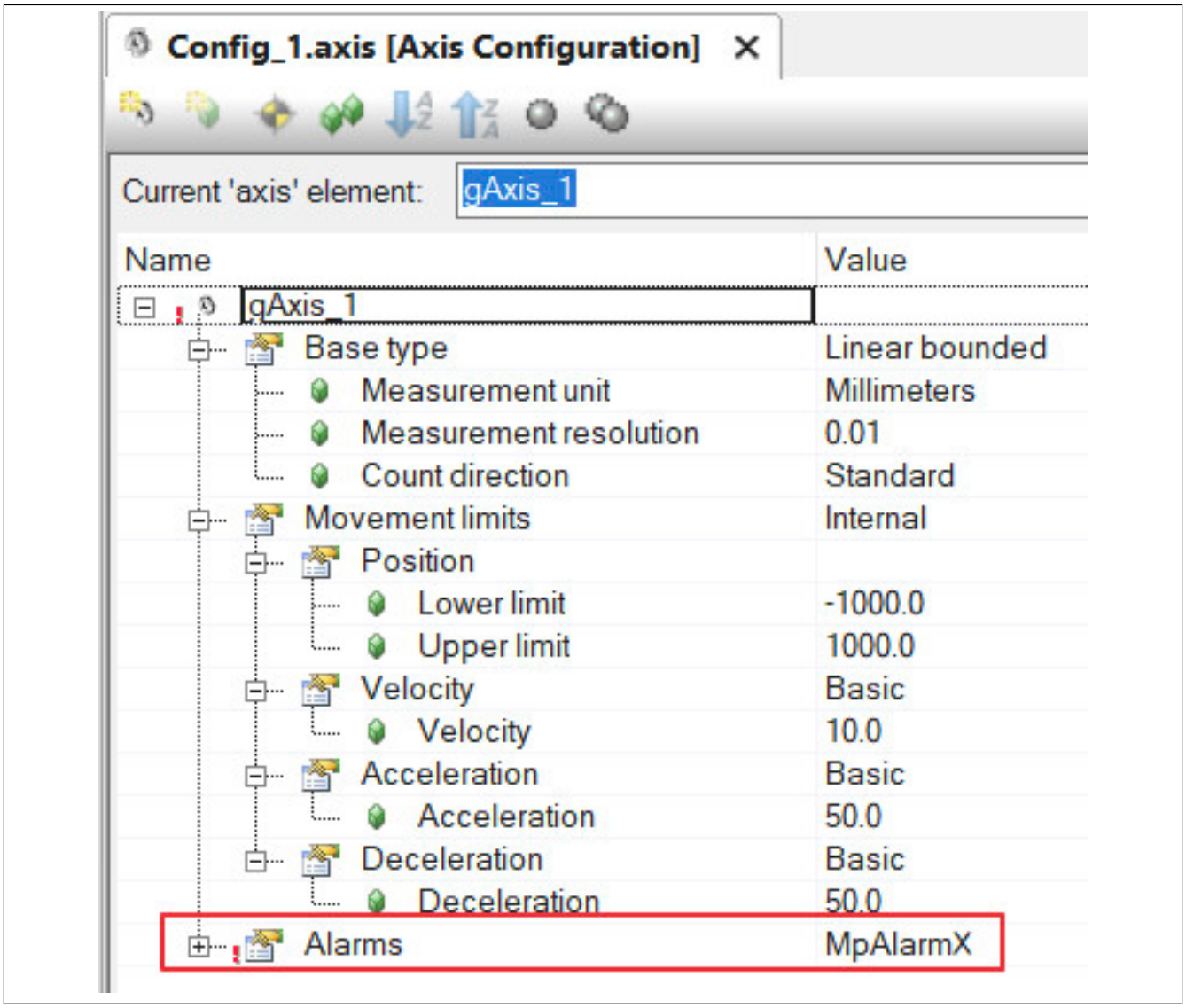


Auf unserem Youtube Kanal sind die Tutorials mapp Recipe Part 1 und mapp Recipe Part 2 welche erklären wie ein Rezeptmanagement implementiert werden kann.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Recipe (GER/ENG) zu finden. (Zugangsdaten)

Achskonfiguration

Um das Förderband in der Feeder-Station anzusteuern, wird die Komponente mapp Axis verwendet. Dafür wird die Konfiguration MpAxis eingefügt. Damit die Achs-Alarme automatisch an das Alarmsystem weitergeleitet werden, wird "MpAlarmX" unter "Alarms" ausgewählt.



Im Getting Started von mapp Axis wird erklärt, wie eine Achse in Betrieb genommen werden kann.

Vorbereiten der Visualisierung

Die Visualisierungen werden mithilfe von mapp View erstellt. Wie eine mapp View Visualisierung erstellt wird, ist hier erklärt.

Damit die erstellten Alarme angezeigt werden können, wird das AlarmList-Widget verwendet. Wie das Widget verwendet wird, ist im Anwendungsfall "Anzeigen von Alarmen in einer Tabelle" erklärt.

Verwendung der mapp Komponenten

Es wird für die Linienüberwachung und pro Station die Komponente **MpAlarmXCore** für die Alarmverwaltung eingefügt.

Da die meisten erstellten Alarme als Alarmverhalten "Level Monitoring" bzw. "Discrete Value Monitoring" besitzen, werden die Alarme automatisch ausgelöst sobald die definierte Bedingung eintritt.

Für jene mit dem Verhalten "Edge Alarm" oder "Persistent Alarm" muss die Funktion **MpAlarmXSet** für das aktivieren des Alarms bzw. **MpAlarmXReset** für das deaktivieren verwendet werden.

Um zu prüfen ob eine bestimmte Reaktion aktiv ist, wie zum Beispiel das Stoppen der Linie, wird die Funktion **MpAlarmXCheckReaction** verwendet. Ist die abgefragte Reaktion aktiv, so kann die Linie gestoppt werden.

1.4.9 Aktualisieren der Alarm-Information

Anforderung

Das Kühlwasser für eine Anlage wird in einem Tank gelagert. Sobald der Wasserstand zu niedrig ist, soll der Alarm "WaterTankLevelLow" ausgelöst werden. Der Alarm soll über eine Visualisierung angezeigt werden. Der aktuelle Wasserstand soll zusätzlich im Alarm mitgeliefert werden.

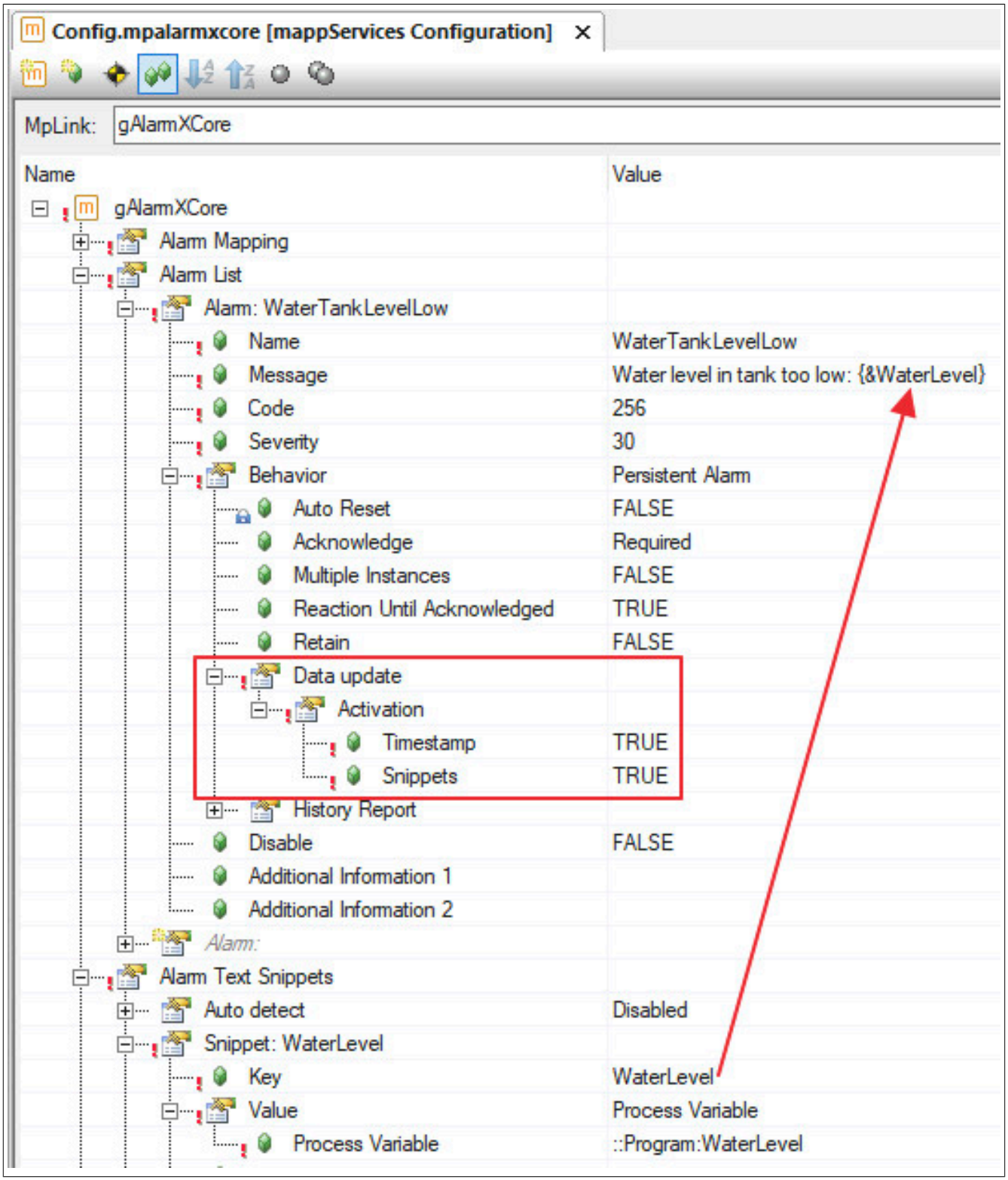
Der Wert im Alarm sowie der Zeitstempel soll aktualisiert werden, sobald der Wasserstand um weitere 500 Liter gefallen ist. Dadurch weiß der Maschinbediener wie kritisch der aktuelle Wasserstand ist und kann dementsprechend reagieren.

MpAlarmXCore Konfiguration

Um das Alarmsystem verwenden zu können, wird die MpAlarmXCore Konfiguration eingefügt.

Es wurde der Alarm "WaterTankLevelLow" definiert. Damit der aktuelle Wasserstand mitgeliefert werden kann, wird das Alarmsnippet "WaterLevel" erstellt und in der Alarmnachricht angegeben.

Damit der Wasserstand sowie der Zeitstempel des Alarms aktualisiert werden kann, wird die "Data update" Funktion verwendet. Dazu werden die erweiterten Parameter der Konfiguration aktiviert. Durch setzten der Parameter auf `TRUE` wird das Alarmsnippet sowie der Zeitstempel des Alarms aktualisiert, sobald der Alarm erneut gesetzt wird.



Vorbereiten der Visualisierung

Die Visualisierung wird mithilfe von mapp View erstellt. Wie eine mapp View Visualisierung erstellt wird, ist hier erklärt.

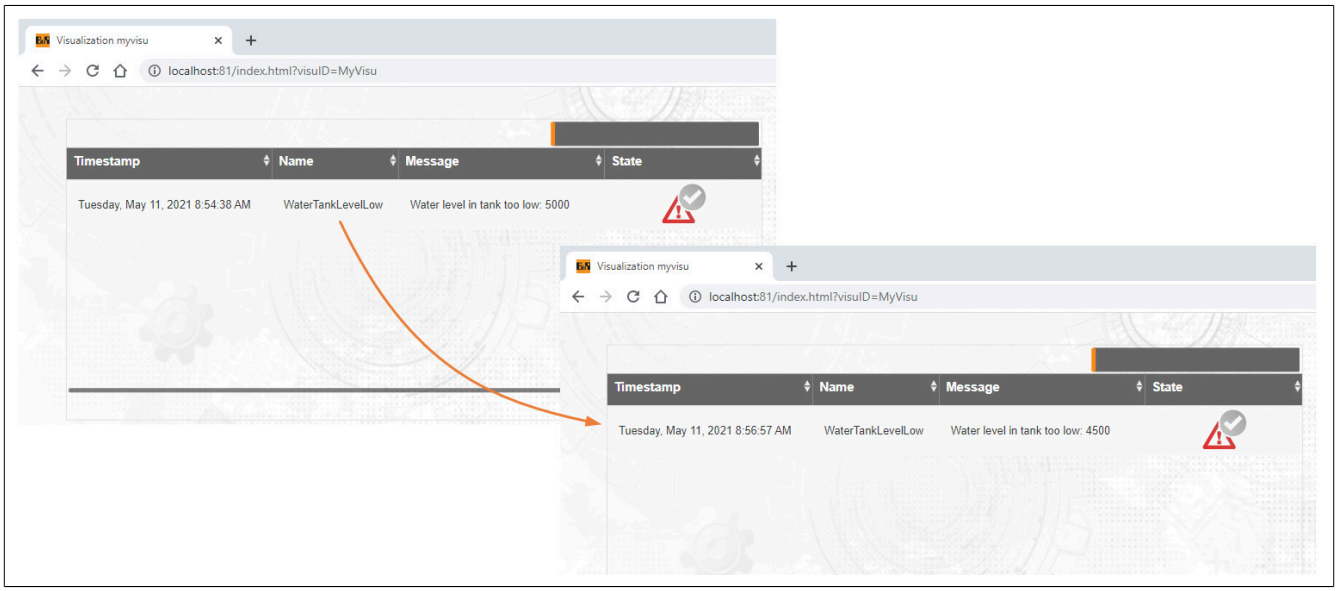
Damit die erstellten Alarme angezeigt werden können, wird das AlarmList-Widget verwendet. Wie das Widget verwendet wird, ist im Anwendungsfall "Anzeigen von Alarmen in einer Tabelle" erklärt.

Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpAlarmXCore** für die Alarmverwaltung eingefügt.

Die Alarme werden über eine mapp View Visualisierung mithilfe des AlarmList-Widgets angezeigt. Wie das Widget parametriert werden muss, ist im Abschnitt Anzeigen von Alarmen erklärt.

Der Alarm wird mithilfe der Funktionen **MpAlarmXSet** sowie **MpAlarmXReset** aktiviert bzw. deaktiviert. Sobald der Wasserstand um weitere 500 Liter gefallen ist, wird der Alarm erneut durch **MpAlarmXSet** gesetzt. Dadurch wird das Alarmsnippet und der Zeitstempel des Alarms aktualisiert.



1.4.10 Verbessern der Alarmperformance

Anforderung

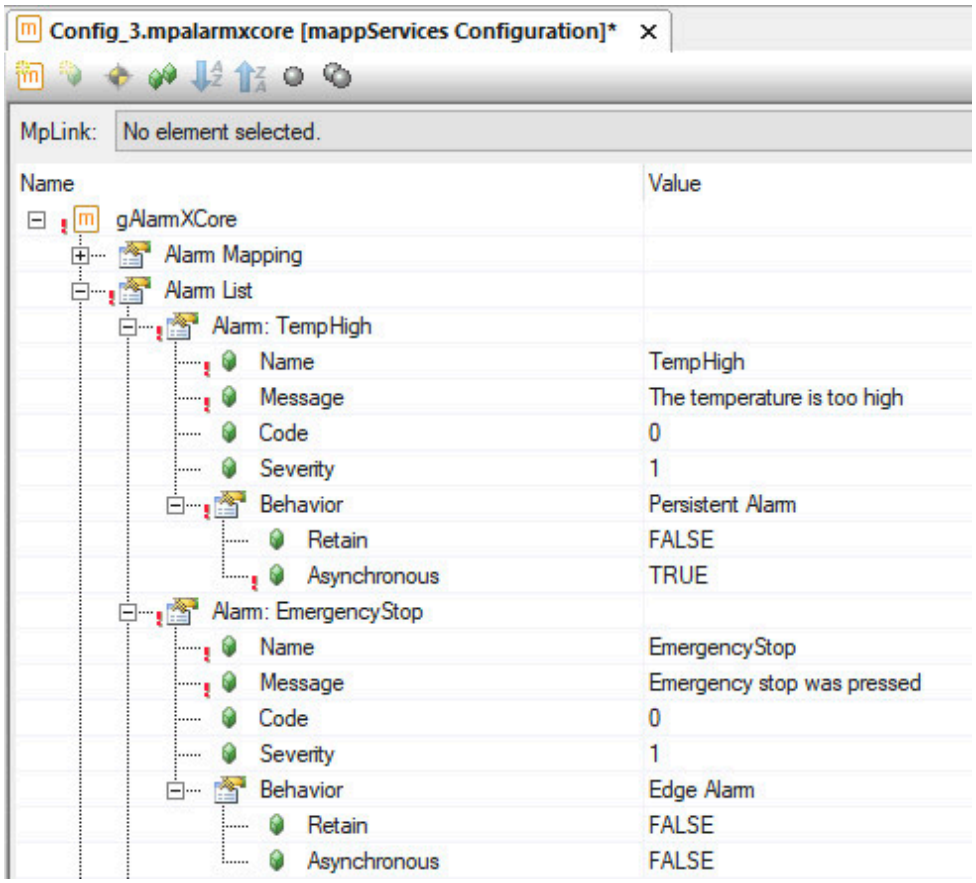
Die Alarmverwaltung soll so effizient wie möglich gestaltet werden.

MpAlarmXCore Konfiguration

Um das Alarmsystem verwenden zu können, wird die MpAlarmXCore Konfiguration eingefügt.

Alarme können synchron oder asynchron verwaltet werden. Synchrone Alarme werden im gleichen Kontext aktiviert, zurückgesetzt oder quittiert wo sie aufgerufen werden. Asynchrone Alarme werden nicht im gleichen Kontext aktiviert, zurückgesetzt und quittiert wo sie aufgerufen wurden, sondern werden an MpAlarmXCore weitergeleitet und dort verarbeitet. Bei dieser Weiterleitung wird lediglich der Zeitstempel des Alarms mitgeliefert.

Um die Alarmperfomance effizient zu gestalten werden alle Alarme auf "Asynchronous = `TRUE`" gestellt. Nur bei einzelnen Alarmen, welche sofort aufgelöst werden müssen damit kein Schaden entsteht, wird "Asynchronous = `FALSE`" konfiguriert und damit etwaige Reaktionen sofort aufgelöst. Für den Alarm "WaterTankLevel" wird ein asynchrones Verhalten festgelegt, für "EmergencyStop" ein synchrones. Für mehr Information siehe hier.



Verwendung der mapp Komponenten

Es wird der Funktionsbaustein MpAlarmXCore für die Alarmverwaltung eingefügt.

Um das Bearbeiten der Alarme noch effizienter zu gestalten werden die Alarme über den Funktionsbaustein MpAlarmXAlarmControl verwaltet. Beim Aktivieren dieses Funktionsbausteins werden bereits die relevanten Alarm-Einstellungen ausgelesen und damit können alle weiteren Funktionen, wie das Setzen und Zurücksetzen, schneller verarbeitet werden.

MpAlarmXCore wird in einer niederprioren Taskklasse mit hoher Toleranz aufgerufen. Dadurch bleibt zum Einen ausreichend Performance für die Applikation und zum anderen genügend Zeit ein erhöhtes Alarmaufkommen in kurzer Zeit zu verarbeiten.

1.4.11 Anzeigen von PDFs über AdditionalInformation Parameter

Anforderung

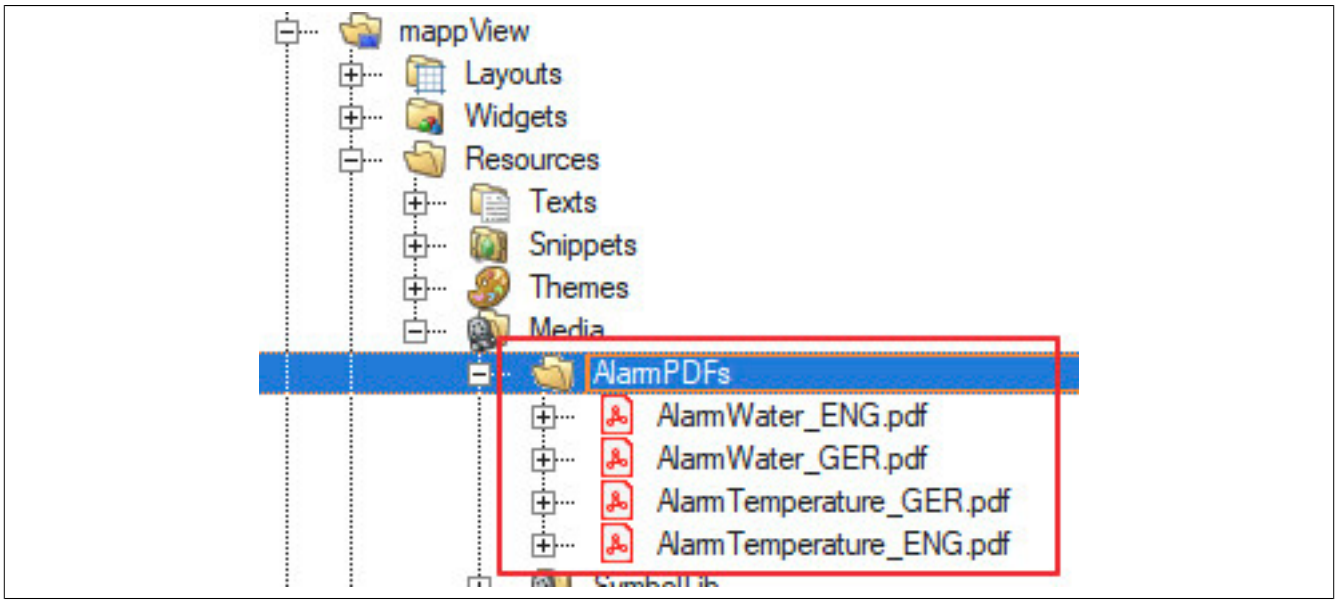
Tritt ein Alarm an der Maschine auf sollen weiterführenden Informationen zu diesem Alarm auf der Visualisierung in einem PDF angezeigt werden. In der PDF-Datei wird erklärt, welche Aktionen an der Maschine erforderlich sind, um die Ursache des Alarms zu beheben.

Vorbereiten der Visualisierung

Die Visualisierung wird mithilfe von mapp View erstellt. Wie eine mapp View Visualisierung erstellt wird, ist hier erklärt.

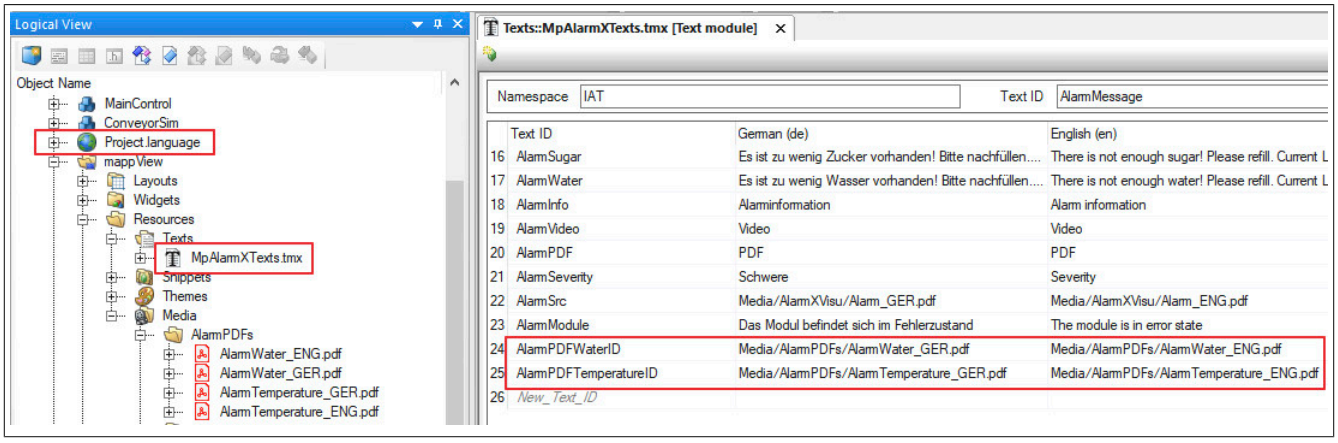
PDF

Die anzuzeigenden PDF-Dateien werden im mapp View Paket unter "Resources/Media/AlarmPDFs" eingefügt.

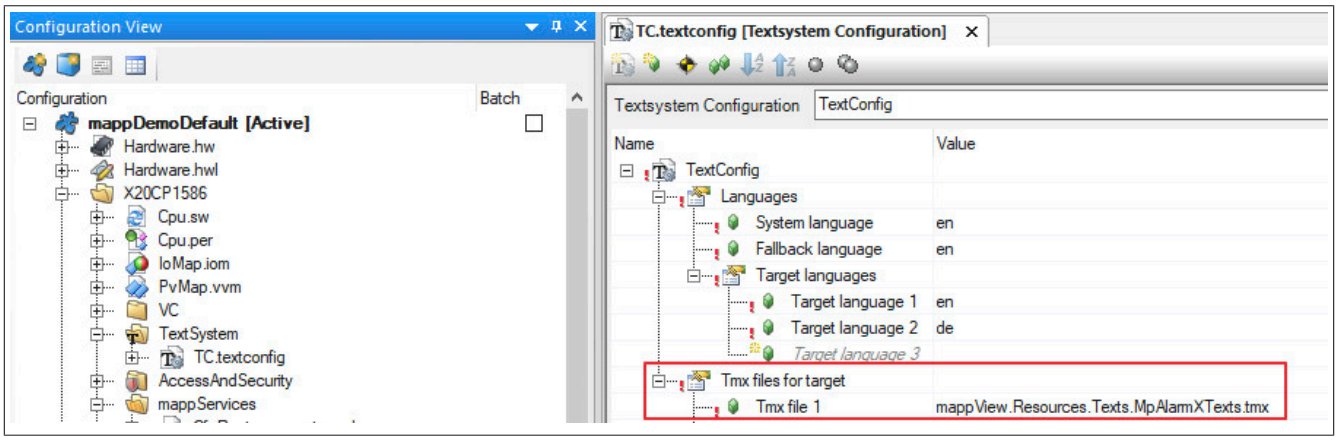


Vorbereitung des Textsystems

Der Pfad zur PDF-Datei wird über das Textsystem definiert. Dazu wird eine TMX-Datei eingefügt, die zu verwendenden Projektsprachen, sowie die Konfiguration des Textsystems. Über die Text-IDs "AlarmPDFWaterID" und "AlarmPDFTemperatureID" wird der Pfad für das englische und deutsche PDF angegeben. Wichtig hierbei ist, dass als Namespace "IAT" verwendet wird.



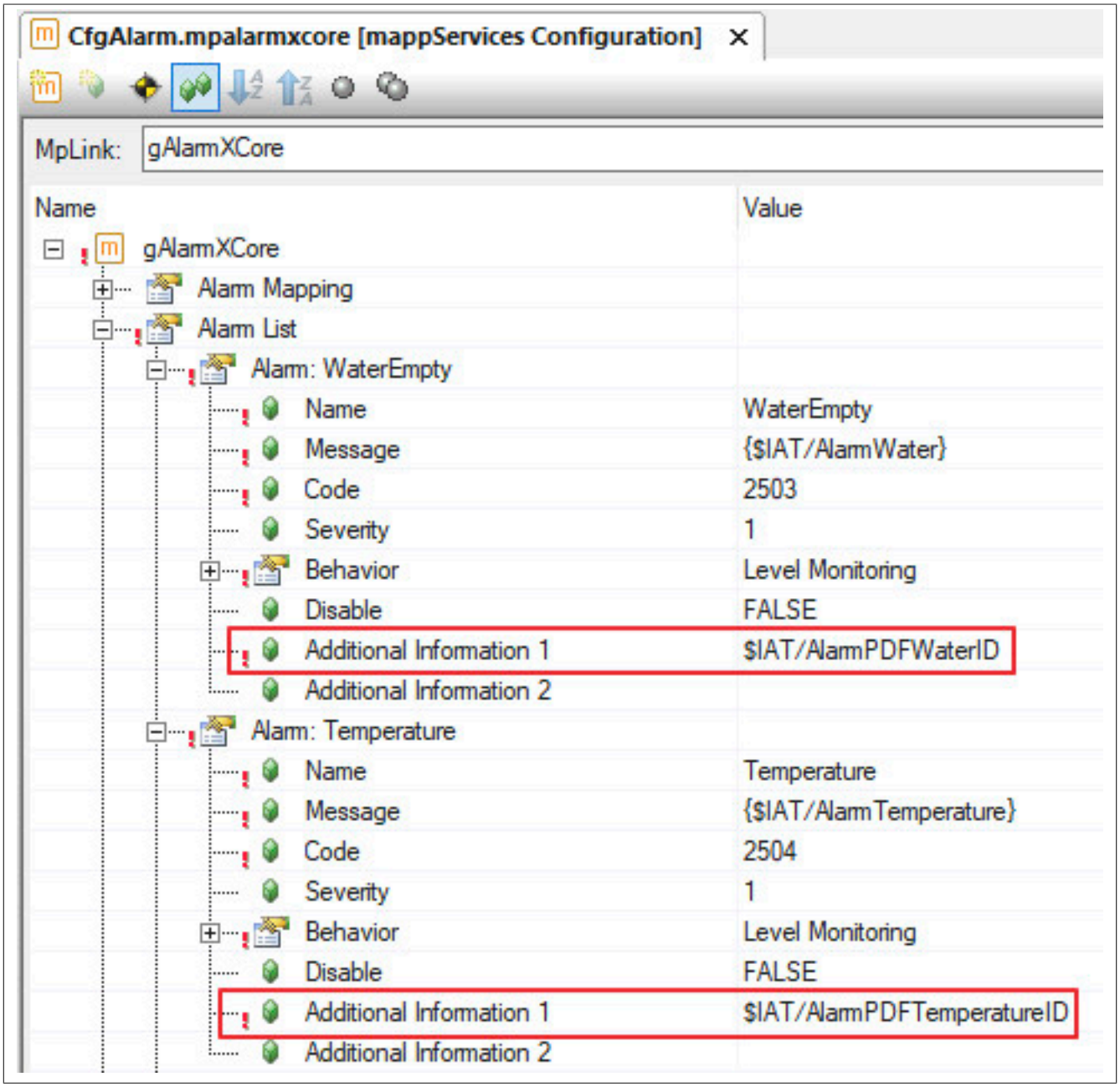
In der Konfiguration muss die TMX-Datei unter "Tmx files for target" angegeben werden. Für mehr Information zum Textsystem siehe Textsystem.



MpAlarmXCore Konfiguration

Um das Alarmsystem verwenden zu können, wird die MpAlarmXCore Konfiguration eingefügt.

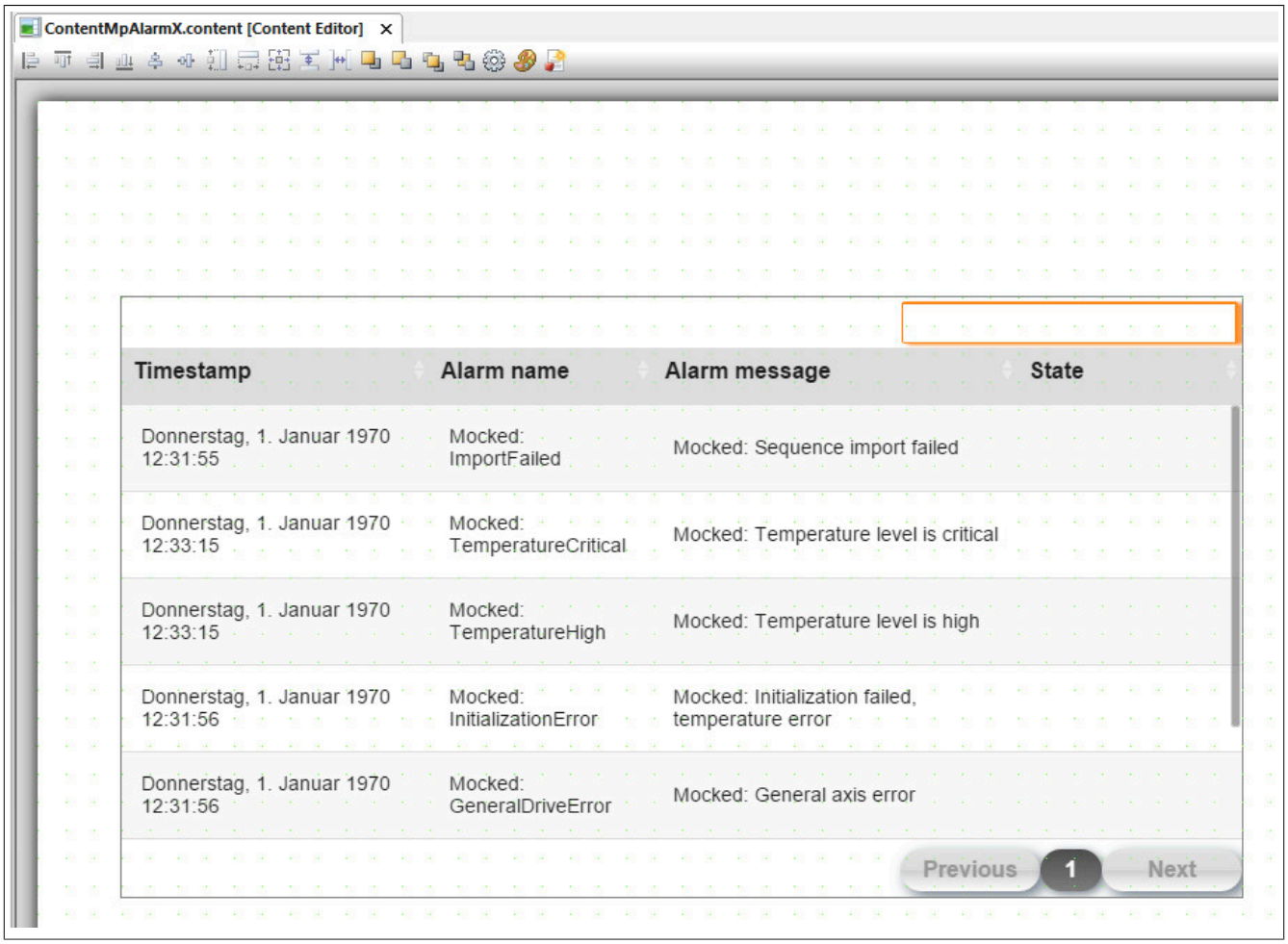
Es wurden zwei Alarme definiert. Unter "Additional Information 1" wird die Text ID ("$IAT/AlarmPDFWaterID", "$IAT/AlarmPDFTemperatureID") angegeben, welche den Pfad zur PDF-Datei enthält. Über diese Info wird später der Pfad zur PDF-Datei mithilfe des Textsystems aufgelöst. Dadurch können die PDFs sprachabhängig angezeigt werden.



Weitere Schritte in der Visualisierung

Einfügen der Alarmliste:

Damit die erstellten Alarme angezeigt werden können, wird das AlarmList-Widget verwendet. Wie das Widget projektiert wird, ist im Anwendungsfall "Anzeigen von Alarmen in einer Tabelle" erklärt.

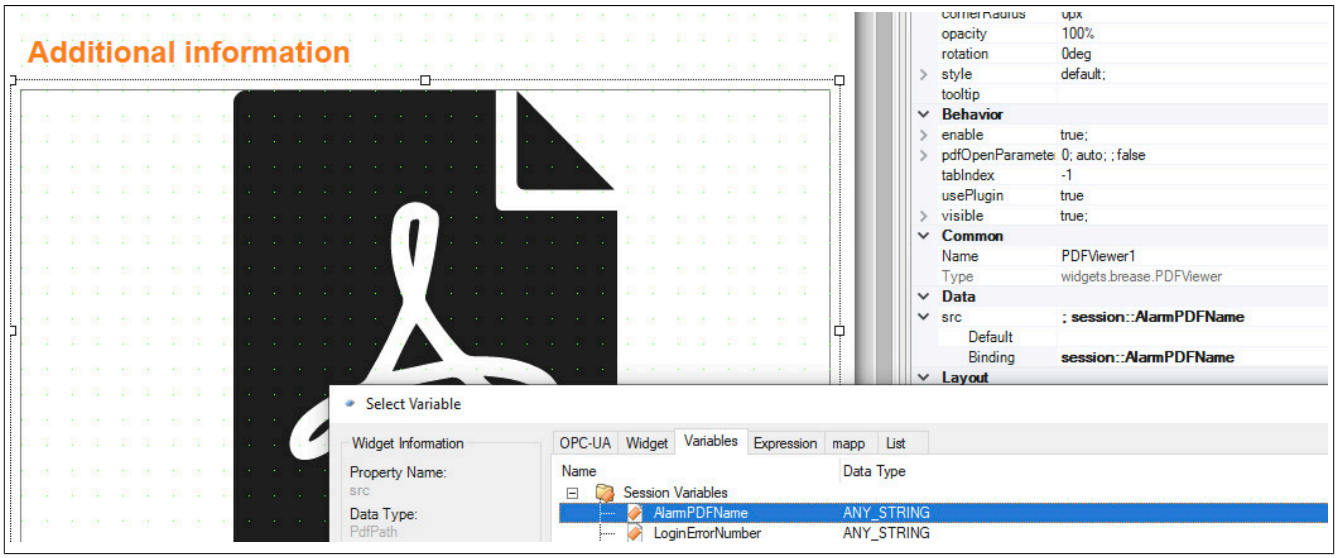


Session-Variable erstellen:

Die Information des PDF-Pfads wird mithilfe einer Session Variable gespeichert und anschließend am PDFViewer-Widget angegeben. Es wird die Session Variable "AlarmPDFName" erstellt.

PDFViewer-Widget:

Damit das PDF angezeigt werden kann, wird das PDFViewer-Widget eingefügt. Unter dem Parameter "src" wird der Pfad zum anzuzeigenden PDF angegeben. Es wird die zuvor definierte Session Variable angegeben.



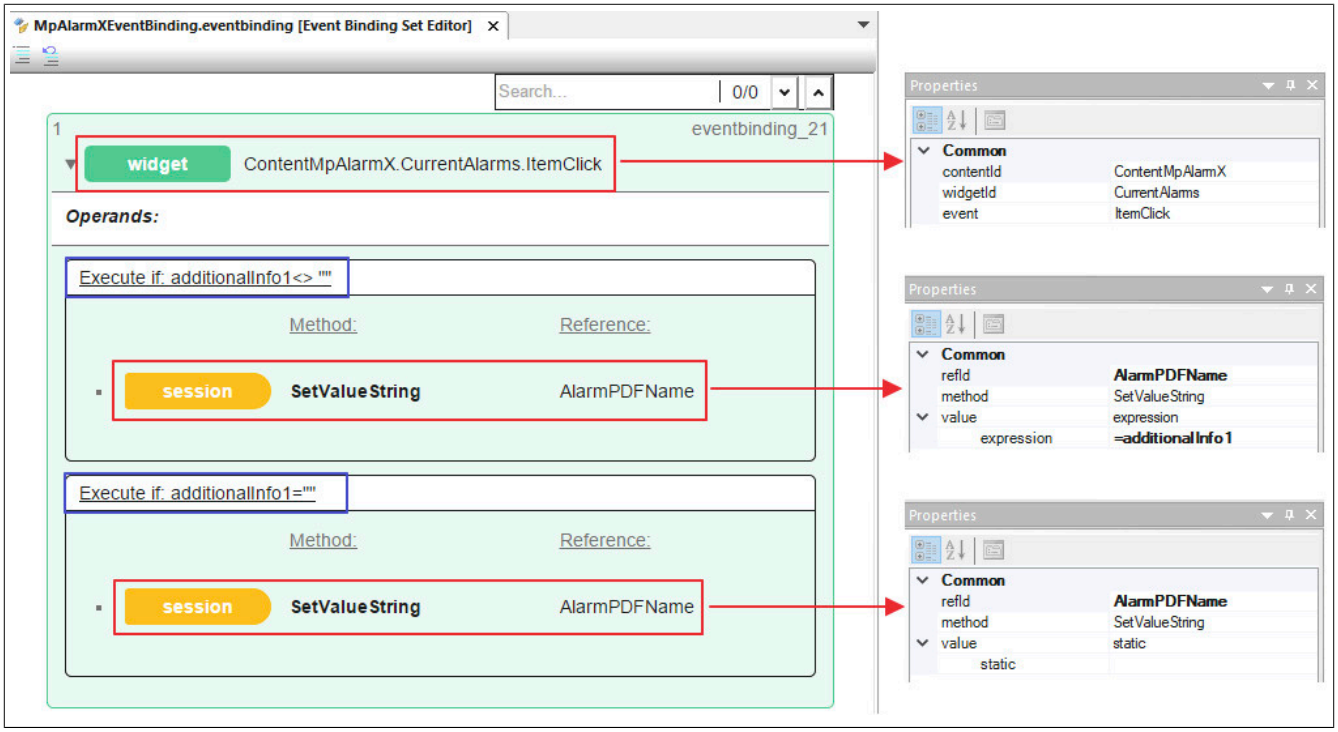
ActionEvent-Binding:

Um den richtigen PDF-Pfad an die Session Variable zu übergeben, wird ein ActionEvent-Binding benötigt. Es wird auf das ItemClick-Ereignis des AlarmList-Widget reagiert.

Wurde ein Alarm selektiert welche eine zusäztliche Information beinhaltet, kann darauf reagiert werden.

Ist eine zusätzliche Information enthalten (additionalInfo1<>""), so wird diese Info auf die zuvor erstellte Session Variable übergeben.

Beinhaltet ein Alarm keine zusätzliche Information, so muss auch die Session Variable dementsprechend geändert werden, damit keine PDF-Datei angezeigt wird. In diesem Fall wird die Session Variable auf " " gesetzt.

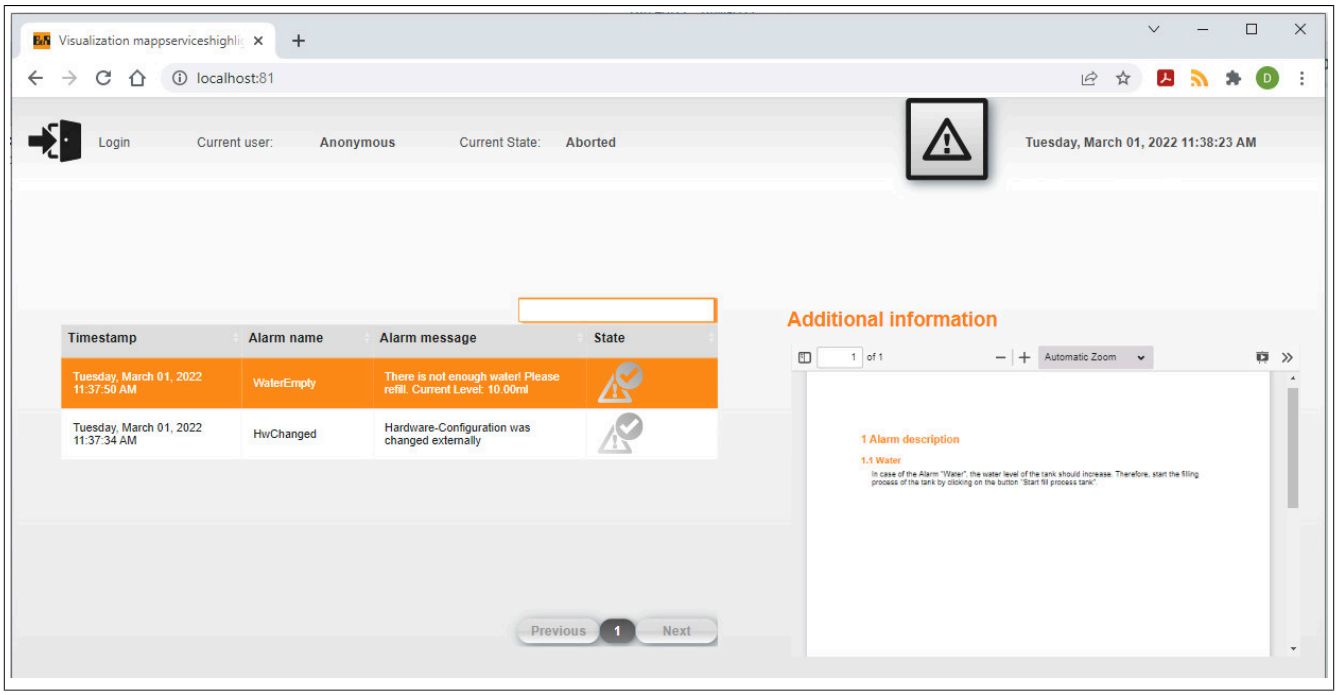


Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpAlarmXCore** für die Alarmverwaltung eingefügt.

Die Alarme werden über eine mapp View Visualisierung mithilfe des AlarmList-Widgets angezeigt.

Wird ein Alarm ausgelöst welcher eine zusätzliche Information beinhaltet, so wird das PDF, sobald der Alarm selektiert wird, am PDFViewer-Widget angezeigt.



1.5 Diagnose

Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

1.5.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp AlarmX auftreten können.

Wie kann ich Alarme in verschiedenen Gruppen verwalten?

Sollen Alarme nicht nur in einer MpAlarmXCore Konfiguration, sondern in verschiedenen verwaltet werden, können die Alarme in Alarmgruppen aufgeteilt werden.

Dazu wird die MpComGroup Konfiguration verwendet.

Wie solche Alarmgruppen in einer Maschine aufgebaut werden können, ist im Anwendungsfall Alarmverwaltung für Druckermaschinenlinie erklärt.

Wieso kann ein Alarm aktiviert werden welcher nicht in der Konfiguration ist?

Alle Alarme werden in der MpAlarmXCore Konfiguration verwaltet.

Wird ein Alarm über MpAlarmXSet aktiviert, welcher nicht in dieser Liste ist, so wird dieser Alarm automatisch erstellt und in der Alarmliste angezeigt. Der Alarm wird mit Standard-Alarmeinstellungen generiert (Edge Alarm, kein Alarmtext, ...)

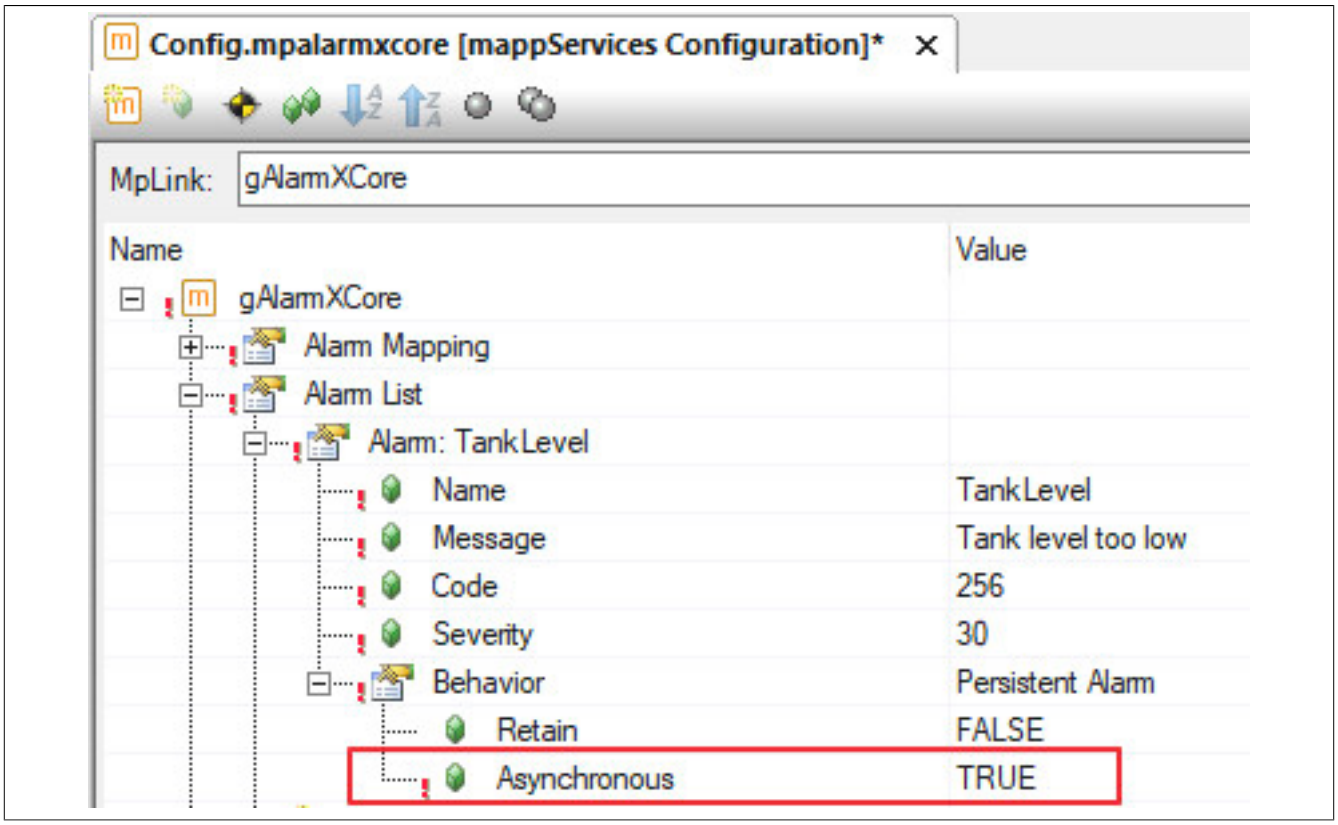
Der Alarm kann über MpAlarmXConfigAlarm weiter konfiguriert werden.

Wie kann ich die Alarmperformance verbessern?

Das Aufrufen vieler Alarme kann zeitintensiv sein. Werden beispielsweise mehrere hundert Alarme in einer Taskklasse aufgerufen, welche eine niedrige Zykluszeit aufweist, kann sich das schlecht auf die Performance auswirken. Um diese zu verbessern kann folgendes beachtet werden:

Asynchrone Alarme

Alarme können synchron oder asynchron verwaltet werden.



Synchrone Alarme werden im gleichen Kontext aktiviert, zurückgesetzt oder quittiert wo sie aufgerufen werden.

Asynchrone Alarme werden nicht im gleichen Kontext aktiviert, zurückgesetzt und quittiert wo sie aufgerufen wurden, sondern werden an MpAlarmXCore weitergeleitet und dort verarbeitet. Bei dieser Weiterleitung wird lediglich der Zeitstempel des Alarms mitgeliefert. Weitere Informationen wie Alarmsnippets oder in der Konfiguration definierte Alarm Mappings werden im Kontext von MpAlarmXCore aufgelöst. Das heißt die Information von Alarmsnippets sowie auch durch Alarm Mappings definierte Aktionen werden zeitverzögert, abhängig davon in welcher Taskklasse sich befindet, aufgelöst.

Um die Alarmperfomance zu verbessern können Alarme auf "Asynchronous = `TRUE`" gestellt werden. Dies kann für Alarme gewählt werden, welche nicht unmittelbar angezeigt werden müssen. Wird beispielsweise ein Alarm "WaterTankLevel" konfiguriert, welcher ausgelöst wird sobald ein kritisches Tank-Level erreicht wurde, kann dieser zeitverzögert angezeigt werden, da innerhalb von wenigen Millisekunden/Sekunden das Wasserlevel nicht um beispielsweise mehrere Meter steigt. Wird allerdings ein Alarm "EmergencyStop" konfiguriert, so sollte dieser im gleichen Kontext angezeigt werden in dem der Alarm aktiviert wurde. Hierbei sollte ein synchrones Alarmverhalten gewählt werden.

Die Alarme können dadurch in niedrigen Zykluszeiten bzw. höher prioren Taskklassen gesetzt werden und die eigentliche Verarbeitung des Alarms findet in einer nieder prioren Taskklasse statt, in welcher sich MpAlarmXCore befindet. Dadurch setzt sich das Aktivieren, Zurücksetzten und Quittieren von Alarmen nicht negativ auf die Performace aus.

Alarm über MpAlarmXControl aufrufen

Wird die Funktion MpAlarmXSet verwendet um einen Alarm zu aktivieren, so wird, sobald die Funktion aufgerufen wird, die Alarmkonfiguration gelesen. Es werden definierte Alarmeigenschaften, Alarmsnippets, Alarm Mappings usw. aufgelöst. Dies geschieht bei jedem Funktionsaufruf. Hiervon ausgenommen sind asynchrone Alarme. In diesem Fall werden Alarmsnippets, Alarm Mappings usw. von MpAlarmXCore aufgelöst.

Beim Verwenden des Funktionsbausteins MpAlarmXAlarmControl wird die Alarmkonfiguration einmalig ausgelesen. Dies geschieht während des Aktivier-Vorgangs ("Enable = `TRUE`") des Funktionsbausteins. Dadurch ist das Aktivieren, Zurücksetzen sowie Quittieren des Alarms weniger zeitintensiv als beim Verwenden von Funktionen. Wird die Alarmkonfiguration zur Laufzeit geändert, so wird die Änderung automatisch an MpAlarmXAlarmControl weitergeleitet.

1.6 Bibliotheken

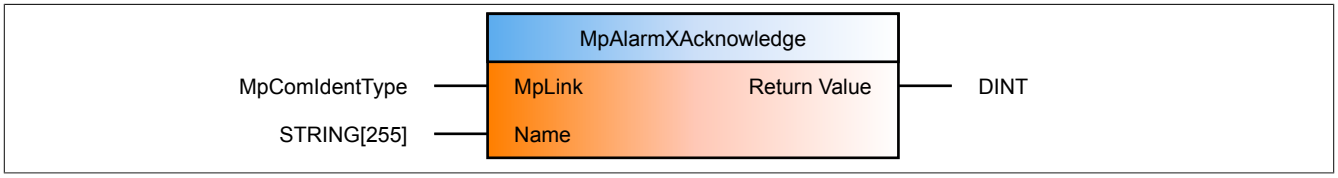
1.6.1 Funktionsbausteine

Unterstützt ab	Name	Beschreibung
5.0.0	MpAlarmXCore	Dieser Funktionsbaustein sammelt und verwaltet Alarme
5.0.0	MpAlarmXListUI	Dieser Funktionsbaustein wird benützt, um eine Verbindung zwischen MpAlarmX und Visual Components 4 zu schaffen. MpAlarmXListUI erstellt eine Liste mit aktiven bzw. nicht quittierten Alarmen
5.0.0	MpAlarmXSet	Diese Funktion aktiviert benutzerdefinierte Alarme
5.0.0	MpAlarmXReset	Diese Funktion setzt benutzerdefinierte Alarme wieder zurück
5.0.0	MpAlarmXResetID	Diese Funktion setzt benutzerdefinierte Alarme wieder zurück über deren Instanz ID
5.0.0	MpAlarmXAcknowledge	Diese Funktion kann angegebene Alarme quittieren
5.0.0	MpAlarmXAcknowledgeID	Diese Funktion kann angegebene Alarme quittieren über deren Instanz ID
5.0.0	MpAlarmXCheckState	Diese Funktion kontrolliert in welchem Status sich der angegebene Alarm befindet
5.0.0	MpAlarmXCheckStateID	Diese Funktion kontrolliert in welchem Status sich der angegebene Alarm befindet über dessen Instanz ID
5.0.0	MpAlarmXCheckReaction	Diese Funktion prüft, ob eine bestimmte Reaktion aktiv ist
5.0.0	MpAlarmXConfigAlarm	Dieser Funktionsbaustein ermöglicht das Erstellen und Konfigurieren eines Alarms zur Laufzeit
5.0.0	MpAlarmXConfigMapping	Dieser Funktionsbaustein ermöglicht das Erstellen und Konfigurieren des Alarm Mappings zur Laufzeit
5.0.0	MpAlarmXHistory	Mit diesem Funktionsbaustein kann die Alarm Historie, auf ein angegebenes Speichermedium, exportiert werden
5.0.0	MpAlarmXHistoryUI	Dieser Funktionsbaustein wird benützt, um eine Verbindung zwischen MpAlarmX und Visual Components 4 zu schaffen. MpAlarmXHistoryUI erstellt eine Liste mit allen Statusänderungen Alarmen
5.3.0	MpAlarmXQuery	Mithilfe von MpAlarmXQuery kann eine Abfrage gestartet werden. Dadurch kann nach bestimmten Alarmen in der Alarmliste oder in der Alarmhistorie gesucht werden.
5.5.0	MpAlarmXClearHistory	Mithilfe dieser Funktion wird die Alarmhistorie gelöscht.
5.7.0	MpAlarmXAcknowledgeAll	Mithilfe von MpAlarmXAcknowledgeAll können alle Alarme quittiert werden.
5.16.0	MpAlarmXAlarmControl	Über diesen Funktionsbaustein kann ein Alarm kontrolliert werden. Es kann ein Alarm aktiviert, zurückgesetzt sowie quittiert werden.

1.6.1.1 MpAlarmXAcknowledge

Diese Funktion kann angegebene Alarme quittieren.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration)
IN	Name	`STRING[255]`	Name des Alarms
OUT	Return Value	`DINT`	Rückgabewert der Funktion

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.1.1 Beschreibung

Diese Funktion verwendet den MpLink einer MpAlarmXCore Konfiguration.

> **Soll ein Alarm quittiert werden, welcher durch die Aktion "Escalate Alarm" zu einer übergeordneten Gruppe hin eskaliert wurde, so muss zum quittieren des Alarms der mapp Link des übergeordneten Alarmsystems angegeben werden. Für mehr Information siehe Alarm Mapping.**

Beim Ausführen dieser Funktion wird ein der über den Parameter "Name" definierte Alarm quittiert.

Es gilt zu beachten, dass Persistente Alarme welche einen Punkt im Alarmnamen beinhalten, wie zum Beispiel "Temp.High", nicht mit MpAlarmXAcknowledge quittiert werden können.

"Return value" liefert den Rückgabewert der Funktion. Dabei können folgende Rückgabewerte auftreten:

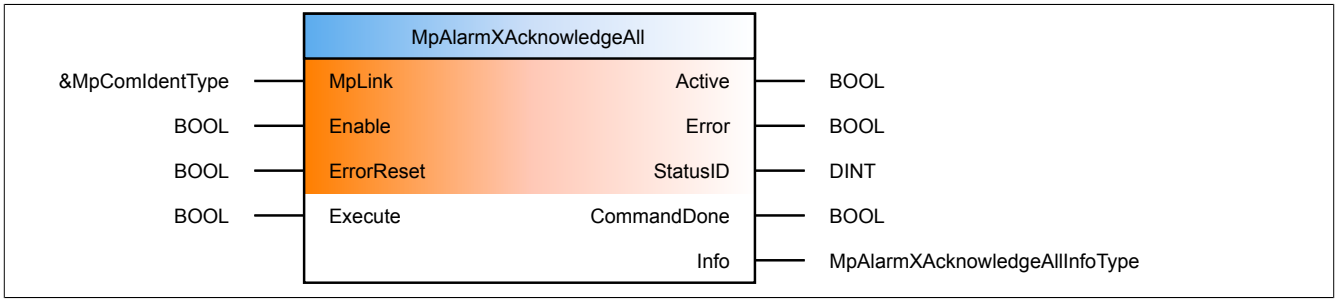
- mpALARMX_ERR_ALARM_NOT_ACTIVE
- mpALARMX_ERR_NAME_EMPTY
- mpALARMX_ERR_NAME_NULL
- mpALARMX_ERR_MPLINK_NULL
- mpALARMX_ERR_MPLINK_INVALID
- mpALARMX_ERR_MPLINK_CORRUPT

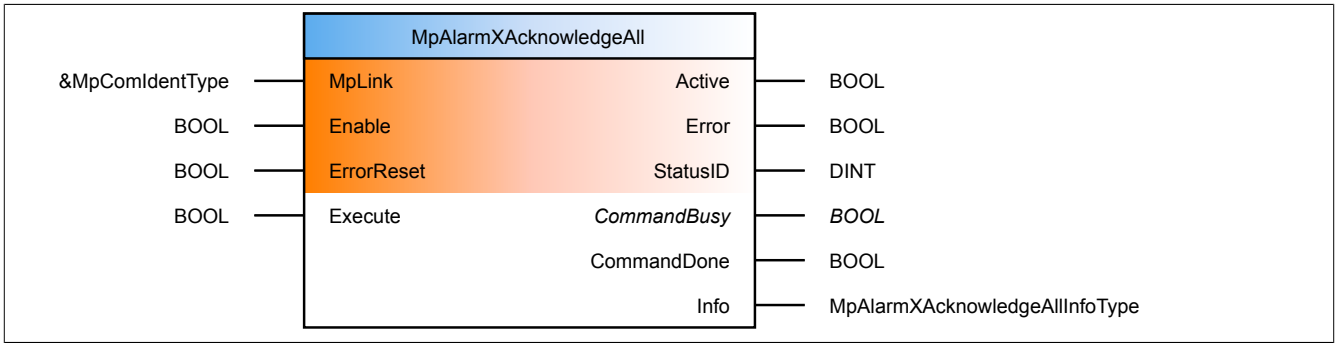
1.6.1.2 MpAlarmXAcknowledgeAll

Mithilfe von MpAlarmXAcknowledgeAll können alle Alarme quittiert werden.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Execute	`BOOL`	Befehl um alle Alarme zu quittieren.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpAlarmXAcknowledgeAllInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.2.1 Beschreibung

Es wird der gleiche MpLink wie an MpAlarmXCore angegeben.

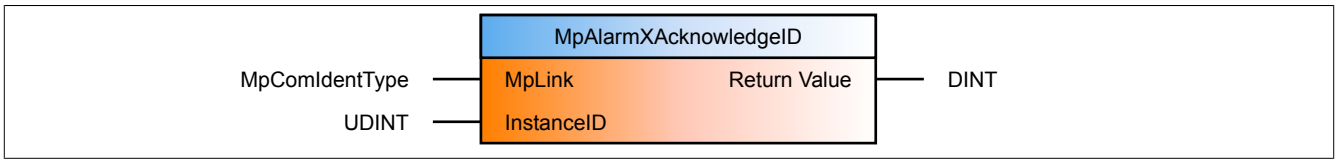
Mithilfe von "Execute = `TRUE`" können alle Alarme, welche in MpAlarmXCore vorhanden sind, quittiert werden.

In der Info-Struktur kann über "UnacknowledgedAlarms" die Anzahl der nicht quittierten Alarme herausgefunden werden.

1.6.1.3 MpAlarmXAcknowledgeID

Diese Funktion kann angegebene Alarme quittieren über deren Instanz ID.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration)
IN	InstanceID	`UDINT`	Instanz ID des Alarms
OUT	Return Value	`DINT`	Rückgabewert der Funktion

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.3.1 Beschreibung

Diese Funktion verwendet den MpLink einer MpAlarmXCore Konfiguration.

> **Soll ein Alarm quittiert werden, welcher durch die Aktion "Escalate Alarm" zu einer übergeordneten Gruppe hin eskaliert wurde, so muss zum quittieren des Alarms der mapp Link des übergeordneten Alarmsystems angegeben werden. Für mehr Information siehe Alarm Mapping.**

Beim Ausführen dieser Funktion wird ein der über den Parameter "InstanceID" definierte Alarm quittiert. Die Instanz ID ist der Rückgabewert der Funktion MpAlarmXSet.

> **Das Arbeiten mit ID Funktionen ist nur dann notwendig, wenn mit mehrfach instanzierten Alarmen gearbeitet wird.**

Beim Ausführen dieser Funktion wird ein der über den Parameter "Name" definierte Alarm quittiert.

"Return value" liefert den Rückgabewert der Funktion. Dabei können folgende Rückgabewerte auftreten:

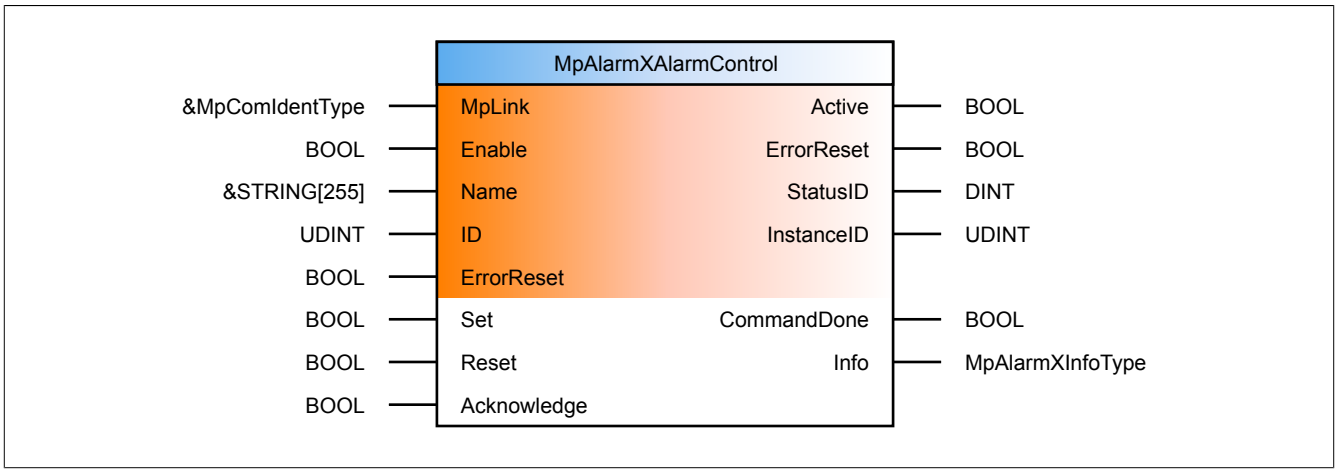
- mpALARMX_ERR_ALARM_NOT_ACTIVE
- mpALARMX_ERR_NAME_EMPTY
- mpALARMX_ERR_NAME_NULL
- mpALARMX_ERR_MPLINK_NULL
- mpALARMX_ERR_MPLINK_INVALID
- mpALARMX_ERR_MPLINK_CORRUPT

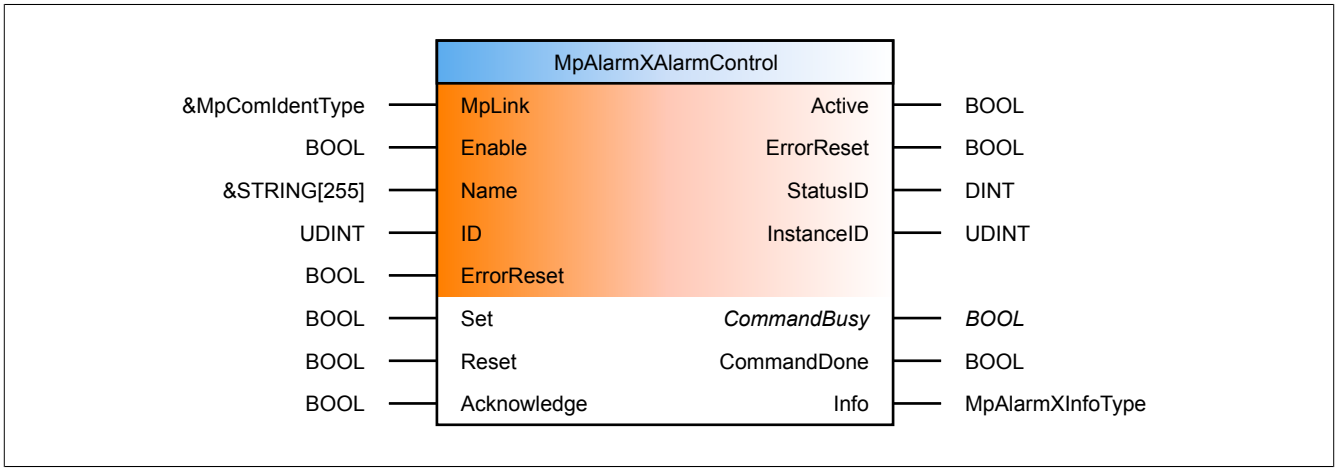
1.6.1.4 MpAlarmXAlarmControl

Über diesen Funktionsbaustein kann ein Alarm kontrolliert werden. Es kann ein Alarm aktiviert, zurückgesetzt sowie quittiert werden.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	Name	`Zeiger auf STRING[255]`	Name des Alarms.
IN	ID	`UDINT`	Instanz ID, welche zur Identifizierung des Alarms dient.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Set	`BOOL`	Befehl um Alarm zu aktivieren.
IN	Reset	`BOOL`	Befehl um Alarm zurückzusetzen.
IN	Acknowledge	`BOOL`	Befehl um Alarm zu quittieren.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	InstanceID	`UDINT`	Instanz ID eines zuletzt aktivierten, zurückgesetzten oder quittierten Alarms.
OUT	CommandBusy	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpAlarmXControlInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.4.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpAlarmXCore Konfiguration verwendet. Mithilfe des Funktionsbausteins wird ein Alarm kontrolliert. Der Alarmname wird dabei über "Name" definiert.

Wird der Funktionsbaustein durch den Eingang "Enable" aktiviert, wird die MpAlarmXCore Konfiguration gelesen. Dadurch weiß der Funktionsbaustein welche Alarme existieren, welche Alarmeigenschaften definiert wurden, ob Alarmsnippets, Alarm Mappings usw. verwendet wurde. Wird zur Laufzeit die MpAlarmXCore Konfiguration geändert, so wird die Änderung automatisch an den Funktionsbaustein weitergeleitet.

Durch "Set", "Reset" und "Acknowledge" kann ein Alarm aktiviert, zurückgesetzt und quittiert werden. Es können auch mehrere Befehle auf einmal gegeben werden, die Reihenfolge wird vom Funktionsbaustein geregelt. Wurden mehrere Befehle gesetzt und alle abgearbeitet, so wird der Ausgange "CommandDone = `TRUE`".

Abhängig davon, ob es sich um einen synchronen oder asynchronen Alarm handelt, wird der Befehl sofort ausgeführt, oder an MpAlarmXCore weitergeleitet.

Wird ein Alarm durch das Alarm Mapping an eine übergeordnete Gruppe weitergeleitet (Escalate Alarm), so wird für das Quittieren der MpLink verwendet, mit welchem der Alarm auch aktiviert wurde. Es wird nicht der MpLink der übergeordneten Gruppe verwendet.

Ob ein Kommando erfolgreich ausgeführt wurde, wird durch den Ausgangsparameter "CommandDone = `TRUE`" angezeigt.

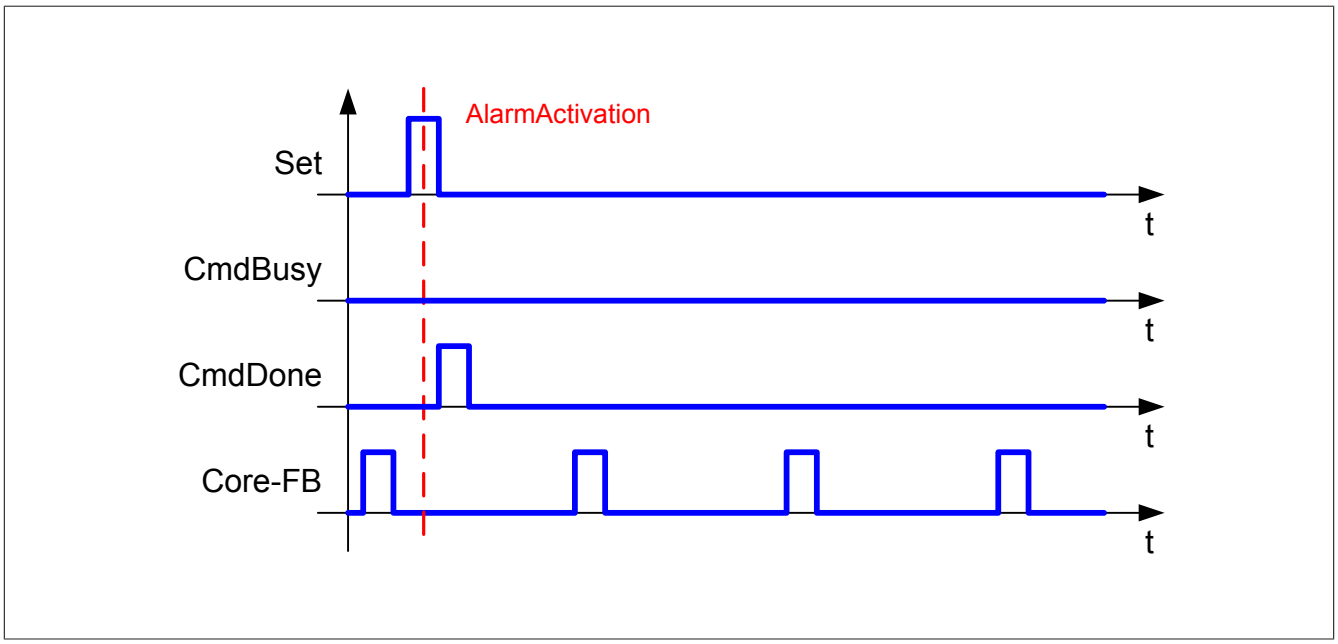
Die Instanz-ID eines Alarms wird am Ausgang "InstanceID" angezeigt. Ob ein Alarm mehrfach instanziert werden kann, wird über die Konfigurations-Eigenschaft "Multiple instances = `TRUE`" geregelt. Wurden mehrere Instanzen eines Alarms erzeugt, so können die einzelnen Instanzen mithilfe des Eingangs "ID" kontrolliert werden. Das heißt einzelne Alarm-Instanzen können aktiviert, zurückgesetzt sowie quittiert werden.

1.6.1.4.2 Zeitdiagramme

Die Zeitdiagramme beinhalten das Verhalten eines synchronen bzw. asynchronen Alarms beim Aktivieren. Das Verhalten ist identisch mit den Befehlen "Reset" und "Acknowledge".

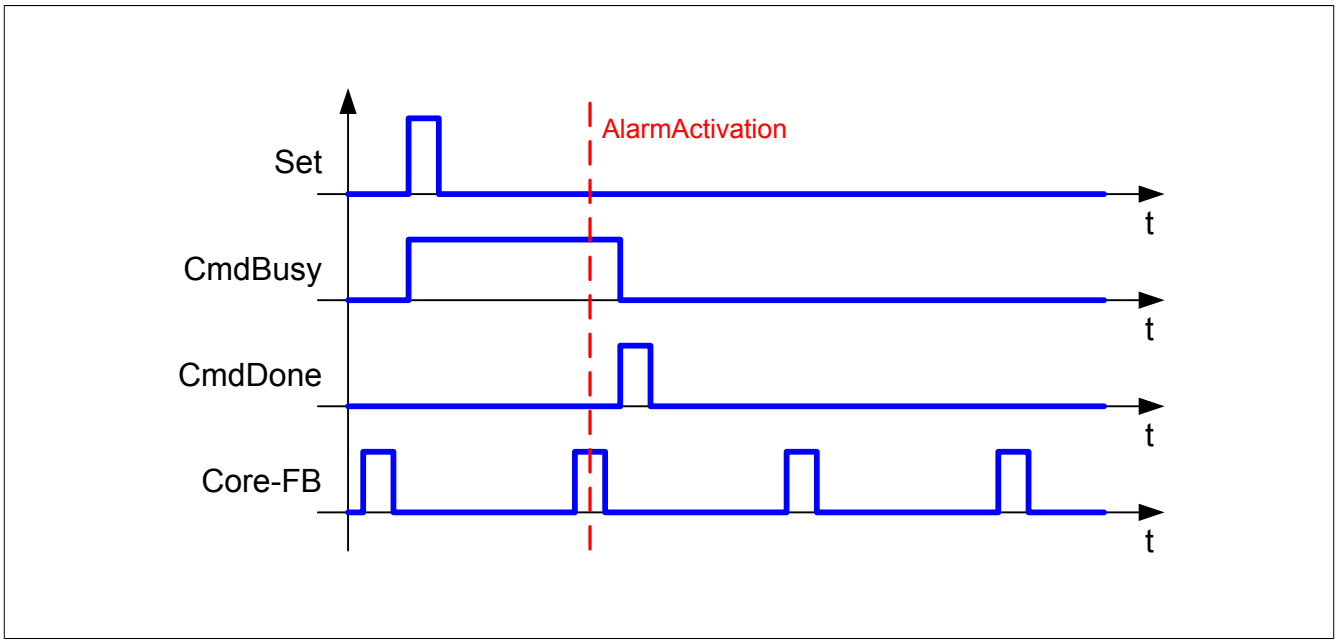
Aktivieren eines synchronen Alarms

In diesem Zeitdiagram wird ein synchroner Alarm gesetzt. Sobald der "Set" Befehl gestartet wurde, wird der Alarm aktiviert. "CommandDone = `TRUE`" zeigt an, dass der Befehl erfolgreich ausgeführt wurde.



Aktivieren eines asynchronen Alarms

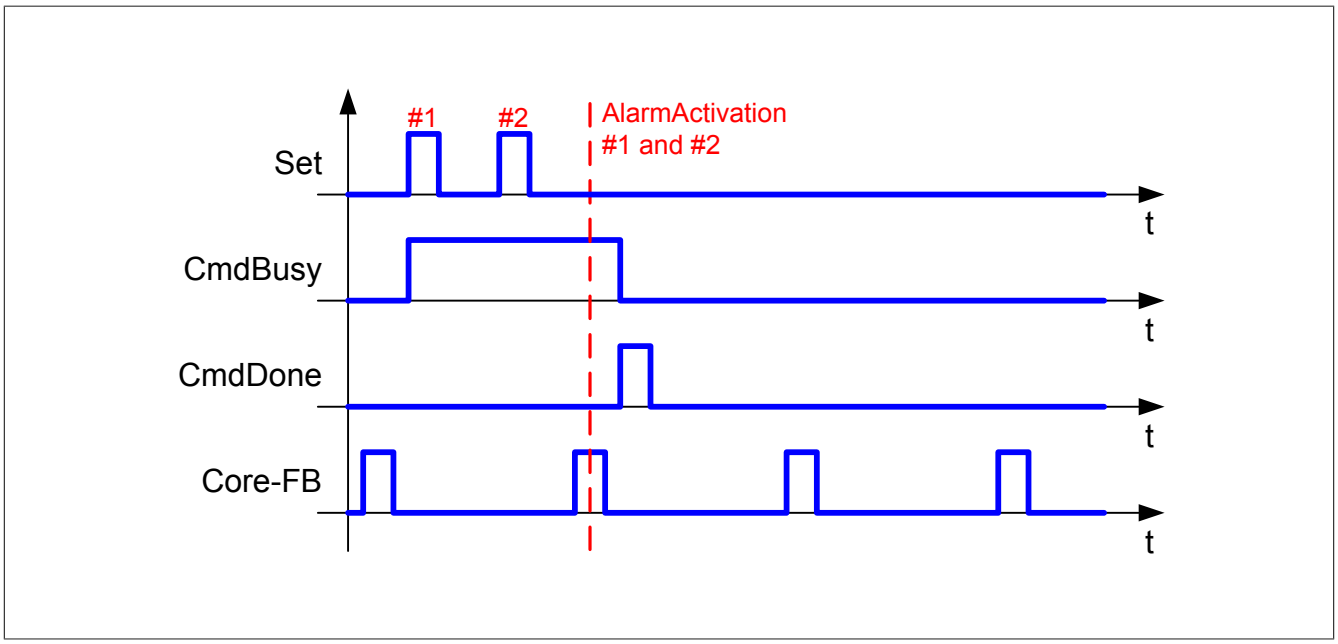
In diesem Zeitdiagramm wird ein asynchroner Alarm gesetzt. Sobald der Befehl durch "Set" gestartet wurde, ist der Ausgang "CommandBusy" so lange `TRUE`, bis dass der Alarm durch MpAlarmXCore aktiviert wird. "CommandDone" zeigt die erfolgreiche Aktivierung des Alarms.



Aktivieren mehrerer asynchroner Alarme/ Setzten von mehreren Befehlen

In diesem Zeitdiagramm werden mehrere asynchrone Alarme durch den Set-Befehl gesetzt. Der Ausgang "CommandBusy" bleibt so lange auf `TRUE`, bis dass die Alarme durch MpAlarmXCore aktiviert werden. "CommandDone" zeigt die erfolgreiche Aktivierung der Alarme.

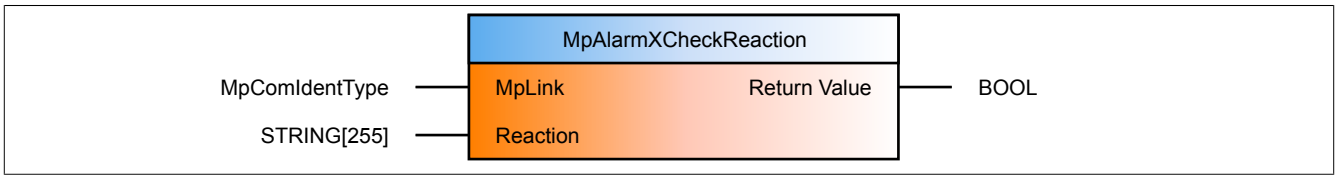
Dieses Verhalten gilt auch beim Setzten von mehreren Befehlen, wie "Set = `TRUE`" und "Acknowledge = `TRUE`". Auch in diesem Fall bleibt "CommandBusy" so lange auf `TRUE`, bis dass alle Befehle ausgeführt wurden. Wurden alle Befehle ausgeführt, so wird dies durch "CommandDone = `TRUE`" angezeigt.



1.6.1.5 MpAlarmXCheckReaction

Diese Funktion prüft, ob eine bestimmte Reaktion aktiv ist.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration)
IN	Reaction	`STRING[255]`	Name der Reaktion
OUT	Return Value	`BOOL`	Status der Reaktion. Wurde ein Alarm getriggert ist der Ausgang TRUE, andernfalls FALSE

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.5.1 Beschreibung

Es wird der MpLink einer MpAlarmXCore Konfiguration verwendet.

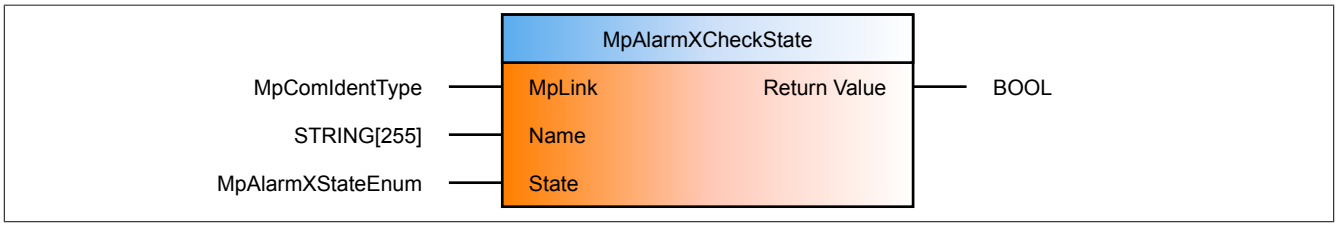
Diese Funktion prüft, ob eine bestimmte Rekation aktiv ist. Dies wird durch den Rückgabewert angezeigt. Ist der Wert *TRUE* ist die Reaktion aktiv.

Die zu prüfende Reaktion wird über den Parameter "Reaction" übergeben.

1.6.1.6 MpAlarmXCheckState

Diese Funktion kontrolliert in welchem Status sich der angegebene Alarm befindet.

Funktion



Schnittstelle

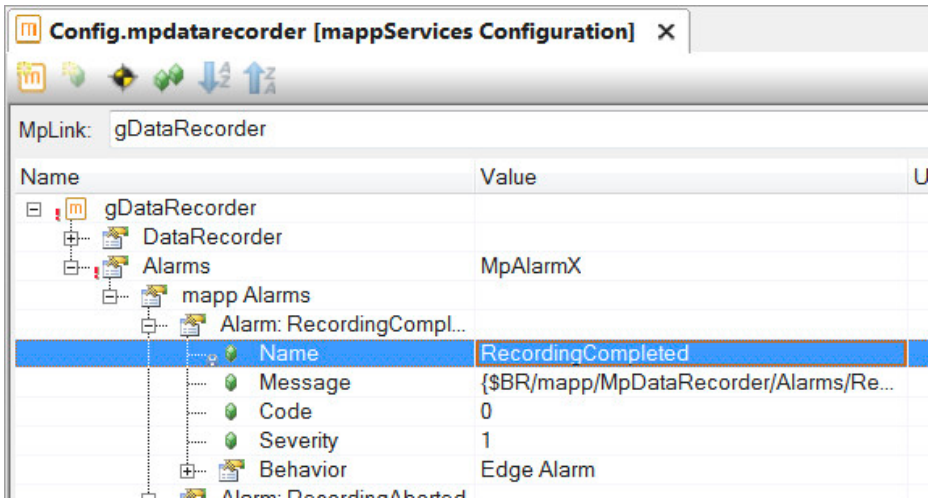
I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp (Abhängig davon welcher Alarm geprüft werden soll, muss der jeweilige MpLink der gewünschten Konfiguration angegeben werden. Das heißt es kann zum Beispiel ein Alarm von mapp Tweet oder mapp Data geprüft werden.)
IN	Name	`STRING[255]`	Name der zur Identifizierung des Alarms dient
IN	State	`MpAlarmXStateEnum`	Angabe des zu prüfenden Status
OUT	ReturnValue	`BOOL`	Kontrolliert ob sich derzeit der Alarm in den angegebenen Status befindet

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

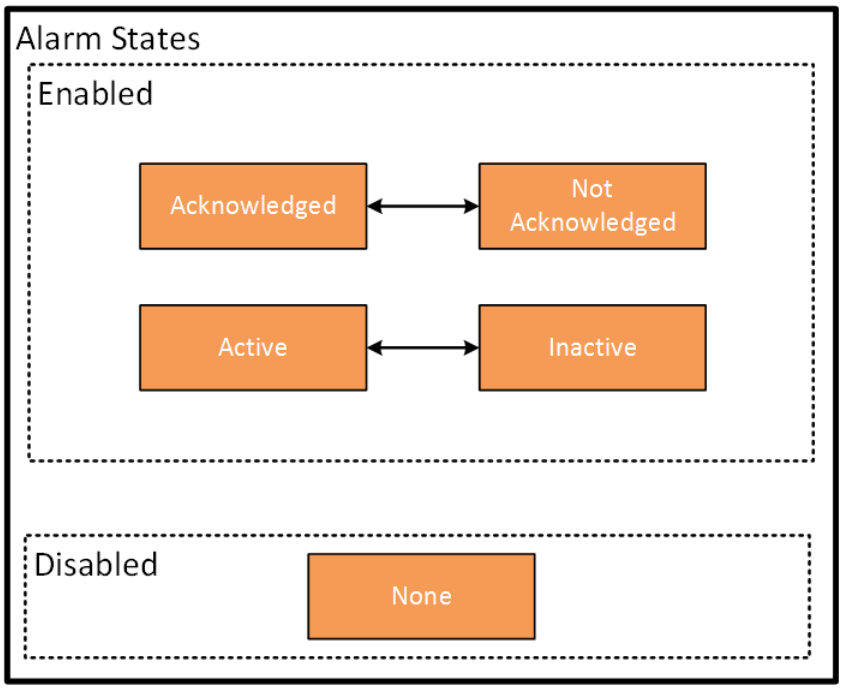
1.6.1.6.1 Beschreibung

Diese Funktion überprüft, ob sich der angegebene Alarm in einem bestimmten Status befindet. Abhängig davon welcher Alarm geprüft werden soll, können unterschiedliche MpLinks angegeben werden. Soll zum Beispiel ein Alarm in mapp Tweet geprüft werden, dann kann der MpLink einer MpTweetCore Konfiguration angegeben werden. Die Alarmnamen können in der jeweiligen mapp Konfiguration gefunden werden, wie zum Beispiel in der mapp Data Konfiguration:



Über den Eingang "State" wird definiert, welcher Zustand des Alarms überprüft werden soll. Befindet sich der Alarm in dem angegebenen Status, wird der Rückgabewert der Funktion *TRUE*.

Das folgende Modell zeigt die verschiedenen Zustände eines Alarms:

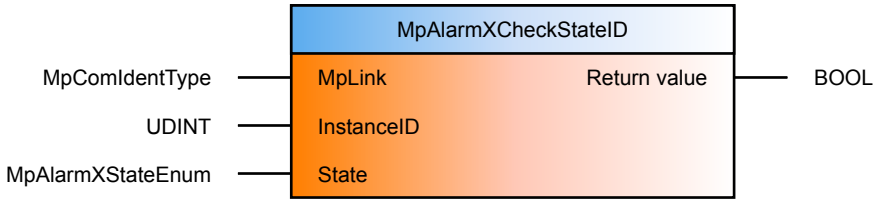


Weitere Details zum den Status eines Alarms sind im Abschnitt Alarm Grundkonzept zu finden.

1.6.1.7 MpAlarmXCheckStateID

Diese Funktion kontrolliert in welchem Status sich der angegebene Alarm befindet.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration)
IN	InstanceID	`UDINT`	Instanz ID, welche zur Identifizierung des Alarms dient
IN	State	`MpAlarmXStateEnum`	Angabe des zu prüfenden Status
OUT	ReturnValue	`BOOL`	Kontrolliert ob sich derzeit der Alarm in den angegebenen Status befindet

mapp Konzept

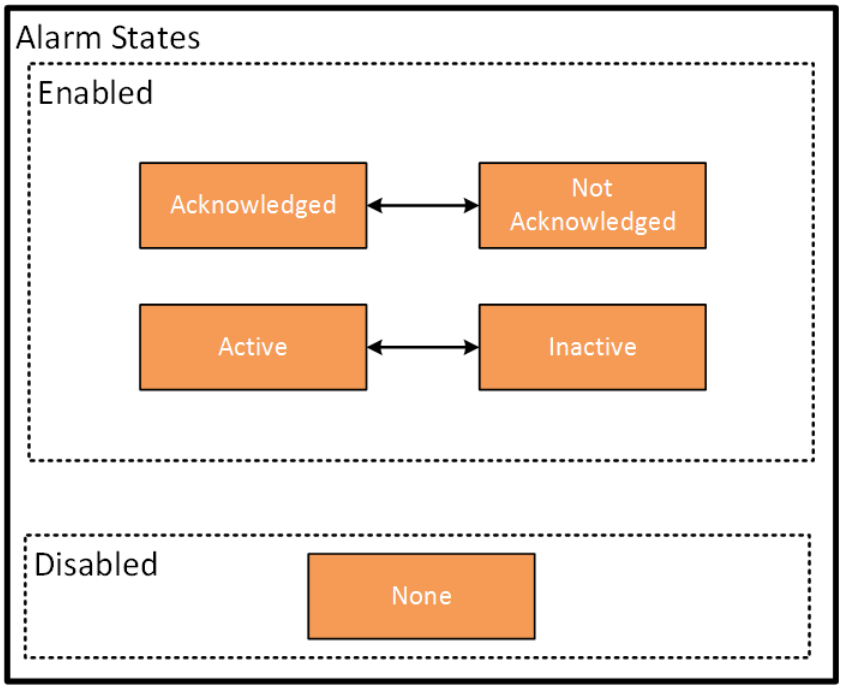
Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.7.1 Beschreibung

Diese Funktion überprüft, ob sich der angegebene Alarm in einem bestimmten Status befindet. Es wird der derselbe MpLink wie von MpAlarmXCore verwendet.

Über den Eingang "State" wird definiert, welcher Zustand des Alarms überprüft werden soll. Befindet sich der Alarm in dem angegebenen Status, wird der Rückgabewert der Funktion *TRUE*.

Das folgende Modell zeigt die verschiedenen Zustände eines Alarms:



Weitere Details zum den Status eines Alarms sind im Abschnitt Alarm Grundkonzept zu finden.

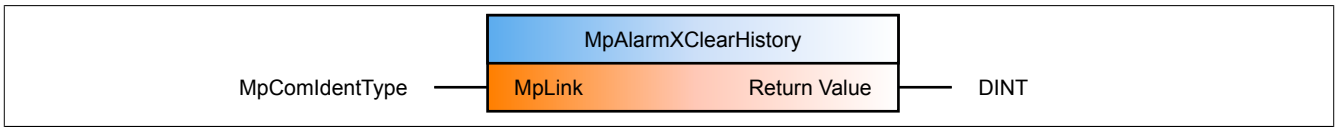
Der Alarm wird über die Instanz ID am Eingang "InstanceID" angegeben. Die Instanz ID ist der Rückgabewert, welcher beim Setzen des Alarms über MpAlarmXSet zurückgegeben wurde.

> **Das Arbeiten mit ID Funktionen ist nur dann notwendig, wenn mit mehrfach instanzierten Alarmen gearbeitet wird.**

1.6.1.8 MpAlarmXClearHistory

Mithilfe dieser Funktion wird die Alarmhistorie gelöscht.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp (MpLink einer MpAlarmXHistory Konfiguration)
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.8.1 Beschreibung

Diese Funktion verwendet den MpLink einer MpAlarmXHistory Konfiguration.

Sobald die Funktion aktiv ist, wird die gesamte Alarmhistorie gelöscht.

"Return value" liefert den Rückgabewert der Funktion. Dabei können folgende Rückgabewerte auftreten:

- mpALARMX_ERR_MPLINK_NULL
- mpALARMX_ERR_MPLINK_INVALID
- mpALARMX_ERR_MPLINK_CORRUPT
- mpALARMX_ERR_MPLINK_CHANGED

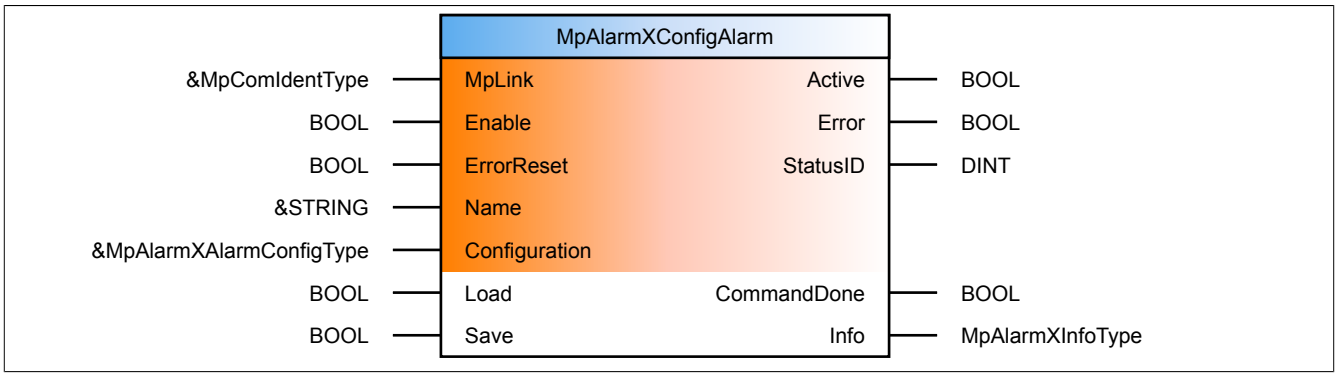
1.6.1.9 MpAlarmXConfigAlarm

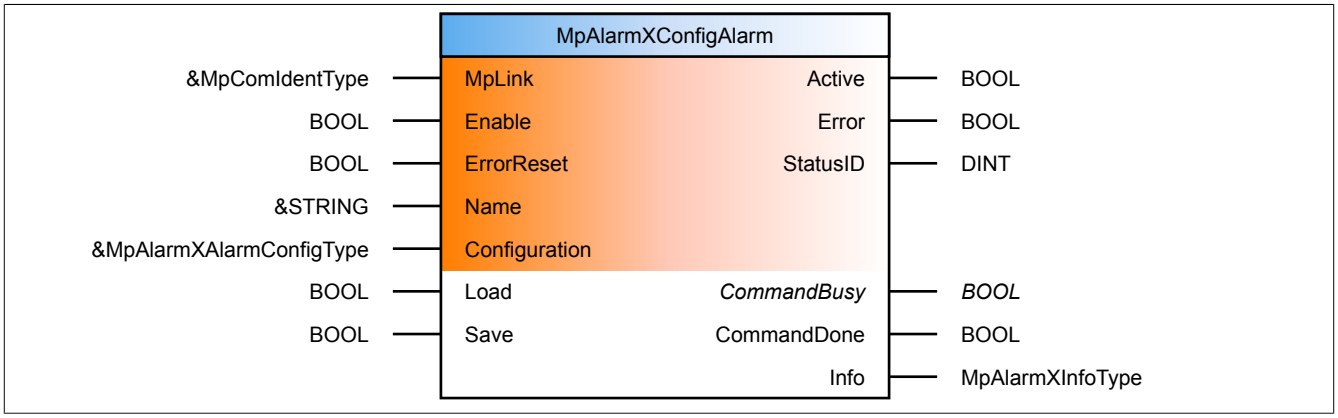
Dieser Funktionsbaustein ermöglicht das Erstellen und Konfigurieren eines Alarms zur Laufzeit.

Es können auch mapp Alarme anderer mapp Komponenten konfiguriert werden. Für mehr Information siehe hier.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration). Sollen Alarme anderer mapp Komponenten konfiguriert werden, so wird der mapp Link der gewünschten Komponente angegeben, wie beispielsweise der mapp Link einer MpRecipeXml Konfiguration.
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.

I/O	Parameter	Datentyp	Beschreibung
IN	Name	`Zeiger auf STRING[255]`	Name des Alarms, welcher erzeugt bzw. konfiguriert werden soll.
IN	Configuration	`Zeiger auf MpAlarmXAlarm-ConfigType`	Struktur, in der die Konfiguration festgelegt wird.
IN	Load	`BOOL`	Lädt die Konfiguration der Komponente.
IN	Save	`BOOL`	Speichert die Konfiguration der Komponente.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpAlarmXInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.9.1 Beschreibung

Für diesen Funktionsbaustein wird derselbe MpLink wie für MpAlarmXCore verwendet. Mithilfe des Funktionsbausteins sind Änderungen der Konfiguration zur Laufzeit möglich.

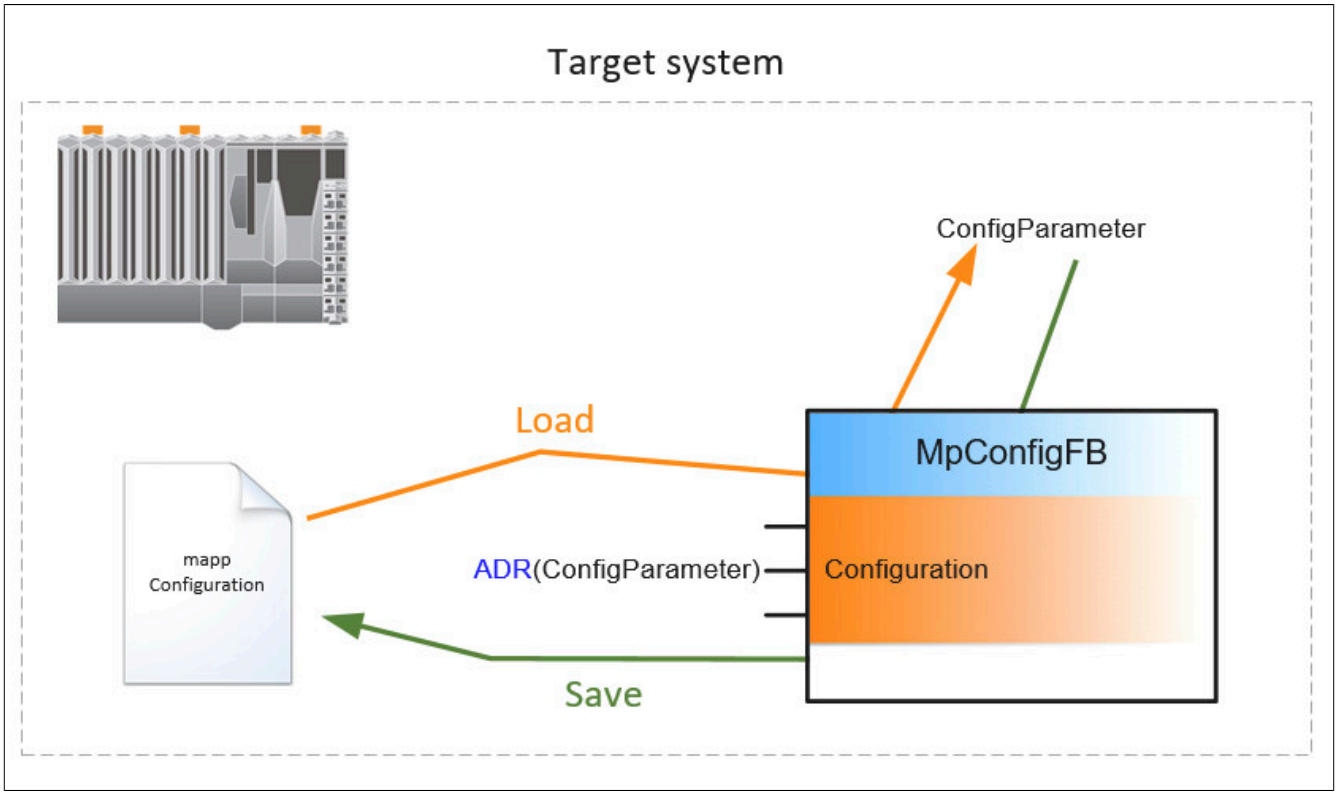
> **Bei Verwendung von zum Beispiel Umlauten (ä, ü,...) sollte die Zusätzliche Information im Abschnitt Best Practise beachtet werden.**

Speichern und Laden der Konfiguration

Der Parameter "Configuration" ermöglicht es die Konfiguration, bzw. einen Teil der Konfiguration zu bearbeiten. Welcher Alarm editiert oder erstellt werden soll wird über den Eingang "Name" festgelegt. Das Erstellen, bzw. Edititieren kann zur Laufzeit erfolgen. Änderungen werden sofort übernommen.

Der Befehl "Load" wird verwendet, um die aktuell konfigurierten Parameter auszulesen. Dabei wird die aktuell aktive Konfiguration der Steuerung in die PV am Eingang "Configuration" geschrieben. Die Daten in der Struktur werden dabei überschrieben. Danach können die Werte beliebig editiert werden.

Über den Befehl "Save" werden die Werte aus der "Configuration" Struktur auf die aktive Konfiguration übernommen und, wenn möglich, direkt verwendet.



Um die Konfigurations-Parameter in der Appliaktion korrekt anzupassen, wird folgendermaßen vorgegangen:

1. Befehl "Load = `TRUE`"

Die aktuelle Konfiguration wird vom Zielsystem in die Konfigurations-Struktur geladen. Der Vorgang ist mit "CommandDone = TRUE" abgeschlossen.

2. Befehl "Load = `FALSE`"

Rücksetzen des "Load" Befehls. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

3. Parametrieren der Konfigurationsstruktur

Die Parameter der Konfigurationsstruktur können nun wie gewünscht angepasst werden.

4. Befehl "Save = `TRUE`" setzen

Die geänderte Konfigurationsstruktur wird auf das Zielsystem übertragen. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

5. Befehl "Save = `FALSE`" setzen

Rücksetzen des "Save" Befehls. Der Vorgang ist mit "CommandDone = `FALSE`" abgeschlossen. Um zu kontrollieren ob die Parameter erfolgreich geändert worden sind, können die Schritte 1 und 2 erneut ausgeführt werden.

Konfigurieren der MpAlarmXAlarmConfigType Struktur

Über diese Konfiguration können bereits existierende Alarme umkonfiguriert werden, bzw. neue Alarme erstellt werden. Folgende Alarm-Eigenschaften können dabei beeinflusst werden:

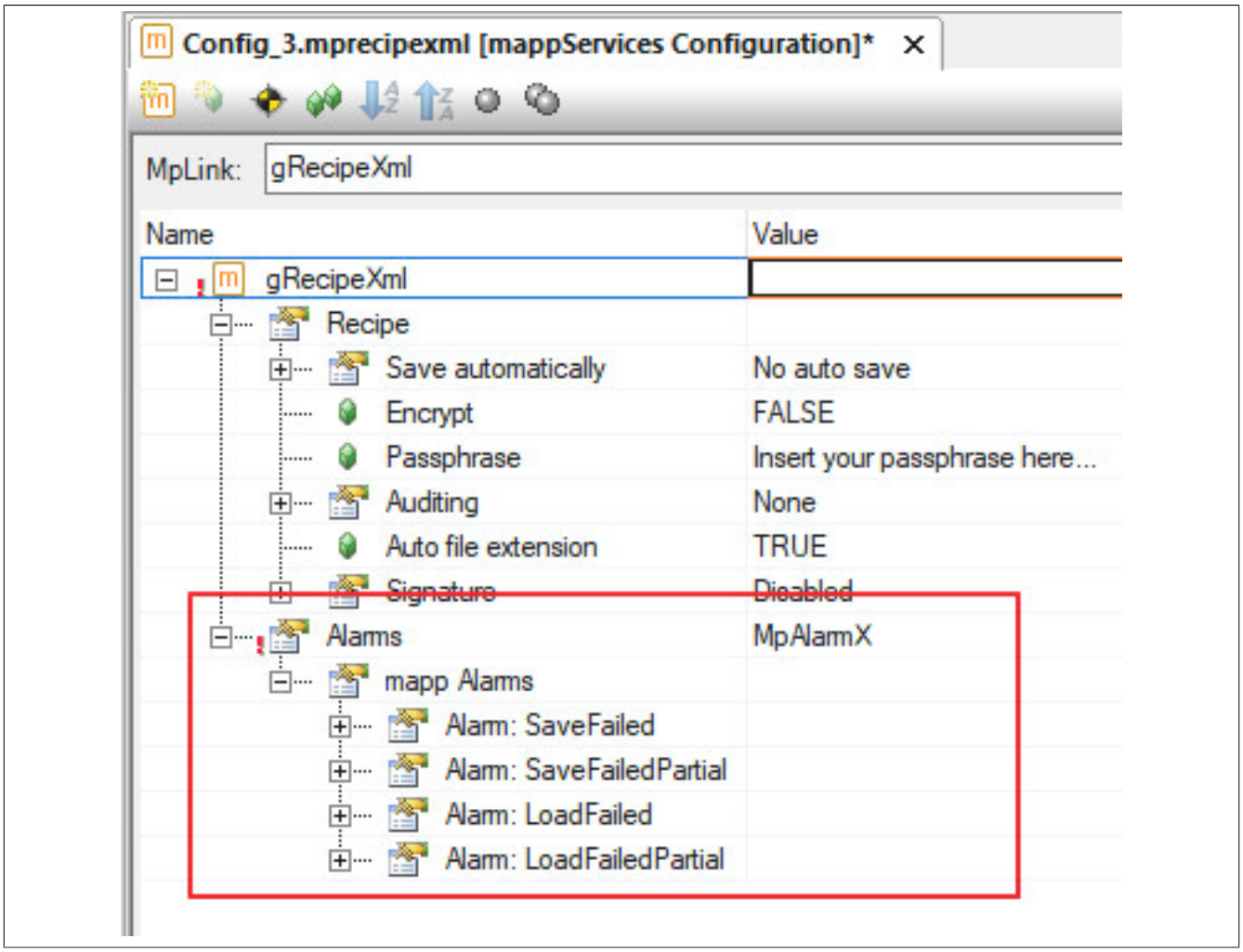
Initialize AlarmConfiguration

Name	Type
AlarmConfiguration	MpAlarmXAlarmConfigType
Message	STRING[255]:=''
Code	UDINT:=0
Severity	UDINT:=1
Behavior	MpAlarmXBehaviorType
AutoReset	BOOL:=FALSE
Acknowledge	MpAlarmXAcknowledgeConfigEnum:=mpALARMX_ACK_REQ
MultipleInstances	BOOL:=FALSE
ReactionUntilAcknowledged	BOOL:=TRUE
HistoryReport	MpAlarmXHistoryReportType
InactiveToActive	BOOL:=TRUE
ActiveToInactive	BOOL:=TRUE
UnacknowledgedToAcknowledged	BOOL:=TRUE
AcknowledgedToUnacknowledged	BOOL:=TRUE
Reactivation	BOOL:=TRUE
DataUpdate	MpAlarmXUpdateBehaviorType
Activation	MpAlarmXActivationUpdateType
Timestamp	BOOL
Snippet	BOOL
Disable	BOOL:=FALSE
AdditionalInformation1	STRING[255]:=''
AdditionalInformation2	STRING[255]:=''

Details zu den einzelnen Eigenschaften sind im Abschnitt Alarmliste zu finden. Über den Datentyp der Konfigurationsstruktur ist für jeden Parameter beschrieben, welcher Wert notwendig ist, um einen Edge oder Persistent Alarm nachzubilden. Wird ein bestehender bzw. neuer Alarm konfiguriert, so wird der Alarm immer mit dem Alarmverhalten "User defined" abgespeichert. Über die Konfigurationsstruktur können keine Monitored-Alarme erstellt werden.

Konfigurieren anderer mapp Alarme

Über den Funktionsbaustein ist es auch möglich die Alarme anderer mapp Komponenten, wie mapp Recipe zu bearbeiten. Wichtig hierbei ist, dass als Alarmsystem in der Konfiguration der mapp Komponente "MpAlarmX" ausgewählt ist:

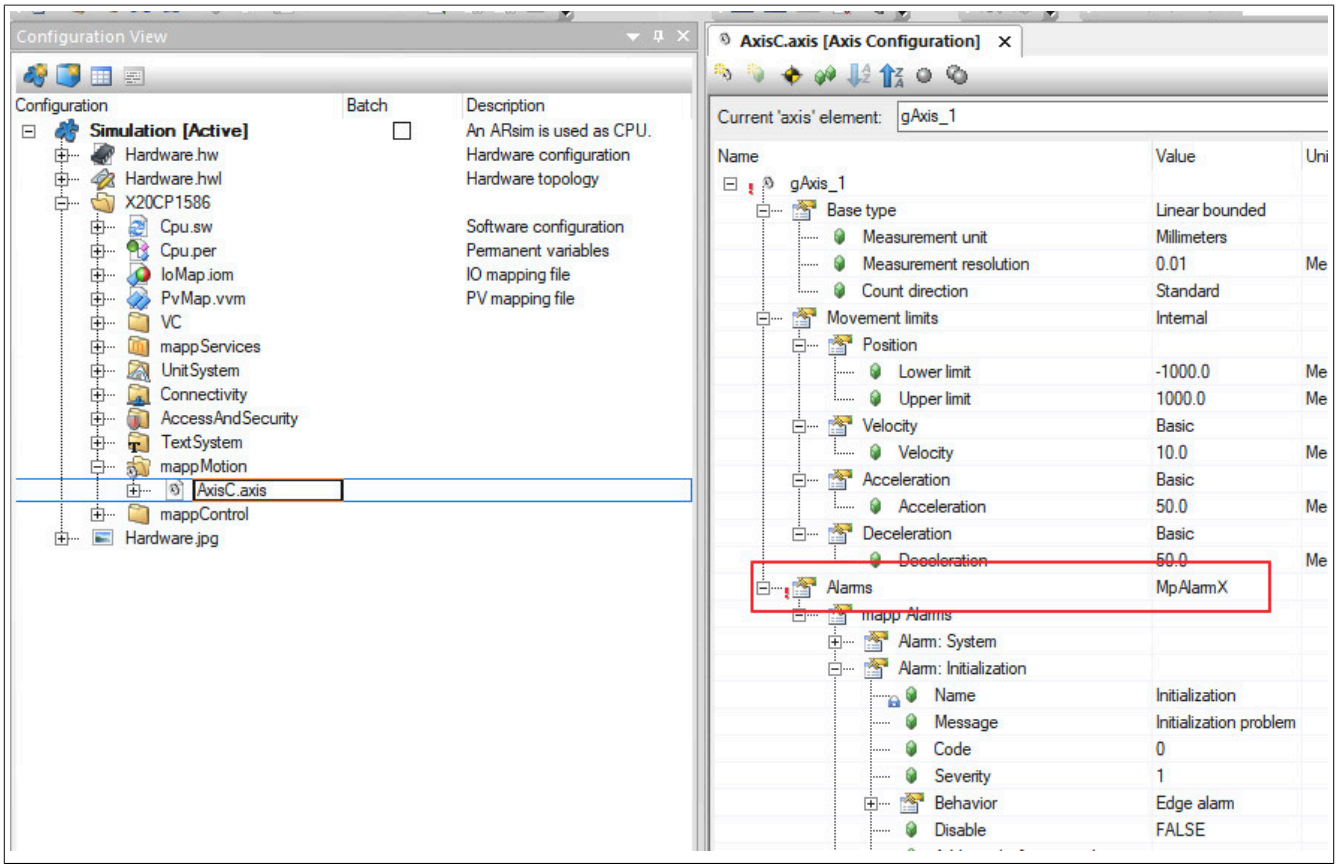


Um nun einen der Alarme konfigurieren zu können, muss als mapp Link für den MpAlarmXConfigAlarm Funktionsbaustein der mapp Link der gewünschten mapp Komponente verwendet werden. Soll ein Rezept-Alarm angepasst werden, so wird der mapp Link einer MpRecipeXml Konfiguration verwendet. Unter "Name" wird der Name des gewünschten Alarms angegeben, wie beispielsweise "SaveFailed". Das ganze kann folgendermaßen im Programm verwendet werden:

```
PROGRAM _INIT

 MpAlarmXConfigAlarm_0.Enable := TRUE;
 MpAlarmXConfigAlarm_0.MpLink := ADR(gRecipeXml);
 MpAlarmXConfigAlarm_0.Name := ADR('SaveFailed');
 MpAlarmXConfigAlarm_0.Configuration := ADR(AlarmConfig);

END_PROGRAM

PROGRAM _CYCLIC

 MpAlarmXConfigAlarm_0();

END_PROGRAM

PROGRAM _EXIT

 MpAlarmXConfigAlarm_0.Enable := FALSE;
 MpAlarmXConfigAlarm_0();

END_PROGRAM
```

Konfigurieren von mapp Motion Alarmen:

Sollen die Alarme einer mapp Motion Komponente konfiguriert werden, so gilt es zu beachten, dass der mapp Link der mapp Motion Komponente mithilfe des Funktionsbausteins MpComGetLink ermittelt werden muss. Der Ausgangsparameter "MpLink" von MpComGetLink wird mit dem Eingangsparameter "MpLink" des MpAlarmXConfigAlarm Funktionsbausteins verbunden. In diesem Beispiel soll der Alarm "Initialization" der MpAxis Komponente editiert werden. Wichtig hierbei ist, dass in der Achskonfiguration, welche für MpAxis benötigt wird, als Alarmsystem "MpAlarmX" verwendet wird.



Die Parametrierung von MpComGetLink und MpAlarmXConfigAlarm kann dabei folgendermaßen aussehen:

```
        MpComGetLink_0.Enable := TRUE;
MpComGetLink_0.ComponentName := ADR('gAxis_1');
AxisMpLink := MpComGetLink_0.MpLink;

MpAlarmXConfigAlarm_0.Enable := TRUE;
MpAlarmXConfigAlarm_0.MpLink := ADR(AxisMpLink);
MpAlarmXConfigAlarm_0.Name := ADR('Initialization');
MpAlarmXConfigAlarm_0.Configuration := ADR(AlarmConfig);
```

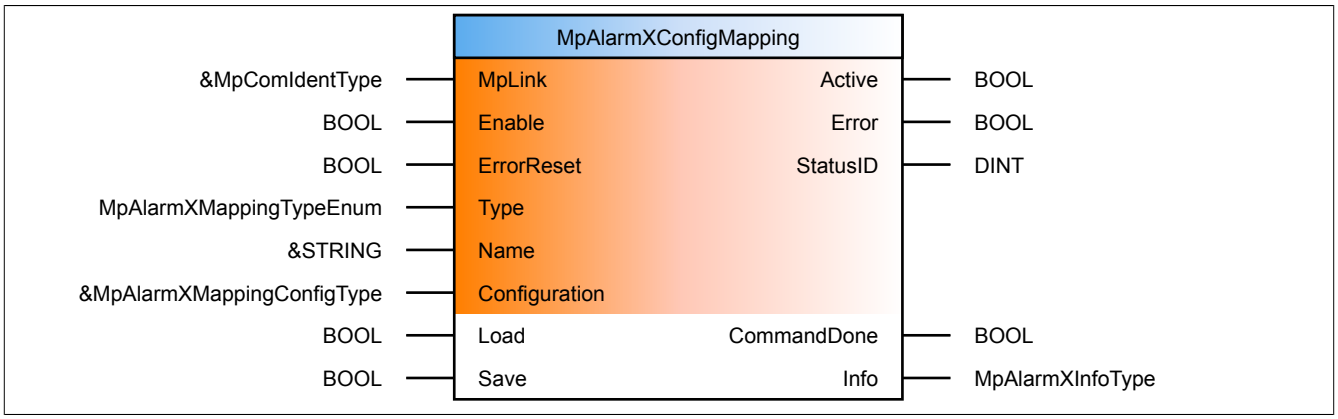
Der MpLink wird in diesem Fall in der Variable "AxisMpLink", welche vom Datentypen MpComIdentType ist, zwischengespeichert und im Anschluss am Eingangsparameter "MpLink" von MpAlarmXConfigAlarm angegeben.

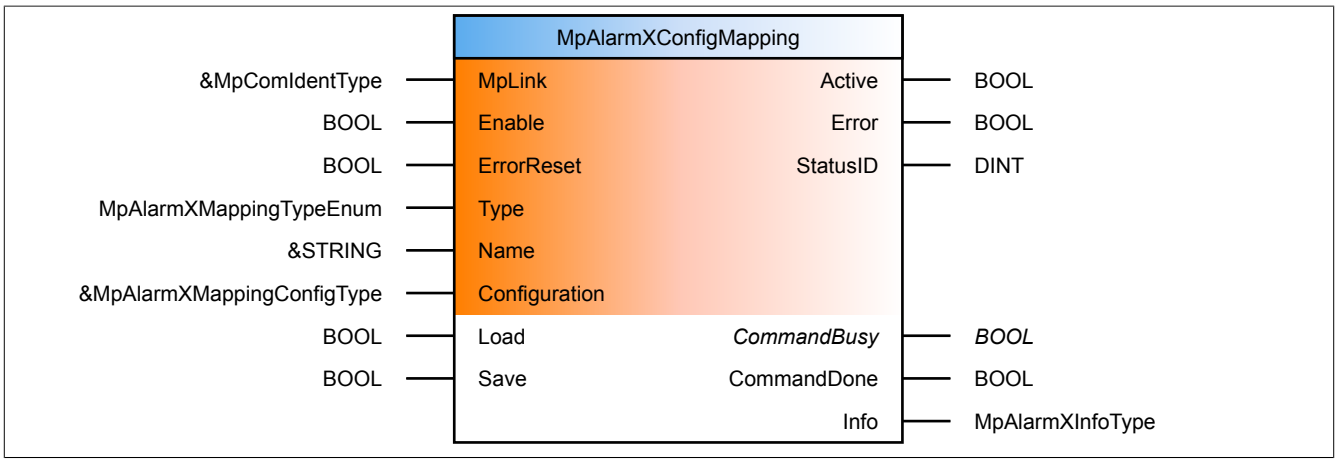
1.6.1.10 MpAlarmXConfigMapping

Dieser Funktionsbaustein ermöglicht das Erstellen und Konfigurieren des Alarm Mappings zur Laufzeit.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp.
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Type	`MpAlarmXMappingTypeEnum`	Definiert den Auslöser.
IN	Name	`Zeiger auf STRING[255]`	Definiert den Namen des Alarms, bzw. den Wertebereich der Severity.
IN	Configuration	`Zeiger auf MpAlarmXMappingConfigType`	Enthält die Aktionen, welche als Reaktion auf den Auslöser durchgeführt werden sollen.
IN	Load	`BOOL`	Lädt die Konfiguration der Komponente.
IN	Save	`BOOL`	Speichert die Konfiguration der Komponente.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpAlarmXInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.10.1 Beschreibung

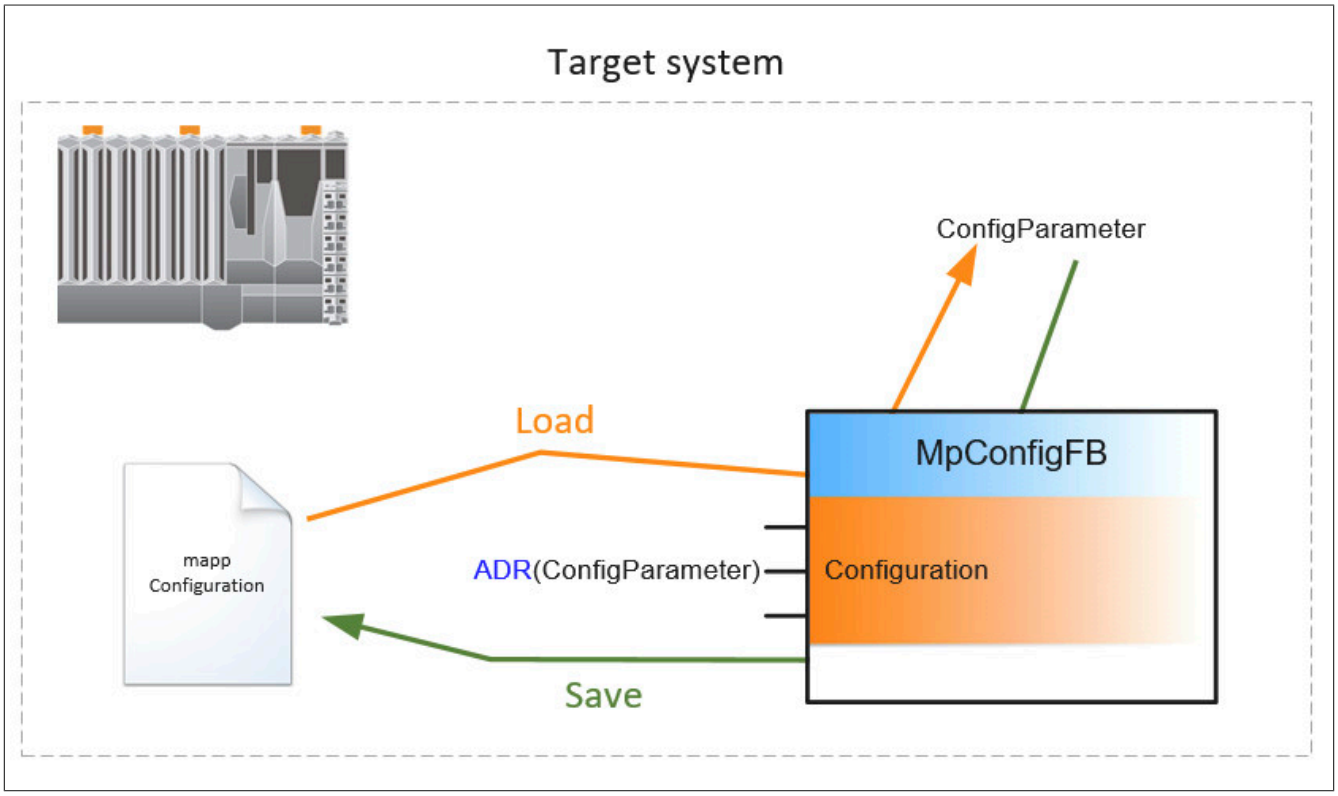
Für diesen Funktionsbaustein wird derselbe MpLink wie für MpAlarmXCore verwendet.

Mithilfe des Funktionsbausteins sind Änderungen der Mapping-Konfiguration zur Laufzeit möglich.

Der Parameter "Configuration" ermöglicht es die Konfiguration, bzw. einen Teil der Konfiguration zu bearbeiten.

Der Befehl "Load" wird verwendet, um die aktuell konfigurierten Parameter auszulesen. Dabei wird die aktuell aktive Konfiguration der Steuerung in die PV am Eingang "Configuration" geschrieben. Die Daten in der Struktur werden dabei überschrieben. Danach können die Werte beliebig editiert werden.

Über den Befehl "Save" werden die Werte aus der "Configuration" Struktur auf die aktive Konfiguration, im speziellen auf das angegebene Mapping, übernommen und, wenn möglich, direkt verwendet. Welches Mapping editiert oder erstellt werden soll wird über die Kombination aus den Eingängen "Type" und "Name" festgelegt.



Um die Konfigurations-Parameter in der Appliaktion korrekt anzupassen, wird folgendermaßen vorgegangen:

1. Befehl "Load = `TRUE`"

Die aktuelle Konfiguration wird vom Zielsystem in die Konfigurations-Struktur geladen. Der Vorgang ist mit "CommandDone = TRUE" abgeschlossen.

2. Befehl "Load = `FALSE`"

Rücksetzen des "Load" Befehls. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

3. Parametrieren der Konfigurationsstruktur

Die Parameter der Konfigurationsstruktur können nun wie gewünscht angepasst werden.

4. Befehl "Save = `TRUE`" setzen

Die geänderte Konfigurationsstruktur wird auf das Zielsystem übertragen. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

5. Befehl "Save = `FALSE`" setzen

Rücksetzen des "Save" Befehls. Der Vorgang ist mit "CommandDone = `FALSE`" abgeschlossen. Um zu kontrollieren ob die Parameter erfolgreich geändert worden sind, können die Schritte 1 und 2 erneut ausgeführt werden.

Über den Befehl "Save" werden die Werte aus der "Configuration" Struktur auf die aktive Konfiguration, im speziellen auf das angegebene Mapping, übernommen und, wenn möglich, direkt verwendet. Welches Mapping editiert oder erstellt werden soll wird über die Kombination aus den Eingängen "Type und "Name" festgelegt.

Konfigurieren der MpAlarmXMappingConfigType Struktur

Über diese Struktur kann ein bereits existierendes Alarm Mapping verändert werden, oder ein komplett neues erstellt werden. Möchte man ein bestehendes Mapping zu einem bestimmten Auslöser verändern, so wird zuerst "Type" und "Name" angegeben und dann das aktuellen Mapping geladen. Über die MpAlarmXMappingConfigType kann dann eine bestimmte Aktion verändert oder hinzugefügt werden. Gibt es die "Type" / "Name" Kombination noch nicht im Alarm Mapping so wird diese neu erstellt.

Wird als Auslöser "Type" ein Alarm definiert, so wird der Alarmname über den Parameter "Name" definiert.

Wird als Auslöser "Type" eine Severity festgelegt, so wird über "Name" der Wertebereich angegeben. Das sieht folgendermaßen aus:

- **Für einen Wertebereich:** 1..10
- **Für einen konkreten Wert:** 11

1.6.1.11 MpAlarmXCore

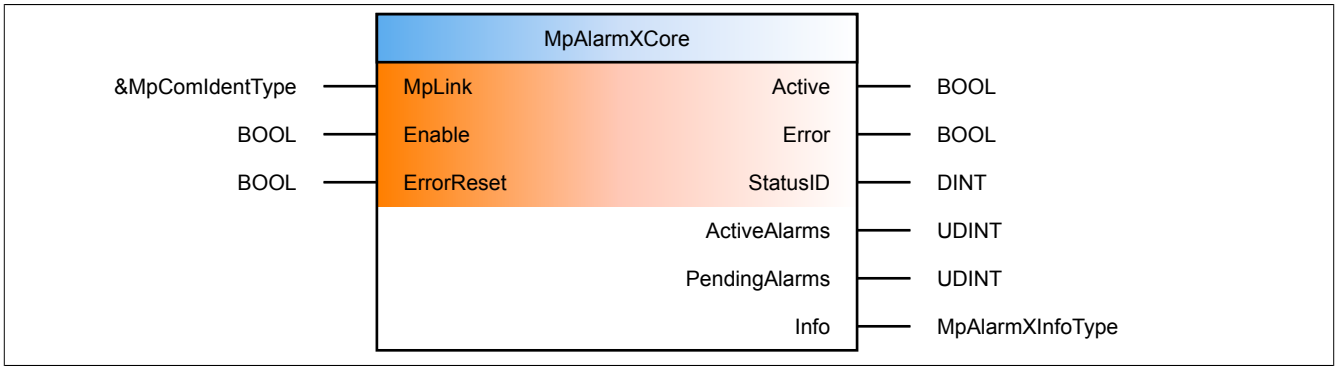
Dieser Funktionsbaustein sammelt und verwaltet Alarme.

> **Bei Verwendung des AlarmList-Widget, muss MpAlarmXCore im Init-Teil des Programms aufgerufen werden!**

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp AlarmX (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal sind viele Tutorials zum Thema zu finden.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	ActiveAlarms	`UDINT`	Anzahl der momentan aktiven Alarme.
OUT	PendingAlarms	`UDINT`	Zeigt die Anzahl jener Alarme an, die entweder aktiv sind, oder noch nicht quittiert wurden. Spiegelt also genau jene Alarme wieder, die noch auf der Visualisierung zu sehen sind.
OUT	Info	`MpAlarmXInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.11.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpAlarmXCore Konfiguration verwendet. Mithilfe dieser Komponente wird das komplette Alarmsystem verwaltet. Dieser Funktionsbaustein sammelt alle Alarme, sowie deren aktueller Status. Die Alarme selbst wurden zuvor über über MpAlarmXCore Konfiguration erstellt und konfiguriert. Der Status wird über verschiedene Funktionen von MpAlarmX verändert (MpAlarmXSet zum Setzen, MpAlarmX-Acknowledge zum Quittieren, MpAlarmXListUI zur visuellen Darstellung, ...). Weiters wird der Zustand aller *Reactions* über diesen Funktionsbaustein verwaltet.Wann genau Alarme aktiviert bzw. im System deaktiviert werden, siehe Best Practise.

> **Um sicherzustellen, dass MpAlarmXCore Alarme sofort beim Hochstart erkennt, wird dazu geraten diesen Funktionsbaustein bereits im INIT Programm über den "Enable" Eingang zu aktivieren. Damit wird die Initialisierung aller Alarme bereits im INIT Programm durchgeführt.**

Alarme zählen

Sobald der Funktionsbaustein aktiv ist, wird am Ausgang "ActiveAlarms" die Anzahl der aktuell aktiven Alarme angezeigt. Angezeigt werden alle mapp Komponenten Alarme und alle benutzerdefinierten Alarme. Inaktive, nicht quittierte Alarme werden nicht miteinberechnet.

Am Ausgang "PendingAlarms" wird die Anzahl aller aktiven und inaktiven, aber nicht quittierten Alarme angezeigt. Somit genau die Alarme, die auch auf der Visualisierung angezeigt werden, falls eine Quittierung erforderlich ist.

Alarmverwaltung in Maschinengruppen

Es kann pro Maschinengruppe nur einen MpAlarmXCore geben (wird nicht mit Gruppen gearbeitet, folg daraus, dass es nur einen MpAlarmXCore im Projekt gibt). Details zu Maschinengruppen sind im Abschnitt Bilden von modularen Applikationen zu finden. Einen Anwendungsfall, wie das für MpAlarmX aussehen kann gibt es bereits: Alarmverwaltung für Druckermaschinenlinie

1.6.1.12 MpAlarmXHistory

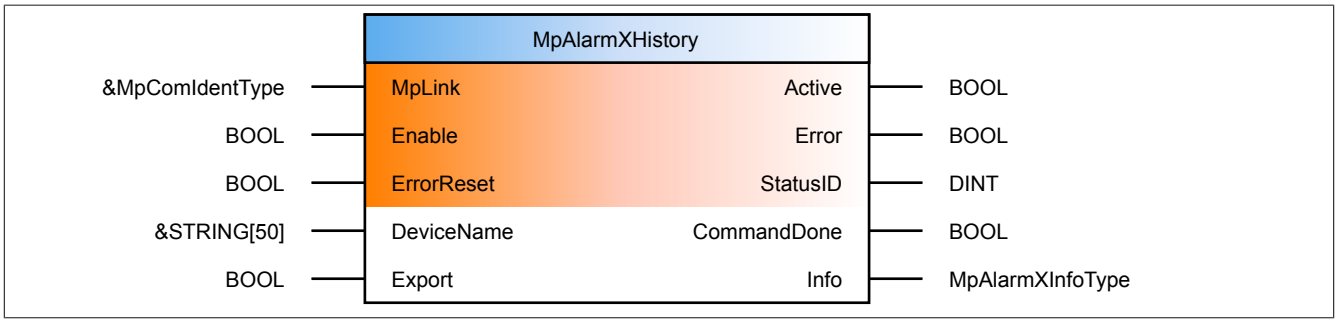
Mit diesem Funktionsbaustein kann die Alarm Historie, auf ein angegebenes Speichermedium, exportiert werden.

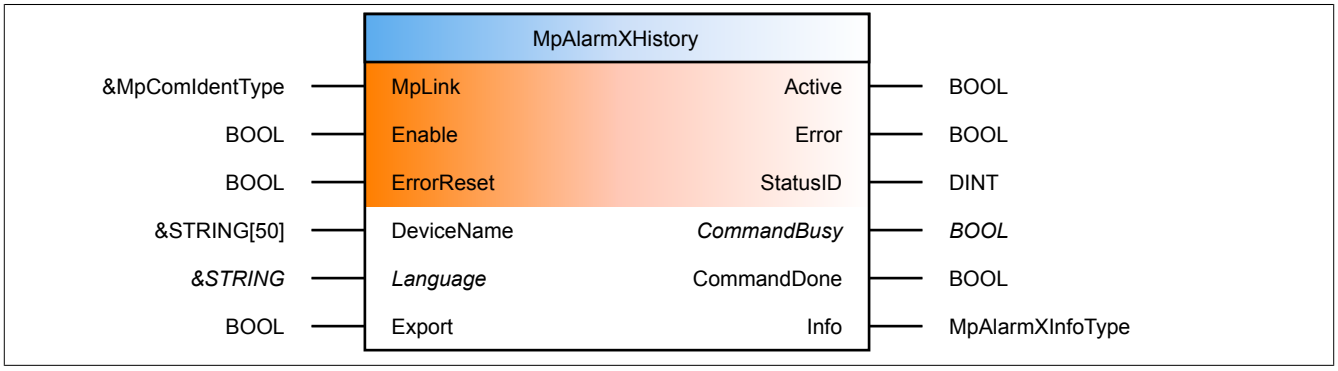
Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp AlarmX (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal sind viele Tutorials zum Thema zu finden.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXHistory Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	DeviceName	`Zeiger auf STRING[50]`	File Device (Speichermedium), auf dem die Dateien abgelegt werden.
IN	Language	`Zeiger auf STRING`[20]	Definiert in welcher Sprache bzw. Einheit die Alarmhistorie exportiert werden soll.
IN	Export	`BOOL`	Speichert die aktuelle Alarm History in einer Datei auf dem angegebenen Speichermedium ("DeviceName").
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpAlarmXInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.12.1 Beschreibung

Dieser Funktionsbaustein verwendet den MpLink einer MpAlarmXHistory Konfiguration.

MpAlarmXHistory wird verwendet, um den Alarmverlauf zu loggen und zu exportieren

Loggen

Jede Statusänderung eines Alarms kann in der Alarm Historie geloggt werden. Welche Statusänderung geloggt werden soll, kann über die Konfiguration pro Alarm festgelegt werden. Folgende Änderungen können geloggt werden:

- Inactive --> Active: Zeitpunkt, an welchem der Alarm aktiviert wurde
- Active --> Inactive: Zeitpunkt, an welchem der Alarm zurückgesetzt wurde
- Unacknowledged --> Acknowledged: Zeitpunkt, an welchem der Alarm quittiert wurde

Wie man das Report Verhalten beeinflusst wird im Abschnitt Alarmeigenschaften genauer beschrieben (siehe *History Report* am Ende der Seite).

Export

Mit dem Befehl "Export" wird die Alarm Historie als CSV Datei auf das angegebene Speichermedium ("DeviceName") exportiert. Die Exportsprache bzw. die Einheit wird über den Eingang "Language" definiert. Bei "CommandDone = `TRUE`" wurde das Kommando fertig ausgeführt. Um ein neues Kommando zu starten, muss der vorherige Befehl auf `FALSE` gesetzt werden. Der Funktionsbaustein ist nun bereit einen neuen Befehl auszuführen.

Um Sprache und Einheitensystem festzulegen ist der Eingang "Language" auf folgende Weise zu verwenden: 'Sprache|Einheitensystem'

Beispiel wäre dafür 'de|metric' oder 'en|imperial-us'. Für mehr Information siehe Einheitenverwaltung.

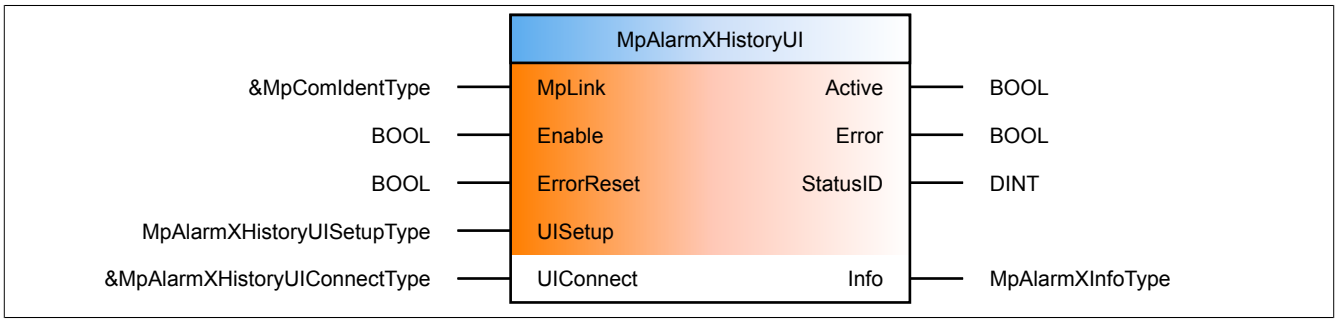
1.6.1.13 MpAlarmXHistoryUI

Dieser Funktionsbaustein wird benützt, um eine Verbindung zwischen MpAlarmX und einer Visualisierung zu schaffen. MpAlarmXHistoryUI erstellt eine Liste mit allen Statusänderungen Alarmen.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp AlarmX (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal sind viele Tutorials zum Thema zu finden.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXHistory Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	UISetup	`MpAlarmXHistoryUISetupType`	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
IN	UIConnect	`Zeiger auf MpAlarmXHistoryUIConnectType`	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.

I/O	Parameter	Datentyp	Beschreibung
OUT	Active	BOOL	FB ist aktiv.
OUT	Error	BOOL	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	DINT	Statusinformation.
OUT	Info	MpAlarmXInfoType	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.13.1 Beschreibung

Der MpLink für diesen Funktonsbaustein ist derselbe, der auch am MpAlarmXHistory Funktonsbaustein verwendet wird. MpAlarmXHistoryUI stellt eine Verbindung zwischen der Alarmhistorie und einer Visualisierung dar. Dazu müssen die Komponenten MpAlarmXHistory und MpAlarmXCore aktiv sein. Der Austausch der Daten zwischen MpAlarmXHistoryUI und der Visualisierung erfolgt über die Struktur MpAlarmXHistoryUIConnectType.

Pro MpAlarmXHistory kann nur ein MpAlarmXHistoryUI verwendet werden. Es ist nicht möglich mehrere MpAlarmXHistoryUI-Instanzen zu verwenden.

MpAlarmXHistoryUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

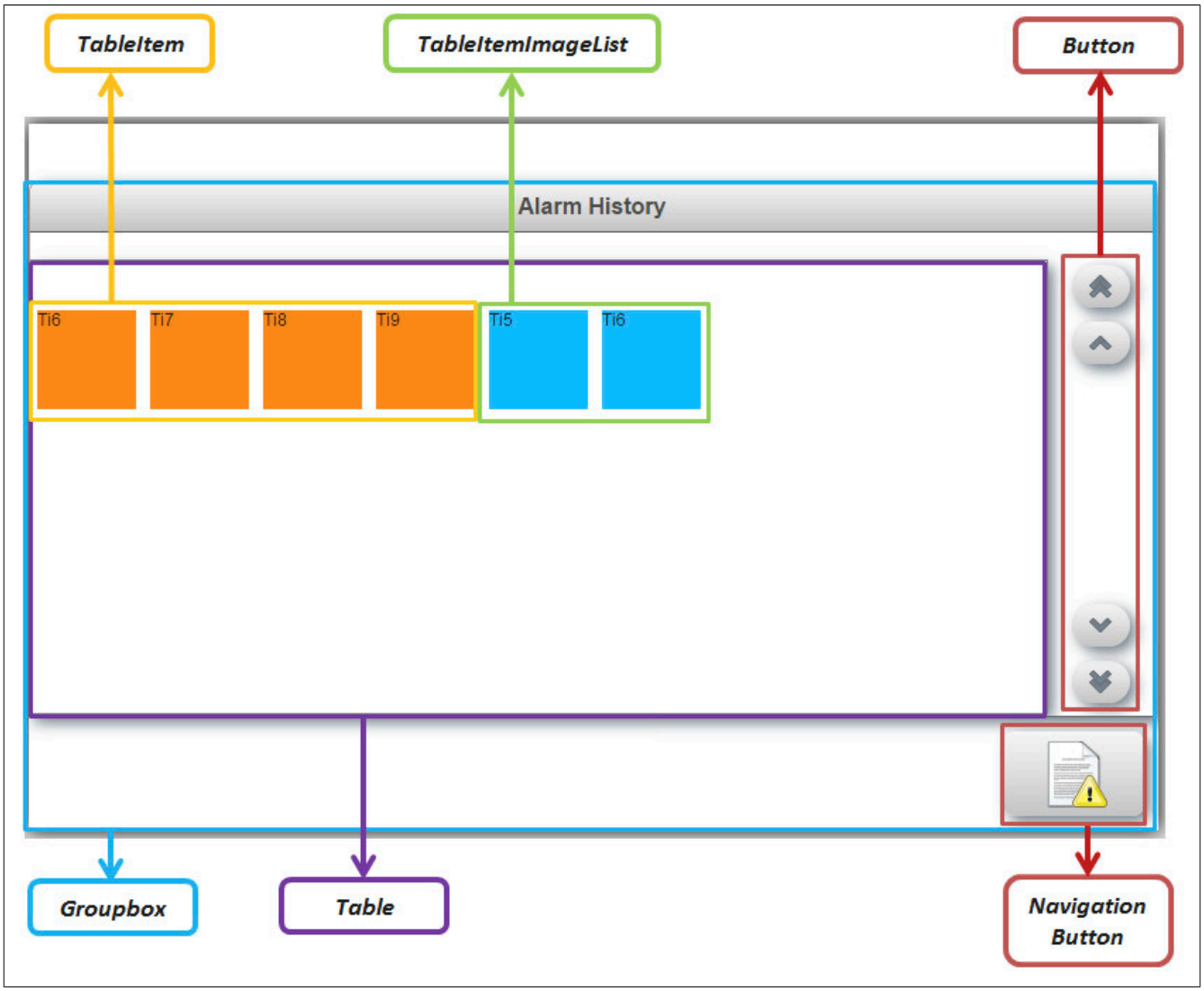
- **Status:** Der Parameter "Status" gibt Auskunft über die aktuelle Aktivität der Alarmhistorie
- **AlarmList:** Durch die Struktur MpAlarmXHistoryUIAlarmListType wird die Alarmhistorie angezeigt
- **Details:** Durch auswählen eines Alarmes wird mithilfe der Struktur MpAlarmXHistoryUIDetailsType zusätzliche Information bereitgestellt
- **Language:** Definiert, in welcher Sprache bzw. Einheit die Alarm Historie angezeigt werden. Um Sprache und Einheitensystem festzulegen ist der Eingang "Language" auf folgende Weise zu verwenden: 'Sprache|Einheitensystem'
  Beispiel wäre dafür 'de|metric' oder 'en|imperial-us'. Für mehr Information siehe Einheitenverwaltung.

MpAlarmXHistoryUISetupType

In dieser Struktur wird über "AlarmListSize" definiert, wieviele Alarme auf einer Seite der Visualisierung angezeigt werden. Über "AlarmListScrollWindow" wird entscheiden wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden.

mappView Anbindung

Wird als Visualisierungstyp mappView verwendet kann die Alarmseite mithilfe folgender Komponenten aufgebaut werden:



Die Anbindung der Parameter wird in den jeweiligen Datentypen beschrieben, wie zum Beispiel bei MpAlarmX-HistoryUIConnectType.

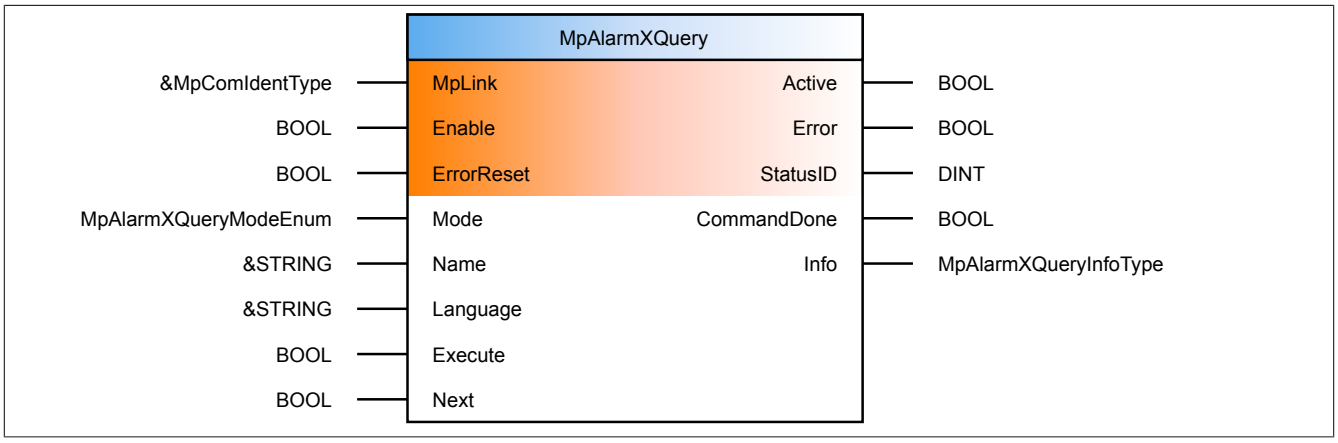
1.6.1.14 MpAlarmXQuery

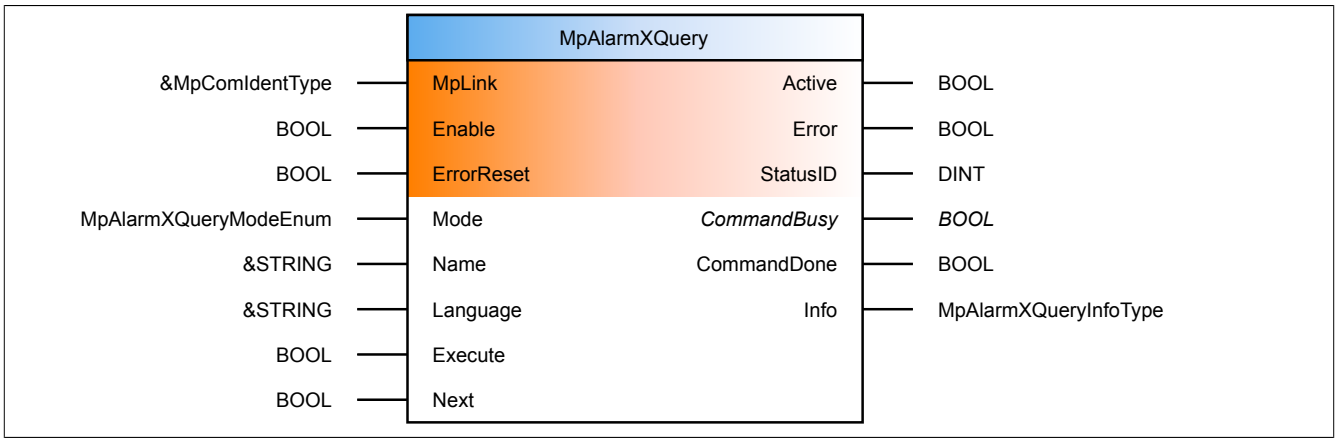
Mithilfe von MpAlarmXQuery kann eine Abfrage gestartet werden. Dadurch kann nach bestimmten Alarmen in der Alarmliste oder in der Alarmhistorie gesucht werden.

Wie der Funktionsbaustein verwendet werden kann, ist im Anwendungsfall Alarmliste nach Schweregrad filtern erklärt.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore bzw. MpAlarmXHistory Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Mode	`MpAlarmXQueryModeEnum`	Definiert, ob alle Einträge angezeigt werden sollen.
IN	Name	`Zeiger auf STRING[50]`	Name der Abfrage. Wird in der MpAlarmXCore Konfiguration definiert.
IN	Language	`Zeiger auf STRING[20]`	Definiert die Sprache und Einheit in welcher die gefilterten Alarme angezeigt werden sollen.
IN	Execute	`BOOL`	Befehl um die Anfrage auszuführen.
IN	Next	`BOOL`	Befehl um weitere Einträge der Anfrage anzuzeigen.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpAlarmXQueryInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.14.1 Beschreibung

Abhängig davon, ob in der Alarmliste oder in der Alarmhistorie Abfragen durchgeführt werden sollen, wird der MpLink einer MpAlarmXCore Konfiguration oder einer MpAlarmXHistory Konfiguration verwendet.

Mithilfe von MpAlarmXQuery kann eine in der MpAlarmXCore bzw. MpAlarmXHistory Konfiguration definierte Abfrage gestartet werden. Die gewünschte Abfrage wird am Eingangsparameter "Name" angegeben.

"Mode" definiert, ob alle vorhandenen Einträge durchsucht werden sollen, oder nur die neuesten. Die neuesten sind in dem Fall jene Alarme, die seit dem letzten "Update count" (siehe MpAlarmXCore bzw. MpAlarmXHistory Konfiguration Abschnitt "Abfrage") erzeugt worden sind. Über den Eingangsparameter "Language" kann definiert werden in welcher Sprache bzw. Einheit die gefilterten Alarme angezeigt werden. Um Sprache und Einheitensystem festzulegen ist der Eingang "Language" auf folgende Weise zu verwenden: 'Sprache|Einheitensystem'

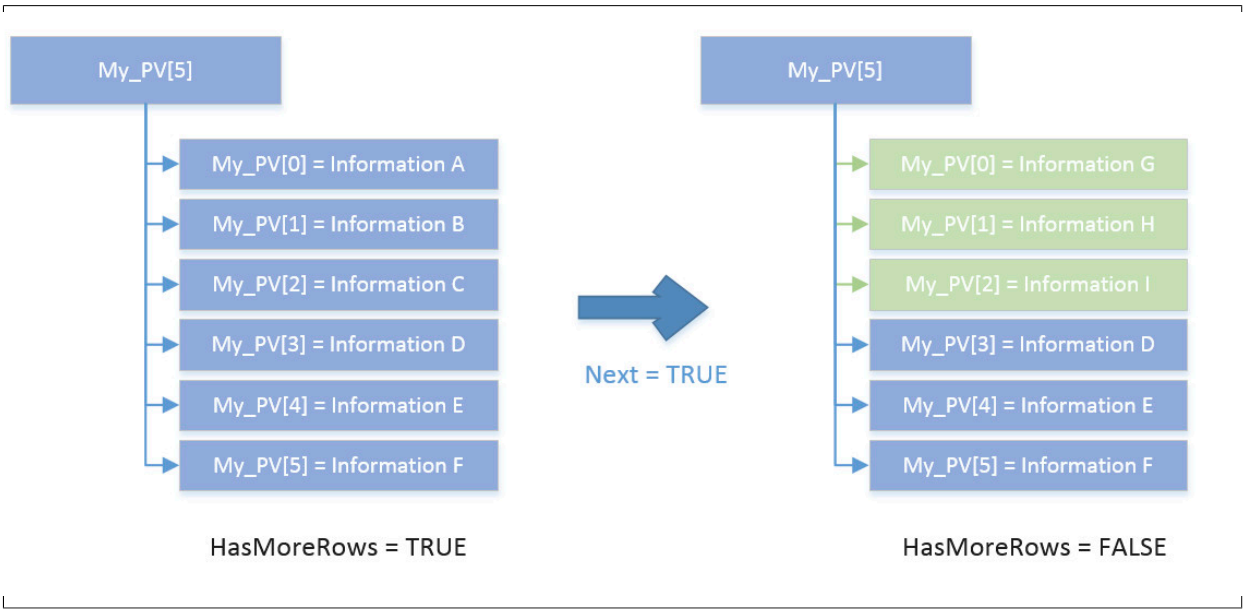
Beispiel wäre dafür 'de|metric' oder 'en|imperial-us'. Für mehr Information siehe Einheiten von Variablen anzeigen.

Durch "Execute = `TRUE`" wird die Anfrage gestartet. Die angefragte Information ist an der Prozess Variable, welche in der MpAlarmXCore bzw. MpAlarmXHistory Konfiguration im Abschnitt "Queries" definiert wurde, ersichtlich. Bei "CommandDone = `TRUE`" wurde das Kommando fertig ausgeführt. Um ein neues Kommando zu starten, muss der vorherige Befehl auf `FALSE` gesetzt werden. Der Funktionsbaustein ist nun bereit einen neuen Befehl auszuführen.

In der Info-Struktur können folgende Informationen entnommen werden:

- **AvailableSpace:** Gibt an wie viel die Platz die Prozess Variable, welche in der MpAlarmXCore bzw. MpAlarmXHistory Konfiguration im Abschnitt "Queries" definiert wurde, bietet. Ist die Prozess Variable ein STRING Array von 100 Elementen, so wird 100 angezeigt.
- **RowsRead:** Zeigt an wie viele Einträge durch die Abfrage gelesen worden sind.
- **HasMoreRows:** Bei "HasMoreRows = `FALSE`" sind alle Einträge der Abfrage in der Prozess Variable sichtbar. Bei "HasMoreRows = `TRUE`" konnten nicht alle Einträge in der Prozess Variable angezeigt werden.

Wurde als Prozessvariable ein STRING Array von 6 Elementen verwendet, es aber 9 Einträge sind, so wird "HasMoreRows = `TRUE`" angezeigt. Um die letzten 3 Einträge zu bekommen, muss der Befehl "Next = `TRUE`" von MpAlarmXQuery verwendet werden. Die letzten 3 Einträge werden auf die Variable geschrieben. Dabei wird die Information auf die ersten Elemente der Prozessvariable geschrieben:



Sind noch weitere Daten vorhanden ("HasMoreRows = `TRUE`") und die Daten haben sich in der Zwischenzeit geändert (es wurden beispielsweise weitere Alarme ausgelöst), so sind die Daten nicht mehr konsistent. Bei der Verwendung des Befehls "Next = `TRUE`" von MpAlarmXQuery, wird die Warnung "Daten inkonsistent" angezeigt. Die Abfrage muss nocheinmal neu durchgeführt werden um sicherzustellen, dass die Daten aktuell sind.

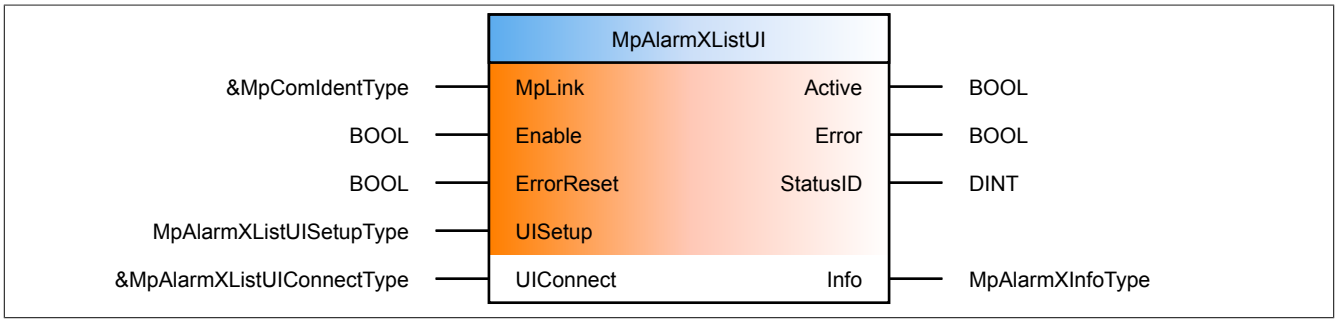
1.6.1.15 MpAlarmXListUI

Dieser Funktionsbaustein wird benützt, um eine Verbindung zwischen MpAlarmX und einer Visualisierung zu schaffen. MpAlarmXListUI erstellt eine Liste mit aktiven bzw. nicht quittierten Alarmen.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp AlarmX (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal sind viele Tutorials zum Thema zu finden.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.

I/O	Parameter	Datentyp	Beschreibung
IN	UISetup	MpAlarmXListUISetupType	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
IN	UIConnect	Zeiger auf MpAlarmXListUIConnectType	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
OUT	Active	BOOL	FB ist aktiv.
OUT	Error	BOOL	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	DINT	Statusinformation.
OUT	Info	MpAlarmXInfoType	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.15.1 Beschreibung

Der MpLink für diesen Funktionsbaustein ist derselbe, der auch am MpAlarmXCore Funktionsbaustein verwendet wird. Die Komponente stellt eine Verbindung zwischen der Alarmverwaltung und einer Visualisierung dar. Der Austausch der Daten zwischen MpAlarmXListUI und der Visualisierung erfolgt über die Struktur MpAlarmXListUIConnectType. Um MpAlarmXListUI zu verwenden muss eine MpAlarmXCore Komponente aktiv sein.

Pro MpAlarmXCore kann nur ein MpAlarmXListUI verwendet werden. Es ist nicht möglich mehrere MpAlarmXListUI-Instanzen zu verwenden.

MpAlarmXListUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

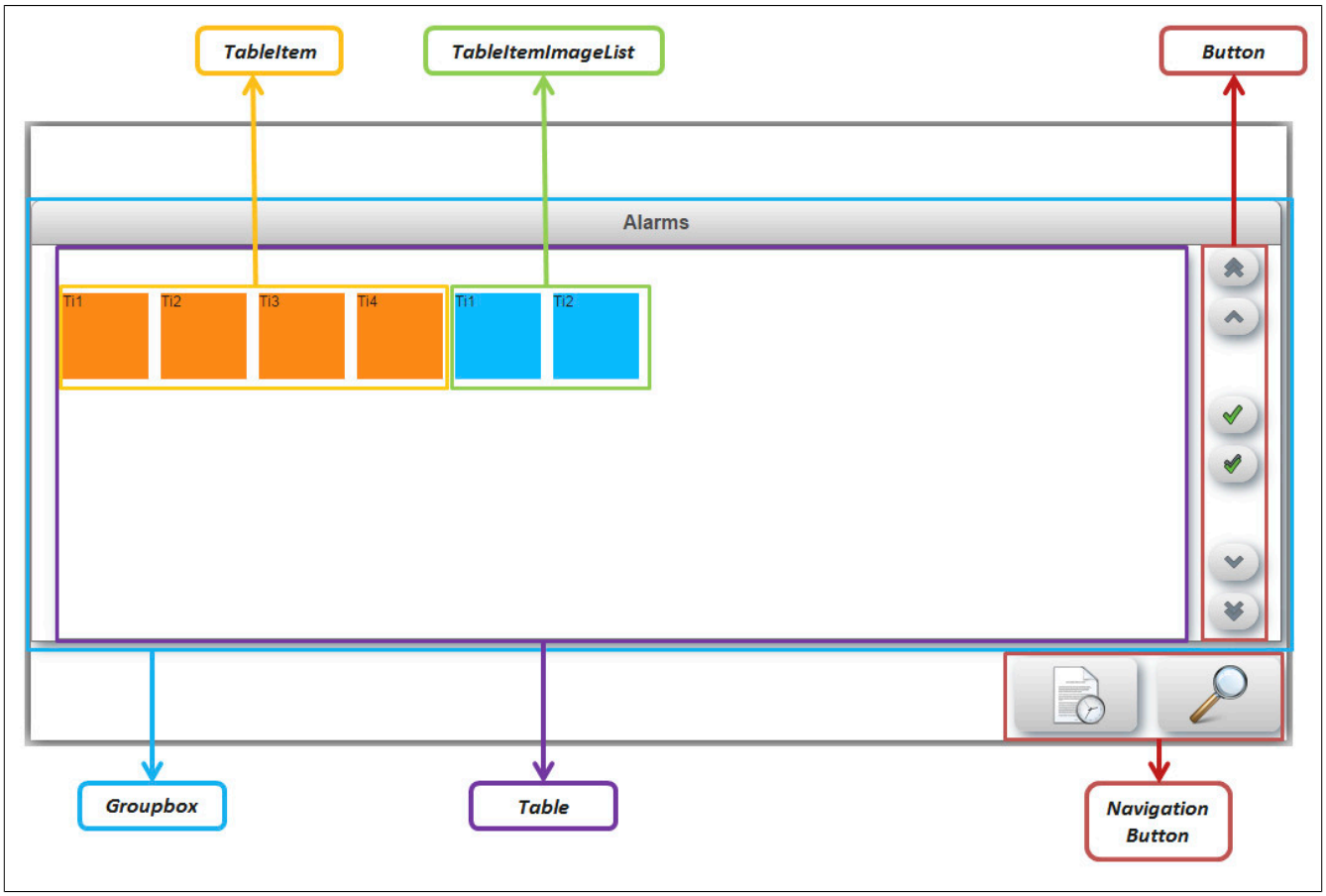
- **Status:** Der "Status" Parameter gibt Auskunft über die aktuelle Aktivität der Alarmliste
- **AlarmList:** Über die Struktur MpAlarmXListUIAlarmListType werden alle aktuellen Alarme angezeigt
- **AcknowledgeAll:** Durch "AcknowledgeAll" können alle angezeigten Alarme quittiert werden
- **Acknowledge:** Durch "Acknowledge" kann ein ausgewählter Alarm quittiert werden
- **Details:** Durch Auswählen eines Alarms wird mithilfe der Struktur MpAlarmXListUIDetailsType zusätzliche Information bereitgestellt. Als Teil der Details wird der Alarm Backtrace zur Verfügung gestellt, welcher, wenn vorhanden, die Fehlerursache im Logger sucht und darstellt
- **Language:** Definiert, in welcher Sprache bzw. Einheit die Alarme angezeigt werden. Um Sprache und Einheitensystem festzulegen ist der Eingang "Language" auf folgende Weise zu verwenden: 'Sprache|Einheitensystem'
  Beispiel wäre dafür 'de|metric' oder 'en|imperial-us'. Für mehr Information siehe Einheitenverwaltung.

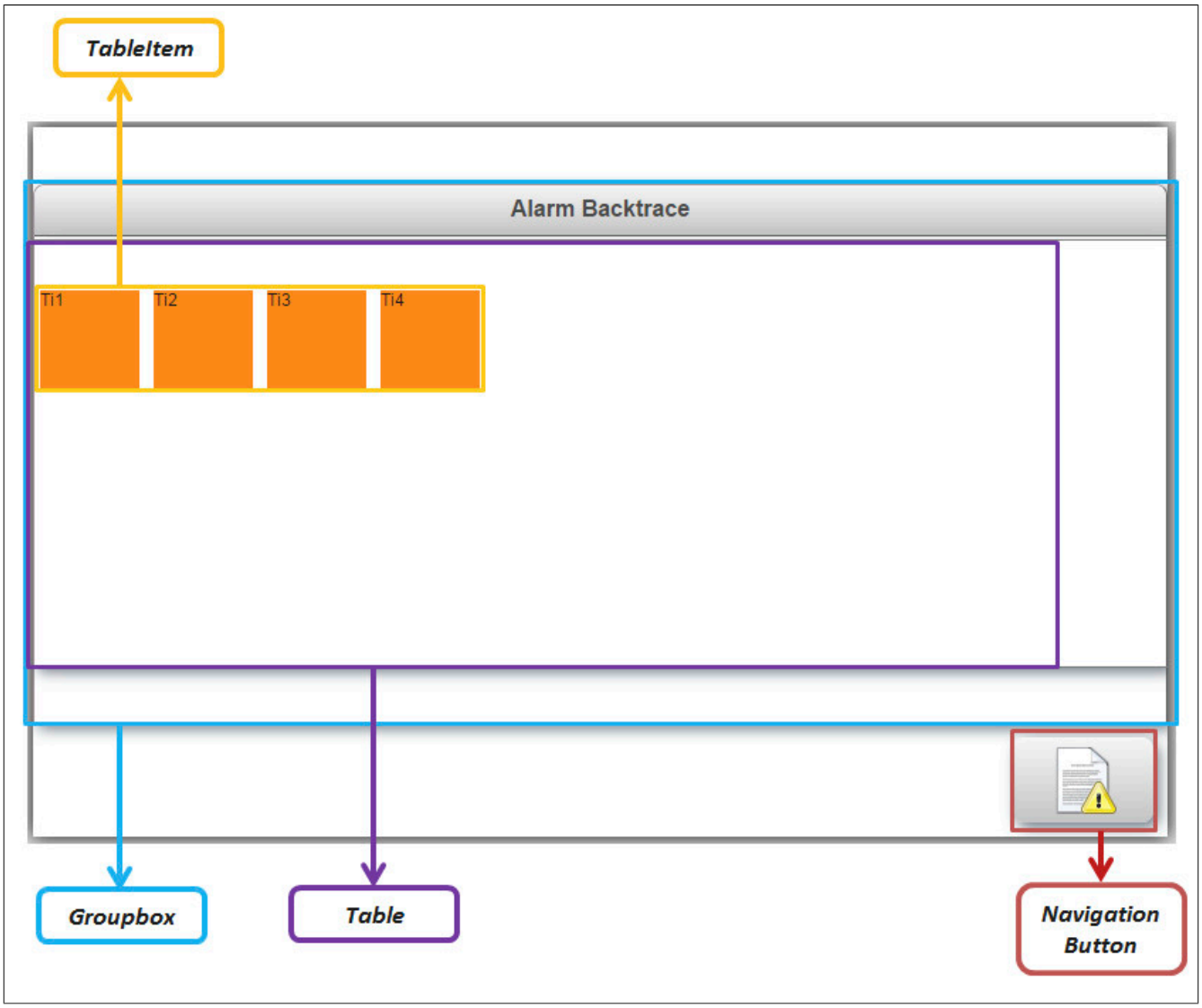
MpAlarmXListUISetupType

In dieser Struktur wird über "AlarmListSize" definiert, wieviele Alarme auf einer Seite der Visualisierung angezeigt werden. Über "AlarmListScrollWindow" wird entscheiden wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden.

mappView Anbindung

Wird als Visualisierungstyp mappView verwendet kann die Alarmseite mithilfe folgender Komponenten aufgebaut werden:


TableItem
TableItemImageList
Button
Alarms
Ti1
Ti2
Ti3
Ti4
Ti1
Ti2
Groupbox
Table
Navigation Button

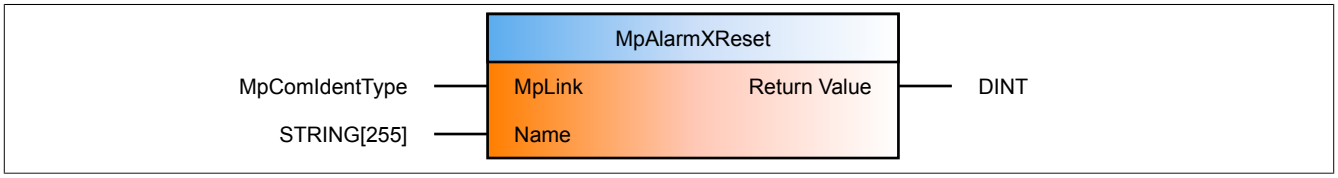


Die Anbindung der Parameter wird in den jeweiligen Datentypen beschrieben, wie zum Beispiel bei MpAlarmXListUIConnectType.

1.6.1.16 MpAlarmXReset

Diese Funktion setzt benutzerdefinierte Alarme wieder zurück.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration).
IN	Name	`STRING[255]`	Name der zur Identifizierung des Alarms dient.
OUT	ReturnValue	`DINT`	Status Information der Funktion.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.16.1 Beschreibung

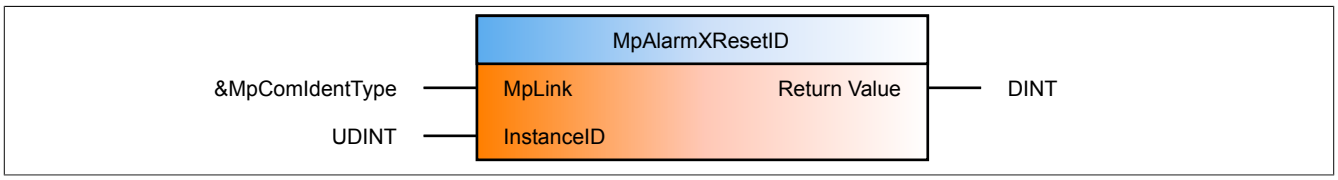
Diese Komponente verwendet den selben mapp Link der zum Setzen des Alarms durch MpAlarmXSet benutzt wurde.

Durch MpAlarmXReset kann ein Alarm, der durch MpAlarmXSet aktiviert wurde, zurückgesetzt werden. Dabei muss der Name des Alarms am Eingang "Name" angegeben werden.

1.6.1.17 MpAlarmXResetID

Diese Funktion setzt benutzerdefinierte Alarme wieder zurück.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration).
IN	InstanceID	`UDINT`	Instanz ID, welche zur Identifizierung des Alarms dient.
OUT	ReturnValue	`DINT`	Status Information der Funktion.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.17.1 Beschreibung

Diese Komponente verwendet den selben mapp Link der zum Setzen des Alarms durch MpAlarmXSet benutzt wurde.

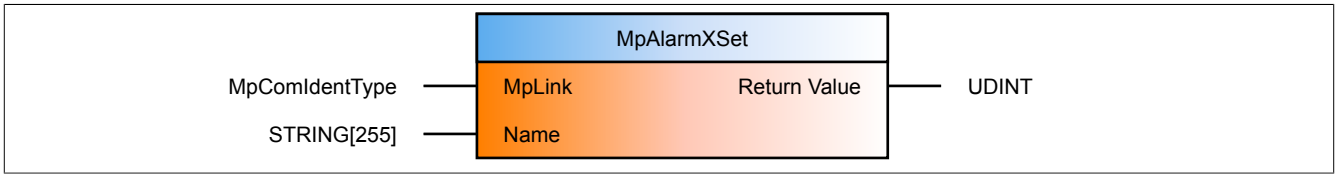
Durch MpAlarmXReset kann ein Alarm, der durch MpAlarmXSet aktiviert wurde, zurückgesetzt werden. Dabei muss die Instanz ID des Alarms am Eingang "InstanceID" angegeben werden. Die Instanz ID ist der Rückgabewert, welcher beim Setzen des Alarms über MpAlarmXSet zurückgegeben wurde.

> **Das Arbeiten mit ID Funktionen ist nur dann notwendig, wenn mit mehrfach instanzierten Alarmen gearbeitet wird.**

1.6.1.18 MpAlarmXSet

Diese Funktion aktiviert benutzerdefinierte Alarme.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmXCore Konfiguration).
IN	Name	`STRING[255]`	Name der zur Identifizierung des Alarms dient.
OUT	Return Value	`UDINT`	Instanz ID des Alarms, welcher aktiviert wurde.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

1.6.1.18.1 Beschreibung

Diese Funktion verwendet den MpLink einer MpAlarmXCore Konfiguration.

Beim Ausführen dieser Funktion wird ein benutzerdefinierter Alarm aktiviert. Benutzerdefinierte Alarme können über die MpAlarmXCore Konfiguration erstellt und konfiguriert werden.

Es wir der "Name" des Alarms übergeben, welcher aktiviert werden soll. Rückgabewert ist die Instanz ID des Alarms, über welche der Alarm bei mehrfacher Instanzierung identifiziert werden kann.

Sobald die Funktion aufgerufen wird, wird die Alarmkonfiguration gelesen. Es werden definierte Alarmeigenschaften, Alarmsnippets, Alarm Mappings usw. aufgelöst. Dies geschieht bei jedem Funktionsaufruf. Hierbei ausgenommen sind asynchrone Alarme. In diesem Fall werden Alarmsnippets, Alarm Mappings usw. von MpAlarmXCore aufgelöst.

Beim aktivieren von Alarmen sollte die Information im Abschnitt Best Practise berücksichtigt werden.

> **Wird eine Alarm aktiviert, welcher in der Konfiguration nicht existiert, so wird ein neuer Alarm erzeugt. Dieser Alarm wird als persistenter Alarm aufgelegt.**

1.6.2 Datentypen und Enumeratoren

1.6.2.1 Datentypen

1.6.2.1.1 MpAlarmXAcknowledgeAllInfoType

Mit diesem Datentyp wird für den MpAlarmXAcknowledgeAll Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
UnacknowledgedAlarms	`UDINT`	Anzahl an nicht quittierten Alarmen.
Diag	`MpAlarmXDiagType`	Eine Diagnosestruktur für die StatusID

1.6.2.1.2 MpAlarmXActionConfigType

Dieser Datentyp beschreibt eine Aktion im Alarm Mapping, welche als Folge eines Auslöser (z.B. Severity >= 10) auftreten kann.

Parameter	Datentyp	Beschreibung	Konfigurationsupdate
Type	`MpAlarmXActionEnum`	Aktion, welche durchgeführt werden soll (z.B. Reaction, Escalate, ...)	
Name	`STRING[255]`	Abhängig vom Typ, hat dieser Parameter eine unterschiedliche Bedeutung: • **Reaction:** Es wird die gewünschte Reaktion über den Namen definiert • **Replace Alarm:** Es wird der Alarmname angegeben, mit welchem der auslösende Alarm ersetzt werden soll • **Aggregate Alarm and Escalate:** Es wird der Alarmname angegeben welcher nach oben hin eskaliert und aggregiert werden soll • **Aggregate Alarm locally:** Es wird der Alarmname angegeben, welcher aggregiert werden soll Ist keine spezielle Bedeutung für eine Aktion aufgelistet, so ist der Parameter Name nicht nötig für diesen Aktionstyp	

1.6.2.1.3 MpAlarmXActivationUpdateType

Über diesen Datentypen kann das Aktualisierungs-Verhalten des Alarms definiert werden.

Parameter	Datentyp	Wert	Beschreibung
Timestamp	`BOOL`	`FALSE`	Definiert ob der Zeitstempel a soll.
Snippet	`BOOL`	`FALSE`	Definiert ob das Snippet aktivie

1.6.2.1.4 MpAlarmXAlarmConfigType

Dieser Datentyp beschreibt alle Eigenschaften eines einzelnen Alarms.

Wie das Konfigurationsupdate der einzelnen Parameter zur Laufzeit durchgeführt wird, ist im Abschnitt Konfiguration während der Laufzeit erklärt.

Parameter	Datentyp	Wert	Beschreibung
Message	STRING[255]	''	Alarmtext, entweder direkt ode das Textsystem Die Referenzierung auf das Te bei folgendermaßen aussehe pace/TextID}' Es gilt zu beachten, dass bei sprache Strukturierter Text da gendermaßen angegeben wer Dadurch ändert sich die Anga ßen: 'Test: {$$Namespace/Tex
Code	UDINT	0	Alarmnummer (optional; Ident plikation erfolgt über den Nam
Severity	UDINT	1	Schweregrad eines Alarms
Behavior	MpAlarmXBehaviorType		Verhalten des Alarms (in der die erweiterte Ansicht zugängl
Disable	BOOL	FALSE	Aktiveren, bzw. Deaktivieren d
AdditionalInformation1	STRING[255]	''	Über diese Parameter könner mationen zu einem Alarm ve beispielsweise ein PDF, ein HTML Datei. Diese Dateien werden, um die genaue Ursac erklären oder Lösungsvorschl Angabe erfolgt dabei statisch das Textsystem.
AdditionalInformation2	STRING[255]	''	Über diese Parameter könner mationen zu einem Alarm ve beispielsweise ein PDF, ein HTML Datei. Diese Dateien werden, um die genaue Ursac erklären oder Lösungsvorschl Angabe erfolgt dabei statisch das Textsystem.

1.6.2.1.5 MpAlarmXBehaviorType

Dieser Datentyp beschreibt das Verhalten eines Alarms.

Parameter	Datentyp	Wert	Beschreibung
AutoReset	BOOL	FALSE	Definiert, ob der Alarm sofort wieder zurückgesetzt werden von MpAlarmXReset ist nicht • **Edge Alarm:** TRUE • **Persistent Alarm:** FA
Acknowledge	MpAlarmXAcknowledgeConfigEnum	mpALARMX_ACK_REQ	Legt fest, ob und wann ein Ala muss • **Edge Alarm:** mpALAR • **Persistent Alar** MX_ACK_REQ
MultipleInstances	BOOL	FALSE	Definiert, ob mehr als ein solc sein. Mehr Infos im Abschnitt • **Edge Alarm:** TRUE • **Persistent Alarm:** FA
ReactionUntilAcknowledged	BOOL	TRUE	*Reactions*, welche eine Folge bleiben aktiv bis der Alarm qui • **Edge Alarm:** TRUE • **Persistent Alarm:** TR
Retain	BOOL	FALSE	Definiert ob ein Alarm nach e Steuerung noch erhalten bleib
Async	BOOL	FALSE	Definiert, ob ein Alarm asynch den soll oder nicht.
HistoryReport	MpAlarmXHistoryReportType		Definiert welche Zustands Alarms in die Alarmhistorie werden sollen
DataUpdate	MpAlarmXUpdateBehaviorType		Definiert, ob die Alarminform werden soll oder nicht.

1.6.2.1.6 MpAlarmXControlInfoType

Mit diesem Datentyp wird für die Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpAlarmXDiagType	Eine Diagnosestruktur für die StatusID

1.6.2.1.7 MpAlarmXDiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	MpAlarmXStatusIDType	StatusID Diagnose Struktur

1.6.2.1.8 MpAlarmXHistoryReportType

Definiert welche Zustandsänderungen eines Alarms in der Alarmhistorie aufgezeichnet werden.

Parameter	Datentyp	Wert	Beschreibung
InactiveToActive	BOOL	TRUE	Zustandswechsel von inaktiv a • **Edge Alarm:** TRUE • **Persistent Alarm:** TR
ActiveToInactive	BOOL	TRUE	Zustandswechsel von aktiv au • **Edge Alarm:** FALSE • **Persistent Alarm:** FA
UnacknowledgedToAcknowledged	BOOL	TRUE	Zustandswechsel von nicht qu • **Edge Alarm:** TRUE • **Persistent Alarm:** TR
AcknowledgedToUnacknowledged	BOOL	TRUE	Zustandswechsel von quittiert • **Edge Alarm:** TRUE • **Persistent Alarm:** TR
Reactivation	BOOL	TRUE	Zeichnet die Re-Aktivierung ei

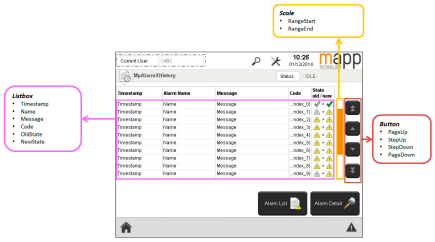
1.6.2.1.9 MpAlarmXHistoryUIAlarmListType

Liste mit allen Alarm - Statusänderungen.

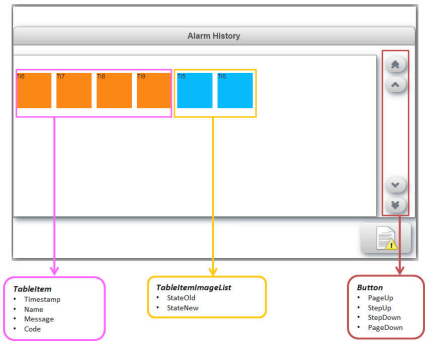
Parameter	Datentyp	Beschreibung
Severity	ARRAY[0.. 49] von UDINT	Level des Alarms
Code	ARRAY[0.. 49] von UDINT	Code des Alarms **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
Message	ARRAY[0.. 49] von WSTRING[255]	Beschreibung des Alarms **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
AdditionalInformation1	ARRAY[0.. 49] von WSTRING[255]	Link mit zusätzlicher Information
AdditionalInformation2	ARRAY[0.. 49] von WSTRING[255]	Link mit zusätzlicher Information
Name	ARRAY[0.. 49] von STRING[255]	Name des Alarms **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
Scope	ARRAY[0.. 49] von STRING[255]	Scope des Alarms
OldState	ARRAY[0.. 49] von MpAlarmXStateEnum	Zustandsänderung des Alarms **mappView Anbindung:** Datapoint von *TableItemImageList* / Data: selectedIndex
NewState	ARRAY[0.. 49] von MpAlarmXStateEnum	Zustandsänderung des Alarms **mappView Anbindung:** Datapoint von *TableItemImageList* / Data: selectedIndex
Timestamp	ARRAY[0.. 49] von STRING[50]	Zeitstempel zum Zeitpunkt der Statusänderung **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
SelectedIndex	UINT	Index des gerade ausgewählten Alarms
MaxSelection	UINT	Index des letzten Alarms
PageUp	BOOL	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten
PageDown	BOOL	Springt ans Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten

Parameter	Datentyp	Beschreibung
StepUp	BOOL	Wählt vorhergehenden Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten
StepDown	BOOL	Wählt nächsten Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten
RangeStart	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von Scale
RangeEnd	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von Scale

VC4 Anbindung:



mappView Anbindung:



EventBinding

Um mithilfe der Buttons in der Tabelle nach oben bzw. nach unten navigieren zu können, muss ein EventBinding für den jeweiligen Button eingefügt werden.

PageUp

```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmHistory" widgetRefId="PageUp" event=
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXHistoryUIConnect.AlarmList.PageU
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                    </Target>
                </Action>
```

```
            </EventHandler>
</EventBinding>
```

PageDown

```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmHistory" widgetRefId="PageDown" even
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXHistoryUIConnect.AlarmList.Page
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                    </Target>
                </Action>
            </EventHandler>
</EventBinding>
```

StepUp

```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmHistory" widgetRefId="StepUp" event
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXHistoryUIConnect.AlarmList.Step
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                    </Target>
                </Action>
            </EventHandler>
</EventBinding>
```

StepDown

```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmHistory" widgetRefId="StepDown" even
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXHistoryUIConnect.AlarmList.Step
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                    </Target>
                </Action>
            </EventHandler>
</EventBinding>
```

1.6.2.1.10 MpAlarmXHistoryUIConnectType

Mithilfe dieses Datentypes wird eine Verbindung zwischen MpAlarmXHistoryUI und der Visualisierung erstellt. Es werden alle aufgetretenen Alarme und deren Zustände aufgelistet.

Parameter	Datentyp	Beschreibung
Status	MpAlarmXHistoryUIStatusEnum	Aktuelle Operation
AlarmList	MpAlarmXHistoryUIAlarmList-Type	Liste aller Alarm - Statusänderungen
Details	MpAlarmXHistoryUIDetailsType	Details des aktuell ausgewählten Alarms
Language	STRING[20]	Definiert, in welcher Sprache bzw. Einheit die Alarme angezeigt werden

1.6.2.1.11 MpAlarmXHistoryUIDetailsType

Struktur mit zusätzlichen Informationen zum aktuell in der Liste ausgewählten Alarm.

Parameter	Datentyp	Beschreibung
Severity	UDINT	Level des Alarms
Code	UDINT	Code des Alarms
Message	WSTRING[255]	Beschreibung des Alarms
AdditionalInformation1	WSTRING[255]	Link mit zusätzlicher Information
AdditionalInformation2	WSTRING[255]	Link mit zusätzlicher Information
Name	STRING[255]	Name des Alarms
Scope	STRING[255]	Scope des Alarms
OldState	MpAlarmXStateEnum	Alter Status des Alarms
NewState	MpAlarmXStateEnum	Neuer Status des Alarms

Parameter	Datentyp	Beschreibung
Timestamp	STRING[50]	Zeitstempel der Statusänderung
DataDescriptions	ARRAY[0..9] von STRING[50]	Liste mit zusätzlichen Informationen des Alarms. Reserviert für spätere Nutzung.
DataValues	ARRAY[0..9] von STRING[255]	Liste mit zusätzlichen Daten des Alarms. Reserviert für spätere Nutzung.
TimestampDT	DATE_AND_TIME	Zeitpunkt, an welchem der Alarm aktiv wurde (Format über MpAlarmXHistoryUISetupType adaptierbar)

1.6.2.1.12 MpAlarmXHistoryUISetupType

Zusätzliche Konfigurationsmöglichkeiten, um die Visualisierung zu beschreiben.

Parameter	Datentyp	Standardwert	Beschreibung
AlarmListSize	UINT	20	Zahl an Alarmen die auf einer Seite der Liste auf der Visualisierung dargestellt werden können
AlarmListScrollWindow	USINT	1	Zeigt an, wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden
TimeStampPattern	STRING[50]	'%Y-%m-%d %H:%M:%S:%L'	Format, in welchem der Zeitstempel auf der Visualisierung dargestellt wird

Der Parameter "TimeStampPattern" gibt an, in welchem Format der Zeitstempel über die Visualisierung angezeigt wird:

- Standardformat: %Y %m %d %H:%M:%S:%L (z.B. '1990 25 03:13:27:253')
- Angepasstes Format (Beispiel): %d.%m.%Y %H:%M:%S:%L (z.B. '25.02.1991 13:27:03:561')

Folgende weitere Möglichkeiten existieren zum Anpassen des Formats:

Kennung	Beschreibung
%H	Stunde im 24 Stunden Format. Null am Anfang falls notwendig: 00 - 23
%I	Stunde im 12 Stunden Format. Null am Anfang falls notwendig: 01- 12
%k	Stunde im 24 Stunden Format: 0 - 23
%M	Minute innerhalb einer Stunde. Null am Anfang falls notwendig: 00 - 59
%S	Sekunden innerhalb einer Minute. Null am Anfang falls notwendig: 00 - 59
%L	Millisekunden innerhalb einer Sekunde. Null am Anfang falls notwendig: 000 - 999
%s	Sekunden seit 1 Januar 1970
%C	Vierstellige Jahreszahl dividiert durch 100. Über zwei Ziffern angezeigt. Null am Anfang falls notwendig: 00 - 99
%Y	Jahr, formatiert als vierstellige Zahl. Null am Anfang falls notwendig
%y	Die letzten beiden Ziffern des Jahres. Null am Anfang falls notwendig: 00 - 99
%m	Monat. Null am Anfang falls notwendig: 01 - 31
%d	Tag des Monats. Null am Anfang falls notwendig: 01 - 31
%e	Tag des Monats: 1 - 31
%R	Zeit als 24 Stunden Anzeige: '%H:%M'
%T	Zeit als 24 Stunden Anzeige: '%H:%M:%S'
%D	Datum formatiet als: '%m/%d/%y'
%F	Datum nach ISO 8601: '%Y-%´m-%d'
%c	Datum und Zeit formatiert als: '%F %T.%L'

1.6.2.1.13 MpAlarmXInfoType

Mit diesem Datentyp wird für die Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpAlarmXDiagType	Eine Diagnosestruktur für die StatusID

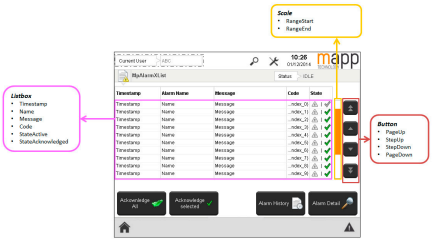
1.6.2.1.14 MpAlarmXListUIAlarmListType

Liste mit Alarmen und die dazugehörigen Navigation.

Parameter	Datentyp	Beschreibung
Severity	ARRAY[0.. 49] von UDINT	Level des Alarms
Code	ARRAY[0.. 49] von UDINT	Code des Alarms **VC4 Anbindung:** Datapoint von *Textsnippet* / Textgruppe: MpAlarmXListCode TextIndexOffset von *Listbox* / Textgruppe: MpAlarmXListCode **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
Message	ARRAY[0.. 49] von WSTRING[255]	Beschreibung des Alarms **VC4 Anbindung:** TextDatapoint von *Listbox* **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
AdditionalInformation1	ARRAY[0.. 49] von WSTRING[255]	Link mit zusätzlicher Information
AdditionalInformation2	ARRAY[0.. 49] von WSTRING[255]	Link mit zusätzlicher Information
InstanceID	ARRAY[0.. 49] von UDINT	Genauer Identifikationsnummer jedes Alarms (bei mehreren Instanzen desselben Alarms hilfreich)
Name	ARRAY[0.. 49] von STRING[255]	Name des Alarms **VC4 Anbindung:** TextDatapoint von *Listbox* **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
Scope	ARRAY[0.. 49] von STRING[255]	Scope des Alarms
StateActive	ARRAY[0.. 49] von BOOL	Zustandsänderung des Alarms **VC4 Anbindung:** IndexDatapoint von *Bitmap* **mappView Anbindung:** Datapoint von *TableItemImageList* / Data: selectedIndex
StateAcknowledged	ARRAY[0.. 49] von BOOL	Zustandsänderung des Alarms **VC4 Anbindung:** IndexDatapoint von *Bitmap* **mappView Anbindung:** Datapoint von *TableItemImageList* / Data: selectedIndex
Timestamp	ARRAY[0.. 49] von STRING[50]	Zeitstempel zum Zeitpunkt der Statusänderung. (Format über MpAlarmXListUISetupType adaptierbar) **VC4 Anbindung:** TextDatapoint von *Listbox* **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
SelectedIndex	UINT	Index des gerade ausgewählten Alarms **VC4 Anbindung:** IndexDatapoint von *Listbox*
MaxSelection	UINT	Index des letzten Alarms **VC4 Anbindung:** MaxDatapoint von *Listbox*
PageUp	BOOL	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten
PageDown	BOOL	Springt ans Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten

Parameter	Datentyp	Beschreibung
StepUp	BOOL	Wählt vorhergehenden Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten
StepDown	BOOL	Wählt nächsten Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 / ResetValue: 0 **mappView Anbindung:** EventBinding, für mehr Information siehe unten
RangeStart	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von Scale
RangeEnd	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von Scale

VC4 Anbindung:



none

Info:

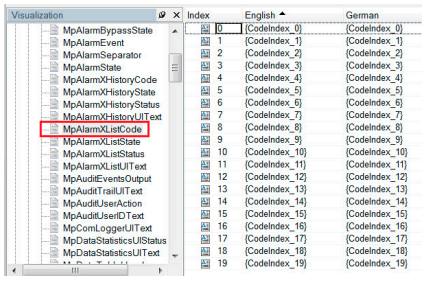


Abbildung 6: Textgruppe: MpAlarmXListCode

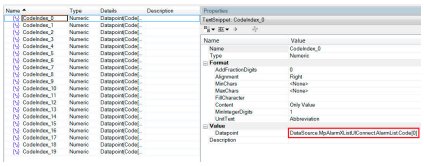
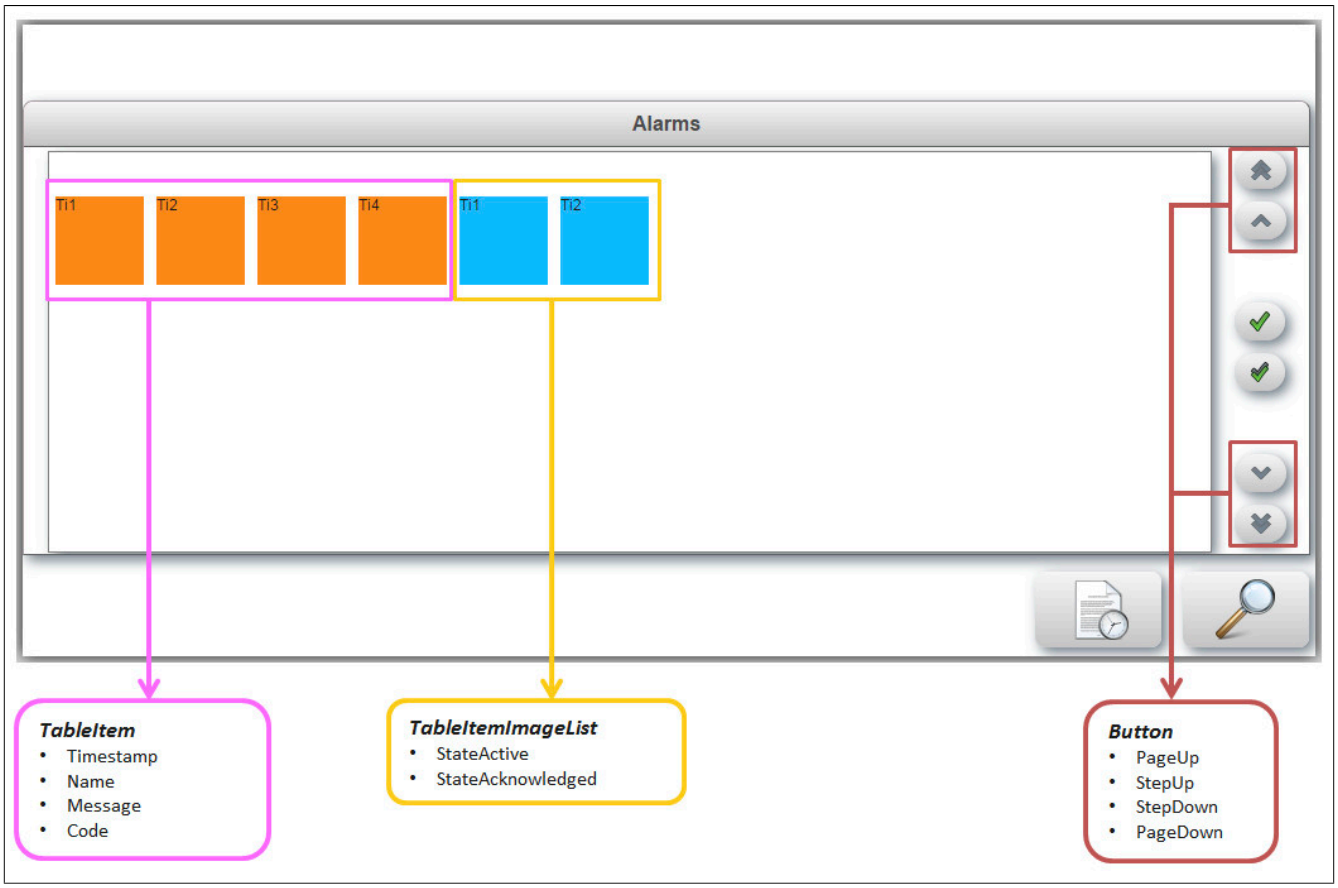


Abbildung 7: Textsnippets der Textgruppe

Abbildung 8: Verbindung zur Textgruppe

mappView Anbindung:



EventBinding

Um mithilfe der Buttons in der Tabelle nach oben bzw. nach unten navigieren zu können, muss ein EventBinding für den jeweiligen Button eingefügt werden.

PageUp

```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmPage" widgetRefId="PageUp" event="Cl
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXListUIConnect.AlarmList.PageUp"
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                    </Target>
                </Action>
            </EventHandler>
</EventBinding>
```

PageDown

```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmPage" widgetRefId="PageDown" event="
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXListUIConnect.AlarmList.PageDown
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                    </Target>
                </Action>
            </EventHandler>
</EventBinding>
```

StepUp

```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmPage" widgetRefId="StepUp" event="Cl
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXListUIConnect.AlarmList.StepUp"
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
```

```
                    </Target>
                </Action>
            </EventHandler>
</EventBinding>
```

StepDown

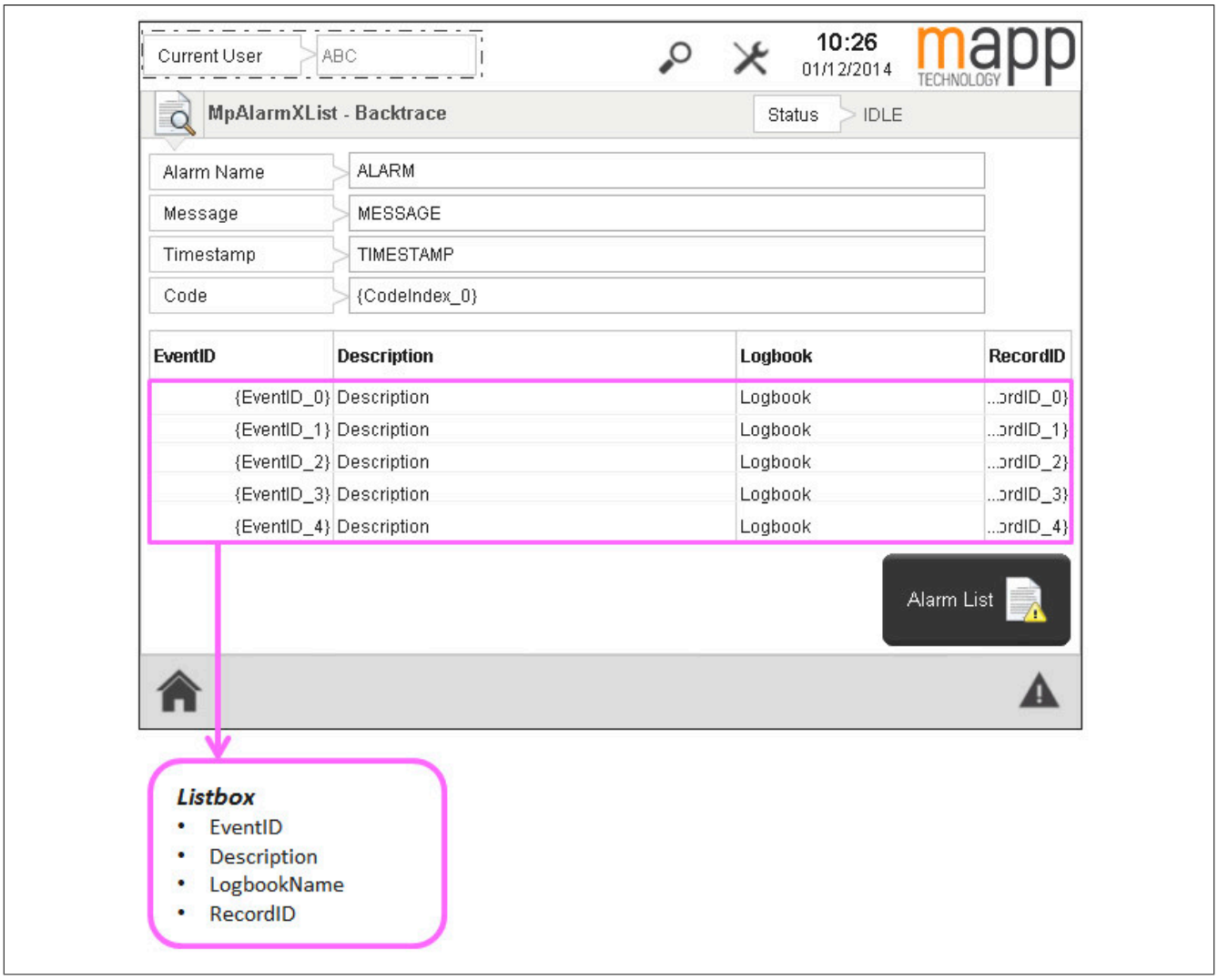
```
<EventBinding>
            <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmPage" widgetRefId="StepDown" event="
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXListUIConnect.AlarmList.StepDown
                        <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                    </Target>
                </Action>
            </EventHandler>
</EventBinding>
```

1.6.2.1.15 MpAlarmXListUIBacktraceType

Über diesen Datetype wird versucht soweit möglich die Fehlerursache für den Alarm anzugeben. Die Ursache wird dabei aus dem Logger ausgelesen. Über die "RecordID", sowie die "OriginID" werden alle Loggereinträge, welche den ausgewählten Alarm betreffen, ausgeforscht und angezeigt.

Parameter	Datentyp	Beschreibung
RecordID	`ARRAY[0..4] von UDINT`	RecordIDs der Fehler **VC4 Anbindung:** Datapoint von *Textsnippet* / Textgruppe: MpAlarmXListRecordID TextIndexOffset von *Listbox* **mappView Anbindung:** Datapoint von *TableItem* / Data: value
LogbookName	`ARRAY[0..4] von STRING[100]`	Logbuchname, in welchem die Fehler vermerkt sind **VC4 Anbindung:** Datapoint von *Listbox* / TextDatapoint **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue
EventID	`ARRAY[0..4] von DINT`	EventIDs der Fehler **VC4 Anbindung:** Datapoint von *Textsnippet* / Textgruppe: MpAlarmXListEventID TextIndexOffset von *Listbox* **mappView Anbindung:** Datapoint von *TableItem* / Data: value
Description	`ARRAY[0..4] von STRING[255]`	Beschreibung der Fehler **VC4 Anbindung:** Datapoint von Listbox / *TextDatapoint* **mappView Anbindung:** Datapoint von *TableItem* / Data: stringValue

VC4 Anbindung:



Info:

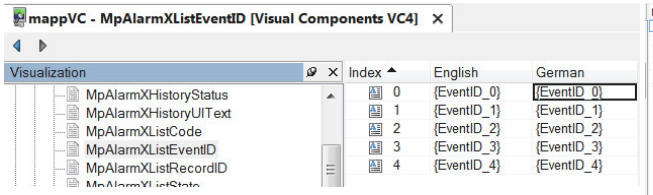


Abbildung 9: Textgruppe: MpAlarmXListEventID

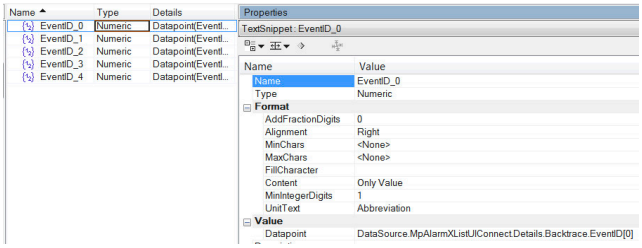


Abbildung 10: Textsnippets der Textgruppe

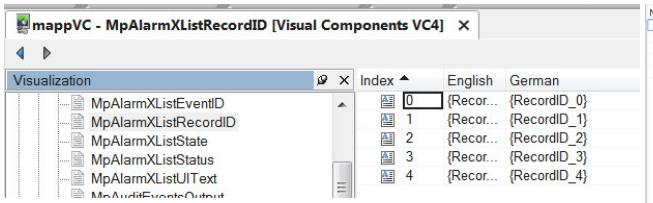
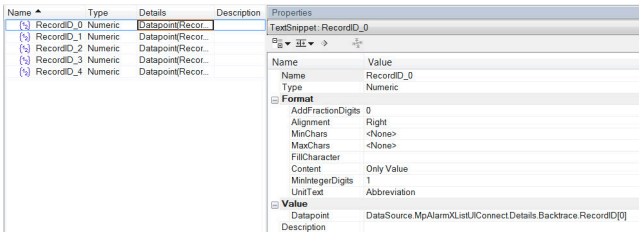
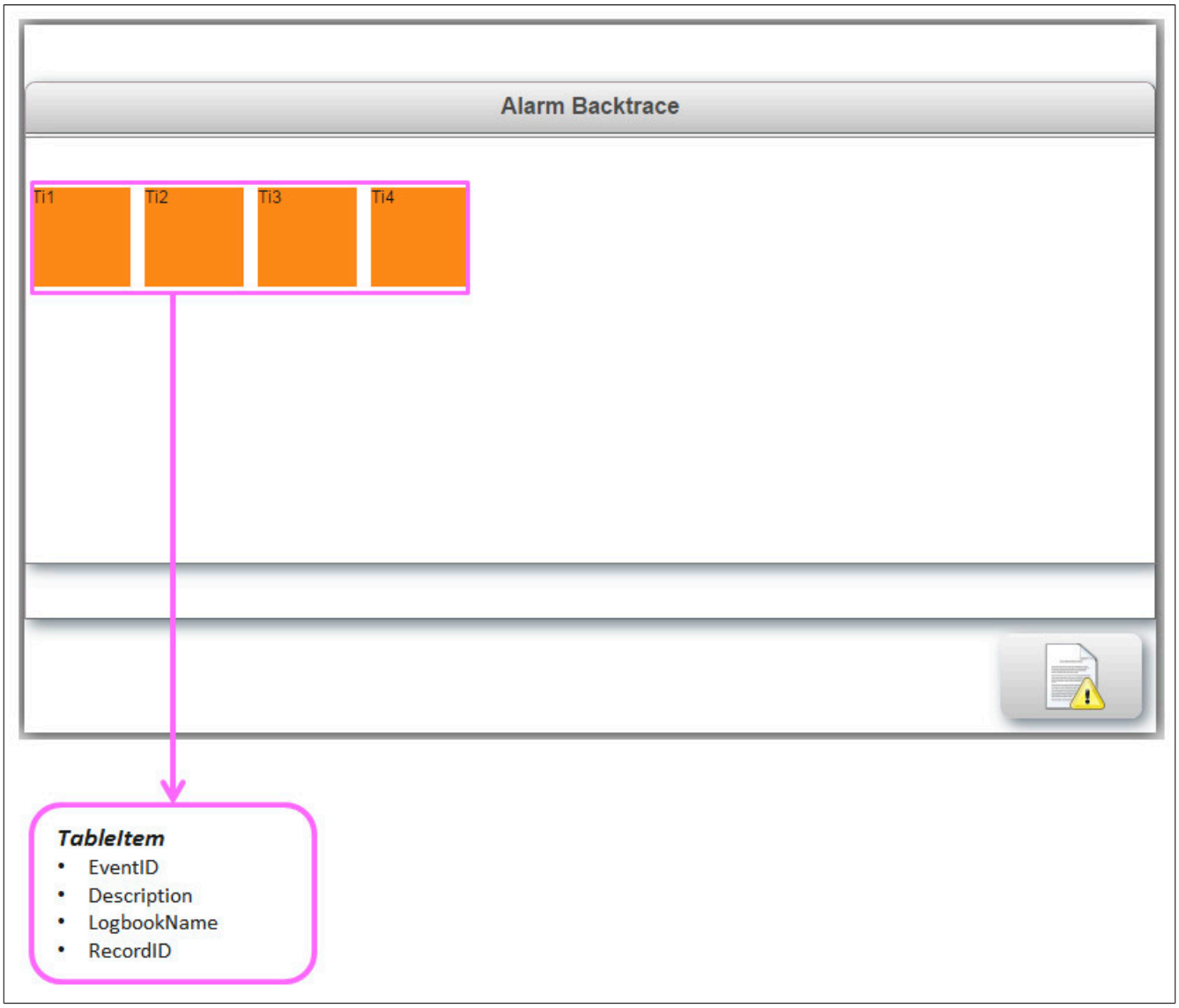


Abbildung 11: Textgruppe: MpAlarmXListRecordID



mappView Anbindung:

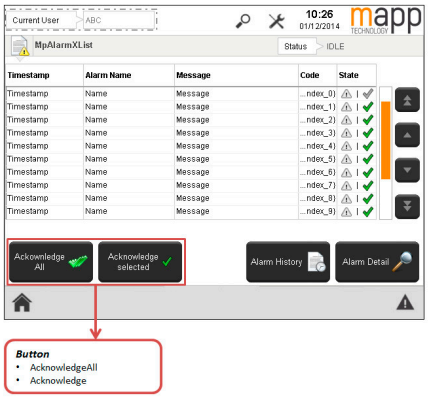


1.6.2.1.16 MpAlarmXListUIConnectType

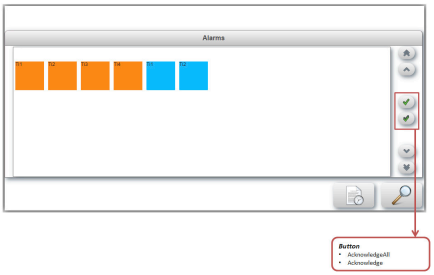
Mithilfe dieses Datentypes wird eine Verbindung zwischen MpAlarmXListUI und der Visualisierung erstellt.

Parameter	Datentyp	Beschreibung
Status	`MpAlarmXListUIStatusEnum`	Aktuelle Operation
AlarmList	`MpAlarmXListUIAlarmListType`	Liste der aktiven bzw. nicht quittierten Alarme
AcknowledgeAll	`BOOL`	Quittiert alle Alarme **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 **mappView Anbindung:** EventBinding, für mehr Information siehe Beispielcode unten
AcknowledgeID	`UDINT`	Quittiert den, über die Instanz ID angegebenen, Alarm **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: Instand ID
Acknowledge	`BOOL`	Quittiert den ausgewählten Alarm **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1 **mappView Anbindung:** EventBinding, für mehr Information siehe Beispielcode unten
Details	`MpAlarmXListUIDetailsType`	Details zu dem Alarm, welcher in der Liste ausgewählt ist
Language	`STRING[20]`	Definiert, in welcher Sprache bzw. Einheit die Alarme angezeigt werden

VC4 Anbindung:



mappView Anbindung:



EventBinding für Button "Acknowledge"

```
<EventBinding>
    <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmPage" widgetRefId="Acknowledge" event="Clic
        <EventHandler>
            <Action>
                <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXListUIConnect.Acknowledge" >
                    <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                </Target>
            </Action>
        </EventHandler>
</EventBinding>
```

EventBinding für Button "AcknowledgeAll"

```
<EventBinding>
    <Source xsi:type="widgets.brease.Button.Event" contentRefId="AlarmPage" widgetRefId="AcknowledgeAll" event="Cl
        <EventHandler>
            <Action>
                <Target xsi:type="opcUa.NodeAction" refId="::AsGlobalPV:AlarmXListUIConnect.AcknowledgeAll" >
                    <Method xsi:type="opcUa.NodeAction.SetValueBool" value="true" />
                </Target>
            </Action>
        </EventHandler>
</EventBinding>
```

1.6.2.1.17 MpAlarmXListUIDetailsType

Struktur mit zusätzlichen Informationen zum aktuell in der Liste ausgewählten Alarm.

Parameter	Datentyp	Beschreibung
Severity	`UDINT`	Level des Alarms
Code	`UDINT`	Code des Alarms
Message	`WSTRING[255]`	Beschreibung des Alarms
AdditionalInformation1	`WSTRING[255]`	Link mit zusätzlicher Information
AdditionalInformation2	`WSTRING[255]`	Link mit zusätzlicher Information
InstanceID	`UDINT`	Genauer Identifikationsnummer jedes Alarms (bei mehreren Instanzen desselben Alarms hilfreich)
Name	`STRING[255]`	Name des Alarms
Scope	`STRING[255]`	Scope des Alarms

Parameter	Datentyp	Beschreibung
StateActive	BOOL	Zeigt an, ob der Alarm aktiv ist
StateAcknowledged	BOOL	Zeigt an, ob der Alarm bereits quittiert wurde
Timestamp	STRING[50]	Zeitpunkt, an welchem der Alarm aktiv wurde. (Format über MpAlarmXListUISetupType adaptierbar)
Backtrace	MpAlarmXListUIBacktraceType	Details zur Fehlerursache (Quelle: Logger)

1.6.2.1.18 MpAlarmXListUISetupType

Zusätzliche Konfigurationsmöglichkeiten, um die Visualisierung zu beschreiben.

Parameter	Datentyp	Standardwert	Beschreibung
AlarmListSize	UINT	20	Zahl an Alarmen die auf einer Seite der Liste auf der Visualisierung dargestellt werden können Es können maximal 50 Alarme angezeigt werden
AlarmListScrollWindow	USINT	1	Zeigt an, wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden
TimeStampPattern	STRING[50]	'%Y-%m-%d %H:%M:%S:%L'	Format, in welchem der Zeitstempel auf der Visualisierung dargestellt wird

Der Parameter "TimeStampPattern" gibt an, in welchem Format der Zeitstempel über die Visualisierung angezeigt wird:

- Standardformat: %Y %m %d %H:%M:%S:%L (z.B. '1990 25 03:13:27:253')
- Angepasstes Format (Beispiel): %d.%m.%Y %H:%M:%S:%L (z.B. '25.02.1991 13:27:03:561')

Folgende weitere Möglichkeiten existieren zum Anpassen des Formats:

Kennung	Beschreibung
%H	Stunde im 24 Stunden Format. Null am Anfang falls notwendig: 00 - 23
%I	Stunde im 12 Stunden Format. Null am Anfang falls notwendig: 01- 12
%k	Stunde im 24 Stunden Format: 0 - 23
%M	Minute innerhalb einer Stunde. Null am Anfang falls notwendig: 00 - 59
%S	Sekunden innerhalb einer Minute. Null am Anfang falls notwendig: 00 - 59
%L	Millisekunden innerhalb einer Sekunde. Null am Anfang falls notwendig: 000 - 999
%s	Sekunden seit 1 Januar 1970
%C	Vierstellige Jahreszahl dividiert durch 100. Über zwei Ziffern angezeigt. Null am Anfang falls notwendig: 00 - 99
%Y	Jahr, formatiert als vierstellige Zahl. Null am Anfang falls notwendig
%y	Die letzten beiden Ziffern des Jahres. Null am Anfang falls notwendig: 00 - 99
%m	Monat. Null am Anfang falls notwendig: 01 - 31
%d	Tag des Monats. Null am Anfang falls notwendig: 01 - 31
%e	Tag des Monats: 1 - 31
%R	Zeit als 24 Stunden Anzeige: '%H:%M'
%T	Zeit als 24 Stunden Anzeige: '%H:%M:%S'
%D	Datum formatiet als: '%m/%d/%y'
%F	Datum nach ISO 8601: '%Y-%´m-%d'
%c	Datum und Zeit formatiert als: '%F %T.%L'

1.6.2.1.19 MpAlarmXMappingConfigType

Dieser Datentyp beschreibt alle Aktionen, welche auf einen Auslöser (z.B. Alarm wird aktiv) hin durchgeführt werden.

Wie das Konfigurationsupdate der einzelnen Parameter zur Laufzeit durchgeführt wird, ist im Abschnitt Konfiguration während der Laufzeit erklärt.

Parameter	Datentyp	Beschreibung
Action	ARRAY[0..19] von MpAlarmXActionConfigType	Liste der Aktionen, welche nach einem definierten Auslöser ausgeführt werden sollen

1.6.2.1.20 MpAlarmXQueryInfoType

Mit diesem Datentyp wird für den MpAlarmXQuery Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
AvailableSpace	UDINT	Maximale Anzahl an Einträgen, welche in der PV angegeben werden können.
RowsRead	UDINT	Anzahl der gelesenen Einträgen.
HasMoreRows	BOOL	Gibt an, ob noch mehr Information vorhanden ist oder nicht.
Diag	MpAlarmXDiagType	Eine Diagnosestruktur für die StatusID

1.6.2.1.21 MpAlarmXStatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	MpAlarmXErrorEnum	Fehlerkode der mapp Komponente
Severity	MpComSeveritiesEnum	Schwere des Fehlers
Code	UINT	Kode der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden

1.6.2.1.22 MpAlarmXUpdateBehaviorType

Über diesen Datentypen kann das Aktualisierungs-Verhalten des Alarms definiert werden

Parameter	Datentyp	Beschreibung
Activation	MpAlarmXActivationUpdateType	Gibt an, welche Alarminformation aktualisiert werden soll.

1.6.2.2 Enumeratoren

1.6.2.2.1 MpAlarmXAcknowledgeConfigEnum

Dieser Aufzählungsdatentyp gibt an, ob und wie sich ein Alarm quittiern lässt.

Aufzählung	Beschreibung
mpALARMX_ACK_DISABLED	Keine Quittierung notwendig, inaktive Alarme verschwinden automatisch aus der Alarmliste
mpALARMX_ACK_REQ	Der Alarm muss quittiert werden. Die Quittierung ist jederzeit möglich
mpALARMX_ACK_REQ_AFTER_ACTIVE	Der Alarm muss quittiert werden. Die Quittierung kann erst erfolgen, wenn der Alarm nicht mehr aktiv ist
mpALARMX_ACK_REQ_RESETABLE	Der Alarm muss quittiert werden. Der Quittier-Zustand wird zurückgesetzt sobald der Alarm nocheinmal aktiviert wurde.

1.6.2.2.2 MpAlarmXActionEnum

Dieser Aufzählungsdatentyp beschreibt die unterschiedlichen Aktionen, welche beim Alarm Mapping als Reaktion auf einen Auslöser definiert werden können.

Aufzählung	Beschreibung
mpALARMX_ACT_NONE	Keine Aktion
mpALARMX_ACT_REACTION	Es soll eine Reaction ausgelöst werden.
mpALARMX_ACT_ESCALATE_ALARM	Der Alarm soll an das nächste Level eskaliert werden. Siehe Escalate Alarm.
mpALARMX_ACT_ESCALATE_REACTION	Die Reaktion soll an das nächste Level eskaliert werden. Siehe Escalate Reaction.
mpALARMX_ACT_REPLACE_ALARM	Der Alarm soll an das nächste Level eskaliert werden und dabei durch einen anderen Alarm ersetzt werden. Siehe Replace Alarm.
mpALARMX_ACT_SEND_MESSAGE	Der Alarm soll über das Nachrichtensystem MpTweet versendet werden. Für mehr Information siehe Send Message.
mpALARMX_ACT_REMAIN	Der Alarm soll lokal behandelt werden. Siehe Remain.
mpALARMX_ACT_AGGREGATE	Der Alarm soll an das nächste Level eskaliert und aggregiert werden. Siehe Aggregate Alarm and Escalate.
mpALARMX_ACT_AGGREGATE_LOCAL	Der Alarm wird aggregiert. Siehe Aggregate Alarm locally.

1.6.2.2.3 MpAlarmXHistoryUIStatusEnum

Dieser Aufzählungsdatentyp gibt Auskunft über die aktuelle Aktivität der Komponente.

Aufzählung	Beschreibung
mpALARMX_HISTORY_UI_STATUS_IDLE	Aktuell ist kein Prozess aktiv
mpALARMX_HISTORY_UI_STATUS_ERROR	Die letzte Operation hat einen Fehler generiert

1.6.2.2.4 MpAlarmXListUIStatusEnum

Dieser Aufzählungsdatentyp gibt Auskunft über die aktuelle Aktivität der Komponente.

Aufzählung	Beschreibung
mpALARMX_LIST_STATUS_IDLE	Aktuell ist kein Prozess aktiv
mpALARMX_LIST_STATUS_ERROR	Die letzte Operation hat einen Fehler generiert

1.6.2.2.5 MpAlarmXMappingTypeEnum

Dieser Enumerator gibt an, welcher Auslöser eine bestimmte Aktion auslöst.

Parameter	Beschreibung
mpALARMX_MAPPING_TYPE_ALARM_NAME	Der Auslöser ist ein Alarm
mpALARMX_MAPPING_TYPE_SEVERITY	Der Auslöser ist die Severity
mpALARMX_MAPPING_TYPE_DEFAULT	Die Default Action soll verwendet werden

1.6.2.2.6 MpAlarmXQueryModeEnum

Dieser Aufzählungsdatentyp gibt an wie die Alarmliste durchsucht werden soll.

Aufzählung	Beschreibung
mpALARMX_QUERY_MODE_ALL	Es werden alle Einträge in der Alarmliste durchsucht.
mpALARMX_QUERY_MODE_NEW	Reserviert für spätere Nutzung. Wird dieser Modus verwendet, wird die Alarmliste wie im Modus "mpALARMX_QUERY_MODE_ALL" sortiert.

1.6.2.2.7 MpAlarmXStateEnum

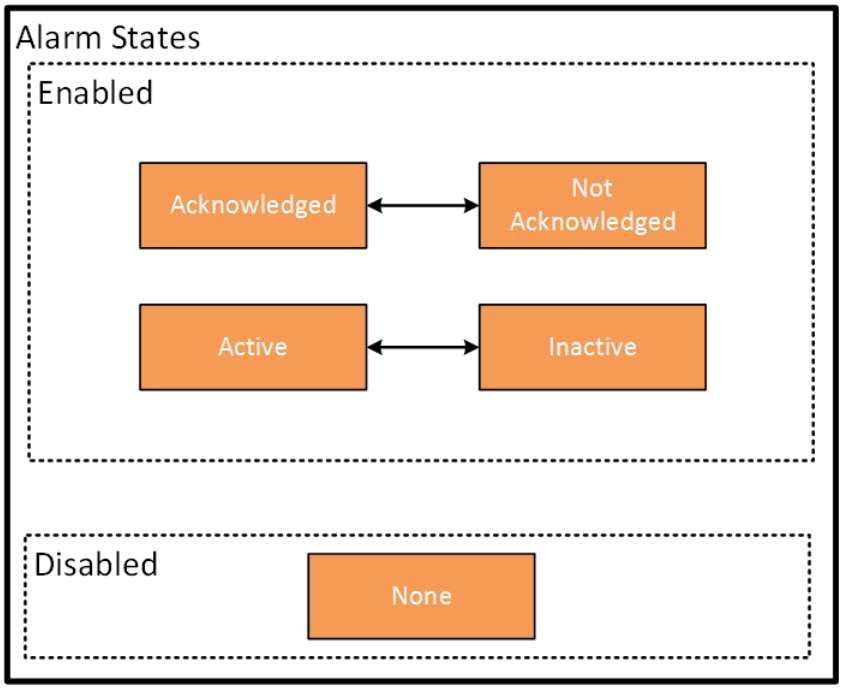
Dieser Aufzählungsdatentyp gibt Auskunft über die aktuelle Aktivität der Komponente. Für mehr Information zu den Aufzählungstypen siehe Alarm Grundkonzept.

Wie die verschiedenen Typen erreicht werden können ist im unteren Abschnitt "Grundzustand eines Alarms" erklärt!

Aufzählung	Beschreibung
mpALARMX_STATE_NONE	Alarm ist nicht eingeschaltet
mpALARMX_STATE_ACTIVE	Alarm ist aktiv
mpALARMX_STATE_INACTIVE	Alarm ist inaktiv
mpALARMX_STATE_ACKNOWLEDGED	Alarm ist quittiert
mpALARMX_STATE_UNACKNOWLEDGED	Alarm ist nicht quittiert

Grundzustand eines Alarms

Folgende Grundzustände kann ein Alarm annehmen:



Die Alarmzustände treten in Kombination miteinander auf. Je nachdem in welcher Reihenfolge der Alarm aktiviert, zurückgesetzt und quittiert wird, können folgende Kombinationen miteinander auftreten:

Alarm aktivieren -> Alarm zurücksetzen -> Alarm quittieren

- **Alarm aktivieren:** Wird der Alarm mithilfe von MpAlarmXSet aktiviert, So hat der Alarm die Zustände "Acitve" und "Not Acknowledged".
- **Alarm zurücksetzen:** Wird der Alarm mithilfe von MpAlarmXReset danach zurückgesetzt, so hat der Alarm die Zustände "Inactive" und "Not Acknowledged".
- **Alarm quittieren:** Wurde anschließend der Alarm quittiert, so hat der Alarm den Zustand "None". Die Quittierung kann in der Applikation über MpAlarmXAcknowledge oder in der Visualisierung über MpAlarmXListUI erfolgen.

Alarm aktivieren -> Alarm quittieren -> Alarm zurücksetzen

- **Alarm aktivieren:** Wird der Alarm mithilfe von MpAlarmXSet aktiviert, So hat der Alarm die Zustände "Acitve" und "Not Acknowledged"
- **Alarm quittieren:** Wurde anschließend der Alarm quittiert, so hat der Alarm die Zustände "Active" und "Acknowledged". Die Quittierung kann in der Applikation über MpAlarmXAcknowledge oder in der Visualisierung über MpAlarmXListUI erfolgen.
- **Alarm zurücksetzen:** Wird der Alarm mithilfe von MpAlarmXReset danach zurückgesetzt, so hat der Alarm den Zustand "None".

Den Status eines Alarms kann man über die Funktion MpAlarmXCheckState herausfinden. Dabei wird einer der obigen fünf Status angegeben und man bekommt boolsches Feedback, ob der angegebene Status aktiv ist. Weiters kann über die Alarm Konfiguration festgelegt werden, ob ein Alarm quittiert werden, muss oder nicht und auch ob ein Alarm zurückgesetzt werden muss oder nicht. Wird ein Alarm über die Konfiguration deaktiviert ("Disable"), so hat der Alarm den Zustand "None".

1.6.3 Statusnummern

1.6.3.1 1083183950: Abfrage abgebrochen

Beschreibung:

Die Abfrage wurde abgebrochen bevor alle Daten gelesen werden konnten.

Ursache / Abhilfe:

- Abfrage erneut ausführen

1.6.3.2 1083367432: MpAlarmXCore oder MpAlarmXHistory ist noch nicht aktiv

Beschreibung:

Der Funktionsbaustein kann erst aktiv werden, wenn MpAlarmXCore bzw. MpAlarmXHistory aktiv ist.

Reaktion:

Der Funktionsbaustein wird diese Information an dem Ausgang "StatusID" anzeigen.

Ursache / Abhilfe:

- MpAlarmXCore bzw. MpAlarmXHistory aktivieren

Konstante:

`mpALARMX_INF_WAIT_CORE_FB`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXHistory
- MpAlarmXListUI
- MpAlarmXHistoryUI

1.6.3.3 1083183915: Speichergröße verändert

Beschreibung:

Die Speichergröße des History-Datenobjektes wurde in der Konfiguration verändert, das Datenobjekt wurde angepasst.

1.6.3.4 1083183914: Speichertyp geändert

Beschreibung:

Der Speichertyp des History-Datenobjektes wurde in der Konfiguration verändert, das Datenobjekt wurde verlagert.

1.6.3.5 1083367438: Keine Daten vorhanden

Beschreibung:

Es sind keine Daten zur angegebenen Abfrage verfügbar.

Konstante

`mpALARMX_INF_QUERY_NO_DATA`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXQuery

1.6.3.6 -2137858045: Fehlender Wert an UIConnect

Beschreibung:

Fehlender Wert an UIConnect

Reaktion

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Prozessvariable für den Funktionsbaustein-Eingang "UIConnect" definieren

Konstante:

`mpALARMX_WRN_MISSING_UICONNECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXListUI
- MpAlarmXHistoryUI

1.6.3.7 -2138041509: SRAM nicht gefunden

Beschreibung:

Es konnte kein SRAM Buffer initialisiert werden. Remanente Daten könnten verloren gehen.

Ursache / Abhilfe:

- SRAM Buffer konnte nicht gefunden werden. Kann nach einem Kaltstart ausgelöst werden.
- Überprüfen ob Speichertyp von "UserROM" auf "BufferedUserROM" umgestellt wurde während der Funktionbaustein deaktiviert war.

1.6.3.8 -2138041538: Konfiguration nicht gefunden

Beschreibung:

Für den Alarm {2:Name} konnte keine Konfiguration gefunden werden. Es wird eine Standardkonfiguration genutzt.

Zusätzliche Informationen

- {2:Name}: Name des Alarms

Ursache / Abhilfe:

- Überprüfen, ob Alarm richtig angegeben wurde

1.6.3.9 -2138041537: MpAlarmBasic erkannt

Beschreibung:

Es wurde eine aktive MpAlarmBasic Komponente entdeckt. MpAlarmBasic und MpAlarmX können nicht im gleichen Projekt benutzt werden.

Ursache / Abhilfe:

- Deaktivieren / entfernen der MpAlarmBasic Komponente

1.6.3.10 -2138041536: Kein Messengerservice gefunden

Beschreibung:

Es wurde kein Messengerservice gefunden als versucht wurde die SendMessage Aktion auszuführen.

Ursache / Abhilfe:

- Einfügen von MpTweet - Nachrichtensystem
- Überprüfen ob MpTweet aktiv ist und mit einem Modem verbunden ist

1.6.3.11 -2138041535: Fehler beim Ausführen von SendMessage

Beschreibung:

Der Messengerservice meldet einen Fehler beim Ausführen der SendMessage Aktion.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

1.6.3.12 -2138041532: Ungültige Monitoring Konfiguration

Beschreibung:

Die angegebene Monitoring-Konfiguration ist ungültig. Es muss ein Monitoring-Name vergeben werden.

Ursache / Abhilfe:

- Monitoring-Name in der MpAlarmXCore Konfiguration angeben

1.6.3.13 -2138041531: Ungültige Monitoring PV-Name

Beschreibung:

Der angegebene PV-Name für das Alarm-Monitoring ist ungültig. Die PV {2:PVName} wurde nicht gefunden.

Ursache / Abhilfe:

- Überprüfen des Variablen-Namen, welcher in der MpAlarmXCore Konfiguration angeben wurde

1.6.3.14 -2138041530: Limit wird nicht überwacht

Beschreibung:

In der Monitoring-Konfiguration wurde beim dynamischen Limit keine PV angegeben. Limit wird nicht überwacht.

Ursache / Abhilfe:

- Variable für das Limit in der MpAlarmXCore Konfiguration angeben

1.6.3.15 -2138041529: Angabe einer ungültigen PV

Beschreibung:

In der Monitoring Konfiguration wurde für das dynamische Limit eine ungültige PV angegeben. Die PV {2:PVName} wurde nicht gefunden.

Zusätzliche Informationen

- {2:PVName}: PV Name

Ursache / Abhilfe:

- Überprüfen der angegebenen Variable in der MpAlarmXCore Konfiguration

1.6.3.16 -2138041528: Monitoring-Konfiguration ungültig

Beschreibung:

Die Monitoring-Konfiguration ist ungültig. Der angegebene PV Typ {3:PVType} der Variable {2:PVName} wird nicht unterstützt.

Zusätzliche Informationen

- {2:PVName}: PV Name
- {3:PVType}: Datentyp

Ursache / Abhilfe:

- Überprüfen der angegebenen Variable in der MpAlarmXCore Konfiguration
- Die unterstützten Datentypen sind hier zu finden.

1.6.3.17 -2138041527: Trigger-Variable ungültig

Beschreibung:

Bei der Konfiguration der Discrete-Value-Monitoring wurde eine ungültige Trigger-Variable angegeben. {2:Trigger} ist kein gültiger Bool-Wert.

Zusätzliche Informationen

- {2:Trigger}: Angegebener Trigger

Ursache / Abhilfe:

- Überprüfen des angegebenen Triggers in der MpAlarmXCore Konfiguration

1.6.3.18 -2138041526: Trigger-Variable ungültig

Beschreibung:

Bei der Konfiguration der Discrete-Value-Monitoring wurde eine ungültige Trigger-Variable angegeben. {2:Trigger} ist kein gültiger Integer-Wert.

Zusätzliche Informationen

- {2:Trigger}: Angegebener Trigger

Ursache / Abhilfe:

- Überprüfen des angegebenen Triggers in der MpAlarmXCore Konfiguration

1.6.3.19 -2138041525: Alarm kann nicht aktiviert werden

Beschreibung:

Der Aalrm {2:AlarmName} kann nicht aktiviert werden, auf den Alarm wird doppelt zugegriffen.

Zusätzliche Informationen

- {2:Name}: Name des Alarms

Ursache / Abhilfe:

- Überprüfen, ob der Alarm aktuell aktiviert wird und gleichzeitig über MpAlarmXConfigAlarm konfiguriert wird
- Wird der Alarm bereits aktiviert obwohl MpAlarmXCore noch nicht aktiv ist, kann es zu diesem Problem kommen

1.6.3.20 -2138041524: Hysterese ungültig

Beschreibung:

Die Hysterese {2:Hysteresis} in der Monitoring-Konfiguration ist ungültig. Die Hysterese kann nicht negativ sein.

Zusätzliche Informationen

- {2:Hysteresis}: Angegebene Hysterese

Ursache / Abhilfe:

- Überprüfen der MpAlarmXCore Konfiguration

1.6.3.21 -2138041521: Mapping Konfiguration ungültig

Beschreibung:

Die Mapping-Konfiguration ist ungültig. Der angegebene Alarm-Monitoring Typ {2:Kind} ist nicht gültig.

Zusätzliche Informationen

- {2:Kind}: Name der Kategorie

Ursache / Abhilfe:

- Überprüfen der MpAlarmXCore Konfiguration

1.6.3.22 -2138041517: Verzögerungs PV leer

Beschreibung:

Die Verzögerungs-Prozessvariable ("Delay"), welche in der MpAlarmXCore Konfiguration angegeben wird, ist leer.

Die Prozessvariable wird bei Monitored-Alarmen verwendet.

Ursache / Abhilfe:

- Angeben einer Prozessvariable

1.6.3.23 -2138041516: Verzögerungs PV nicht gefunden

Beschreibung:

Die Verzögerungs-Prozessvariable ("Delay") in der MpAlarmXCore Konfiguration wurde nicht gefunden.

Die Prozessvariable wird bei Monitored-Alarmen verwendet.

Ursache / Abhilfe:

- Überprüfen der MpAlarmXCore Konfiguration
- Überprüfen der Prozessvariable
- berprüfen ob Prozessvariable im Programm verwendet wird

1.6.3.24 -2138041515: Hysterese PV leer

Beschreibung:

Die Prozessvariable für die Hysterese, welche in der MpAlarmXCore Konfiguration definiert wird, ist leer.

Die Prozessvariable wird bei Monitored-Alarmen verwendet.

Ursache / Abhilfe:

- Angeben einer Prozessvariable

1.6.3.25 -2138041514: Hysterese PV nicht gefunden

Beschreibung:

Die Hysterese-Prozessvariable wurde in der MpAlarmXCore Konfiguration nicht gefunden.

Die Prozessvariable wird bei Monitored-Alarmen verwendet.

Ursache / Abhilfe:

- Überprüfen der MpAlarmXCore Konfiguration
- Überprüfen der angegebenen Prozessvariable

1.6.3.26 -2138041513: PV für Zeitkonstante leer

Beschreibung:

Die Prozessvariable für die Zeitkonstante ("Time constant"), welche in der MpAlarmXCore Konfiguration definiert wird, ist leer.

Die Prozessvariable wird bei Monitored-Alarmen verwendet.

Ursache / Abhilfe:

- Angeben einer Prozessvariable

1.6.3.27 -2138041512: PV für Zeitkonstante nicht gefunden

Beschreibung:

Die Prozessvariable für die Zeitkonstante ("Time constant"), welche in der MpAlarmXCore Konfiguration definiert wird, konnte nicht gefunden werden.

Die Prozessvariable wird bei Monitored-Alarmen verwendet.

Ursache / Abhilfe:

- Überprüfen der MpAlarmXCore Konfiguration
- Überprüfen der Prozessvariable

1.6.3.28 -2137858033: Daten inkonsistent

Beschreibung:

Die Daten der Abfrage sind seit dem letzten Abfruf inkonsistent.

Reaktion

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Abfrage mit MpAlarmXQuery nochmal neu ausführen

Konstante:

`mpALARMX_WRN_QUERY_INCONSISTENT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXQuery

1.6.3.29 -1064299732: Fehler beim Ändern des Speichertyps

Beschreibung:

Der Speichertyp des Datenobjektes konnte nicht geändert werden.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers. Fehlernummer ist aus der Bibliothek DataObj

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.30 -1064299731: Fehler beim Erstellen

Beschreibung:

Fehler beim Erstellen eines neuen Verlaufsdatenobjektes.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers. Fehlernummer ist aus der DataObj Bibliothek

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.31 -1064299730: Fehler beim Öffnen

Beschreibung:

Fehler beim Ändern der Datenobjekt-Größe. Temporäres Datenobjekt kann nicht angelegt werden. Fehlerursache {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.32 -1064299729: Fehler beim Erstellen

Beschreibung:

Fehler beim Ändern der Datenobjekt-Größe. Neues Datenobjekt kann nicht angelegt werden.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.33 -1064299728: Fehler beim Initialisieren

Beschreibung:

Fehler beim Ändern der Datenobjekt-Größe. Neues Datenobjekt kann nicht formatiert werden.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.34 -1064299727: Fehler beim Kopieren

Beschreibung:

Fehler beim Ändern der Datenobjekt-Größe. Datenobjekt kann nicht kopiert werden.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.35 -1064299726: Fehler beim Löschen

Beschreibung:

Fehler beim Ändern der Datenobjekt-Größe. Altes Datenobjekt kann nicht gelöscht werden.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.36 -1064299725: Fehler beim Öffnen oder Initialisieren

Beschreibung:

Fehler biem Öffnen oder Initialisieren des Verlaufsdateiformates.

Zusätzliche Informationen

- {1:ErrorNumber}: Interne Fehlernummer

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.37 -1064299724: Fehler beim Erstellen einer neuen Exportdatei

Beschreibung:

Fehler beim Erstellen einer neuen Exportdatei {2:File} für {3:Device}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers
- {2:File}: Zu erstellende Datei
- {3:Device}: Das Gerät, das die Datei nutzen sollte

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.38 -1064299723: Fehler beim Schreiben der Exportdatei

Beschreibung:

Fehler beim Schreiben der Exportdatei {2:File} für {3:Device}.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers
- {2:File}: Zu schreibende Datei
- {3:Device}: Das Gerät, das die Datei nutzen sollte

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

1.6.3.39 -1064299718: Konfiguration kann nicht gelesen werden

Beschreibung:

Ein Abschnitt in der Konfiguration kann nicht gelesen werden.

Ursache / Abhilfe:

- Überprüfen der Konfiguration
- Ein Abschnitt in der Konfiguration kann nicht gelesen werden

1.6.3.40 -1064299716: PV nicht gefunden

Beschreibung:

Die Prozessvariable mit dem Namen {2:PVName} konnte nicht gefunden werden und ist nicht als Alarm Textsnippet verfügbar.

Zusätzliche Informationen

- {2:PVName}: Name der Prozessvariable

Ursache / Abhilfe:

- Überprüfen sie die Schreibweise des Textsnippets
- Überprüfen der Variable welche angegeben wurde

1.6.3.41 -1064299715: PV nicht unterstützter Datentyp

Beschreibung:

Alarm Textsnippet mit der PV {2:PVName} nutzt einen nicht unterstützten Datentyp.

Zusätzliche Informationen

- {2:PVName}: Name der Prozessvariable

Ursache / Abhilfe:

- Ändern des Datentyps der Prozessvariable. Für mehr Information zu den nicht unterstützten Datentypen siehe "Einbinden von Applikationsdaten in den Alarmtext" in Alarm Texte.

1.6.3.42 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXCore
- MpAlarmXHistory

Konstante:

`mpALARMX_ERR_ACTIVATION`

1.6.3.43 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXCore
- MpAlarmXCheckReaction
- MpAlarmXSet
- MpAlarmXReset
- MpAlarmXHistory

Konstante:

`mpALARM_ERR_MPLINK_NULL`

1.6.3.44 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXCore
- MpAlarmXCheckReaction
- MpAlarmXSet
- MpAlarmXReset
- MpAlarmXHistory

Konstante:

`mpALARMX_ERR_MPLINK_INVALID`

1.6.3.45 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXCore
- MpAlarmXHistory

Konstante:

`mpALARMX_ERR_MPLINK_CHANGED`

1.6.3.46 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXCore
- MpAlarmXCheckReaction

- MpAlarmXSet
- MpAlarmXReset
- MpAlarmXHistory

Konstante:

`mpALARMX_ERR_MPLINK_CORRUPT`

1.6.3.47 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXCore
- MpAlarmXHistory

Konstante:

`mpALARMX_ERR_MPLINK_IN_USE`

1.6.3.48 -1064239091: Ungültige Konfiguration

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Konfiguration ist beschädigt
- Die Funktion ist in der Konfiguration nicht aktiviert

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXCore
- MpAlarmXHistory

Konstante:

`mpALARMX_ERR_CONFIG_INVALID`

1.6.3.49 -1064239095: Konfigurations Zeiger ist keine PV

Beschreibung:

Der angegebene Zeiger ist kein Zeiger auf eine gültige PV. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "Configuration" Eingang prüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXConfigMapping
- MpAlarmXConfigAlarm

Konstante:

`mpALARMX_ERR_CONFIG_NO_PV`

1.6.3.50 -1064116224: Angegebener Name ist NULL

Beschreibung:

Der angegebene Alarmname ist NULL.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Alarmnamen angeben

Konstante:

`mpALARMX_ERR_NAME_NULL`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXAcknowledge
- MpAlarmXSet
- MpAlarmXReset
- MpAlarmXCheckState

1.6.3.51 -1064116223: Kein Name angegeben

Beschreibung:

Angegebener String für Alarmnamen ist leer.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Alarmnamen überprüfen

Konstante:

`mpALARMX_ERR_NAME_EMPTY`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXAcknowledge
- MpAlarmXSet

- MpAlarmXReset
- MpAlarmXCheckState

1.6.3.52 -1064116221: Fehlender Wert an UIConnect

Beschreibung:

An "UIConnect" wurde NULL angehängt.

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Eingang "UIConnect" vergessen

Konstante:

`mpALARMX_ERR_MISSING_UICONNECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmXListUI
- MpAlarmXHistoryUI

1.6.3.53 -1064116220: Alarm kann nicht deaktiviert/quittiert werden

Beschreibung:

Der angegebene Alarm ist unbekannt / nicht aktiv und kann deshalb nicht deaktiviert/quittiert werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Prüfen, ob der Alarm aktiv ist
- Alarmnamen überprüfen
- Konfiguration überprüfen

Konstante:

`mpALARMX_ERR_ALARM_NOT_ACTIVE`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXReset
- MpAlarmXAcknowledge

1.6.3.54 -1064116219: Kein Alarm ausgewählt

Beschreibung:

Es wurde kein Alarm in der Liste ausgewählt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Alarm auswählen

Konstante:

`mpALARMX_ERR_ALARM_NOT_SELECTED`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXListUI

1.6.3.55 -1064116218: Quittieren des Alarms nicht erlaubt

Beschreibung:

Der Alarm kann über disen MpAlarmXListUI nicht quittiert werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Wird der Alarm über mehrere MpAlarmXListUI angezeigt (z.B. ist einmal MpAlarmXListUI direkt an MpAxisBasic angehängt um nur die mapp Alarme dieser Komponente anzuzeigen und einmal an MpAlarmXCore um alle Alarme anzuzeigen), so kann er nur an einer Position quittiert werden und zwar an dem MpAlarmXListUI, welcher über den MpLink mit MpAlarmXCore verbunden ist.

Konstante:

`mpALARMX_ERR_ACK_NOT_ALLOWED`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXListUI

1.6.3.56 -1064116217: Ungültiges File Device

Beschreibung:

Das angegebene File Device am Eingang "DeviceName" ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Am Eingang "DeviceName" ist nichts angehängt
- Das angehängte File Device existiert nicht
- Es konnte keine Verbindung zum gewünschten File Device erstellt werden

Konstante:

`mpALARMX_ERR_INVALID_FILE_DEV`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXHistory

1.6.3.57 -1064116215: MpAlarmXCore bereits vorhanden

Beschreibung:

In dieser Applikation / diesem Applikationsmodul / dieser Gruppe gibt es bereits einen MpAlarmXCore.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Es ist nur ein MpAlarmXCore pro Applikation / Applikationsmodul möglich

Konstante:

`mpALARMX_ERR_CORE_INST_EXISTS`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXCore

1.6.3.58 -1064116214: Fehler bei Speicherinitialisierung

Beschreibung:

Fehler bei der Initialisierung des Speichers für die Alarmhistorie. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfung des angegebenen Speichermediums

Konstante:

`mpALARMX_ERR_MEMORY_INIT`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXHistory

1.6.3.59 -1064116213: Fehler beim Exportieren

Beschreibung:

Die Datei konnte nicht exportiert werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfung des angegebenen Speichermediums

Konstante:

`mpALARMX_ERR_WRITE_EXPORT_FILE`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXHistory

1.6.3.60 -1064116212: Alarmname nicht gefunden

Beschreibung:

Der Alarm mit dem angegebenen Namen wurde nicht gefunden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Kann nach einem "Load" Befehl auftreten, wenn noch kein Mapping für die angegebene Aktion existiert (z.B. für den Alarm "TempHigh" existiert noch kein Eintrag im Alarm Mapping)
- "Name" überprüfen

Konstante:

`mpALARMX_ERR_NAME_NOT_FOUND`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXConfigAlarm

1.6.3.61 -1064116211: Anfrage nicht gefunden

Beschreibung:

Die angegebene Anfrage konnte nicht gefunden werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen ob die Anfrage in der MpAlarmXCore Konfiguration existiert

Konstante:

`mpALARMX_ERR_QUERY_NOT_FOUND`

Diese Funktionsbausteine/ Funktionen können diesen Fehler melden:

- MpAlarmXQuery

1.6.3.62 -1064299694: Name der Kategorie fehlt

Beschreibung:

Für die Kategorie wurde kein Name festgelegt.

Ursache / Abhilfe:

- Angabe eines Namens für die Kategorie in der MpAlarmXCore Konfiguration

1.6.3.63 -1064299695: Name der Kategorie existiert bereits

Beschreibung:

Der Name der Kategorie {2:Name} existiert bereits.

Zusätzliche Informationen

- {2:Name}: Name der Kategorie

Ursache / Abhilfe:

- Angabe eines anderen Namens für die Kategorie in der MpAlarmXCore Konfiguration

1.6.3.64 -1064299696: Wertebereich für Kategorie ungültig

Beschreibung:

Der Wertebereich für die Kategorie {2:Name} ist ungültig

Zusätzliche Informationen

- {2:Name}: Name der Kategorie

Ursache / Abhilfe:

- Überprüfen des angegebenen Wertebereichs für "Severity" oder "Code" in der MpAlarmXCore Konfiguration

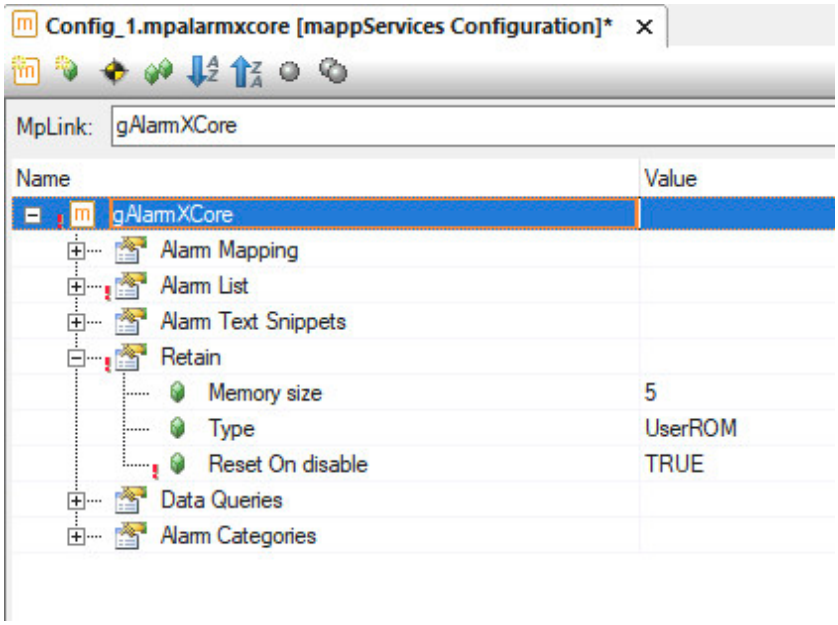
1.6.3.65 -1064299687: Speichern fehlgeschlagen

Beschreibung:

Der definierte Retain-Alarm kann nicht mehr gespeichert werden. Es ist zu wenig Speicher vorhanden.

Ursache / Abhilfe:

- Vergößern des Speichers in der MpAlarmXCore Konfiguration. Dafür wird der "Memory size" Parameter im Abschnitt "Retain" verwendet:



2 mapp Recipe: Rezeptverwaltung

mapp Recipe liefert alle Funktionalitäten, die notwendig sind für eine einfache und schnelle Rezeptverwaltung. Dazu gehören das Lesen und Schreiben, sowie eine Anbindung an Visual Components 4.

Diese Komponente ist kompatibel mit allen anderen mapp Komponenten und kann daher als zentrale Rezeptverwaltung die Parameter der kompletten Maschineninfrastruktur vereinen.

Auf unserem Youtube Kanal sind die Tutorials mapp Recipe Part 1 und mapp Recipe Part 2 welche erklären wie ein Rezeptmanagement implementiert werden kann.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Recipe (GER/ENG) zu finden. (Zugangsdaten)

2.1 Konzept

Allgemein

- Als Applikationist möchte ich, dass meine Rezeptparameter in unterschiedliche Rezepte gespeichert werden können, sowie aus unterschiedlichen Rezepten geladen werden können
- Als Applikationist möchte ich meine Parameter über unterschiedliche Kategorien (PID, Maschinenparameter, ...) zu verschiedenen Rezepten zuordnen können
- Als Applikationist möchte ich eine einfache Möglichkeit meine Prozessvariablen (Strings, Arrays, Strukturen, ...) in Rezepte zu verpacken

Export-Funktion

- Als Endnutzer möchte ich meine Rezepte manipulationssicher auf einem Speichermedium verwalten

Rezepte visualisieren

- Als Endnutzer möchte ich verschiedene Rezepte verwalten bzw. verwenden
- Als Maschinenbediener möchte ich zu Schichtbeginn das entsprechende Rezept laden
- Als Schichtleiter möchte ich Rezepte verändern und Speichern können

2.1.1 Aufzeichnen verschiedener Rezepte

Mithilfe einer oder mehreren MpRecipeXML-Instanzen können verschiedene Rezeptdateien, wie beispielsweise ein Rezept welches PID-Parameter oder Achs-Parameter beinhaltet, aufgzeichnet werden. Dies wird mithilfe des "Category" Eingangsparameters an MpRecipeCsv/MpRecipeXml bzw. MpRecipeRegPar/MpRecipeRegParSync realisiert.

Je nach Anwendungsfall kann dieser Eingangsparameter unterschiedlich verwendet werden.

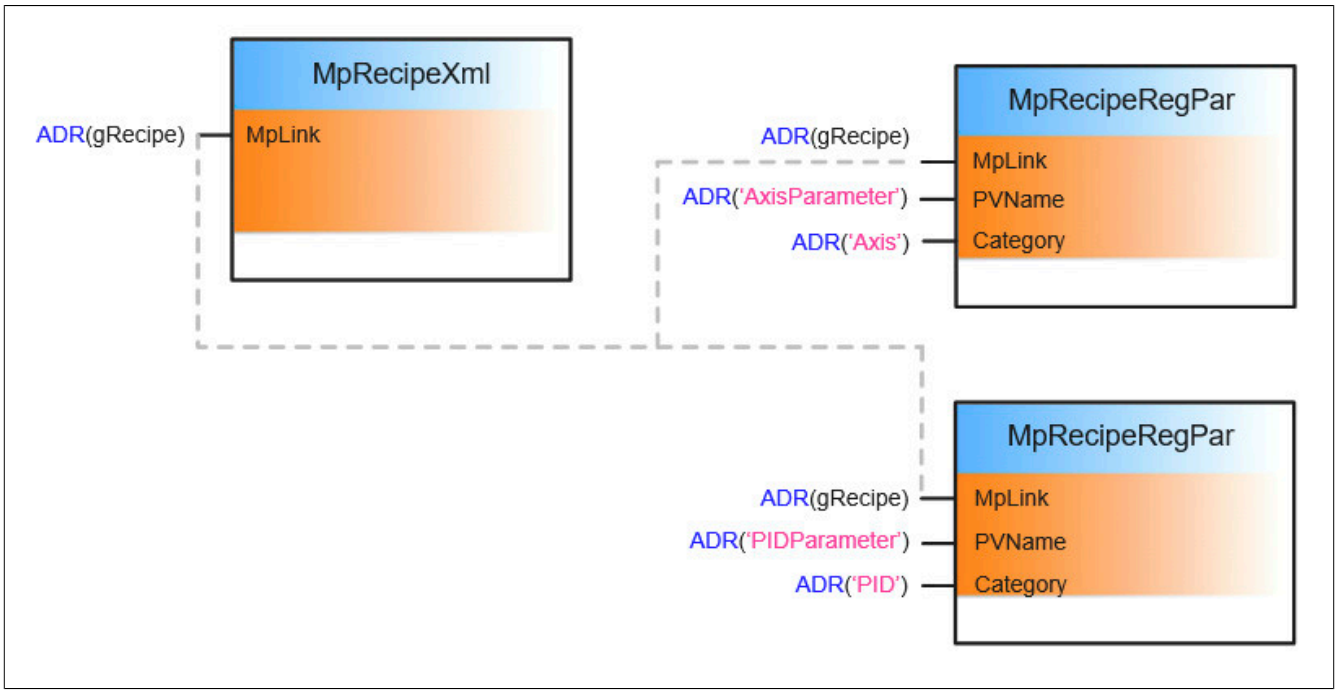
Zu diesem Thema gibt es auch einen Anwendungsfall Maschine mit Produkt- und Maschinenparametern.

Variante 1

Im folgenden Beispiel werden in einer Maschine zwei verschiedene Prozessvariablen mithilfe von MpRecipeRegPar registriert. In der Prozessvariable "AxisParameter" werden Werte wie Geschwindigkeit, Beschleunigung etc. gespeichert. Mithilfe der zweiten Prozessvariable "PIDParameters" werden Parameter gespeichert welche für eine Temperatur-Zone benötigt werden.

Der Funktionsbaustein welcher die Prozessvariable "AxisParameter" registriert, verwendet die Kategorie "Axis", der andere verwendet die Kategorie "PID".

Mithilfe des Funktionsbausteins MpRecipeCsv/MpRecipeXml werden die registrierten Prozessvariablen in einer Rezeptdatei abgespeichert.



Das gespeicherte Rezept kann folgendermaßen aussehen:

MachineParameters.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA>
    <Element Name="AxisParameter" Type="PvParameter">
        <Group ID="AxisParameter">
            <Property ID="Velocity" DataType="REAL" Value="130" />
            <Property ID="Acceleration" DataType="REAL" Value="50" />
            <Property ID="Position" DataType="LREAL" Value="340" />
        </Group>
    </Element>
    <Element Name="PIDParameter" Type="PvParameter">
        <Group ID="PIDParameter">
            <Property ID="ProportionalGain" DataType="REAL" Value="1.2" />
            <Property ID="IntegrationTime" DataType="REAL" Value="2" />
            <Property ID="DerviativeTime" DataType="REAL" Value="2.1" />
        </Group>
    </Element>
</DATA>
```

Je nachdem welches Rezept aktuell geladen werden muss, kann die gewünschte Kategorie "Axis" oder "PID" am Eingangsparameter "Category" an MpRecipeCsv/MpRecipeXml angegeben werden.

Das bedeutet in einer Rezeptdatei können unterschiedliche Kategorien abgespeichert werden. Je nachdem welche Prozessvariable aktuell benötigt wird, kann ein Teil der gesamten Rezept-Stuktur geladen werden. Sollen also die PID-Parameter geladen werden, so wird am Funktionsbaustein MpRecipeCsv/MpRecipeXml die Kategorie "PID" angegeben. Durch "Load = `TRUE`" wird das Rezept für die PID-Parameter geladen.

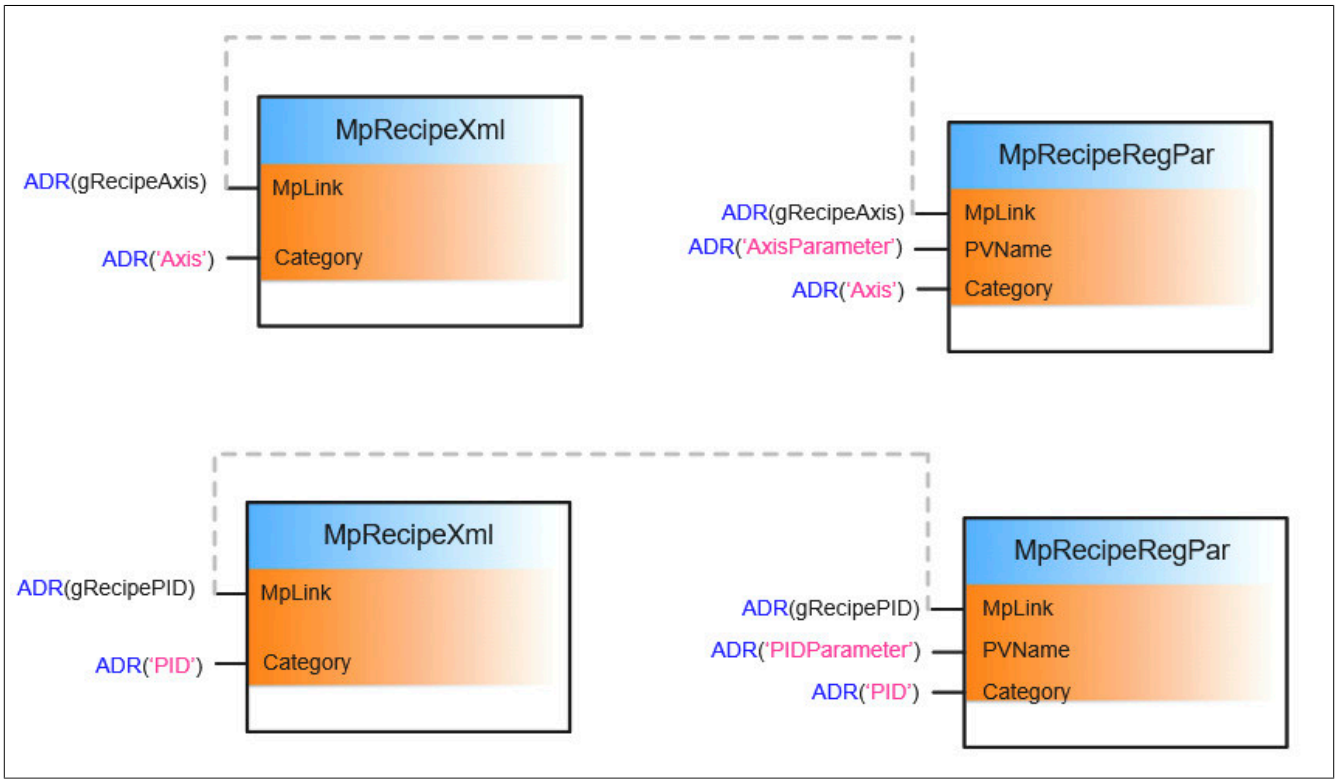
Variante 2

Falls gewünscht können die Prozessvariablen auch in unterschiedliche Rezepte aufgeteilt werden.

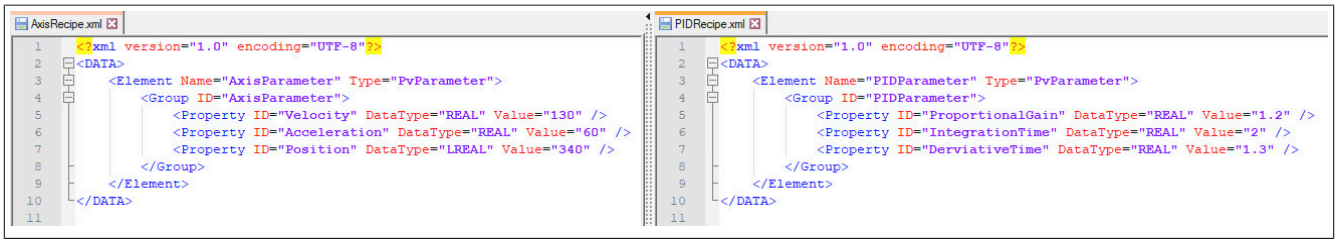
Im folgenden Beispiel existieren zwei verschiedene Prozessvariablen welche jeweils in einem Rezept abgespeichert werden sollen.

In der Prozessvariable "AxisParameter" werden Werte wie Geschwindigkeit, Beschleunigung etc. gespeichert. Mithilfe der zweiten Prozessvariable "PIDParameters" werden Parameter gespeichert welche für eine Temperatur-Zone benötigt werden. Das bedeutet die Achsparameter sollen in einem Rezept gespeichert werden sowie die PID-Parameter in einem anderen Rezept.

Um dies zu realisieren werden jeweils zwei MpRecipeCsv/MpRecipeXml Instanzen verwendet. Eine Instanz ist für die Achs-Parameter verantwortlich, die zweite für die PID-Parameter. Um zwischen den verschiedenen Rezepten unterscheiden zu können, wird der Eingangsparameter "Category" an den jeweiligen Funktionsbausteinen verwendet:



Werden die jeweiligen Rezepte abgespeichert, so werden zwei verschiedene Rezept-Dateien erstellt:



AxisRecipe.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA>
    <Element Name="AxisParameter" Type="PvParameter">
        <Group ID="AxisParameter">
            <Property ID="Velocity" DataType="REAL" Value="130" />
            <Property ID="Acceleration" DataType="REAL" Value="60" />
            <Property ID="Position" DataType="LREAL" Value="340" />
        </Group>
    </Element>
</DATA>
```

PIDRecipe.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA>
    <Element Name="PIDParameter" Type="PvParameter">
        <Group ID="PIDParameter">
            <Property ID="ProportionalGain" DataType="REAL" Value="1.2" />
            <Property ID="IntegrationTime" DataType="REAL" Value="2" />
            <Property ID="DerviativeTime" DataType="REAL" Value="1.3" />
        </Group>
    </Element>
</DATA>
```

2.1.2 Auditing der Rezeptverwaltung

Für diese Komponente können Ereignisse über MpAudit aufgezeichnet werden. Für mehr Information wie ein Eintrag erstellt werden kann, siehe Abschnitt Definieren des Ereignistexts.

Die Identifikationsnummer kann anhand des Kürzels **%act** in den jeweiligen Eintrag eingebunden werden.

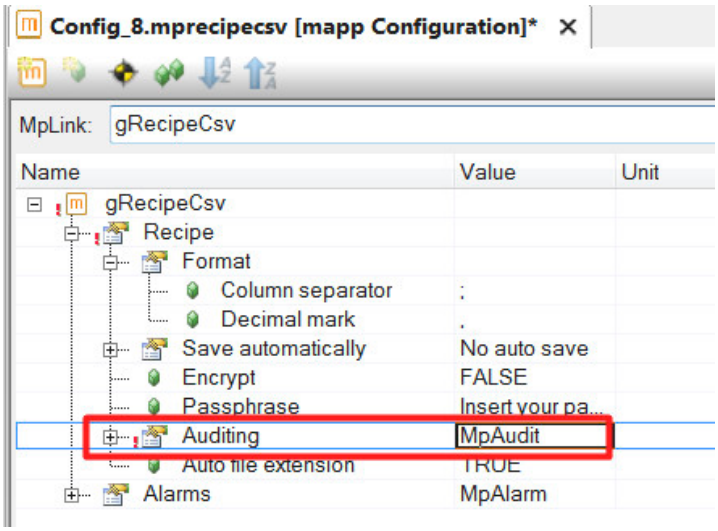
Folgende Ereignisse können für diese mapp Komponenten auftreten:

Rezeptsystem Ereignis

Durch diesen Ereignistyp können Ereignisse in der Rezeptverwaltung, wie zum Beispiel welches Rezept geladen wurde, aufgezeichnet werden.

> **Werden Ereignisse über eine mapp View Visualisierung ausgelöst, können manche Ereignisse nicht einem Benutzer zugeordnet werden. Dies muss, bei der Verwendung des "op" Kürzels beachtet werden. Für weitere Information siehe Abschnitt Ereignisse einem Benutzer zuordnen.**

Um die Ereignisse aufzuzeichnen, muss in der MpRecipeCsv Konfiguration bzw. MpRecipeXml Konfiguration die Option "Auditing" freigeschaltet sein:



Ereignis ID 48	Komponenten MpRecipe Komponenten
Ereignisse	
op	Name des Benutzers welcher beim Auftreten des Ereignisses eingeloggt war
idx	Liefert den Index des aktuell generierten Ereignisses. Der Wert wird bei jedem neuen Ereignis inkrementiert. Das erste generierte Ereignis in der Ereignisliste hat somit den Wert 1, dass fünfzigste Ereignis hat den Index 50, das hundertste den Index 100 und so weiter.
ev	Ereignis ID
file	Datei welche verwendet wurde (z.B. Rezept, welches geladen wurde)
dev	Speichermedium welches verwendet wurde (z.B. USB Stick, von welchem ein Rezept geladen wurde)
act	Aktion, welche in einer mapp Komponente ausgeführt wurde. Die ID für die Aktion wird zurükgegeben (z.B. 1 für "Laden des Rezepts wurde beendet"). Für mehr Infomation siehe folgende Tabelle:

Ereignis	Identifikationsnummer (wird über %act zurückgegeben)	Zusatzinformation
Laden eines Rezepts wurde gestartet	0	• {&op}/[%op] - Benutzer • {&file}/[%file] - Rezept • {&dev}/[%dev] - Speichermedium
Laden eines Rezepts wurde beendet	1	• {&op}/[%op] - Benutzer • {&file}/[%file] - Rezept • {&dev}/[%dev] - Speichermedium
Speichern eines Rezepts	2	• {&op}/[%op] - Benutzer • {&file}/[%file] - Rezept • {&dev}/[%dev] - Speichermedium
Umbenennen eines Rezepts	3	• {&op}/[%op] - Benutzer • {&old}/[%old] - Alter Name des Rezepts • {&new}/[%new] - Neuer Name des Rezepts • {&dev}/[%dev] - Speichermedium
Löschen eines Rezepts	4	• {&op}/[%op] - Benutzer • {&file}/[%file] - Rezept • {&dev}/[%dev] - Speichermedium

Tabelle 2: MpRecipe Ereignis

Wie der Ereignistext angegeben werden kann ist in Abschnitt Verwendung des neuen Formats erklärt.

Beispiel für neues Format:

- **'Rezept {&file} des Speichermediums {&dev} wurde gespeichert'** ausgewertet: 'Rezept Cappuccino des Speichermediums USB1 wurde gespeichert.'

Beispiel für altes Format:

- **'Rezept [%file] des Speichermediums [%dev] wurde gespeichert'** ausgewertet: 'Rezept Cappuccino des Speichermediums USB1 wurde gespeichert.'

2.1.3 MpRecipe in einer modularen Applikation

In diesem Abschnitt wird erklärt was man bei der Implementierung von MpRecipe in modularen Applikationen beachten muss. Allgemeine Infos zu mapp in Applikationsmodulen / einer Maschinengruppe sind Bilden von modularen Applikationen

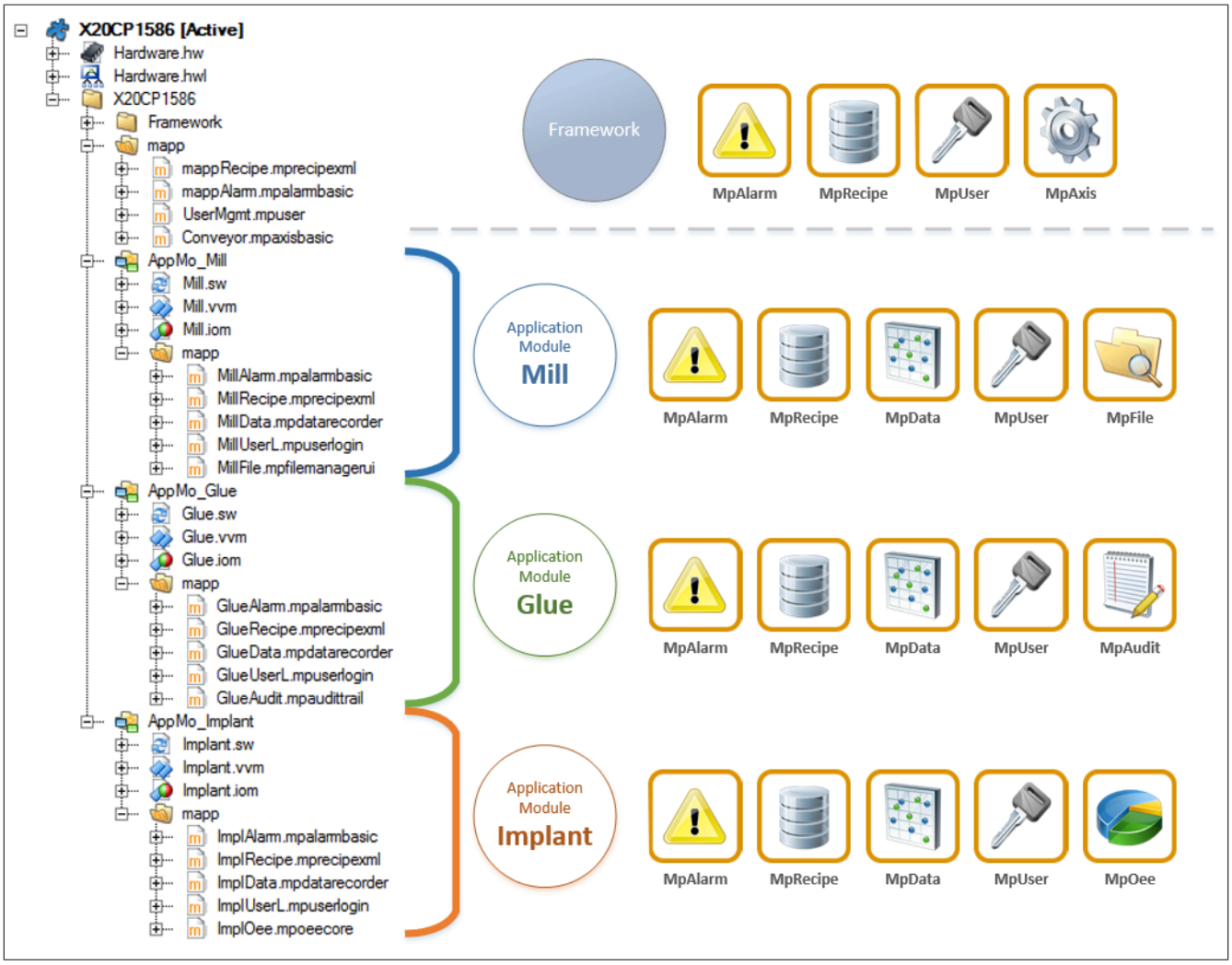
Registrieren von PVs in einem Applikationsmodul

Befindet sich eine Rezeptverwaltung in einem Applikationsmodul, muss bei der Angabe der Prozess Variable (PV) der Komponente MpRecipeRegPar darauf geachtet werden, den Applikationsmodulnamen vor den Variablennamen anzugeben.

- Angabe einer globalen Prozess Variable: Applikationsmodulname::ProzessVariable
- Angabe einer lokalen Prozess Variable: Applikationsmodulname::Taskname:ProzessVariable

Verhalten in einer modularen Applikation

Hier wird erklärt wie sich MpRecipe verhält, wenn es in Kombination mit Applikationsmodulen bzw. in einer Maschinengruppe. Folgendes Bild zeigt eine mögliche Applikation, welche MpRecipe in Applikationsmodulen verwendet:

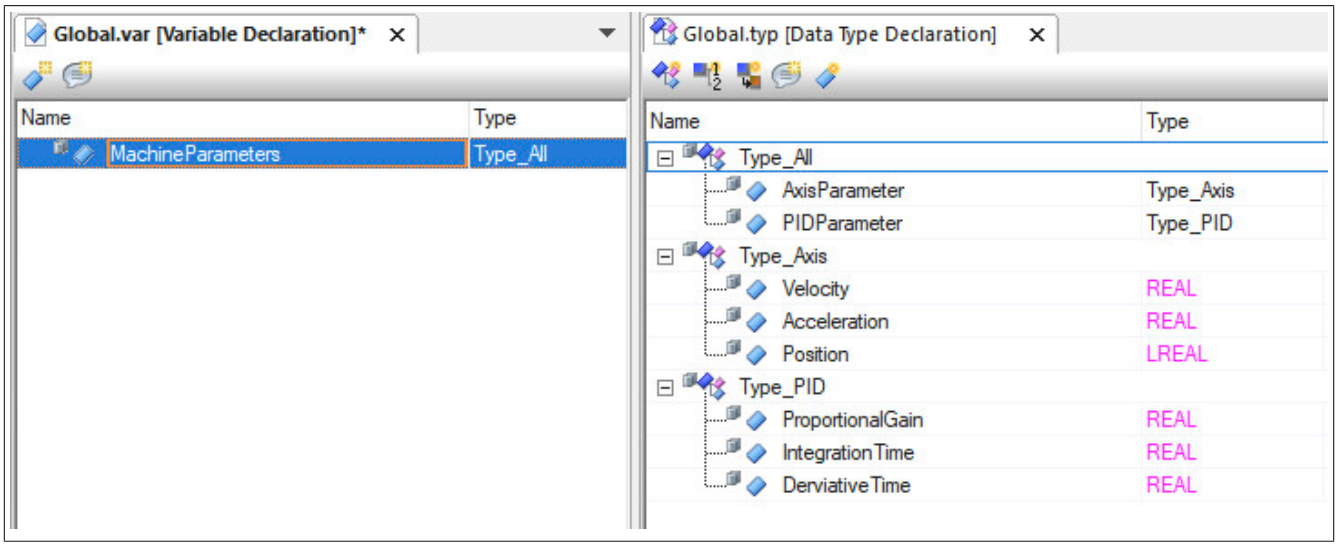


Abhängig von der Position, in wecher das Rezepthandling verwendet wird, ändern sich die Parameter, welche im Rezept abgespeichert werden. Das Rezepthandling nimmt nur nur Parameter ins Rezept, welche in demselben Applikationsmodul bzw. in derselben Maschinengruppe oder darunter vorkommen.

Für obiges Beispiel bedeutet das, dass das Rezepthandling im Modul **Mill** und im Modul **Glue** komplett unabhängig voneinander agieren. Jeder Parameter der im Applikationsmodul **Mill** über MpRecipeRegPar oder MpRecipeRegParSync registriert wird, kommt in ein Rezept das nur Parameter aus diesem Applikationsmodul enthält. Davon unabhängig gibt es ein weiteres Rezept, in dem alle Parameter gesammelt werden die im Applikationsmodul **Glue** registriert werden. Das Rezepthandling im Framework Projekt sammelt die Parameter, welche im Framework Projekt registriert werden, aber auch alle Parameter in den Applikationsmodulen. Arbeitet man mit Maschinengruppen (mapp Hierarchie), dann sammelt eine Maschinengruppe auch alle Parameter, der darunter liegenden Gruppen.

2.1.4 Aufbau der Rezept-Datei

Der Inhalt einer Rezept-Datei wird über die von MpRecipeRegPar bzw. MpRecipeRegParSync registrierte Prozessvariable bestimmt. Abhängig davon, ob MpRecipeCsv oder MpRecipeXml für die Rezept-Erstellung verwendet wurde, unterscheidet sich der Aufbau der Rezept-Datei. Im nächsten Abschnitt wird erklärt wie ein XML-Rezept bzw. ein CSV-Rezept aufgebaut ist, dafür wird die Prozessvariable "MachineParameters" verwendet:



Angabe der Versionsinformation

Über MpRecipeCsv oder MpRecipeXml kann mithilfe des Eingangsparameters "Header" eine Versionsinformation für das Rezept definiert werden. Dafür wird eine Prozessvariable vom Datentypen MpRecipeXmlHeaderType erstellt. Über diese Struktur kann ein Name, eine Beschreibung, die Version sowie das Datum und die Uhrzeit definiert werden, in der das Rezept erstellt wurde. Die Angabe des Headers ist optional. Bei der Angabe eines Headers wird die definierte Information im Rezept zusammen mit den Rezeptparametern abgespeichert.

XML-Rezept

Die XML-Struktur für ein Rezept und die dazugehörige registrierte Prozessvariable kann beispielsweise folgendermaßen aussehen:

AllRecipe.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <DATA>
3      <Element Name="MachineParameters" Type="PvParameter">
4          <Group ID="MachineParameters">
5              <Group ID="AxisParameter">
6                  <Property ID="Velocity" DataType="REAL" Value="0" />
7                  <Property ID="Acceleration" DataType="REAL" Value="0" />
8                  <Property ID="Position" DataType="LREAL" Value="0" />
9              </Group>
10             <Group ID="PIDParameter">
11                 <Property ID="ProportionalGain" DataType="REAL" Value="0" />
12                 <Property ID="IntegrationTime" DataType="REAL" Value="0" />
13                 <Property ID="DerviativeTime" DataType="REAL" Value="0" />
14             </Group>
15         </Group>
16     </Element>
17 </DATA>
```

Angabe einer Versionsinformation

Bei der Angabe einer Versionsinformation ("Header") scheinen die zuvor definierten Parameter im XML-Rezept auf:

```
MachineParameters.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <DATA>
3      <Element Name="MpRecipe.Header" Type="HeaderData">
4          <Property ID="Name" DataType="STRING" Value="MachineParameters" />
5          <Property ID="Description" DataType="STRING" Value="Definition of the machine parameters" />
6          <Property ID="Version" DataType="STRING" Value="1.0.0" />
7          <Property ID="Created" DataType="UDINT" Value="1614006620" />
8          <Property ID="CreatedReadable" DataType="STRING" Value="2021-02-22 15:10:20" />
9      </Element>
10     <Element Name="MachineParameters" Type="PvParameter">
11         <Group ID="MachineParameters">
12             <Group ID="AxisParameter">
13                 <Property ID="Velocity" DataType="REAL" Value="120" />
```

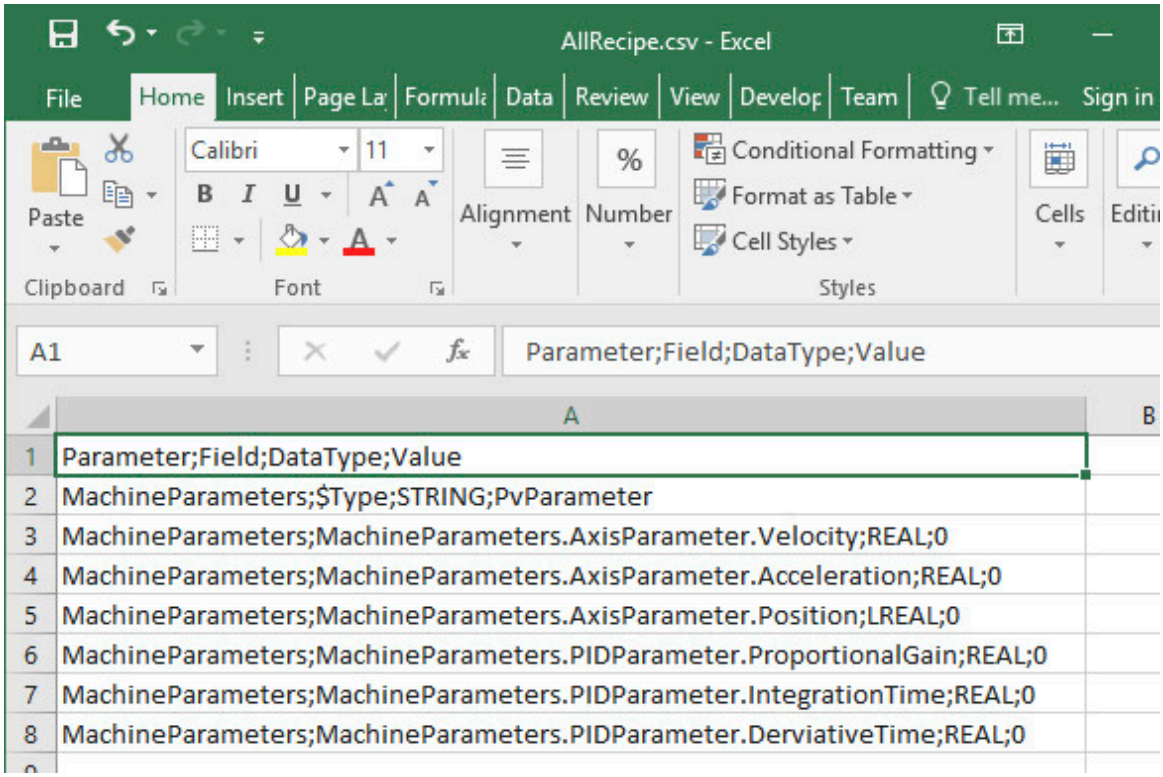
Die Angabe der Zeit, Version, etc. muss dabei selbst definiert werden.

XML Schema:

Hier angehängt die XML Schema Definition für alle Rezepte mit Dateiformat .xml: MpRecipeXml

CSV-Rezept

Die CSV-Struktur für ein Rezept und die dazugehörige registrierte Prozessvariable kann beispielsweise folgendermaßen aussehen:



In der ersten und zweiten Zeile sind die sogenannte Header-Informationen des Rezepts, welches für das Laden des Rezepts benötigt werden. Danach folgen die Rezept-Informationen.

Es gilt zu beachten, dass sobald die Header-Information des Rezepts nicht vorhanden ist, die Rezept-Datei nicht korrekt geladen werden kann. Aus diesem Grund darf die Header-Information nicht gelöscht werden!

Angabe einer Versionsinformation

Bei der Angabe einer Versionsinformation ("Header") scheinen die zuvor definierten Parameter im CSV-Rezept auf:

	A
1	Parameter;Field;DataType;Value
2	MpRecipe.Header;$Type;STRING;HeaderData
3	MpRecipe.Header;Name;STRING;MachineParameters
4	MpRecipe.Header;Description;STRING;Definition of the machine parameters
5	MpRecipe.Header;Version;STRING;1.0.0
6	MpRecipe.Header;Created;UDINT;1614006615
7	MpRecipe.Header;CreatedReadable;STRING;2021-02-22 15:10:15
8	MachineParameters;$Type;STRING;PvParameter
9	MachineParameters;MachineParameters.AxisParameter.Velocity;REAL;120
10	MachineParameters;MachineParameters.AxisParameter.Acceleration;REAL;20
11	MachineParameters;MachineParameters.AxisParameter.Position;LREAL;340
12	MachineParameters;MachineParameters.PIDParameter.ProportionalGain;REAL;2
13	MachineParameters;MachineParameters.PIDParameter.IntegrationTime;REAL;1
14	MachineParameters;MachineParameters.PIDParameter.DerviativeTime;REAL;1
15	

Die Angabe der Zeit, Version, etc. muss dabei selbst definiert werden.

2.1.5 Rahmenbedingung

In diesem Abschnitt werden die Rahmenbedingungen von mapp Recipe aufgelistet

Allgemeines

- Der Rezeptname darf folgende Zeichen nicht enthalten: " * / : < > ? \ |

2.2 Getting Started

2.2.1 Erstellen einer Rezeptverwaltung

In diesem Abschnitt wird Schritt für Schritt beschrieben wie man mithilfe von mapp Recipe ein Rezept erstellt und bearbeitet.

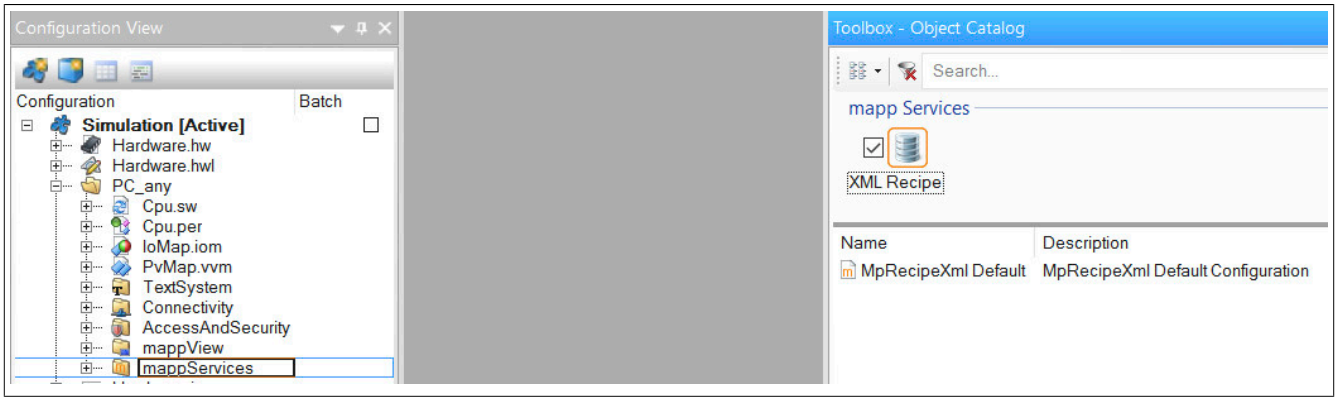
2.2.1.1 Projekt erstellen

Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

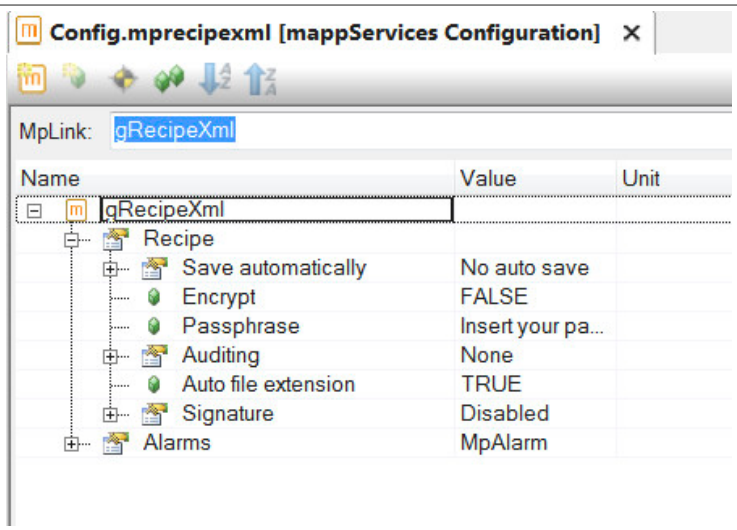
2.2.1.2 mapp Komponente einfügen

Konfiguration MpRecipeXml einfügen

Um eine Rezeptverwaltung zu erstellen, wird die MpRecipeXml Konfiguration eingefügt.



Hier befindet sich der mapp Link, welcher die mapp Komponente in der "world of mapp" eindeutig identifiziert. Der mapp Link stellt die Verbindung zwischen Programmierung und Konfiguration her.

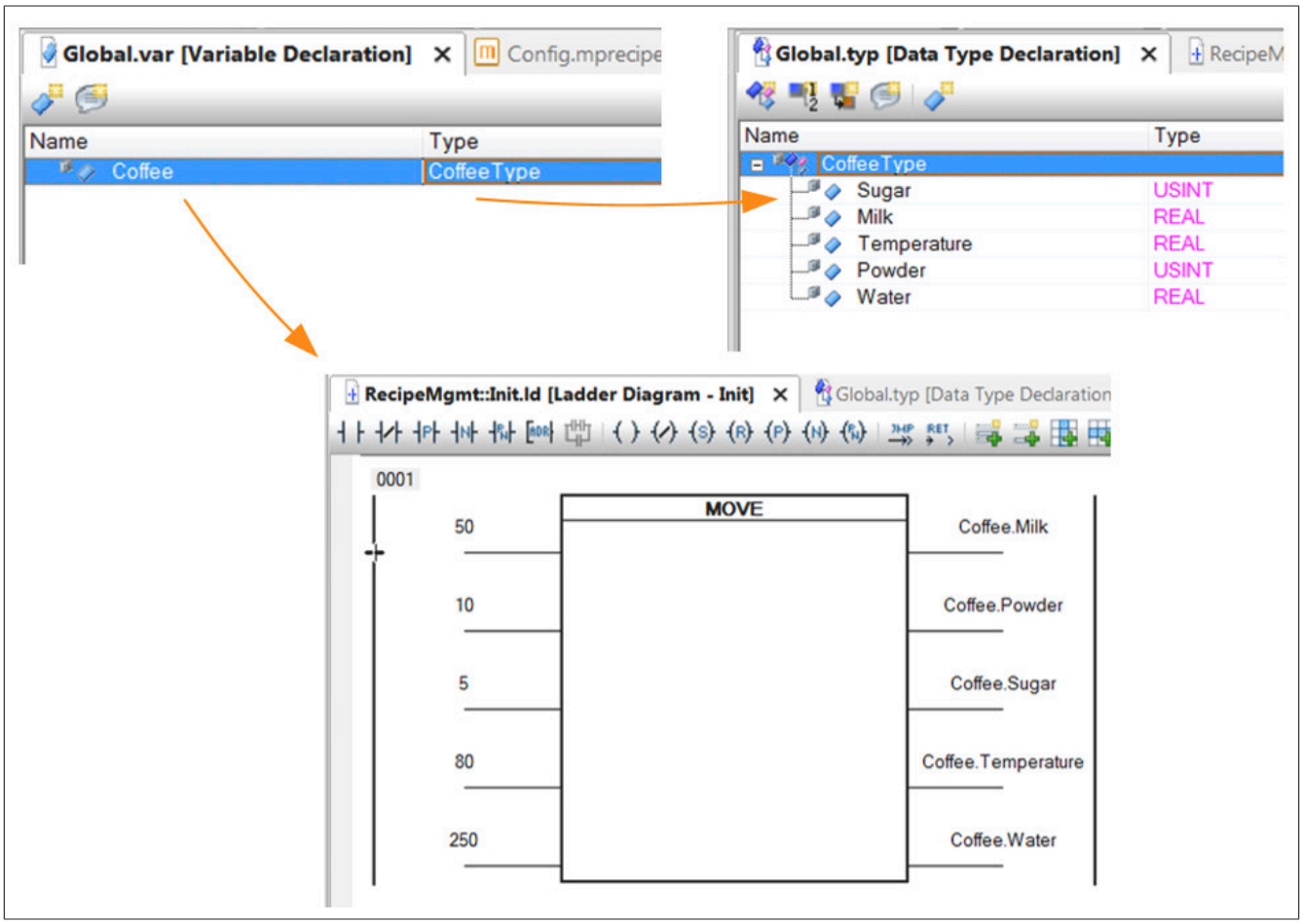


2.2.1.3 Programm hinzufügen

Der nächste Schritt ist das Einfügen eines Kontaktplan-Programms. Für mehr Information siehe hier.

2.2.1.4 Daten fürs Rezept erzeugen

Es wird die Stuktur CoffeeType erstellt, die diese Parameter beinhaltet. Die Variable Coffee soll in einem Rezept abgespeichert werden. Die Prozess Variable muss in der Software verwendet werden, damit Speicher dafür bereitgestellt wird, darum wird die Variable im INIT-Teil des Programms aufgerufen.



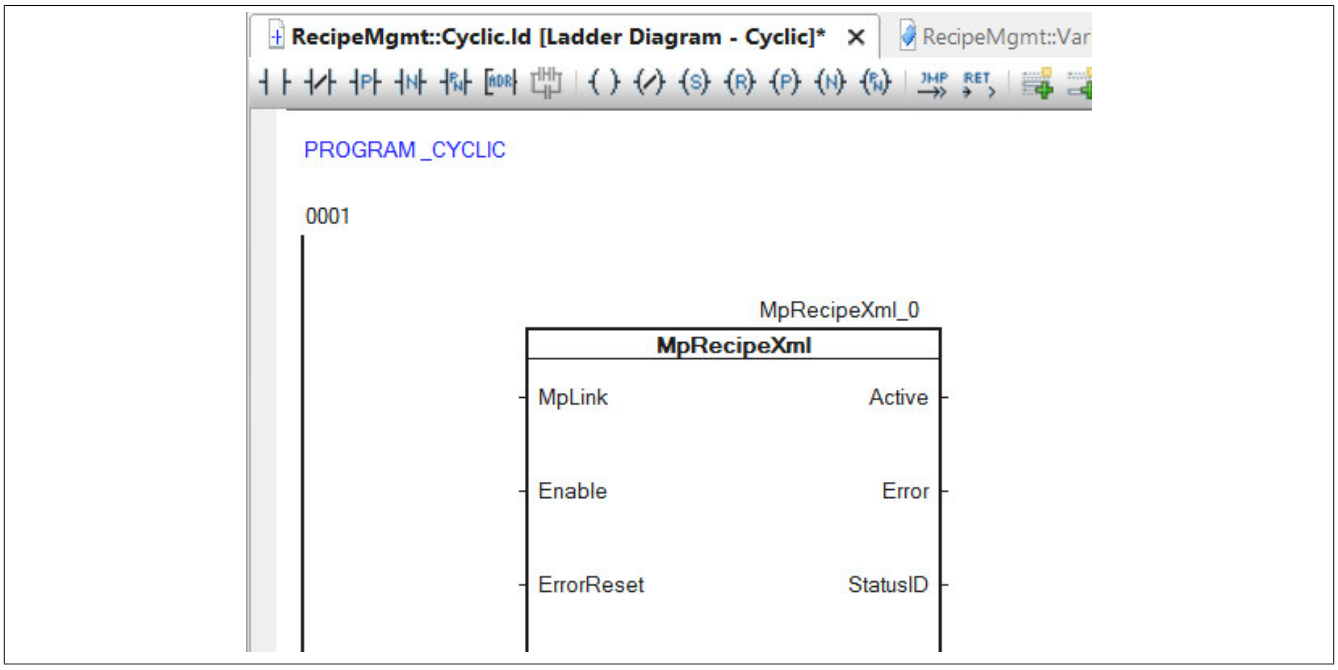
2.2.1.5 Funktionsbausteine einfügen und parametrieren

In diesem Abschnitt wird erklärt, welche Funktionsbausteine eingefügt und parametriert werden.

2.2.1.5.1 MpRecipeXml einfügen

Funktionsbaustein einfügen

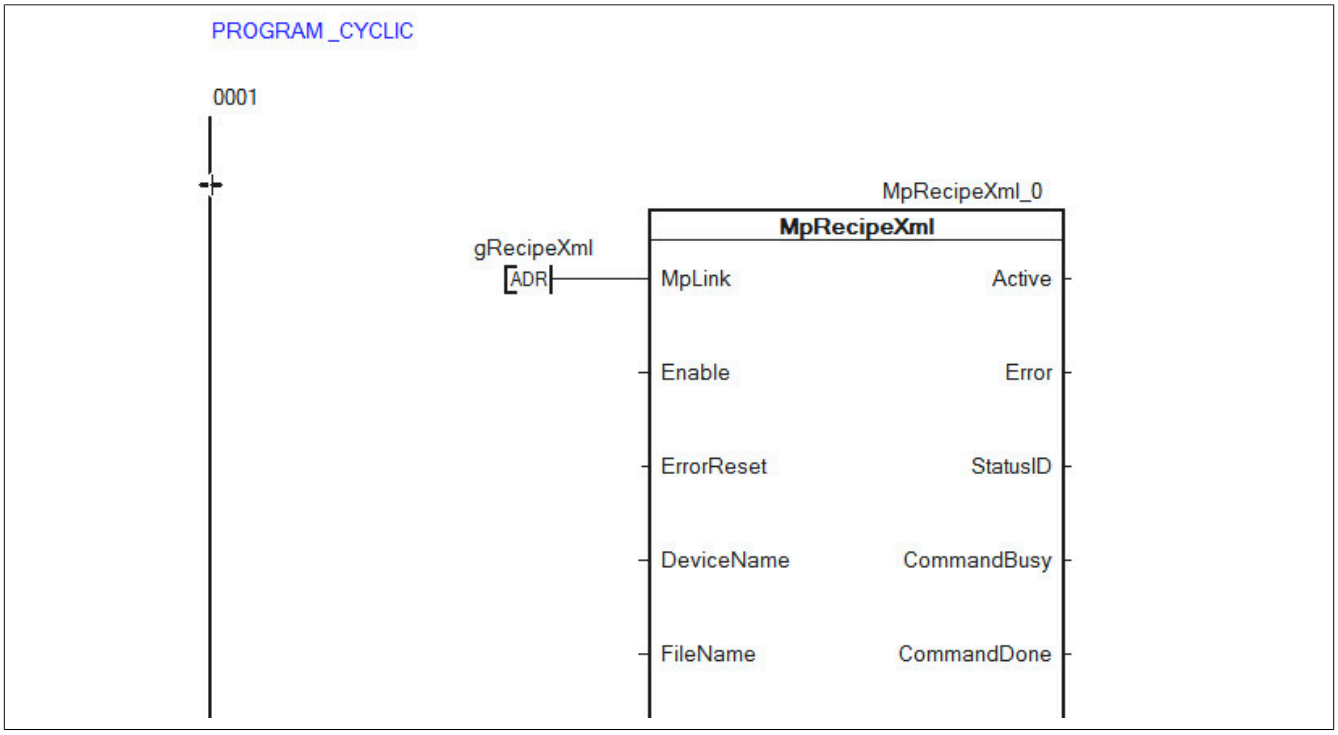
Es wird der Funktionsbaustein MpRecipeXml eingefügt. Mithilfe dieses Funktionsbausteins können Rezepte geladen und gespeichert werden.



2.2.1.5.2 Parametrieren von MpRecipeXml

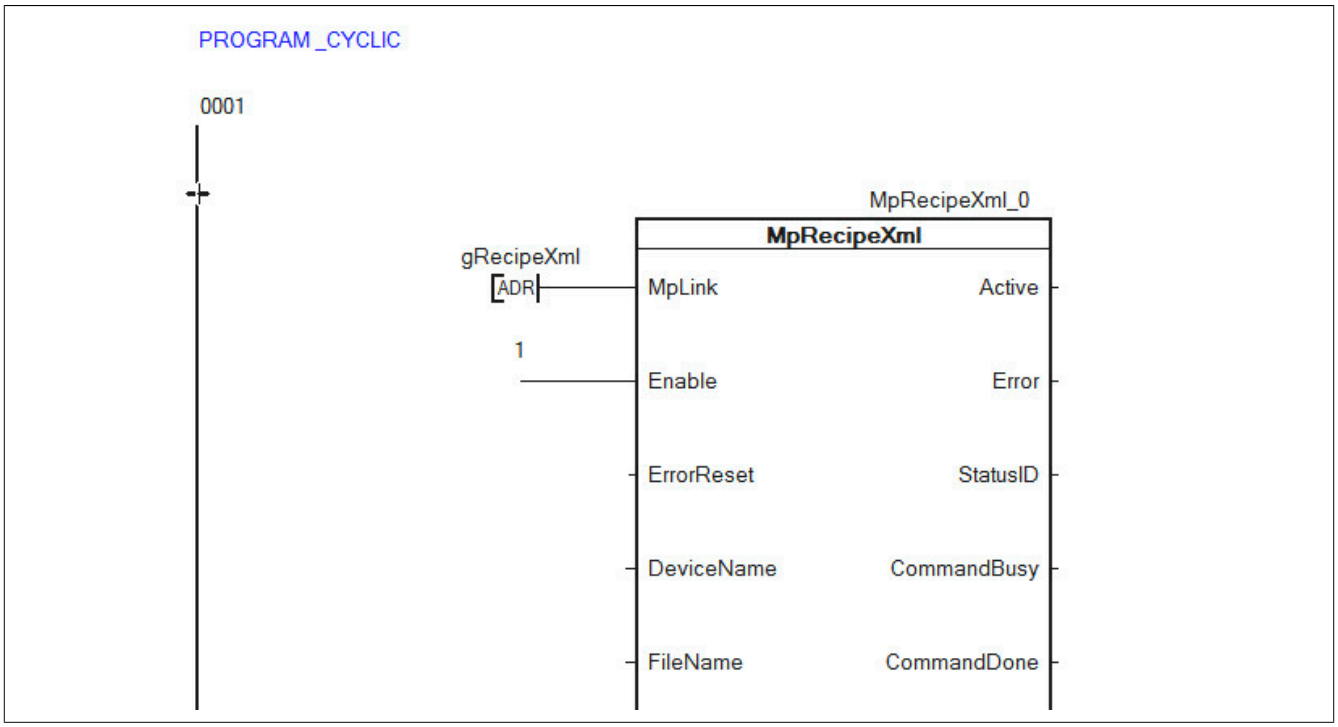
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gRecipeXml erstellt haben.



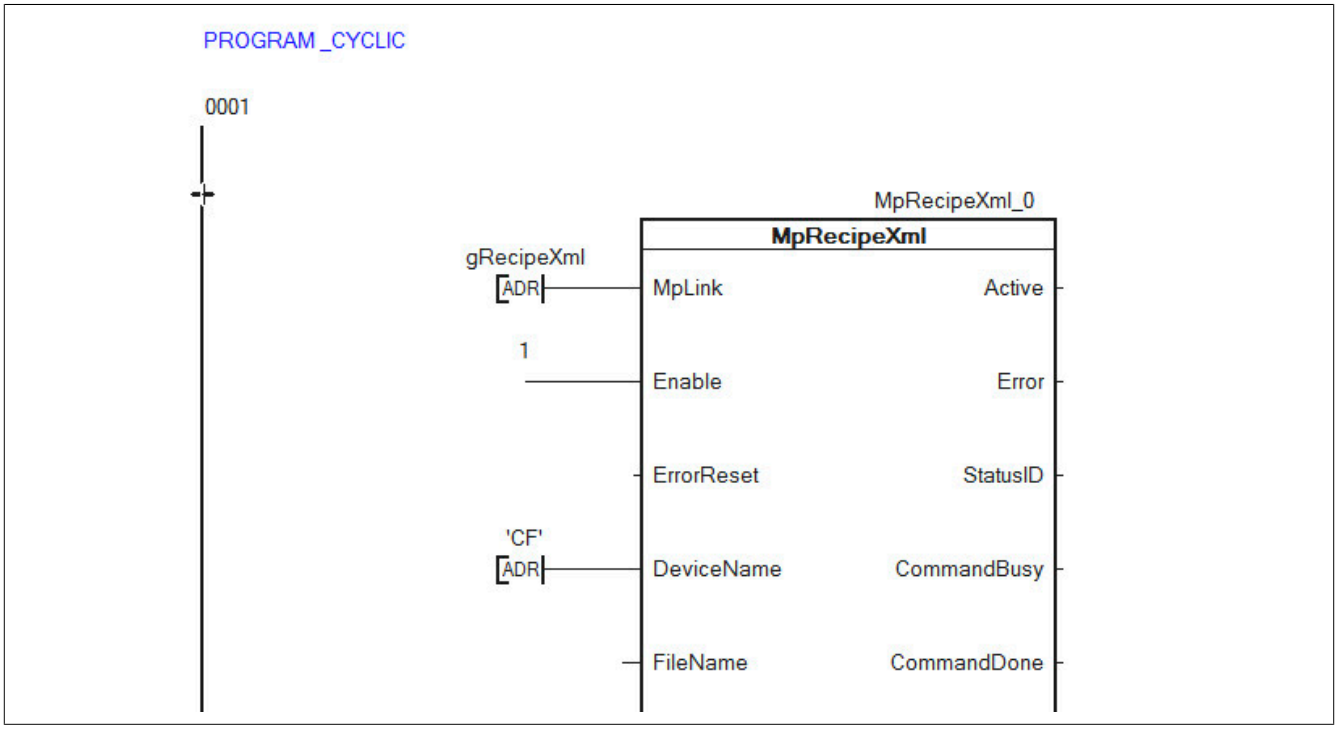
Aktivieren des Funktionsbausteins

Der Funktionsbaustein wird durch den Eingang "Enable" aktiviert.



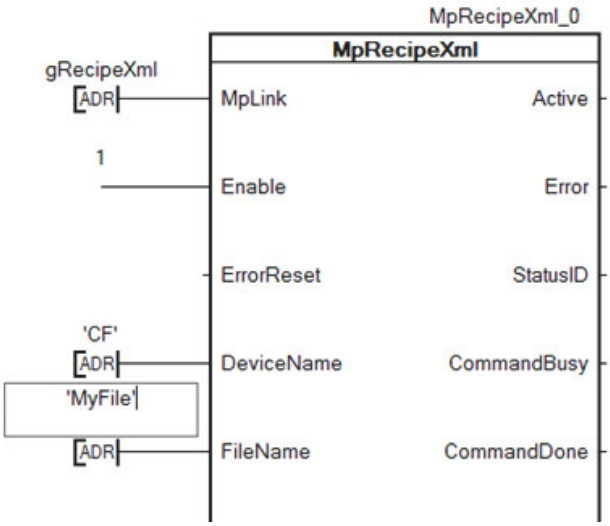
Speichermedium bestimmen

Das Speichermedium für die Rezeptverwaltung wird über den Eingang "DeviceName" bestimmt. Es wird als Adresse übergeben.



Rezept benennen

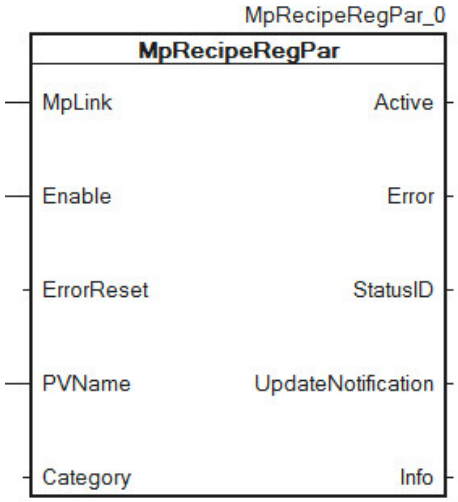
Über den Eingang "FileName" wird der gewünschte Name des Rezeptes eingegeben. Der Dateiname wird als Adresse übergeben.



2.2.1.5.3 MpRecipeRegPar einfügen

Funktionsbaustein einfügen

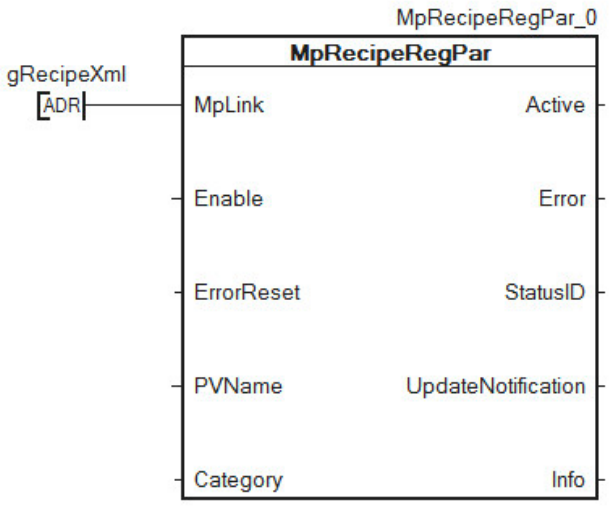
Es wird der Funktionsbaustein MpRecipeRegPar eingefügt. Mithilfe dieses Funktionsbausteins kann eine Prozessvariable für die Rezeptverwaltung registriert werden.



2.2.1.5.4 Parametrieren von MpRecipeRegPar

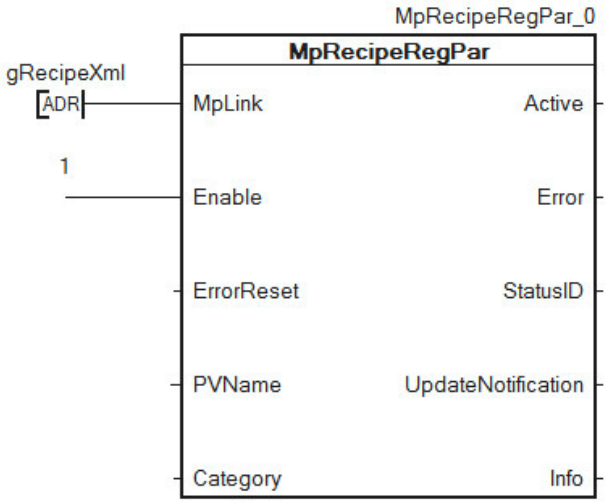
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gRecipeXml erstellt haben.



Aktivieren des Funktionsbausteins

Der Funktionsbaustein wird durch den Eingang "Enable" aktiviert.



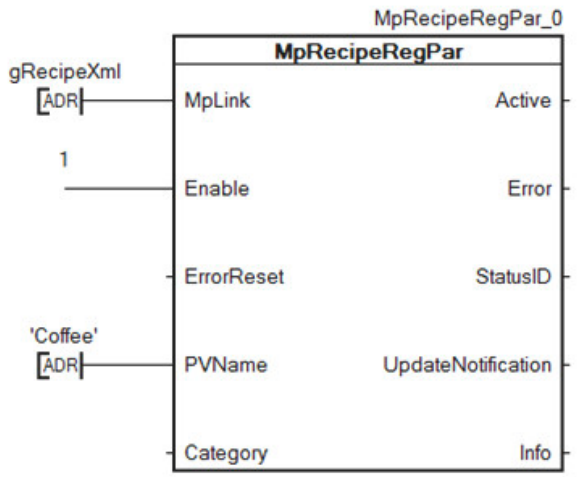
Variable registrieren

Über den Eingang "PVName" wird die erstellte Variable im Rezept registriert. Die PV wird über den Namen übergeben. Dieser setzt sich aus Programm und PV Name zusammen.

ProgrammName:PVName

Da die zu registrierende Variable eine globale Variable ist, muss nur der PV Name angegeben werden.

Dieser String wird als Adresse an MpRecipeRegPar übergeben.



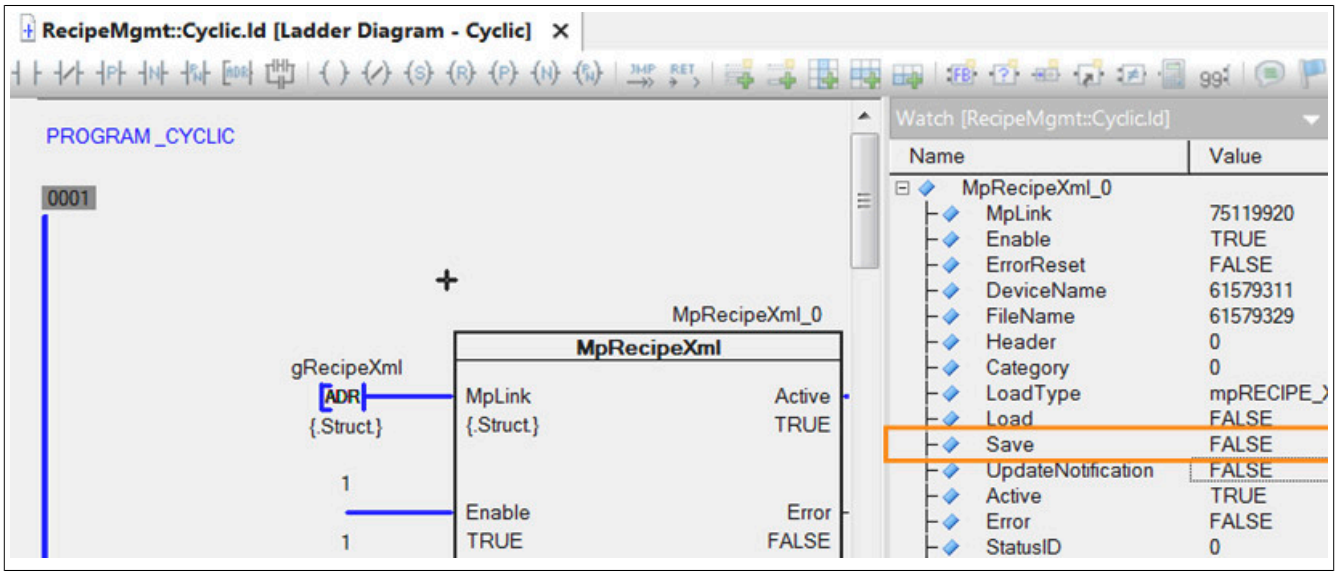
2.2.1.6 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

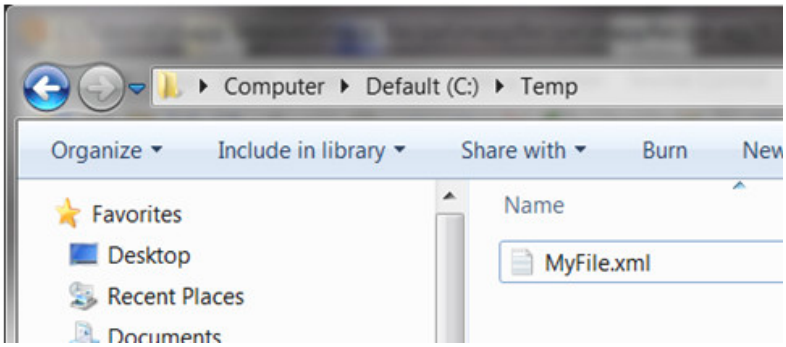
2.2.1.7 Testen des Programms

Nach einem Download der Änderungen kann das Programm getestet werden. Der Monitor-Modus wird aktiviert.

Um ein Rezept zu speichern, wird der Befehl "Save = `TRUE`" an MpRecipeXml gestartet.



Das Kaffeerezept wird am angegeben Speichermedium abgelegt.



Die Datei kann mit einem beliebigen Bearbeitungsprogramm geöffnet und editiert werden.

MyFile.xml - Notepad

File Edit Format View Help

```
<?xml version="1.0"?>
<DATA>
<Element Name="Coffee" Type="PvParameter">
        <Group ID="Coffee">
                <Property ID="Sugar" DataType="USINT" Value="5" />
                <Property ID="Milk" DataType="REAL" Value="50" />
                <Property ID="Temperature" DataType="REAL" Value="80" />
                <Property ID="Powder" DataType="USINT" Value="10" />
                <Property ID="Water" DataType="REAL" Value="250" />
        </Group>
</Element></DATA>
```

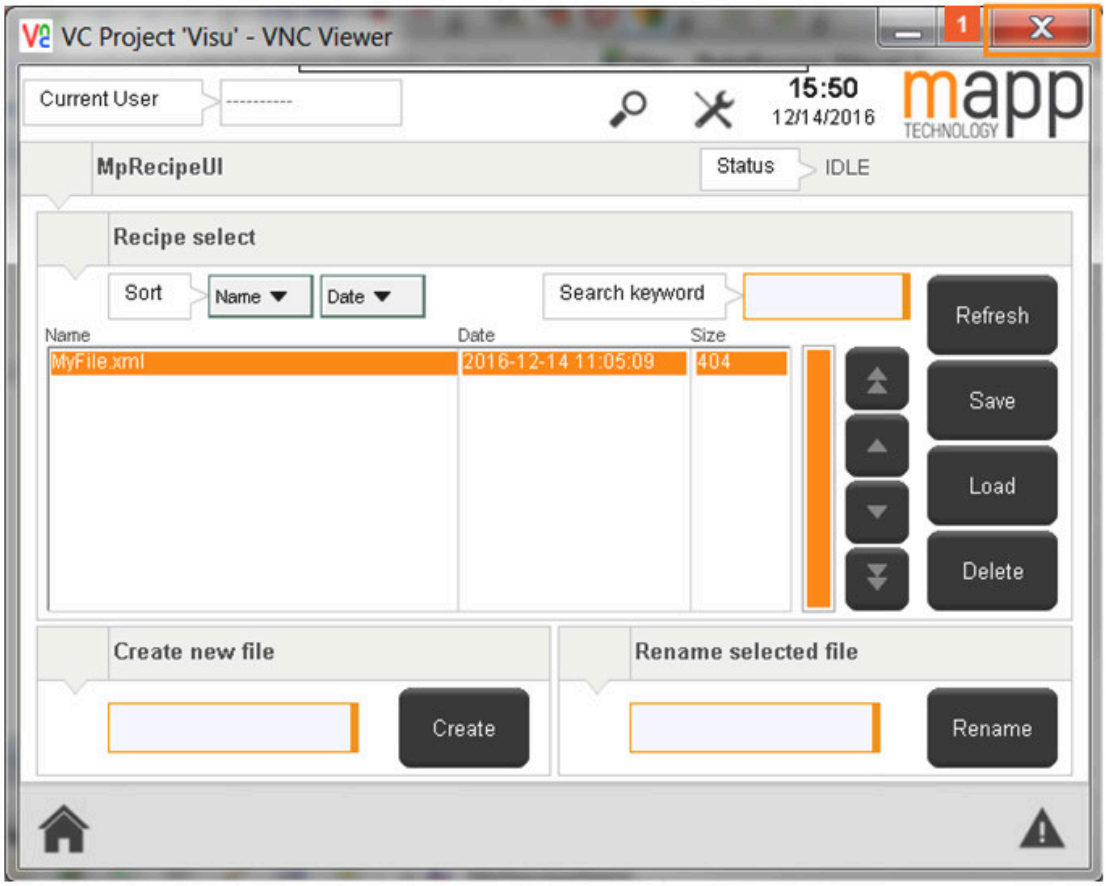
Die bearbeitete Datei kann anschließend mithilfe von "Load = `TRUE`" an MpRecipeXml wieder geladen werden.

Wie die Rezeptverwaltung auf einer VC4-Visualsierung verwendet werden kann, ist im Abschnitt Rezeptverwaltung mit VC4-Visualisierung erklärt.

2.2.2 Rezeptverwaltung mit VC4-Visualisierung

In diesem Abschnitt wird Schritt für Schritt beschrieben wie man mithilfe von MpRecipeUI eine Rezeptverwaltung auf einer VC4-Visualisierung angzeigen lassen kann.

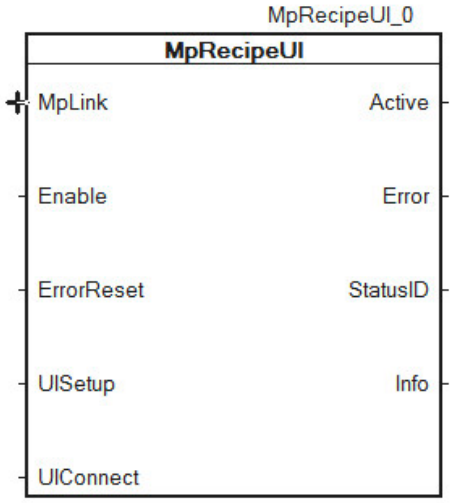
Wie eine Rezeptverwaltung erstellt werden kann, ist im Abschnitt Erstellen einer Rezeptverwaltung erklärt.



2.2.2.1 MpRecipeUI im Programm einfügen

Der Funktionsbaustein kann in ein neues Programm oder in ein bereits bestehendes eingefügt werden.

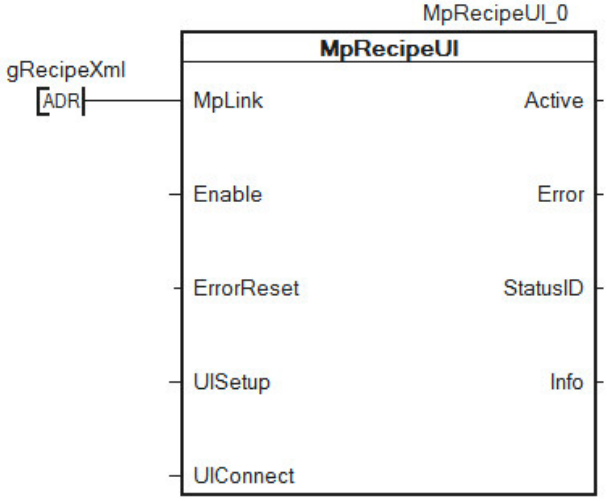
Es wird der Funktionsbaustein MpRecipeUI eingefügt. Mithilfe dieses Funktionsbausteins kann eine Verbindung zu einer VC4-Visualisierung erstellt werden.



2.2.2.2 Parametrieren von MpRecipeUI

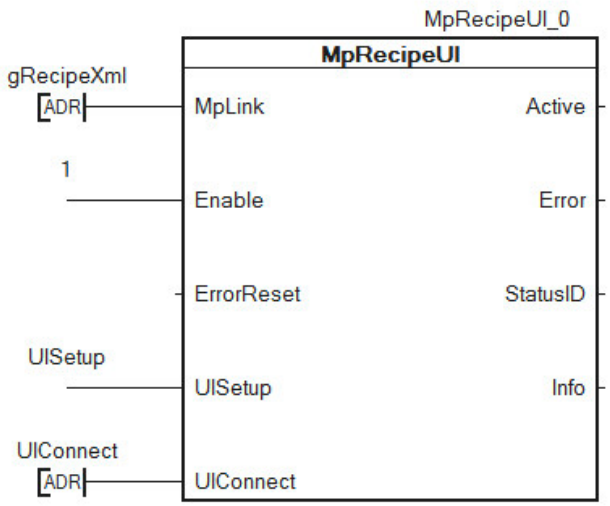
Verbinden des MpLinks

Es wird derselbe MpLink wie für MpRecipeXml verwendet. Er wird über die Adresse übergeben.



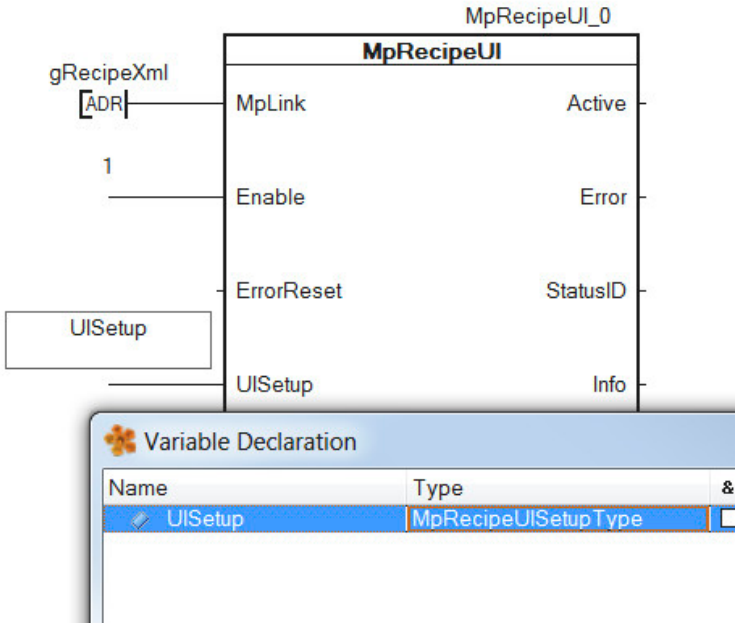
Aktivieren des Funktionsbausteins

Durch Setzen des "Enable" Eingangs wird der Funktionsbaustein aktiviert. Das wird über den Ausgang "Active" angezeigt.



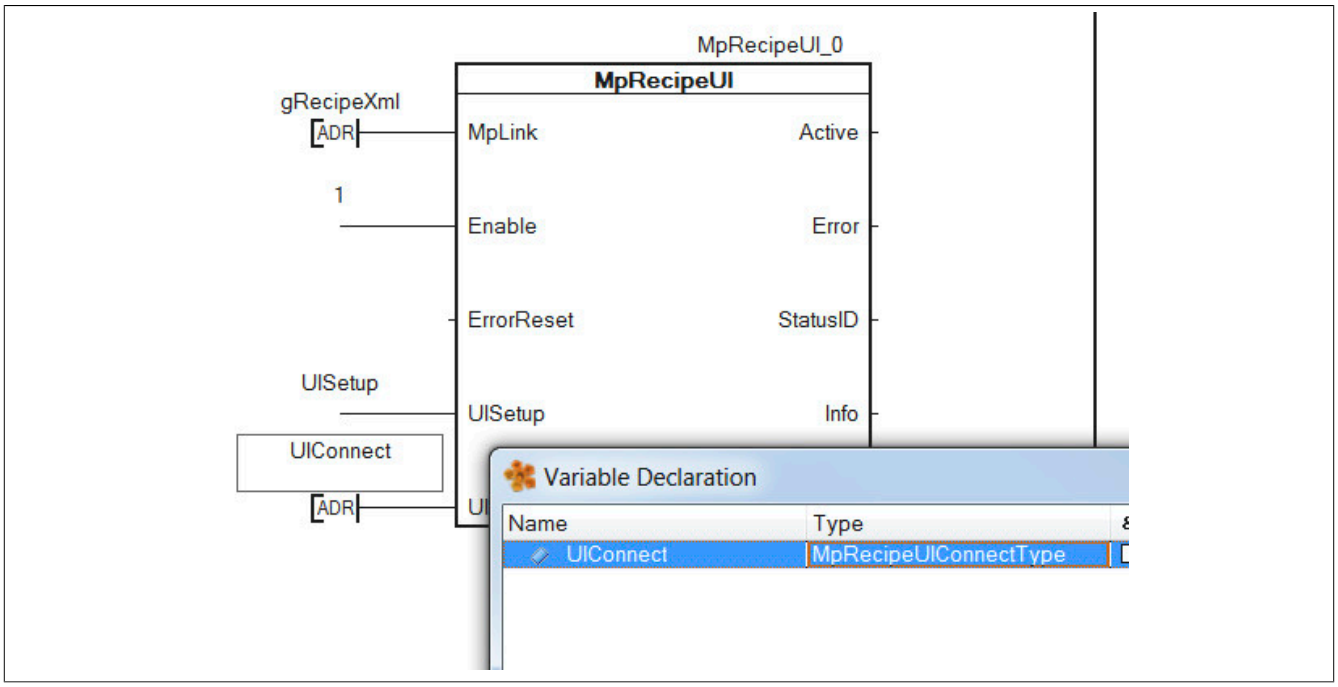
UISetup anhängen

Danach wird die Struktur UISetup (MpRecipeUISetupType) an den Funktionsbaustein angeschlossen. Diese ermöglicht es zu konfigurieren, wieviele Rezepte pro Seite angezeigt werden sollen.



UIConnect anhängen

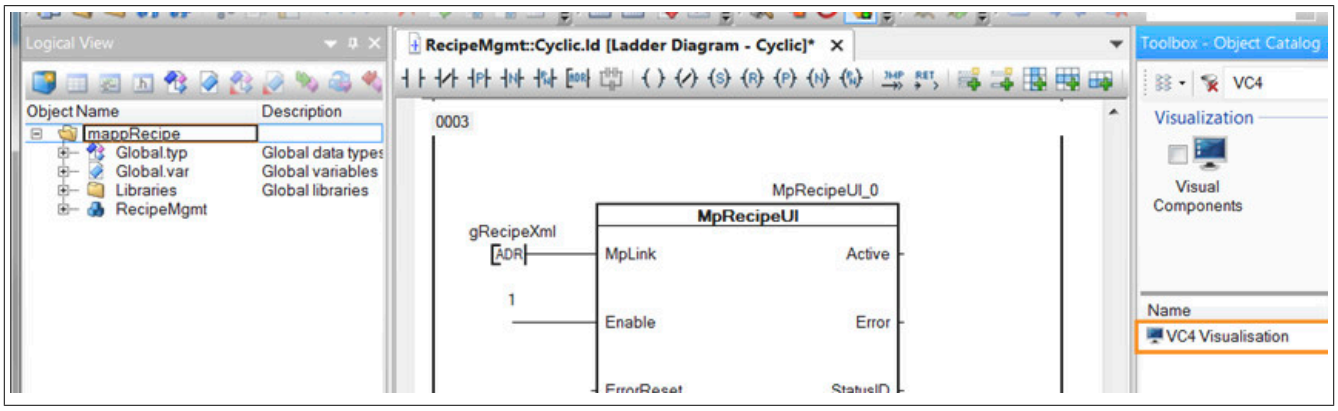
Die UIConnect Struktur (MpRecipeUIConnectType), wird über die Adresse mit MpRecipeUI verbunden. Der Inhalt der Struktur wird später mit den Datenpunkten der VC4 Visualisierung verbunden.



2.2.2.3 Visualisierung erstellen

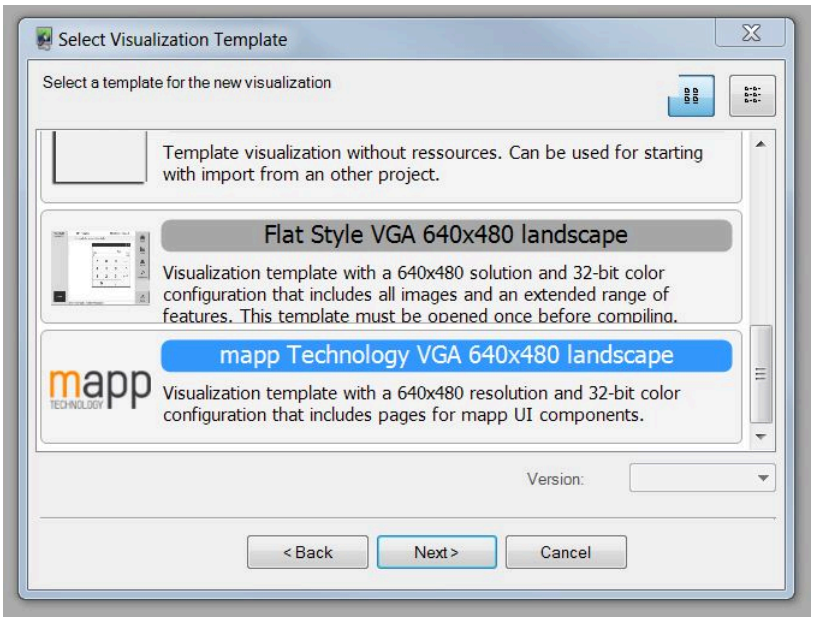
Visualisierung einfügen

Um die Rezeptverwaltung anzeigen zu lassen wird eine VC4-Visualisierung eingfügt.



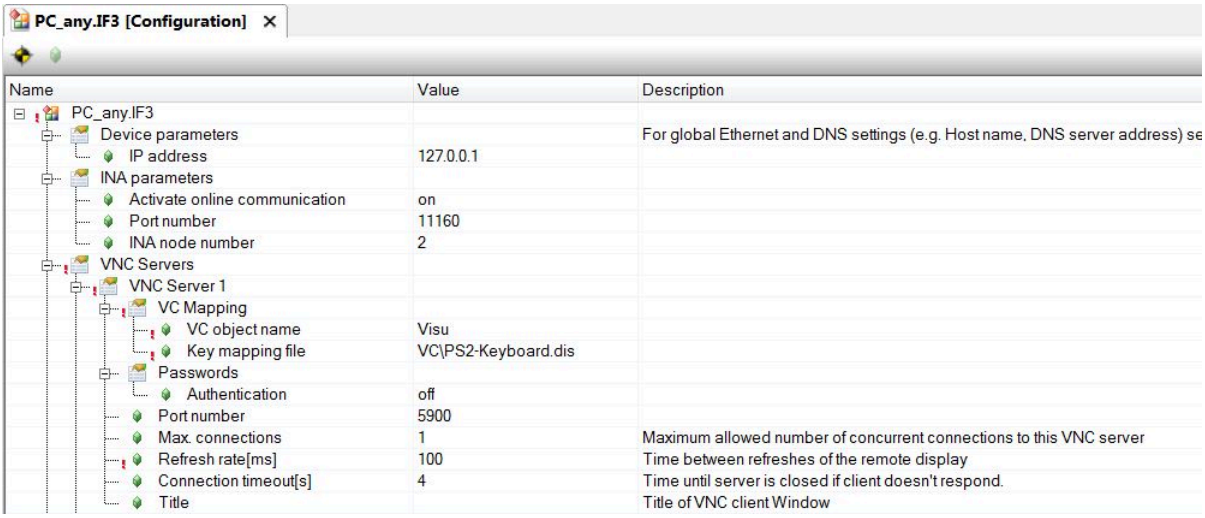
Vorlage auswählen

Damit die mapp Visualisierung angezeigt werden kann, muss die Vorlage "mapp Technology VGA 640x480 landscape" ausgewählt werden.



VNC Server für die Visualisierung konfigurieren

Danach wird die Visualisierung mit einem VNC Server verbunden.

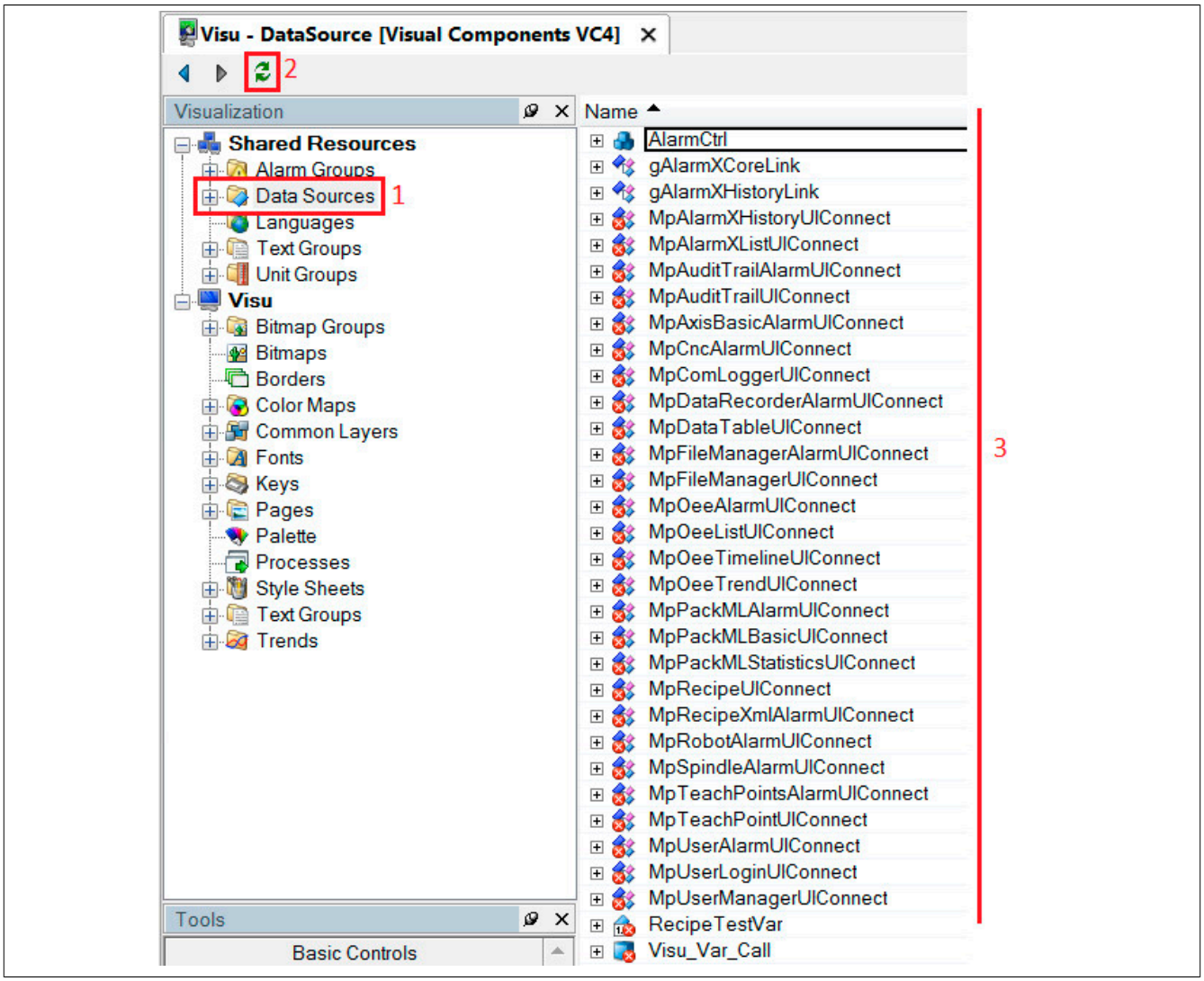


2.2.2.4 Anbinden der Vorlage an das eigene Projekt

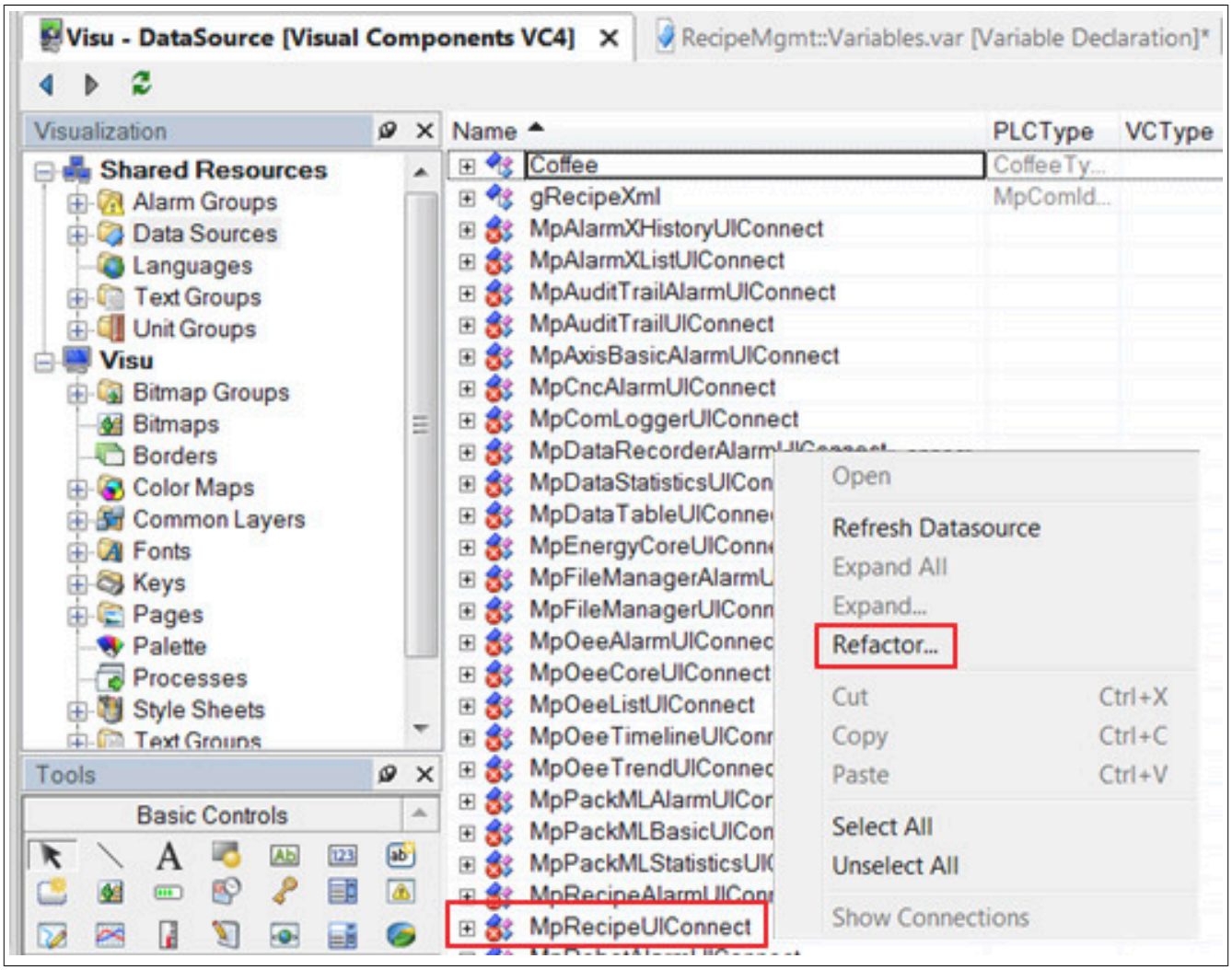
Damit die Rezepte angezeigt werden können, muss die Bibliothek MpFile eingefügt werden!

Erstellen der Verbindung zwischen Task und Visualisierung

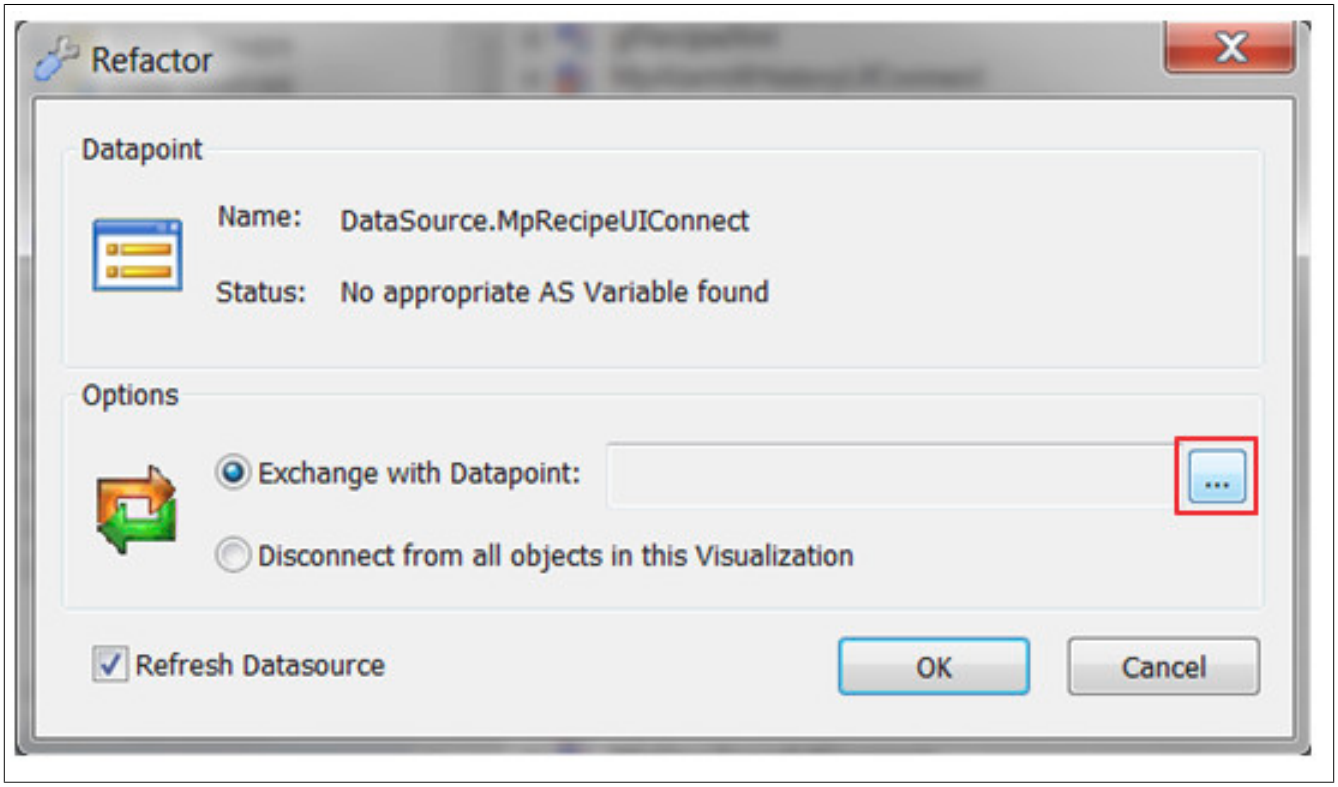
Nach dem Öffnen der Visualisierung wird zunächst die Datenquelle ausgewählt[1] und aktualisiert[2]. Nach dem aktualisieren wird sichtbar welche Variablen in der Visualisierung verbunden aber im Projekt nicht vorhanden sind[3].

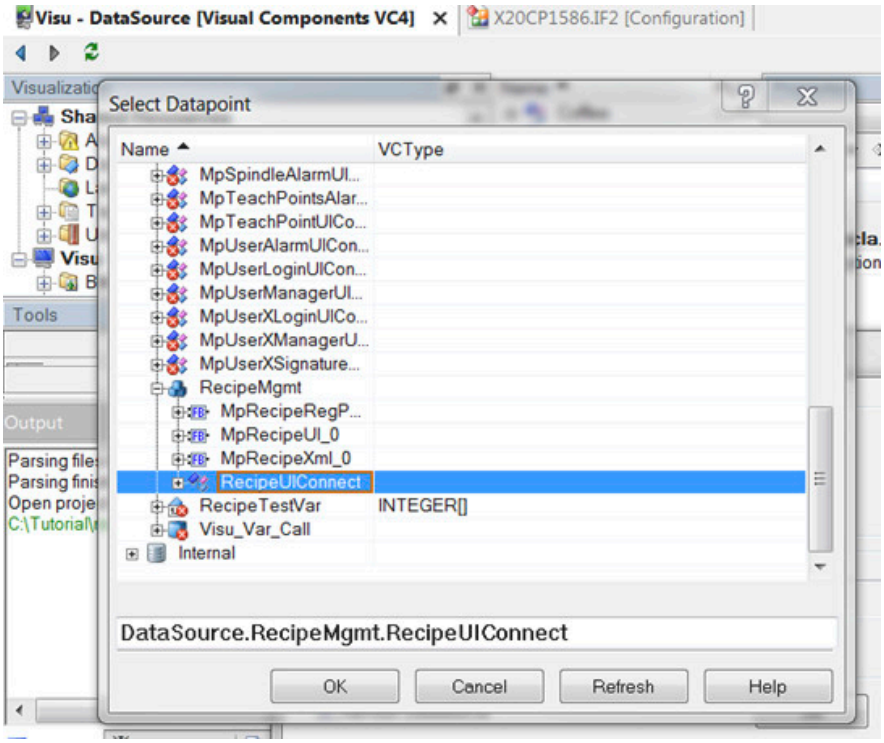


Aus dieser Liste wird die Variable ausgewählt welche mit der eigenen ersetzt werden soll. Mit einem Rechtsklick auf *MpRecipeUIConnect* wählt man den Eintrag *Refactor* aus.



Jetzt wird über das Dialogfenster, welches sich öffnet, die eigene "UIConnect" Struktur ausgewählt.



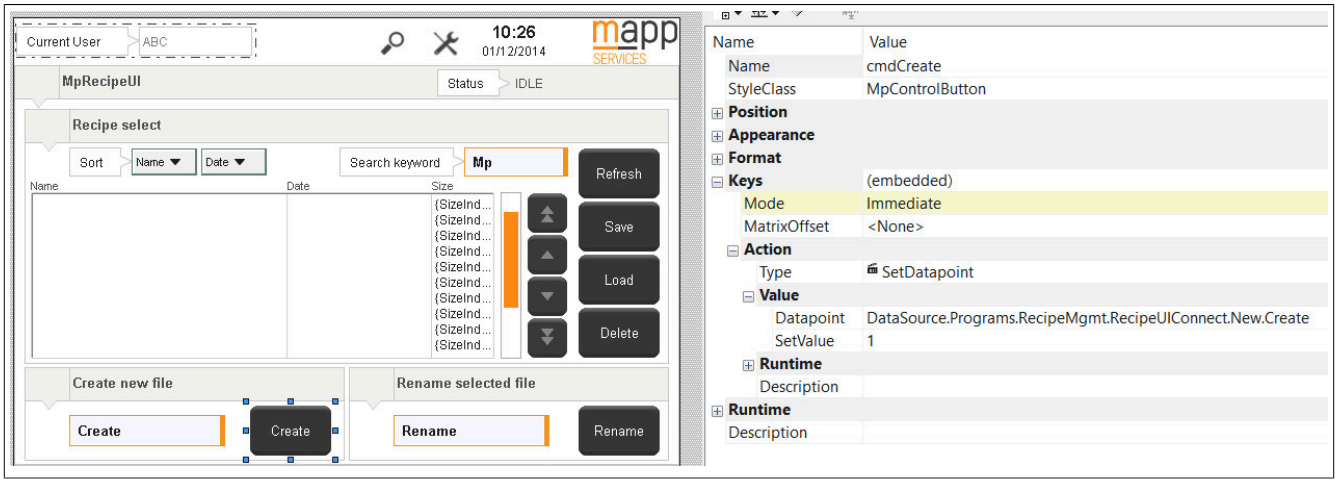


Danach wird der Austausch bestätigt. Alle Verbindungen werden jetzt automatisch erstellt.

Nach dem *Refactoring* wird die Datenquelle wieder aktualisiert. Jetzt sollte die vorhergehende Verbindung *MpRecipeUIConnect* nicht mehr vorhanden sein.

Rezeptseite

Alle vorhandenen Datenpunkte der Rezeptseite wurden durch die eigene Struktur ersetzt.



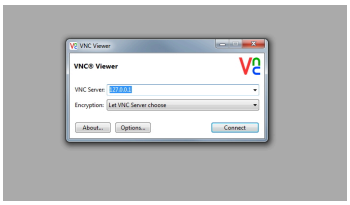
2.2.2.5 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

2.2.2.6 Visalisierung testen

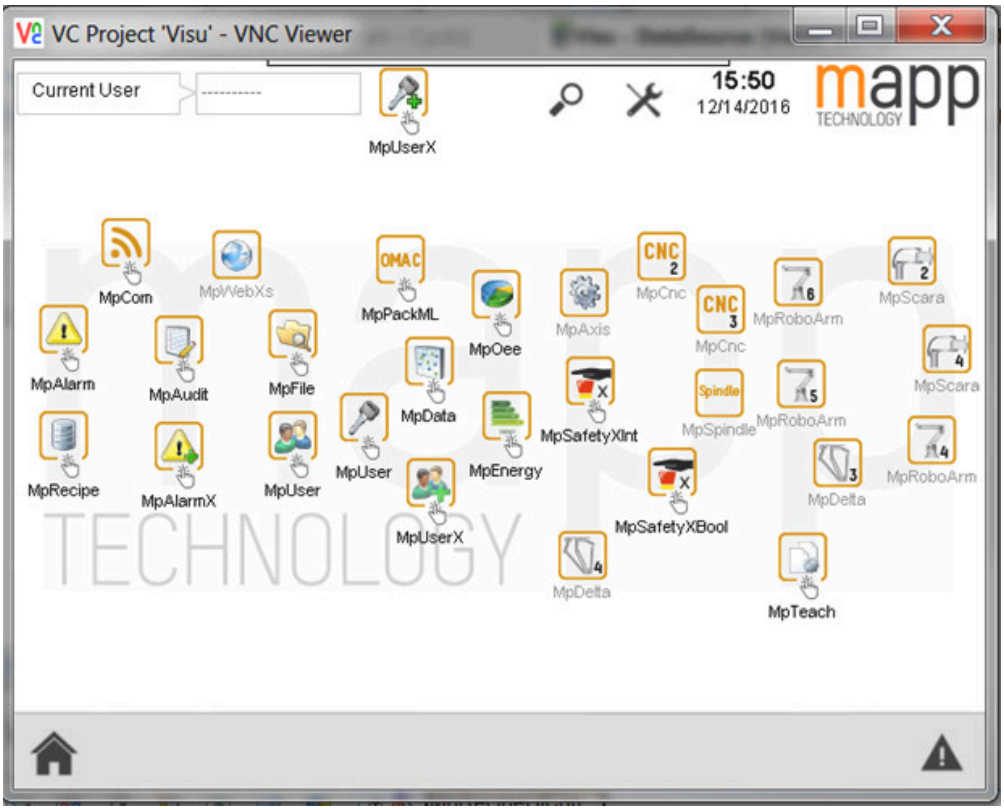
VNC Viewer öffnen

Die Visualisierung kann über den VNC Viewer geöffnet werden.



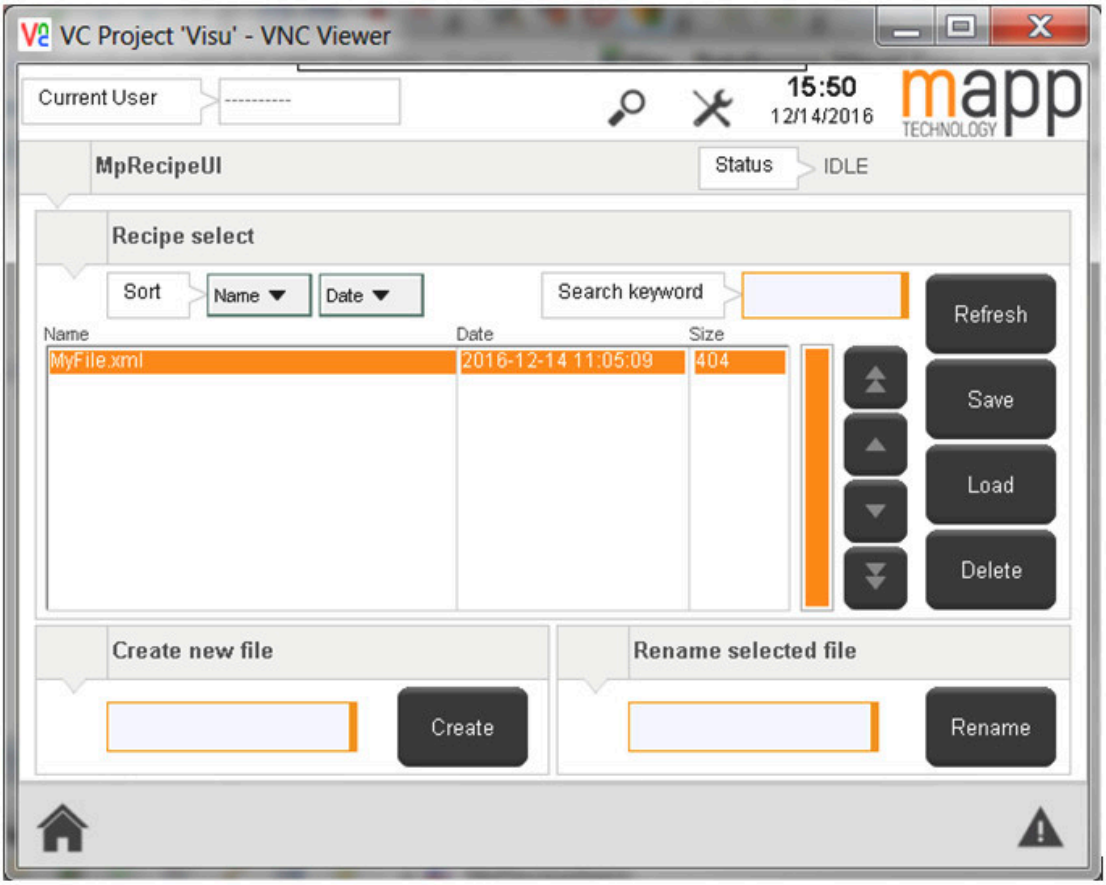
Startseite der Visualisierung

Hat man sich erfolgreich mit der Visualisierung durch den VNC Viewer verbunden, wird die Startseite angezeigt. Durch klicken des Buttons "MpRecipe" gelangt man zu der Rezeptseite.



Rezeptseite

Auf der Rezeptseite können nun die zuvor erstellten Rezepte angezeigt werden. Wie ein Rezept erstellt werden kann, ist hier erklärt. Es können neue Rezepte erstellt, gespeichert, geladen, umbenannt und vieles mehr werden.



2.3 Konfiguration

2.3.1 MpRecipeXml Konfiguration

Die hier in der Konfiguration existierenden Passwörter können verschlüsselt werden. Für mehr Information siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" unter Rahmenbedingungen.

Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein MpRecipeXml de-, und wieder aktiviert werden.

Allgemeine Einstellungen

Im ersten Abschnitt der Konfiguration können verschiedene Speichereinstellungen festgelegt werden. Über "Save automatically" kann eine automatische Abspeicherung der Rezeptdatei aktiviert werden. Dabei muss definiert werden, in welchem Zeitabstand die Datei gespeichert werden soll.

Mithilfe von "Encrypt" kann die Rezeptdatei verschlüsselt werden. "Passphrase" ist dabei der Schlüssel mit dem die Datei kodiert wird. Als Kodierung wird 256 Bit AES verwendet. Der Benutzer kann die Datei *nicht* selbst entschlüsseln. Die "PassPhrase" wird nur benötigt, um dem Schreiber der Applikation selbst die Möglichkeit zu geben den Sicherheitsgrad zu beinflussen. Um eine sinnvolle Verschlüsselung zu gewährleisten **muss** die "PassPhrase" verändert werden! Ziel der Verschlüsselung ist es, dass das Rezept nur durch mapp Recipe gelesen und gespeichert werden kann. Ein externes öffnen der Rezeptdatei wird dadurch verhindert. Das bedeutet, ein extern verschlüsseltes Rezept kann auch nicht durch mapp Recipe entschlüsselt und gelesen werden.

Über "Auditing" kann festgelegt werden, ob Ereignisse in der Rezeptverwaltung aufgezeichnet werden sollen oder nicht. Für mehr Information siehe Auditing der Rezeptverwaltung.

"Auto file extension" legt fest, ob die Dateiendung (.xml) automatisch angehängt wird oder nicht.

Signieren von Rezepten

Über den Parameter *Signature* kann ein Rezept signiert werden. Dadurch wir sichergestellt, dass es nur geladen werden kann, wenn das Rezept nicht manipuliert wurde. Die Signatur wird erstellt aus einer Kombination aus *Passphrase* und dem Inhalt des Rezepts. Beim Laden des Rezepts wird nun nach einer Datei im selben File Device mit demselben Namen wie das Rezept, aber mit der Endung *.mprecipesgn* gesucht. Mithilfe der darin enthaltenen Signatur wird geprüft, ob der Inhalt des Rezepts gültig ist.

Wird der Inhalt verändert, so muss auch die Signatur angepasst werden. Das ist auf zwei Arten machbar:

- Speichern über MpRecipe
- Externes erstellen des Rezepts inkl. der Signatur

Signieren über MpRecipe

Wird das Rezept von MpRecipe gespeichert, wird eine neue Signatur erstellt. Der Inhalt wird über die Applikation definiert. Die *Passphrase* wird in der Konfiguration angegeben. Beim Speichern wird die Datei, welche die Signatur enthält angepasst, basiernd auf der *Passphrase* und dem Inhalt des Rezepts.

Externes erstellen des Rezepts inkl. Signatur

Zum Erstellen der Signatur wird ein HMAC-SHA256 - Hash Algorithmus verwendet. Dazu müssen der genaue Inhalt des Rezepts, sowie die *Passphrase* bekannt sein. Danach wird eine Datei mit dem Namen des Rezepts und der Endung *.mprecipesgn* erstellt.

Diese Endung wird nach der .xml-Endung angegeben, wie zum Beispiel "MyRecipe.xml.mprecipesgn"!

In diese Datei wird die Signatur kopiert. Der Inhalt dieser Datei besteht ebenfalls aus XML-Syntax. Beide Dateien, Rezept- und Signaturdatei, werden nun auf die Steuerung in denselben Ordner kopiert.

Wie eine externe Signatur erstellt werden kann, ist im Anwendungsfall Erstellen eines signierten Rezepts mit einem externen Tool erklärt.

2.3.2 MpRecipeCsv Konfiguration

Die hier in der Konfiguration existierenden Passwörter können ab der mapp Services Version >= 5.11.x verschlüsselt werden. Für mehr Information siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" unter Rahmenbedingungen.

Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein MpRecipeCsv de-, und wieder aktiviert werden.

Allgemeine Einstellungen

Im ersten Abschnitt der Konfiguration können verschiedene Speichereinstellungen festgelegt werden. Über "Save automatically" kann eine automatische Abspeicherung der Rezeptdatei aktiviert werden. Dabei muss definiert werden, in welchem Zeitabstand die Datei gespeichert werden soll.

Mithilfe von "Encrypt" kann die Rezeptdatei verschlüsselt werden. "Passphrase" ist dabei der Schlüssel mit dem die Datei kodiert wird. Als Kodierung wird 256 Bit AES verwendet. Der Benutzer kann die Datei *nicht* selbst entschlüsseln. Die "PassPhrase" wird nur benötigt, um dem Schreiber der Applikation selbst die Möglichkeit zu geben den Sicherheitsgrad zu beinflussen. Um eine sinnvolle Verschlüsselung zu gewährleisten **muss** die "PassPhrase" verändert werden! Ziel der Verschlüsselung ist es, dass das Rezept nur durch mapp Recipe gelesen und gespeichert werden kann. Ein externes öffnen der Rezeptdatei wird dadurch verhindert. Das bedeutet, ein extern verschlüsseltes Rezept kann auch nicht durch mapp Recipe entschlüsselt und gelesen werden.

Über "Auditing" kann festgelegt werden, ob Ereignisse in der Rezeptverwaltung aufgezeichnet werden sollen oder nicht. Für mehr Information siehe Auditing der Rezeptverwaltung.

"Auto file extension" legt fest, ob die Dateiendung (.xml) automatisch angehängt wird oder nicht.

Signieren von Rezepten

Über den Parameter *Signature* kann ein Rezept signiert werden. Dadurch wir sichergestellt, dass es nur geladen werden kann, wenn das Rezept nicht manipuliert wurde. Die Signatur wird erstellt aus einer Kombination aus *Passphrase* und dem Inhalt des Rezepts. Beim Laden des Rezepts wird nun nach einer Datei im selben File Device mit demselben Namen wie das Rezept, aber mit der Endung .mprecipesgn gesucht. Mithilfe der darin enthaltenen Signatur wird geprüft, ob der Inhalt des Rezepts gültig ist.

Wird der Inhalt verändert, so muss auch die Signatur angepasst werden. Das ist auf zwei Arten machbar:

- Speichern über MpRecipe
- Externes erstellen des Rezepts inkl. der Signatur

Signieren über MpRecipe

Wird das Rezept von MpRecipe gespeichert, wird eine neue Signatur erstellt. Der Inhalt wird über die Applikation definiert. Die *Passphrase* wird in der Konfiguration angegeben. Beim Speichern wird die Datei, welche die Signatur enthält angepasst, basiernd auf der *Passphrase* und dem Inhalt des Rezepts.

Externes erstellen des Rezepts inkl. Signatur

Zum Erstellen der Signatur wird ein HMAC-SHA256 - Hash Algorithmus verwendet. Dazu müssen der genaue Inhalt des Rezepts, sowie die *Passphrase* bekannt sein. Danach wird eine Datei mit dem Namen des Rezepts und der Endung .mprecipesgn erstellt.

Diese Endung wird nach der .csv-Endung angegeben, wie zum Beispiel "MyRecipe.csv.mprecipesgn"!

In diese Datei wird die Signatur kopiert. Der Inhalt dieser Datei besteht aus XML-Syntax. Beide Dateien, Rezept- und Signaturdatei, werden nun auf die Steuerung in denselben Ordner kopiert.

Wie eine externe Signatur erstellt werden kann, ist im Anwendungsfall Erstellen eines signierten Rezepts mit einem externen Tool erklärt.

2.4 Anwendungsfälle

2.4.1 Maschine mit einfacher Rezeptverwaltung

Anforderung

Eine Maschine soll Rezeptdaten wie Zutaten und Mengen eines Produktes (z.B. Kaffee) in einer Datei ablegen. Jedes Produkt (z.B. Cappuccino, Espresso, Latte Macchiato) hat dabei seine eigene Rezeptdatei. Eine der drei oben genannten Kaffeetypen kann in der Applikation ausgewählt und geladen werden.

Lösung

Komponentenliste

- ***MpRecipeXml*** (Eigener MpLink): Speichern und Laden der Rezeptdateien
- ***MpRecipeRegPar*** (MpLink von MpRecipeXml): Registrierung von Prozessvariablen. Einzelvariablen oder ganze Strukturen können registriert werden

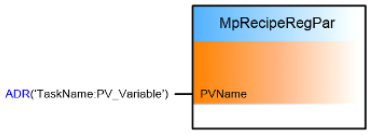
Verbindungsschema

Verwendung der mapp Komponenten

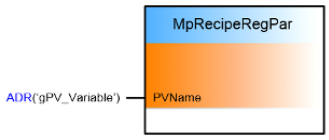
Mit der **MpRecipeXml** Komponente kann ein Rezept geladen und gespeichert werden. Der Inhalt wird über die **MpRecipeRegPar** Komponente definiert. Über **MpRecipeRegPar** wird die gesamte Struktur (Zutaten und Mengen) des Kaffees registriert.

Setup / Einstellungen

Registrierung einer **lokalen** Variable:



Registrierung einer **globalen** Variable



2.4.2 Maschine mit Produkt- und Maschinenparametern (Kategorie)

Anforderung

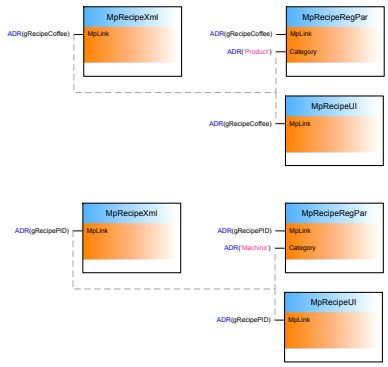
Eine Maschine soll Produktparameter wie Zutaten, Mengen und Daten sowie unterschiedliche Maschinenparameter (PID) speichern. Sowohl die Produkt- als auch die Maschinenparameter sollten mit Hilfe von XML-Dateien auf zwei verschiedenen Speichermedien abgelegt und unabhängig voneinander geladen werden können. Die Produkt- und Maschinenparameter sollen zudem über eine spezielle Service Seite auf der Visualisierung bedienbar sein.

Lösung

Komponentenliste

- ***MpRecipeXml*** (Eigener MpLink): Speichern und Laden der Rezeptdatei
- ***MpRecipeRegPar*** (MpLink von MpRecipeXml): Registrierung der Prozess Variablen. Mit dieser Komponenten werden Rezeptstruktur und Einstellungen registriert
- ***MpRecipeUI*** (MpLink von MpRecipeXml): Verbindung zur VC4 Visualisierung

Verbindungsschema



Verwendung der mapp Komponenten

MpRecipeXML wird verwendet, um Prozessvariablen in XML Dateien zu speichern und zu laden. Da die Dateien in zwei separaten Rezepten gespeichert werden sollen, werden zwei unterschiedliche **MpRecipeXML** Instanzen verwendet. Um die Produkt- und die Maschinenparameter zu trennen, werden die Produktparameter und die Maschinenparameter in zwei verschiedene Strukturen aufgeteilt.

Die Produktparameter Struktur wird mittels des **MpRecipeRegPar** unter der Kategorie "Product" registriert. Die Kategorie wird über den Eingang "Category" bestimmt. Alle Parameter, für welche eine Kategorie festgelegt ist, werden im Rezept derselben Kategorie gespeichert.

Die zweite Struktur für die Maschinenparameter wird mit einem weiteren **MpRecipeRegPar** unter der Kategorie "Maschine" registriert.

Die gleichen Kategorienamen werden ebenfalls für die beiden **MpRecipeXml** Instanzen verwendet. Dies bedeutet, dass die eine **MpRecipeXml** Instanz die Produkt- und die zweite Instanz die Maschinenparameter speichert.

Um die Trennung der beiden Rezepte auch auf der Visualisierung zu sehen, werden zwei verschiedene **MpRecipeUI** verwendet. Diese können dann unabhängig voneinander die beiden Rezepte verwalten. Einer ist am **MpRecipeXml** der Kategorie "Produkt" und der andere mit dem **MpRecipeXml** mit der Kategorie "Maschine" verbunden (Verwendung des gleichen MpLinks).

2.4.3 Rezeptverwaltung mit Synchronisierung beim Speichern

Anforderung

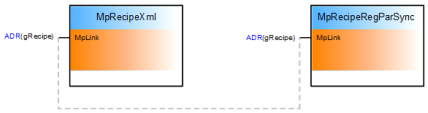
Eine Maschine soll in der Lage sein, Rezepte in Form von XML Dateien zu speichern. Rezeptparameter können über die Visualisierung geändert werden. Bevor ein Rezept mit geänderten Daten gespeichert wird, soll eine Synchronisation stattfinden. Damit wird sichergestellt, dass nur vollständige Rezepte und ein konsistentes Datenabbild in den Rezeptdateien gespeichert werden.

Lösung

Komponentenliste

- ***MpRecipeXml*** (Eigener MpLink): Speichern und Laden der Rezeptdatei
- ***MpRecipeRegParSync*** (MpLink von MpRecipeXml): Registrierung der Prozessvariablen (PV). Es können auch ganze Strukturen registriert werden

Verbindungsschema



Verwendung der mapp Komponenten

Mit **MpRecipeXml** lassen sich PV Daten in einer XML Datei speichern. Da es möglich sein soll, die Parameter über die Visualisierung aus zu bearbeiten, muss der **MpRecipeRegParSync** für die Durchführung einer synchronisierten Speicherung verwenden werden. Dadurch wird sichergestellt, dass die Rezepte nur als komplettes Set von Parameterwerten abgespeichert werden.

Eine Parameter Struktur ist mit der Visualisierung verbunden, eine weitere Parameter Struktur wird mit **MpRecipeRegParSync** registriert. Wenn nun der Speicherbefehl am **MpRecipeXml** aufgerufen wird, wird die „SaveNotification“ am **MpRecipeRegParSync** gesetzt. Dies ist eine Benachrichtigung, dass ein Speichervorgang beginnen soll. Nun kann die Applikation für ein konsistentes Datenabbild sorgen, indem sie die Parameter Struktur welche mit der Visualisierung verbunden ist, umkopiert und gleichzeitig auf Plausibilität prüft.

Sobald die Daten kopiert sind, wird der Befehl auf „ConfirmSave“ am **MpRecipeRegParSync** gesetzt.

Über ConfirmSave wird dem **MpRecipeXml** mitgeteilt, dass die Parameter sicher und konsistent sind, und dass die PV Daten nun in die Rezeptdatei geschrieben werden können.

2.4.4 Maschine mit neuen Rezeptdaten sicher aktualisieren (UpdateNotification)

Anforderung

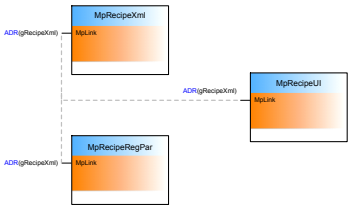
Eine Spritzgießmaschine soll in der Lage sein, Rezepte in Form von XML Dateien zu speichern. Rezeptparameter können über die Visualisierung geändert und jederzeit geladen werden. Bevor ein Rezept mit den geänderten Daten von der Maschine verwendet wird, soll sichergestellt sein, dass sich der Produktionszyklus dabei am Ende befindet. Somit werden die neuen Rezeptdaten nicht während des Einspritzvorgangs übernommen, was zu einer Fehlproduktion führen wurde, sondern erst wenn das Produkt fertig ist und ein neuer Zyklus sicher stattfinden kann.

Lösung

Komponentenliste

- ***MpRecipeXml*** (Eigener MpLink): Speichern und Laden der Rezeptdatei
- ***MpRecipeRegPar*** (MpLink von MpRecipeXml): Registrierung der Prozessvariablen (PV). Es können auch ganze Strukturen registriert werden
- ***MpRecipeUI*** (MpLink von MpRecipeXml): Verbindung zur VC4 Visualisierung

Verbindungsschema



Verwendung der mapp Komponenten

Mit **MpRecipeXml** lassen sich PV Daten in einer XML Datei speichern. Der Inhalt wird über die **MpRecipeRegPar** Komponente definiert.

Über **MpRecipeRegPar** wird die gesamte Struktur des Produkts registriert. In unseren Fall haben wir zwei Parameterstrukturen. Eine für die aktuellen Produktparameter, welche über MpRecipeRegPar registriert ist. Die zweite Struktur ist direkt mit der Maschine verbunden, das heißt sie beeinflusst zum Beispiel den Einspritzvorgang.

Ein Ladebefehl ("Load = *TRUE*") ist jederzeit möglich. Damit die geänderten Parameter aber erst am Ende eines Produktzyklus übernommen werden, muss der Parameter "UpdateNotification" verwendet werden. Dieser Parameter muss auf *TRUE* gesetzt werden, sobald das Ausgangssignal "CommandDone = TRUE" ist.

Durch setzen von "UpdateNotification" wird der Komponente **MpRecipeRegPar** signalisiert, dass ein neues Rezept zur Verfügung steht. Dies kann am Ausgangssignal ("UpdateNotification = TRUE") von **MpRecipeRegPar** abgelesen werden. Befindet sich der Produktionszyklus am Ende (z.B.: "ProductCycleEnd = *TRUE*") und das Ausgangssignal "UpdateNotification = TRUE" können die Rezeptdaten von der Maschine verwendet werden, beispielsweise indem genau und nur zu diesem Zeitpunkt die Rezept-Parameter, also die Struktur mit den aktuellen Parametern, auf die Parameterstruktur umkopiert werden, welche direkten Einfluss auf die den Einspritzvorgang haben.

2.4.5 Schützen der Rezeptdaten gegen Manipulation

Anforderung

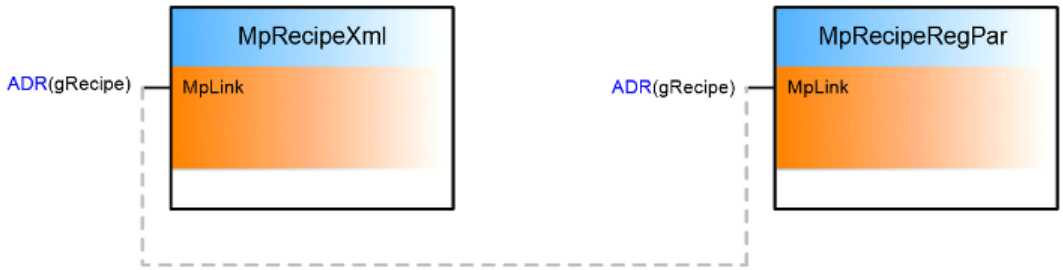
Die Rezeptdaten sollen gegen externe Manipulation geschützt sein, ohne dass die gesamte Datei verschlüsselt wird. Es soll nur autorisierten Personen gestattet sein, die Rezeptdaten zu ändern. Wurde das Rezept durch eine nicht autorisierte Person verändert, muss das Laden des Rezepts verhindert werden.

Lösung

Komponentenliste

- ***MpRecipeXml*** (Eigener MpLink): Speichern und Laden der Rezeptdatei
- ***MpRecipeRegPar*** (MpLink von MpRecipeXml): Registrierung der Prozessvariablen (PV). Es können auch ganze Strukturen registriert werden.

Verbindungsschema



Konfiguration der Komponenten

gRecipeXml		
Recipe		
Save automatically	No auto save	
Encrypt	FALSE	Encrypt recipe file
Passphrase	Insert your passph...	Password (or phrase) for the recipe
Auditing	None	
Auto file extension	TRUE	Set to TRUE to add a file extension automatically if the file extension is not specified in the recipe name.
Signature	Required	
Passphrase	MyPassphrase	Password (or phrase) for the recipe signature.
Alarms	MpAlarm	

Verwendung der mapp Komponenten

Die MpRecipe Konfiguration erlaubt, ein Passwort zum Signieren der Rezeptdatei anzugeben. Mit dieser Option wird für jedes Rezept eine Signaturdatei erstellt, welche einen HMAC-Wert der Rezeptdatei enthält. Dieser HMAC-Wert wird von MpRecipe beim Laden verwendet, um sicherzustellen, dass die Rezeptdatei nicht verändert wurde. Die Signatur ist einzigartig für jede Rezeptdatei und das definierte Passwort. Details zum Thema Signieren von Rezepten sind hier zu finden.

Beim Laden des Rezepts überprüft MpRecipe die Signatur mit dem Inhalt der Rezeptdatei. Wenn die Signatur nicht mit dem Inhalt übereinstimmt (der Dateiinhalt wurde bearbeitet), werden die Daten nicht geladen und der Funktionsbaustein zeigt den Fehler "Fehler beim Laden eines Rezepts" an. Weitere Infos werden im Logger angezeigt. Dabei erscheint die Meldung "Das Rezept oder die Signatur wurde verändert".

2.4.6 Erstellen eines signierten Rezepts mit einem externen Tool

Anforderung

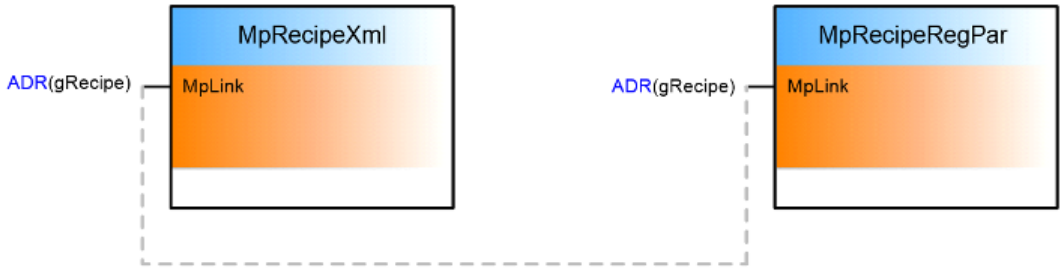
Das Rezept mit den Maschinendaten wird vom Maschinenbauer erstellt. Der Maschinenbauer möchte verhindern, dass der Endkunde die Maschinendaten verändert und damit die Maschine beschädigt. Dazu soll das Rezept mit den Maschinendaten signiert werden.

Lösung

Komponentenliste

- ***MpRecipeXml*** (Eigener MpLink): Speichern und Laden der Rezeptdatei.
- ***MpRecipeRegPar*** (MpLink von MpRecipeXml): Registrierung der Prozessvariablen (PV). Es können auch ganze Strukturen registriert werden.

Verbindungsschema



Konfiguration der Komponente

Die MpRecipe Konfiguration erlaubt, ein Passwort zum Signieren der Rezeptdatei anzugeben.

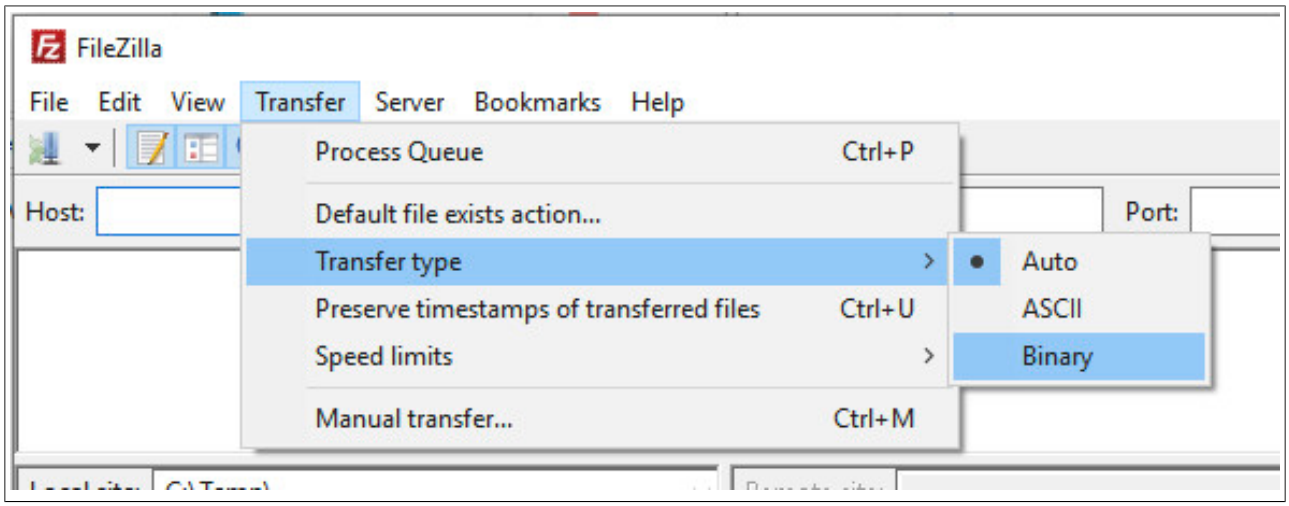
gRecipeXml		
Recipe		
Save automatically	No auto save	
Encrypt	FALSE	Encrypt recipe file
Passphrase	Insert your passph...	Password (or phrase) for the recipe
Auditing	None	
Auto file extension	TRUE	Set to TRUE to add a file extension automatically if the file extension is not specified in the recipe name.
Signature	Required	
Passphrase	MyPassphrase	Password (or phrase) for the recipe signature.
Alarms	MpAlarm	

Verwendung der mapp Komponenten

Wird nun ein Rezept geladen, wird anhand des Inhalts, sowie des definierten Passworts überprüft, ob das Rezept gültige Daten enthält.

Das Rezept wird extern erstellt.

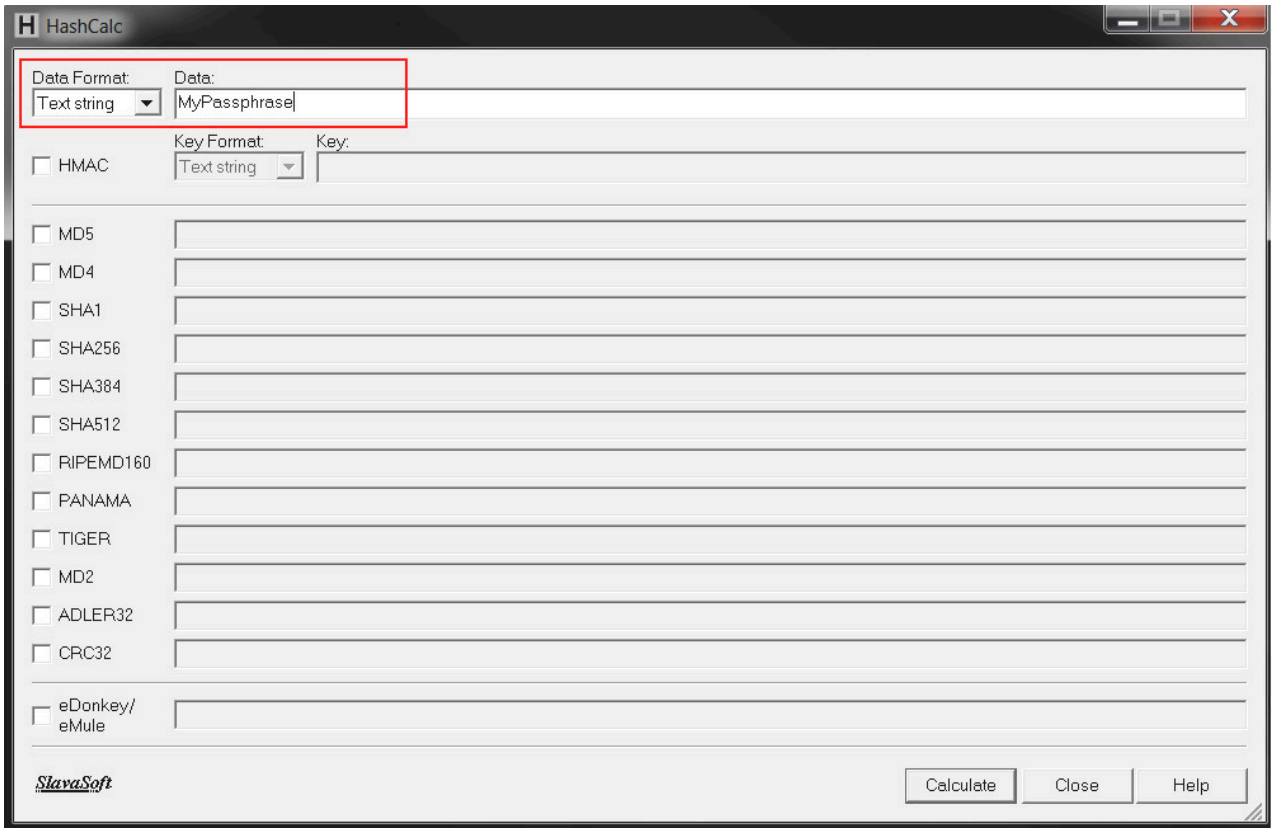
> **Wird das Rezept von der Steuerung mithilfe eines FTP-Client Programms (Beispiel FileZiller), heruntergeladen, so muss folgendes beachtet werden:**
>
> **Die Datei muss binär transferiert werden. Dadurch wird sichergestellt, dass die Binär-Dateien der gewünschten Datei korrekt transferiert werden. Nur dadurch kann eine gültige Signatur erstellt werden.**



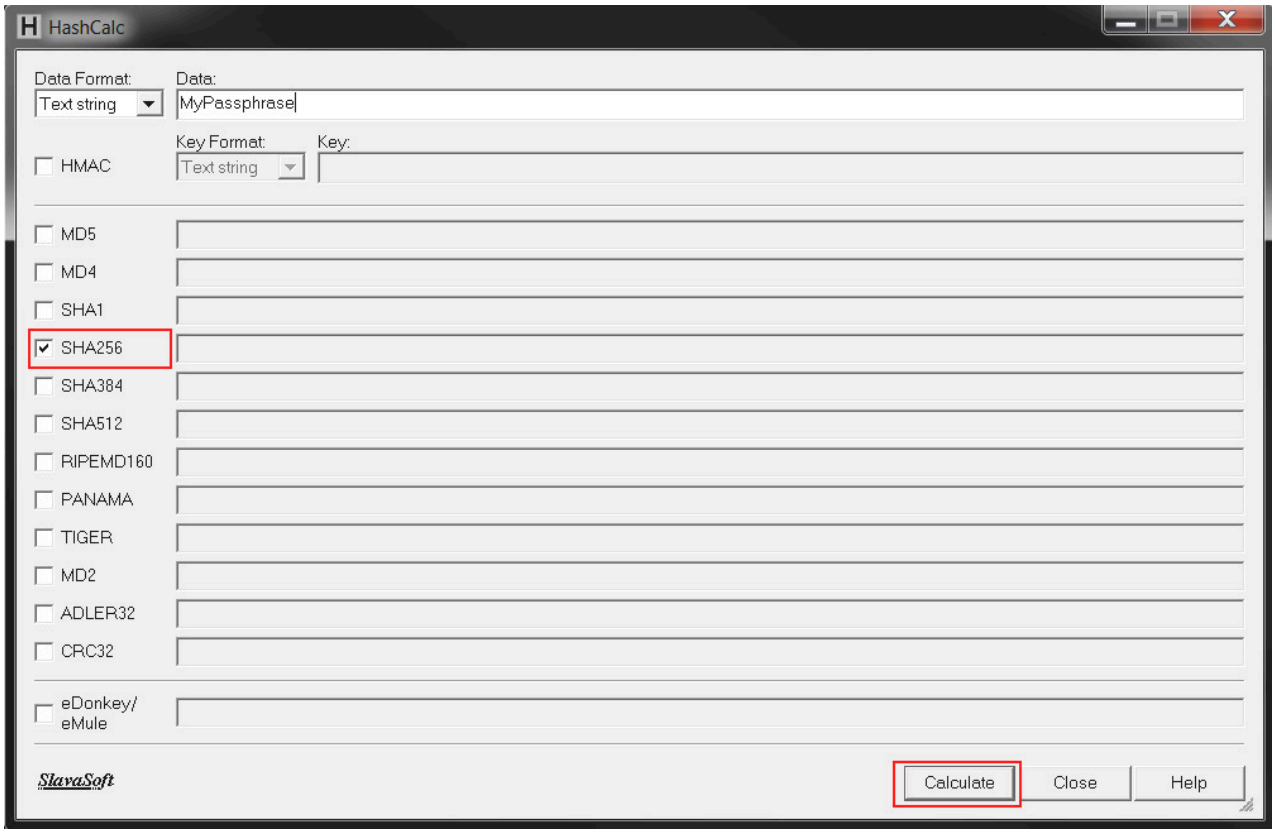
Danach muss die Signatur generiert werden und in einer zweiten Datei abgespeichert werden. Die Signatur ist ein SHA256 basierter HMAC-Wert. Diese Signatur kann wie folgt generiert werden:

Für unseren Anwendungsfall wird die Signatur über folgendes Tool generiert: Hash-Calculator

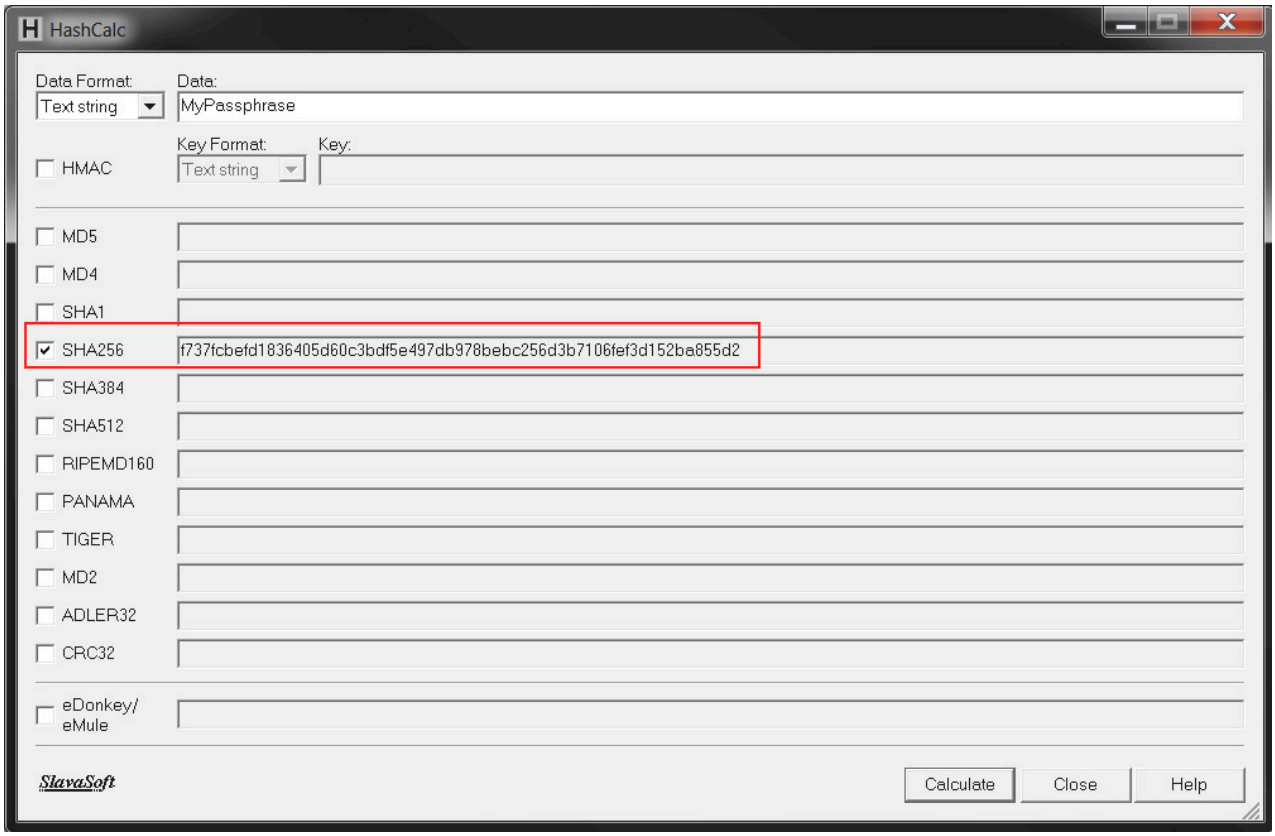
- Eingeben der Passphrase:



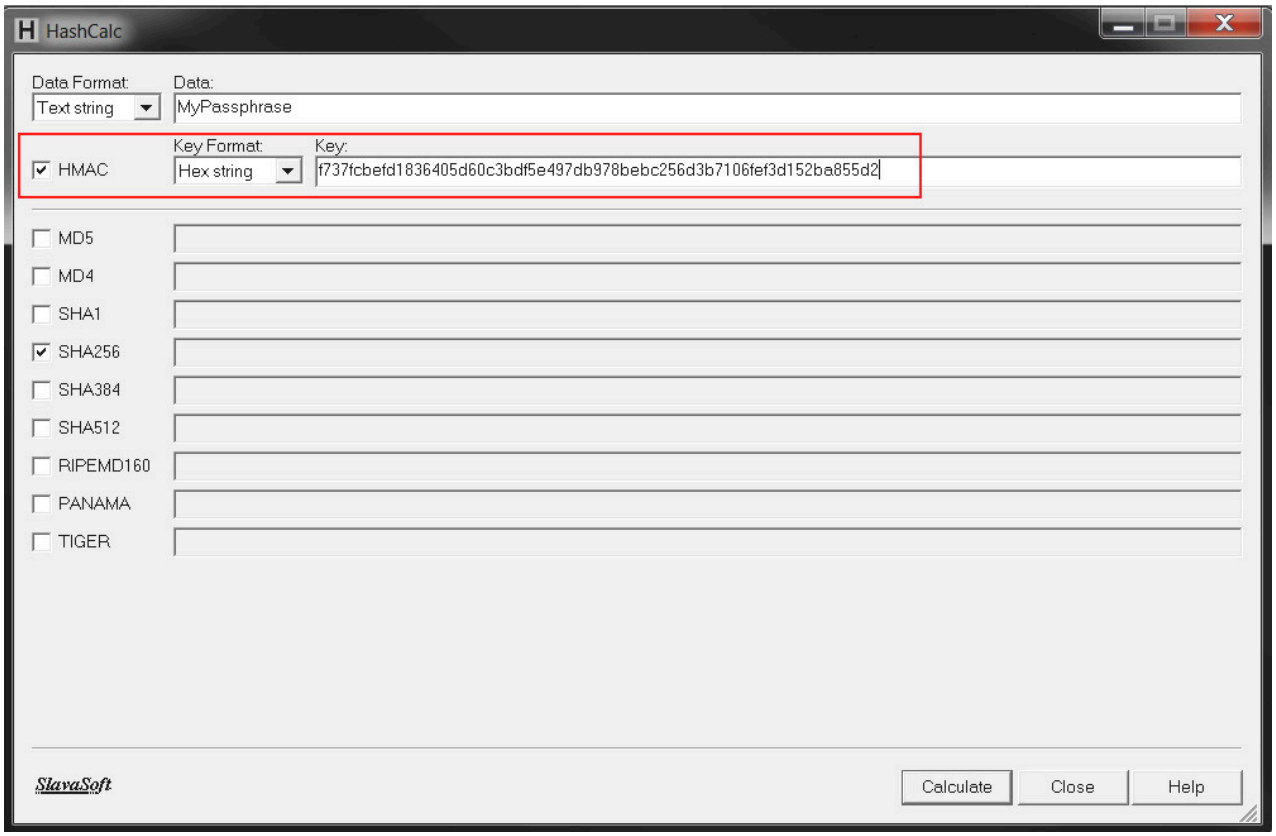
- Zum generieren des SHA256-Wertes "SHA256" auswählen und auf "Calculate" drücken:



- Der berechnete SHA256-Wert wird dann hier angezeigt:

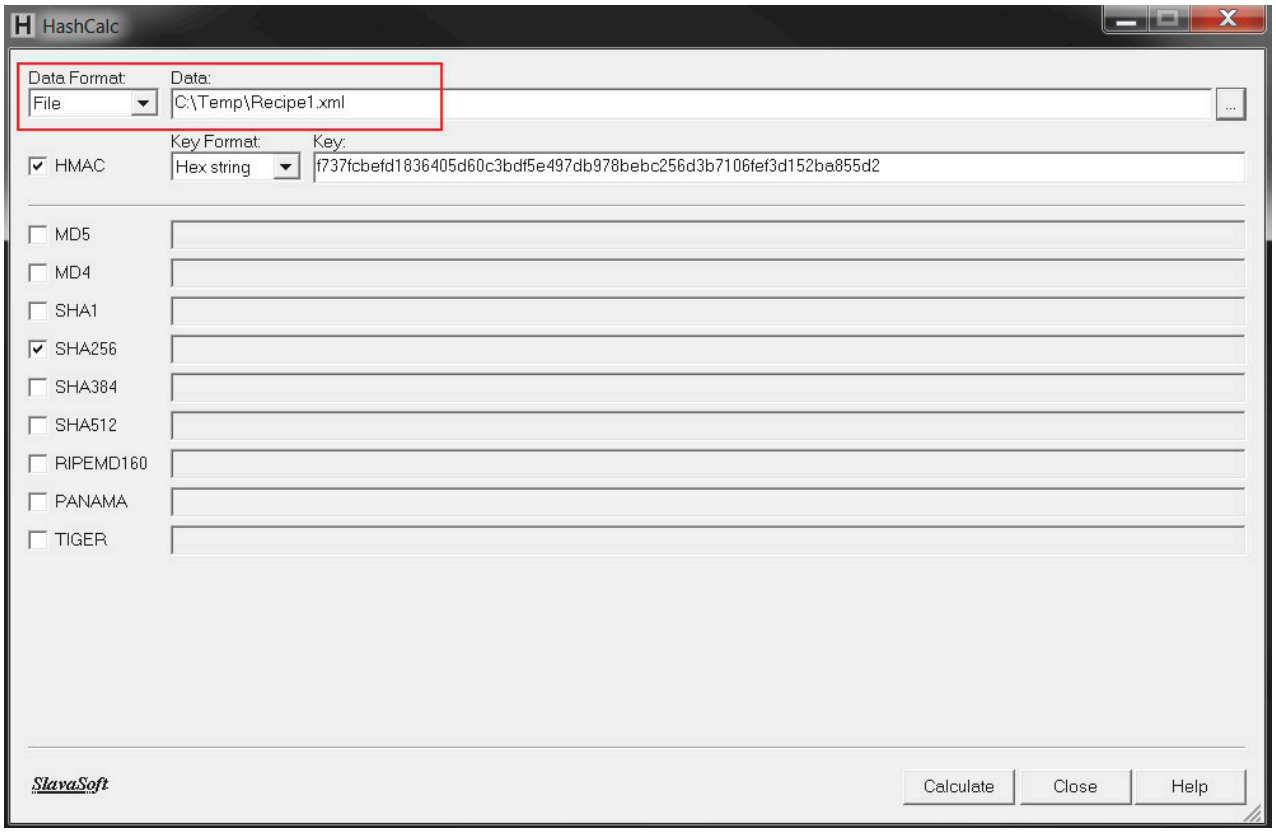


- Auswählen des HMAC-Keys:
  Zuerst wird der SHA256-Wert kopiert. Danach wird HMAC ausgewählt und der Key als "Hex String" eingefügt.



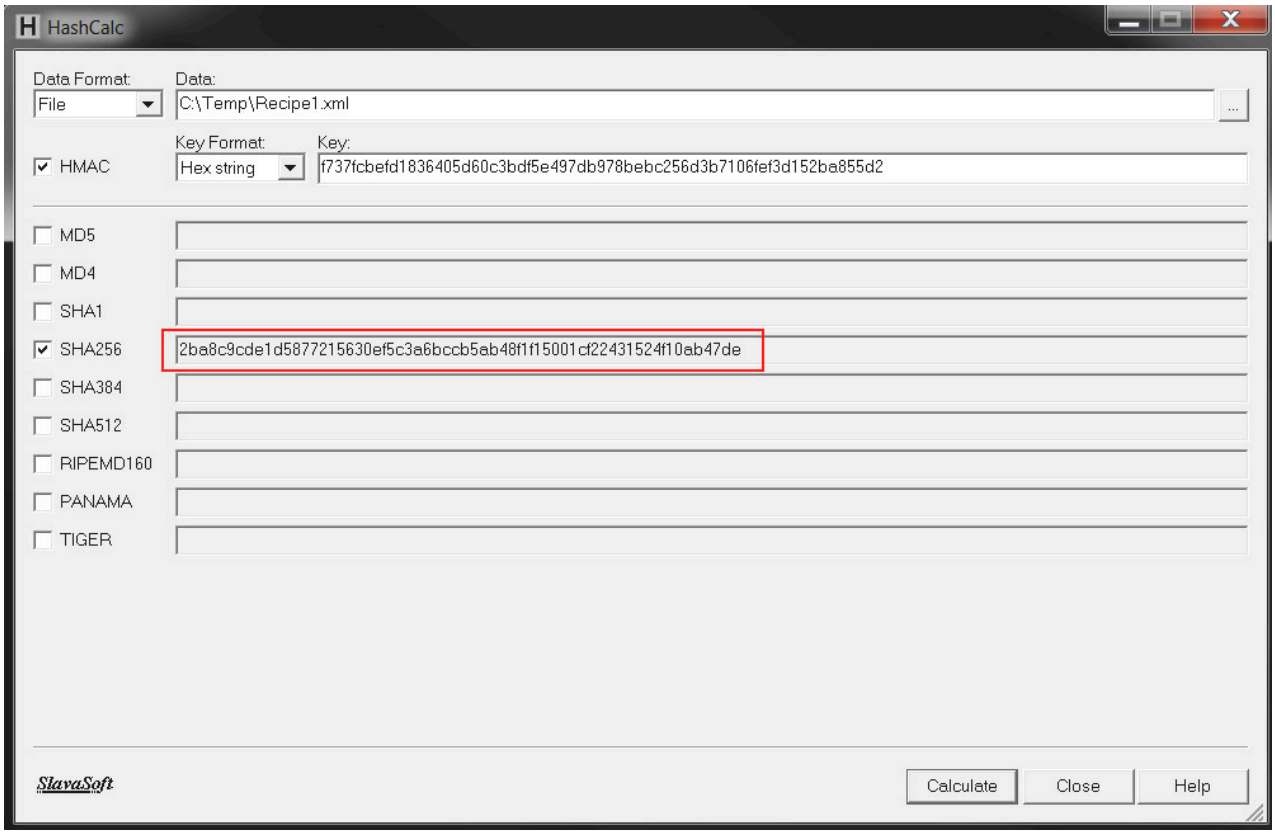
- Auswählen der Rezeptdatei:

  Als nächstes muss die Rezeptdatei angegeben werden. Als Data Format "File" wählen und die Rezeptdatei auswählen.

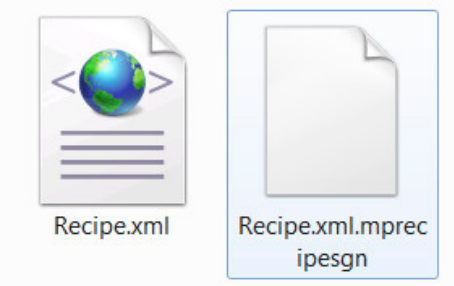


  Nach dem Auswählen der Datei "Calculate" drücken.

- Die Signatur wird dann hier angezeigt:



Dieser Wert wird dann in eine neue Datei gespeichert. Die Datei muss den gleichen Namen haben wie die Rezeptdatei, einzig die Dateiendung ist .mprecipesgn (z.b. Recipe1.xml.mprecipesgn).



Die Struktur der Datei muss so aussehen, wobei unter Value der vorher generierte SHA256-Wert eingetragen wird.

Wenn die Rezeptdatei und die Signaturdatei mit dem selben Namen und auf dem gleichen Speichermedium (definiert über "DeviceName") vorhanden sind, und die Signieroption in der Konfiguration aktiviert ist, wird MpRecipe beim Laden die Signatur prüfen. Wenn die Rezeptdatei nicht verändert wurde, ist die Prüfung erfolgreich. Ansonsten wird ein Fehler angezeigt.

2.5 Diagnose

Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

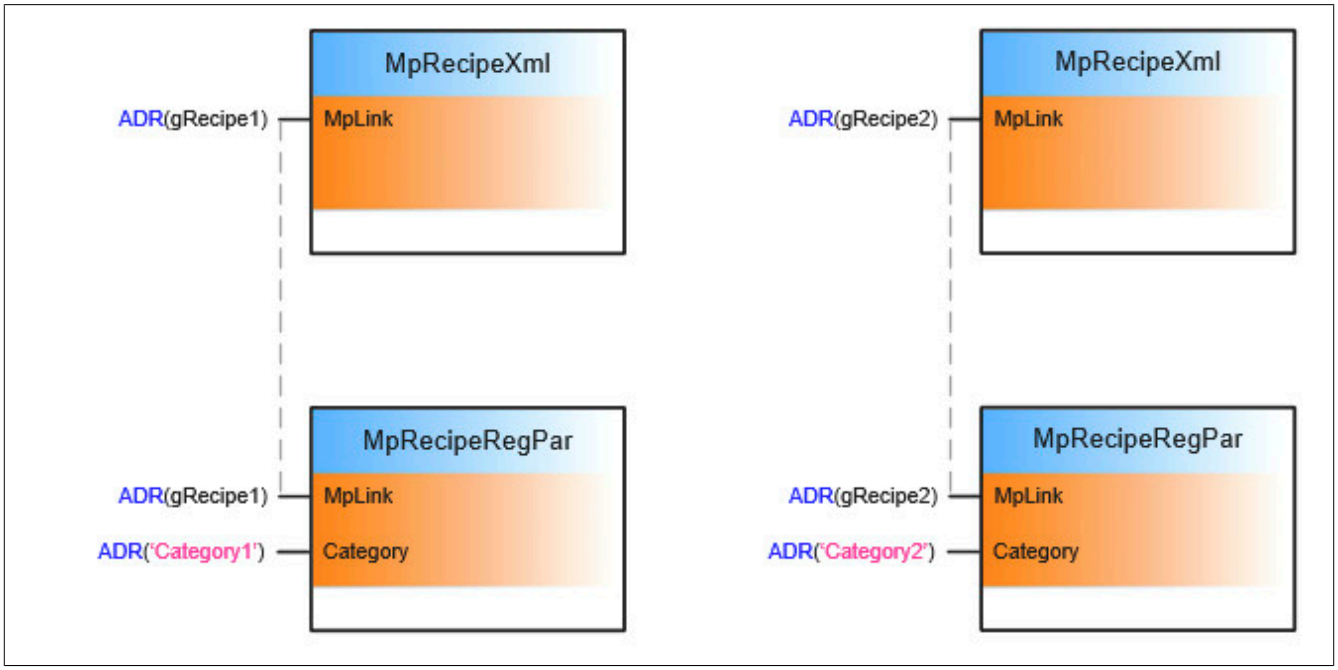
2.5.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp Recipe auftreten können.

Bei Verwendung von mehreren mapp Recipe Instanzen werden immer alle Parameter in einem Rezept abgespeichert, obwohl unterschiedliche MpLinks verwendet werden

Sollen verschiedene Rezepte abgespeichert werden, wie zum Beispiel Maschinenparameter und PID-Parameter, so können zwei verschiedene mapp Recipe Instanzen verwendet werden.

Damit die Rezepte seperat abgespeichert werden können, muss der Eingang "Category" an MpRecipeRegPar verwendet werden.



Beim abspeichern der Maschinenparameter wird beispielsweise die Kategorie "Category1" verwendet und beim abspeichern der PID-Parameter "Category2".

Zu diesem Thema gibt es den Anwendungsfall Maschine mit Produkt- und Maschinenparametern (Kategorie).

Darstellen von Sonderzeichen

Werden Umlaute in der Rezeptdatei nicht korrekt dargestellt und stattdessen als Sonderzeichen oder Rechteck angegeben, sollte folgendes beachtet werden:

Rezeptdateien werden standardmäßig mit einer UTF-8 Codierung abgespeichert. Mithilfe des Datentypen STRING werden ASCII Zeichen korrekt wiedergegeben. Für Zeichen außerhalb des ASCII Zeichensatz ist der WSTRING Datentyp zu verwenden.

Werden also Zeichen durch beispielsweise "?" ersetzt, so wurde versucht ein Unicode Zeichen außerhalb des ASCII Zeichenraums in einen STRING zu schreiben. Als Abhilfe dafür sollte anstelle des STRING Datentyp ein WSTRING verwendet werden

2.6 Bibliotheken

2.6.1 Funktionsbausteine

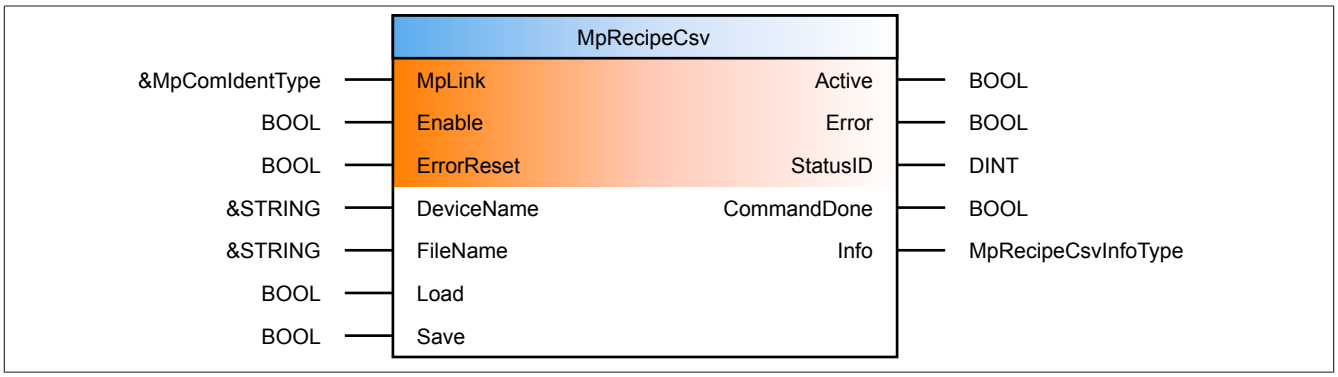
Unterstützt ab	Name	Beschreibung
5.0.0	MpRecipeXml	Dieser Funktionsbaustein bietet die Möglichkeit, Parameter von einer bestimmten Datei zu laden bzw. Parameter in diese Datei zu schreiben (XML Format)
5.0.0	MpRecipeCsv	Dieser Funktionsbaustein bietet die Möglichkeit, Parameter von einer bestimmten Datei zu laden bzw. Parameter in diese Datei zu schreiben (CSV Format)
5.0.0	MpRecipeRegPar	Dieser Funktionsbaustein ermöglicht es, eine PV als Rezept Parameter zu deklarieren
5.0.0	MpRecipeRegParSync	Dieser Funktionsbaustein ermöglicht es, eine PV als Rezept Parameter zu deklarieren und beim Speichern des Rezeptes eine Synchronisation auszuführen
5.0.0	MpRecipeUI	Dieser Funktionsbaustein erstellt eine eine Verbindung zu Visual Components 4 Visualisierungen, um eine Liste möglicher Rezepte anzuzeigen

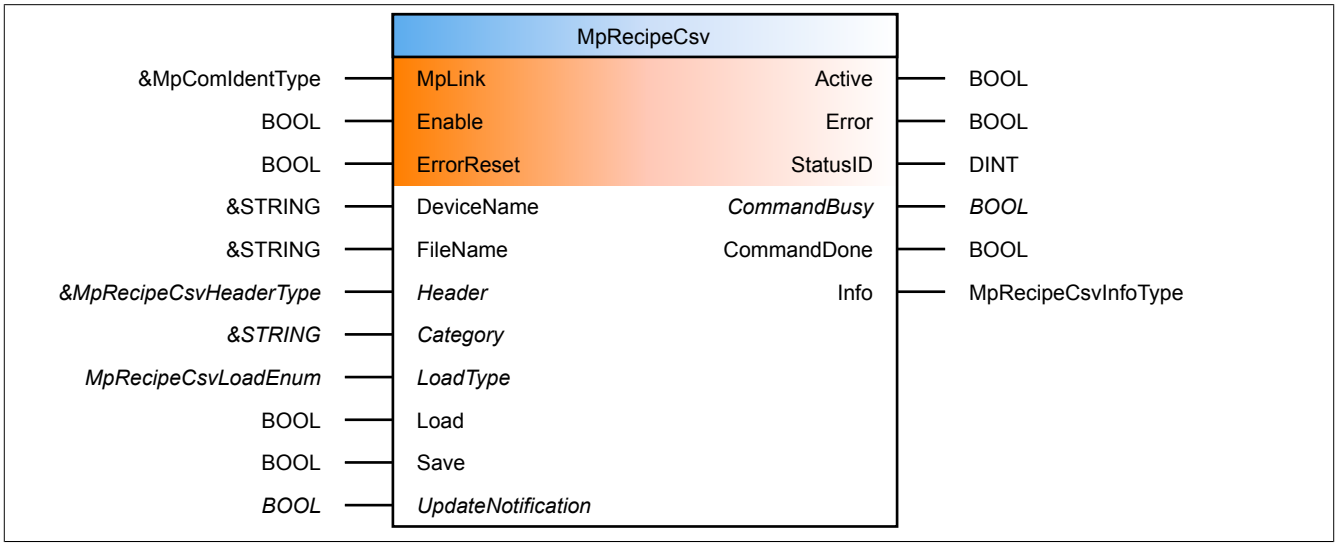
2.6.1.1 MpRecipeCsv

Dieser Funktionsbaustein bietet die Möglichkeit, Parameter von einer bestimmten Datei zu laden bzw. Parameter in diese Datei zu schreiben.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpRecipeCsv Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	DeviceName	`Zeiger auf STRING[50]`	File Device (Speichermedium), auf dem die Dateien abgelegt werden.
IN	FileName	`Zeiger auf STRING[255]`	Name der Datei, welche das Rezept beinhaltet. Die Datei kann auch in einem Ordner auf dem angegebenen Speichermedium abgelegt werden. Dabei muss der Ordnername mitangegeben werden, wie zum Beispiel "Ordnername/FileName" Die Symbole "/", "\", ".", "?" und "*" dürfen nicht im Dateinamen enthalten sein.

I/O	Parameter	Datentyp	Beschreibung
IN	*Header*	Zeiger auf MpRecipeCsvHeaderType	Headerinformation, die entweder geschrieben oder gelesen wird (abhängig vom Befehl). Es wird kein Header in die Datei eingefügt wenn NULL angehängt wird, bzw. wenn die Headerstruktur leer ist.
IN	*Category*	Zeiger auf STRING[50]	Name der Kategorie, die für das Organisieren der Parameter verwendet wird.
IN	*LoadType*	MpRecipeCsvLoadEnum	Definiert, *wie* das Rezept geladen wird. Standardmäßig wird das *gesamte* Rezept geladen.
IN	Load	BOOL	Lädt die Rezeptparameter aus der angegebenen Datei ("FileName")
IN	Save	BOOL	Speichert die aktuellen Rezeptparameter in eine definierte Datei
IN	*UpdateNotification*	BOOL	Eine positive Flanke an diesem Eingang veranlasst, dass alle MpRecipeRegPar und MpRecipeRegParSync an deren Ausgang "UpdateNotification" einen Zyklus lang TRUE sind. Damit ist es möglich die verbundenen Komponenten über neue Rezeptwerte zu informieren.
OUT	Active	BOOL	FB ist aktiv.
OUT	Error	BOOL	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	DINT	Statusinformation.
OUT	*CommandBusy*	BOOL	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	BOOL	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	MpRecipeCsvInfoType	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

2.6.1.1.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpRecipeCsv Konfiguration verwendet.

Die Hauptfunktionen dieses Funktionsbausteins sind:

- **Laden:** Es können Rezeptdaten in die jeweiligen Prozess Variablen geladen werden
- **Speichern:** Die aktuellen Daten der registrierten PVs können in Rezepte geschrieben werden.
  Existiert das angegebene Rezeot noch nicht, so wird ein neues erstellt. Wird keine Endung angegeben bei einem neuen Rezept, so wird automatisch .csv angehängt. Das automatische anhängen der Endung kann in der MpRecipeCsv Konfiguration unter dem Abschnitt *Auto file extension* deaktiviert werden.
  Existieren in einer durch MpComGroup erstellte Gruppe mehrere Instanzen von MpRecipeCsv mit unterschiedlich registrierten PVs, so werden durch "Save=TRUE" alle PVs der Gruppe in Rezepte geschrieben.
- **Versionskontrolle:** Durch das Lesen der "Header" Information kann überprüft werden, welche Version des Rezepts aktuell vorliegt

Die Rezepte, die von dieser mapp Komponente verarbeitet werden, sind CSV - Dateien. Wie die CSV-Datei aufgebaut ist, ist im Abschnitt Aufbau der Rezept-Datei zu finden.

Das Dateiformat CSV ist offen und kann, wenn nicht verschlüsselt, mit einem einfachen Texteditor manipuliert werden.

Es gilt zu beachten, dass beim Setzten eines Kommandos sichergestellt sein muss, dass der Ausgangsparameter "CommandDone = FALSE" ist. Das bedeutet es kann immer nur ein Kommando aktiv ("TRUE") sein!

Beispiel:

Soll ein Rezept geladen werden ("Load = TRUE"), muss das Kommando "Save = FALSE" sein.

Beim Arbeiten auf Linux-Systemen kann auf keine Systemordner mit Zugangsbeschränkung zugegriffen werden. Dies kann zum Beispiel der "lost+found" Ordner sein. Wird versucht auf so einen Ordner zuzugreifen, wird der Fehler "Ungültiges File Device" ausgelöst. Betroffen sind alle Linux Geräte mit EXT2/3/4 Dateisystemen zb. MP71xx oder T30.

```
C:\Users\kratzern>dir \\10.43.37.3\usershare
 Volume in drive \\10.43.37.3\usershare is usershare
 Volume Serial Number is 9325-608F

 Directory of \\10.43.37.3\usershare

12/12/2017  07:42 PM    <DIR>          .
01/19/2018  02:01 PM    <DIR>          ..
12/12/2017  07:27 PM    <DIR>          TestDir
01/01/1970  02:00 AM    <DIR>          lost+found
               0 File(s)              0 bytes
               4 Dir(s)     473,661,440 bytes free
```

Das Laden eines Rezepts

Bevor ein Befehl zum Laden gegeben wird, müssen ein gültiges Speichermedium ("DeviceName") und ein gültiges Rezept ("FileName") übergeben werden. Danach muss über den Eingang "LoadType" ausgewählt werden, ob das gesamte Rezept oder nur die Headerinformation geladen wird. Falls eine Struktur am Eingang "Header" (MpRecipeCsvHeaderType) angeschlossen ist, werden dort die Headerinformationen des Rezepts abgelegt. Die möglichen Modi von "LoadType" sind:

- **mpRECIPE_FILE_LOAD_HEADER:** Es wird nur der Inhalt des Rezeptheaders gelesen und in die obige Struktur geschrieben. Falls keine Variable am Eingang "Header" verbunden ist, wird nichts geschrieben. Dieser Modus ist speziell für die Versionskontrolle hilfreich (siehe weiter unten)
- **mpRECIPE_FILE_LOAD_ALL:** Der gesamte Inhalt des Rezepts wird ausgelesen. Die Headerinformationen werden in die Struktur am Eingang "Header" (falls vorhanden) geschrieben, die Rezeptdaten in die jeweiligen PVs

Mit dem Befehl "Load" werden die Daten werden aus dem Rezept gelesen. Das erfolgreiche Laden wird durch "CommandDone = `TRUE`" bestätigt.

Wird ein Rezept geladen, welche Parameter beinhaltet die nicht registriert sind, werden nur jene Parameter geladen welche auch aktuell registriert sind. Nehmen wir an, wir laden das Rezept "Kaffee", welche die Parameter Zucker, Milch, Kaffeepulver und Wasser beinhaltet. Aktuell sind aber nur die Parameter Zucker, Milch und Kaffeepulver registriert, somit wird der Parameter Wasser nicht geladen. Es scheint die Warnung mpRECIPE_WRN_LOAD_WITH_WARN.

Wird aber ein Rezept geladen, welches nicht alle aktuell registrierten Parameter beinhaltet, bleiben die Parameter welche nicht neu geladen werden unverändert. Es wird also das Rezept "Kaffee" mit den Parametern Zucker, Milch und Kaffeepulver geladen. Aktuell sind aber die Parameter Zucker, Milch, Kaffeepulver und Wasser registriert. Somit bleibt der Wert, welcher aktuell am Parameter Wasser war, bestehen. Es scheint ebenfalls die Warnung mpRECIPE_WRN_LOAD_WITH_WARN.

Wird eine CSV Datei geladen, welches ein anderes Trennzeichen (";" anstatt ",") verwendet, so kann die Datei nicht korrekt gelesen werden. Das Rezept kann somit nicht geladen werden.

Nach dem Laden kann allen MpRecipeRegPar und MpRecipeRegParSync Funktionsbausteinen mitgeteilt werden, dass neue Parameter aus einem Rezept geladen wurden. Dieser Befehl muss manuell am Eingang "UpdateNotification" aktiviert werden. Dieser Vorgang ist dann von Nutzen, wenn der aktuelle Prozess nicht direkt mit den Rezeptdaten betrieben wird, sondern diese erst übernommen werden müssen. Das kann geschehen wenn neue Daten geladen wurden und vom Benutzer das Signal "UpdateNotification" weitergegeben wird.

Wie das Kommando "UpdateNotification" verwendet werden kann, ist im Anwendungsfall Maschine mit neuen Rezeptdaten sicher aktualisieren erklärt.

Das Speichern eines Rezepts

Bei "Save = `TRUE`", werden die aktuellen PV Daten in das Rezept geschrieben. Dieser Vorgang verläuft unterschiedlich für die Funktionsbausteine MpRecipeRegPar und MpRecipeRegParSync:

- **MpRecipeRegPar:** Im Fall, dass eine Prozess Variable über diesen Funktionsbaustein registriert wurde, werden die neuen Daten bei einem Speicher - Befehl direkt in dem Rezept abgelegt
- **MpRecipeRegParSync:** Bei Verwendung dieses Funktionsbausteins muss das Übertragen der Daten von der PV ins Rezept bestätigt werden. Bei "Save = `TRUE`" kann man an allen MpRecipeRegParSync Funktionsbausteinen den Status "SaveNotification = `TRUE`" finden. Dieser zeigt an, dass die aktuellen PV Daten in das Rezept geschrieben werden können. Zuvor muss das jedoch quittiert werden. Über den Eingang "ConfirmSave" am MpRecipeRegParSync wird das Speichern bestätigt und erst dann werden die Daten wirklich von der PV ins Rezept übertragen. Damit hat der Benutzer hier die Möglichkeit, bevor die Daten gespeichert werden, mögliche Dateninkonsistenz zu verhindern bzw. die Daten noch zu beeinflussen. Solange die Daten durch "SaveNotification" nicht bestätigt werden ist kein neuer Save-Befehl möglich. Das heißt während eines Speichervorgangs kann "Save" deaktiviert und wieder aktiviert werden. Dies wirkt sich nicht auf den Speichervorgang aus. Erst wenn "SaveNotification = TRUE" ist, kann ein neuer Speichervorgang gestartet werden.

Wenn der Name der Datei ("FileName") keine Endung hat (kein Punkt im Namen), so wird automatisch *.csv* angehängt. Das automatische anhängen der Endung kann in der MpRecipeCsv Konfiguration unter dem Abschnitt *Auto file extension* deaktiviert werden.

Sobald das Speichern erfolgreich war, wird der Ausgang "CommandDone = `TRUE`". Falls wegen aktiver MpRecipeRegParSync Funktionsbausteinen das Speichern verzögert wird (es kommt keine Bestätigung durch "ConfirmSave"), bleibt der Befehl "Save" aktiv, was durch "CommandBusy = `TRUE`" ersichtlich ist. Erst wenn der Prozess von allen Teilnehmern bestätigt wurde, werden die PV Daten gespeichert.

In der MpRecipeCsv Konfiguration kann das automatische Speichern durch "Save automatically" aktiviert werden. Dabei muss ein Intervall zum Abspeichern festgelegt werden. Über "Start immediately" wird definiert, ob beim aktivieren von MpRecipeCsv das Rezept sofort abgespeichert werden soll.

Beim Verwenden des gleichen Rezeptnamens, wird das Rezept auf dem Speichermedium automatisch überschrieben.

Tritt beim Speichern eines Rezepts ein Fehler auf, so kann detailierte Information immer im Logger gefunden werden.

Beim Verwenden von Sonderzeichen in einer Rezeptdatei, sollten die genannten Punkte im Abschnitt FAQ berücksichtigt werden.

Wird während des Speichervorgangs die Stromzufuhr der Steuerung unterbrochen (Stromausfall), so kann es sein, dass das Rezept nicht gespeichert wird. Wird eine Rezeptdatei durch den Speichervorgang überschrieben (gleicher Rezeptname), so wird die Ursprungs-Datei gelöscht und das neu abzuspeichernde Rezept wird gespeichert. Bei einem Stromausfall kann es also passieren, dass das vorhergehende Rezept gelöscht wird und das neue Rezept nicht rechzeitig abgespeichert wird.

Versionskontrolle

Mithilfe der Eingänge "Header" und "LoadType" kann eine Versionskontrolle eingeführt werden. Da man sich entscheiden kann, nur den Header eines Rezepts zu laden, ist es möglich, herauszufinden welche Version des Rezepts vorliegt. Danach kann ein anderes Rezept geladen oder das Laden des kompletten Rezepts durchgeführt werden. Für mehr Informationen siehe Abschnitt Aufbau einer Rezept-Datei.

Rezeptkategorien

Über den Eingang "Category" können unterschiedliche PVs in mehrere Rezepte aufgeteilt werden. Wird die gleiche Kategorie beim Registrieren einer PV (MpRecipeRegPar oder MpRecipeRegParSync) und beim Verwalten eines Rezeptes (MpRecipeCsv) angegeben, so wird der PV Wert in das Rezept mit derselben Kategorie abgelegt. Mehr Infos dazu unter Aufzeichnen verschiedener Rezepte.

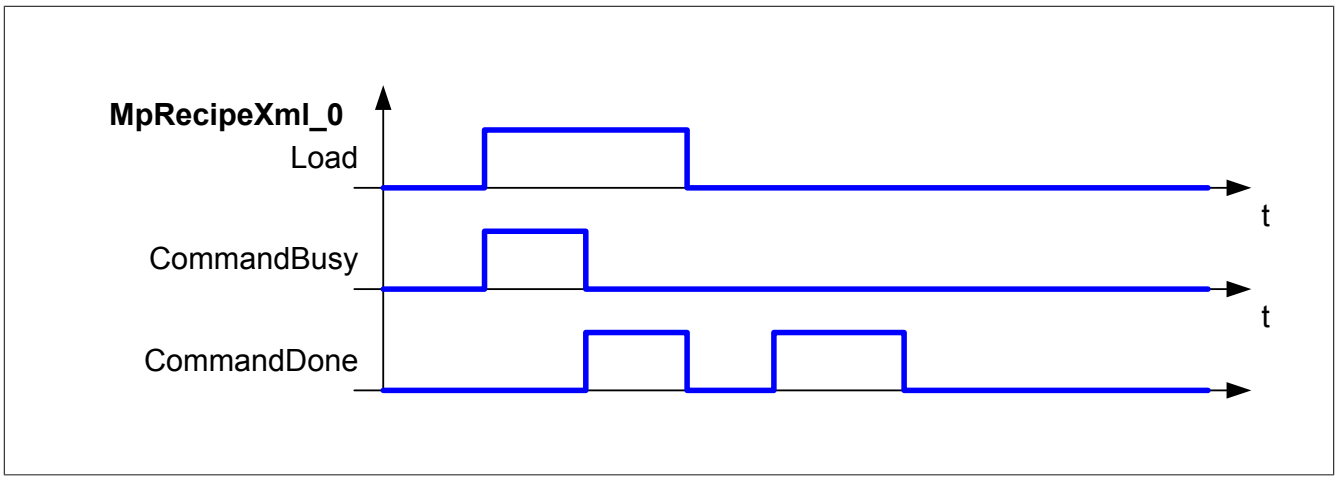
> **Für jede PV ist ein eigener MpRecipeRegPar oder MpRecipeRegParSync Funktionsbaustein notwendig.**

2.6.1.1.2 Zeitdiagramme

Die Zeitdiagramme für das Verwalten der Rezepte behandeln das Speichern und Laden sowie die Synchronisation der Parameter.

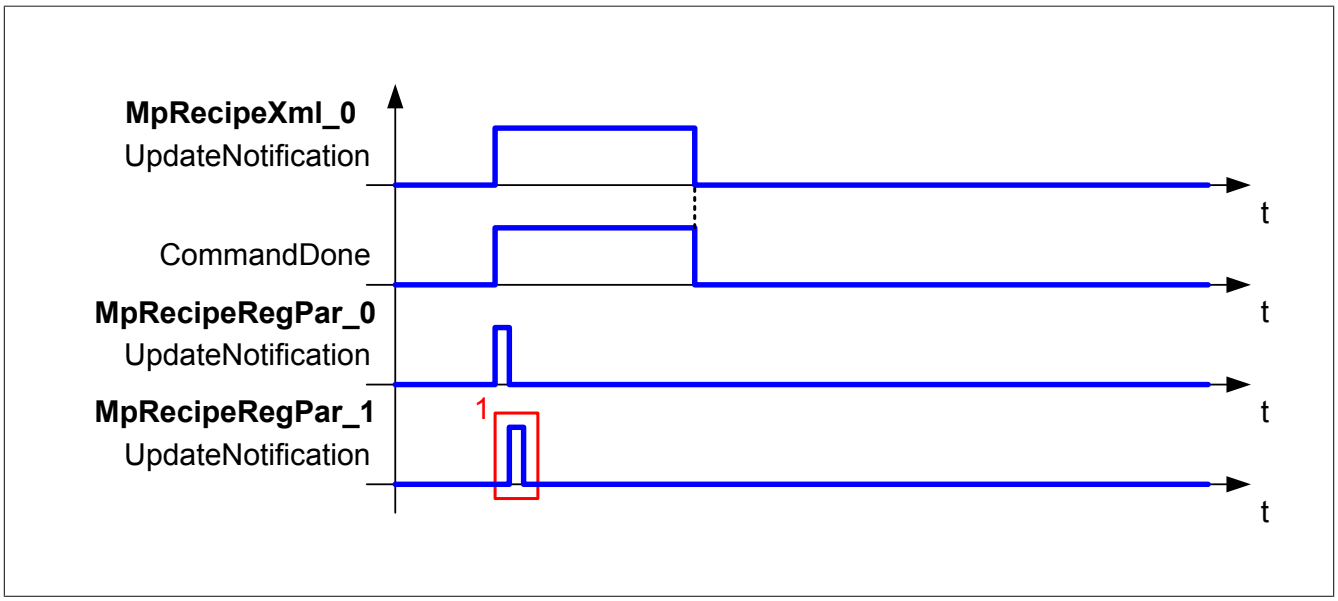
Laden eines Rezepts

Hier wird zuerst das Laden eines Rezepts gezeigt. Danach wird mithilfe des Befehls "UpdateNotification" (welcher durch die Applikation ausgelöst wird[1]) angezeigt, dass neue Werte in die PVs geschrieben wurden. Diese Ausgänge sind nur für einen Zyklus aktiv und werden dann automatisch zurückgesetzt. In diesem Fall wird das Signal des Funktionsbausteins *MpRecipeRegPar_1* verzögert, da es in einem Task mit niedrigerer Zykluszeit läuft[2].



Funktionsweise einer "UpdateNotification"

Der Befehl "UpdateNotification" wird am MpRecipeCsv ausgelöst und erzeugt an allen verbundenen MpRecipeRegPar Funktionsbausteinen ein Statussignal "UpdateNotification", welches zeigt, dass neue Werte in die PVs der geschrieben wurden. Der Ausgang ist für nur einen Zyklus aktiv und wird dann automatisch zurückgesetzt. In diesem Beispiel wird das Signal des Funktionsbausteins *MpRecipeRegPar_1* verzögert, da es in einem Task mit niedrigerer Zykluszeit läuft[1].

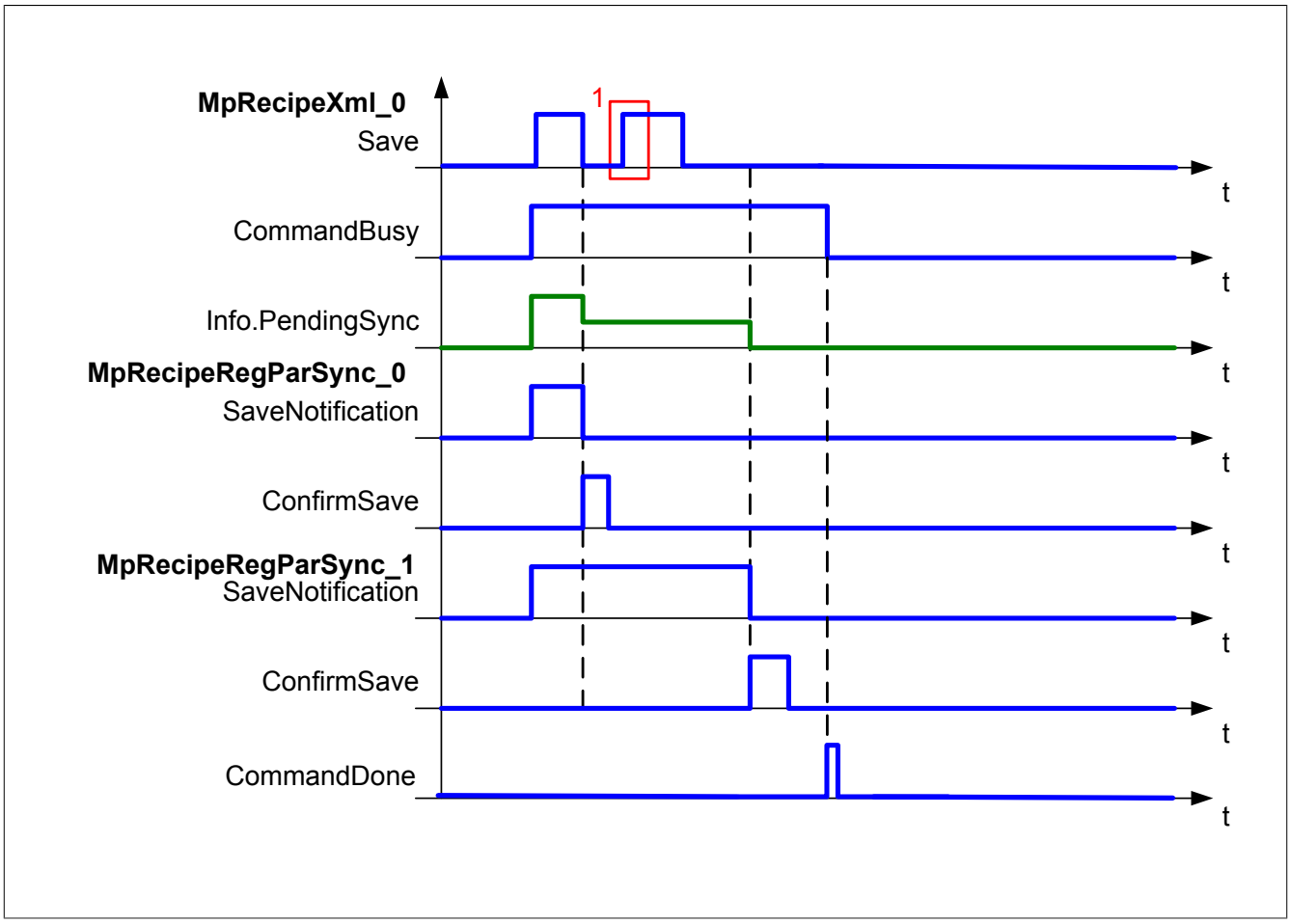


Speichern eines Rezepts

Speichern eines Rezepts mit MpRecipeRegParSync

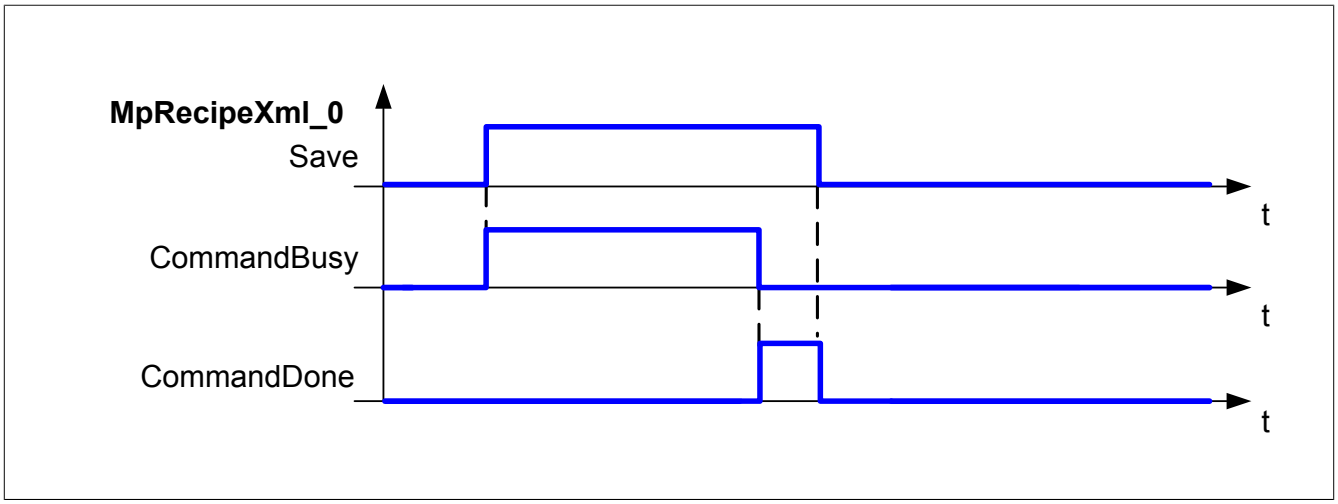
Wird "Save = `TRUE`" am MpRecipeCsv Funktionsbaustein, wird ein Speichervorgang gestartet. In diesem Beispiel werden einige MpRecipeRegParSync Funktionsbausteine verwendet und daher wird das Rezept nicht gespeichert ohne eine Bestätigung an diesen Funktionsbausteinen. Sobald der Speicherbefehl ankommt, werden die Status "SaveNotification" gesetzt, welche anzeigen, dass die aktuellen Parameter in das Rezept abgespeichert werden sollen. Wird dieser Status durch den Befehl "ConfirmSave" bestätigt, werden die Parameter übertragen.

Solange die Daten durch "SaveNotification" nicht bestätigt werden ist kein neuer Save-Befehl möglich. Das heißt während eines Speichervorgangs kann "Save" deaktiviert und wieder aktiviert werden. Dies wirkt sich nicht auf den Speichervorgang aus[1]. Erst wenn "SaveNotification = TRUE" ist, kann ein neuer Speichervorgang gestartet werden.



Speichern eines Rezepts mit MpRecipeRegPar

Wird "Save = `TRUE`" am MpRecipeCsv Funktionsbaustein, wird ein Speichervorgang gestartet. In diesem Beispiel wird ein MpRecipeRegPar Funktionsbaustein verwendet. Sobald der Speicherbefehl kommt, wird das Rezept am angegebenen Speichermedium gespeichert. "CommandDone = `FALSE`" sobald "Save = `FALSE`" ist.



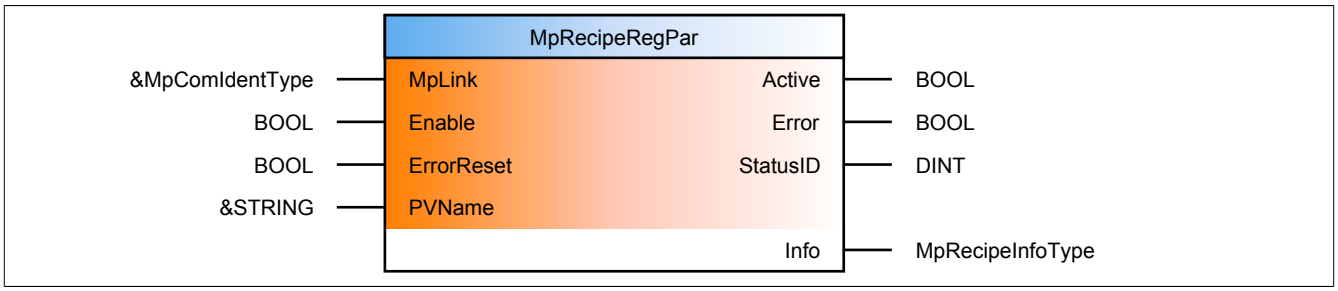
2.6.1.2 MpRecipeRegPar

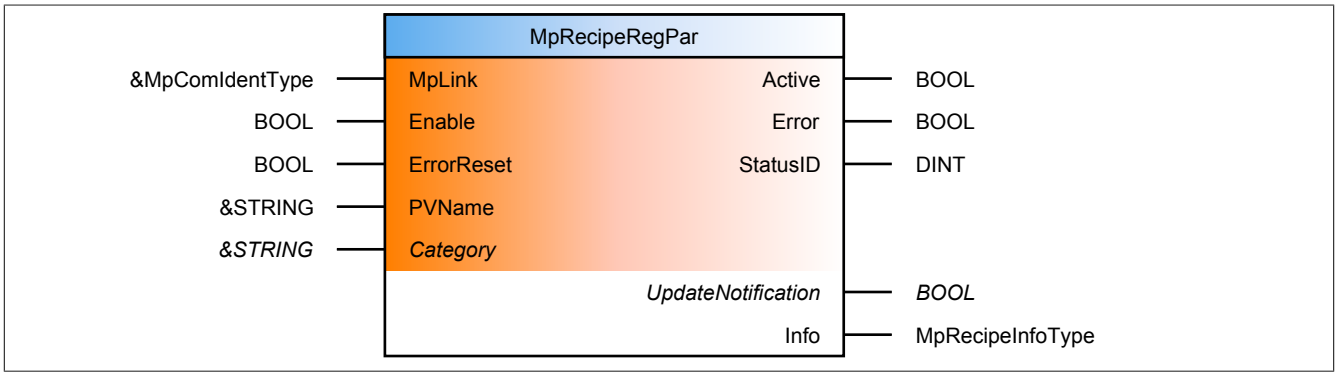
Dieser Funktionsbaustein ermöglicht es dem Benutzer eine PV als Rezept Parameter zu registrieren.

> **Abgeleitete Datentypen, Arrays welche nicht mit Index 0 starten und Mehrdimensionale Arrays können erst ab der AR Version D4.26 bzw. AR 4.34 registriert werden.**

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpRecipeCsv/MpRecipeXml Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	PVName	`Zeiger auf STRING[100]`	Name der PV, welche ins Rezept gespeichert werden soll.
IN	*Category*	`Zeiger auf STRING[50]`	Name der Kategorie, die für das Organisieren der Parameter benützt wird.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*UpdateNotification*	`BOOL`	`TRUE`, bei einer positiven Flanke des Eingangs "UpdateNotification" von MpRecipeXml oder MpRecipeCsv. Wird automatisch nach einem Zyklus zurückgesetzt.
OUT	Info	`MpRecipeInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

2.6.1.2.1 Beschreibung

Es wird der Mplink von einer MpRecipeXml / MpRecipeCsv Konfiguration verwendet.

Mithilfe dieses Funktionsbausteins wird eine Prozess Variable in die Rezeptverwaltung eingebunden. Wird eine PV registriert, wird deren Inhalt beim Laden und Speichern von Rezepten berücksichtigt. Ob die PV gültig ist, wird bei einer positiven Flanke am "Enable" Eingang geprüft. Sobald die Komponente aktiv ist ("Active" = `TRUE`), ist die PV registriert. Ist die Komponente nicht mehr aktiv, so ist auch die PV nicht mehr registriert. Für jede PV, welche im Rezept sein soll, muss ein eigener MpRecipeRegPar oder MpRecipeRegParSync verwendet werden.

Das Speichern und Laden erfolgt bei diesem Funktionsbaustein unverzüglich. Wird durch einen der Funktionsbausteine MpRecipeXml oder MpRecipeCsv eine "UpdateNotification" ausgesendet, so wird dies am Ausgang "UpdateNotification" ersichtlich.

Über den Eingang "Category" kann die PV einem bestimmten Rezept zugeordnet werden (siehe Aufzeichnen verschiedener Rezepte).

Eine neue PV kann jederzeit registriert werden (Angabe des Namens von Prozessvariablen). Dabei wird folgendes Format verwendet:

- **Globale PV:** <PV_name> (z.B. gTestVariable, gControl.Parameter.Distance)
- **Lokale PV:** <task_name>:<PV_name> (z.B. Task1:LokaleTestVariable, AxisCtrl:Basic.Command.MoveVelocity)

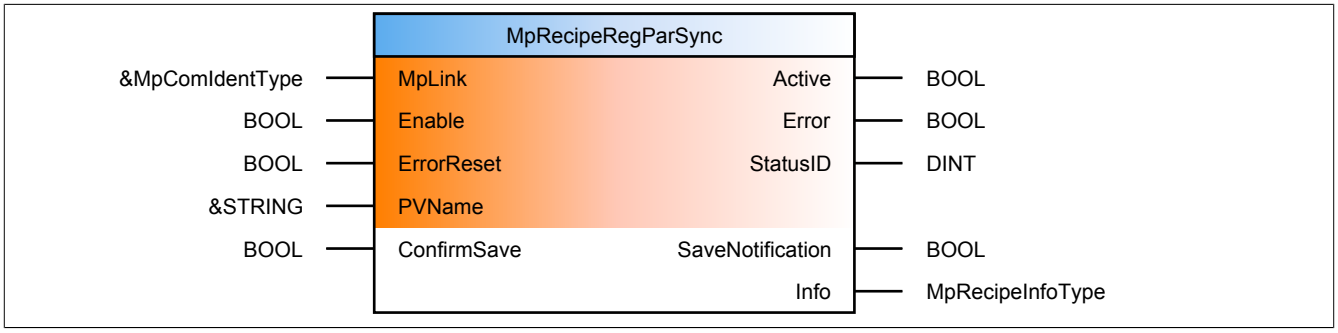
2.6.1.3 MpRecipeRegParSync

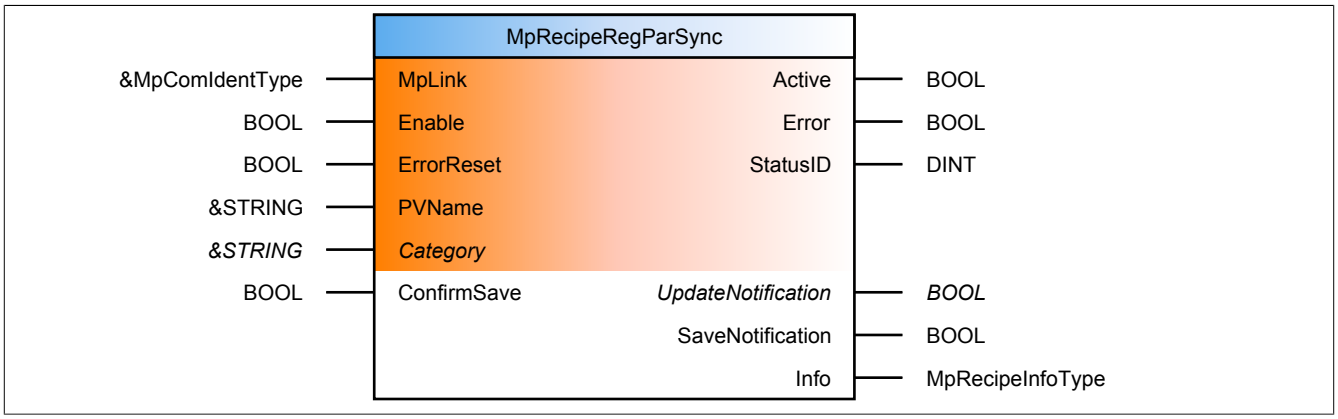
Dieser Funktionsbaustein ermöglicht es, eine PV als Rezept Parameter zu registrieren und beim Speichern des Rezeptes eine Synchronisation auszuführen.

> **Abgeleitete Datentypen, Arrays welche nicht mit Index 0 starten und Mehrdimensionale Arrays können erst ab der AR Version D4.26 bzw. AR 4.34 registriert werden.**

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpRecipeCsv/MpRecipeXml Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	PVName	`Zeiger auf STRING[100]`	Name der PV, welche ins Rezept gespeichert werden soll.
IN	*Category*	`Zeiger auf STRING[50]`	Name der Kategorie, die für das Organisieren der Parameter benützt wird.
IN	ConfirmSave	`BOOL`	Bestätigt, dass die aktuellen Daten der registrierten PV ins Rezept abgespeichert werden können. Benützt in Verbindung mit "SaveNotification".
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*UpdateNotification*	`BOOL`	`TRUE`, wenn ein neues Rezept geladen wurde und die registrierte PV neue Werte enthält. Wird automatisch nach einem Zyklus zurückgesetzt.
OUT	SaveNotification	`BOOL`	TRUE, wenn ein Rezept gespeichert wird und neue Daten ins Rezept kopiert werden sollten, aber bevor die Daten wirklich ins Rezept übertragen werden. Benützt in Verbindung mit "ConfirmSave".
OUT	Info	`MpRecipeInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

2.6.1.3.1 Beschreibung

Es wird der Mplink von einer MpRecipeXml / MpRecipeCsv Konfiguration verwendet.

Mithilfe dieses Funktionsbausteins wird eine Prozess Variable in die Rezeptverwaltung eingebunden. Wird eine PV registriert, wird deren Inhalt beim Laden und Speichern von Rezepten berücksichtigt. Ob die PV gültig ist, wird bei einer positiven Flanke am "Enable" Eingang geprüft. Sobald die Komponente aktiv ist ("Active" = `TRUE`), ist die PV registriert. Ist die Komponente nicht mehr aktiv, so ist auch die PV nicht mehr registriert. Für jede PV, welche im Rezept sein soll, muss ein eigener MpRecipeRegPar oder MpRecipeRegParSync verwendet werden.

Das Laden erfolgt bei diesem Funktionsbaustein unverzüglich. Wird durch einen der Funktionsbausteine MpRecipeXml oder MpRecipeUI eine "UpdateNotification" ausgesendet, so wird dies am Ausgang "UpdateNotification" ersichtlich.

Das Speichern wird verzögert, um die Daten vor dem Abspeichern ins Rezept noch beeinflussen zu können. Wird "Save = `TRUE`", setzt diese Komponente den Status "SaveNotification" am Ausgang. Daraufhin muss über den Eingang "ConfirmSave" bestätigt werden, dass die Daten der PV ins Rezept übertragen werden sollen.

Über den Eingang "Category" kann die PV einem bestimmten Rezept zugeordnet werden (siehe Aufzeichnen verschiedener Rezepte).

Eine neue PV kann jederzeit registriert werden (Angabe des Namens von Prozessvariablen). Dabei wird folgendes Format verwendet:

- **Globale PV:** <PV_name> (z.B. gTestVariable, gControl.Parameter.Distance)
- **Lokale PV:** <task_name>:<PV_name> (z.B. Task1:LokaleTestVariable, AxisCtrl:Basic.Command.MoveVelocity)

Wie der Funktionsbaustein verwendet werden kann, ist im Anwendungsfall Rezeptverwaltung mit Synchronisierung beim Speichern erklärt.

2.6.1.4 MpRecipeUI

Dieser Funktionsbaustein erstellt eine Verbindung zu Visual Components 4 Visualisierungen, um eine Liste möglicher Rezepte anzuzeigen.

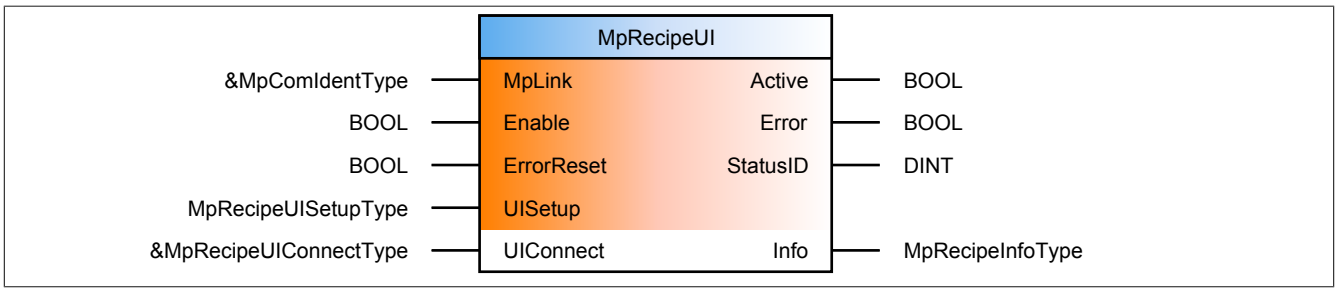
Information:

Es ist nötig die Bibliothek MpFile auf die Steuerung zu übertragen, um diesen Funktionsbaustein nützen zu können.

Auf unserem Youtube Kanal sind die Tutorials mapp Recipe Part 1 und mapp Recipe Part 2 welche erklären wie ein Rezeptmanagement implementiert werden kann.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Recipe (GER/ENG) zu finden. (Zugangsdaten)

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpRecipeCsv/MpRecipeXml Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	UISetup	`MpRecipeUISetupType`	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
IN	UIConnect	`Zeiger auf MpRecipeUIConnectType`	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpRecipeInfoType`	Zusätzliche Funktionsbausteins-Information.

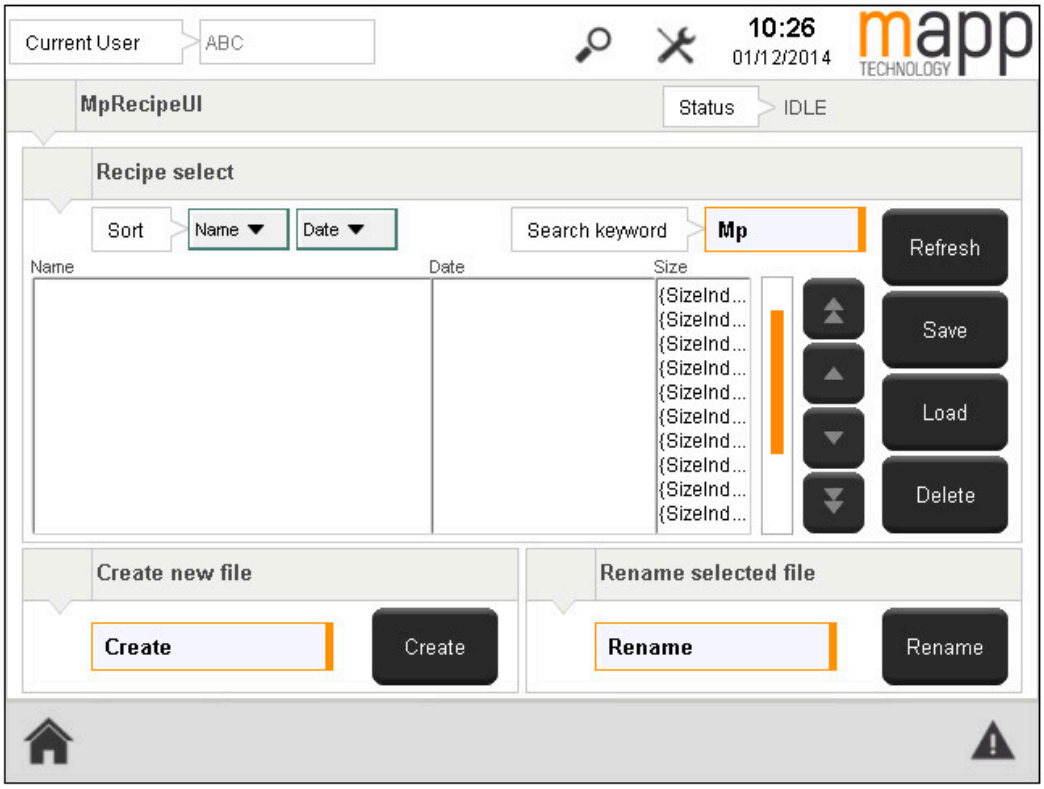
mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

2.6.1.4.1 Beschreibung

Für diesen Funktionsbaustein wird derselbe MpLink verwendet, wie für MpRecipeXml. Diese Komponente stellt eine Verbindung zwischen Rezeptverwaltung und einer Visual Components 4 Visualisierung dar. Der Austausch der Daten zwischen MpRecipeUI und der Visualisierung erfolgt über die Struktur MpRecipeUIConnectType.

Pro MpRecipeXml kann nur ein MpRecipeUI verwendet werden. Es ist nicht möglich mehrere MpRecipeUI-Instanzen zu verwenden.



MpRecipeUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

- **Status:** Der "Status" Parameter gibt Auskunft über die aktuelle Aktivität der Rezeptverwaltung
- **Recipe:** In der Struktur MpRecipeUIRecipeType ist eine Liste mit verfügabren Rezepten zu finden. Das Speichermedium (File Device), von dem die Rezepte gelesen werden, wird über den Eingang "DeviceName" an MpRecipeXml bestimmt. Die Liste kann sortiert und gefiltert werden, um die Suche nach einem Rezept zu vereinfachen. Auch das Laden, Speichern und Löschen erfolgt über diese Struktur. Nach dem Löschen eines Rezepts wird die Rezeptliste automatisch erneuert. Weiters wird über diese Struktur der Header des aktuell ausgewählten Rezepts direkt angezeigt, damit können die Beschreibung sowie die Version des Rezepts angezeigt werden ohne das gesamte Rezept zu laden
- **New:** Über die Struktur MpRecipeUINewType kann ein neues Rezept erstellt werden. Nach dem Erstellen eines Rezepts wird die Rezeptliste automatisch erneuert.

Suchfunktion

Mit dem Parameter "Filter" im Datentyp MpRecipeUIRecipeType kann nach bestimmten Dateien gesucht werden. Es ist möglich in der Suchfunktion das Platzhaltersymbol "*" zu verwenden. Wird als Filter, zum Beispiel "*Test*.xml" verwendet, können die Dateien "Test01.xml", "MyTest.xml" oder "MyTest01.xml" angezeigt werden. Bei Nichtbenützung des Platzhalters "*" wird direkt nach der Datei "Test.xml" gesucht.

Wird kein Filter angegeben oder nur das Symbol "*", werden alle Dateien und Ordner angezeigt die vorhanden sind.

MpRecipeUISetupType

In dieser Struktur wird über "RecipeListSize" definiert, wieviele Einträge der Dateiliste auf einer Seite der Visualisierung angezeigt werden. Über "RecipeListScrollWindow" wird entscheiden wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden. Mit dem Parameter "Confirmation" können unterschiedliche Dialogfenster ein- bzw. ausgeschaltet werden.

> **Es ist nötig die Bibliothek MpFile auf die Steuerung zu übertragen, um diesen Funktionsbaustein nützen zu können.**

2.6.1.5 MpRecipeXml

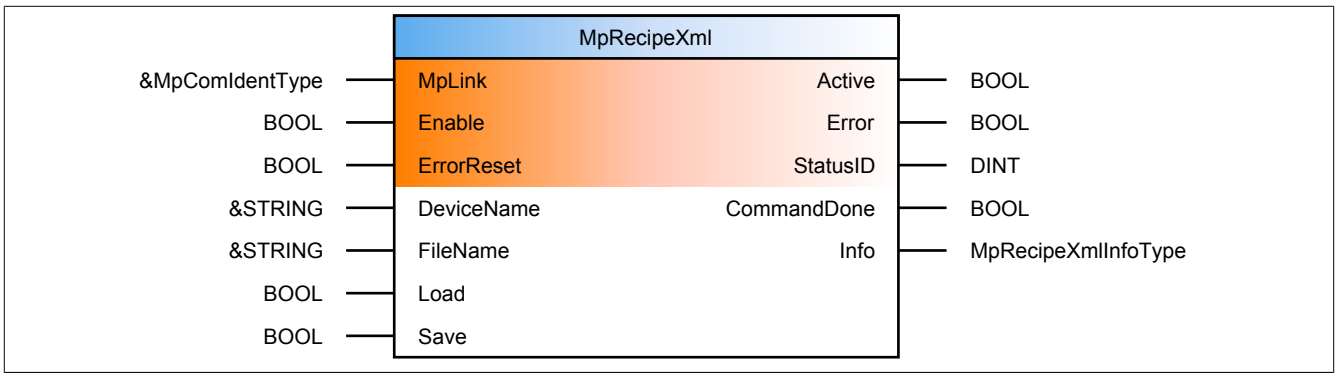
Dieser Funktionsbaustein bietet die Möglichkeit, Parameter von einer bestimmten Datei zu laden bzw. Parameter in diese Datei zu schreiben.

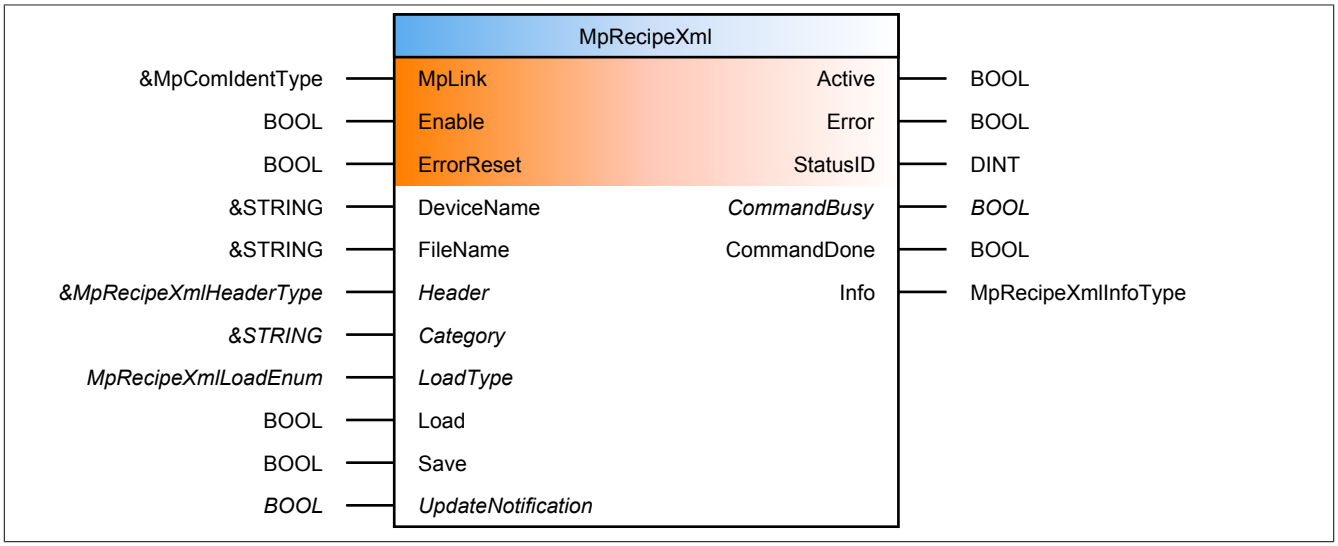
Auf unserem Youtube Kanal sind die Tutorials mapp Recipe Part 1 und mapp Recipe Part 2 welche erklären wie ein Rezeptmanagement implementiert werden kann.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Recipe (GER/ENG) zu finden. (Zugangsdaten)

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpRecipeXml Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	DeviceName	`Zeiger auf STRING[50]`	File Device (Speichermedium), auf dem die Dateien abgelegt werden.
IN	FileName	`Zeiger auf STRING[255]`	Name der Datei, welche das Rezept beinhaltet. Die Datei kann auch in einem Ordner auf dem angegebenen Speichermedium abgelegt werden. Dabei muss der Ordnername mitangegeben werden, wie zum Beispiel "Ordnername/FileName" Die Symbole "/", "\", ".", "?" und "*" dürfen nicht verwendet werden.

I/O	Parameter	Datentyp	Beschreibung
IN	*Header*	Zeiger auf MpRecipeXmlHeaderType	Headerinformation, die entweder geschrieben oder gelesen wird (abhängig vom Befehl). Es wird kein Header in die Datei eingefügt wenn NULL angehängt wird, bzw. wenn die Headerstruktur leer ist.
IN	*Category*	Zeiger auf STRING[50]	Name der Kategorie, die für das Organisieren der Parameter verwendet wird.
IN	*LoadType*	MpRecipeXmlLoadEnum	Definiert, *wie* das Rezept geladen wird. Standardmäßig wird das *gesamte* Rezept geladen.
IN	Load	BOOL	Lädt die Rezeptparameter aus der angegebenen Datei ("FileName").
IN	Save	BOOL	Speichert die aktuellen Rezeptparameter in eine definierte Datei.
IN	*UpdateNotification*	BOOL	Eine positive Flanke an diesem Eingang veranlasst, dass alle MpRecipeRegPar und MpRecipeRegParSync an deren Ausgang "UpdateNotification" einen Zyklus lang TRUE sind. Damit ist es möglich die verbundenen Komponenten über neue Rezeptwerte zu informieren.
OUT	Active	BOOL	FB ist aktiv.
OUT	Error	BOOL	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	DINT	Statusinformation.
OUT	*CommandBusy*	BOOL	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	BOOL	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	MpRecipeXmlInfoType	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

2.6.1.5.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpRecipeXml Konfiguration verwendet.

Die Hauptfunktionen dieses Funktionsbausteins sind:

- **Laden:** Es können Rezeptdaten in die jeweiligen Prozess Variablen geladen werden
- **Speichern:** Die aktuellen Daten der registrierten PVs können in Rezepte geschrieben werden. Existiert das angegebene Rezept noch nicht, so wird ein neues erstellt. Wird keine Endung angegeben bei einem neuen Rezept, so wird automatisch .xml angehängt. Das automatische anhängen der Endung kann in der MpRecipeXml Konfiguration unter dem Abschnitt *Auto file extension* deaktiviert werden.
  Existieren in einer durch MpComGroup erstellte Gruppe mehrere Instanzen von MpRecipeXml mit unterschiedlich registrierten PVs, so werden durch "Save=TRUE" alle PVs der Gruppe in Rezepte geschrieben.
- **Versionskontrolle:** Durch das Lesen der "Header" Information kann überprüft werden, welche Version des Rezepts aktuell vorliegt

Die Rezepte, die von dieser mapp Komponente verarbeitet werden, sind XML - Dateien. Wie die XML-Datei aufgebaut ist, ist im Abschnitt Aufbau der Rezept-Datei zu finden.

Das Dateiformat XML ist offen und kann, wenn nicht verschlüsselt, mit einem einfachen Texteditor manipuliert werden.

Es gilt zu beachten, dass beim Setzten eines Kommandos sichergestellt sein muss, dass der Ausgangsparameter "CommandDone = FALSE" ist. Das bedeutet es kann immer nur ein Kommando aktiv ("TRUE") sein!

Beispiel:

Soll ein Rezept geladen werden ("Load = TRUE"), muss das Kommando "Save = FALSE" sein.

Beim Arbeiten auf Linux-Systemen kann auf keine Systemordner mit Zugangsbeschränkung zugegriffen werden. Dies kann zum Beispiel der "lost+found" Ordner sein. Wird versucht auf so einen Ordner zuzugreifen, wird der Fehler "Ungültiges File Device" ausgelöst. Betroffen sind alle Linux Geräte mit EXT2/3/4 Dateisystemen zb. MP71xx oder T30.

```
C:\Users\kratzern>dir \\10.43.37.3\usershare
 Volume in drive \\10.43.37.3\usershare is usershare
 Volume Serial Number is 9325-608F

 Directory of \\10.43.37.3\usershare

12/12/2017  07:42 PM    <DIR>          .
01/19/2018  02:01 PM    <DIR>          ..
12/12/2017  07:27 PM    <DIR>          TestDir
01/01/1970  02:00 AM    <DIR>          lost+found
               0 File(s)              0 bytes
               4 Dir(s)     473,661,440 bytes free
```

Das Laden eines Rezepts

Bevor ein Befehl zum Laden gegeben wird, müssen ein gültiges Speichermedium ("DeviceName") und ein gültiges Rezept ("FileName") übergeben werden. Danach muss über den Eingang "LoadType" ausgewählt werden, ob das gesamte Rezept oder nur die Headerinformation geladen wird. Falls eine Struktur am Eingang "Header" (MpRecipeXmlHeaderType) angeschlossen ist, werden dort die Headerinformationen des Rezepts abgelegt. Die möglichen Modi von "LoadType" sind:

- **mpRECIPE_FILE_LOAD_HEADER:** Es wird nur der Inhalt des Rezeptheaders gelesen und in die obige Struktur geschrieben. Falls keine Variable am Eingang "Header" verbunden ist, wird nichts geschrieben. Dieser Modus ist speziell für die Versionskontrolle hilfreich (siehe weiter unten)
- **mpRECIPE_FILE_LOAD_ALL:** Der gesamte Inhalt des Rezepts wird ausgelesen. Die Headerinformationen werden in die Struktur am Eingang "Header" (falls vorhanden) geschrieben, die Rezeptdaten in die jeweiligen PVs

Mit dem Befehl "Load" werden die Daten werden aus dem Rezept gelesen. Das erfolgreiche Laden wird durch "CommandDone = `TRUE`" bestätigt.

Wird ein Rezept geladen, welche Parameter beinhaltet die nicht registriert sind, werden nur jene Parameter geladen welche auch aktuell registriert sind. Nehmen wir an, wir laden das Rezept "Kaffee", welche die Parameter Zucker, Milch, Kaffeepulver und Wasser beinhaltet. Aktuell sind aber nur die Parameter Zucker, Milch und Kaffeepulver registriert, somit wird der Parameter Wasser nicht geladen. Es scheint die Warnung mpRECIPE_WRN_LOAD_WITH_WARN.

Wird aber ein Rezept geladen, welches nicht alle aktuell registrierten Parameter beinhaltet, bleiben die Parameter welche nicht neu geladen werden unverändert. Es wird also das Rezept "Kaffee" mit den Parametern Zucker, Milch und Kaffeepulver geladen. Aktuell sind aber die Parameter Zucker, Milch, Kaffeepulver und Wasser registriert. Somit bleibt der Wert, welcher aktuell am Parameter Wasser war, bestehen. Es erscheint ebenfalls die Warnung mpRECIPE_WRN_LOAD_WITH_WARN.

Wird ein Rezept geladen, welches beispielsweise extern bearbeitet wurde und dadurch die XML-Struktur beschädigt wurde, so kann das Rezept nicht geladen werden. Es erscheint der Fehler mpRECIPE_ERR_LOAD_WITH_ERRORS.

Nach dem Laden kann allen MpRecipeRegPar und MpRecipeRegParSync Funktionsbausteinen mitgeteilt werden, dass neue Parameter aus einem Rezept geladen wurden. Dieser Befehl muss manuell am Eingang "UpdateNotification" aktiviert werden. Dieser Vorgang ist dann von Nutzen, wenn der aktuelle Prozess nicht direkt mit den Rezeptdaten betrieben wird, sondern diese erst übernommen werden müssen. Das kann geschehen wenn neue Daten geladen wurden und vom Benutzer das Signal "UpdateNotification" weitergegeben wird.

Wie das Kommando "UpdateNotification" verwendet werden kann, ist im Anwendungsfall Maschine mit neuen Rezeptdaten sicher aktualisieren erklärt.

Das Speichern eines Rezepts

Bei "Save = `TRUE`", werden die aktuellen PV Daten in das Rezept geschrieben. Dieser Vorgang verläuft unterschiedlich für die Funktionsbausteine MpRecipeRegPar und MpRecipeRegParSync:

- **MpRecipeRegPar:** Im Fall, dass eine Prozess Variable über diesen Funktionsbaustein registriert wurde, werden die neuen Daten bei einem Speicher - Befehl direkt in dem Rezept abgelegt
- **MpRecipeRegParSync:** Bei Verwendung dieses Funktionsbausteins muss das Übertragen der Daten von der PV ins Rezept bestätigt werden. Bei "Save = `TRUE`" kann man an allen MpRecipeRegParSync Funktionsbausteinen den Status "SaveNotification = `TRUE`" finden. Dieser zeigt an, dass die aktuellen PV Daten in das Rezept geschrieben werden können. Zuvor muss das jedoch quittiert werden. Über den Eingang "ConfirmSave" am MpRecipeRegParSync wird das Speichern bestätigt und erst dann werden die Daten wirklich von der PV ins Rezept übertragen. Damit hat der Benutzer hier die Möglichkeit, bevor die Daten gespeichert werden, mögliche Dateninkonsistenz zu verhindern bzw. die Daten noch zu beeinflussen. Solange die Daten durch "SaveNotification" nicht bestätigt werden ist kein neuer Save-Befehl möglich. Das heißt während eines Speichervorgangs kann "Save" deaktiviert und wieder aktiviert werden. Dies wirkt sich nicht auf den Speichervorgang aus. Erst wenn "SaveNotification = `TRUE`" ist, kann ein neuer Speichervorgang gestartet werden.

Wenn der Name der Datei ("FileName") keine Endung hat (kein Punkt im Namen), so wird automatisch *.xml* angehängt. Das automatische anhängen der Endung kann in der MpRecipeXml Konfiguration unter dem Abschnitt *Auto file extension* deaktiviert werden.

Sobald das Speichern erfolgreich war, wird der Ausgang "CommandDone = `TRUE`". Falls wegen aktiver MpRecipeRegParSync Funktionsbausteine das Speichern verzögert wird (es kommt keine Bestätigung durch "ConfirmSave"), bleibt der Befehl "Save" aktiv, was durch "CommandBusy = `TRUE`" ersichtlich ist. Erst wenn der Prozess von allen Teilnehmern bestätigt wurde, werden die PV Daten gespeichert.

In der MpRecipeXml Konfiguration kann das automatische Speichern durch "Save automatically" aktiviert werden. Dabei muss ein Intervall zum Abspeichern festgelegt werden. Über "Start immediately" wird definiert, ob beim aktivieren von MpRecipeXml das Rezept sofort abgespeichert werden soll.

Beim Verwenden des gleichen Rezeptnamens, wird das Rezept auf dem Speichermedium automatisch überschrieben.

Tritt beim Speichern eines Rezepts ein Fehler auf, so kann detailierte Information immer im Logger gefunden werden.

Beim Verwenden von Sonderzeichen in einer Rezeptdatei, sollten die genannten Punkte im Abschnitt FAQ berücksichtigt werden.

Wird während des Speichervorgangs die Stromzufuhr der Steuerung unterbrochen (Stromausfall), so kann es sein, dass das Rezept nicht gespeichert wird. Wird eine Rezeptdatei durch den Speichervorgang überschrieben (gleicher Rezeptname), so wird die Ursprungs-Datei gelöscht und das neu abzuspeichernde Rezept wird gespeichert. Bei einem Stromausfall kann es also passieren, dass das vorhergehende Rezept gelöscht wird und das neue Rezept nicht rechzeitig abgespeichert wird.

Versionskontrolle

Mithilfe der Eingänge "Header" und "LoadType" kann eine Versionskontrolle eingeführt werden. Da man sich entscheiden kann, nur den Header eines Rezepts zu laden, ist es möglich, herauszufinden welche Version des Rezepts vorliegt. Danach kann ein anderes Rezept geladen oder das Laden des kompletten Rezepts durchgeführt werden. Für mehr Information siehe Abschnitt Aufbau der Rezept-Datei.

Rezeptkategorien

Über den Eingang "Category" können unterschiedliche PVs in mehrere Rezepte aufgeteilt werden. Wird die gleiche Kategorie beim Registrieren einer PV (MpRecipeRegPar oder MpRecipeRegParSync) und beim Verwalten eines Rezeptes (MpRecipeXml) angegeben, so wird der PV Wert in das Rezept mit derselben Kategorie abgelegt. Mehr Infos dazu unter Aufzeichnen verschiedener Rezepte.

Wie die Rezeptkategorie verwendet werden kann, ist im Anwendungsfall Maschine mit Produkt- und Maschinenparametern erklärt.

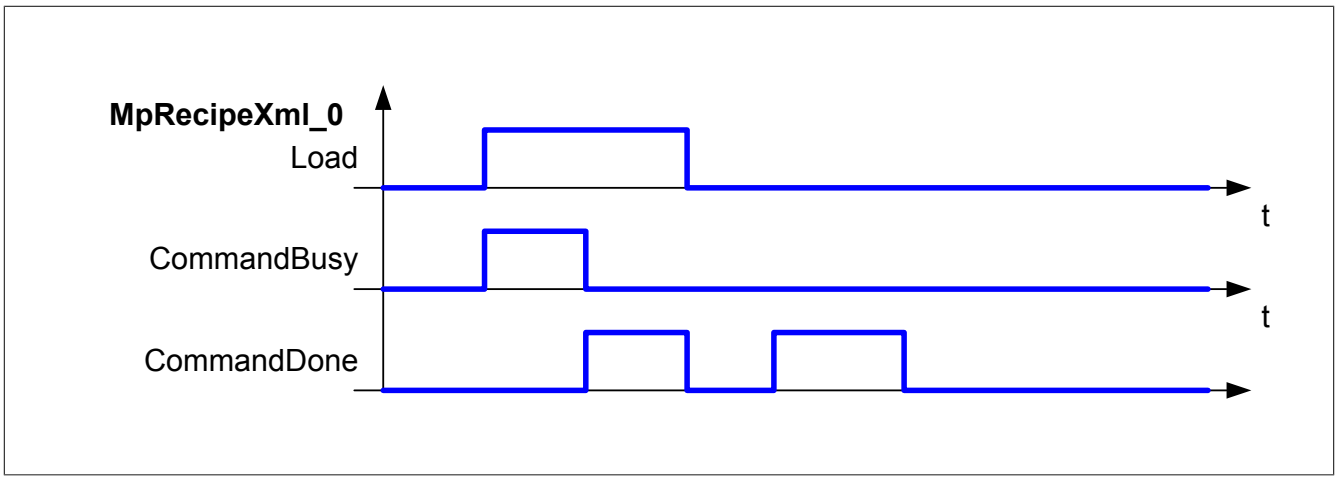
Für jede PV ist ein eigener MpRecipeRegPar oder MpRecipeRegParSync Funktionsbaustein notwendig.

2.6.1.5.2 Zeitdiagramme

Die Zeitdiagramme für das Verwalten der Rezepte behandeln das Speichern und Laden sowie die Synchronisation der Parameter.

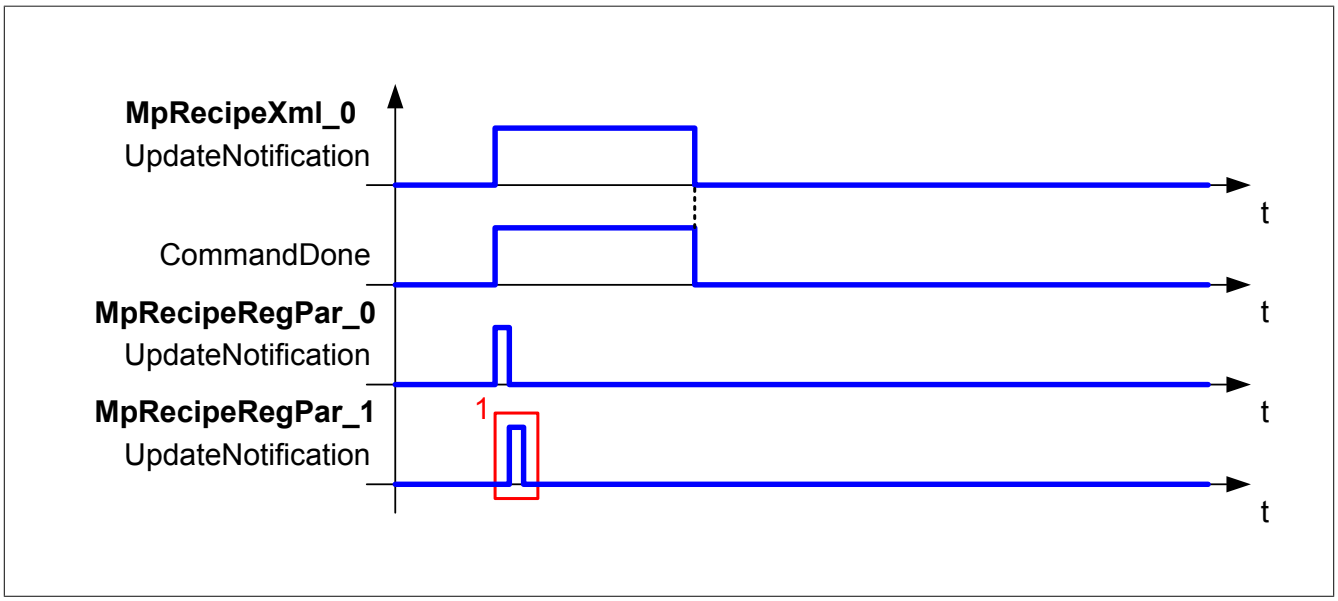
Laden eines Rezepts

Hier wird zuerst das Laden eines Rezepts gezeigt. Danach wird mithilfe des Befehls "UpdateNotification" (welcher durch die Applikation ausgelöst wird[1]) angezeigt, dass neue Werte in die PVs geschrieben wurden. Diese Ausgänge sind nur für einen Zyklus aktiv und werden dann automatisch zurückgesetzt. In diesem Fall wird das Signal des Funktionsbausteins *MpRecipeRegPar_1* verzögert, da es in einem Task mit niedrigerer Zykluszeit läuft[2].



Funktionsweise einer "UpdateNotification"

Der Befehl "UpdateNotification" wird am MpRecipeXml ausgelöst und erzeugt an allen verbundenen MpRecipeRegPar Funktionsbausteinen ein Statussignal "UpdateNotification", welches zeigt, dass neue Werte in die PVs der geschrieben wurden. Der Ausgang ist für nur einen Zyklus aktiv und wird dann automatisch zurückgesetzt. In diesem Beispiel wird das Signal des Funktionsbausteins *MpRecipeRegPar_1* verzögert, da es in einem Task mit niedrigerer Zykluszeit läuft[1].

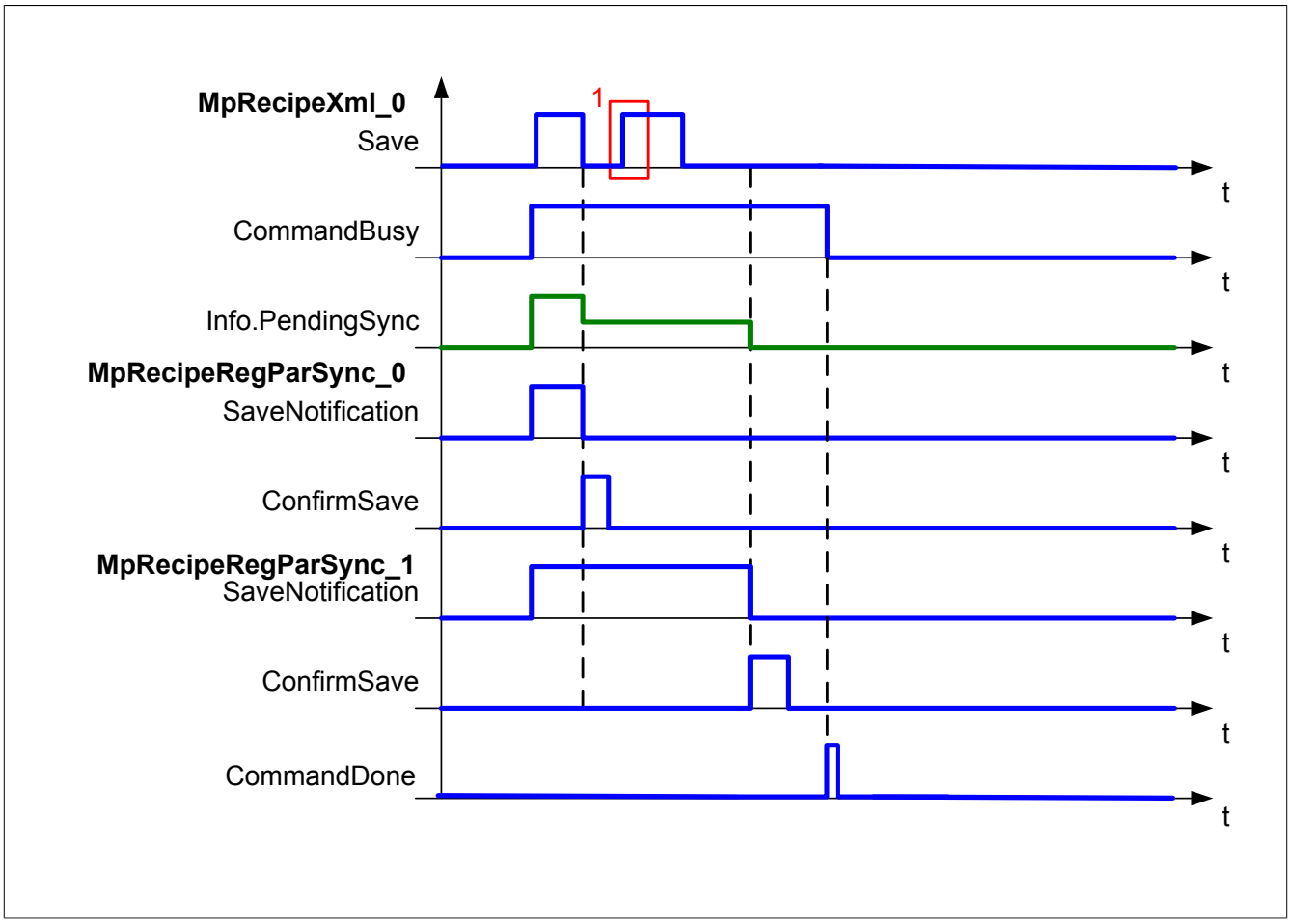


Speichern eines Rezepts

Speichern eines Rezepts mit MpRecipeRegParSync

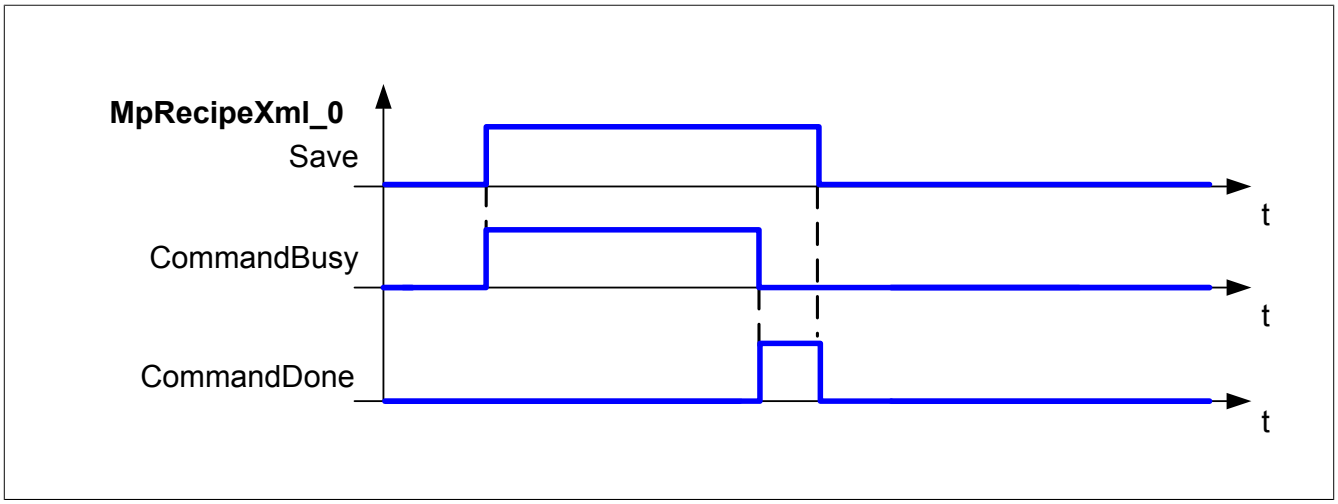
Wird "Save = `TRUE`" am MpRecipeXml Funktionsbaustein, wird ein Speichervorgang gestartet. In diesem Beispiel werden einige MpRecipeRegParSync Funktionsbausteine verwendet und daher wird das Rezept nicht gespeichert ohne eine Bestätigung an diesen Funktionsbausteinen. Sobald der Speicherbefehl ankommt, werden die Status "SaveNotification" gesetzt, welche anzeigen, dass die aktuellen Parameter in das Rezept abgespeichert werden sollen. Wird dieser Status durch den Befehl "ConfirmSave" bestätigt, werden die Parameter übertragen.

Solange die Daten durch "SaveNotification" nicht bestätigt werden ist kein neuer Save-Befehl möglich. Das heißt während eines Speichervorgangs kann "Save" deaktiviert und wieder aktiviert werden. Dies wirkt sich nicht auf den Speichervorgang aus[1]. Erst wenn "SaveNotification = TRUE" ist, kann ein neuer Speichervorgang gestartet werden.



Speichern eines Rezepts mit MpRecipeRegPar

Wird "Save = `TRUE`" am MpRecipeXml Funktionsbaustein, wird ein Speichervorgang gestartet. In diesem Beispiel wird ein MpRecipeRegPar Funktionsbaustein verwendet. Sobald der Speicherbefehl kommt, wird das Rezept am angegebenen Speichermedium gespeichert. "CommandDone = `FALSE`" sobald "Save = `FALSE`" ist.



2.6.2 Datentypen und Enumeratoren

2.6.2.1 Datentypen

2.6.2.1.1 MpRecipeCsvHeaderType

In dieser Struktur wird die Header Information für Rezepte abgespeichert. Die Header Struktur wird beim Erzeugen / Speichern / Lesen mithilfe der MpRecipeCsv Komponente verwendet.

Parameter	Datentyp	Beschreibung
Name	STRING[100]	Der Name des Rezepts
Description	STRING[255]	Kommentar, welcher das Rezept beschreibt
Version	STRING[20]	Version des Rezepts
DateTime	DATE_AND_TIME	Das Datum und die Uhrzeit, wann das Rezept erstellt wurde

2.6.2.1.2 MpRecipeCsvInfoType

Mit diesem Datentyp wird für die MpRecipeCsv Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
FileSize	UDINT	Die Größe des Rezepts, das auf dem jeweiligen Speichermedium abgelegt ist
PendingSync	UDINT	Die Nummer der anstehenden Speicher-Benachrichtigungen ("SaveNotification"), welche noch nicht bestätigt wurden
LastLoadedRecipe	STRING[255]	Das letzte geladene Rezept
Diag	MpRecipeDiagType	Eine Diagnosestruktur für die StatusID

2.6.2.1.3 MpRecipeDiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	MpRecipeStatusIDType	StatusID Diagnose Struktur

2.6.2.1.4 MpRecipeInfoType

Mit diesem Datentyp wird für die MpRecipe Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpRecipeDiagType	Diagnosestruktur für die StatusID

2.6.2.1.5 MpRecipeStatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	MpRecipeErrorEnum	Fehlerkode der mapp Komponente
Severity	MpComSeveritiesEnum	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler)
Code	UINT	Kode der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden

2.6.2.1.6 MpRecipeUIConnectType

Mithilfe dieses Datentypes wird eine Verbindung zwischen der MpRecipeUI Komponente und der Visual Components 4 Visualisierung erstellt. Dazu gehören eine Liste an möglichen Rezepten, sowie die Möglichkeit neue zu Erstellen.

Parameter	Datentyp	Beschreibung
Status	MpRecipeUIStatusEnum	Aktuelle Operation
Recipe	MpRecipeUIRecipeType	Zugriff auf bereits existierende Rezepte
New	MpRecipeUINewType	Zum Erstellen von neuen Rezepten
MessageBox	MpRecipeUIMessageBoxType	Bedienung der Dialogfenster
DefaultLayerStatus	UINT	Statusdatenpunkt für den Default Layer der Visualisierungsseite, auf der die Rezeptverwaltung angezeigt wird **VC4 Anbindung:** StatusDatapoint von *Layer*

2.6.2.1.7 MpRecipeUIHeaderType

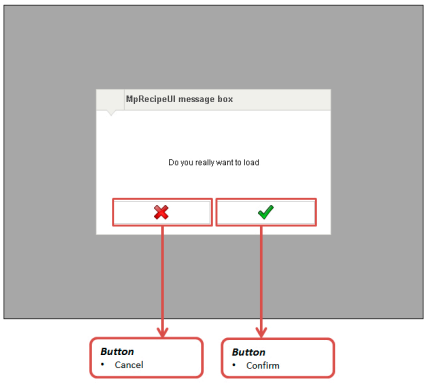
In dieser Struktur wird die Header Information von Rezepten angezeigt. Die Header Struktur wird automatisch geladen, wenn ein Rezept über die Visualisierung ausgewählt wird.

Parameter	Datentyp	Beschreibung
Name	STRING[100]	Der Name des Rezepts
Description	STRING[255]	Kommentar, welcher das Rezept beschreibt
Version	STRING[20]	Version des Rezepts
DateTime	DATE_AND_TIME	Das Datum und die Uhrzeit, wann das Rezept erstellt wurde

2.6.2.1.8 MpRecipeUIMessageBoxType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern.

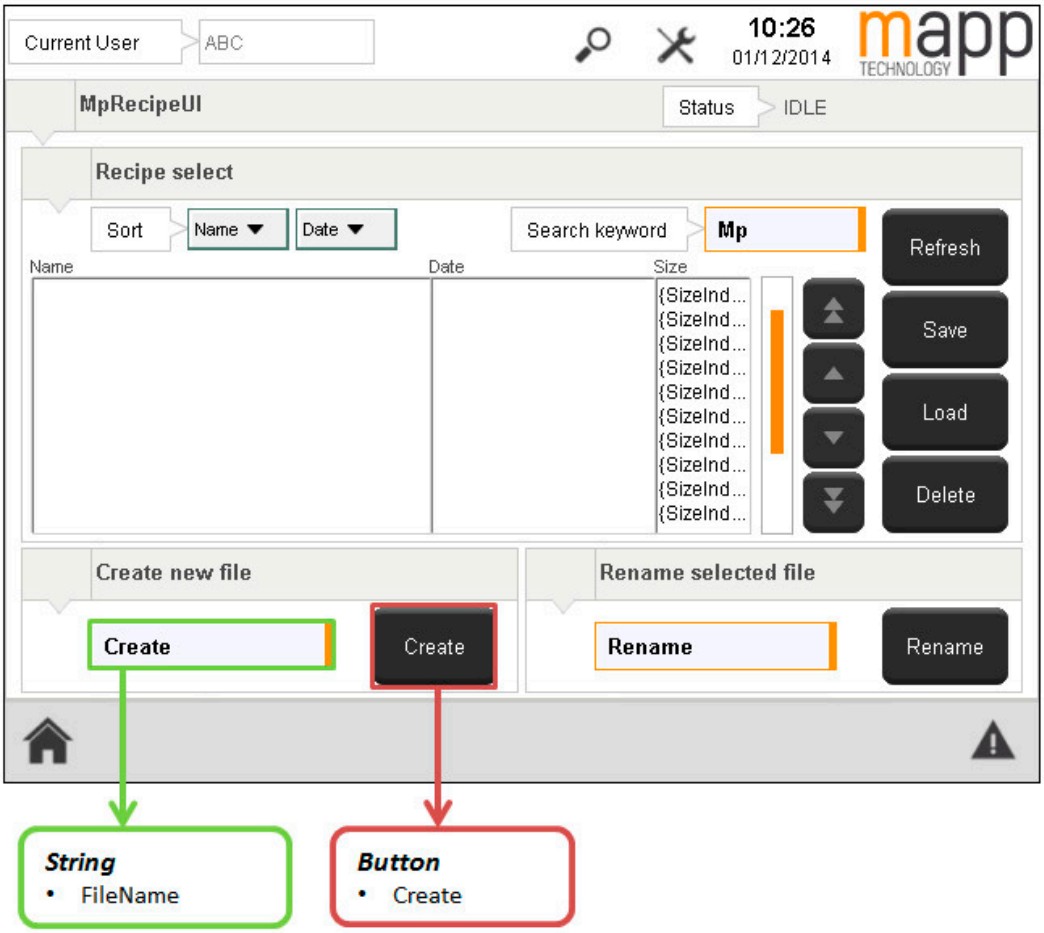
Parameter	Datentyp	Beschreibung
LayerStatus	UINT	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Type	MpRecipeUIMessageEnum	Art des Dialogfensters
Confirm	BOOL	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	BOOL	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1



2.6.2.1.9 MpRecipeUINewType

Hier findet man alle nötigen Punkte, um ein neues Rezept zu erstellen. Nach dem Erstellen eines Rezepts wird die Rezeptliste automatisch erneuert. Wird keine Endung angegeben bei einem Rezept, so wird automatisch .xml angehängt.

Parameter	Datentyp	Beschreibung
FileName	STRING[255]	Name des neuen Rezepts **VC4 Anbindung:** Datapoint von *String*
Create	BOOL	Erstellt ein neues Rezept **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0

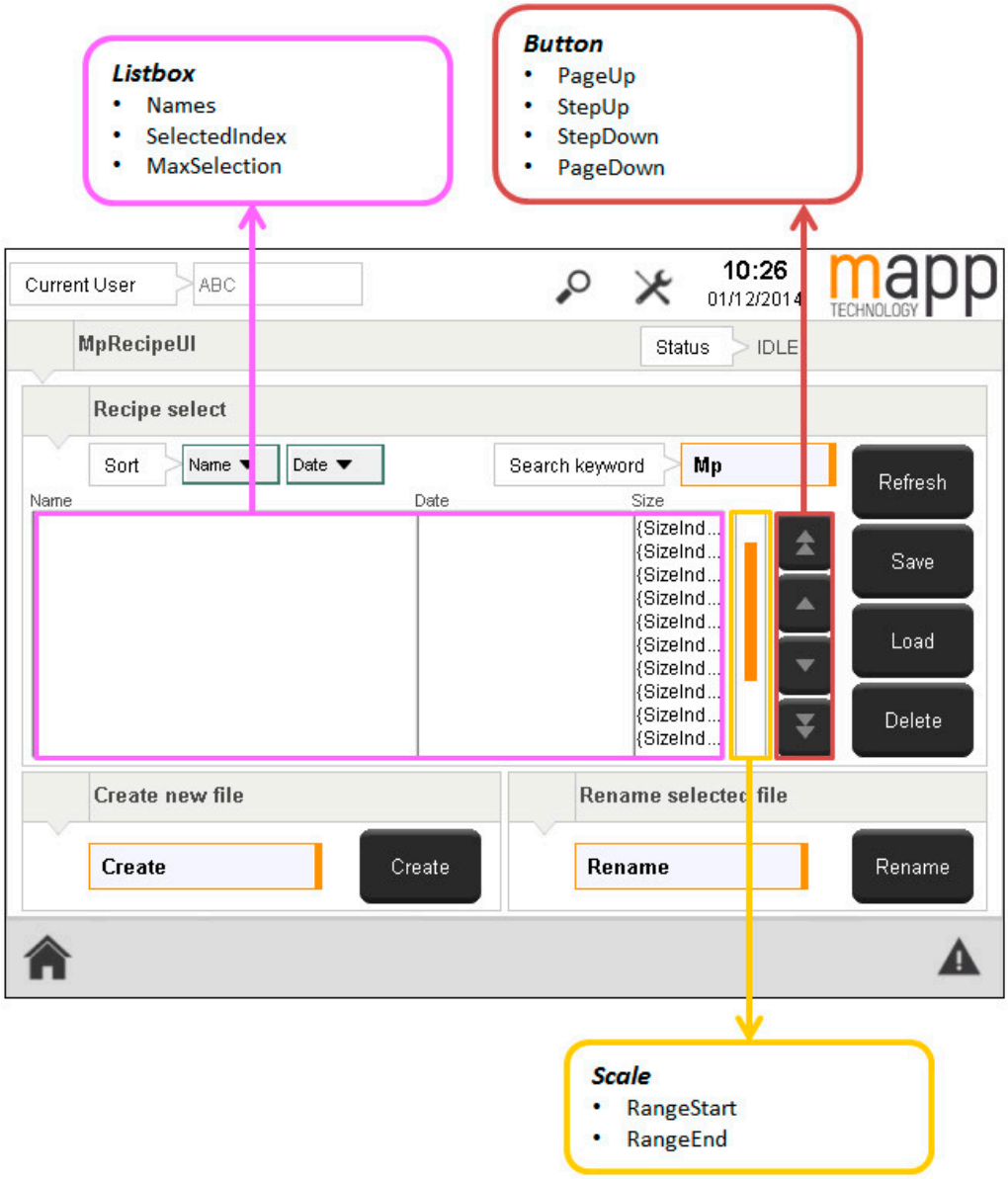


2.6.2.1.10 MpRecipeUIRecipeListType

Liste mit möglichen Rezepten und die dazugehörige Navigation.

Parameter	Datentyp	Beschreibung
Names	`ARRAY[0..19] von STRING[255]`	Liste aller verfügbaren Rezepte (Dateinamen) **VC4 Anbindung:** TextDatapoint von *Listbox*
SelectedIndex	`UINT`	Index des aktuell ausgewählten Eintrages in der Liste **VC4 Anbindung:** IndexDatapoint von *Listbox*
MaxSelection	`UINT`	Index des letzten Eintrages in der Liste **VC4 Anbindung:** MaxDatapoint von *Listbox*
PageUp	`BOOL`	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. Die Größe der Seite wird über den Parameter "RecipeListSize" der Struktur MpRecipeUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
PageDown	`BOOL`	Springt ans Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. Die Größe der Seite wird über den Parameter "RecipeListSize" der Struktur MpRecipeUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepUp	`BOOL`	Wählt vorhergehenden Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepDown	`BOOL`	Wählt nächsten Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0

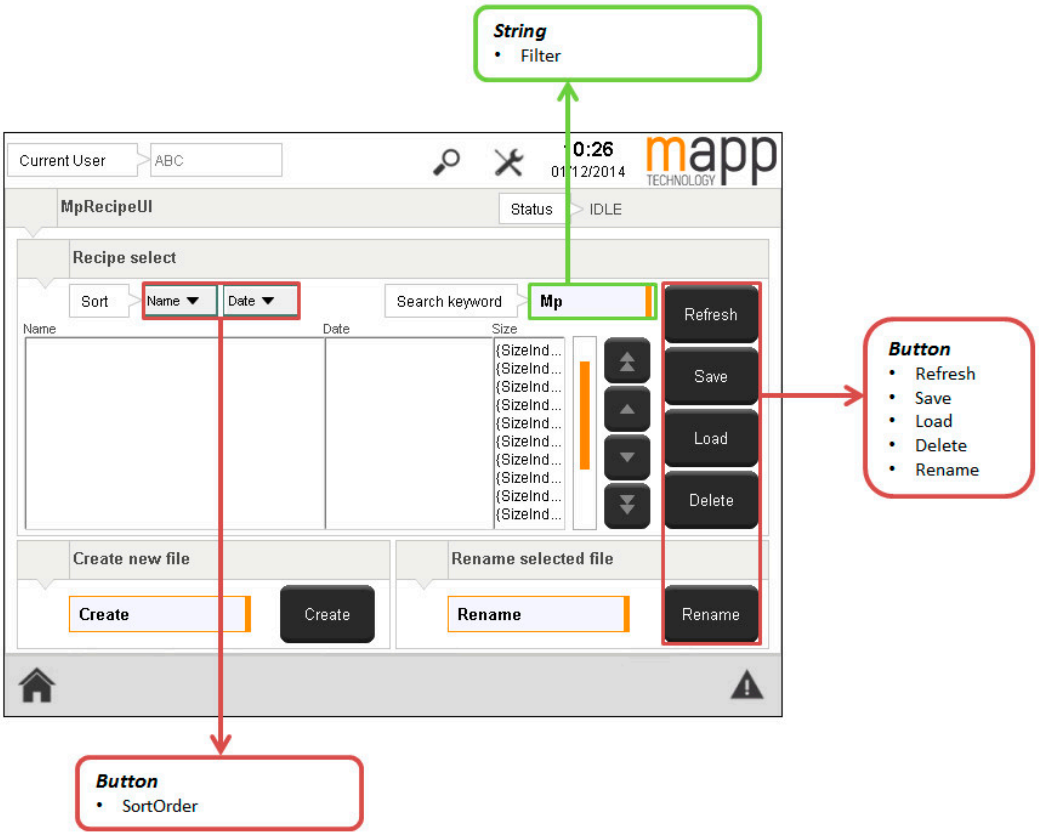
Parameter	Datentyp	Beschreibung
RangeStart	`REAL`	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von *Scale*
RangeEnd	`REAL`	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von *Scale*
Sizes	`ARRAY[0..19] von UDINT`	Größe der Dateien
LastModified	`ARRAY[0..19] von STRING[50]`	Zeitpunkt, an dem die Dateien das letze mal verändert / bearbeitet wurden



2.6.2.1.11 MpRecipeUIRecipeType

Alle Daten die beim Verwalten der Rezepte benötigt werden. Nach dem Erstellen eines Rezepts wird die Rezeptliste automatisch erneuert.

Parameter	Datentyp	Beschreibung
List	MpRecipeUIRecipeListType	Liste aller verfügbaren Rezepte (Dateinamen), sowie die Navigation für die Liste
Load	BOOL	Laden des ausgewählten Rezepts **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
Save	BOOL	Speichern des ausgewählten Rezepts **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
Filter	STRING[255]	Filter, mit dem Rezepte aussortiert werden können. Der Filterstring bezieht sich auf die Rezeptnamen und zeigt die Rezepte an, die den String im Namen enthalten. Es ist möglich in der Suchfunktion das Platzhaltersymbol "*" zu verwenden. Für mehr Information zur Suchfunktion siehe Abschnitt "'Suchfunktion" in der Beschreibung von MpRecipeUI **VC4 Anbindung:** Datapoint von *String*
SortOrder	MpRecipeUISortOrderEnum	Sortieren der Rezeptliste **VC4 Anbindung:** Datapoint von *Button* / Type: ToggleDatapoint / SetValue: 1 / ResetValue: 0
Refresh	BOOL	Aktualisieren der Rezeptliste **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
UpdateNotification	BOOL	Anfrage an alle mapp Komponenten, um die aktuellen Werte der Rezeptparameter einzutragen **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Delete	BOOL	Löschen des ausgewählten Rezepts **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
Rename	BOOL	Umbenennen des ausgewählten Rezepts **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
NewFileName	STRING[255]	Neuer Name des Rezepts **VC4 Anbindung:** Datapoint von *String*
Header	MpRecipeUIHeaderType	Header Daten des aktuell ausgewählten Rezepts



2.6.2.1.12 MpRecipeUISetupConfirmType

Über diesen Datentyp kann definiert werden, wann Bestätigungs Dialoge angezeigt werden sollen.

Parameter	Datentyp	Beschreibung
RecipeLoad	`BOOL`	Aktiviert einen Bestätigungs Dialog für das Laden eines Rezepts
RecipeSave	`BOOL`	Aktiviert einen Bestätigungs Dialog für das Überschreiben eines Rezepts: • Wird ein bestehendes Rezept gespeichert, welches bereits existiert, wird dieser Dialog angezeigt • Wird versucht ein neues Rezept zu erzeugen, welches aber bereits existiert, wird dieser Dialog angezeigt Damit die Dialog-Box während der oben genannten Fälle angezeigt werden kann, muss "RecipeCreate = `TRUE`" sein.
RecipeCreate	`BOOL`	Aktiviert einen Bestätigungs Dialog für das Erstellen eines Rezepts
RecipeDelete	`BOOL`	Aktiviert einen Bestätigungs Dialog für das Löschen eines Rezepts
RecipeRename	`BOOL`	Aktiviert einen Bestätigungs Dialog für das Umbenennen eines Rezepts

2.6.2.1.13 MpRecipeUISetupType

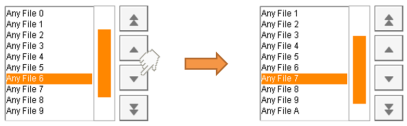
Zusätzliche Konfigurationsmöglichkeiten, um die Visualisierung zu beschreiben.

Parameter	Datentyp	Wert	Beschreibung
RecipeListSize	UINT	20	Zahl an Rezepten, die auf einer Seite der Liste auf der Visualisierung dargestellt werden können Es können maximal 20 Rezepte angezeigt werden
RecipeListScrollWindow	USINT	1	Zeigt an, wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden
Confirmation	MpRecipeUISetupConfirmType		Anzeige der Bestätigungs Dialoge
AutoLoadHeader	BOOL		Definiert, ob der Header eines Rezepts automatisch geladen werden soll, wenn es in der Rezeptliste ausgewählt wird (zu finden sind die Header Daten in der Struktur MpRecipeUIRecipeType)

RecipeListSize

RecipeListSize gibt an, wie viele Dateien auf einer Seite einer Listbox im VC4 angezeigt werden. Die Listbox in der Visualisierung zeigt je nach Größe unterschiedlich viele Elemente an. Wie viele Elemente die Listbox anzeigt, kann durch definieren eines Simulationswertes (SimulationValue) in den Listbox Eigenschaften ermittelt werden (siehe Bild oben: "Any File"). Die Anzahl dieser Elemente muss mit dem Wert von RecipeListSize übereinstimmen.

RecipeListScrollWindow



RecipeListScrollWindow gibt an, wie viele Einträge beim Scrollen "vorrausschauend" angezeigt werden sollen. In diesem Beispiel wurde RecipeListScrollWindow = 3 definiert. Befindet sich nun die aktuelle Auswahl an 4. letzter Stelle in der Liste, so wird bei erneutem Drücken des StepDown Buttons der nächste Eintrag (Any File 7) ausgewählt und vorrauschschauend die nächsten 3 Einträge (Any File 8, Any File 9 und Any File A) angezeigt.

2.6.2.1.14 MpRecipeXmlHeaderType

In dieser Struktur wird die Header Information für Rezepte abgespeichert. Die Header Struktur wird beim Erzeugen / Speichern / Lesen mithilfe der MpRecipeXml Komponente verwendet.

Parameter	Datentyp	Beschreibung
Name	STRING[100]	Der Name des Rezepts
Description	STRING[255]	Kommentar, welcher das Rezept beschreibt
Version	STRING[20]	Version des Rezepts
DateTime	DATE_AND_TIME	Das Datum und die Uhrzeit, wann das Rezept erstellt wurde

2.6.2.1.15 MpRecipeXmlInfoType

Mit diesem Datentyp wird für die MpRecipeXml Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
FileSize	UDINT	Die Größe des Rezepts, das auf dem jeweiligen Speichermedium abgelegt ist
PendingSync	UDINT	Die Nummer der anstehenden Speicher-Benachrichtigungen ("SaveNotification"), welche noch nicht bestätigt wurden
LastLoadedRecipe	STRING[255]	Das letzte geladene Rezept
Diag	MpRecipeDiagType	Eine Diagnosestruktur für die StatusID

2.6.2.2 Enumeratoren

2.6.2.2.1 MpRecipeCsvLoadEnum

Diese Enumeration entscheidet, in welcher Form das Rezept geladen wird, nur der Header oder das gesamte Rezept.

Aufzählung	Beschreibung
mpRECIPE_CSV_LOAD_ALL	Der gesamte Inhalt des Rezepts wird geladen
mpRECIPE_CSV_LOAD_HEADER	Nur der Header des Rezepts wird geladen

2.6.2.2.2 MpRecipeUIMessageEnum

Dieser Aufzählungsdatentyp informiert über die Art des Dialogfensters. Die Dialgofenster werden nur angezeigt, wenn sie in der Struktur MpRecipeUISetupConfirmType aktiviert wurden.

Aufzählung	Beschreibung
mpRECIPE_UI_MSG_NONE	Kein Dialogfenster
mpRECIPE_UI_MSG_CONFIRM_LOAD	Dialogfenster, dass zur Bestätigung des Ladevorganges auffordert
mpRECIPE_UI_MSG_CONFIRM_SAVE	Dialogfenster, dass zur Bestätigung des Speichervorganges auffordert
mpRECIPE_UI_MSG_CONFIRM_CREATE	Dialogfenster, dass zur Bestätigung für das Erstellen eines neuen Rezepts auffordert
mpRECIPE_UI_MSG_CONFIRM_DELETE	Dialogfenster, dass zur Bestätigung für das Löschen eines neuen Rezepts auffordert
mpRECIPE_UI_MSG_CONFIRM_RENAME	Dialogfenster, dass zur Bestätigung für das Umbenennen eines neuen Rezepts auffordert

2.6.2.2.3 MpRecipeUISortOrderEnum

Dieser Aufzählungsdatentyp entscheidet, in welcher Form sortiert wird. Dabei hat man die Möglichkeit, zwischen ansteigender und absteigender alphabetischer Reihenfolge zu wählen.

Aufzählung	Beschreibung
mpRECIPE_UI_SORT_ASCENDING	Sortierung in ansteigender (alphabetischer) Reihenfolge
mpRECIPE_UI_SORT_DESCENDING	Sortierung in absteigender (alphabetischer) Reihenfolge
mpRECIPE_UI_SORT_DATE_ASCENDING	Sortierung in ansteigender Reihenfolge über den Zugriffszeitpunkt
mpRECIPE_UI_SORT_DATE_DESCENDING	Sortierung in absteigender Reihenfolge über den Zugriffszeitpunkt

2.6.2.2.4 MpRecipeUIStatusEnum

Dieser Aufzählungsdatentyp gibt Auskunft über die aktuelle Aktivität der Komponente.

Aufzählung	Beschreibung
mpRECIPE_UI_STATUS_IDLE	Aktuell ist kein Prozess aktiv
mpRECIPE_UI_STATUS_LOAD	Ein Rezept wird geladen
mpRECIPE_UI_STATUS_SAVE	Ein Rezept wird gespeichert
mpRECIPE_UI_STATUS_CREATE	Ein Rezept wird erstellt
mpRECIPE_UI_STATUS_REFRESH	Die Rezeptliste wird aufgefrischt
mpRECIPE_UI_STATUS_NOTIFY	mapp Komponenten werden unterrichtet, dass neue Parameter zu verwenden sind
mpRECIPE_UI_STATUS_DELETE	Ein Rezept wird gelöscht
mpRECIPE_UI_STATUS_RENAME	Ein Rezept wird umbenannt
mpRECIPE_UI_STATUS_ERROR	Die letzte Operation hat einen Fehler generiert

2.6.2.2.5 MpRecipeXmlLoadEnum

Diese Enumeration entscheidet, in welcher Form das Rezept geladen wird, nur der Header oder das gesamte Rezept.

Aufzählung	Beschreibung
mpRECIPE_XML_LOAD_ALL	Der gesamte Inhalt des Rezepts wird geladen
mpRECIPE_XML_LOAD_HEADER	Nur der Header des Rezepts wird geladen

2.6.3 Statusnummern

2.6.3.1 1083342870: MpRecipeXml/Csv ist noch nicht aktiv

Beschreibung:

Dieser Funktionsbaustein kann erst aktiv werden, wenn MpRecipeXml / MpRecipeCsv aktiv ist.

Reaktion:

Der Funktionsbaustein wird diese Information an dem Ausgang "StatusID" anzeigen.

Ursache / Abhilfe:

- MpRecipeXml aktivieren
- MpRecipeCsv aktivieren

Konstante:

`mpRECIPE_INF_WAIT_RECIPE_FB`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.3.2 -2138044264: Synchrones Speichern nicht unterstützt

Beschreibung:

Synchronisiertes Speichern wird durch den aktuellen Parameter nicht unterstützt.

2.6.3.3 -2138044263: Synchronisiertes Speichern ausgeführt

Beschreibung:

Synchronisiertes Speichern wurde mit einem Mix aus synchronisierten und nicht synchronisierten Parametern ausgeführt.

2.6.3.4 -2138044262: CSV Datei enthält unerwartete Anzahl an Spalten

Beschreibung:

In der CSV Datei wurde eine Reihe gefunden, welche nicht die erwartete Anzahl an Spalten beinhaltet.

Ursache / Abhilfe:

- Kontrolle der CSV Datei

2.6.3.5 -2138044261: Inhalt kann nicht zu einem 32-bit Integer geparst werden

Beschreibung:

Der Inhalt {2:Content} in Zeile {3:Line} und Spalte {4:Column}, welche sich in der Datei {5:FileName} befindet, kann nicht zu einem 32-bit Integer geparst werden.

Zusätzliche Informationen

- {2:Content}: Der Inhalt, der geparst werden sollte.
- {3:Line}: Die Zeile, in der sich der Inhalt befindet.
- {4:Column}: Die Spalte, in der sich der Inhalt befindet.
- {5:FileName}: Die Datei, in der sich der Inhalt befindet.

Ursache

- Die Datei wurde extern editiert.
- Jeder String welche keine Nummer repräsentiert, kann nicht geparst werden.

2.6.3.6 -2138044260: Inhalt kann nicht zu einem 32-bit UINT geparst werden

Beschreibung:

Der Inhalt {2:Content} in Zeile {3:Line} und Spalte {4:Column} in der Datei {5:FileName} kann nicht zu einem 32-bit UINT geparst werden.

Zusätzliche Informationen

- {2:Content}: Der Inhalt, der geparst werden sollte.
- {3:Line}: Die Zeile, in der sich der Inhalt befindet.
- {4:Column}: Die Spalte in der sich der Inhalt befindet.
- {5:FileName}: Die Datei, in der sich der Inhalt befindet.

Ursache

- Die Datei wurde extern editiert.
- Jeder String welche keine Nummer repräsentiert, kann nicht geparst werden.

2.6.3.7 -2138044259: Der Inhalt kann nicht auf einen Boolean geparst werden.

Beschreibung:

Der Inhalt {2:Content} in Zeile {3:Line} und Spalte {4:Column} in der Datei {5:FileName} kann nicht auf einen Boolean geparst werden.

Zusätzliche Informationen

- {2:Content}: Der Inhalt, der geparst werden sollte.
- {3:Line}: Die Zeile, in der sich der Inhalt befindet.
- {4:Column}: Die Spalte, in der sich der Inhalt befindet.
- {5:FileName}: Die Datei, in der sich der Inhalt befindet.

Ursache

- Die Datei wurde extern editiert.
- Jeder String welche keinen Boolschen Wert repräsentiert, kann nicht geparst werden.

2.6.3.8 -2138044258: Inhalt konnte nicht auf einen 64-bit Float geparst werden

Beschreibung:

Der Inhalt {2:Content} in Zeile {3:Line} und Spalte {4:Column} in Datei {5:FileName} konnte nicht auf einen 64-bit Float geparst werden.

Zusätzliche Informationen

- {2:Content}: Der Inhalt, der geparst werden sollte.
- {3:Line}: Die Zeile, in der sich der Inhalt befindet.
- {4:Column}: Die Spalte, in der sich der Inhalt befindet.
- {5:FileName}: Die Datei, in der sich der Inhalt befindet.

Ursache

- Die Datei wurde extern editiert
- Jeder String welche keine Nummer repräsentiert, kann nicht geparst werden.

2.6.3.9 -2138044255: PV konnte nicht lokalisiert werden

Beschreibung:

Prozessvariable {2:PVName} konnte nicht lokalisiert werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:PVName}: Name der Prozessvariable.

2.6.3.10 -2138044254: Datentyp wird nicht unterstützt

Beschreibung:

Der Datentyp {2:DataType} der Variable {3:PVName} ist kein bekannter IEC-Datentyp und wird daher nicht unterstützt.

Zusätzliche Informationen

- {2:DataType}: Der angegebene Datentyp.
- {3:PVName}: Der Name der Prozessvariable.

Ursache / Abhilfe:

- Ändern des Datentypen
- Für mehr Information über nicht unterstützte Datentypen siehe hier

2.6.3.11 -2138044253: PV-Datentyp anders als auf dem Zielsystem

Beschreibung:

Der Datentyp der Variable {2:PVName} ist nicht gleich dem auf dem Zielsytem verwendeten Datentyp.

Zusätzliche Informationen

- {2:PVName}: Name der betroffenen Prozessvariable.

Ursache / Abhilfe:

- Ändern des Datentypen zu ursprünglichen Datentypen

2.6.3.12 -2138044252: Wert der Variable außerhalb des Wertebereichs

Beschreibung:

Der Wert der Variable {2:PVName} ist außerhalb des Wertebereiches des Datentyps.

Zusätzliche Informationen

- {2:PVName}: Name der betroffenen Prozessvariable.

Ursache / Abhilfe:

- Ändern des Wertebereichs

2.6.3.13 -2138044251: String-Wert der PV zu groß

Beschreibung:

Der Stringwert der Variable {2:PVName} passt nicht in den auf dem Zielsystem reservierten Speicher.

Zusätzliche Informationen

- {2:PVName}: Name der Prozessvariable.

Ursache / Abhilfe:

- Allokieren eines größeren Speicherblockes.

2.6.3.14 -2138044250: PV konnte nicht Teil des Rezeptes werden

Beschreibung:

Prozessvariable {2:PVName} zu einem Teil des Rezeptes zu machen ist fehlgeschlagen. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:PVName}: Name der betroffenen Prozessvariable.

Ursache / Abhilfe:

- Überprüfung der Prozessvariable auf korrekten Datentypen
- Überprüfung der Prozessvariable auf korrekte Angabe

2.6.3.15 -2138044248: Datentyp nicht zulässig

Beschreibung:

Der Datentyp wird nicht als Teil eines Rezeptes unterstützt.

Ursache / Abhilfe:

- Umändern des Datentyps
- Für mehr Information zu nicht unterstütze Datentypen siehe hier

2.6.3.16 -2138044247: Rezept Formatfehler

Beschreibung:

Rezept-Formatfehler bei {2:Context}.

Zusätzliche Informationen

- {2:Context}: Text, welcher Formatfehler beinhaltet

2.6.3.17 -2138044246: Ungültiger Rezeptinhalt

Beschreibung:

Ungültiger Rezeptinhalt in {2:Context}.

Zusätzliche Informationen

- {2:Context}: Text, welcher ungültigen Rezeptinhalt beinhaltet

Ursache / Abhilfe:

- Wurde die Datei extern editiert, muss kontrolliert werden, ob der Inhalt der Datei nicht beschädigt wurde. Das kann, zum Beispiel, eine Angabe eines falschen xml-Tags sein.

2.6.3.18 -2138044244: Erwartetes Attribut fehlt in Element

Beschreibung:

Erwartetes Attribut {2:Attribute} fehlt in Element {3:Element}.

Zusätzliche Informationen

- {2:Attribute}: Erwartetes Attribut
- {3:Element}: Element

2.6.3.19 -2138044242: Erwartetes Element fehlt

Beschreibung:

Erwartetes Element {2:Element} fehlt.

Zusätzliche Informationen

- {2:Element}: Element

2.6.3.20 -2138044241: Unerwartetes Element gefunden

Beschreibung:

Unerwartetes Element {2:Element} gefunden.

Zusätzliche Informationen

- {2:Element}: Element

2.6.3.21 -2138044240: Parameter Inhaltstyp nicht bekannt

Beschreibung:

Parameter Inhaltstyp {2:Type} konnte nicht erkannt werden. Parsen des Inhalts kann nicht ausgeführt werden.

Zusätzliche Informationen

- {2:Type}: Der Typ des Parameters.

Ursache

- Der Fehler tritt auf wenn die Datei extern editiert wurde
- In der Datei darf als Type nur HeaderData und PvParameter verwendet werden:

```
<Element Name="MpRecipe.Header" Type="HeaderData">
    <Property ID="Name" Type="STRING" Value="TestHeader" />
    <Property ID="Description" Type="STRING" Value="TestDescription" />
    <Property ID="Version" Type="STRING" Value="0.0.99" />
    <Property ID="Created" Type="UDINT" Value="0" />
</Element>
<Element Name="MpRecipeXm:UserNestedStruct" Type="PvParameter">
    <Group ID="MpRecipeXm:UserNestedStruct">
        <Property ID="DeviceName" DataType="USINT" Value="0" />
        <Group ID="open">
            <Property ID="Commandname" DataType="STRING" Value="" />
```

2.6.3.22 -2138044233: Datentyp der Header-Eigenschaft ist nicht gültig

Beschreibung:

Datentyp der Header-Eigenschaft {2:Property} ist nicht gültig. Erwartet {3:Expected}, Aktuell {4:Actual}.

Zusätzliche Informationen

- {2:Property}: Header-Eigenschaft.
- {3:Expected}: Erwarteter Datentyp.
- {4:Actual}: Aktueller Datentyp.

Ursache / Abhilfe:

- Überprüfung des angegebenen Headers
- Umändern des Headers

2.6.3.23 -2138044232:Ungültiger Header-Inhalt

Beschreibung:

Beim Laden der Datei {3:File} wurde ein ungültiger Header Inhalt bei {2:Context} festgestellt. Fehlerursache: {1:ErrorNumber}

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache
- {2:Context}: Inhalt, welcher ungültige Information beinhaltet
- {3:File}: Betroffene Datei

2.6.3.24 -2137882618: Warnungen während dem Speichern eines Rezepts

Beschreibung:

Während dem Speichern eines Rezepts wurden Warnungen ausgelöst. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist. Der Speichervorgang wird fortgeführt und es werden soviele registrierte PVs in das Rezept gespeichert wie möglich.

Ursache / Abhilfe:

-

Konstante:

`mpRECIPE_WRN_SAVE_WITH_WARN`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.25 -2137882617: Warnungen während dem Laden eines Rezepts

Beschreibung:

Während dem Laden eines Rezepts wurden Warnungen ausgelöst. Weitere Details siehe Logger (unter anderem der PV Name).

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist. Der Ladevorgang wird fortgeführt und es werden soviele Parameter aus dem Rezept in die registrierten PVs geladen wie möglich.

Ursache / Abhilfe:

- Die Parameter im Rezept überprüfen (Datentyp, beschädigte XML Struktur im Rezept, ...)
- Überprüfen ob die String-Länge eines Parameters verändert wurde (z.B.: STRING[80] auf STRING[40])

Konstante:

`mpRECIPE_WRN_LOAD_WITH_WARN`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.26 -2137882611: Ungültiger Wert für "UIConnect"

Beschreibung:

Am Eingang "UIConnect" des Funktionsbausteins MpRecipeUI ist ein ungültiger Wert (NULL Zeiger) angeschlossen.

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Die Struktur am Eingang "UIConnect" des Funktionsbausteins MpRecipeUI nachprüfen (NULL angehängt)

Konstante:

`mpRECIPE_WRN_MISSING_UICONNECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.3.27 -2137882606: Keine PV registriert

Beschreibung:

Es wurde keine Prozess Variable registriert.

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Es wurde weder ein MpRecipeRegPar noch ein MpRecipeRegParSync Funktionsbaustein gefunden

Konstante:

`mpRECIPE_WRN_NO_PV_REGISTERED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv

2.6.3.28 -2137882603: Ein leeres Rezept wurde erstellt

Beschreibung:

Ein leeres Rezept wurde gespeichert.

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Parameter registiert, aber falsche Kategorie angegeben

Konstante:

`mpRECIPE_WRN_EMPTY_RECIPE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.29 -2137882600: Keine PV vorhanden

Beschreibung:

Es wurde versucht ein Rezept zu laden, in dem keine einzige der registrierten Prozess Variablen gefunden wurde.

Reaktion:

Der Funktionsblock wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Rezept enthält keine validen Parameter
- Wird der Funktionsbaustein MpRecipeCsv verwendet und die Option "Encrypt = `TRUE`" in der MpRecipeCsv Konfiguration, so muss sichergestellt sein, dass ein Rezept mit der gleichen Passphrase geladen wird mit der es auch gespeichert/erstellt wurde. Wurde die Passphrase in der Zwischenzeit verändert (Passphrase = 3), so kann das Rezept welches mit der Passphrase = 1 erstellt wurde nicht geöffnet werden.

Konstante

`mpRECIPE_NO_PV_FOUND`

Diese Funktionsblöcke / Funktionen können diesen Fehler melden:

- MpRecipeCsv
- MpRecipeXml

2.6.3.30 -2137882599: Listgröße außerhalb des Limits

Beschreibung:

Die Listgröße (UISetup.RecipeListSize) ist außerhalb des Limits (1-20).

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Anpassen der Listgröße

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.3.31 -1064302433: Eine Datei mit gleichem Namen existiert bereits

Beschreibung:

Eine Datei mit gleichem Namen existiert bereits.

Ursache / Abhilfe:

- Angabe eines anderen Dateinamens

2.6.3.32 -1064302432: Information über PV konnte nicht lokalisiert werden

Beschreibung:

Information über Prozessvariable {2:PVName} konnte nicht lokalisiert werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:PVName}: Name der Prozessvariable.

Ursache / Abhilfe:

- Überprüfung der Prozessvariable.
- Prozessvariable wurde falsch angegeben.

2.6.3.33 -1064302421: Benötigtes Attribut fehlt

Beschreibung:

Benötigtes Attribut {2:Attribute} fehlt in Element {3:Element}.

Zusätzliche Informationen

- {2:Attribute}: Benötigtes Attribut
- {3:Element}: Element in welches Attribut fehlt

Ursache

- Dieser Fehler tritt auf, wenn die Datei extern editiert wurde.
- Die Datei muss die Attribute *Name* und *Type* enthalten.
  Beispiel: <Element *Name*="MpRecipeXml:UserNestedStruct" *Type*="PvParameter">

2.6.3.34 -1064302419: Benötigtes Element fehlt

Beschreibung:

Benötigtes Element {2:Element} fehlt.

Zusätzliche Informationen

- {2:Element}: Element

Ursache / Abhilfe:

- Kontrollieren ob Prozess Variable registriert wurde
- Kontrollieren ob der "Category" Eingangsparameter verwendet wurde und ob richtige Kategorie angegeben wurde.

2.6.3.35 -1064302415: Fehler beim Erstellen eines Datenmodells

Beschreibung:

Fehler beim Erstellen eines Datenmodells für das Laden / Speichern eines Rezepts. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.

2.6.3.36 -1064302414: Fehler beim Schließen eines Datenmodells

Beschreibung:

Fehler beim Schließen eines Datenmodells nachdem ein Rezept geladen / gespeichert wurde. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.

2.6.3.37 -1064302413: Fehler beim Initialisieren eines Datenmodells

Beschreibung:

Fehler beim Initialisieren eines Datenobjekt, welches benutzt wird um ein neues Rezept zu erstellen. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.

2.6.3.38 -1064302412: Parameter mit undefiniertem Namen gefunden

Beschreibung:

Parameter mit undefiniertem Namen gefunden.

Ursache / Abhilfe:

- Vergeben eines Namens

2.6.3.39 -1064302411: MpFile konnte keine Dateien auflisten

Beschreibung:

MpFile konnte keine Dateien für das ausgewählte Laufwerk auflisten. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.

Ursache / Abhilfe:

- MpFile Bibliothek hinzufügen
- Angegebenes Speichermedium kontrollieren

2.6.3.40 -1064302410: Service konnte nicht hinzugefügt werden

Beschreibung:

Hinzufügen des Service für das Rezeptsystem fehlgeschlagen. Fehlerursache: {1:ErrorNumber}

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache

2.6.3.41 -1064302407: Das Rezept oder die Signatur wurde verändert

Beschreibung:

Das Verifizieren der Signatur ist fehlgeschlagen. Weitere Details siehe Logger.

Ursache / Abhilfe:

- Die Rezeptdatei oder die Signaturdatei wurde extern verändert

2.6.3.42 -1064302406: Signaturdatei fehlt

Beschreibung:

Die für das Rezept benötigte Signaturdatei wurde nicht gefunden aber wird benötigt. Weitere Details siehe Logger.

Ursache / Abhilfe:

- Überprüfen, ob die Signaturdatei existiert.

2.6.3.43 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom

- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeRegPar
- MpRecipeRegParSync
- MpRecipeUI

Konstante:

`mpRECIPE_ERR_ACTIVATION`

2.6.3.44 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeRegPar
- MpRecipeRegParSync
- MpRecipeUI

Konstante:

`mpRECIPE_ERR_MPLINK_NULL`

2.6.3.45 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeRegPar

- MpRecipeRegParSync
- MpRecipeUI

Konstante:

`mpRECIPE_ERR_MPLINK_INVALID`

2.6.3.46 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeRegPar
- MpRecipeRegParSync
- MpRecipeUI

Konstante:

`mpRECIPE_ERR_MPLINK_CHANGED`

2.6.3.47 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeRegPar
- MpRecipeRegParSync
- MpRecipeUI

Konstante:

`mpRECIPE_ERR_MPLINK_CORRUPT`

2.6.3.48 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeRegPar
- MpRecipeRegParSync
- MpRecipeUI

Konstante:

`mpRECIPE_ERR_MPLINK_IN_USE`

2.6.3.49 -1064239091: Ungültige Konfiguration

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Konfiguration ist beschädigt
- Die Funktion ist in der Konfiguration nicht aktiviert

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

- MpRecipeXml
- MpRecipeCsv

Konstante:

`mpRECIPE_ERR_CONFIG_INVALID`

2.6.3.50 -1064140799: Fehler beim Speichern des Rezepts

Beschreibung:

Ein Fehler ist aufgetreten während dem Speichern eines Rezepts. Das deutet auf ein Problem mit dem Dateisystem hin.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Dateinamen am Eingang "FileName" überprüfen
- Das File Device am Eingang "DeviceName" überprüfen
- Der Dateiname darf folgende Zeichen nicht beinhalten: \ / : * ? < > |
- Überprüfen ob auf dem angegebenen Speichermedium genügend Speicherplatz vorhanden ist
- Überprüfen ob ein externes Programm, wie zum Beispiel Excel, auf die Datei zugreift
- Bei der Verwendung von verschlüsselten Passwörtern in der Konfiguration muss darauf geachtet werden, dass das Passwort im Nachhinein nicht manipuliert wurde, wie beispielsweise hinzufügen oder löschen von Zeichen. Für mehr Information siehe hier im Abschnitt "Passwortverschlüsselung in der Konfiguration"

Konstante:

`mpRECIPE_ERR_SAVE_DATA`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.51 -1064140798: Fehler beim Laden eines Rezepts

Beschreibung:

Beim Laden eines Rezepts ist ein Fehler aufgetreten. Die Parameter wurden nicht übertragen.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Sicherstellen, dass die Rezeptdatei eine gültige XML bzw. CSV Struktur besitzt
- Überprüfen ob, auf das File Device am Eingang "DeviceName" zugegriffen werden kann
- Bei der Verwendung von verschlüsselten Passwörtern in der Konfiguration muss darauf geachtet werden, dass das Passwort im Nachhinein nicht manipuliert wurde, wie beispielsweise hinzufügen oder löschen von Zeichen. Für mehr Information siehe hier im Abschnitt "Passwortverschlüsselung in der Konfiguration"

Konstante:

`mpRECIPE_ERR_LOAD_DATA`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.52 -1064140797: Ungültiges File Device

Beschreibung:

Das angegebene File Device am Eingang "DeviceName" ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Am Eingang "DeviceName" ist nichts angehängt
- Das angehängte File Device existiert nicht
- Es konnte keine Verbindung zum gewünschten File Device erstellt werden
- Beim Arbeiten auf Linux-Systemen kann auf keine Systemordner mit Zugangsbeschränkung zugegriffen werden. Dies kann zum Beispiel der "lost+found" Ordner sein.

```
C:\Users\kratzern>dir \\10.43.37.3\usershare
 Volume in drive \\10.43.37.3\usershare is usershare
 Volume Serial Number is 9325-608F

 Directory of \\10.43.37.3\usershare

12/12/2017  07:42 PM    <DIR>          .
01/19/2018  02:01 PM    <DIR>          ..
12/12/2017  07:27 PM    <DIR>          TestDir
01/01/1970  02:00 AM    <DIR>          lost+found
               0 File(s)              0 bytes
               4 Dir(s)     473,661,440 bytes free
```

Konstante:

`mpRECIPE_ERR_INVALID_FILE_DEV`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.53 -1064140796: Ungültiger Dateiname

Beschreibung:

Der Dateiname am Eingang "FileName" ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Am Eingang "FileName" ist nichts angehängt
- Die angehängte Datei existiert nicht
- Keine Zugriffsberechtigung

Konstante:

`mpRECIPE_ERR_INVALID_FILE_NAME`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.54 -1064140795: Die Komponente führt bereits einen Befehl aus

Beschreibung:

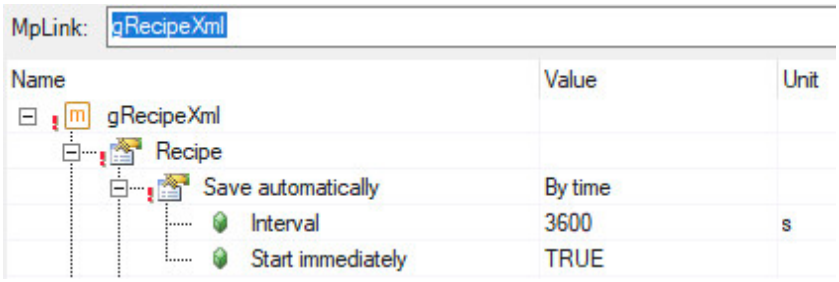
Kommt vor, wenn die Komponente von mehreren Quellen zur gleichen Zeit in Anspruch genommen wird.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Kann auftreten, wenn der Funktionsbaustein aus mehreren Quellen zur selben Zeit Befehle bekommt (z.B. ein neuer Lesebefehl in der Struktur MpRecipeUIConnectType von MpRecipeUI, während ein Befehl "Load" des Funktionsbausteins MpRecipeXml aktiv ist)
- Es gilt zu beachten, wie die Option "Save automatically" in der mapp Recipe Konfigurationen gesetzt ist. Wurde beispielsweise die automatische Speicherung des Rezeptes für jede Stunde festgelegt, so kann in der Zeit in der das Rezept abgespeichert wird, keine weiteres Kommando gesetzt werden. Wird zu dieser Zeit, zum Beispiel ein "Save" Befehl am Funktionsbaustein MpRecipeXml gesetzt, so wird dieser Fehler angezeigt.



Konstante:

`mpRECIPE_ERR_CMD_IN_PROGRESS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.55 -1064140792: Fehler beim Konvertieren der Parameter des Rezepts

Beschreibung:

Beim Konvertieren der Parameter eines Rezepts ist es zu einem Fehler gekommen. Es wurde kein Rezept abgespeichert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar. Der Speichervorgang wird abgebrochen. Es werden keine Daten in das Rezept gespeichert.

Ursache / Abhilfe:

- Die abzuspeichernden Parameter überprüfen
- Arrays, welche nicht mit Index 0 starten können nicht registriert werden
- Multidimensionale Arrays können nicht registriert werden
- Arrays die mehr als 65535 Elemente besitzen können nicht registriert werden
- Abgeleitete Datentypen
- Variablen mit dem Datentyp TIME_OF_DAY können nicht registriert werden
- Variablen mit dem Datentyp BYTE können nicht registriert werden
- Variablen mit dem Datentyp WORD können nicht registriert werden
- Variablen mit dem Datentyp DWORD können nicht registriert werden
- Variablen mit dem Datentyp LWORD können nicht registriert werden
- Variablen mit dem Datentyp WSTRING können nicht registriert werden
- Variablen mit dem Datentyp LINT können nicht registriert werden

Konstante:

`mpRECIPE_ERR_SAVE_WITH_ERRORS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.56 -1064140791: Fehler beim Konvertieren des Rezepts in Parameter

Beschreibung:

Beim Konvertieren des Rezepts in Parameter ist es zu einem Fehler gekommen. Es wurde kein Rezept gelesen. Weitere Details siehe Logger (unter anderem der PV Name).

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar. Der Ladevorgang wird abgebrochen. Es werden keine Daten in die PVs gespeichert.

Ursache / Abhilfe:

- Die Parameter im Rezept überprüfen (Datentyp, beschädigte XML Struktur im Rezept, ...)

Konstante:

`mpRECIPE_ERR_LOAD_WITH_ERRORS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv
- MpRecipeUI

2.6.3.57 -1064140790: Keine Verbindung zur Parent Komponente

Beschreibung:

Es wurde keine MpRecipeXml / MpRecipeCsv Komponente gefunden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob die fehlergebende Komponente einen Link zur MpRecipeXml / MpRecipeCsv Komponente hat
- Prüfen, ob MpRecipeXml / MpRecipeCsv richtig arbeitet

Konstante:

`mpRECIPE_ERR_MISSING_RECIPE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeRegPar
- MpRecipeRegParSync
- MpRecipeUI

2.6.3.58 -1064140789: Keine Verbindung zum Dateimanager

Beschreibung:

Es wurde keine MpFile Komponente gefunden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob die Bibliothek MpFile auf die Steuerung übertragen wurde

Konstante:

`mpRECIPE_ERR_MISSING_MPFILE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.3.59 -1064140788: Die gewählte Art zu Sortieren ist ungültig

Beschreibung:

Die ausgewählte Art zu Sortieren am Eingang "SortOrder" ist nicht zulässig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am Eingang "SortOrder" in der Struktur MpRecipeUIConnectType überprüfen

Konstante:

`mpRECIPE_ERR_INVALID_SORT_ORDER`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.3.60 -1064140786: Keine gültige PV

Beschreibung:

Unter dem angegebenen Namen konnte kein PV gerfunden werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar. Ein erneute Evaluierung der PV wird bei einer positiven Flanke am "Enable" Eingang durchgeführt.

Ursache / Abhilfe:

- PV mit diesem Namen wird vom System nicht verwendet, Name/Adresse der PV überprüfen

Konstante:

`mpRECIPE_ERR_INVALID_PV_NAME`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeRegPar
- MpRecipeRegParSync

2.6.3.61 -1064140785: LoadType ungültig

Beschreibung:

Der Wert am Eingang "LoadType" ist ungültig.

Reaktion:

Der Funktionsbaustein onsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Wert am Eingang "LoadType" überprüfen, mögliche Optionen sind in der MpRecipeXmlLoadEnum zu finden

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv

2.6.3.62 -1064140784: Fehler beim Erstellen der Dateiliste

Beschreibung:

Es konnte keine neue Rezeptliste erstellt werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Verbindung zum Speichermedium überprüfen ("DeviceName" des MpRecipeXml Funktionsblocks)

Konstante:

`mpRECIPE_ERR_LISTING_FILES`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.3.63 -1064140783: PV Name ist NULL

Beschreibung:

Es wurde ein ungültiger PV Name (NULL) an die Funktion übergeben.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar. Ein erneute Evaluierung der PV wird bei einer positiven Flanke am "Enable" Eingang durchgeführt.

Ursache / Abhilfe:

- PV Name ist NULL

Konstante:

`mpRECIPE_ERR_PV_NAME_NULL`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeRegPar
- MpRecipeRegParSync

2.6.3.64 -1064140781: Speichern aktiv

Beschreibung:

Ein synchronisierter Speichervorgang ist bereits aktiv.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Warte bis der aktuelle Speichervorgang abgeschlossen ist
- Einer der Parameter, welcher gespeichert werden soll, hat noch keine "SaveNotification" gesendet

Konstante:

`mpRECIPE_ERR_SYNC_SAVE_ACTIVE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeXml
- MpRecipeCsv

2.6.3.65 -1064140780: Fehler beim Löschen

Beschreibung:

Beim Löschen ist ein Fehler aufgetreten.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Datei zum Löschen nicht vorhanden
- Die Datei darf nicht gelöscht werden (schreibgeschützt)

Konstante:

`mpRECIPE_ERR_DELETING_FILE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.3.66 -1064140777: Datei kann nicht umbenannt werden

Beschreibung:

Ausgewählte Datei kann nicht umbenannt werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Dateinamen angeben

Konstante:

`mpRECIPE_ERR_RENAMING_FILE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpRecipeUI

2.6.4 Alarme

2.6.4.1 mpRECIPE_ALM_SAVE_FAILED: Speichern des Rezepts ist fehlgeschlagen

Beschreibung:

Während dem Speichern ist ein Fehler aufgetreten. Es wurde kein Rezept geschrieben.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Ein möglicher Grund ist ein falsches File Device am Eingang "DeviceName"
- Überprüfen, ob das File Device (USB Stick, Speicherkarte) richtig angesteckt ist

Verhalten

Edge Alarm

2.6.4.2 mpRECIPE_ALM_SAVE_FAILED_PART: Speichern des Rezepts teilweise fehlgeschlagen

Beschreibung:

Das Rezept konnte nur teilweise erfolgreich gespeichert werden. Die Parameter wurden nicht alle richtig abgespeichert. Weitere Details im Logger.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Die abzuspeichernden Parameter überprüfen

Verhalten

Edge Alarm

2.6.4.3 mpRECIPE_ALM_LOAD_FAILED: Laden des Rezepts ist fehlgeschlagen

Beschreibung:

Während dem Laden ist ein Fehler aufgetreten. Es wurden keine Daten geladen.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Ein möglicher Grund ist ein falsches File Device am Eingang "DeviceName"
- Überprüfen, ob das Dateiformat stimmt

Verhalten

Edge Alarm

2.6.4.4 mpRECIPE_ALM_LOAD_FAILED_PART: Laden des Rezepts teilweise fehlgeschlagen

Beschreibung:

Das Rezept konnte nur teilweise erfolgreich geladen werden. Die Parameter wurden nicht alle richtig geladen. Weitere Details siehe Logger.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Die Parameter im Rezept überprüfen (Datentyp, beschädigte XML Struktur im Rezept, ...)

Verhalten

Edge Alarm

3 mapp Backup: Backup erstellen und installieren

mapp Backup kann Sicherungsdateien des Zielsystems, sogenannte Backups, auf ein Speichermedium, wie zum Beispiel ein USB-Stick, ablegen oder ein bestehendes Backup installieren. Mithilfe von mapp Backup können auch Updates installiert werden.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Backup (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal ist das Tutorial mapp Backup Part 1 zu finden, welches erklärt wie ein Backup manuell erstellt werden kann.

3.1 Konzept

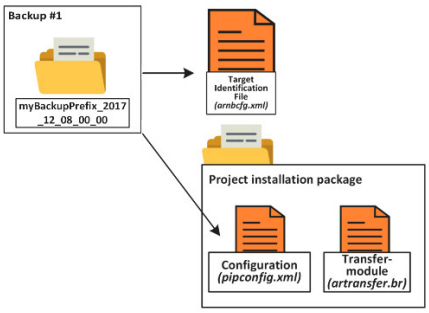
Allgemeines

- Als OEM möchte ich täglich automatisch ein Backup meiner Steuerung erstellen und auf einen USB-Stick abspeichern
- Als Servicetechniker möchte ich die Maschine schnell und einfach auf den neuesten Softwarestand updaten
- Als Servicetechniker möchte ich, wenn eine Steuerung beschädigt ist, die Steuerung austauschen und das zuletzt abgespeicherte Backup auf die Steuerung laden

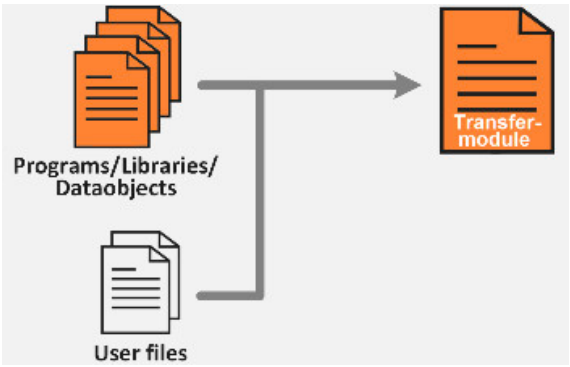
3.1.1 Aufbau eines Backups

Ein Backup ist ein Ordner der die unten angeführten Dateien beinhaltet. Der Name des Ordners definiert auch den Namen des Backups. Wird ein Backup installiert oder erstellt ist genau dieser Name zu verwenden.

Ein Backup besteht aus folgenden Dateien:



- **Haupt-Ordner:** In diesem Beispiel hat der Haupt-Ordner den Namen "myBackupPrefix_2017_12_08_00_00". Wird ein Backup erstellt, so wird ein Haupt-Ordner mit allen Dateien und Unterordnern abgespeichert. Der Name des Ordners wird dabei über den "Name" Parameter an MpBackupCore bzw. über den "NamePrefix"-Parameter der MpBackupCore Konfiguration definiert. Soll ein Backup installiert werden, so muss der Name des Haupt-Ordners angegeben werden.
- **Target Identification File:** In dieser Datei wird bestimmt auf welchem Zielsystem das Backup installiert werden soll. Per default erlaubt mapp Backup die Installation auf jedem Zielsystem. Je nach Bedarf kann aber in dieser Datei eine Einschränkung definiert werden, sodass das Backup nur auf Zielsystemen installiert werden kann, die z. B. die selbe ConfigID, MAC-Adresse, Stationnumber oder den selben Hostnamen aufweisen. Für mehr Information dazu siehe Aufbau des Projektinstallationspakets.
- **Project installation package:** In diesem Paket befinden sich zwei Dateien. Die Konfigurationsdatei (pipconfig.xml) und das zu installierende Transfermodul. In der Konfigurationsdatei wird festgelegt wie das Projekt installiert werden soll. Das Transfermodul enthält einen in sich konsistenten Projektstand mitallen nötigen Informationen um das Projekt am Zielsystem zu installieren. Es beinhaltet alle Applikationsmodule, Systemmodule sowie Benutzerdateien, falls vorhanden, welche auf der Benutzerpartition liegen. Das sind zum Beispiel alle Programme, Bibliotheken, Datenobjekte oder AR-Version:



Es gilt zu beachten, dass in der Erstellung der Konfigurationsdatei (pipconfig.xml) auch Installations-Einstellungen festgelegt werden. Diese sind in der Datei unter "InstallationSettings" zu finden.

```
<PIPCONFIG>
    <InstallationSettings>
        <ForceInitialInstallation>1</ForceInitialInstallation>
        <AllowInitialInstallation>0</AllowInitialInstallation>
        <IgnoreVersion>0</IgnoreVersion>
        <AllowPartitioning>0</AllowPartitioning>
        <Reboot>0</Reboot>
        <CiR>0</CiR>
        <KeepPvValues>0</KeepPvValues>
        <ExecuteEI>0</ExecuteEI>
        <ConsistentInstallation>0</ConsistentInstallation>
    </InstallationSettings><!--Warning: Do not edit beyond this line-->
    <Info>
        <ConfigId>MpBackup</ConfigId>
        <ConfigVersion>1.0.1</ConfigVersion>
        <PartitionInfos>
            <Part1>57009664</Part1>
            <Part2>10763264</Part2>
            <Part3>9810944</Part3>
            <Part4>232448</Part4>
        </PartitionInfos>
        <OrderNo>X20CP1586</OrderNo>
        <CompatibleCpuCode/>
        <ArType>AR Embedded</ArType>
    </Info>
    <Commands>
        <processTrsfm srcPath="artransfer.br"/>
    </Commands>
</PIPCONFIG>
```

Standardmäßig wird die Installationseinstellung "ForceInitialInstallation" verwendet. Das heißt sobald das Backup installiert wird, wird eine Initial-Installation durchgeführt.

Die Daten welche durch MpBackupCore erstellt werden sind inhaltlich ident mit den Daten eines Projektinstallationspakets, welches durch Automation Studio oder dem Runtime Utility Center erstellt werden kann. Sie unterscheiden sich nur in der Größe, da das PIP in Automation Studio/ Runtime Utility Center bei der Erstellung anders komprimiert wird als über MpBackupCore.

Mehr Information zum Backup und wie dieses konfiguriert werden kann, ist im Abschnitt Aufbau des Projektinstallationspakets zu finden.

MpBackupCore kann Backups erstellen und installieren. Für mehr Information wie ein Backup verwendet werden kann siehe hier.

Durch MpBackupCore kann aber ein Update installiert werden. Für mehr Information dazu siehe Abschnitt Installieren eines Updates.

3.1.2 Verwenden eines Backups

Ein Backup ist eine Sicherungsdatei des verwendeten Zielsystems. Es enthält Applikationsmodule, Systemmodule sowie Benutzerdateien, falls vorhanden, welche auf der Benutzerpartition liegen. Für mehr Information siehe Aufbau eines Backups .

In diesem Abschnitt wird erklärt, wie ein Backup verwendet werden kann:

Manuelles erstellen/installieren eines Backups

Ein Backup kann manuell von MpBackupCore erstellt werden. Dabei muss ein Name sowie ein Speichermedium für das Backup angegeben werden.

Über den Funktionsbaustein MpBackupCore kann auch ein Backup installiert werden. Dabei muss das gewünschte Backup sowie das Speichermedium wo sich das Backup befindet, angegeben werden.

Für mehr Information siehe hier.

Automatisches erstellen eines Backups

Zu diesem Thema gibt es auch einen Anwendungsfall

MpBackupCore bietet auch die Möglichkeit ein Backup automatisch zu erstellen. Soll eine Backup automatisch erstellt werden, so kann dies über die MpBackupCore Konfiguration definiert werden. Zur Laufzeit kann die Konfiguration vom Funktionsbaustein MpBackupCoreConfig angepasst werden. Bei einem automatisch erstellten Backup, wird beim Namen des Backups automatisch der Zeitstempel wie zum Beispiel "myPrefix_%Y_%m_%d_%H_%M_%S" angehängt.

3.1.3 Installieren eines Updates

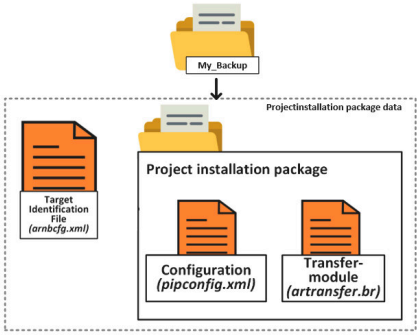
Mithilfe von MpBackupCore kann auch ein Update installiert werden. Wurde zum Beispiel die Applikation der Maschine verändert, kann diese Änderung in Form eines Updates am Zielsystem installiert werden.

Dies kann mithilfe eines Projektinstallationspakets realisiert werden:

Manuelles Update

Zu diesem Thema gibt es auch einen Anwendungsfall.

Wie im Abschnitt Aufbau eines Backups erklärt wird, sind die von MpBackupCore generierten Backup-Daten ident mit den Daten welche über das Projektinstallationspakets in Automation Studio bzw. dem Runtime Utility Center (RUC) generiert werden. Das bedeutet die Daten des Projektinstallationspakets können auch über MpBackupCore installiert werden. Damit dies funktioniert, müssen die Projektinstallationspaket-Dateien in einem Ordner abgespeichert werden. Der Name des Ordners ("My_Backup") kann dann anschließen an MpBackupCore angegeben werden. Dadurch kann ein Update am Zielsystem durchgeführt werden.



Automatisch nach Updates suchen

Wie ein Update automatisch angzeigt werden kann, ist im Anwendungsfall Update für Maschinensoftware automatisch anzeigen erklärt

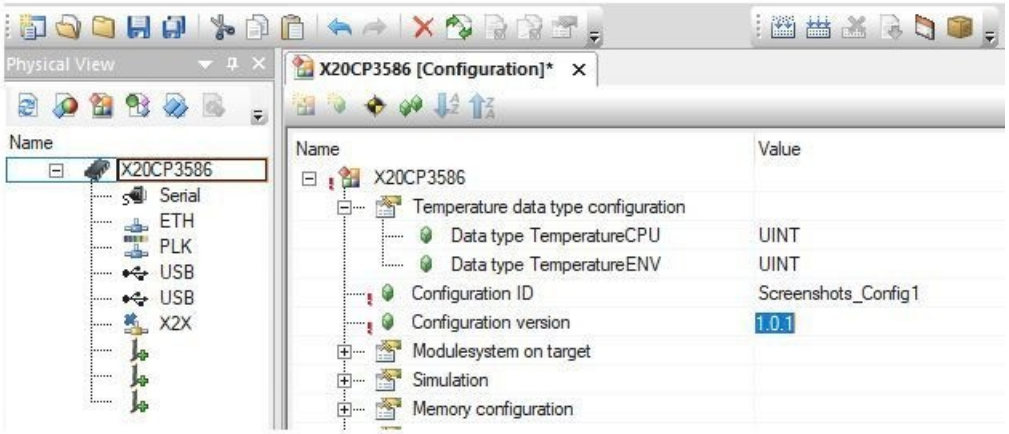
mapp Backup bietet die Funktionalität automatisch nach Updates zu suchen. Dies kann über die MpBackupCore Konfiguration oder den Funktionsbaustein MpBackupCoreConfig aktiviert werden. Folgende Schritte müssen beachtet werden:

1. Änderungen im Projekt durchführen (optional, nur für Ersteller des Updates)

Um ein Update an einer Maschine durchzuführen, muss zuerst die gewünschte Änderung im Automation Studio Projekt durchgeführt werden. Dabei kann zum Beispiel eine neue Funktion implementiert werden.

2. Konfigurations-Version anpassen (optional, nur für Ersteller des Updates)

Damit das Update im Anschluss von MpBackupCore automatisch erkannt wird, muss die Konfigurations-Version im geänderten AS-Projekt angepasst werden. Dies ist über die Konfiguration des Zielsystems möglich:



3. PIP erstellen (optional, nur für Ersteller des Updates)

Damit das Update installiert werden kann, muss ein Projektinstallationspaket (PIP) erstellt werden. Für mehr Information siehe den oberen Punkt "Manuelles Update"

4. Update automatisch anzeigen lassen

Die automatische Update-Suche muss in der MpBackupCore Konfiguration bzw. am Funktionsbaustein MpBackupCoreConfig aktiviert werden.

Je nachdem welcher Modus für die Suche nach Updates verwendet wird, wird nach Updates am angegebenen Speichermedium gesucht.

Ob ein Update vorhanden ist, wird über den "IsAvailable" Parameter an der Info-Struktur am Funktionsbaustein MpBackupCore sichtbar.

Um zu erkennen ob es sich nun um ein Update handelt oder nicht, wird über die Konfigurations-Version bestimmt. Ist die Version im Update höher als die Konfigurations-Version am Zielsystem, so wird automatisch ein Update angezeigt ("IsAvailable = `TRUE`"). Befinden sich mehrere Updates am Speichermedium wird automatisch das Update mit der höchsten Konfigurations-Version ausgewählt.

Sobald "IsAvailable = `TRUE`" ist, kann das Update mithilfe von MpBackupCore installiert werden. Als Name wird dabei der Name angegeben, welcher über den MpBackupAutoUpdateInfoType Datentypen ersichtlich ist.

3.1.4 Auditing von mapp Backup

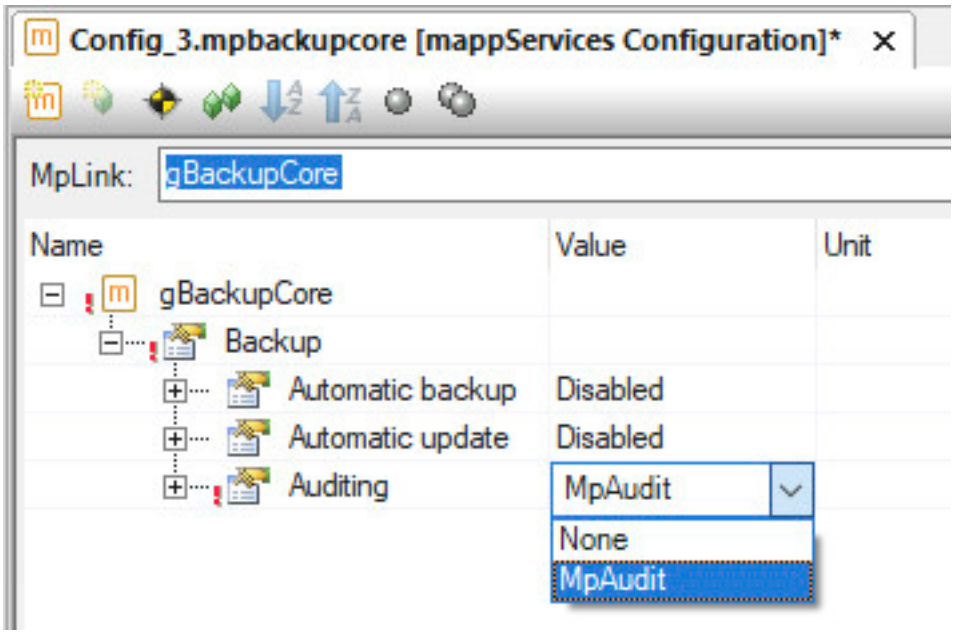
Ereignisse in mapp Backup können über mapp Audit aufgezeichnet werden. Für mehr Information wie ein Eintrag erstellt werden kann, siehe Abschnitt Definieren des Ereignistexts.

Durch diesen Ereignistyp kann aufgezeichnet werden, ob ein Backup erstellt bzw. installiert wurde.

> **Bei automatischer Erstellung eines Backups wird die Benutzer-Information (Kürzel "op") nicht hinzugefügt.**

> **Werden Ereignisse über eine mapp View Visualisierung ausgelöst, können manche Ereignisse nicht einem Benutzer zugeordnet werden. Dies muss, bei der Verwendung des "op" Kürzels beachtet werden. Für weitere Information siehe Abschnitt Ereignisse einem Benutzer zuordnen.**

Um die Ereignisse aufzuzeichnen, muss in der MpBackupCore Konfiguration die Option "Auditing" freigeschaltet sein:



Ereignis ID 149	Funktionsbaustein MpBackupCore
Ereignisse	
idx	Liefert den Index des aktuell generierten Ereignisses. Der Wert wird bei jedem neuen Ereignis inkrementiert. Das erste generierte Ereignis in der Ereignisliste hat somit den Wert 1, dass fünfzigste Ereignis hat den Index 50, das hundertste den Index 100 und so weiter.
op	Name des Benutzers welcher beim Auftreten des Ereignisses eingeloggt war
ev	Ereignis ID
map	Name der mapp Komponente, welcher das Ereignis über den MpLink zugeordnet ist
dev	Speichermedium welches verwendet wurde (z.B. USB Stick, von welchem ein Backup geladen wurde)
file	Dateiname des Backups
msg	Gibt die Aktion als Ereignis-Text zurück. Für mehr Information siehe folgende Tabelle: **Ereignis** / **Ereignis als Text (wird über %msg zurückgegeben)** Backup-Erstellung gestartet / StartCreate Backup-Erstellung abgeschlossen / EndCreate Backup wird installiert / StartInstall
act	Aktion, welche in einer mapp Komponente ausgeführt wurde. Die ID für die Aktion wird zurückgegeben (z.B. 1 für "Backup-Erstellung abgeschlossen"). Für mehr Information siehe folgende Tabelle: **Ereignis** / **Identifikationsnummer (wird über %act zurückgegeben)** Backup-Erstellung gestartet / 0 Backup-Erstellung abgeschlossen / 1 Backup wird installiert / 2

3.1.5 Konfiguration während der Laufzeit

Die Konfigurations Parameter können zur Laufzeit über MpBackupCoreConfig verändert werden. Dabei ist für die unterschiedlichen Parameter jedoch zu beachten in welchem Zustand sich der Funktionsbaustein befinden muss um eine Änderung zuzulassen. Folgende Zustände sind möglich:

- : Dieses Symbol bedeutet, dass dieser Parameter in jedem Zustand geändert werden kann
- : Dieses Symbol bedeutet, dass dieser Parameter mit einer neue positiven Flanke am "Enable" Eingang von MpBackupCore übernommen wird.

Jeder Parameter der Konfigurations Struktur (MpBackupCoreConfigType) ist mit einem dieser Symbole ausgestattet.

3.1.6 Rahmenbedingungen

In diesem Abschnitt werden die Rahmenbedingungen von mapp Backup aufgelistet

- Backups können in Simulation nicht erstellt oder installiert werden.
- Läuft auf dem Zielsystem ARWin so kann kein Backup erstellt oder installiert werden
- Um mapp Backup verwenden zu können muss AR >= F4.34 installiert werden

- Es darf keine Installation eines Backups durchgeführt werden, wenn sich dieses am internen Speicher (z.B. Compact Flash oder eingebauter Flash-Speicher) befindet
- Wird ein Update/Backup auf einem B&R Hypervisor installiert, so gilt es zu beachten, dass bei der Verwendung eines UEFI Systems keine Umpartionierung möglich ist.

3.2 Getting Started

3.2.1 Erstellen eines Backups

In diesem Abschnitt wird Schritt für Schritt beschrieben wie man mithilfe von MpBackupCore ein Backup erstellen kann.

Ziel ist es die Daten der Maschinenapplikation in einem Backup zu sichern.

In Simulation kann kein Backup erstellt oder installiert werden!

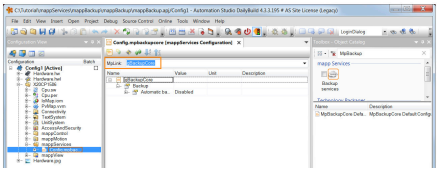
3.2.1.1 Projekt erstellen

Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

3.2.1.2 mapp Komponente einfügen

Konfiguration MpBackupCore einfügen

Um ein Backup erstellen zu können, wird die Konfiguration MpBackupCore eingefügt.



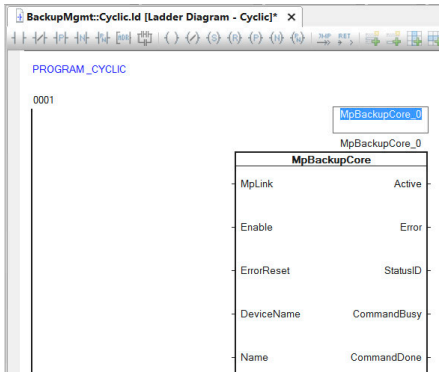
Hier befindet sich der mapp Link, welcher die mapp Komponente in der „world of mapp“ eindeutig identifiziert. Der mapp Link stellt die Verbindung zwischen Programmierung und Konfiguration her.

3.2.1.3 Programm hinzufügen

Der nächste Schritt ist das Einfügen eines Kontaktplan-Programms. Für mehr Information siehe hier.

3.2.1.4 MpBackupCore einfügen

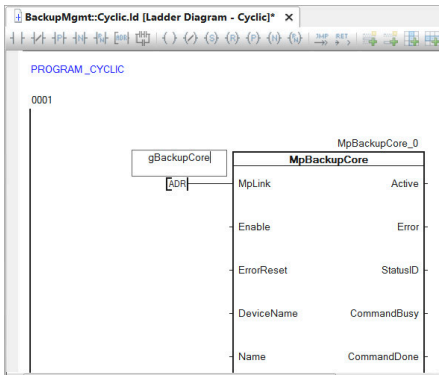
Im nächsten Schritt wird der Funktionsbaustein MpBackupCore eingefügt. Mithilfe von MpBackupCore können Backups erstellt oder installiert werden.



3.2.1.5 Parametrieren von MpBackupCore

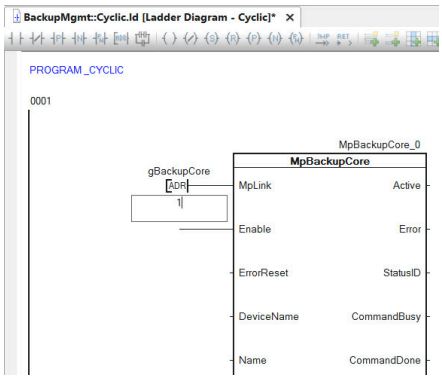
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gBackupCore erstellt haben.



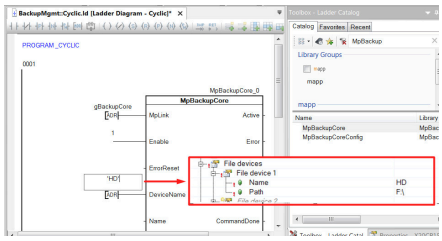
Aktivieren des Funktionsbausteins

Der Funktionsbaustein wird durch den Eingang "Enable" aktiviert.



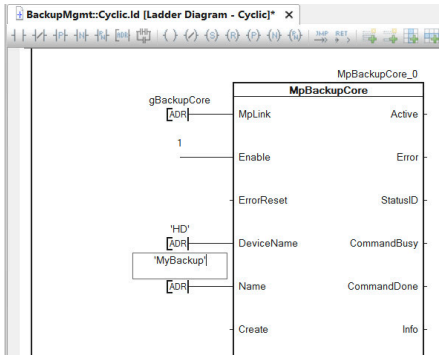
Speichermedium definieren

Damit das Backup abgespeichert werden kann, muss ein Speichermedium angegeben werden. Dieses wurde in der Konfiguration der Hardware, mit dem Namen HD erstellt.



Name für Backup definieren

Zum Schluss muss der Name des zu erstellenden Backups definiert werden. Es wird "MyBackup" angegeben.



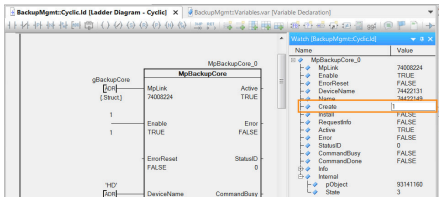
3.2.1.6 Übertragen des Projekts

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

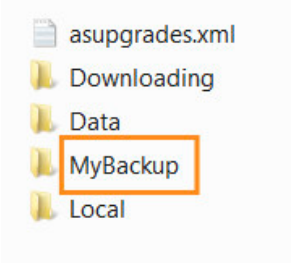
3.2.1.7 Testen des Programms

Nach einem Download der Änderungen kann das Programm getestet werden. Der Monitor-Modus wird aktiviert.

Über den Befehl "Create = `TRUE`" wird ein Backup erstellt.



Das Backup wird am angegebenen Speichermedium (In diesem Fall in der Benutzerpartition) abgelegt. Welche Daten das Backup beinhaltet kann im Abschnitt Aufbau eines Backups nachgelesen werden.



3.3 Konfiguration

3.3.1 MpBackupCore Konfiguration

> **Die Konfiguration kann zur Laufzeit mithilfe des Funktionsbausteins MpBackupCoreConfig angepasst werden.**

Automatisches Backup

In der MpBackupCore Konfiguration kann über "Automatic backup" definiert werden, ob ein Backup automatisch erstellt werden soll oder nicht.

Wie ein Backup automatisch erstellt werden kann, ist im Anwendungsfall Wöchentliches erstellen eines Backups erklärt.

Soll ein Backup automatisch erstellt werden, so können folgende Parameter konfiguriert werden:

- **Name prefix:** Definiert den Namen des Backups. Es wird automatisch nach dem Namen ein Zeitstempel angehängt.Das kann zum Beispiel folgendermaßen aussehen: "myPrefix_%Y_%m_%d_%H_%M_%S"
- **Device name:** Angabe des Speichermediums, wo das Backup abgelegt werden soll.
- **Mode:** Definiert in welchen Zeitintervall das Backup abgespeichert werden soll. Das Backup kann wöchentlich oder täglich abgespeichert werden. Soll die Datei wöchentlich abgespeichert werden, so wird über "Day" der Tag definiert. Time legt fest um welche Uhrzeit das Backup erstellt werden soll. Die Zeit wird in Sekunden beginnend von 00:00 angegeben. Soll die Datei um 06:00 morgens abgespeichert werden, so muss eine Zeit von 21600 s angegeben werden.
- **Overwrite oldest:** Definiert ob ältere Backups gelöscht werden sollen oder nicht. Die maximale Anzahl an Dateien wird dabei über den Parameter "Maximum number of backups" definiert.

Die Konfiguration kann zur Laufzeit über den Funktionsbaustein MpBackupCoreConfig verändert werden.

> **Wird Option "Override oldest" zur Laufzeit von MpBackupCoreConfig aktiviert und es befinden sich am angegebenen Speichermedium 30 Backups, "MaximumNumberOfBackups" aber nur 10 Backups erlaubt, so werden die ältesten 20 Backups gelöscht.**

Automatisches Update

Über "Automatic Update" kann definiert werden, ob automatisch nach neuen Updates gesucht werden soll oder nicht. Dabei wird die Konfigurations-Version geprüft. Ist die Konfigurations-Version des Updates höher als die des Projekts welches aktuell auf dem Zielsystem läuft, so wird dies in der Info-Struktur unter "Automatic" angezeigt. Ist der Parameter "IsAvailable = `TRUE`" kann das Update mithilfe von MpBackupCore installiert werden. Der zu verwendende Update-Name ist im Datentypen MpBackupAutoUpdateInfoType über "Name" ersichtlich.

Wie ein Update automatisch angezeigt werden kann, ist im Anwendungsfall Update für Maschinensoftware automatisch anzeigen erklärt.

Für mehr Information siehe Abschnitt Installieren eines Updates.

Aktiviert man die Funktion, so können folgende Parameter eingestellt werden:

- **Device name:** Unter "Device name" kann das Speichermedium angegeben werden, welches zuvor in der Konfiguration des Zielsystems definiert wurde. Bei Bedarf kann zusätzlich noch der gewünschte Pfad definiert werden, wie zum Beispiel "DEV1/Path1/Folder2". In diesem Speichermedium/Pfad wird nach neuen Updates gesucht.
- **Check:** Unter "Check" wird festgelegt wann nach Updates gesucht werden soll. Dabei gibt es 3 Auswahlmöglichkeiten.
  Daily: Es wird täglich nach Updates gesucht. Dabei muss unter "Time" der Zeitpunkt gewählt werden. Die Zeit wird in Sekunden beginnend von 00:00 Uhr bestimmt. Das heißt soll nach einem Update um 06:00 Uhr morgens gesucht werden, so wird 21600 Sekunden angegeben.
  Weekly: Es kann auch auf wöchentlicher Basis nach Updates gesucht werden. Dabei wird der Tag und die Zeit bestimmt. Die Zeit wird in Sekunden beginnend von 00:00 Uhr bestimmt. Das heißt soll nach einem Update um 06:00 Uhr morgens gesucht werden, so wird 21600 Sekunden angegeben.
  On enable: Sobald der Funktionsbaustein MpBackupCore aktiviert ist, wird automatisch einmalig nach einem Update gesucht. Das heißt, bei jeder positiven Flanke am Eingangsparameter "Enable" wird einmal nach einem Update gesucht.

Auditing

Unter "Auditing" kann festgelegt werden, ob Ereignisse, wie beispielsweise das Erstellen von Backups, von mapp Audit aufgezeichnet werden sollen. Für mehr Information siehe Auditing von mapp Backup.

3.4 Anwendungsfälle

3.4.1 Wöchentliches erstellen eines Backups

Anforderung

Die aktuelle Applikation samt Benutzerdaten einer Maschine soll aus Sicherheitsgründen wöchentlich auf einem Speichermedium, wie zum Beispiel einen USB-Stick, abgespeichert werden.

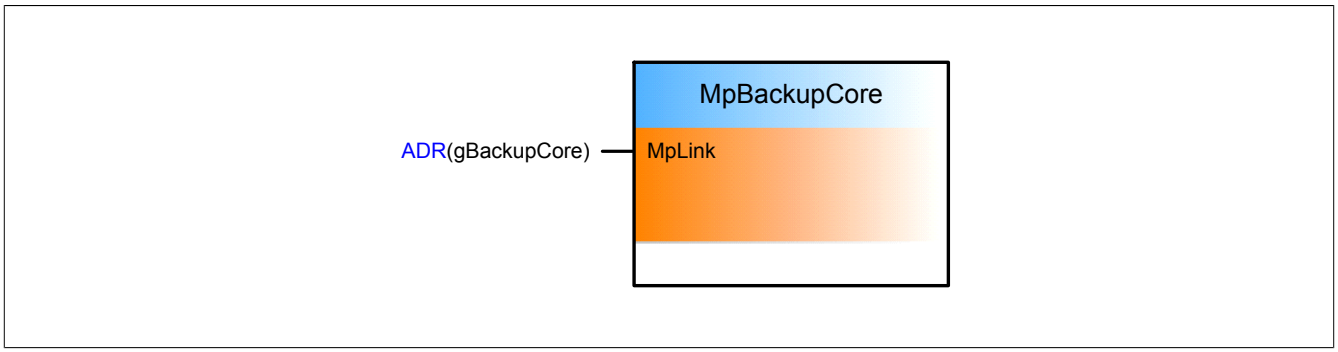
Die Sicherungsdatei soll dabei automatisch jeden Freitag erstellt und abgespeichert werden.

Lösung

Komponentenliste

- ***MpBackupCore*** (Eigener MpLink): Erstellt die Sicherungsdatei und speichert es auf ein angegebenes Speichermedium.

Verbindungsschema



Konfiguration

Um eine Sicherungsdatei zu erstellen, wird die Konfiguration MpBackupCore eingefügt.

Damit das Backup wöchentlich erstellt wird, wird der Parameter "Automatic backup" auf `TRUE` gesetzt. Der Name der Sicherungsdatei wird über "Name prefix" definiert. Beim erstellen der Datei wird am Namen automatisch ein Zeitstempel angehängt. Um den Speicherort der Datei zu definieren, wird an "Device Name" das gewünschte Speichermedium angegeben. Dieses muss zuvor in der Konfiguration des Zielsystems angegeben werden.

Damit das Backup automatisch wöchentlich erstellt wird, muss der Parameter "Mode" auf "Weekly" gesetzt werden. Anschließend kann über "Day" und "Time" der gewünschte Tag und die gewünschte Uhrzeit ausgewählt werden. Die Uhrzeit wird in Sekunden, beginnend von 00:00 Uhr angegeben. Soll das Backup um 06:00 Uhr in der Früh erstellt werden, so wird die Zeit 21600 angegeben.

Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpBackupCore** eingefügt und parametriert.

Es wird, durch die vorher vorgenommenen Einstellungen in der MpBackupCore Konfiguration, automatisch eine Sicherungsdatei am angegebenen Speichermedium abgelegt.

3.4.2 Installieren einer neuen Software-Version an der Maschine

Anforderung

Für eine Maschine, welche bereits mehrere Monate im Einsatz ist, soll ein Software-Update durchgeführt werden.

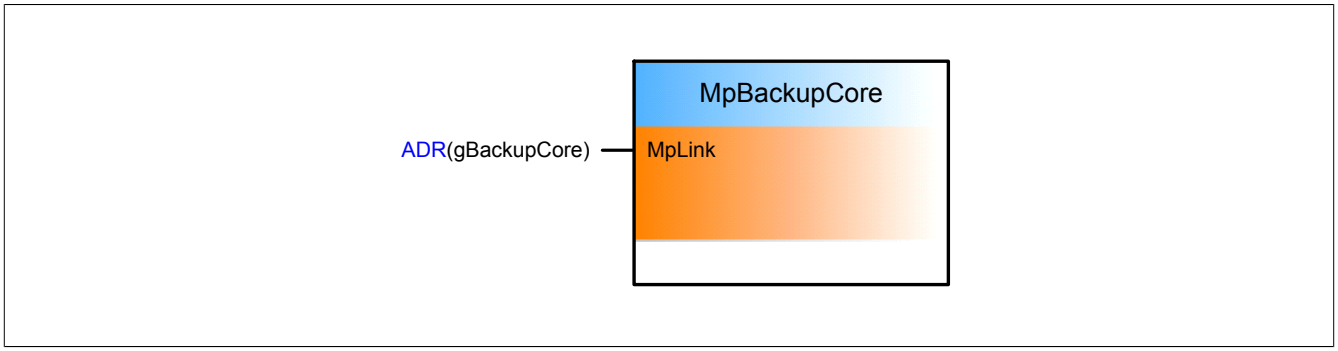
Das Software-Update soll mithilfe eines USB-Sticks an der Maschine installiert werden.

Lösung

Komponentenliste

- ***MpBackupCore*** (Eigener MpLink): Erstellt und installiert Backups bzw. Updates.

Verbindungsschema

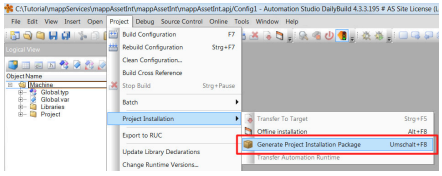


Konfiguration

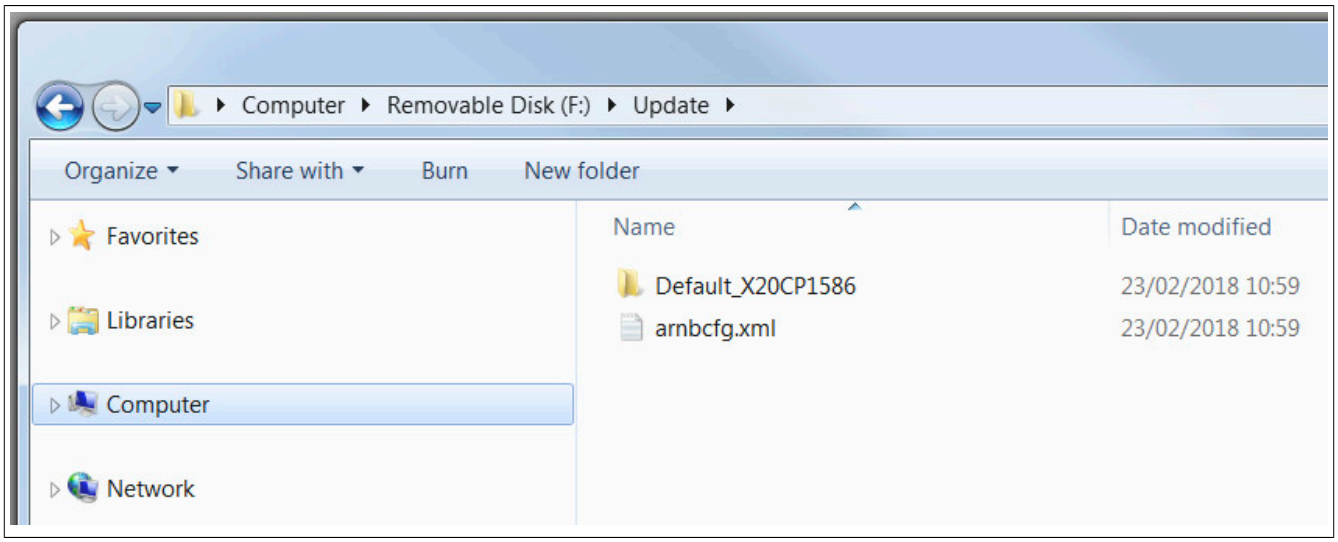
Damit ein Software-Update durchgeführt werden kann, muss die Konfiguration MpBackupCore eingefügt werden.

Softwareupdate

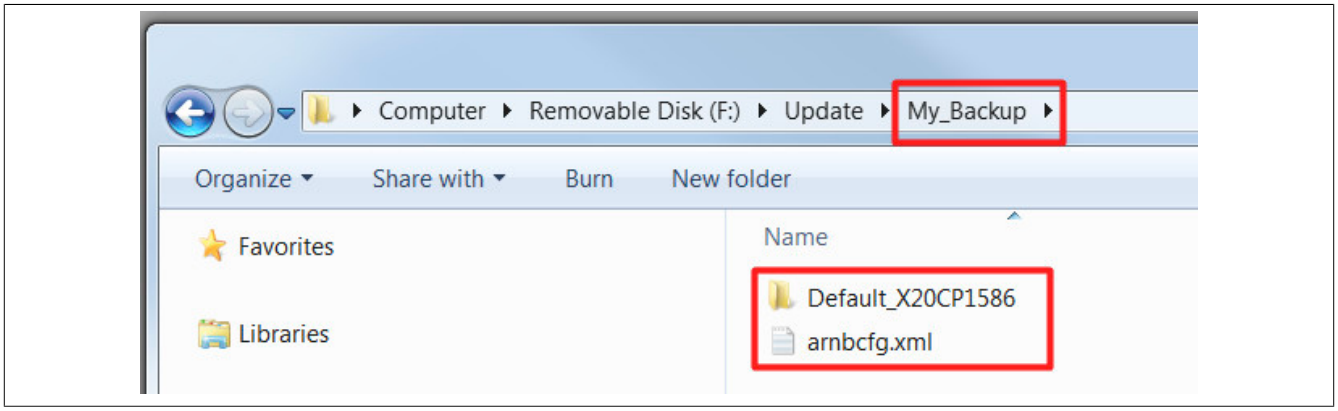
Das Projekt über Automation Studio angepasst. Damit die neue Software auf die Maschine übertragen werden kann, wird ein Projektinstallationspakets erstellt. Für mehr Information wie ein Projektinstallationspaket erstellt und verwendet wird, siehe hier.



Wurde das Projektinstallationspaket erfolgreich erstellt, so sind die Daten auf einem Speichermedium, wie zum Beispiel einen USB-Stick, zu finden:



Damit die Daten nun von mapp Backup verwendet werden können, müssen die Daten in einen Ordner kopiert werden. Die Daten wurden in den Ordner "My_Backup" kopiert. Die Daten werden auf einen USB-Stick übertragen.



Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpBackupCore** eingefügt und parametriert.

Über eine Visualisierung kann mit **MpBackupCore** interagiert werden.

Am Eingangsparameter "Name" wird der Name des vorher erstellten Ordners "My_Backup" angegeben. Die Datei liegt auf einem USB-Stick welcher an der Maschine angeschlossen ist.

Über das Kommando "Install = `TRUE`" wird das Update für die Maschine installiert.

3.4.3 Wiederherstellen der Machinensoftware auf einer neuen Steuerung

Anforderung

Die Steuerung einer Maschine wurde beschädigt und muss gewechselt werden. Es existiert ein Backup der vorhergehenden Steuerung. Das Backup wurde, wie hier erklärt, erstellt. Dieses Backup soll auf die neue Steuerung mithilfe eines USB-Sticks übertragen werden.

Wiederherstellen der neuen Steuerung anhand eines Backups

Das vorhandene Backup der vorhergehenden Steuerung soll auf die neue Steuerung übertragen werden. Abhängig vom verwendeten Zielsystem und der Runtime-Version des Zielsystems gibt es verschiedene Lösungswege. Hier sind drei verschiedene Lösungswege aufgelistet. Für die aufgelisteten Lösungswege wird eine Steuerung **ohne** Compact Flash Karte verwendet.

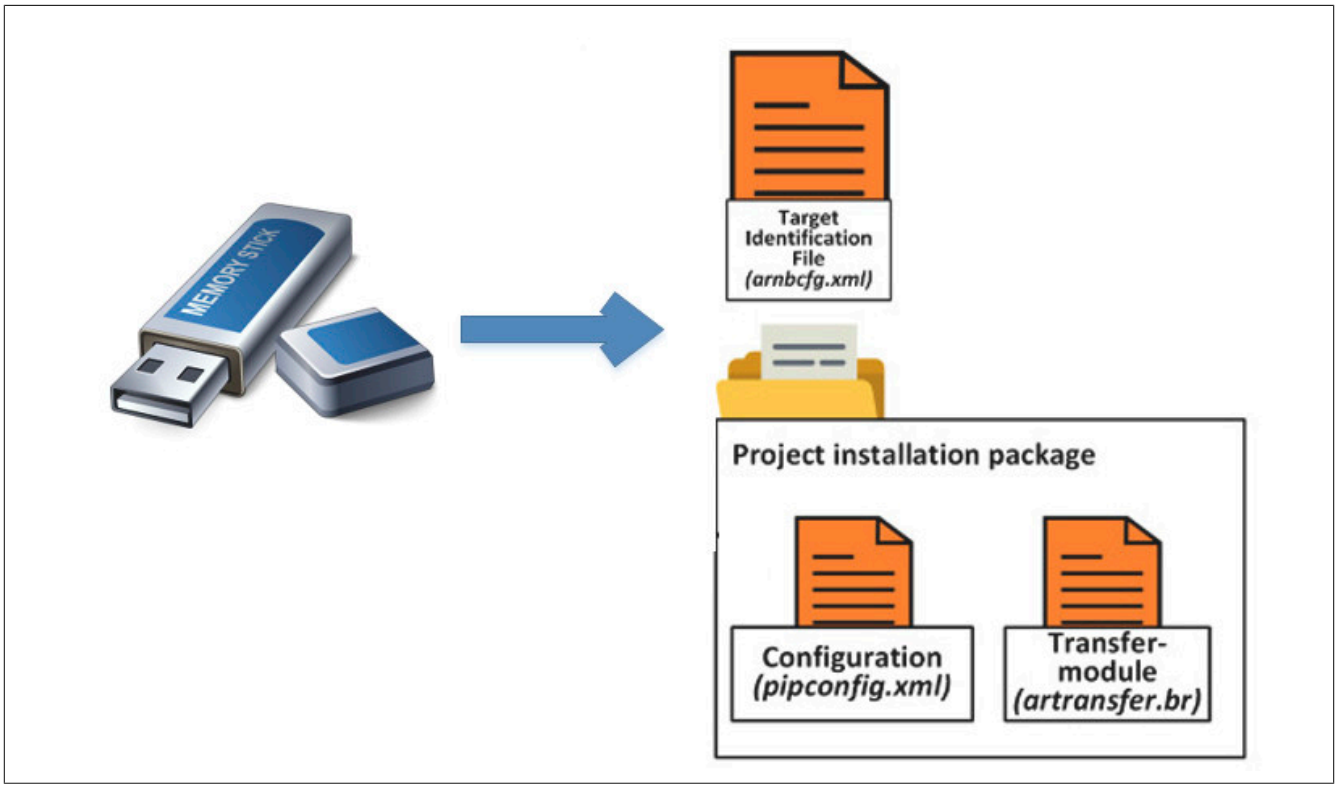
- Steuerung mit Default AR >= 4.33
- Steuerung mit default AR < 4.33
- Steuerung mit BIOS und ohne default AR

Um herauszufinden welche Default Automation Runtime Version das Zielsystem verwendet, kann man sich zum Beispiel mit der Steuerung über Automation Studio verbinden. Für mehr Information siehe hier.

3.4.3.1 Steuerung mit Default AR >= 4.33

USB-Stick mit Backup vorbereiten:

Damit das Backup am Zielsystem installiert werden kann, muss sich der Inhalt des Backups auf einem USB-Stick befinden. Die Backup-Dateien dürfen sich in keinen Unterordner befinden. Das bedeutet das "Target Identification File" Datei muss sich im Root-Verzeichnis des USB-Stick befinden.



Backup installieren

Der USB-Stick wird am Zielsystem angeschlossen

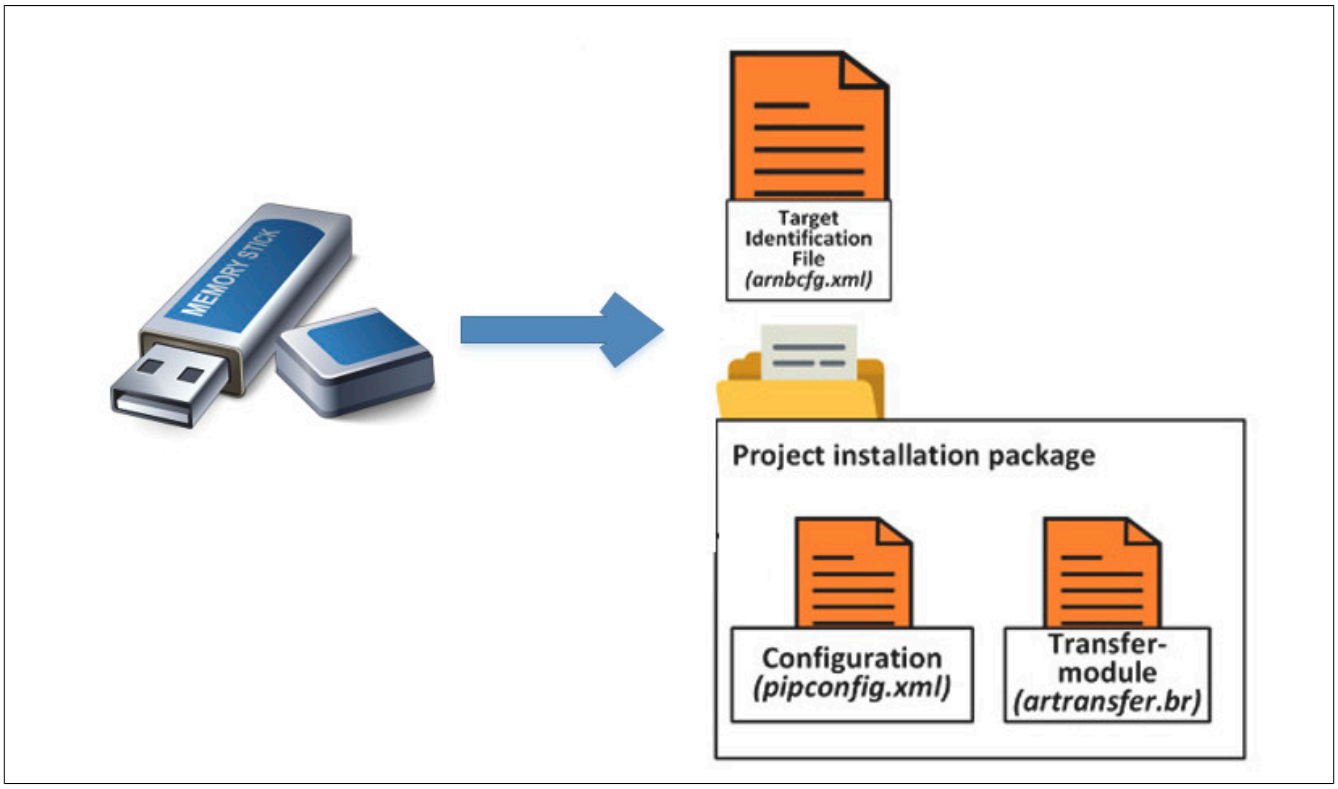
Damit das Backup vom USB-Stick installiert werden kann, muss die Steuerung im Betriebsmodus BOOT sein. Dies funktioniert zum Beispiel mit dem Betriebsmodus-Schalter oder der Reset-Taste der Steuerung. Das Backup wird automatisch auf der Steuerung installiert.

3.4.3.2 Steuerung mit default AR < 4.33

Ist auf der Steuerung eine Default AR Version < 4.33, so müssen folgende Vorbereitungen getroffen werden bevor das Backup installiert werden kann:

USB-Stick mit Backup vorbereiten:

Damit das Backup am Zielsystem installiert werden kann, muss sich der Inhalt des Backups auf einem USB-Stick befinden. Die Backup-Dateien dürfen sich in keinen Unterordner befinden. Das bedeutet das "Target Identification File" Datei muss sich im Root-Verzeichnis des USB-Stick befinden.

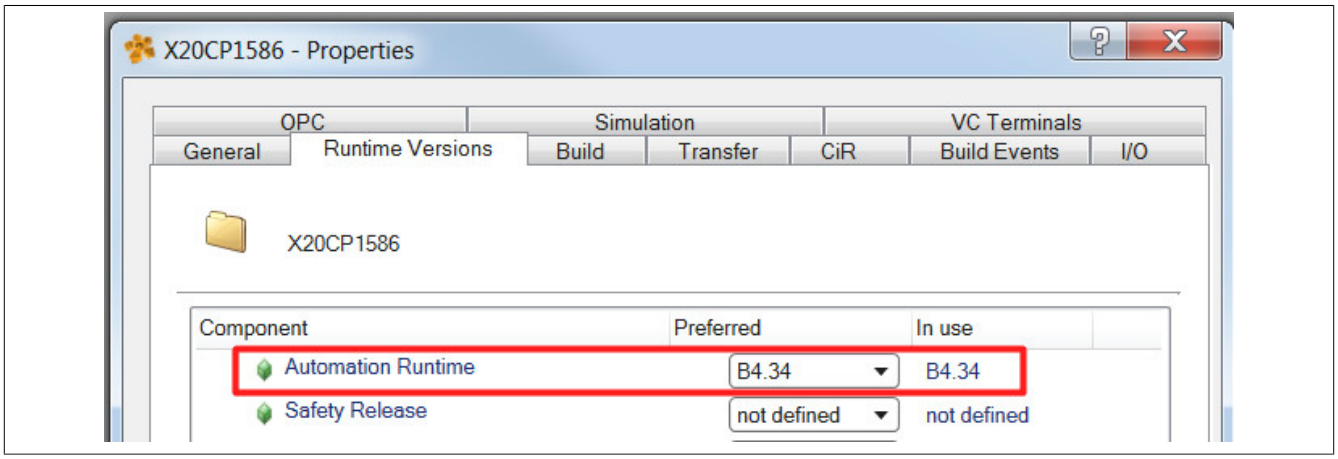


Projektinstallationspaket erstellen

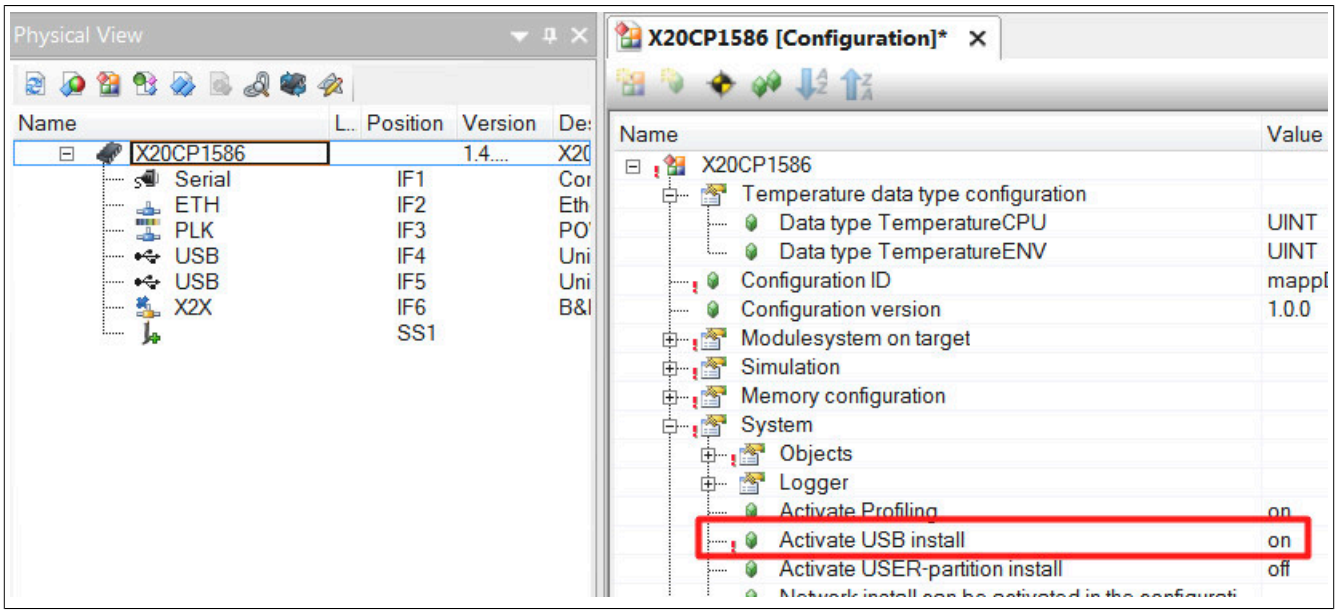
Um das Backup zu installieren, wird die AR Version >= 4.33 benötigt.

Damit das Zielsystem die benötigte Runtime Version hat, wird ein Projektinstallationspaket erstellt.

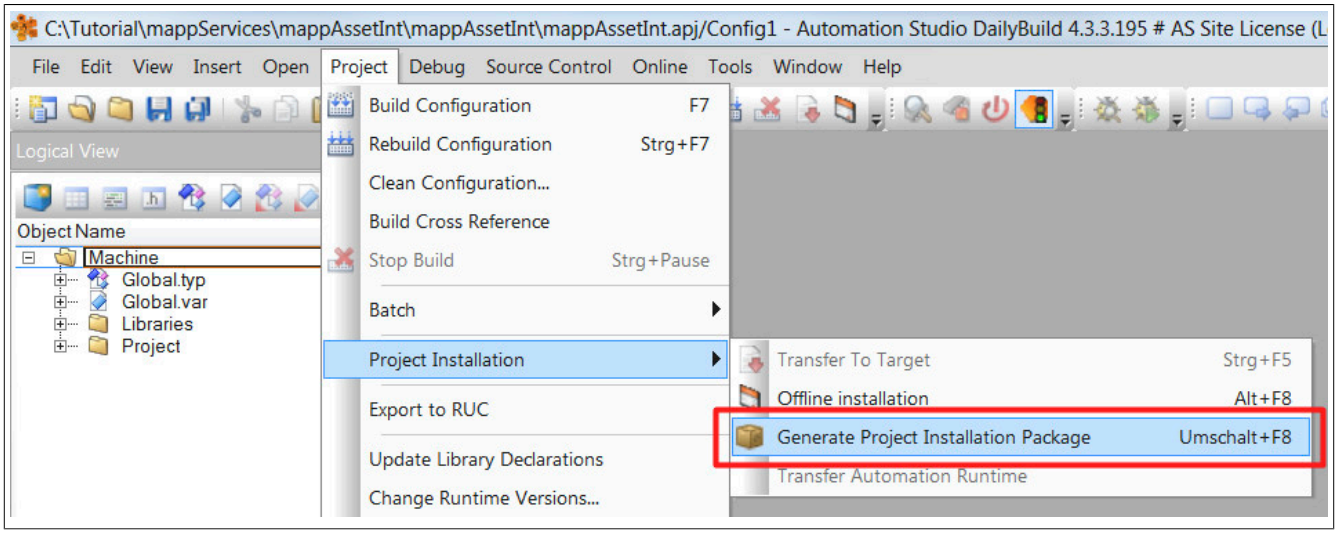
Das bedeutet es wird in Automation Studio ein Projekt erstellt, welches die Hardware des verwendeten Zielsystem besitzt. Außerdem muss im Projekt eine Runtime-Version >= 4.33 eingestellt sein.



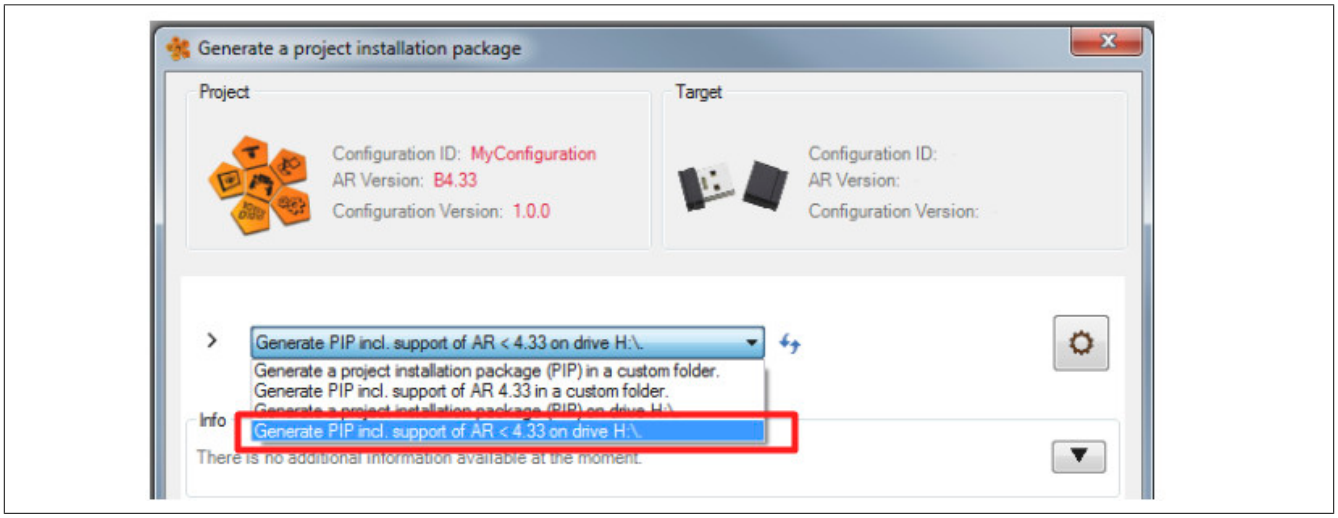
Damit das Backup später über einen USB-Stick installiert werden kann, wird zum Schluss noch die Option "Activate USB-Stick install" in der Konfiguration des Zielsytems aktiviert.



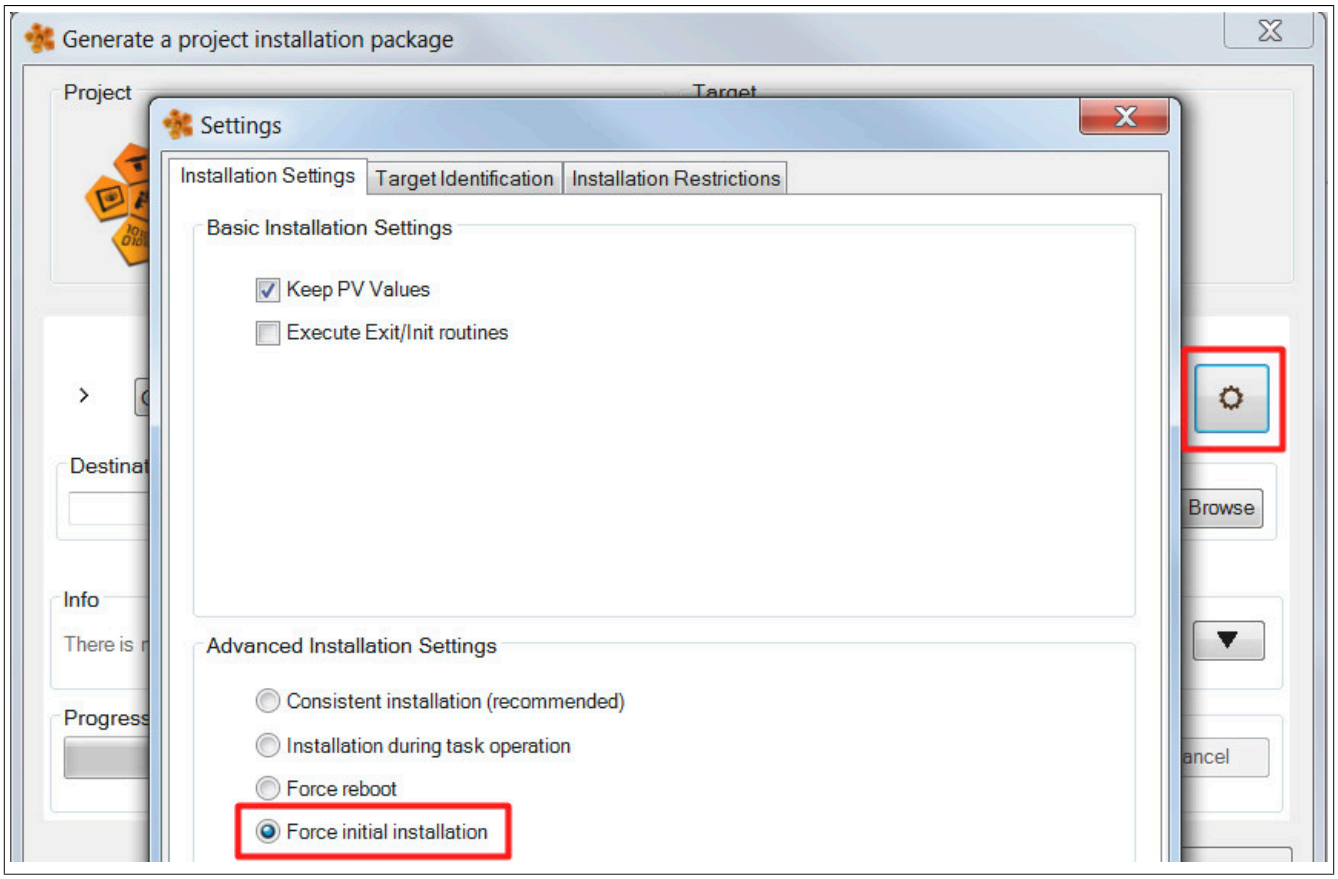
Über "Generate Project installation package", kann nun ein Projektinstallationspaket erstellt werden. Für mehr Information siehe hier.



Wichtig hierbei ist, dass die Option "Generate PIP incl. support of AR < 4.33 ..." verwendet wird. Nur so kann das Projektinstallationspaket am Zielsystem installiert werden.



In den Transfereinstellungen kann angegeben werden wie das Paket anschließend auf dem Zielsystem installiert werden soll. Es wird empfohlen die Option "Force initial transfer" auszuwählen.



Die Daten welche durch das Projektinstallationspaket erstellt wurden, werden auf einen USB-Stick übertragen und am Zielsystem angschlossen. Wichtig hierbei ist, dass sich die Daten im Root-Verzeichnis des USB-Sticks befinden, das heißt die Daten dürfen sich in keinem Unterordner befinden.

Damit die Daten vom USB-Stick installiert werden können, muss die Steuerung im Betriebsmodus BOOT sein. Dies funktioniert zum Beispiel mit dem Betriebsmodus-Schalter der Steuerung. Das Projektinstallationspaket wird nun automatisch auf der Steuerung installiert. Ist das Paket installiert, so kann der USB-Stick vom Zielsystem entfernt werden.

Backup installieren

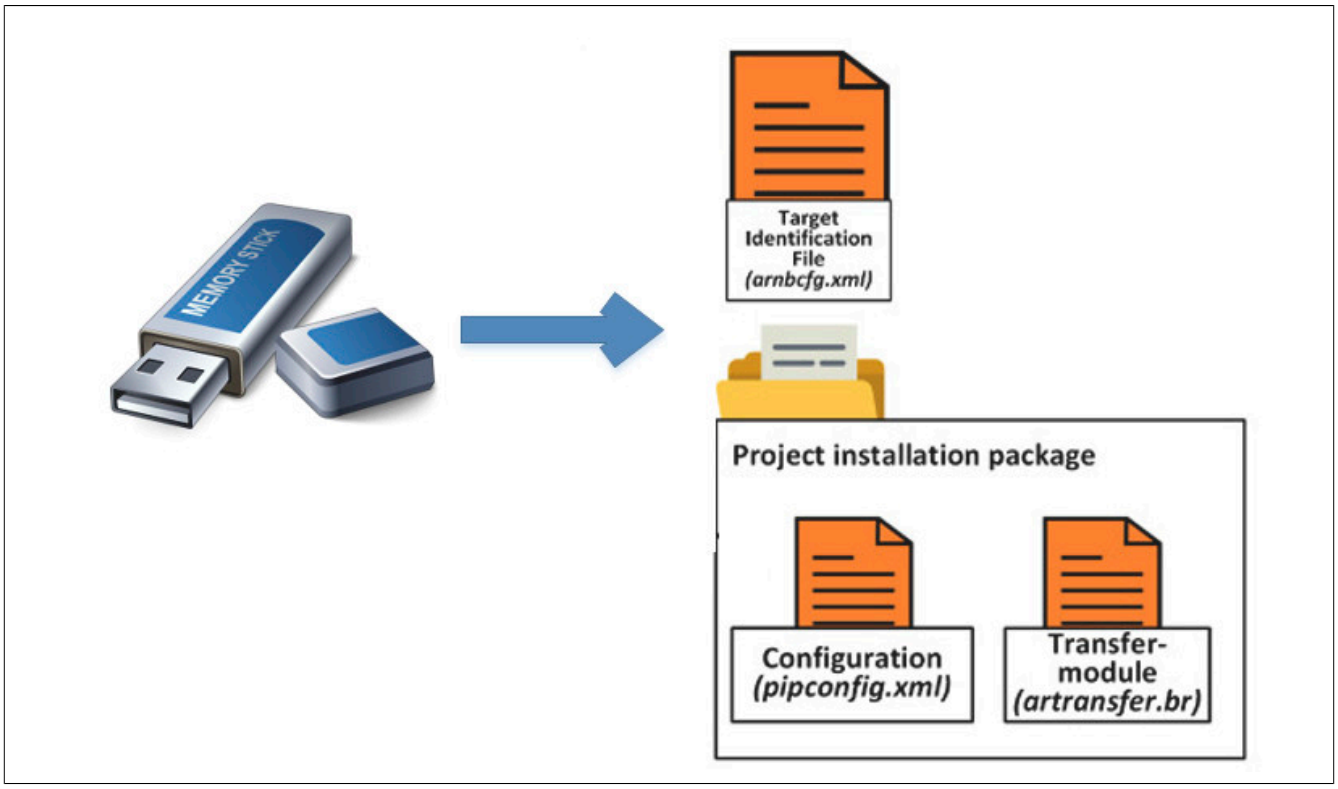
Hat das Zielsystem nun die AR-Version >= 4.33 kann der USB-Stick am Zielsystem angeschlossen werden, welcher das zu installierende Backup beinhaltet. Damit die Daten installiert werden, muss die Steuerung neu gestartet werden. Dadurch, dass die Option "Activate USB install" vorher aktiviert wurde, wird beim Hochfahren der Steuerung das Backup am USB-Stick installiert.

3.4.3.3 Steuerung mit BIOS und ohne default AR

Ist auf der Steuerung ein BIOS und keine default Automation Runtime vorhanden, so müssen folgende Vorbereitungen getroffen werden:

USB-Stick mit Backup vorbereiten:

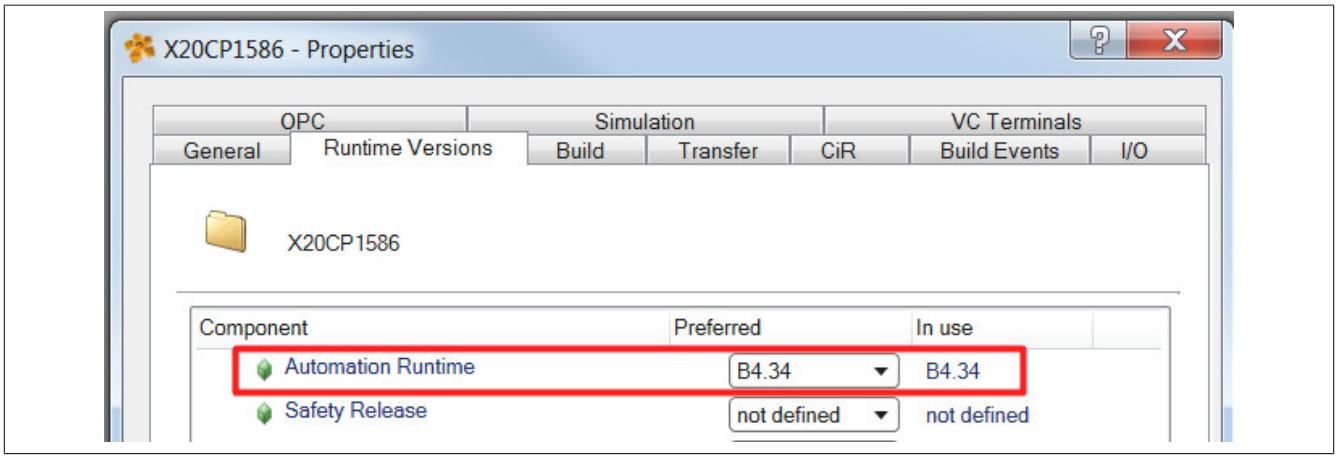
Damit das Backup am Zielsystem installiert werden kann, muss sich der Inhalt des Backups auf einem USB-Stick befinden. Die Backup-Dateien dürfen sich in keinen Unterordner befinden. Das bedeutet das "Target Identification File" Datei muss sich im Root-Verzeichnis des USB-Stick befinden.



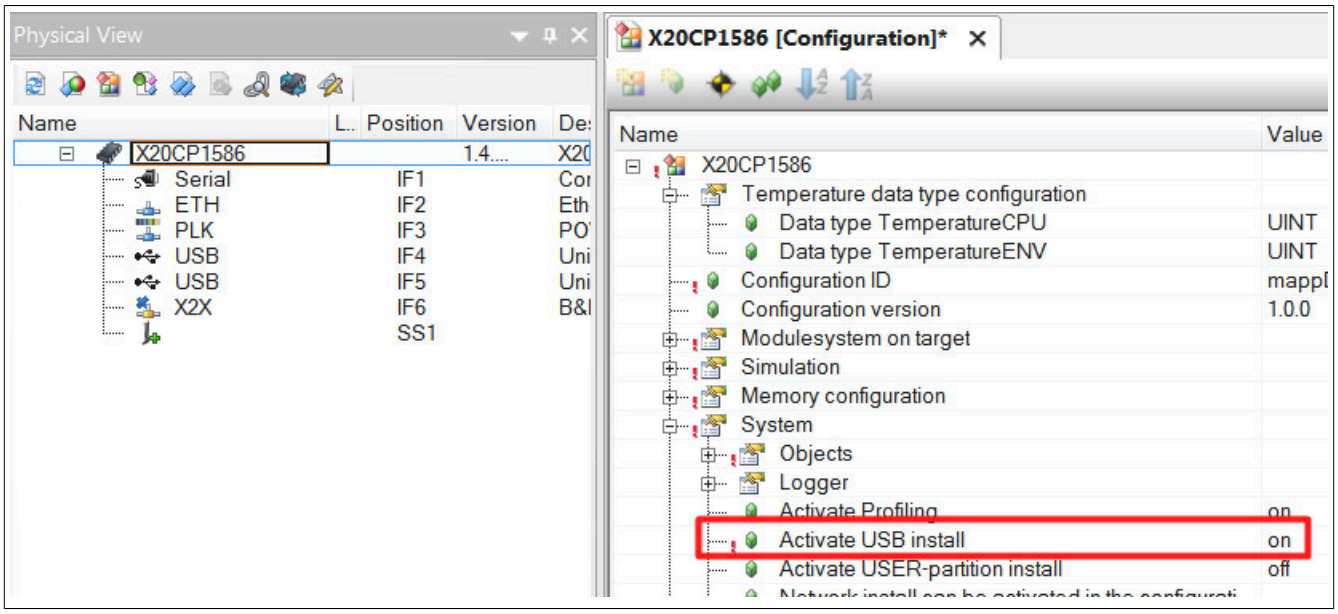
Offline Install erstellen

Um das Backup zu installieren, wird die AR Version >= 4.33 benötigt.

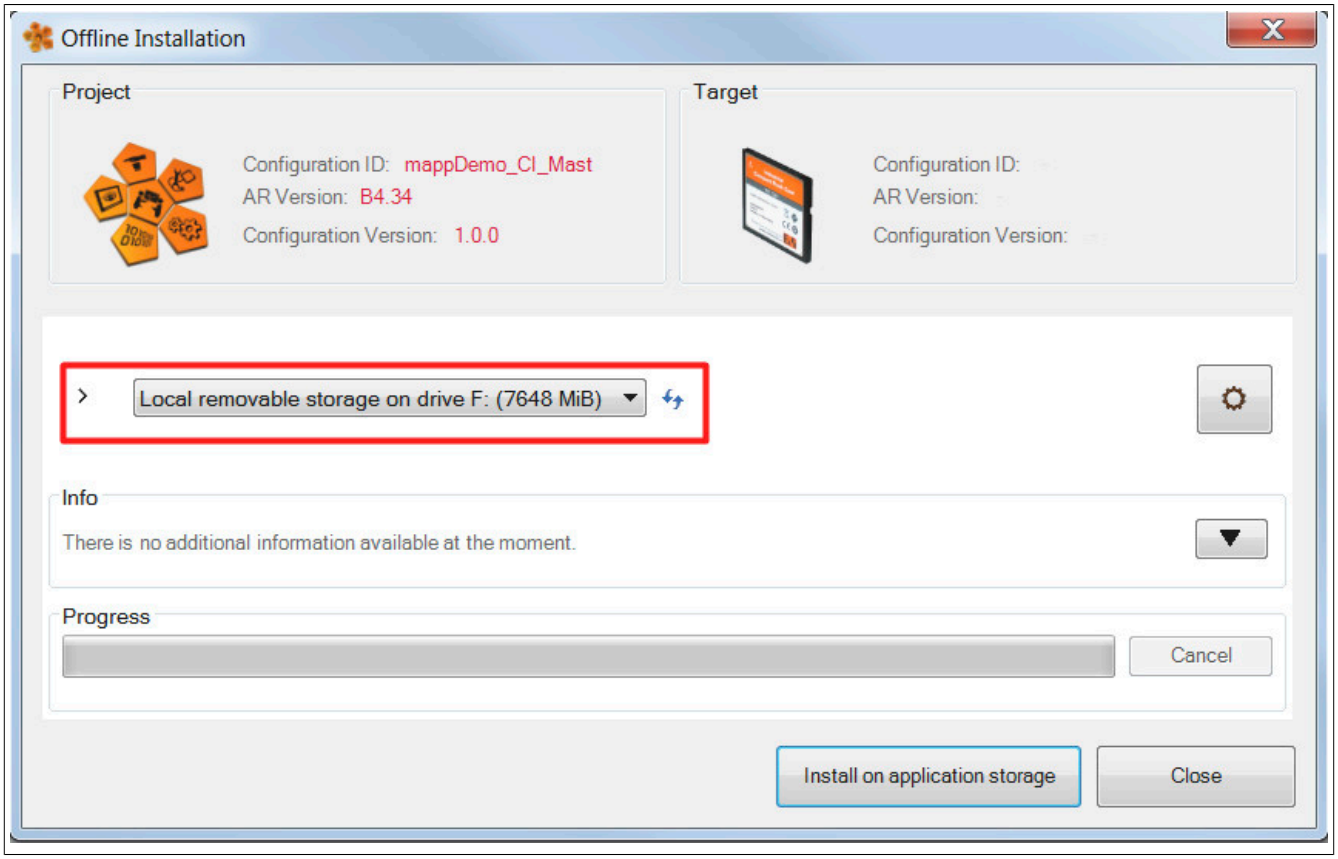
Das bedeutet es wird in Automation Studio ein Projekt erstellt, welches die Hardware des verwendeten Zielsystem besitzt. Außerdem muss im Projekt eine Runtime-Version >= 4.33 eingestellt sein.



Damit das Backup später über einen USB-Stick installiert werden kann, wird zum Schluss noch die Option "Activate USB-Stick install" in der Konfiguration des Zielsytems aktiviert.



Damit die neue Runtime Version am Zielsystem übertragen werden kann muss ein Offline Install durchgeführt werden. Die durch den Offline Install generierten Daten müssen auf einen USB-Stick abgespeichert werden:



Der USB-Stick mit den Daten wird anschließend am Zielsystem angeschlossen.

Das BIOS am Zielsystem muss anschließend so eingestellt werden, dass der Boot-Vorgang vom angesteckten USB-Stick gestartet wird. Dadurch wird das Projekt, welches sich am USB-Stick befindet, installiert. Ist das Projekt installiert kann der USB-Stick vom Zielsystem entfernt werden. Die Einstellung im BIOS System kann wieder zurückgesetzt werden.

Backup installieren

Hat das Zielsystem nun die AR-Version >= 4.33 kann der USB-Stick am Zielsystem angeschlossen werden, welcher das zu installierende Backup beinhaltet. Damit die Daten installiert werden, muss die Steuerung neu gestartet werden. Dadurch, dass die Option "Activate USB install" vorher aktiviert wurde, wird beim Hochfahren der Steuerung das Backup am USB-Stick installiert.

3.4.4 Anwendungsfall 4: Update für Maschinensoftware automatisch anzeigen

Anforderung

Beim Hochstarten einer Maschine soll automatisch kontrolliert werden, ob neue Software-Updates vorhanden sind.

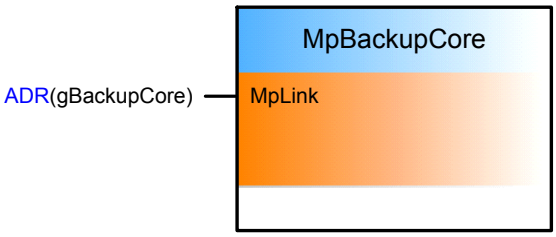
Sobald ein neues Update verfügbar ist, soll dies auf der Visualisierung angezeigt werden. Für den Service-Mitarbeiter soll es möglich sein das Update über die Visualisierung zu starten.

Lösung

Komponentenliste

- ***MpBackupCore*** (Eigener MpLink): Erstellt und installiert Backups bzw. Updates.

Verbindungsschema

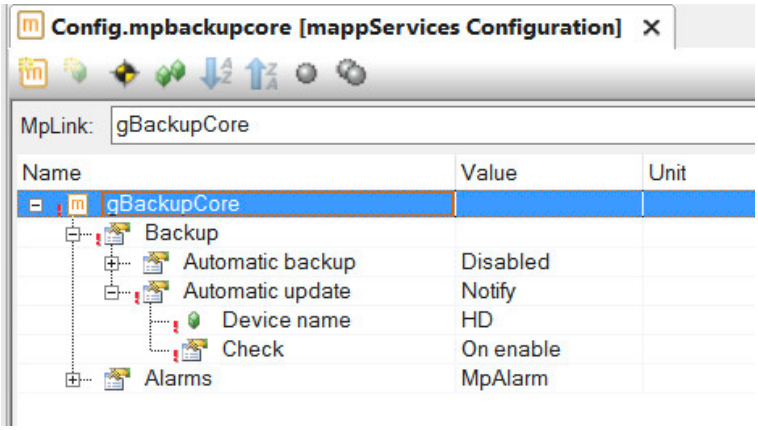


Konfiguration

Die Prüfung ob neue Software-Updates vorhanden sind, ist Teil von mapp Backup. Der erste Schritt ist dabei das Einfügen der MpBackupCore Konfiguration.

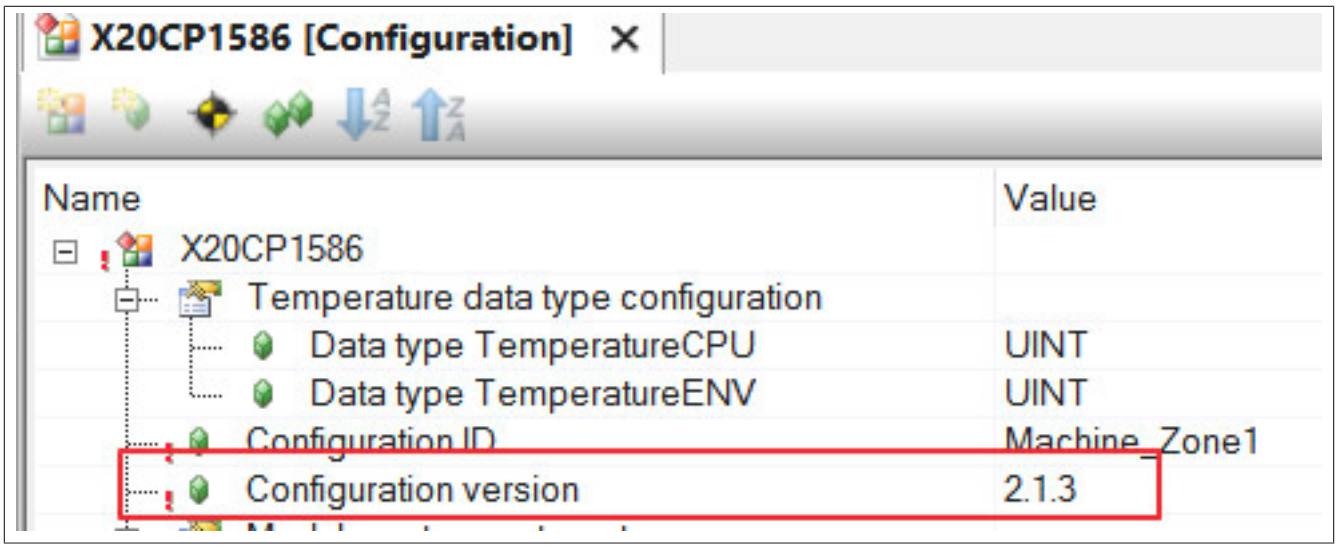
Die Option "Automatic update" wird auf "Notify" eingestellt. Damit beim Hochstarten der Maschine nach Updates gesucht werden kann, wird unter "Check" die Option "On enable" definiert. Sobald der Funktionsbaustein MpBackupCore aktiviert (Enable = `TRUE`) wird, wird automatisch nach Updates gesucht.

Unter "Device Name" wird das Speichermedium angegeben, auf dem nach Updates gesucht wird. Das Speichermedium wurde zuvor in der Konfiguration des Zielsystems definiert. Dies kann beispielsweise ein FTP Server oder ein USB Stick sein.

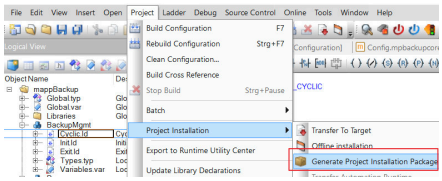


Softwareupdate vorbereiten

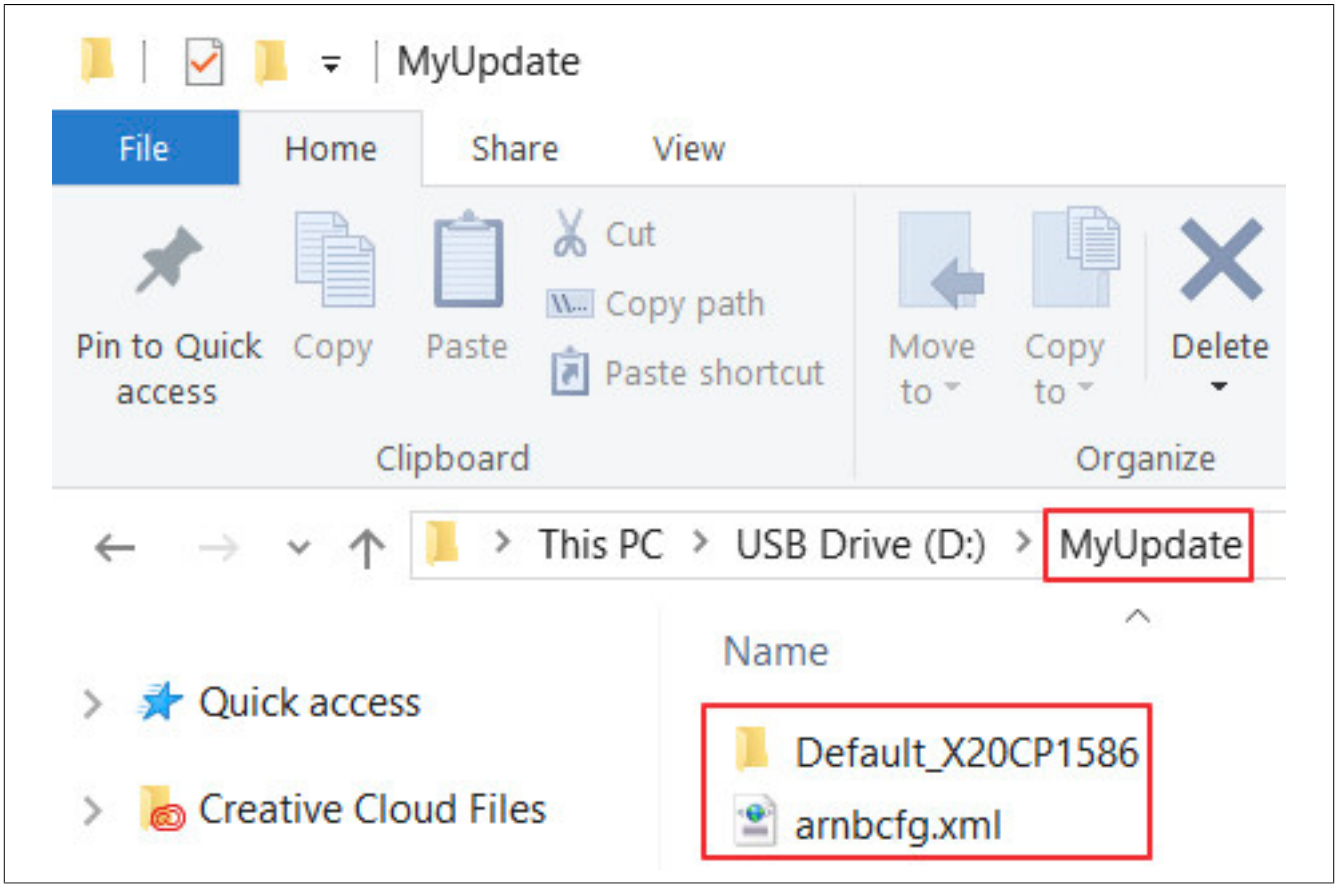
Im nächsten Schritt wird das Software-Update vorbereitet. Die gewünschten Änderungen werden in der Maschinenapplikation in Automation Studio durchgeführt. Damit das Software-Update automatisch angezeigt werden kann, muss die Konfigurations-Version angepasst werden. Es muss eine höhere Version angegeben werden. Diese Einstellung ist in der Konfiguration des Zielsystems zu finden.



Im Anschluss wird ein Projektinstallationspakets erstellt. Für mehr Information dazu, siehe hier.



Bei der Erstellung des PIP wird angegeben wo die Daten abgelegt werden sollen:



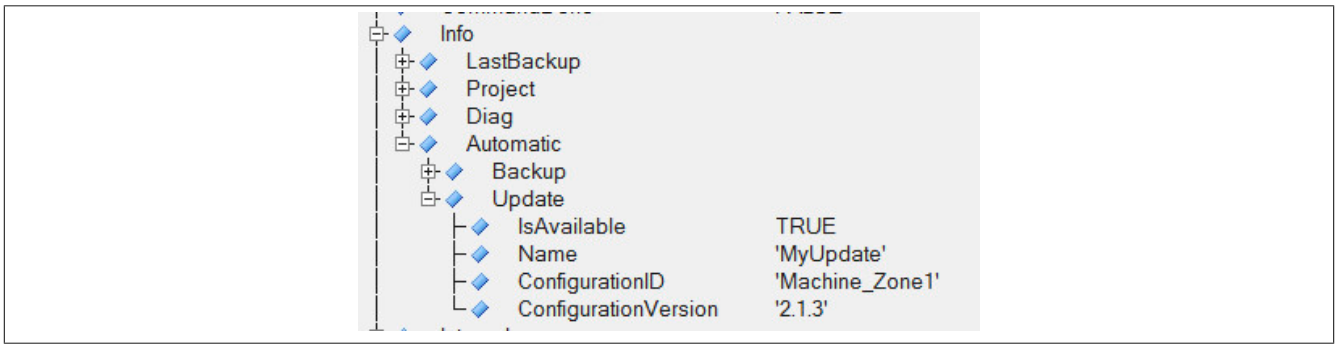
Die PIP-Daten werden in dem Ordner "MyUpdate" abgespeichert. Der Ordner wird am Speichermedium abgelegt, welches in der MpBackupCore Konfiguration unter "Automatic update" angegeben wurde.

Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpBackupCore** eingefügt und parametriert.

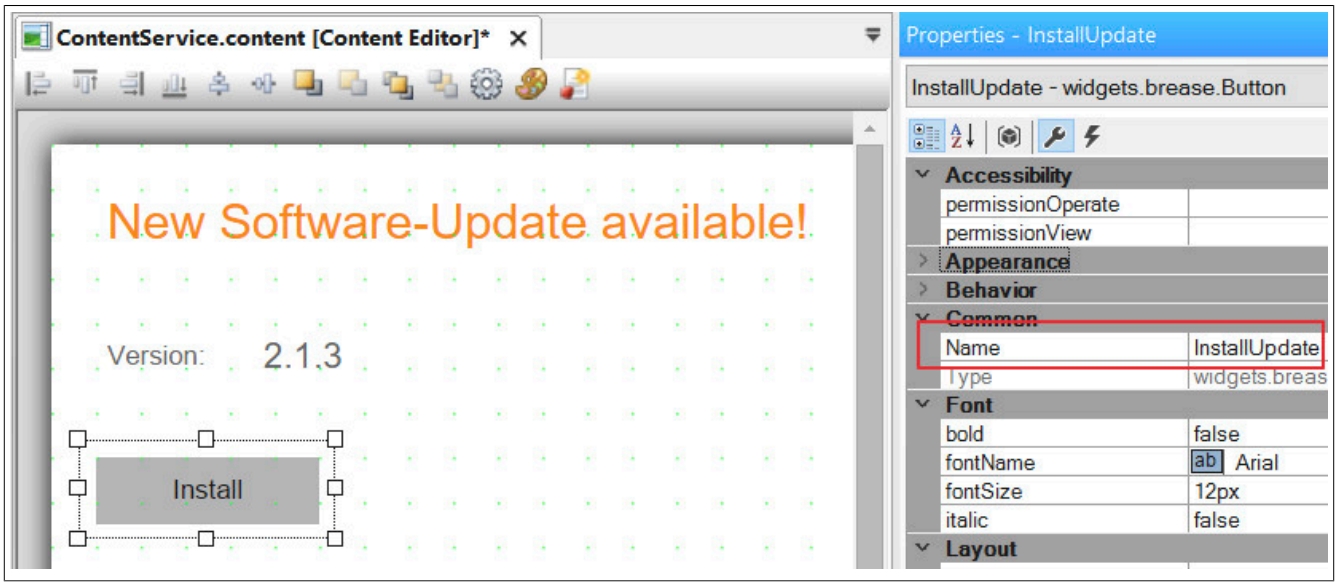
Über eine Visualisierung kann mit **MpBackupCore** interagiert werden. Beim Hochstarten der Maschine wird **MpBackupCore** aktiviert. Dabei wird automatisch nach neuen Updates gesucht.

Ist ein neues Update vorhanden, so wird dies in der Info-Struktur an **MpBackupCore** angezeigt.



Der Update-Name welcher in der Info-Struktur angegeben wird, muss am Eingangsparameter "Name" an MpBackupCore verwendet werden. Durch "Install = `TRUE`" wird das Software-Update installiert.

Um die Installation über eine mapp View Visualisierung zu starten, wird ein Button-Widget eingefügt. Bei einem Klick auf den Button soll der Eingangsparameter "Install" des MpBackupCore Funktionsbausteins auf `TRUE` gesetzt werden.



Die Aktion, die bei einem Klick auf den Button ausgelöst wird, kann dabei folgendermaßen aussehen:

```
<Bindings>
    <EventBinding id="ContentTest.InstallUpdate.Click">
        <Source contentRefId="ContentTest" widgetRefId="InstallUpdate" xsi:type="widgets.brease.Button.Event" even
            <EventHandler>
                <Action>
                    <Target xsi:type="opcUa.NodeAction" refId="::BackupMgmt:MpBackupCore_0.Install" >
                        <Method xsi:type="opcUa.NodeAction.SetValueBool " value="true" />
                    </Target>
                </Action>
            </EventHandler>
    </EventBinding>
</Bindings>
```

Sobald also ein neues Update vorhanden ist, kann der Maschinenbediener/ Service-Techniker das Update an der Machine durchführen.

3.5 Diagnose

Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

3.5.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp Backup auftreten können.

Mein Backup kann nicht erstellt/installiert werden

Es gilt zu beachten, dass Backups in Simulation nicht erstellt oder installiert werden können. Läuft auf dem Zielsystem ARWin so kann ebenfalls kein Backup erstellt oder installiert werden.

Es darf keine Installation eines Backups durchgeführt werden, wenn sich dieses am internen Speicher (z.B. Compact Flash oder eingebauter Flash-Speicher) befindet.

3.6 Bibliotheken

3.6.1 Datentypen und Enumeratoren

3.6.1.1 Datentypen

3.6.1.1.1 MpBackupAutoBackupInfoType

Dieser Datentyp liefert Information über das automatische Erstellen von Backups.

Parameter	Datentyp	Beschreibung
InProgress	`BOOL`	Gibt an, ob ein automatisches Backup erstellt wird oder nicht.

3.6.1.1.2 MpBackupAutoBackupType

Dieser Datentyp legt fest, wann und wie das Backup abgespeichert werden soll.

Parameter	Datentyp	Beschreibung	Konfigurationsupdate
Enabled	`BOOL`	Gibt an ob automatisch ein Backup erstellt werden soll ("Enabled = `TRUE`") oder nicht ("Enabled = `FALSE`")	
NamePrefix	`STRING[50]`	Definiert den Namen des Backups. Es wird automatisch nach dem Namen ein Zeitstempel angehängt. Das kann zum Beispiel folgendermaßen aussehen: "myPrefix_%Y_%m_%d_%H_%M_%S"	
DeviceName	`STRING[255]`	Angabe des Speichermediums, wo das Backup abgelegt werden soll.	
Mode	`MpBackupIntervalType`	Definiert in welchen Zeitintervall das Backup abgespeichert werden soll.	
OverwriteOldest	`MpBackupOverwriteOldestType`	Definiert ob ältere Backups gelöscht werden sollen oder nicht.	

3.6.1.1.3 MpBackupAutoInfoType

Dieser Datentyp liefert Information über Backups und Updates.

Parameter	Datentyp	Beschreibung
Backup	`MpBackupAutoBackupInfoType`	Gibt an, ob ein automatisches Backup erstellt wird oder nicht.
Update	`MpBackupAutoUpdateInfoType`	Gibt an, ob ein neues Update verfügbar ist oder nicht.

3.6.1.1.4 MpBackupAutoUpdateInfoType

Dieser Datentyp liefert Information über Updates.

Parameter	Datentyp	Beschreibung
IsAvailable	`BOOL`	Definiert, ob ein neues Update verfügbar ist.
Name	`STRING[255]`	Gibt den Namen des Updates an.
ConfigurationID	`STRING[255]`	Konfigurations ID des Updates.
ConfigurationVersion	`STRING[64]`	Konfigurations-Version des Updates.

3.6.1.1.5 MpBackupAutoUpdateType

Dieser Datentyp liefert Information über die automatische Suche nach Updates.

Parameter	Datentyp	Beschreibung	Konfigurationsupdate
Mode	MpBackupAutoUpdateModeEnum	Gibt an, wie nach Updates gesucht werden soll.	
DeviceName	STRING[255]	Speichermedium	
Check	MpBackupUpdateCheckType	Zeitangabe für die Suche nach Updates	

3.6.1.1.6 MpBackupCoreConfigType

Dieser Datentyp legt fest, ob das Backup automatisch abgespeichert werden soll.

Wie das Konfigurationsupdate der einzelnen Parameter zur Laufzeit durchgeführt wird, ist im Abschnitt Konfiguration während der Laufzeit erklärt.

Parameter	Datentyp	Beschreibung
AutomaticBackup	MpBackupAutoBackupType	Über die Struktur kann definiert werden, ob ein Backup automatisch erstellt werden soll oder nicht.
AutomaticUpdate	MpBackupAutoUpdateType	Über die Stuktur kann definiert werden, ob automatisch nach Updates gesucht werden soll oder nicht.

3.6.1.1.7 MpBackupCoreInfoType

Mit diesem Datentyp wird für den Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
LastBackup	MpBackupLastBackupType	Information über das zuletzt erstellte Backup.
Project	MpBackupProjectInfoType	Information über das zu installierende bzw. angefragte Backup.
Diag	MpBackupDiagType	Eine Diagnosestruktur für die StatusID.
Automatic	MpBackupAutoInfoType	Gibt Information über automatisch erstellte Backups und über Updates.

3.6.1.1.8 MpBackupDiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	MpBackupStatusIDType	StatusID Diagnose Struktur.

3.6.1.1.9 MpBackupInfoType

Mit diesem Datentyp wird für den Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpBackupDiagType	Eine Diagnosestruktur für die StatusID.

3.6.1.1.10 MpBackupIntervalType

Dieser Datentyp beschreibt in welchem Intervall das Backup automatisch erstellt werden soll.

Parameter	Datentyp	Beschreibung	Konfigurationsupdate
Interval	MpBackupIntervalEnum	Der Parameter "Interval" legt fest, in welchen Intervall die Datei abgespeichert werden soll. Die Datei kann wöchentlich oder täglich abgespeichert werden.	
Day	MpBackupDayEnum	Legt fest an welchen Tag das Backup erstellt werden soll.	
Time	TIME_OF_DAY	Legt fest um welche Uhrzeit das Backup erstellt werden soll. Die Zeit wird in Sekunden beginnend von 00:00 angegeben. Soll die Datei um 06:00 morgens abgespeichert werden, so muss eine Zeit von 21600 s angegeben werden.	

3.6.1.1.11 MpBackupLastBackupType

Dieser Datentyp enthält Informationen über das zuletzt erstellte Backup.

Parameter	Datentyp	Beschreibung
Name	STRING[255]	Name des zuletzt erstellten Backups.
DateTime	DATE_AND_TIME	Zeitpunkt in dem das Backup erstellt wurde.
GenerationType	MpBackupGenerationTypeEnum	Gibt an ob das letzte Backup manuell oder automatisch erstellt worden ist.

3.6.1.1.12 MpBackupOverwriteOldestType

Dieser Datentyp gibt an, ob das älteste Backup überschrieben werden soll oder nicht.

> **Wird diese Option aktiviert und es befinden sich am angegebenen Speichermedium 30 Backups, "MaximumNumberOfBackups" aber nur 10 Backups erlaubt, so werden die ältesten 20 Backups gelöscht.**

Parameter	Datentyp	Beschreibung	Konfigurationsupdate
Enabled	BOOL	Legt fest, ob ältere Backups überschrieben werden soll oder nicht.	
MaximumNumberOfBackups	UINT	Definiert die Anzahl an möglichen Backups. Statt diese zu überschreiten wird das älteste vorhandene Backup überschrieben.	

3.6.1.1.13 MpBackupProjectInfoCurrentType

Dieser Datentyp gibt Auskunft über das aktive Backup am Zielsystem. Für mehr Information zu den beiden Parametern siehe hier.

Parameter	Datentyp	Beschreibung
ConfigurationID	STRING[255]	Konfigurations-ID des installierten Projekts.
ConfigurationVersion	STRING[64]	Konfigurations-Version des installierten Projekts.

3.6.1.1.14 MpBackupProjectInfoRequestType

Dieser Datentyp gibt Auskunft über das zuletzt installierte Backup am Zielsystem.

Parameter	Datentyp	Beschreibung
Name	STRING[255]	Name des Backups von welchem Information angefordert wurde.
ConfigurationID	STRING[255]	Konfigurations-ID des Projekts.
ConfigurationVersion	STRING[64]	Konfigurations-Version des Projekts.

3.6.1.1.15 MpBackupProjectInfoType

Dieser Datentyp gibt Auskunft über Backups.

Parameter	Datentyp	Beschreibung
Current	MpBackupProjectInfoCurrentType	Projektinformation des aktuellen Projekts.
Requested	MpBackupProjectInfoRequestType	Projektinformation des angefragen Backups.

3.6.1.1.16 MpBackupStatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	MpBackupErrorEnum	Fehlerkode des Funktionsbausteins.
Severity	MpComSeveritiesEnum	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler).
Code	UINT	Code der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden.

3.6.1.1.17 MpBackupUpdateCheckType

Dieser Datentyp gibt an wann nach einem Update gesucht werden soll.

Parameter	Datentyp	Beschreibung	Konfigurationsupdate
Type	MpBackupUpdateCheckEnum	Gibt an, wie nach Updates gesucht werden soll.	
Day	MpBackupDayEnum	Definiert den Tag nach dem ein Update gesucht werden soll.	
Time	TIME_OF_DAY	Definiert die Zeit nach dem nach einem Update gesucht werden soll. Die Zeit wird in Sekunden beginnend von 00:00 Uhr bestimmt. Das heißt soll nach einem Update um 06:00 Uhr morgens gesucht werden, so wird 21600 Sekunden angegeben.	

3.6.1.2 Enumeratoren

3.6.1.2.1 MpBackupAutoUpdateModeEnum

Dieser Aufzählungsdatentyp gibt an ob automatisch nach Updates gesucht werden soll oder nicht.

Aufzählung	Beschreibung
mpBACKUP_AUTO_UPDATE_DISABLED	Das automatische Suchen nach Updates ist deaktiviert.
mpBACKUP_AUTO_UPDATE_NOTIFY	Das automatische Suchen nach Updates ist aktiviert.

3.6.1.2.2 MpBackupDayEnum

Dieser Aufzählungsdatentyp gibt an, an welchem Tag ein automatisches Backup erstellt werden soll.

Aufzählung	Beschreibung
mpBACKUP_DAY_MONDAY	Datei wird an einem Montag erstellt.
mpBACKUP_DAY_TUESDAY	Datei wird an einem Dienstag erstellt.
mpBACKUP_DAY_WEDNESDAY	Datei wird an einem Mittwoch erstellt.
mpBACKUP_DAY_THURSDAY	Datei wird an einem Donnerstag erstellt.
mpBACKUP_DAY_FRIDAY	Datei wird an einem Freitag erstellt.
mpBACKUP_DAY_SATURDAY	Datei wird an einem Samstag erstellt.
mpBACKUP_DAY_SUNDAY	Datei wird an einem Sonntag erstellt.

3.6.1.2.3 MpBackupGenerationTypeEnum

Dieser Aufzählungsdatentyp gibt an wie das Backup erstellt wurde.

Aufzählung	Beschreibung
mpBACKUP_GENERATION_AUTO	Backup wurde automatisch erstellt.
mpBACKUP_GENERATION_MANUAL	Backup wurde manuell erstellt.

3.6.1.2.4 MpBackupIntervalEnum

Dieser Aufzählungsdatentyp gibt an in welchem Intervall ein Backup erstellt werden soll.

Aufzählung	Beschreibung
mpBACKUP_INTERVAL_DAILY	Backup wird täglich erstellt.
mpBACKUP_INTERVAL_WEEKLY	Backup wird wöchentlich erstellt.

3.6.1.2.5 MpBackupUpdateCheckEnum

Dieser Aufzählungsdatentyp gibt an in welchem Intervall ein Backup erstellt werden soll.

Aufzählung	Beschreibung
mpBACKUP_UPDATE_CHECK_DAILY	Es wird täglich nach Updates gesucht.
mpBACKUP_UPDATE_CHECK_WEEKLY	Es wird wöchentlich nach Updates gesucht.
mpBACKUP_UPDATE_CHECK_ON_ENABLE	Es wird bei jeder positiven Flanke an "Enable" vom Funktionsbaustein MpBackupCore nach Updates gesucht.

3.6.2 Funktionsbausteine

Unterstützt ab	Name	Beschreibung
5.3.0	MpBackupCore	Mithilfe dieses Funktionsbausteins kann ein Backup bzw. Update auf ein Speichermedium abgelegt oder installiert werden.
5.3.0	MpBackupCoreConfig	Mithilfe dieses Funktionsbausteins kann die automatische Erstellung einer Sicherung, des sogenannten Backups, konfiguriert werden.

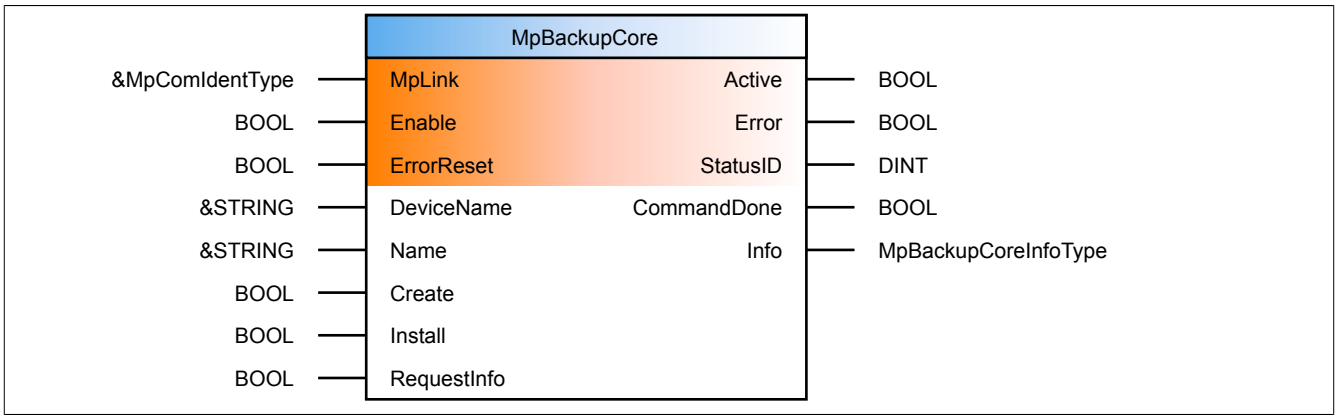
3.6.2.1 MpBackupCore

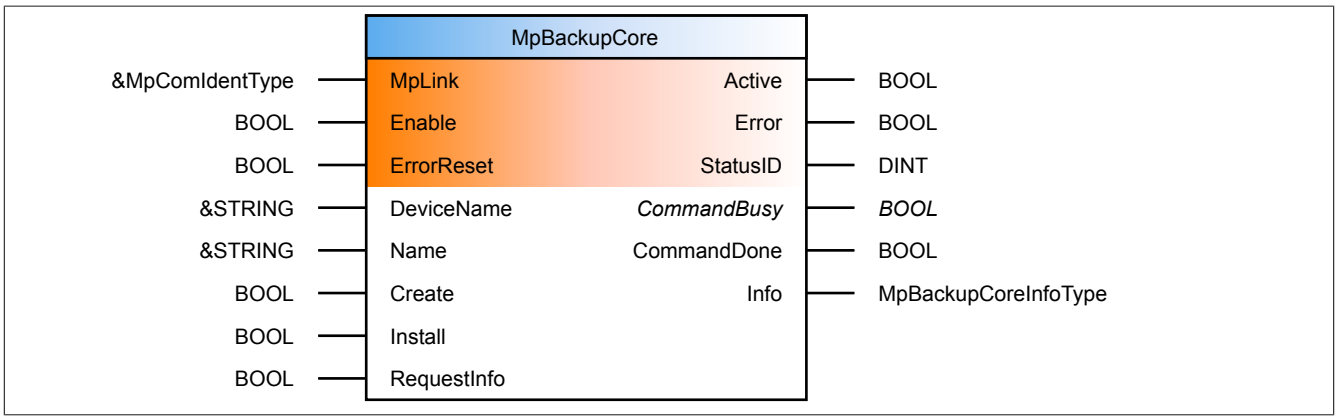
Mithilfe dieses Funktionsbausteins kann ein Backup bzw. Update auf ein Speichermedium abgelegt oder installiert werden.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Backup (GER/ENG) zu finden. (Zugangsdaten)

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpBackup Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	DeviceName	`Zeiger auf STRING[50]`	File Device (Speichermedium), auf dem die Dateien abgelegt werden.
IN	Name	`Zeiger auf STRING[255]`	Name des Backups/ Update. Die Datei kann auch in einem Ordner auf dem angegebenen Speichermedium abgelegt/gefunden werden. Dabei muss der Ordnername mitangegeben werden, wie zum Beispiel "Ordnername/FileName"
IN	Create	`BOOL`	Befehl um Backup zu erstellen.
IN	Install	`BOOL`	Befehl um Backup zu installieren.
IN	RequestInfo	`BOOL`	Über diesen Befehl können Informationen des Backups angefordert werden. Die Information ist am Ausgangsparameter *Info* sichtbar.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpBackupCoreInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

3.6.2.1.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpBackupCore Konfiguration verwendet.

Mithilfe dieses Funktionsbausteins kann ein Backup erstellt oder installiert werden. Wie ein Backup aufgebaut ist, ist im Abschnitt Aufbau eines Backups erklärt. Wie ein Backup verwendet werden kann, ist im Abschnitt Verwenden eines Backups erklärt.

Durch MpBackupCore kann aber auch ein Update installiert werden. Für mehr Information siehe Abschnitt Installieren eines Updates.

Bei "CommandDone = `TRUE`" wurde das Kommando fertig ausgeführt. Um ein neues Kommando zu starten, muss der vorherige Befehl auf `FALSE` gesetzt werden. Der Funktionsbaustein ist nun bereit einen neuen Befehl auszuführen.

Backup erstellen

Durch das Kommando "Create = `TRUE`" kann ein Backup erstellt werden. Die Datei wird auf dem an "DeviceName" angegebenen Speichermedium abgelegt. Der Name des Backup wird über den Eingangsparameter "Name" bestimmt. Die Datei kann auch in einem Ordner auf dem angegebenen Speichermedium abgelegt werden. Dabei muss der Ordnername mitangegeben werden, wie zum Beispiel "Ordnername/FileName". Wird ein Backup automatisch erstellt, so wird beim Namen des Backups automatisch ein Zeitstempel angehängt. Bei einem manuell erstellten Backup wird kein Zeitstempel angehängt. Für mehr Information siehe Verwenden eines Backups.

Backup/Update installieren

Mithilfe des Kommandos "Install = `TRUE`" kann ein vorhandenes Backup/Update auf dem Zielsystem installiert werden. In welchen Speichermedium das gewünschte Backup/Update abgelegt ist, wird über den Parameter "DeviceName" bestimmt. Welches Backup/Update geladen werden soll wird am Parameter "Name" angegeben. Das Backup/Update kann auch in einem Ordner auf dem angegebenen Speichermedium abgelegt werden. Dabei muss der Ordnername mitangegeben werden, wie zum Beispiel "Ordnername/FileName".

Information eines Backups anfordern

Mithilfe des Kommandos "RequestInfo = `TRUE`" kann die Konfigurations-ID und die Konfigurations-Version eines Backups angefordert werden. Für mehr Information zu den beiden Parametern siehe hier.
Dabei muss am Parameter "Name" der Name des gewünschten Backups angegeben werden, sowie der Speicherort des Backups ("DeviceName"). Die angeforderte Information wird in der Struktur MpBackupProjectInfoRequestType der Info-Stuktur angezeigt. Dadurch kann kontrolliert werden, ob die Konfigurations-ID des abgefragten Backups identisch ist mit der Konfiguration-ID des aktuell installierten Backups.

Info-Struktur

In der Info - Struktur können zusätzliche Informationen zu den erstellten Backups gefunden werden:

- **LastBackup:** Der Parameter "LastBackup" gibt an welches Backup zuletzt über das "Create"-Kommando erstellt wurde.
- **Project:** Der Parameter "Project" gibt an, welches Projekt auf der Steuerung installiert ist. Zusätzlich wird in der Info-Stuktur die Information angezeigt welche über "RequestInfo" angefragt wurde.

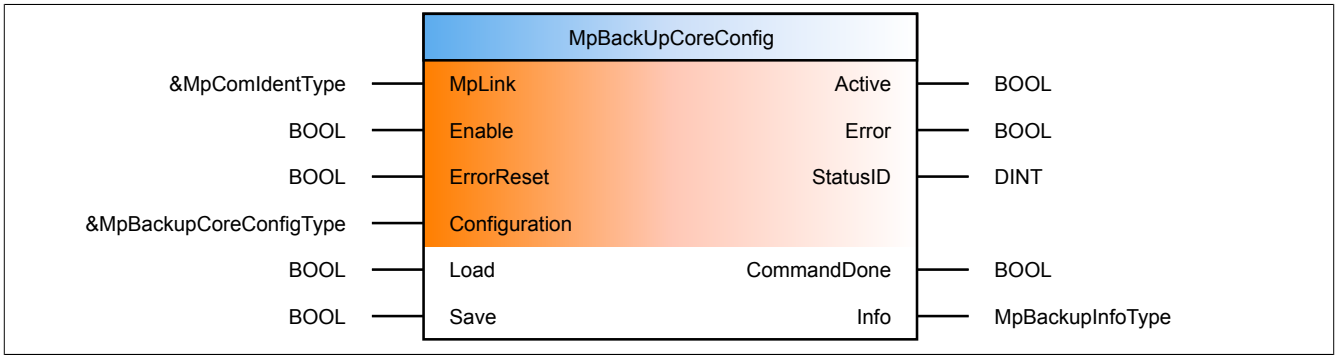
> **Die Daten welche durch MpBackupCore erstellt werden sind inhaltlich ident mit den Daten eines Projektinstallationspakets, welches durch Automation Studio oder dem Runtime Utility Center erstellt werden kann. Sie unterscheiden sich nur in der Größe, da das PIP in Automation Studio/ Runtime Utility Center bei der Erstellung anders komprimiert wird als über MpBackupCore.**
> **Für mehr Information siehe Abschnitt Aufbau des Projektinstallationspakets.**

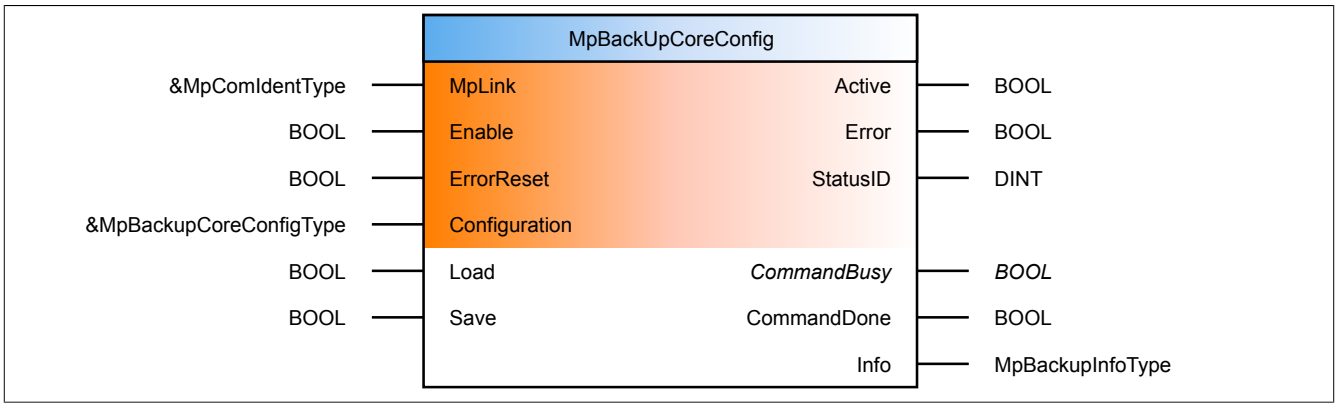
3.6.2.2 MpBackupCoreConfig

Mithilfe dieses Funktionsbausteins kann die automatische Erstellung einer Sicherung, des sogenannten Backups, konfiguriert werden.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpBackUpCore Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Configuration	`Zeiger auf MpBackupCoreConfigType`	Struktur, in der die Konfiguration festgelegt wird.
IN	Load	`BOOL`	Lädt die Konfiguration der Komponente.
IN	Save	`BOOL`	Speichert die Konfiguration der Komponente.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpBackupInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

3.6.2.2.1 Beschreibung

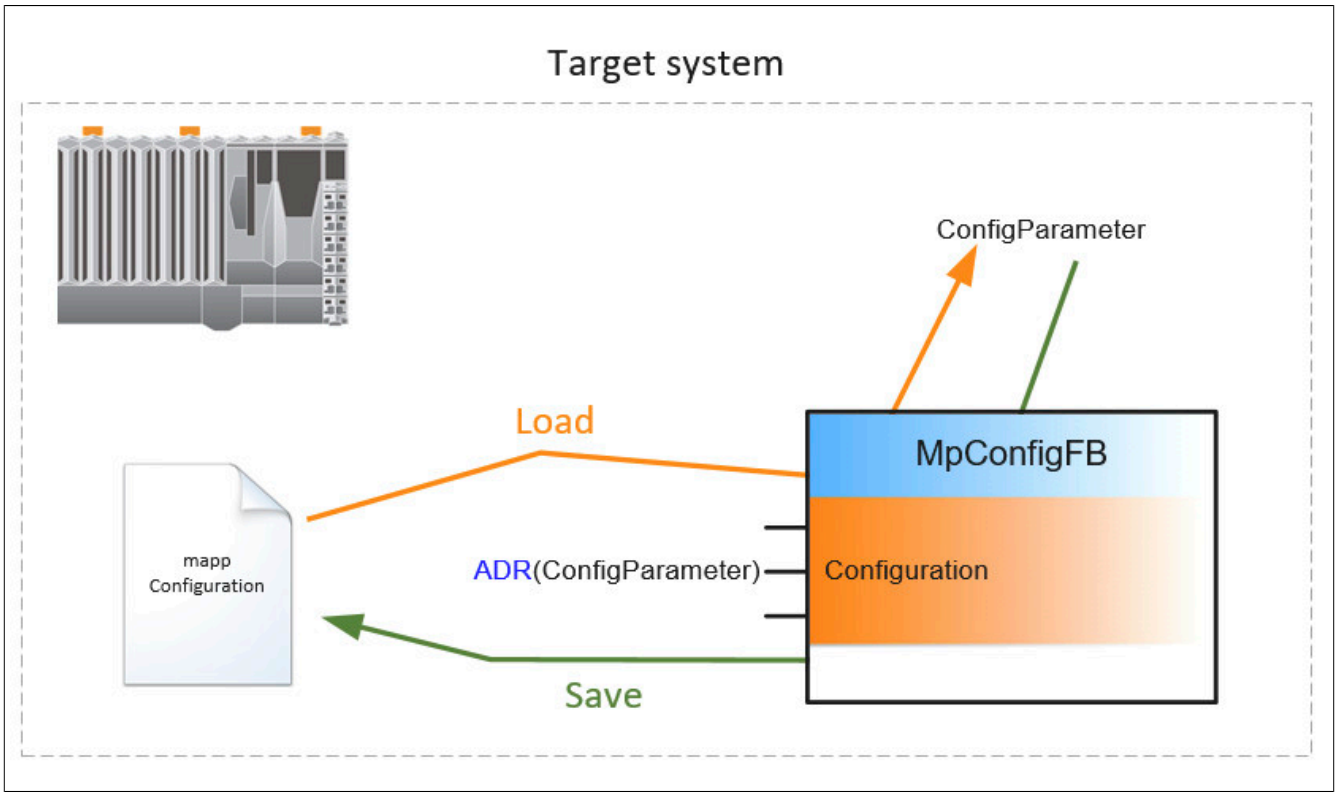
Für diesen Funktionsbaustein wird derselbe MpLink wie für MpBackupCore verwendet.

Mithilfe des Funktionsbausteins sind Änderungen der Konfiguration zur Laufzeit möglich.

Der Parameter "Configuration" ermöglicht es die Konfiguration, bzw. einen Teil der Konfiguration zu bearbeiten.

Der Befehl "Load" wird verwendet, um die aktuell konfigurierten Parameter auszulesen. Dabei wird die aktuell aktive Konfiguration der Steuerung in die PV am Eingang "Configuration" geschrieben. Die Daten in der Struktur werden dabei überschrieben. Danach können die Werte beliebig editiert werden.

Über den Befehl "Save" werden die Werte aus der "Configuration" Struktur auf die aktive Konfiguration übernommen und, wenn möglich, direkt verwendet.



Um die Konfigurations-Parameter in der Appliaktion korrekt anzupassen, wird folgendermaßen vorgegangen:

1. Befehl "Load = `TRUE`"

Die aktuelle Konfiguration wird vom Zielsystem in die Konfigurations-Struktur geladen. Der Vorgang ist mit "CommandDone = TRUE" abgeschlossen.

2. Befehl "Load = `FALSE`"

Rücksetzen des "Load" Befehls. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

3. Parametrieren der Konfigurationsstruktur

Die Parameter der Konfigurationsstruktur können nun wie gewünscht angepasst werden.

4. Befehl "Save = `TRUE`" setzen

Die geänderte Konfigurationsstruktur wird auf das Zielsystem übertragen. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

5. Befehl "Save = `FALSE`" setzen

Rücksetzen des "Save" Befehls. Der Vorgang ist mit "CommandDone = `FALSE`" abgeschlossen. Um zu kontrollieren ob die Parameter erfolgreich geändert worden sind, können die Schritte 1 und 2 erneut ausgeführt werden.

Konfigurieren der MpBackupCoreConfigType-Struktur

Über diese Konfiguration wird festgelegt ob das Backup automatisch erstellt wird oder nicht.

Wird das Backup automatisch erstellt können folgende Einstellungen getroffen werden:

- **NamePrefix:** Definiert den Namen des Backups. Es wird automatisch nach dem Namen ein Zeitstempel angehängt. Das kann zum Beispiel folgendermaßen aussehen: "myPrefix_%Y_%m_%d_%H_%M_%S"
- **DeviceName:** Angabe des Speichermediums, wo das Backup abgelegt werden soll.
- **Mode:** Definiert in welchen Zeitintervall das Backup abgespeichert werden soll. Das Backup kann wöchentlich oder täglich abgespeichert werden.
- **OverwriteOldest:** Definiert ob ältere Backups gelöscht werden sollen oder nicht.

3.6.3 Statusnummern

3.6.3.1 -1064159488: Installation fehlgeschlagen

Beschreibung:

Das angegebene Backup/Update {2:Name} konnte nicht installiert werden. Fehlernummer: {1:ErrorNumber}

Zusätzliche Information

- {2:Name}: Name des Backup/Update
- {1:ErrorNumber}: Fehlernummer

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen ob Backup am Eingangsparameter "Name" angegeben wurde
- Überprüfen ob das angegebene Speichermedium existiert
- Für weitere Information siehe Logger

Konstante:

`mpBACKUP_ERR_INSTALL_FAILED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore

3.6.3.2 -1064159487: Erstellung fehlgeschlagen

Beschreibung:

Das angegebene Backup/Update {2:Name} konnte nicht erstellt werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Information

- {2:Name}: Name des Backup/Update
- {1:ErrorNumber}: Fehlernummer

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen ob Backup am Eingangsparameter "Name" angegeben wurde
- Überprüfen ob das angegebene Speichermedium existiert
- Überprüfen ob Schreibrechte am angegebenen Speichermdium existieren
- Für weitere Inforamtion siehe Logger

Konstante:

`mpBACKUP_ERR_CREATE_FAILED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore

3.6.3.3 -1064159486: Information konnte nicht abgefragt werden

Beschreibung:

Die Information des gewünschten Backups/Update {2:Name} konnte nicht abgefragt werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Information

- {2:Name}: Name des Backup/Update
- {1:ErrorNumber}: Fehlernummer

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen ob Backup am Eingangsparameter "Name" angegeben wurde
- Überprüfen ob das angegebene Speichermedium existiert
- Für weitere Information siehe Logger

Konstante:

`mpBACKUP_ERR_REQUEST_INFO_FAILED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore

3.6.3.4 -2137980917: Warnung beim Laden der Konfiguration

Beschreibung:

Warnung beim Laden der Konfigurationsdatei (in die PV). Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Konfigurationsdatei beschädigt
- Unterschied zwischen dem Inhalt der Konfigurationsdatei und der PV (z.B. mehr Daten in der Konfigurationsdatei als in der PV)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCoreConfig

3.6.3.5 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore
- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_MPLINK_NULL`

3.6.3.6 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore

Konstante

`mpBACKUP_ERR_ACTIVATION`

3.6.3.7 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore
- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_MPLINK_INVALID`

3.6.3.8 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore
- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_MPLINK_CHANGED`

3.6.3.9 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore
- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_MPLINK_CORRUPT`

3.6.3.10 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore

Konstante

`mpBACKUP_ERR_MPLINK_IN_USE`

3.6.3.11 -1064239096: Konfigurations Struktur ist ein Nullzeiger

Beschreibung:

Der "Configuration" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "Configuration" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_CONFIG_NULL`

3.6.3.12 -1064239095: Konfigurations Zeiger ist keine PV

Beschreibung:

Der angegebene Zeiger ist kein Zeiger auf eine gültige PV. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "Configuration" Eingang prüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_CONFIG_NO_PV`

3.6.3.13 -1064239094: Fehler beim Laden der Konfiguration

Beschreibung:

Fehler beim Laden der Konfigurationsdatei (in die PV). Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Konfigurationsdatei beschädigt
- Unterschied zwischen dem Inhalt der Konfigurationsdatei und der PV (z.B. unterschiedliche Datentypen, Daten außerhalb der gültigen Limits, ...)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_CONFIG_LOAD`

3.6.3.14 -1064239092: Fehler beim Speichern der Konfiguration

Beschreibung:

Fehler beim Speichern in die Konfigurationsdatei (aus der PV). Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Konfigurationsdatei beschädigt
- Unterschied zwischen dem Inhalt der Konfigurationsdatei und der PV (z.B. unterschiedliche Datentypen, Daten außerhalb der gültigen Limits, ...)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_CONFIG_SAVE`

3.6.3.15 -1064239091: Ungültige Konfiguration

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Konfiguration ist beschädigt
- Die Funktion ist in der Konfiguration nicht aktiviert

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore
- MpBackupCoreConfig

Konstante

`mpBACKUP_ERR_CONFIG_INVALID`

3.6.3.16 -1064289140: Lesen des Backups/Update fehlgeschlagen

Beschreibung:

Das Backup/Update {3:Name} konnte nicht vom Speichermedium {2:DeviceName} installiert werden. Die Datei konnte nicht gelesen werden.

Zusätzliche Information

- {2:DeviceName}: Angegebenes Speichermedium
- {3:Name}: Name des Backups/ Updates

Ursache / Abhilfe:

- Überprüfen ob das angegebene Speichermedium existiert
- Daten des Backups überprüfen
- Für mehr Information siehe Logger

3.6.3.17 -1064289139: Öffnen/Lesen der Datei fehlgeschlagen

Beschreibung:

Das Öffnen und Lesen der Datei {3:FileName} vom vom Speichermedium {2:DeviceName} fehlgeschlagen.

Zusätzliche Information

- {3:FileName}: Name der Datei
- {2:DeviceName}: Angegebenes Speichermedium

Ursache / Abhilfe:

- Überprüfen ob das angegebene Speichermedium existiert
- Überprüfen des angegebenen Datei-Namen
- Überprüfen ob Datei extern manipuliert wurde

3.6.3.18 -1064289138: Inhalt Zielidentifizierungsdatei ungültig

Beschreibung:

Die Zielidentifizierungsdatei {2:FileName} ist ungültig

Zusätzliche Information

- {3:FileName}: Name der Datei

Ursache / Abhilfe:

- Überprüfen ob das angegebene Speichermedium existiert
- Überprüfen des angegebenen Datei-Namen
- Überprüfen des Datei-Inhalts
- Überprüfen ob Datei extern manipuliert wurde

3.6.3.19 -1064289137: Update/Backup kann nicht auf Zielsystem installiert werden

Beschreibung:

Das Update/Backup kann nicht installiert werden, da es nicht mit dem verwendeten Zielsystem übereinstimmt.

Ursache / Abhilfe:

- Verwenden eines anderen Backup/Updates
- Verwenden eines anderen Zielsystems

3.6.3.20 -1064289136: Projekt konnte nicht installiert werden

Beschreibung:

Projekt konnte nicht installiert werden. Fehlerursache: {2:FailureCause}

Zusätzliche Information

- {2:FailureCause}: Fehlernummer

Ursache / Abhilfe:

- Überprüfen ob gewünschtes Backup am Zielsystem installiert werden darf (Übereinstimmung der Hardware)
- Für weitere Details siehe Logger

3.6.3.21 -1064289135: Speichermedium für PIP nicht gefunden

Beschreibung:

Das angegebene Speichermedium für das Projektinstallationspaket (PIP) wurde nicht gefunden.

Ursache / Abhilfe:

- Überprüfen ob das angegebene Speichermedium existiert

3.6.3.22 -1064289134: PIP ungültig

Beschreibung:

Das Projektinstallationspaket (PIP) kann nicht installiert werden oder existiert nicht am Speichermedium

Ursache / Abhilfe:

- Inhalt des PIP kontrollieren
- Überprüfen ob PIP vorhanden ist

3.6.3.23 -1064289133: PIP kann nicht installiert werden

Beschreibung:

Das Projektinstallationspaket (PIP) kann nicht installiert werden.

Ursache / Abhilfe:

- Überprüfen ob das angegebene Speichermedium existiert
- Überprüfen des angegebenen Datei-Namen

3.6.3.24 -1064289132: PIP nicht verwendbar am Zielsystem

Beschreibung:

Das Projektinstallationspaket (PIP) kann nicht installiert werden, da es vom Zielsystem nicht unterstützt wird.

Ursache / Abhilfe:

- Runtime Version am Zielsystem muss >=4.33 sein

3.6.3.25 -1064289131: Erzeugen des Backups fehlgeschlagen

Beschreibung:

Das Backup {4:Name} konnte auf dem Speichermedium {3:DeviceName} nicht erstellt werden. Fehlerursache: {2:FailureCause}

Zusätzliche Information

- {2:FailureCause}: Fehlerursache
- {3:DeviceName}: Speichermedium
- {4:Name}: Name des Backups

Ursache / Abhilfe:

- Überprüfen ob auf dem angegebenen Speichermedium Schreibrechte existieren
- Für weitere Informationen siehe Logger

3.6.3.26 -1064289130: Erstellung der Zielidentifizierungsdatei fehlgeschlagen

Beschreibung:

Die Zielidentifizierungsdatei konnte auf dem Speichermedium {3:DeviceName}/{2:RealtivePath} nicht erstellt werden.

Zusätzliche Information

- {2:RealtivePath}: Pfad
- {3:DeviceName}: Speichermedium

Ursache / Abhilfe:

- Überprüfen ob angegebenes Speichermedium existiert
- Überprüfen ob auf dem angegebenen Speichermedium Schreibrechte existieren

3.6.3.27 -1064289129: Erstellung des Ordners fehlgeschlagen

Beschreibung:

Der Ordner {4:FolderName} am Speichermedium {3:DeviceName}/{2:RelativePath} konnte nicht erstellt werden.

Zusätzliche Information

- {2:FailureCause}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen ob am angegebenen Speichermedium Schreibrechte existieren
- Für weitere Informationen siehe Logger

3.6.3.28 -1064289128: PIP Pfad ungültig

Beschreibung:

Der Pfad für das Projektinstallationspaket (PIP) ist ungültig

3.6.3.29 -1064289127: PIP kann nicht erstellt werden

Beschreibung:

Projektinstallationspaket (PIP) konnte nicht erstellt werden.

Ursache / Abhilfe:

- Überprüfen des Speichermediums
- Für weitere Inforamtion siehe Logger

3.6.3.30 -1064289125: Lesen der Projektinformation fehlgeschlagen

Beschreibung:

Das Lesen der Projektinformation ist fehlgeschlagen. Fehlerursache: {2:FailureCause}

Zusätzliche Information

- {2:FailureCause}: Fehlerursache

Ursache / Abhilfe:

- Für weitere Information siehe Logger

3.6.3.31 -1064289124: Lesen der Projektinformation des Zielsystems fehlgeschlagen

Beschreibung:

Das Lesen der Projektinformation des Zielsystems ist fehlgeschlagen. Fehlerursache: {2:FailureCause}

Zusätzliche Information

- {2:FailureCause}: Fehlerursache

Ursache / Abhilfe:

- Für weitere Inforamtion siehe Logger

3.6.3.32 -1064289123: Speichermedium ungültig

Beschreibung:

Das Speichermedium {2:DeviceName} ist ungültig

Zusätzliche Information

- {2:DeviceName}: Speichermedium

Ursache / Abhilfe:

- Überprüfen ob angegebenes Speichermedium existiert

3.6.3.33 -1064289122: Verzeichnis ungültig

Beschreibung:

Das Verzeichnis {2:DirectoryName} ist ungültig.

Zusätzliche Information

- {2:DirectoryName}: Name des Verzeichnis

Ursache / Abhilfe:

- Überprüfen ob das Verzeichnis existiert

3.6.3.34 -1064289121: Verzeichnis existiert bereits

Beschreibung:

Das angegebene Verzeichnis {2:DirectoryName} existiert bereits.

Zusätzliche Information

- {2:DirectoryName}: Name des Verzeichnis

Ursache / Abhilfe:

- Verwenden eines anderen Verzeichnis-Namen

3.6.3.35 -1064289120: Pfad ungültig

Beschreibung:

Der Pfad {2:Path} ist ungültig

Zusätzliche Information

- {2:Path}: Angegebener Pfad

Ursache / Abhilfe:

- Überprüfen ob Pfad angegeben wurde
- Überprüfen ob Pfad existiert

3.6.3.36 -1064289119: FileIO FB wird nicht unterstützt

Beschreibung:

Der FileIO Funktionsbaustein wird am Speichermedium {2:DeviceName} nicht unterstützt.

Zusätzliche Information

- {2:DeviceName}: Speichermedium

Ursache / Abhilfe:

- Das angegebene Speichermedium kann nicht verwendet werden

3.6.3.37 -1064289118: AR Version zu klein

Beschreibung:

Das Projekt konnte nicht installiert werden. Fehlerursache: {2:FailureCause}

Zusätzliche Information

- {2:FailureCause}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen ob angegebenes Speichermedium existiert
- Überprüfen des angegebenen Datei-Namens

3.6.3.38 -1064289117: Pfadangabe falsch

Beschreibung:

Der angegebene Pfad {3:FolderName} existiert nicht auf dem Speichermedium {2:DeviceName}.

Zusätzliche Information

- {2:DeviceName}: Speichermedium
- {3:FolderName}: Ordner-Name

Ursache / Abhilfe:

- Angegebenen Pfad überprüfen

3.6.3.39 -1064289116: Keine Handles verfügbar

Beschreibung:

Es sind keine Handles mehr verfügbar. Es kann auf das Speichermedium/Ordner nicht zugeriffen werden.

Ursache / Abhilfe:

- Überprüfen ob Zugriff auf das angegebene Speichermedium/Ordner existiert
- Überprüfen ob Ordner bereits geöffnet wurde

3.6.3.40 -1064289115: Kein Zugriff auf Ordner

Beschreibung:

Es kann auf den angegebenen Ordner nicht zugegriffen werden

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

3.6.3.41 -1064289114: Ungültiges Handle

Beschreibung:

Ungültiges Handle. Der angegebene Ordner kann nicht geöffnet oder geschlossen werden.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

3.6.3.42 -1064289113: Datenlänge zu klein

Beschreibung:

Die Datenlänge wurde zu klein festgelegt. Der Ordner kann nicht geöffnet werden.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

3.6.3.43 -1064289112: Ungültiger Zeiger

Beschreibung:

Es wurde ein ungültiger Zeiger definiert. Ordner kann nicht gelesen werden.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

3.6.3.44 -1064289111: Öffnen fehlgeschlagen

Beschreibung:

Der Ordner {3:FolderName} auf dem Speichermedium {2:DeviceName} konnte nicht geöffnet werden.

Zusätzliche Information

- {2:DeviceName}: Verwendetes Speichermedium
- {3:FolderName}: Ordner-Name

Ursache / Abhilfe:

- Überprüfen des angegeben Ordners
- Überprüfen ob auf den Ordner zugegriffen werden darf

3.6.3.45 -1064289110: Lesen fehlgeschlagen

Beschreibung:

Der Ordner {3:FolderName} auf dem Speichermedium {2:DeviceName} konnte nicht gelesen werden.

Zusätzliche Information

- {2:DeviceName}: Verwendetes Speichermedium
- {3:FolderName}: Ordner-Name

Ursache / Abhilfe:

- Überprüfen des angegebenen Ordner

3.6.3.46 -1064289109: Schließen fehlgeschlagen

Beschreibung:

Der Ordner {3:FolderName} auf dem Speichermedium {2:DeviceName} konnte nicht geschlossen werden.

Zusätzliche Information

- {2:DeviceName}: Verwendetes Speichermedium
- {3:FolderName}: Ordner-Name

Ursache / Abhilfe:

- Überprüfen des angegebenen Ordner

3.6.3.47 -1064289108: Daten können nicht gelöscht werden

Beschreibung:

Pfad enthält Daten welche nicht gelöscht werden können

3.6.3.48 -1064289107: Automatische Suche fehlgeschlagen

Beschreibung:

Suche nach automatisch erstellten Backup an {3:DeviceName}/{2:RelativePath} ist fehlgeschlagen.

Zusätzliche Information

- {2:DeviceName}: Verwendetes Speichermedium
- {3:RelativePath}: Pfad

Ursache / Abhilfe:

- Überprüfen ob angegebenes Speichermedium existiert
- Überprüfen des angegebenen Pfad

3.6.3.49 -1064289106: Backup löschen fehlgeschlagen

Beschreibung:

Das Backup {4:Name} an {3:DeviceName}/{2:RelativePath} konnte nicht gelöscht werden.

Zusätzliche Information

- {2:RelativePath}: Pfad
- {3:DeviceName}: Speichermedium
- {4:Name}: Name des Backups

3.6.3.50 -1064289105: Ethernet Schnittstelle kann nicht ermittelt werden

Beschreibung:

Die Ethernet Schnittstelle konnte nicht ermittelt werden

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

3.6.3.51 -1064289104: MAC Adresse kann nicht ermittelt werden

Beschreibung:

Die MAC Adresse des Zielsystems konnte nicht ermittelt werden. Fehlerursache: {2:FailureCause}

Zusätzliche Information

- {2:FailureCause}: Fehlerursache

Ursache / Abhilfe:

- Für weitere Information siehe Logger

3.6.3.52 -1064289103: Node-Nummer kann nicht ermittelt werden

Beschreibung:

Die Node-Nummer konnte nicht ermittelt werden.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

3.6.3.53 -1064289102: Host-Name kann nicht ermittelt werden

Beschreibung:

Der Host-Name des Zielsystems konnte nicht ermittelt werden. Fehlerursache: {2:FailureCause}

Zusätzliche Information

- {2:FailureCause}: Fehlerursache

Ursache / Abhilfe:

- Für weitere Information siehe Logger

3.6.3.54 -1064289101: Suche nach Backups fehlgeschlagen

Beschreibung:

Suche nach Backups/Updates auf {3:DeviceName}/{2:RelativePath} ist fehlgeschlagen

Zusätzliche Information

- {2:DeviceName}: Verwendetes Speichermedium
- {3:RelativePath}: Pfad

Ursache / Abhilfe:

- Überprüfen des angegebenen Speichermediums
- Überprüfen des Pfads

3.6.3.55 -2137901309: Suche nach Update fehlgeschlagen

Beschreibung:

Die Suche nach einem Update ist fehlgeschlagen. Fehlerursache: {1:ErrorNumber}

Zusätzliche Information

- {1:ErrorNumber}: Fehlernummer

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Überprüfen ob das angegebene Speichermedium existiert
- Für weitere Information siehe Logger

Konstante:

`mpBACKUP_WRN_UPDATE_CHECK_FAILED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpBackupCore

3.6.4 Alarme

3.6.4.1 mpBACKUP_ALM_CREATE_FAILED: Erstellen des Backup fehlgeschlagen

Beschreibung:

Das Backup konnte nicht erstellt werden.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Überprüfen ob ein Name für das Backup vergeben wurde
- Überprüfen ob das angegebene Speichermdium existiert
- Überprüfen ob Schreibrechte auf den angegebenen Speichermedium vorhanden sind

Verhalten

Edge Alarm

3.6.4.2 mpBACKUP_ALM_INSTALL_FAILED: Installieren des Backups fehlgeschlagen

Beschreibung:

Das gewünschte Backup konnte nicht installiert werden.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Überprüfen ob ein Name für das Backup vergeben wurde
- Überprüfen ob das angegebene Speichermdium existiert

Verhalten

Edge Alarm

3.6.4.3 mpBACKUP_ALM_UPDATE_CHECK_FAILED: Überprüfung auf Updates fehlgeschlagen

Beschreibung:

Die automatische Überprüfung ob neue Updates vorhanden sind ist fehlgeschlagen.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Überprüfen des angegebenen Speichermediums

Verhalten

Edge Alarm

4 mapp IO: Hardware Konfiguration verwalten

mapp IO stellt die Möglichkeit, eine Hardware-Konfiguration zur Laufzeit zu verändern, zur Verfügung.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp IO (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal sind viele Tutorials zum Thema zu finden.

4.1 Konzept

Allgemein

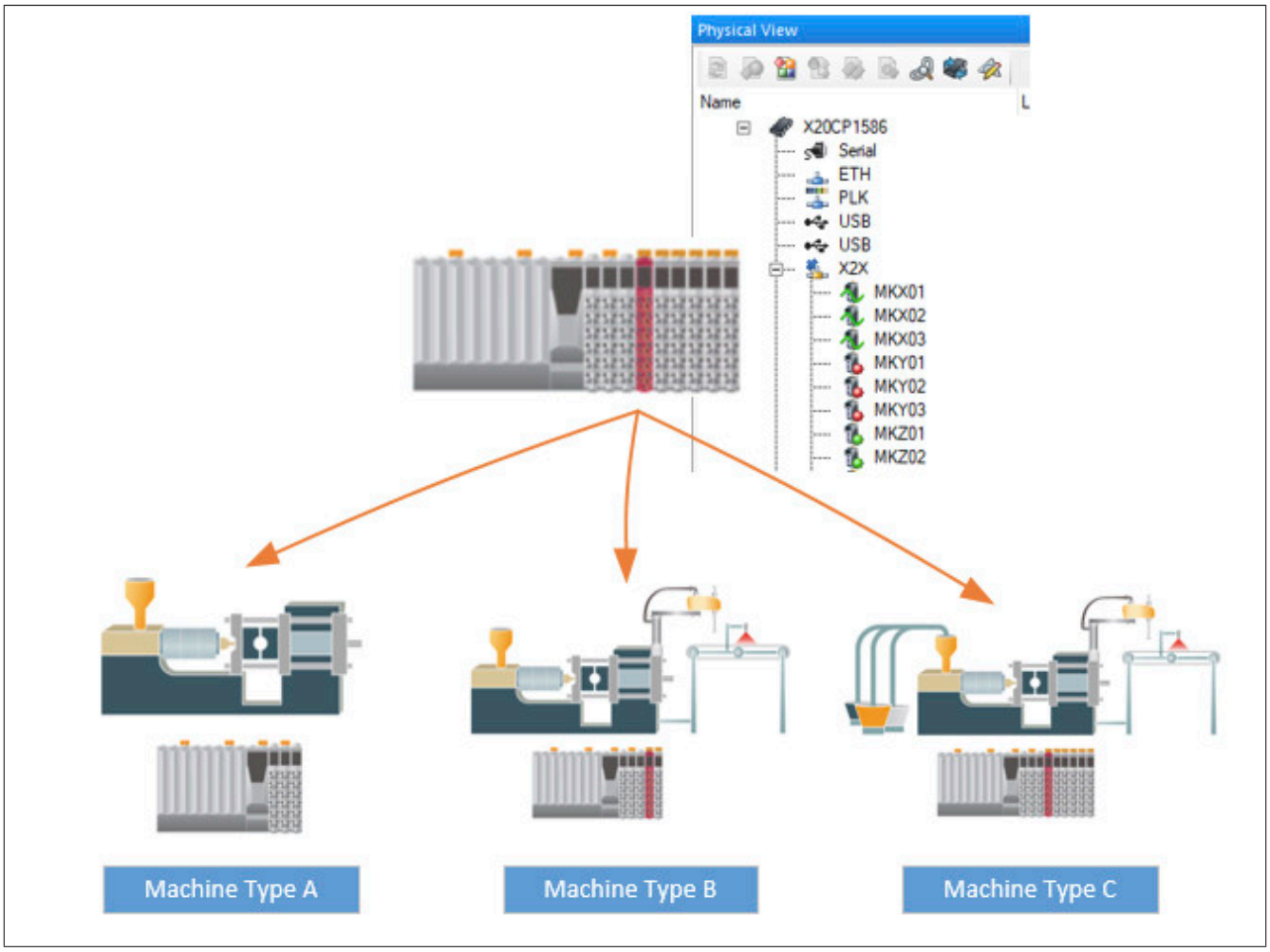
- Als Applikationist möchte ich nicht für jedes einzelne Modul prüfen müssen, ob ein Alarm gesetzt werden muss, da das Modul nicht richtig funktioniert
- Als Applikationist möchte ich nur eine Software für alle möglichen Maschinen-Konfigurationen warten müssen

Hardware Konfiguration über Visualisierung verwalten

- Als Maschinenbauer möchte ich die Flexibilität die Hardware-Konfiguration über ein externen Tool in einem gängigen Format zu erstellen und direkt an der Maschine zu Importieren
- Als Servicetechniker möchte ich den aktuellen Zustand aller Hardware-Module über die Visualisierung überprüfen können
- Als Servicetechniker möchte ich einen I/O Datenpunkt zur Laufzeit und ohne Neustart auf einen anderen I/O Datenpunkt umhängen können
- Als Servicetechniker möchte ich ein defektes Hardware Modul durch ein anderes (lagerndes) Modul, eines ähnlichen Typs, ersetzen können ohne dafür Automation Studio bedienen können zu müssen

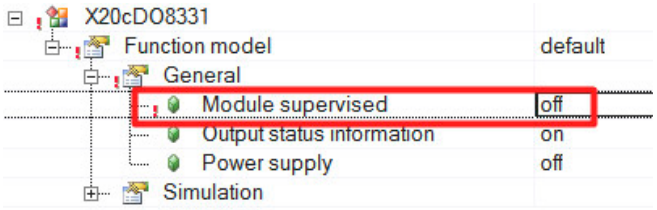
4.1.1 Grundprinzip

Mithilfe von mapp IO kann die Hardwarekonfiguration zur Laufzeit verwaltet werden. Dadurch können, je nach verwendeter Hardware, unterschiedliche Maschinentypen realisiert werden.



Damit unterschiedliche Maschinentypen verwendet werden können, muss als Basis ein Projekt erstellt werden, welche einmal alle benötigten Module für die verschiedenen Maschinentypen beinhaltet. Dabei werden nicht nur alle notwendigen Module einmalig eingefügt, sondern auch für alle benötigten Modulkonfigurationen muss ein Modul eingefügt werden. Beispielsweise brauchen alle Maschinentypen ein X20AT2402. Manche davon benötigen das Modul mit Sensortyp J und andere verwenden den Sonsortyp K. In diesem Fall muss das Modul zweimal eingefügt werden. Eines welches den Sensortypen J verwendet und eines welches den Sensortyp K verwendet.

Die Modulüberwachung ("Module supervised") sollte bei allen Modulen deaktiviert sein. Ansonsten startet die Steuerung im Service-Modus wenn die Hardware-Module nicht am angegebenen Steckplatz vorhanden sind.



Danach wird mapp IO implementiert. Dies wird Schritt für Schritt im Getting Started Hardwarekonfiguration zur Laufzeit verwalten erklärt.

Im Anschluss wird das Projekt auf die Steuerung übertragen. Dadurch ist die Automation Runtime Konfiguration (AR-Konfiguration), welche später als Referenz für Imports dient, auf der Steuerung vorhanden. Zustätzlich werden dadurch die Firmware-Version für alle notwendigen Module auf die Steuerung übertragen.

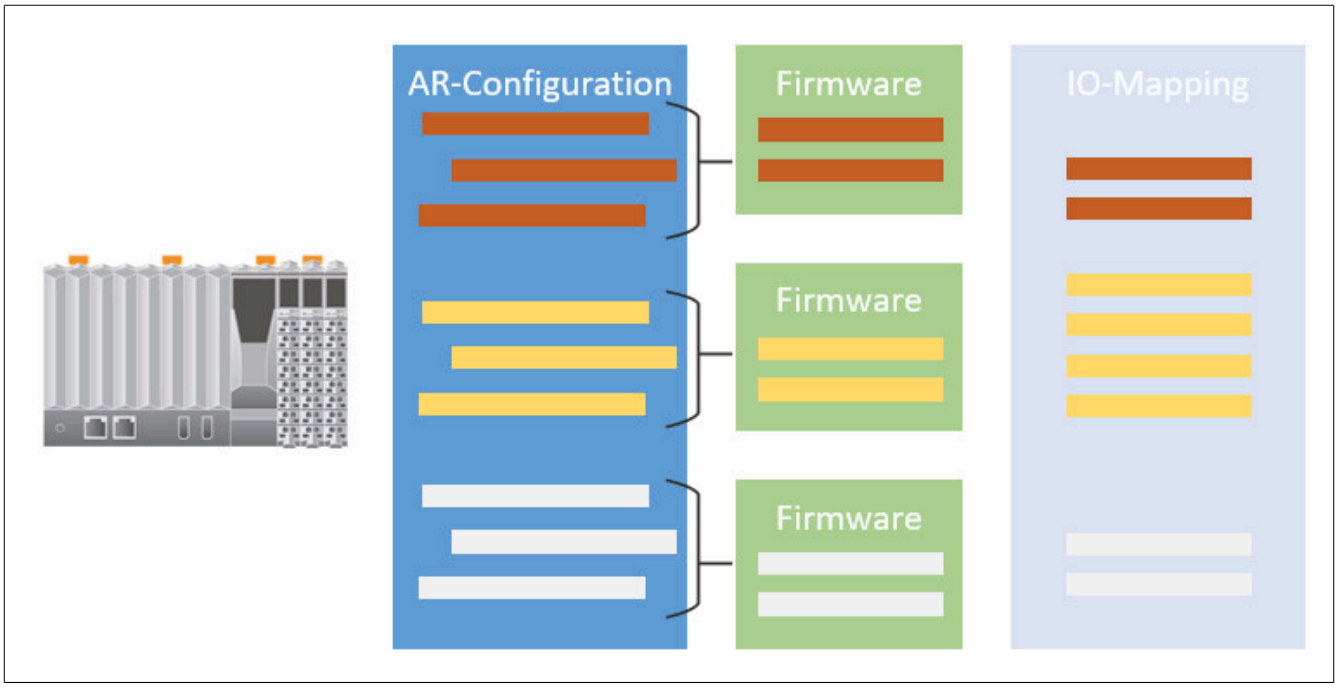
Welcher Maschinentyp geladen werden soll, wird mithilfe einer Import-Datei (XML-Format) definiert. In der Datei werden die Hardware-Module, sowie auch die IO-Zuordnung angegeben, welche für den Maschinentypen benötigt werden. Werden drei verschiedene Maschinentypen verwendet, müssen drei unterschiedliche Import-Dateien existieren. Über den Funktionsbaustein MpIOImport kann die Import-Datei geladen werden. Wie die Import-Datei aufgebaut ist, ist im Abschnitt Aufbau der Import-Datei erklärt.

Der Import kann auch mithilfe von Template-Dateien durchgeführt werden. Dazu wird pro Modul eine Template-Datei benötigt. Mehr Information dazu kann im Abschnitt Template Dateien gefunden werden.

Allgemeine Modul-Information

Um ein Hardware-Modul zu betreiben werden drei Dinge benötigt:

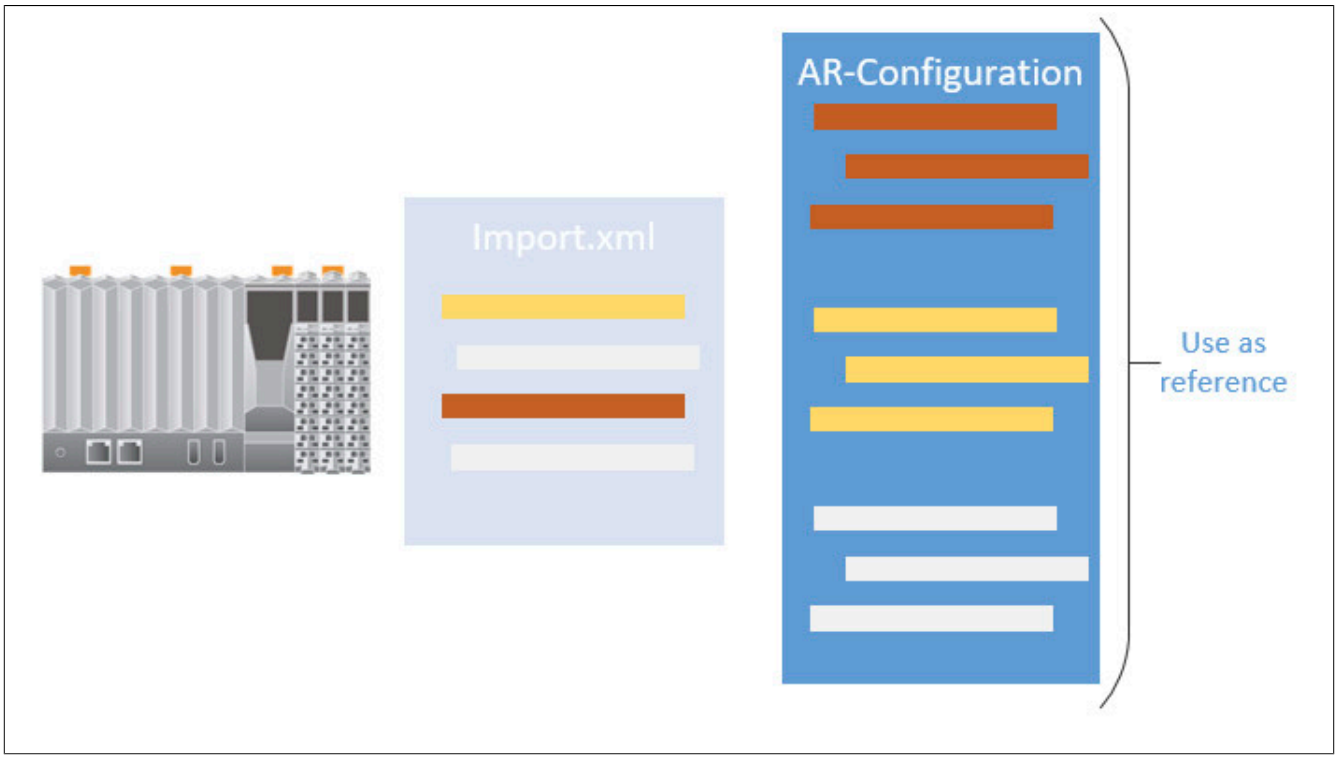
- **AR-Konfiguration:** Die AR-Konfiguration enthält die Funktionweise des Moduls, sowie die möglichen Parameter und Einstellungen, wie beispielsweise die Art des Temperatursensors, oder ob ein analoger Eingang auf Spannung oder Strom achtet. Die AR-Konfiguration eines Moduls kann nach dem Einfügen des Moduls im Hardwarebaum verändert werden.
- **Firmware:** Im Modul eingebettete Software, welche die Daten der AR-Konfiguration empfängt und das Modul den Einstellungen entsprechend betreibt. Die Firmware wird automatisch mit auf die Steuerung übertragen, sobald das Modul im Hardwarebaum eingefügt wurde.
- **I/O Zuordnung:** Über die I/O Zuordnung wird die Verbindung zwischen Software und Hardware definiert. Dabei kann zum Beispiel festgelegt werden an welcher Prozessvariable der analoge Eingang seinen aktuellen Wert überträgt. Die I/O Zuordnung eines Moduls kann nach dem Einfügen des Moduls im Hardwarebaum verändert werden.



Verändern der Hardware-Konfiguration über eine Import-Datei

In der Import-Datei wird definiert, welche Module an welcher Stelle verwendet werden sollen. Das heißt die AR-Konfiguration sowie die I/O Zuordnung kann beliebig über eine Import-Datei definiert werden. Wie eine Import-Datei aufgebaut ist, wird im Abschnitt Aufbau der Import-Datei erklärt.

Mithilfe der Import-Datei werden die gewünschten Module referenziert, welche in der ursprünglichen AR-Konfiguration vorhanden waren und generiert daraus eine neue AR-Konfiguration. Der genaue Import Prozess wird im Abschnitt Verwalten der Hardware-Konfiguration beschrieben.



Ist die neue AR-Konfiguration generiert, wird diese auch direkt aktiviert, was mit einem Neustart des Zielsystems verbunden ist. Wurde jedoch nur die I/O Zuordnung durch die Import-Datei geändert, wird kein Neustart durchgeführt. Das neue IO-Mapping wird direkt angewandt.

4.1.2 Rahmenbedingungen

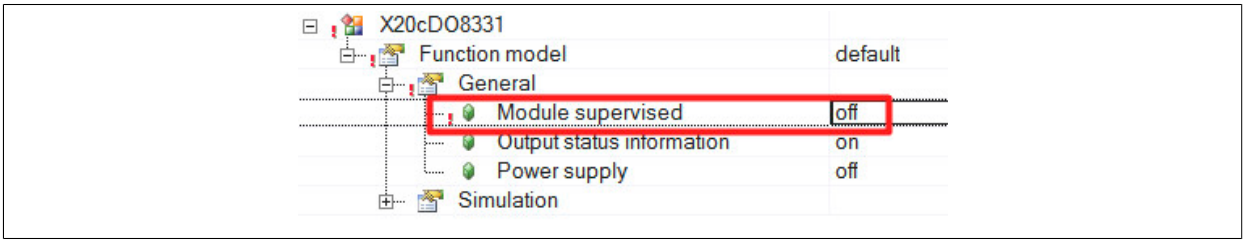
In diesem Abschnitt werden Rahmenbedingungen von mapp IO aufgelistet:

Antriebstechnik

- mapp IO kann nur in Kombination mit ACP10/ARNC0 verwendet werden! Aktuell wird mapp Motion nicht unterstützt.
- CNC-Kanäle können nicht konfiguriert werden.
- SDC Achsen können nicht konfiguriert werden.
- ACOPOS P3 kann im Projekt verwendet werden. Beim importieren der Hardware darf sich aber die Position des ACOPOS P3 nicht ändern.
- Bei der Verwendung von Template Dateien können Generische Interface Karten nicht verwendet werden.

Module

- Es müssen alle gewünschten Module in der Import-Datei angegeben werden, welche auch im Automation Studio sichtbar sind. Manche Module sind mit sogenannten versteckten Modulen verlinkt welche nicht in Automation Studio sichbar sind. Dazu gehört zum Beispiel das Modul X67BC8321.L12. Es enthält ein verstecktes Modul (X67BC8321.L12io). Da dieses aber auch nicht in Automation Studio angezeigt wird, muss es auch nicht in der Import-Datei angegeben werden!
- Die Modul-Überwachung ("Module supervised") sollte bei allen Modulen deaktiviert sein. Ansonsten startet die Steuerung im Service-Modus wenn die Hardware-Module nicht am angegebenen Steckplatz vorhanden sind.

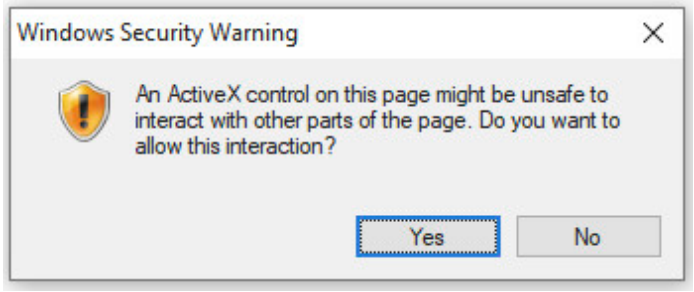


- Schnittstellenmodule können nicht in Kombination mit mapp IO verwendet werden.
- Variablen, welche auf Ein- und Ausgänge verbunden sind, dürfen keine direkt abgeleiteten Datentypen sein.

Nicht verwendbare Modul-Parameter

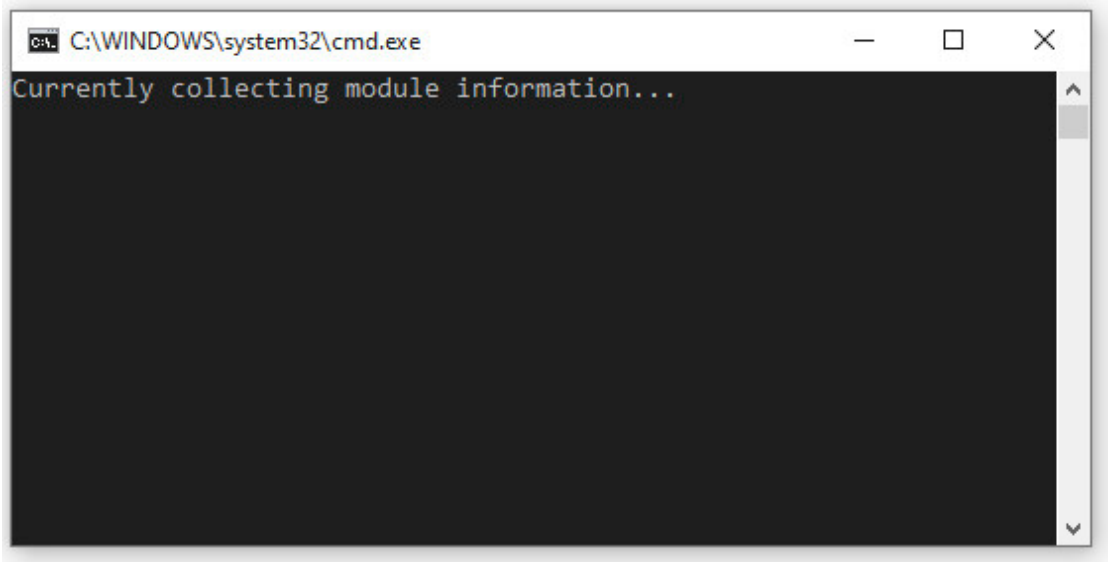
Manche Modul-Parameter können über die Import-Datei nicht geändert werden. Um zu überprüfen welche Parameter davon betroffen sind, kann eine Modulliste erstellt werden. Dabei werden die installierten Module geprüft und im Abschnitt Modulliste aufgelistet.

Durch klicken des Buttons wird eine ActiveX-Objekt gestartet. Der Start muss bestätigt werden. Bei einem Klick auf "No" wird die Erstellung der Modulliste nicht durchgeführt.



Die Erstellung der Modulliste kann mehrere Minuten dauern, da jedes installierte Modul geprüft wird.

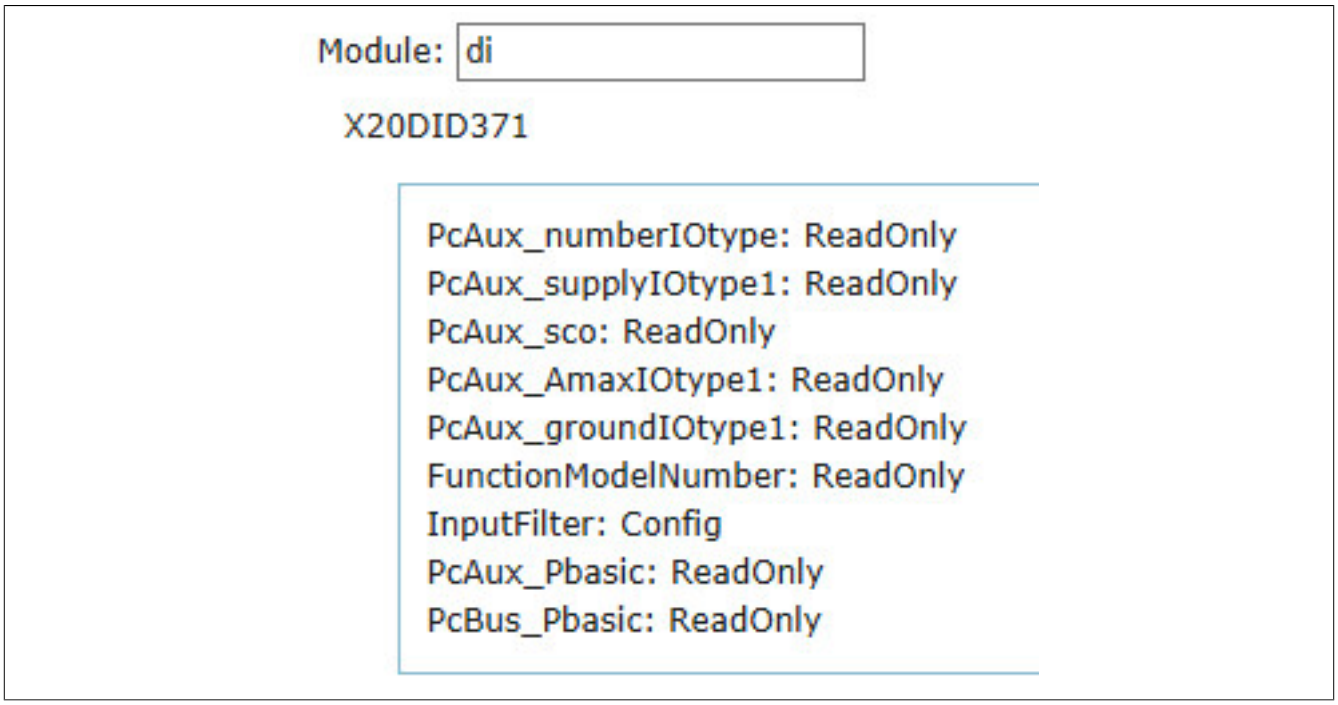
Währenddessen erscheint ein Kommandofenster. Ist die Erstellung abgeschlossen kann das Fenster geschlossen und die Modulliste geprüft werden.



Für die Erstellung wird Powershell benötigt, welches standarmäßig seit >= Windows 7 installiert ist. Wurde das Programm nachträglich gelöscht, kann die Modulliste nicht erstellt werden.

In der Modulliste werden alle Module-Parameter aufgelistet, welche nicht verwendet werden können. Über das Such-Fenster kann nach einem Modul gesucht werden. Wildcard-Symbole, wie beispielsweise "*" können für die Suche nicht verwendet werden. Wird auf ein Modul geklickt, werden die Parameter angezeigt welche nicht verwendet werden dürfen.

Zeigt ein Parameter "ReadOnly" an, so darf der Parameter nicht geändert werden. Änderungen können ein nicht definiertes Verhalten auslösen. Zeigt ein Parameter "Config" an, so hat eine Änderung keine Auswirkung. Die Einstellungen kann durch mapp IO nicht geändert werden.



4.1.2.1 Modulliste

ModuleListToHtml

xmlout

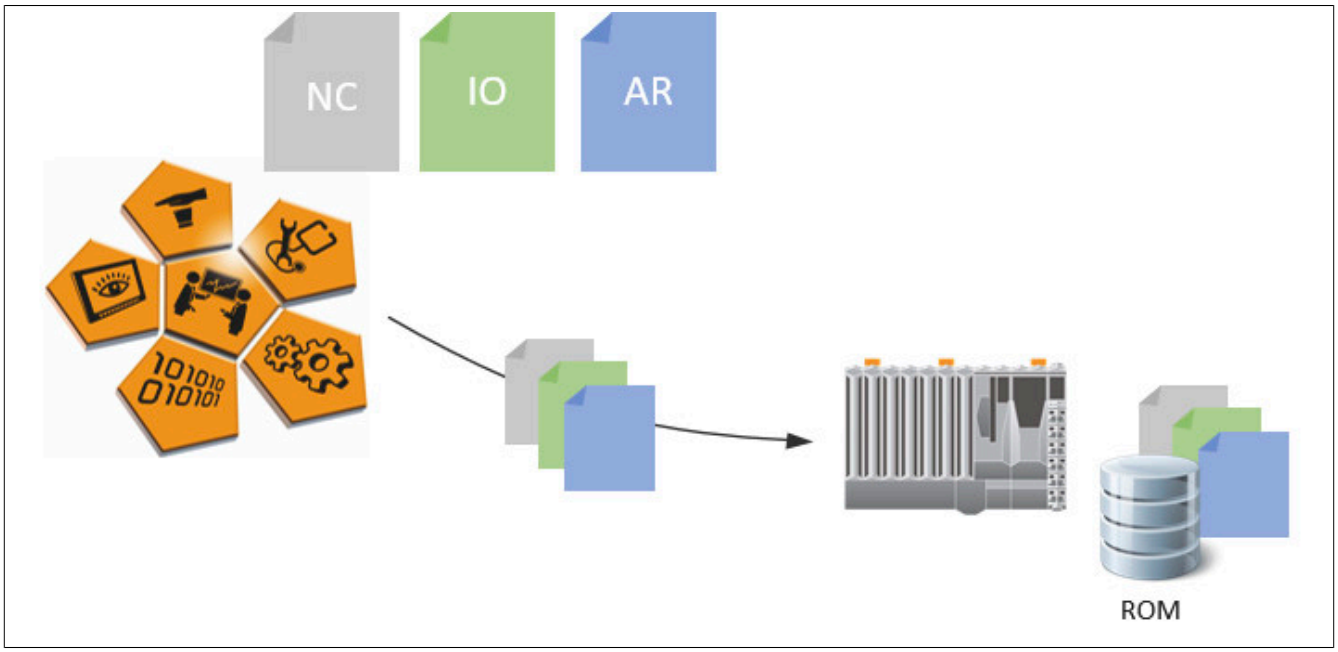
TestBatch

ReadInfo

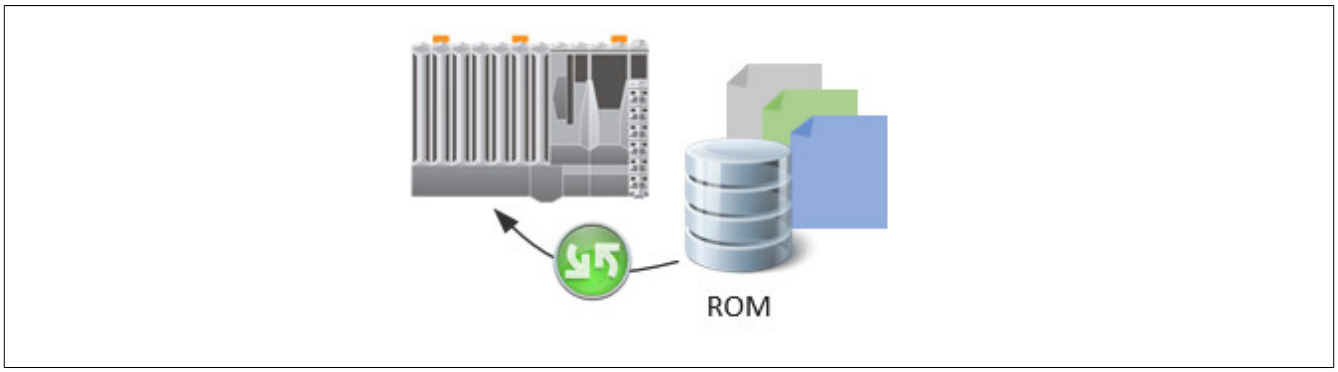
In diesem Abschnitt wird die Modulliste angezeigt. Werden noch keine Module angezeigt, so kann die Modulliste hier erstellt werden.

4.1.3 Verwalten der Hardware-Konfiguration

Wird ein Automation Studio Projekt auf die Steuerung übertragen, dann inkludiert das die AR-Konfiguration (AR), die I/O Zuordnung (IO) und auch die NC-Zuordnungstabelle (NC). Diese Informationen sind neben der Firmware notwendig, um die Hardware richtig zu konfigurieren und die Verbindung zur Software zu schaffen. Alle dafür notwendigen Daten werden auf dem ROM gespeichert. Für mehr Information zu den Speicher-Typen siehe hier.

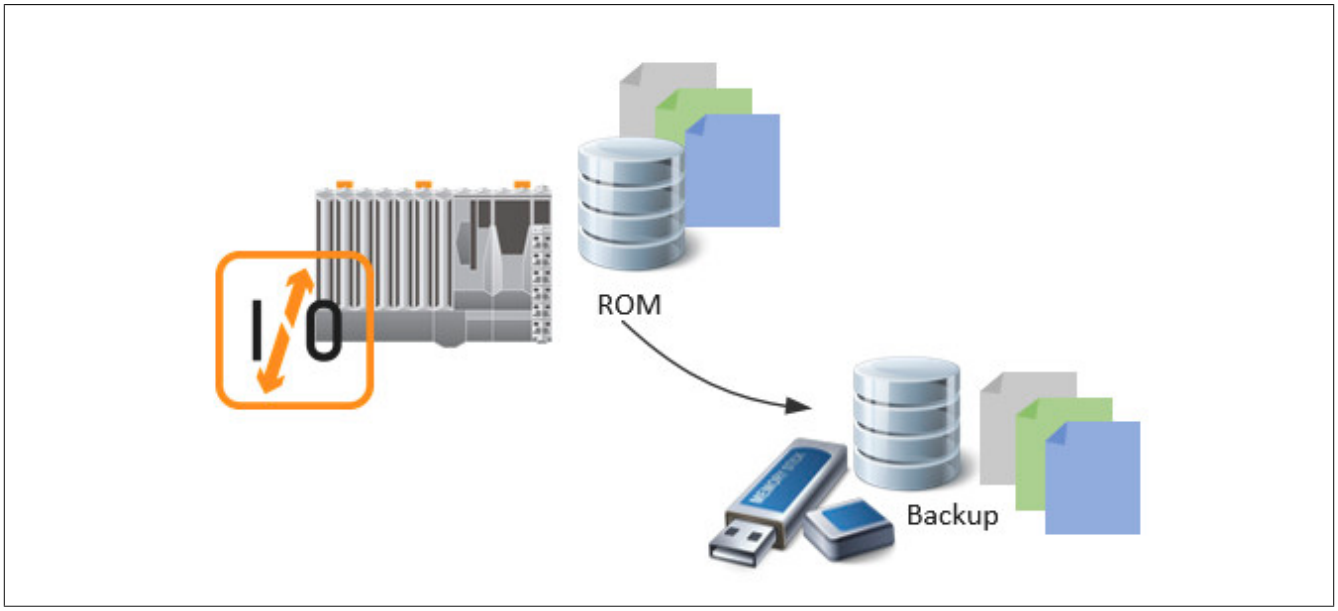


Die Dateien werden im ROM abgelegt und bei jedem Neustart von dort ins DRAM geladen.



Verwendung von MpIO

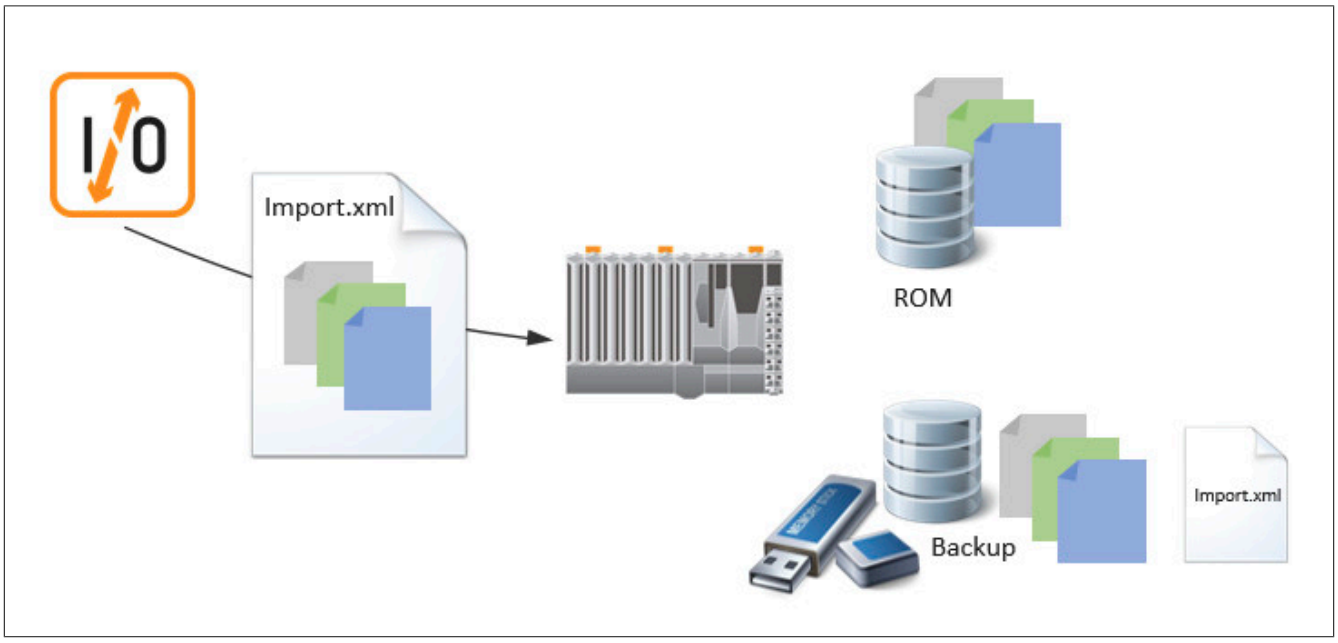
Soll die Konfiguration zur Laufzeit, ohne Einwirken von Automation Studio, geändert werden, kann dies über mapp IO erreicht werden. Wird der Funktionsbaustein MpIOImport aktiviert, so wird zuerst ein Backup der aktiven Konfigurations-Dateien erstellt.



Dabei werden die Konfigurations-Dateien (genauer die AR-Konfiguration, die I/O Zuordnung(en) und auch die NC-Zuordnungstabelle(n)) in ein Backup File Device kopiert, welches in der Konfiguration von mapp IO festgelegt wird.

Import-Befehl

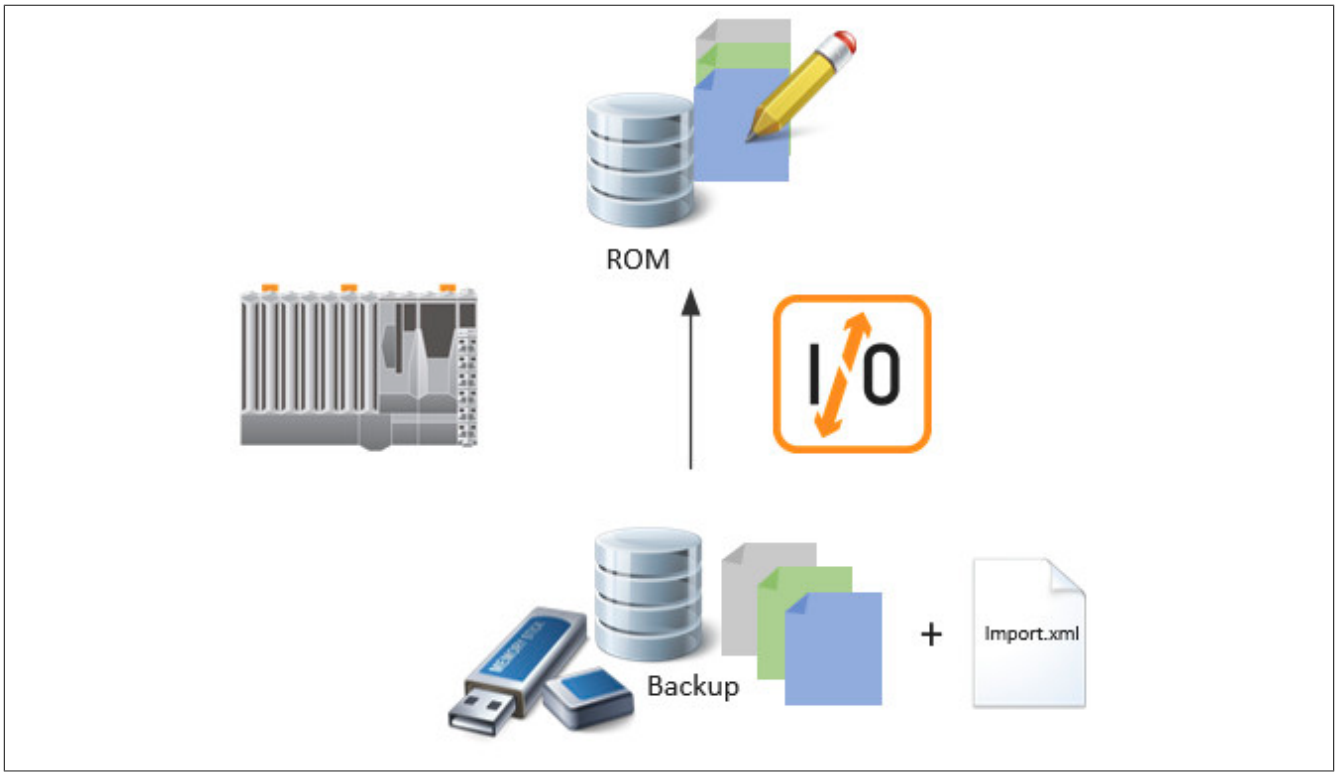
Bei einem "Import" Befehl kopiert MpIOImport die über den Eingang "FileName" definierte Import-Datei zusätzlich in das Backup File Device.



Zu diesem Zeitpunkt ist immer noch die anfangs über Automation Studio definierte Hardware-Konfiguration aktiv. Im nächsten Schritt wird aus den am Backup File Device vorhandenen Daten die neue Hardware-Konfiguration generiert. Abhängig von dem Parameter "Master Configuration" in der MpIO Konfiguration dient die Automation Studio Hardware-Konfiguration als Basis, oder die Import-Datei:

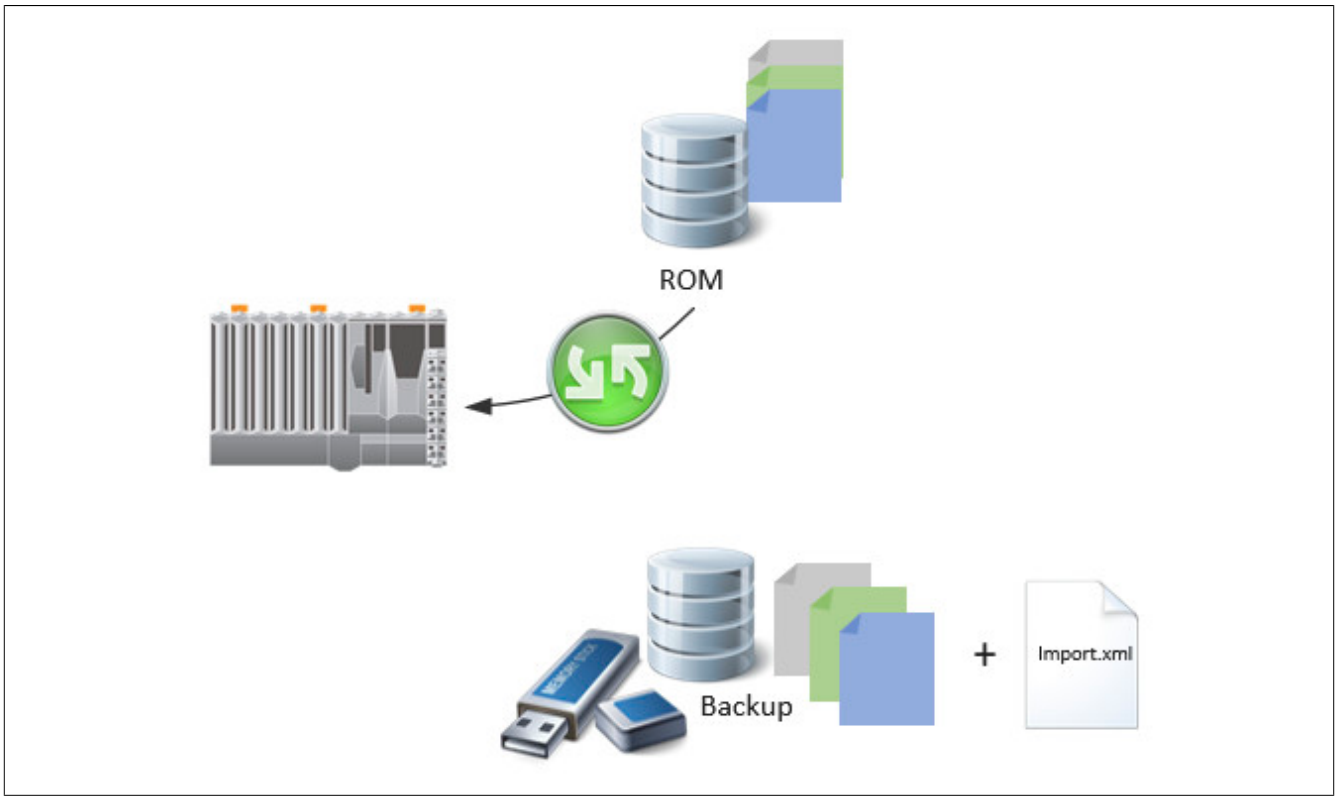
- **None:** Die Basis der neuen Hardware-Konfiguration ist ausschließlich die Import-Datei. Die von Automation Studio gelieferten Daten dienen nur noch als Referenz.
- **Automation Runtime:** Die Basis bleibt die von Automation Studio gelieferte Hardware-Konfiguration und die Import-Datei wird zum Erweitern einer bestimmte Schnittstelle (z.B. IF6, X2X bei einer X20CP1586) verwendet.

Wenn die neue Hardware-Konfiguration aus den am Backup File Device vorhandenen Daten generiert wurde, wird diese direkt übernommen.



Nachdem die neue Hardware-Konfiguration übernommen wurde startet das Zielsystem neu. Wurde jedoch nur die I/O Zuordnung geändert wird kein Neustart durchgeführt. Die neue Zuordnung wird direkt übernommen. Bei allen anderen Änderungen wird ein Neustart durchgeführt.

Es wird weiterhin bei jedem Start der Steuerung die Hardware-Konfiguration aus dem ROM geladen, nur das diese nicht mehr der Hardware-Konfiguration, welche über Automation Studio transferriert wurde entspricht, sondern der Hardware-Konfiguration welche über MpIOImport importiert wurde.



4.1.4 Wiederherstellen der Hardware-Konfiguration

mapp IO bietet einen Mechanismus der überwacht, ob die Hardware-Konfiguration extern (= nicht durch mapp IO) verändert wurde, beispielweise über einen Automation Studio Download oder über das Runtime Utility Center. Wird die Hardware-Konfiguration auf diese Weise verändert, während mapp IO aktiv ist, wird der Alarm "Hardware-Konfiguration wurde verändert" ausgelöst. Nun kann entschieden werden, ob die Hardware-Konfiguration wiederhergestellt werden soll oder nicht. Die Entscheidung wird über den Parameter "ImportAllowed" an MpIOImport gesteuert.

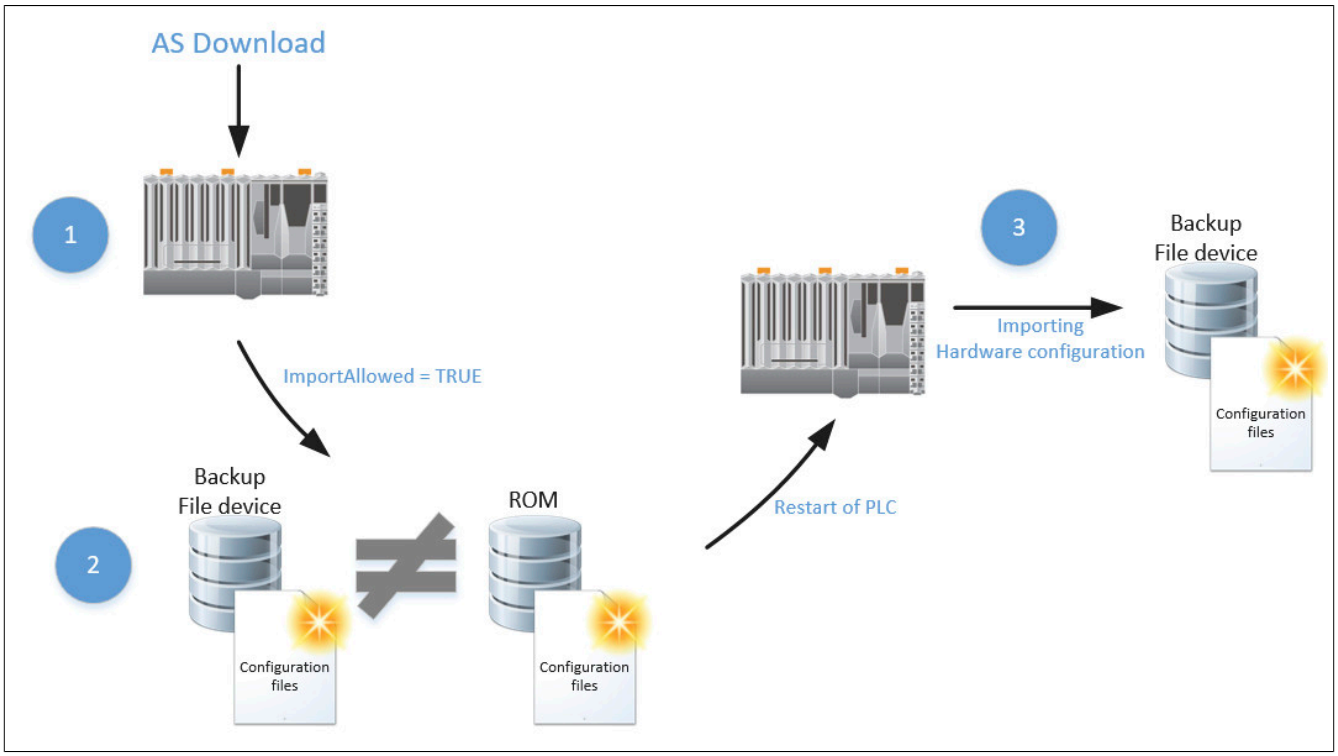
Überprüfung ob Hardware-Konfiguration Widerhergestellt werden soll

Bei der Aktivierung von mapp IO wird automatisch geprüft, ob die Hardware-Konfiguration vom Backup File Device und die aus der ROM geladene Hardwarekonfiguration, identisch sind. Ist das nicht der Fall, dann wurde die Hardware-Konfiguration extern (= nicht durch mapp IO) verändert. mapp IO steht nun vor der Entscheidung die neue Hardware-Konfiguration zu behalten, oder die zuletzt über mapp IO importierte Konfiguration erneut zu aktivieren. Diese Entscheidung kann der Benutzer durch den Parameter "ImportAllowed" an MpIOImport steuern.

Ist der Eingang "ImportAllowed = `TRUE`", startet die Steuerung neu, und nach dem Neustart ist wieder die importierte Hardware-Konfiguration aktiv. Ist diese automatische Wiederherstellung nicht erwünscht, kann sie mithilfe des Eingangs "ImportAllowed = `FALSE`" an MpIOImport verhindert werden.

Das heißt auch nach einem Automation Studio Download (1) kann die Steuerung automatisch nocheinmal starten ("ImportAllowed = `TRUE` "), falls die Konfigurationen nicht übereinstimmen (2). Nach dem Neustart wird die Konfiguration welche am Backup File Device liegt importiert (3). Mithilfe von "ImportAllowed=`FALSE`" ist die automatische Wiederherstellung ausgeschalten.

> **Durch "ImportAllowed" können zwei verschiedene Funktionen in mapp IO verwendet werden:**
>
> - **Wiederherstellen der Hardwarekonfiguration**
> - **Sicherheitsmechanismus beim Importieren einer Konfiguration**
>
> **Für mehr Information siehe hier.**



4.1.5 Aufbau der Import-Datei

In diesem Kapitel wird der Aufbau der Import-Datei beschrieben. Das Format der Import-Datei ist kompatibel mit der IO-Zuordnungstabelle, der AR-Konfiguration und der NC-Zuordnungstabelle, welche von Automation Studio erstellt werden.

Wie der Einsatz einer Import-Datei zur Laufzeit aussehen kann, wird im Anwendungsfall Auswählen der Hardware-Konfiguration zur Laufzeit erklärt.

> **Es müssen alle gewünschten Module in der Import-Datei angegeben werden, welche auch im Automation Studio sichtbar sind. Manche Module sind mit sogenannten versteckten Modulen verlinkt welche nicht in Automation Studio sichbar sind. Dazu gehört zum Beispiel das Modul X67BC8321.L12. Es enthält ein verstecktes Modul (X67BC8321.L12io). Da dieses aber auch nicht in Automation Studio angezeigt wird, muss es auch nicht in der Import-Datei angegeben werden!**

Übersicht der XML-Elemente

Folgende XML-Elemente können in der Import-Datei verwendet werden:

- **IOCFG**: Über "IOCFG" wird der Wurzelknoten angegeben
- **Module**: Durch "Module" kann ein Modul definiert werden
- **Parameter**: Mithilfe dieses Elements kann die Konfiguration eines Modules verändert werden.
- **IO Channel**: "IO Channel" definiert die Angabe des IO-Kanals. Muss nur angegeben werden, falls die IO-Mapping Struktur verändert werden soll
- **LN**: Über "LN" kann die IO-Mapping Informationen für das Modul angegeben werden
- **NcObject**: Durch dieses Element kann die NC-Mapping Informationen für das Modul bestimmt werden

> **Die Reihenfolge wie die Elemente angegeben werden, wie beispielsweise <Parameter>, <LN> oder <IO> spielt in der Import-Datei keine Rolle. Beim Importieren holt sich mapp IO automatisch die Information in der benötigten Reihenfolge.**

Hier angehängt die XML Schema Definition für die Importdatei: Import

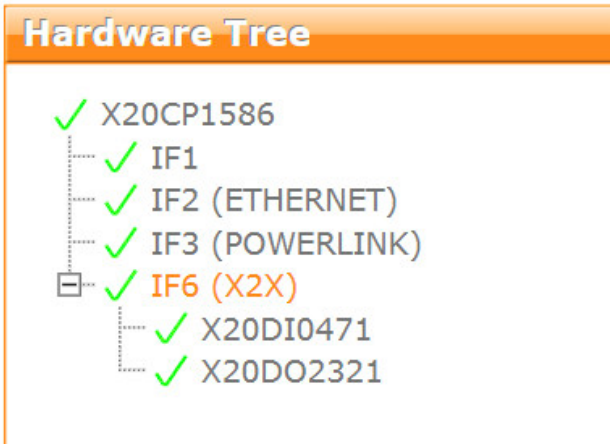
Beispiel einer vollständigen Import-Datei

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />
```

```
    <Module ID="IF6.ST1" Source="AR" SourceName="MyInputModule" />
    <Module ID="IF6.ST1.IO">
        <LN ID="DigitalInput01" DPName="diInput1" />
    </Module>

    <Module ID="IF6.ST2" Source="AR" SourceName="MyOutputModule" >
        <LN ID="DigitalOutput01" DPName="doOutput1" />
    </Module>
</IOCFG>
```

Über den Module-Knoten mit der ID "$root" wird die CPU aus der aktuellen Konfiguration übernommen.

Nach einem Import mit dieser Datei wird die Hardware-Konfiguration so aussehen:



4.1.5.1 IOCFG

Der IOCFG-Knoten ist der Wurzelknoten der XML-Datei und wird daher in jeder Import-Datei benötigt. Innerhalb dieses Wurzelknotens können nun die gewünschten Module, IO-Anpassungen usw. angegeben werden.

Attribut

- **Version:** Über dieses Attribut kann die Version der Konfiguration definiert werden. Die Version dient dem Benutzer zur Information.

Beispiel:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <!-- Definition of modules -->

</IOCFG>
```

4.1.5.2 Module

Durch "Module" wird ein neues Modul in der Konfiguration definiert. Dabei kann eine Modulkonfiguration sowie ein IO-Mapping enthalten sein. Wird ein Modul importiert, wird auch dessen Schnittstellen-Eigenschaften importiert. Das bedeutet, wenn ein Modul eine bestimmte Schnittstelle verwendet, so wird diese auch automatisch mitimportiert.

Attribute

- **ID:** Definiert die Adresse / das Interface, an dem das Modul nach dem Import angeschlossen ist (z.B. "IF6.ST1"). Dieser Parameter muss immer definiert werden.
- **Hardware:** Eindeutiger Identifikationsstring des I/O Moduls (z.B. "X20DI9371").
- **SourceName:** Name des Moduls in der referenzierten Konfiguration (z.B. "MyModule").
- **SourceID:** Definiert die Adresse/ das Interface des Moduls in der referenzierten Konfiguration (z.B. "IF6.ST7").
- **Source:** Definiert, wo die Konfiguration des Moduls gefunden werden kann (z.B. "AR"). Dieser Parameter kann optional angegeben werden, da standardmäßig immer "AR" verwendet wird.
- **ImportMode:** Definiert, wie das Modul importiert werden soll ("z.B. "ModuleOnly"). Dieser Parameter kann optional angegeben werden, da standardmäßig immer "ModuleWithSubslots" verwendet wird.

Die Attribute "Hardware", "SourceName" und "SourceID" geben an, welches Modul importiert werden soll. Eines von den drei Attributen kann verwendet werden. Bei der Verwendung von Template Dateien wird immer "SourceName" verwendet.

> **Es kann sein, dass beim Importieren von Modulen zusätzliche Parameter definiert werden müssen. In manchen AO-Modulen muss zum Beispiel ein oberes oder unteres Limit angegeben werden. Ob und wie diese Parameter definiert werden müssen, kann durch die arconfig.br Datei herausgefunden werden. Das gewünschte Modul wird eingefügt und das Projekt wird kompiliert. In der daraus entstehenden arconfig.br Datei können diese Parameter überprüft werden.**

Beispiel:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />

    <Module ID="IF6.ST1" Source="AR" SourceName="MyInputModule" ImportMode="ModuleOnly" />
    <Module ID="IF6.ST1.IO">
        <LN ID="DigitalInput01" DPName="diInput1" />
    </Module>

    <Module ID="IF6.ST2" Source="AR" SourceName="MyOutputModule" >
        <LN ID="DigitalOutput01" DPName="doOutput1" />
    </Module>
</IOCFG>
```

In diesem Beispiel wird das Element "LN" verwendet um das IO-Mapping anzupassen. Für mehr Information dazu siehe hier.

Mithilfe der Anweisung "$root" wird die Information der Master-Konfiguration in die Import-Datei kopiert. Dadurch können die Module der Master-Konfiguration in der Import-Datei verwendet werden. Als Master-Konfiguration wird die Konfiguration des originales Automation Studio Projekts bezeichnet.

ID

Die ID definiert an welcher Stelle in der Hardware-Topologie das Modul eingefügt wird. Die Position des Moduls wird definiert als:

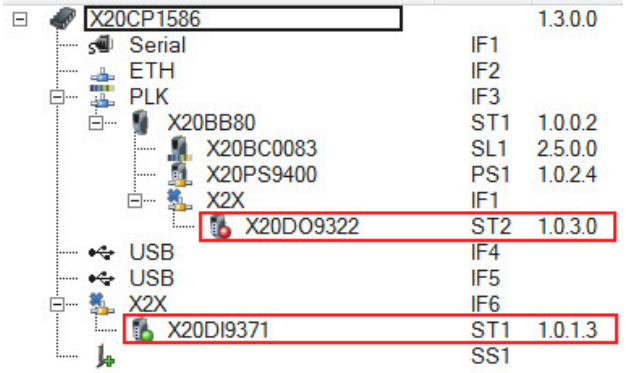
```
[SL<slotnr>][.SS<sslnr>] [[.IF<ifnr>.ST<stanr>][.SL<staslotnr>][.SS<stasslnr>]].<chnr>
```

Attribut	Beschreibung
slotnr	Slot Nummer
sslnr	Subslot Nummer
ifnr	Interface Nummer
stanr	Stations-Nummer
staslotnr	Stations-Slot Nummer
stasslnr	Stations-Subslot Nummer
chnr	Kanalnummer

Beispiel:

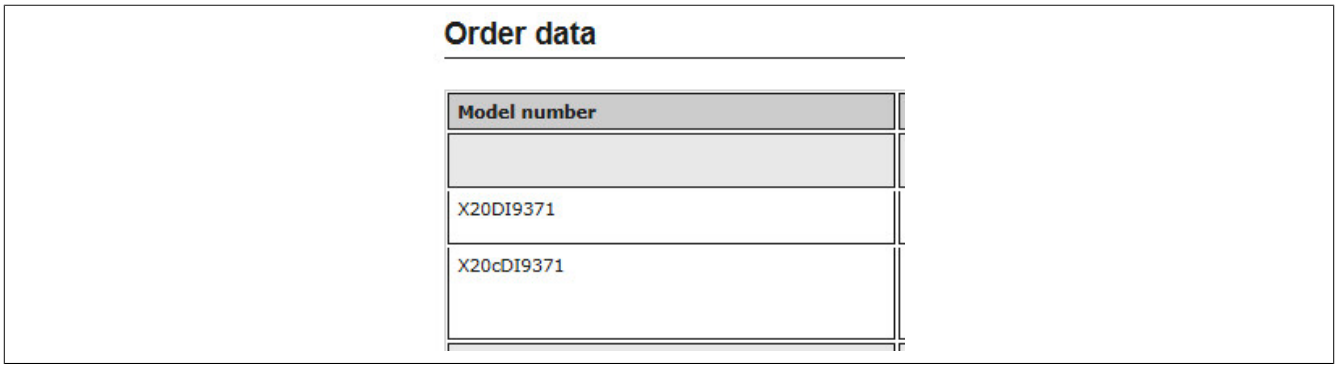
```
<Module ID="IF6.ST1" Hardware="X20DI9371" />
<Module ID="IF3.ST1.IF1.ST2" Hardware="X20DO9322" />
```

In diesem Beispiel werden die folgenden rot markierten Module eingefügt:



Hardware

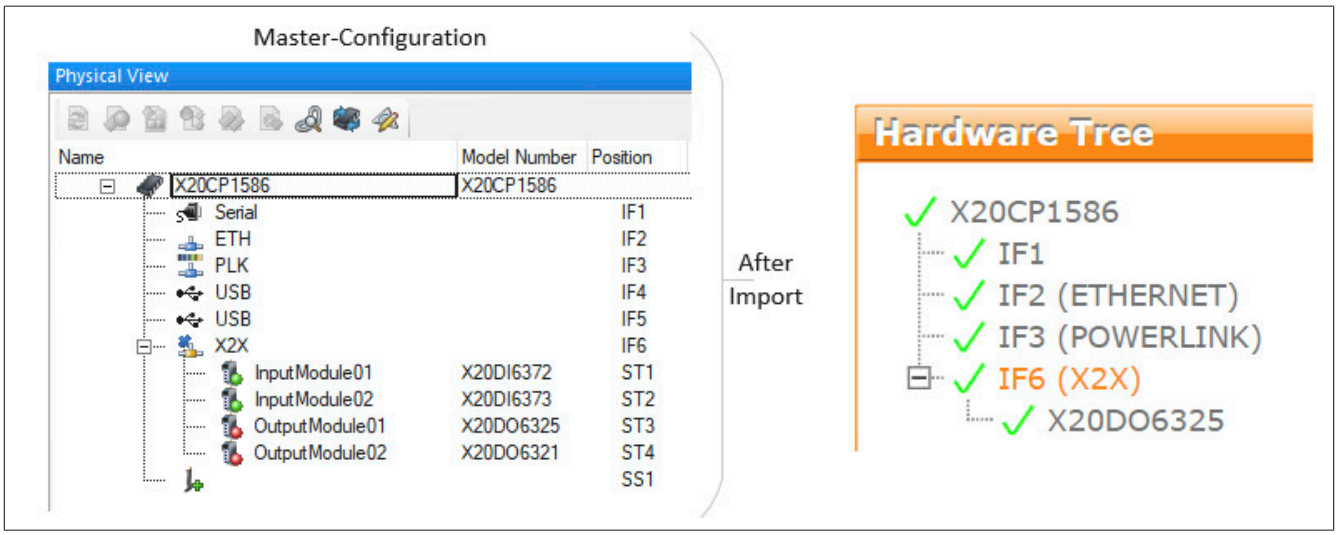
Eindeutige Identifikation des Moduls. Der Identifikationsstring kann der Hilfe des Moduls entnommen werden. Anstelle von "Hardware" kann "SourceName" bzw. "SourceID" verwendet werden.



SourceName

Name des Hardwaremoduls in der referenzierten Konfiguration. Kann verwendet werden, wenn das Modul in der Automation Studio Konfiguration benannt ist. Anstelle von "SourceName" kann "Hardware" bzw. "SourceID" verwendet werden.

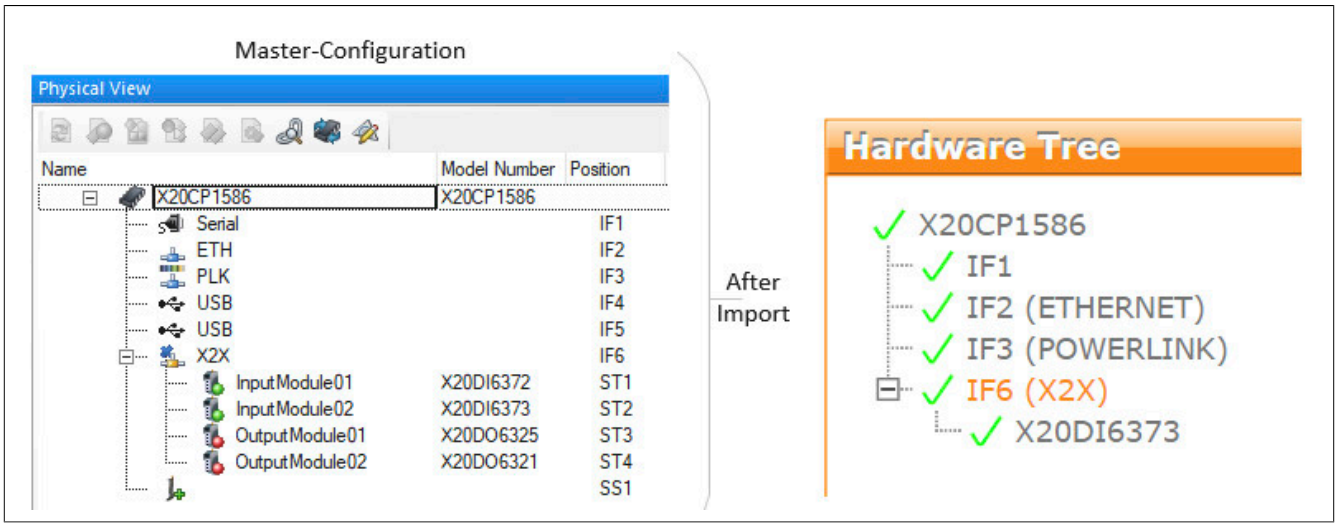
```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />
      <Module ID="IF6.ST1" Source="AR" SourceName="OutputModule01" />

</IOCFG>
```



SourceID

Definiert die Adresse/ das Interface des Moduls in der referenzierten Konfiguration. Anstelle von "SourceID" kann "Hardware" bzw. "SourceName" verwendet werden.

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

     <Module ID="$root" Source="AR" SourceID="$root" />
     <Module ID="IF6.ST1" Source="AR" SourceID="IF6.ST2" />

</IOCFG>
```



Source

Dieser Parameter kann optional angegeben werden. "Source" definiert wo die Konfiguration der Module gefunden werden kann. Standardwert ist "AR".

AR: Konfiguration aus der Master-Konfiguration wird verwendet.

Template: Konfiguration aus den Template Dateien wird verwendet.

```
<Module ID="IF3.ST1" Source="Template" SourceName="DI2377" />
```

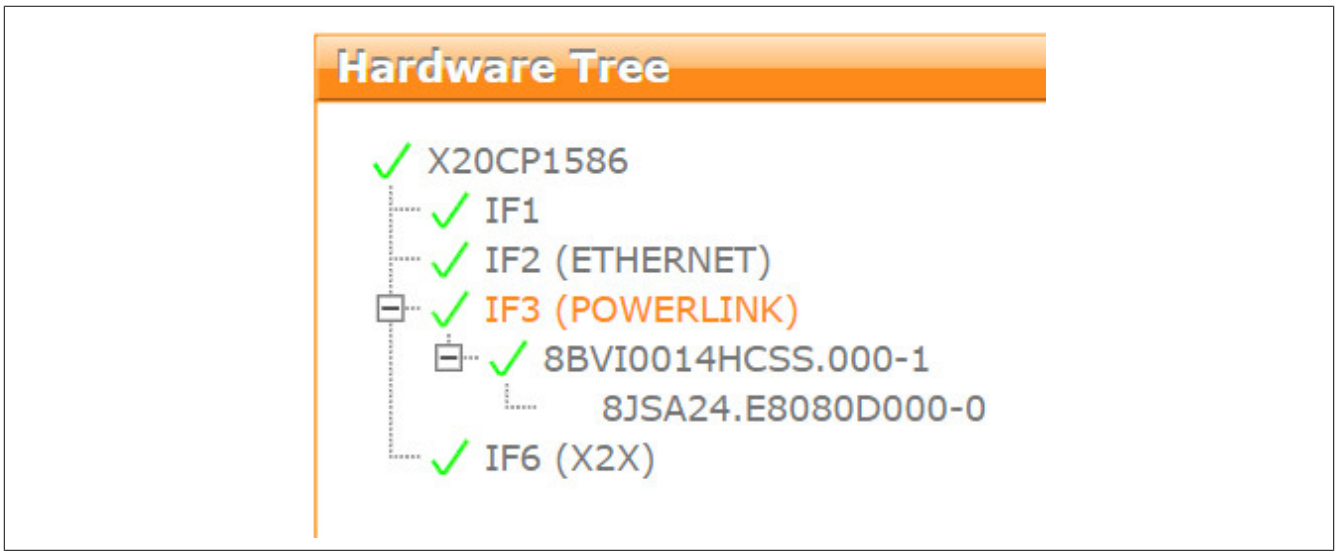
In diesem Beispiel wurden Template-Dateien mit dem Namen "DI2377" erstellt. Wie Template-Dateien verwendet werden können siehe Abschnitt Template-Dateien.

ImportMode

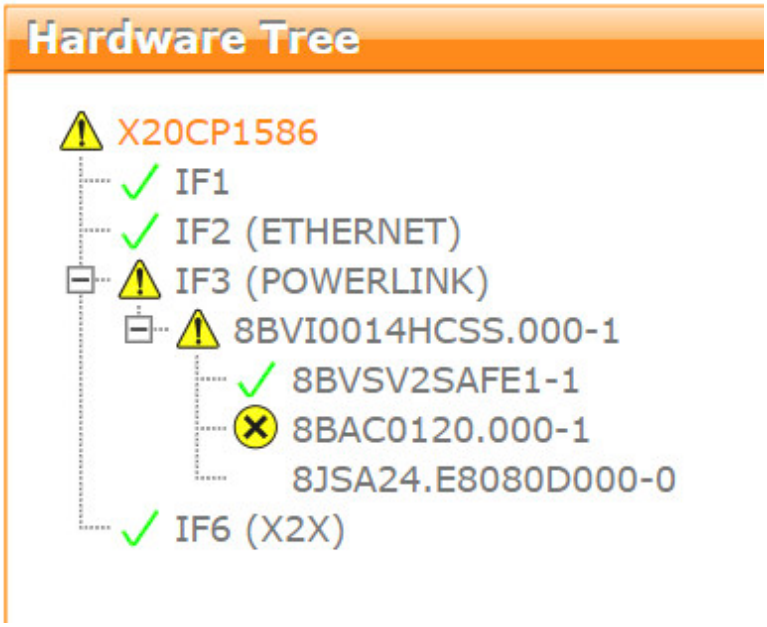
Dieser Parameter ist optional. Die folgenden Beispiele importieren einen 8BVI0014HCSS.000-1-Modul, welches bereits in der AR-Konfiguration vorhanden ist, mit verschiedenen Import-Modi. Wird kein "ImportMode" angegeben, wird immer *ModuleWithSubslots* verwendet.

Der Import-Modus kann definiert werden als:

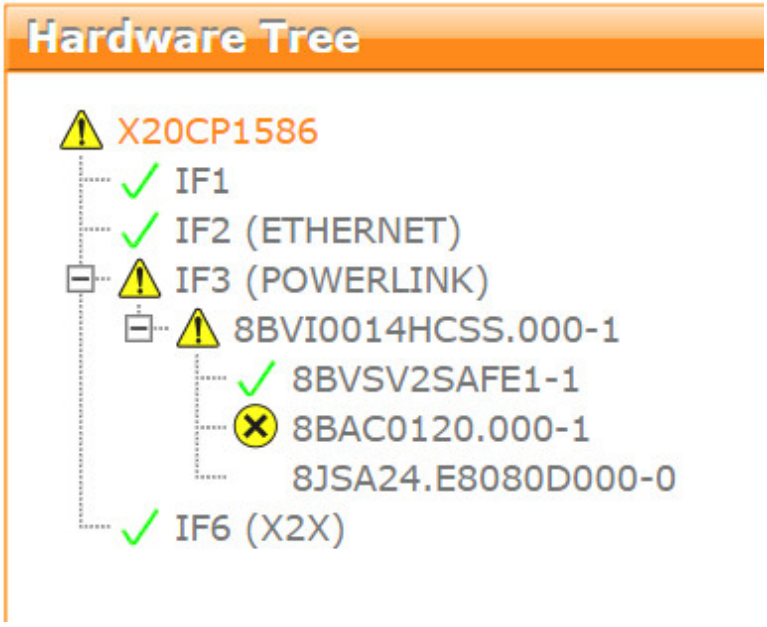
ModuleOnly: Es wird nur das angegebene Module eingefügt.



ModuleWithSubslots: Es wird das angegebene Module mit allen seinen Subslots eingefügt.



Branch: Es wird das angegebene Modul mit allen Subslots und Submodulen eingefügt.



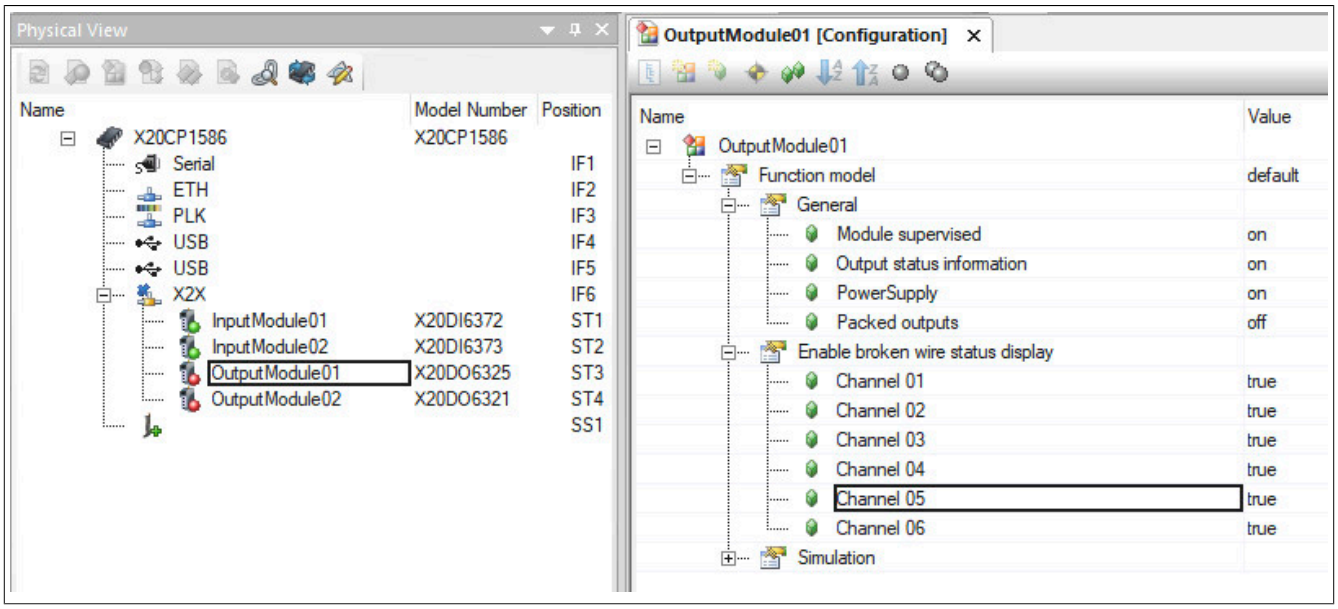
4.1.5.3 Parameter

Mithilfe dieses Elements kann die Konfiguration für jedes Modul individuell verändert werden.

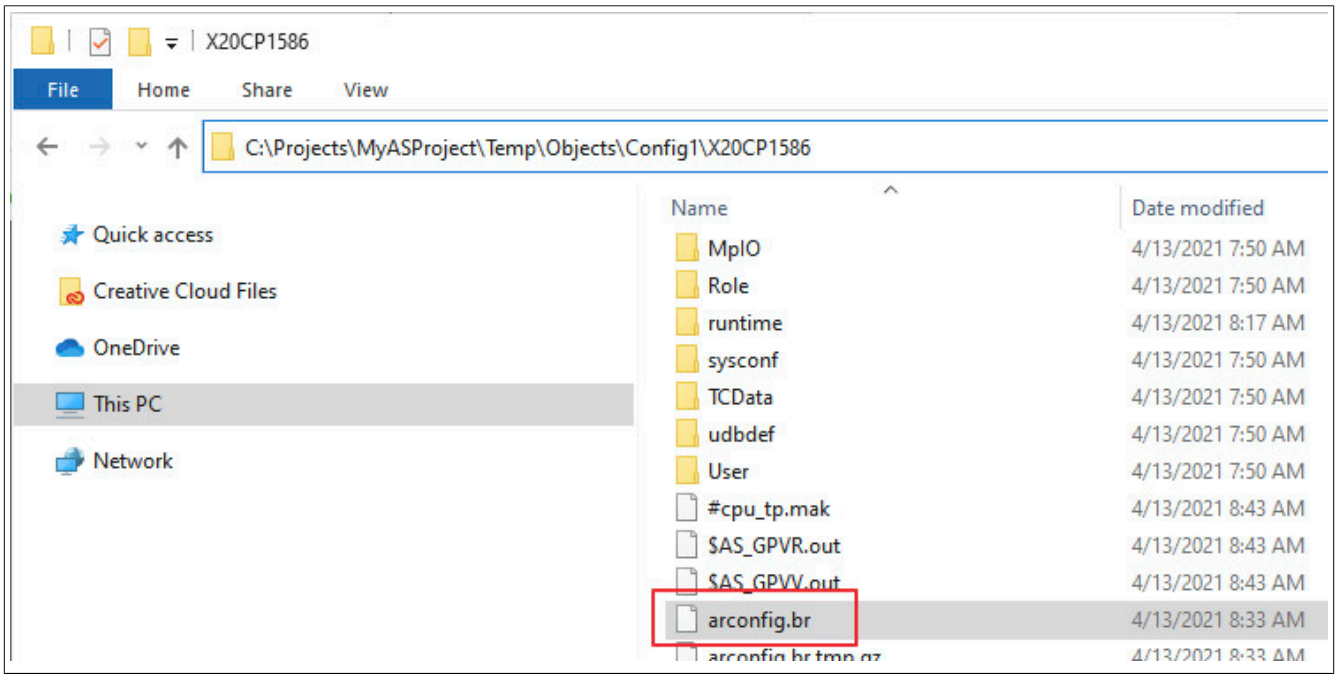
Attribute

- **ID**: Eindeutige Identifikation eines Parameters in einer Gruppe oder der Konfiguration.
- **Value**: Neuer Wert des Parameters.

Ein Modul kann unterschiedlichste Konfigurations-Parameter aufweisen:



Um den Namen des Konfigurations-Parameters zu bekommen, wird die arconfig.br Datei des Projekts verwendet. Die arconfig.br Datei befindet sich im "Temp" Ordner des Automation Studio Projektes.



Die arconfig.br Datei kann mit einem Texteditor geöffnet werden. In der Datei werden die Parameter-Namen des Moduls angezeigt. Befindet sich der gewünschte Parameter im IO-Bereich, muss "IO" unter "Module ID" (z.B.: ""IF6.ST1.**IO**"") angegeben werden.

```
MyImport.xml   arconfig.br
132     <Parameter ID="FunctionModelNumber" Value="0" Type="USINT" />
133     <Parameter ID="Supervision" Value="on" />
134     <Parameter ID="InputFilter" Value="10" Type="USINT" />
135     <Parameter ID="DigitalInputsPacked" Value="off" />
136   </Module>
137   <Module ID="IF6.ST3" Hardware="X20DO6325">
138     <Parameter ID="HardwareModuleName" Value="OutputModule01" />
139   </Module>
140   <Module ID="IF6.ST3.IO" Hardware="X20DO6325">
141     <Parameter ID="FunctionModel" Value="Standard" />
142     <Parameter ID="FunctionModelNumber" Value="0" Type="USINT" />
143     <Parameter ID="Supervision" Value="on" />
144     <Parameter ID="StatusInfo" Value="on" />
145     <Parameter ID="PowerSupply01" Value="on" />
146     <Parameter ID="DigitalOutputsPacked" Value="off" />
147     <Parameter ID="OLEnableOutput01" Value="true" />
148     <Parameter ID="OLEnableOutput02" Value="true" />
149     <Parameter ID="OLEnableOutput03" Value="true" />
150     <Parameter ID="OLEnableOutput04" Value="true" />
151     <Parameter ID="OLEnableOutput05" Value="true" />
152     <Parameter ID="OLEnableOutput06" Value="true" />
153   </Module>
```

Die Änderung eines Paramters kann in einer Import-Datei folgendermaßen aussehen:

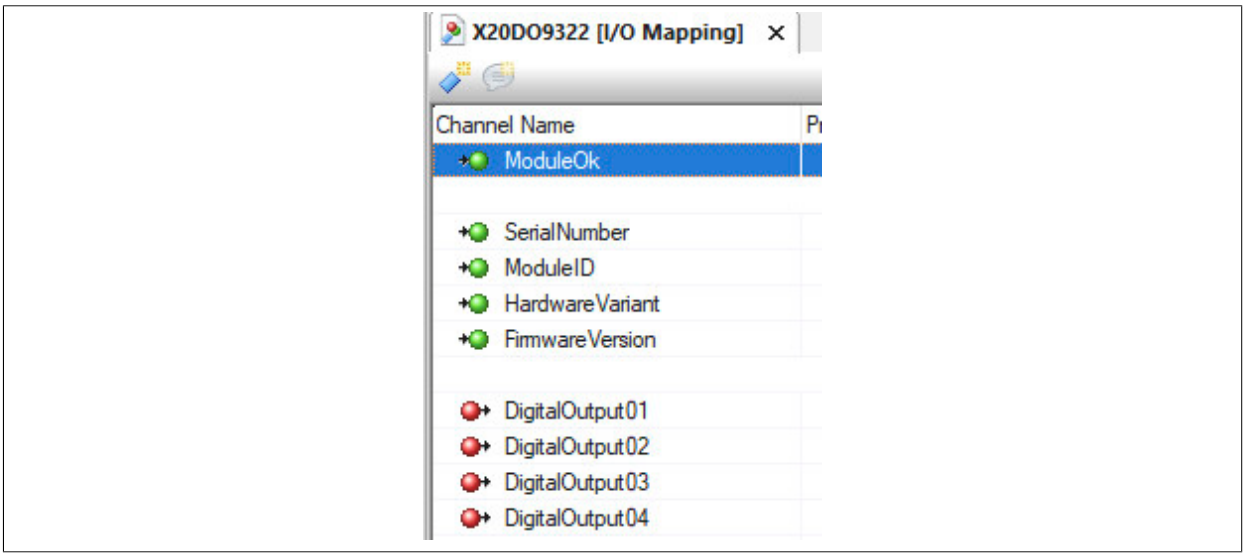
```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />
    <Module ID="IF6.ST1.IO" Source="AR" SourceID="IF6.ST3" >
        <Parameter ID="PowerSupply01" Value="off" />
    </Module>

</IOCFG>
```

4.1.5.4 LN

Mithilfe dieses Elements kann für ein Modul ein IO-Mapping definiert werden.

Attribute

- **ID**: Name des Datenpunktes, für welchen ein IO-Mapping erstellt wird (z.B. "DigitalOutput03")



- **DPName**: Name der Prozessvariable, die verbunden wird (z.B. "Program:MyPV")

- **Device**: Definiert, in welcher Task-Klasse der Wert aktualisiert wird. Wird kein Wert angegeben, wird immer "TC#1-CPYDEV" (erste Task-Klasse) als Task-Klasse genommen. Bei der Angabe von "TC#2-CPYDEV" wird die zweite Task-Klasse verwendet, bei "TC#3-CPYDEV" die dritte usw.
- **Inverted:** Mithilfe von "Inverted" kann ein Eingang invertiert werden. Für Eingänge bedeutet dies, dass der am jeweilige Kanal anliegende Wert genommen, invertiert und auf die zugeordnete Variable übertragen wird. Für Ausgänge ist diese Funktionalität nicht verfügbar. Soll der Eingang invertiert werden, so kann Inverted="`true`" angegeben werden. Wird nichts angegeben, so wird der Eingang auch nicht invertiert.
- **Mode**: Definiert, wie bereits existierende Einträge behandelt werden sollen (z.B. "replace"). Für den "Mode" gibt es folgende Möglichkeiten:
  - **Add**: Fügt das definierte Mapping hinzu, ohne bereits vorhandenene zu Entfernen. Dieser Modus ist nur für Eingänge möglich und ist für diese der Standardwert.
  - **Replace**: Ersetzt bereits existierende Mappings mit dem in der Import-Datei definierten. Standard-Modus für Ausgänge.
  - **Clear**: Löscht ein existierendes Mapping ohne ein neues zu definieren.

Das IO-Mapping für ein Modul wird folgendermaßen definiert:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

     <Module ID="IF3.ST1.IF1.ST2" Hardware="X20DO9322" >
        <LN ID="DigitalOutput03" DPName="doOutput2" />
    </Module>

  </IOCFG>
```

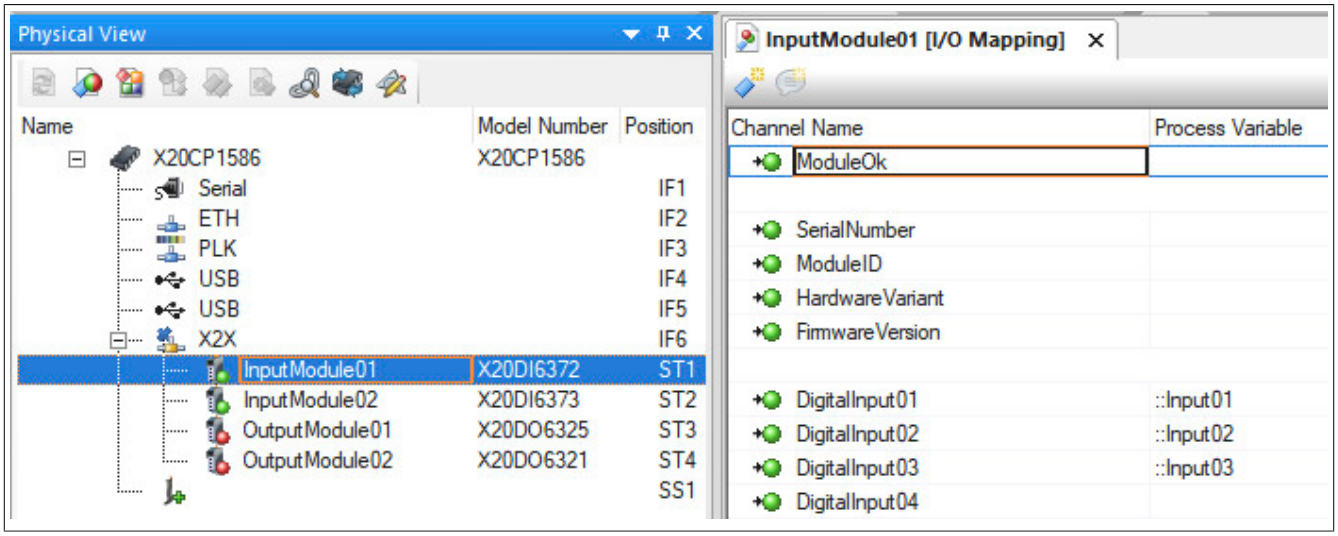
Diese Angabe gilt auch für 3rd party Hardware. Für mehr Information dazu siehe Umgang mit spezieller Hardware.

Der Module-Knoten wird erst nach allen LN-Knoten geschlossen!

Beispiele:

Replace

In diesem Beispiel wird erklärt, wie der Modus "Replace" verwendet werden kann. Im Projekt sind folgende Variablen auf die digitalen Eingänge gemappt:

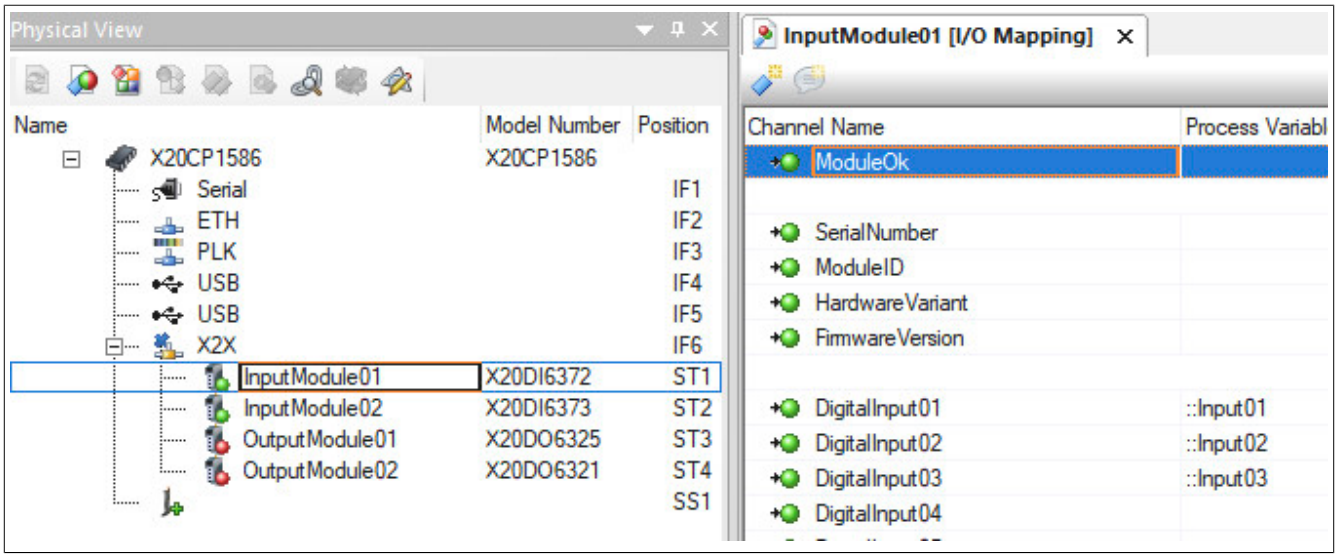


Sollen die gemappten Prozessvariablen getauscht werden, so kann folgendes in der Import-Datei angegeben werden:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />
 <Module ID="IF6.ST1" Source="AR" SourceName="InputModule01" >
   <LN ID="DigitalInput01" Mode="Replace" DPName="::Input02" />
   <LN ID="DigitalInput02" Mode="Replace" DPName="::Input01" />
 </Module>

</IOCFG>
```

Clear

In diesem Beispiel wird erklärt, wie der Modus "Clear" verwendet werden kann. Im Projekt sind folgende Prozessvariablen auf die digitalen Eingägne gemappt:



Soll das Mapping gelöscht werden, so kann folgendes in der Import-Datei angegeben werden:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />
 <Module ID="IF6.ST1" Source="AR" SourceName="InputModule01" >
   <LN ID="DigitalInput01" Mode="Clear" />
   <LN ID="DigitalInput02" Mode="Clear" />
   <LN ID="DigitalInput03" Mode="Clear" />
 </Module>

</IOCFG>
```

4.1.5.5 NcObject

Durch "NcObject" wird ein neues NC-Mapping für das Modul importiert. Weitere Infos zu Objekten aus der Antriebstechnik kann hier gefunden werden.

mapp IO kann nur in Kombination mit ACP10/ARNC0 verwendet werden!

Attribute

- **Name**: Name der Achsenvariable, welche auf das Modul verbunden wird.
- **Channel**: Der Kanal des Moduls, auf den die Variable verbunden wird.
- **InitParameterModule:** Angabe der Init-Parameter Tabelle. Diese Tabelle muss sich auf der Steuerung befinden. Die Datei wird nicht durch mapp IO erstellt!
- **AcoposParameterModule:** Angabe der Acopos-Parameter Tabelle. Diese Tabelle muss sich auf der Steuerung befinden. Die Datei wird nicht durch mapp IO erstellt!

Beispiel:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

     <Module ID="IF3.ST1" Source="AR" SourceName="MAcopos" >
        <NcObject Name="myAxis" Channel="1" InitParameterModule="gAxis01i" AcoposParameterModule="gAxis01a"/>
    </Module>

</IOCFG>
```

Der Module-Knoten wird erst nach allen NcObject-Knoten geschlossen!

4.1.5.6 Ändern der IO-Mapping Struktur

Durch "IO Channel" wird der IO-Kanal angegeben. Dieser wird nur definiert, falls die IO-Mapping Struktur verändert werden soll.

Attribute

- **Kind:** Gibt an wie der IO-Kanal geändert werden soll. (Für löschen wird zum Beispiel Kind="Clear" verwendet).
- **Type:** Mit "Type" muss der Datentyp des IO-Kanals angegeben werden.

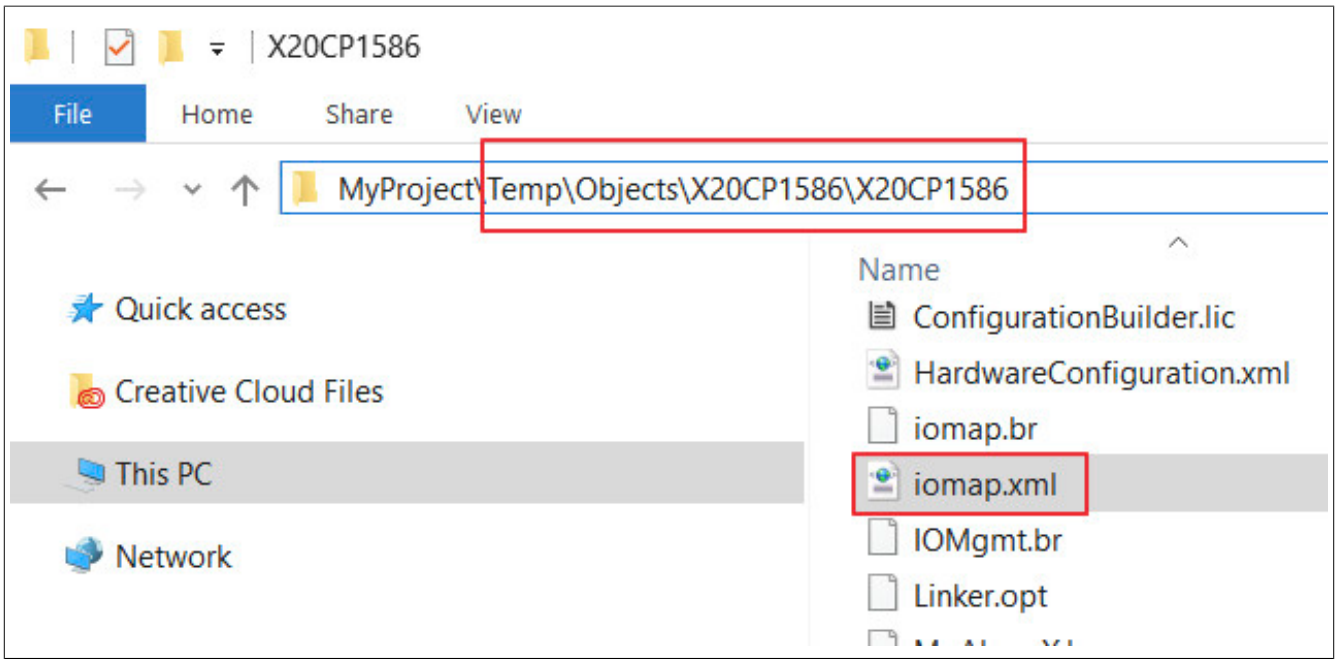
Bei manchen Mischmodulen, wie zum Beispiel das X67BC7321 Modul, kann ein IO-Kanal die Richtung ändern. Soll die Richtung geändert werden, so muss dies im Parameter-Knoten vermerkt werden. Sind zum Beispiel alle IO-Kanäle als Eingänge definiert und es soll einer davon als Ausgang angegeben werden, so wird dies folgendermaßen definiert:

```
<Parameter ID="Direction04" Value="output" />
```

Dadurch ändert sich die Richtung vom IO-Kanal 4 zu einem Ausgang. Damit ist es notwendig den "DigitalInput04" aus dem IO-mapping zu entfernen (Kind="Clear") und "DigitalOutpu04" hinzuzufügen. Dann kann man optional eine PV zu diesem Ausgang mappen:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

     <Module ID="IF7.ST1" Source="AR" SourceName="MyMixedModule">
        <Parameter ID="Direction04" Value="output" />
        <IO Channel="DigitalInput04" Kind="Clear" />
        <IO Channel="DigitalOutput04" Kind="Out" Type="BOOL" />
        <LN ID="DigitalOutput04" DPName="MyTestVar"/>
    </Module>

</IOCFG>
```

Welche Information der Digitale Ausgang 04 hat, kann in der iomap.xml Datei herausgefunden werden. Dabei muss die gewünschte Änderung in AS durchgeführt werden (Eingang auf Ausgang umändern) und kompiliert werden. Im Temp-Verzeichnis kann anschließend zur iomap.xml Datei navigiert werden:

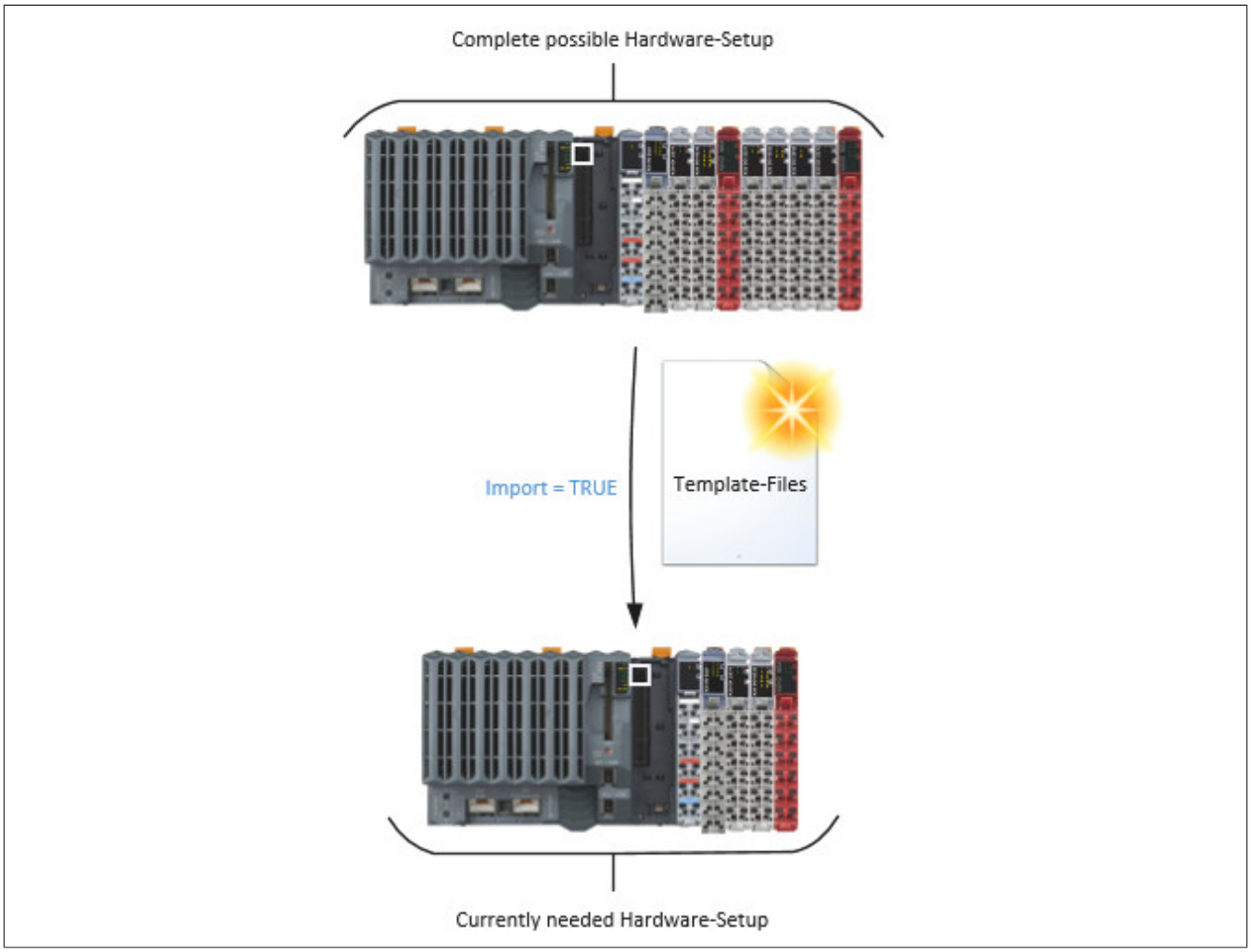


4.1.6 Template Dateien

Zu diesem Thema gibt es das Getting Started Hardware Konfiguration zur Laufzeit über Template-Dateien verwalten.

Template Dateien sind Vorlagen für bestimmte Hardware-Module und stellen einer weitere Variante dar Hardware-Module in die Hardware-Konfiguration einzubinden.

Im Abschnitt Grundprinzip wurde die Variante vorgestellt, eine Hardware-Konfiguration aus Automation Studio mit allen möglichen Modulen auf die Steuerung zu übertragen. Über die Import-Datei wird dann entschieden, welche Module an welcher Position in der Hardware-Konfiguration sitzen sollen.



Template Dateien bieten eine weitere, flexiblere, Variante Hardware-Module zu refernzieren. Der Hauptunterschied, besteht darin, dass beim Parameter "Source" in der zu importierenden Konfiguration nicht *AR,* also die über Automation Studio übertragene Hardware-Konfiguration, ausgewählt wird, sondern *Template* (siehe Code-Beispiel unten). Damit wird nach einer Template Datei im Template File Device ("Template Device Name" in der MpIO Konfiguration) gesucht. Pro Modul muss es zwei Template Dateien geben:

- **MyTemplate.ar:** (z.B. *X20DI9371.ar*) Diese Datei enthält die gesamte Modul-Konfiguration in gleicher Weise, wie sie auch in der AR-Konfiguration enthalten sind.
- **MyTemplate.io:** (z.B. *X20DI9371.io*) Diese Datei enthält die gesamte I/O Zuordnung in gleicher Weise, wie sie auch in der I/O Zuordnungstabelle enthalten sind.

Der Name der Template Dateien kann, bis auf die Endung, beliebig gewählt werden. In der Import-Datei wird das Template über den Dateinamen, ohne Endung, angegeben:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">
    <Module ID="$root" Source="AR" SourceID="$root" />
    <!-- ... -->
    <Module ID="IF6.ST1" Source="Template" SourceName="MyTemplate" />
</IOCFG>
```

> **In jeder Template Datei darf nur ein Hardware-Modul definiert werden. Eine Ausnahme dazu bilden Bus-Koppler. Bei diesen Modulen können in der Template Datei auch alle Submodule / Subslots hinzugefügt werden, welche, je nach *ImportMode*, ebenfalls importiert werden.**

Erstellen von Template Dateien

Im unteren Teil des Abschnitts kann ein Beispiel für eine Template-Datei heruntergeladen werden.

Template Dateien werden erstellt, indem die arconfig.br Datei einer bestehenden Konfiguration geöffnet wird. Dazu wird der Pfad zu einem AS Projekt geöffnet, welches das gewünschte Modul beinhaltet. Im "MyASProject/Temp/Object" Ordner befindet sich die "arconfig.br" Datei. In dieser Datei sind alle verwendeten Module und deren Eigenschaften eingetragen. Als Beispiel wird hier das X20DI9371-Modul verwendet:

```
<Module ID="IF6.ST1" Hardware="X20DI9371">
    <Parameter ID="HardwareModuleName" Value="X20DI9371" />
</Module>
<Module ID="IF6.ST1.IO" Hardware="X20DI9371">
    <Parameter ID="FunctionModel" Value="Standard" />
    <Parameter ID="FunctionModelNumber" Value="0" Type="USINT" />
    <Parameter ID="Supervision" Value="on" />
    <Parameter ID="InputFilter" Value="10" Type="USINT" />
    <Parameter ID="DigitalInputsPacked" Value="off" />
</Module>
```

Die Modul ID in der Template-Datei kann eine beliebige ID sein. An welchem Platz die Template-Datei eingefügt wird, wird über die zu importierende Datei bestimmt!

Diese Parameter werden in eine Datei mit dem Namen "DI9371.ar" kopiert. Die Stuktur dieser Datei muss wie folgt aussehen:

```
<?AutomationRuntimeIOSystem Version = 1.0?>
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">
    <!-- Add Module Konfiguration here.-->
</IOCFG>
```

Innerhalb des Knotens <IOCFG> wird der oben gezeigte Inhalt eingefügt, damit ist die Template Datei mit der Modul-Konfiguration fertig.

Als nächstes wird die Template Datei mit der I/O Zuordnung erstellt.In der iomap.br können alle Informationen gefunden werden, welche benötigt werden, um die I/O Zuordnung des Moduls in die zweite Template Datei zu packen. Auch die iomap.br Datei kann im Pfad "MyASProject/Temp/Object" gefunden werden. Hier ist ein kleiner Ausschnitt zu dem obigen Beispiel:

```
<LN ID="%IX.IF6.ST1.ModuleOk" Type="BOOL">
    <Prod Device="IF6.ST1" DPName="ModuleOk" Kind="io" />
</LN>
<LN ID="%ID.IF6.ST1.SerialNumber" Type="UDINT">
    <Prod Device="IF6.ST1" DPName="SerialNumber" Kind="io" />
</LN>
<LN ID="%IW.IF6.ST1.ModuleID" Type="UINT">
    <Prod Device="IF6.ST1" DPName="ModuleID" Kind="io" />
</LN>
<LN ID="%IX.IF6.ST1.DigitalInput01" Type="BOOL">
    <Prod Device="IF6.ST1" DPName="DigitalInput01" Kind="io"/>
    <Cons Device="TC#4-CPYDEV" DPName="gdiVariable" Kind="pv"/>
</LN>
```

Alle IO-Datenpunkte des Moduls müssen in einer Template-Datei Namens "DI9371.io" kopiert werden. Die Struktur dieser Datei ist wie folgt:

```
<?AutomationRuntimeIOSystem Version="1.0"?>
<IO xmlns="http://www.br-automation.com/AR/IO">
    <Links>
        <!-- Add Module IO-Information here. -->
    </Links>
</IO>
```

Innerhalb des Knotens <IO> wird der oben gezeigte Inhalt eingefügt, damit ist die Template Datei mit der I/O Zuordnung fertig.

Sind diese beiden Dateien fertig, gibt es ein Template für das Hardware Modul X20DI9371.

Beispiel für eine Template Datei

In diesem Abschnitt kann eine Template-Datei heruntergeladen werden und zu Testzwecken verwendet werden:

- .io Datei: Template_DI2377
- .ar Datei: Template_DI2377

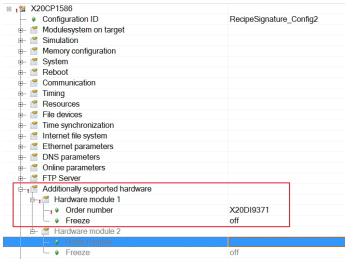
  In einer Import-Datei muss anschließend auf die Template-Datei referenziert werden. Das kann zum Beispiel folgendermaßen aussehen:

  ```
  <IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">
      <Module ID="$root" Source="AR" SourceID="$root" />
  ```

```
    <Module ID="IF6.ST1" Source="Template" SourceName="Template_DI2377" />
</IOCFG>
```

Firmware für Module in Template Dateien

Um sicherzustellen, dass auch die Firmware für Module, welche über Template Dateien referenziert werden, auf der Steuerung ist, wird die Funktionalität "Additionally supported hardware" von Automation Studio verwendet. Diese Option findet man in der Konfiguration der Steuerung:



Befindet sich ein Modul nicht in der Master-Konfiguration und wurde es auch nicht unter "Additionally supported hardware" definiert, befindet sich die Firmware des Moduls nicht auf dem Zielsystem, dadurch kann das Modul zur Laufzeit nicht verwendet werden.

Umgang mit Antriebstechnik bei der Verwendung von Templates

mapp IO bietet auch die Möglichkeit Antriebstechnik über Templates zu verwalten. Dabei wird die Template-Datei nach dem gleichen Prinzip erstellt wie für Module. Folgende Dateien werden verwendet und erstellt:

arconfig.br

Zuerst wird die arconfig.br Datei einer bestehenden Konfiguration geöffnet. Die Daten des verwendeten Antriebs werden in eine neue Datei, wie zum Beispiel "MyTemplate.br" kopiert:

```
<?AutomationRuntimeIOSystem Version = 1.0?>
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">
<Module ID="IF3.ST1" Hardware="8B0P0110HC00.000-1">
<Parameter ID="HardwareModuleName" Value="8B0P0110HC00.000-1" />
<Parameter ID="Mode" Value="CN" />
<Parameter ID="ResponseTimeout" Value="25" Type="UDINT" />
<Parameter ID="MuxNetworkSlot" Value="0" Type="UDINT" />
<Parameter ID="MinimalInputLatency" Value="off" />
<Parameter ID="EPL_Chained" Value="off" />
<Parameter ID="EnableDNA" Value="off" />
<Parameter ID="FunctionModel" Value="DoubleAxis" />
<Parameter ID="InSize" Value="152" Type="UDINT" />
<Parameter ID="OutSize" Value="64" Type="UDINT" />
</Module>
</IOCFG>
```

iomap.br (Optional)

Diese Datei enthält die gesamte I/O Zuordnung. Muss nur verwendet werden, wenn eine I/O Zuordnung gewünscht ist. Wird eine I/O Zuordnung verwendet, so muss die Datei "MyTemplate.io" erstellt werden.

Acp10map.br

In dieser Datei sind Informationen bezüglich des Antriebs gespeichert. Die Information wird kopiert und in eine neu erstellte Template-Datei "MyTemplate.nc" gespeichert:

```
<?AutomationStudio Version="4.4.4.112"?>
<NcDeployment NcSwId="ACP10">  <NcObject Name="gAxis01" InterfaceType="Powerlink" Type="ncAXIS" AcoposSimulation="
<NcObject Name="8B0P0110HC00.000-1_ncAXIS2" InterfaceType="Powerlink" Type="ncAXIS" AcoposSimulation="Off" InitPa
<NcObject Name="8B0P0110HC00.000-1_ncV_AXIS1" InterfaceType="Powerlink" Type="ncV_AXIS" InitParameterModule="" Ac
<NcObject Name="8B0P0110HC00.000-1_ncV_AXIS2" InterfaceType="Powerlink" Type="ncV_AXIS" InitParameterModule="" Ac
</NcDeployment>
```

In der Import-Datei wird die zu verwendende Template-Datei an der gewünschten Stelle ("IF5.ST1") referenziert:

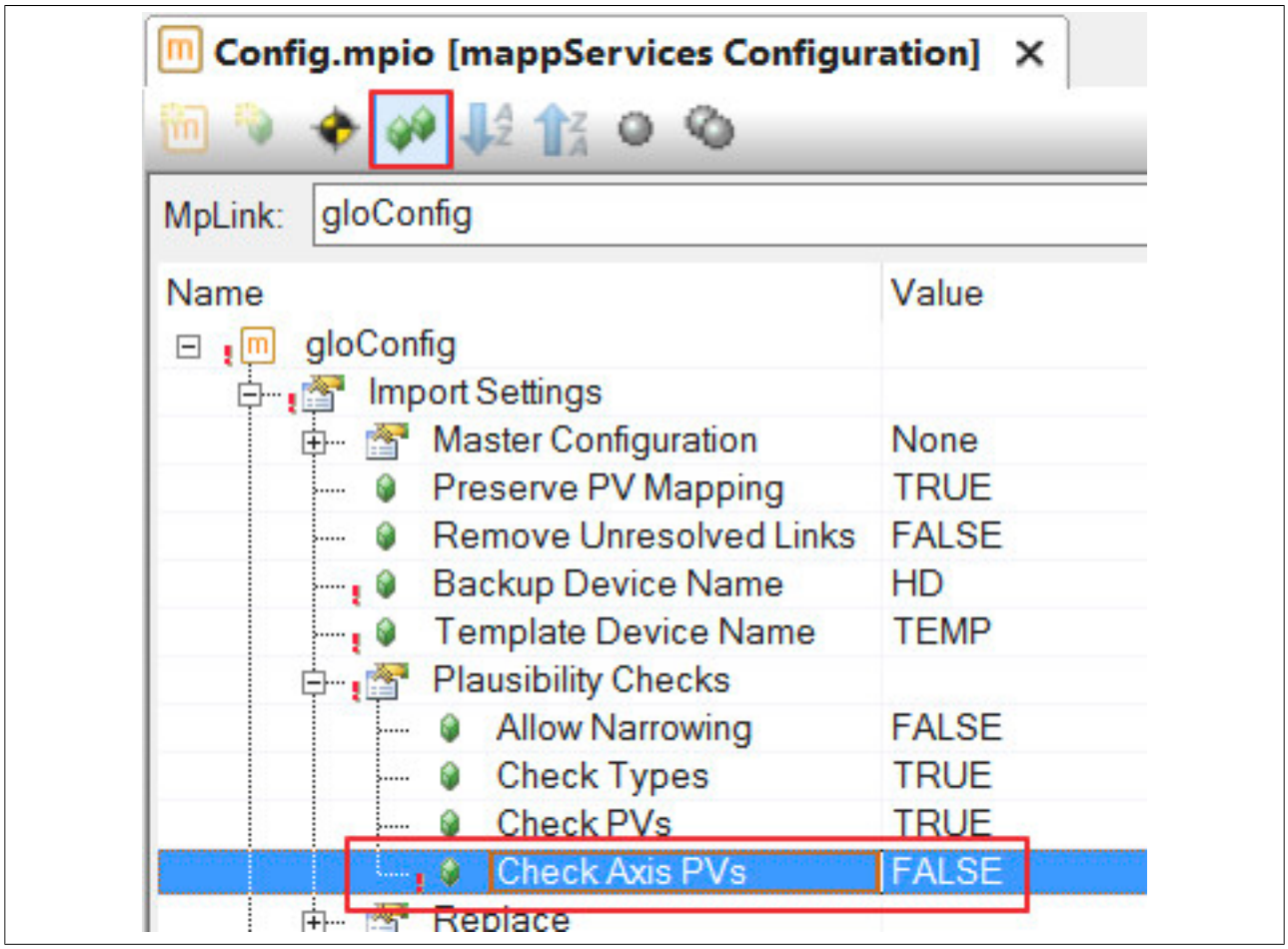
```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">
<Module ID="$root" Source="AR" SourceID="$root" />
<!-- ... -->
<Module ID="IF5.ST1" Source="Template" SourceName="MyTemplate" ImportMode="ModuleWithSubslots" />
```

```
</IOCFG>
```

Über den Import-Modus kann zusätzlich noch definiert werden, ob nur das Modul des angegebenen Templates, das Modul und die zugehörigen Sub-Slots oder der komplette Branch des Templates importiert werden soll. Standardmäßig wird das Modul und die zugehörigen Sub-Slots importiert. Für mehr Information siehe hier im Abschnitt "Module".

> **Die Modul ID in der Template-Datei kann eine beliebige ID sein. An welchem Platz die Template-Datei eingefügt wird, wird über die zu importierende Datei bestimmt!**

> **Wird die Import-Datei geladen, wird automatisch geprüft ob zum Beispiel die Variable gAxis01 auf dem Zielsystem vorhanden ist. Das bedeutet es muss die Variable erstellt werden, oder die Prüfung der Variable wird deaktiviert. Dies funktioniert über die MpIO Konfiguration:**



4.1.7 Umgang mit Spezieller Hardware

Umgang mit Safety Modulen

Sicherheitsmodule werden von MpIO wie graue I/O-Module behandelt. Der einzige Unterschied zu grauen I/O Modulen sind die Cross-Links, Querverweise zwischen Sicherheitsmodul und SafeLogic, welche in der Hardware-Konfiguration der Sicherheitsmodule hinterlegt sind.

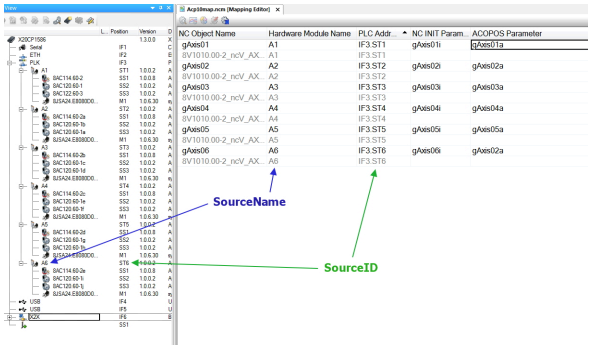
Wird ein Sicherheitsmodul über MpIO bewegt, hinzugefügt oder entfernt sind die Cross-Links obsolet. Das kann zu Fehlern beim Import führen. Um diese Fehler zu vermeiden kann in der MpIO Konfiguration die Eigenschaft "Remove Unresolved Crosslinks" verwendet werden. Wird diese Eigenschaft auf TRUE gesetzt, werden bei einem Import alle Cross-Links, welche nicht mehr aktuell sind, automatisch entfernt.

> **MpIO kann nicht das Programm oder die Funktionsweise der SafeLogic verändern! Wird die Hardware-Konfiguration verändert, muss sichegestellt werden, dass auch die SafeLogic mit den veränderten Bedingungen zurechtkommt.**

Umgang mit Antriebstechnik (ACP10)

mapp IO ermöglicht auch das Zuordnen von Achsobjekten auf einen bestimmten Antrieb. Im Abschnitt NcObject wird beschrieben, wie man ein Achsobjekt in der Import-Datei mit einem Antrieb verbindet. In der Import-Datei kann festgelegt werden welches Achsobjekt sich auf welcher Hardware-Position wiederfindet.

Verändert wird dabei das Achsobjekt, sowie die Zielposition im Hardware-Baum. Nicht verändert werden NC Init Parameter Modul und ACOPOS Parameter Tabelle. Diese werden von der Quelle (definiert über *SourceID* oder *SourceName* in der Import-Datei) übernommen.



Im Bild sieht man eine NC-Zuordnungstabelle mit sechs Achsen. Beim Import sollen die Achsen A2 und A3, sowie A5 und A6 ausgetauscht werden. Außerdem existiert die Achse A4 in der Konfiguration des Kunden nicht. Die dazugehörige Import-Datei könnte folgendermaßen aussehen:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">
    <Module ID="$root" Source="AR" SourceID="$root" />

    <Module ID="IF3.ST1" Source="AR" SourceName="A1" >
        <NcObject Name="gAxis01" Channel="1" />
    </Module>
    <Module ID="IF3.ST2" Source="AR" SourceName="A3" >
        <NcObject Name="gAxis03" Channel="1" />
    </Module>
    <Module ID="IF3.ST3" Source="AR" SourceName="A2" >
        <NcObject Name="gAxis02" Channel="1" />
    </Module>
    <Module ID="IF3.ST5" Source="AR" SourceID="IF3.ST6" >
        <NcObject Name="gAxis06" Channel="1" />
    </Module>
    <Module ID="IF3.ST6" Source="AR" SourceID="IF3.ST5" >
        <NcObject Name="gAxis05" Channel="1" />
    </Module>
</IOCFG>
```

- In der Import-Datei bleibt die Achse A1 unverändert an der Interface-Adresse IF3.ST1, Powerlink Interface mit Knotennummer 1
- Die Achsen A2 und A3 tauschen die Position am Powerlink Interface. Bei der *Module ID* wird für die Achse A3, welche über *SourceName* identifiziert wird, die Adresse IF3.ST2 angegeben. Dadurch ändert sich die Position des Antriebs im Hardwarebaum. Das für Achse A3 verwendete NC Init Parameter Modul und die ACOPOS Parameter Tabelle verändern sich nicht
- Bei Achse A2 ist es genau umgekehrt, die*Module ID* wid auf IF3.ST3 festgelegt
- Die Achse A4 wird komplett weggelassen, beispielsweise weil der Endkunde eine bestimmte Option beim Kauf der Maschine nicht braucht
- Die Achsen A5 und A6 tauschen wieder die Position am Powerlink Interface. Der Unterschied zu dem Tausch zwischen A2 und A3 ist, dass in diesem Fall die Achsen nicht über den *SourceName*, sondern die *SourceID* idenitifiziert werden. Das heißt über die ursprüngliche Adresse in der Hardware-Konfiguration

Umgang mit 3rd-Party Hardware

3rd-Party Hardware wird von mapp IO behandelt wie B&R-Module. Wenn die Hardware über Automation Studio projektiert wird, kann über die Import-Datei die I/O-Zuordnung und die Konfiguration der 3rd-Party Hardware verändert werden.

3rd-Party Hardware kann **immer** wie folgt angegeben werden:

```
<Module ID="IF6.ST2" Source="AR" SourceName="My3rdPartyHardware" >
```

```
    <LN ID="DigitalOutput01" DPName="doOutput1" />
</Module>
```

4.1.8 Auditing in mapp IO

Für diese Komponente können Ereignisse über MpAudit aufgezeichnet werden. Für mehr Information wie ein Eintrag erstellt werden kann, siehe Abschnitt Definieren des Ereignistexts.

Folgende Ereignisse können für diese mapp Komponenten auftreten:

IO Ereignis

Durch diesen Ereignistyp werden Ereignisse in mapp IO aufgezeichnet.

> **Werden Ereignisse über eine mapp View Visualisierung ausgelöst, können manche Ereignisse nicht einem Benutzer zugeordnet werden. Dies muss, bei der Verwendung des "op" Kürzels beachtet werden. Für weitere Information siehe Abschnitt Ereignisse einem Benutzer zuordnen.**

Ereignis ID **96**	**Komponenten** **mapp IO Funktionsbausteine**
Ereignisse	
idx	Liefert den Index des aktuell generierten Ereignisses. Der Wert wird bei jedem neuen Ereignis inkrementiert. Das erste generierte Ereignis in der Ereignisliste hat somit den Wert 1, dass fünfzigste Ereignis hat den Index 50, das hundertste den Index 100 und so weiter.
desc	Beschreibung des Ereignisses
op	Name des Benutzers welcher beim Auftreten des Ereignisses eingeloggt war
ev	Ereignis ID
map	Name der mapp Komponente, welcher das Ereignis über den MpLink zugeordnet ist
file	Name der Datei
dev	Name des Speichermediums

4.2 Getting Started

4.2.1 Hardwarekonfiguration zur Laufzeit verwalten

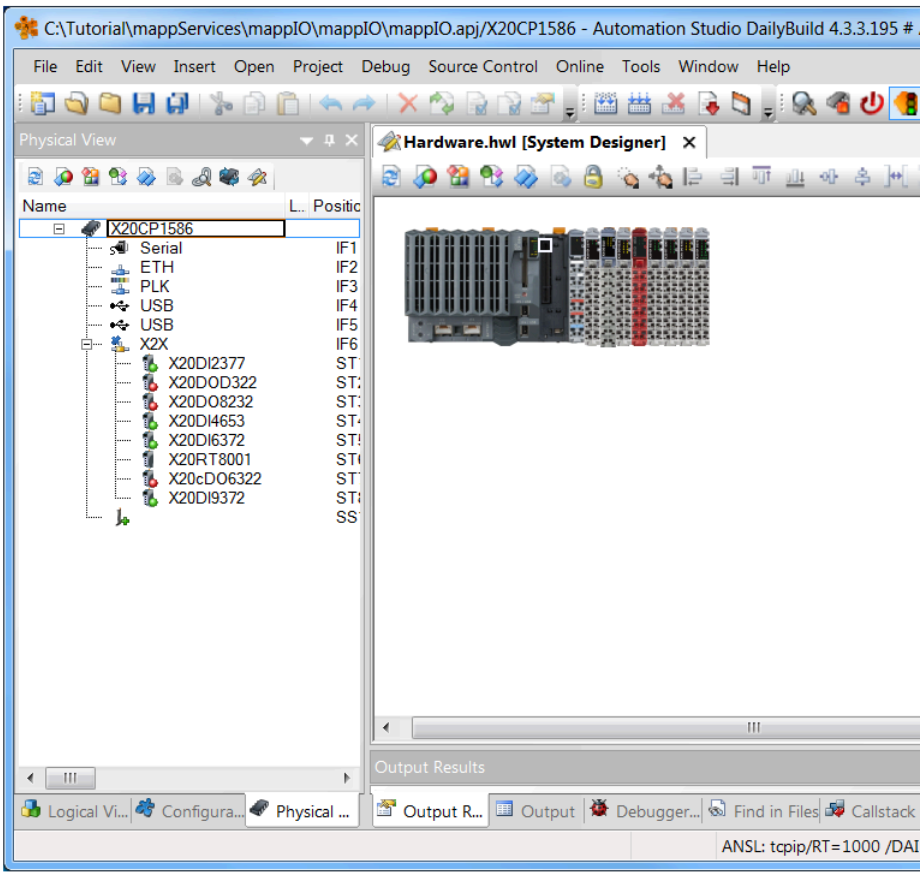
In diesem Abschnitt wird erklärt, wie man zur Laufzeit die Hardwarekonfiguration mithilfe von MpIOImport verändern kann.

4.2.1.1 Projekt erstellen

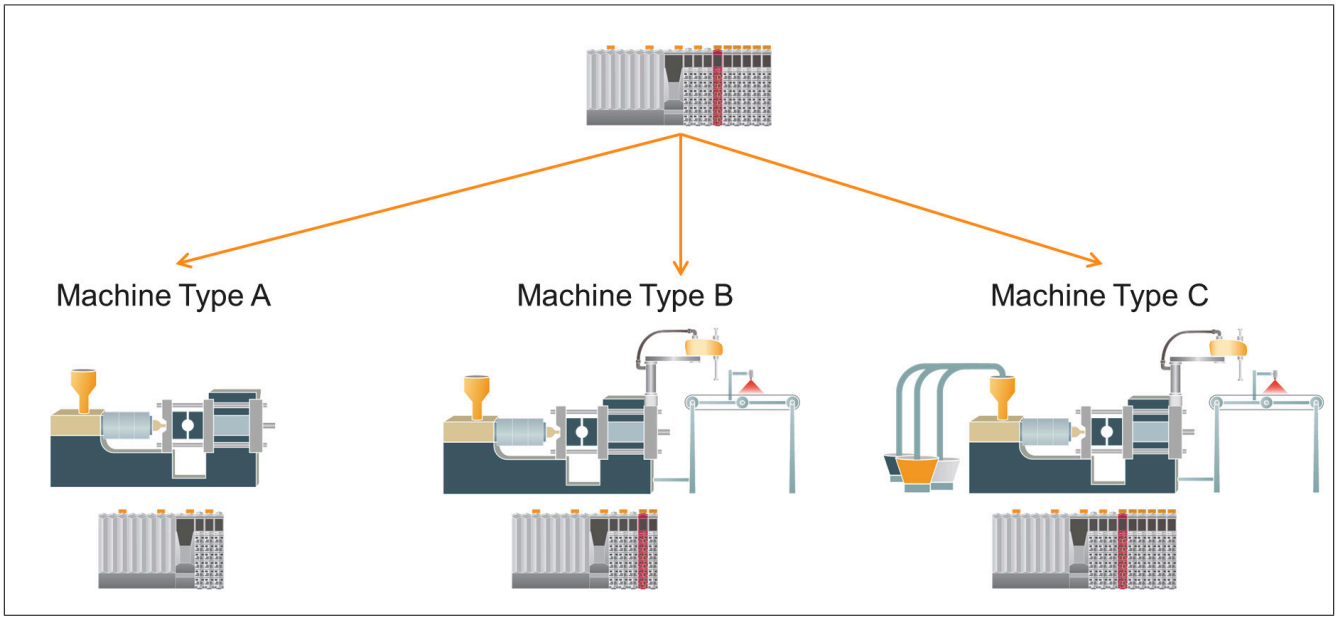
Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

4.2.1.2 Hardwarekonfiguration

Im Projekt wurde die Hardware mit allen möglichen verwendbaren Modulen eingefügt.

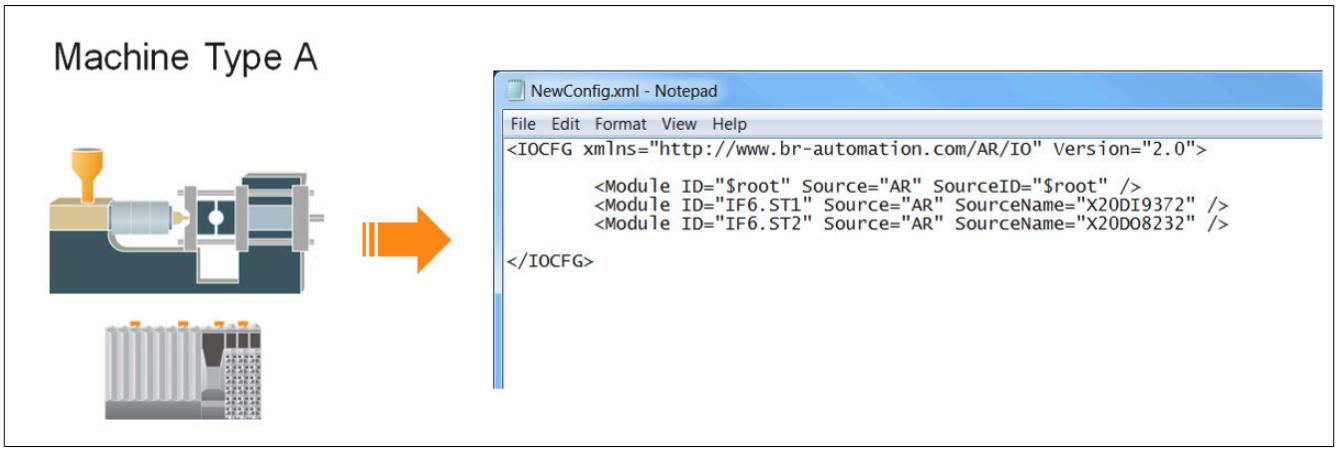


Dadurch können verschiedene Maschinentypen realisiert werden. Zur Laufzeit sollen die Module importiert werden, welche für den gewünschten Maschinentypen benötigt werden:



4.2.1.3 Neue Hardwarekonfiguration

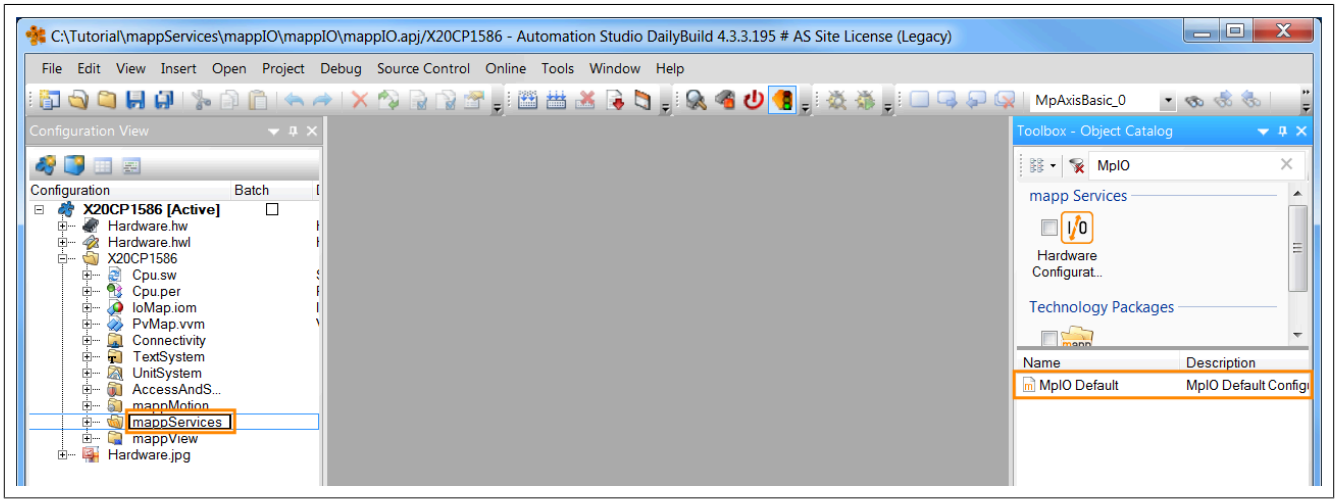
Als nächstes wird die zu importierende Hardwarekonfiguration definiert. Ziel ist es zur Laufzeit eine Hardwarekonfiguration für den Maschinentypen A zu erstellen. Das bedeutet zur Laufzeit muss ein Ein- und Ausgangsmodul importiert werden. Die Import-Datei wurde dafür bereits erstellt. Für mehr Information siehe Aufbau der Import-Datei. Über den Module-Knoten mit der ID "$root" wird die CPU aus der aktuellen Konfiguration übernommen. Anschließend soll am Steckplatz 1 der Steuerung das Modul X20DI9372 und am Steckplatz 2 das Modul X20DO8232 verwendet werden. "Source" definiert wo die Konfiguration des Moduls gefunden werden kann.



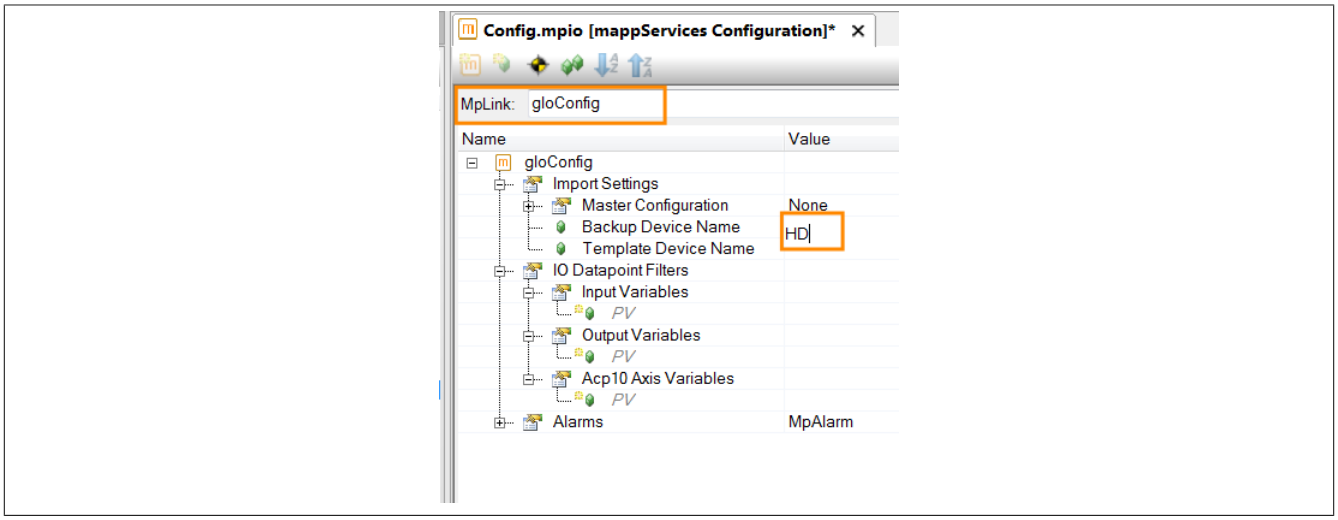
4.2.1.4 mapp Komponente einfügen

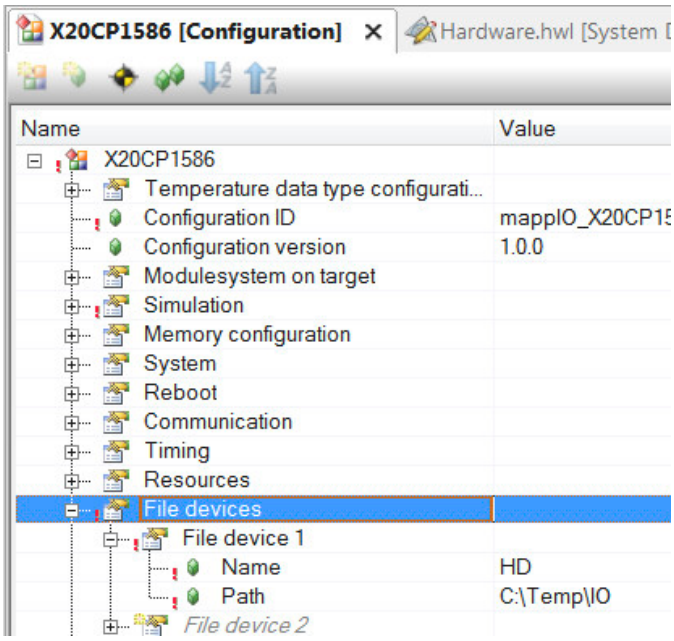
Konfiguration MpIO einfügen

Um die Hardwarekonfiguration zur Laufzeit zu verändern wird die Konfiguration MpIO eingefügt.



Hier befindet sich der mapp Link, welcher die mapp Komponente in der „world of mapp“ eindeutig identifiziert. Der mapp Link stellt die Verbindung zwischen Programmierung und Konfiguration her. Unter „Backup Device Name“ wird ein Speichermedium angegeben auf dem während eines Imports Backup-Dateien der Original-Konfiguration gespeichert werden. Das Speichermedium wurde zuvor in der Konfiguration der Hardware definiert.





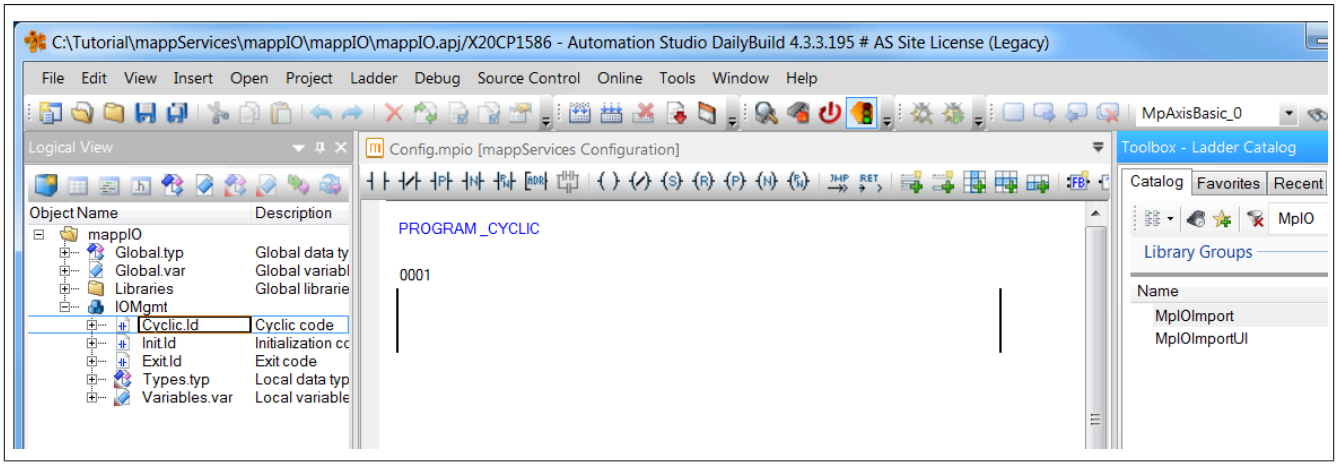
4.2.1.5 Programm hinzufügen

Der nächste Schritt ist das Einfügen eines Kontaktplan-Programms. Für mehr Information siehe hier.

4.2.1.6 MpIOImport im Programm einfügen

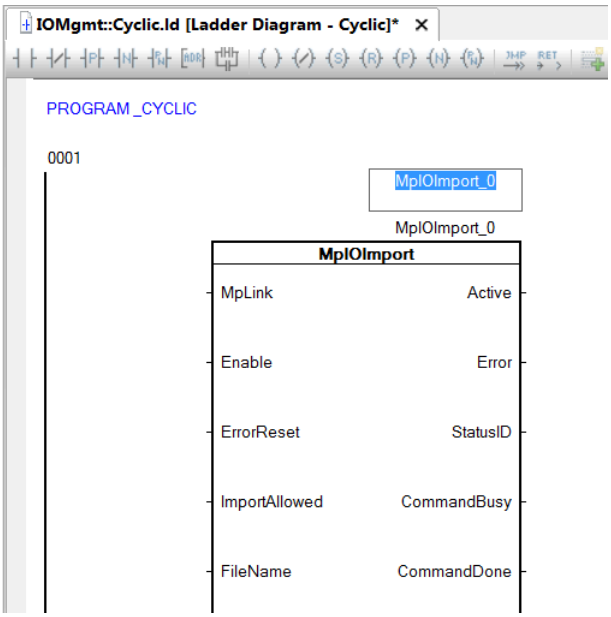
MpIO im Kontaktplan suchen

Im Katalog in Kontaktplan kann nach MpIO gesucht werden. Es werden alle MpIO Komponenten aufgelistet.



Funktionsbaustein auswählen

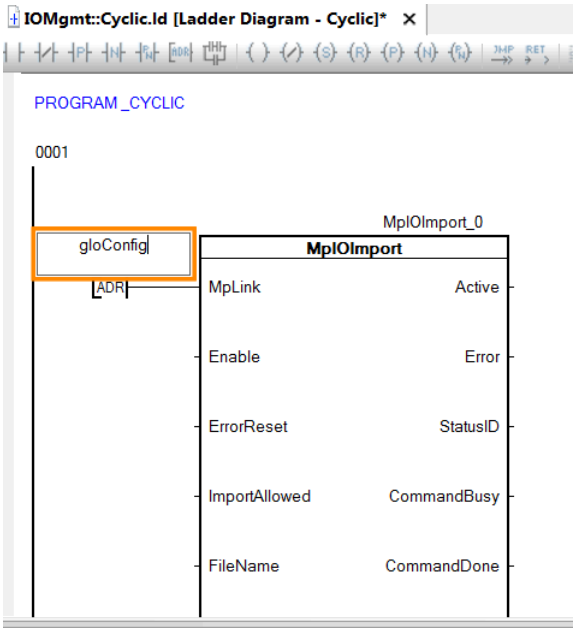
Anschließend wird der Funktionsbaustein MpIOImport eingefügt.



4.2.1.7 Parametrieren von MplOImport

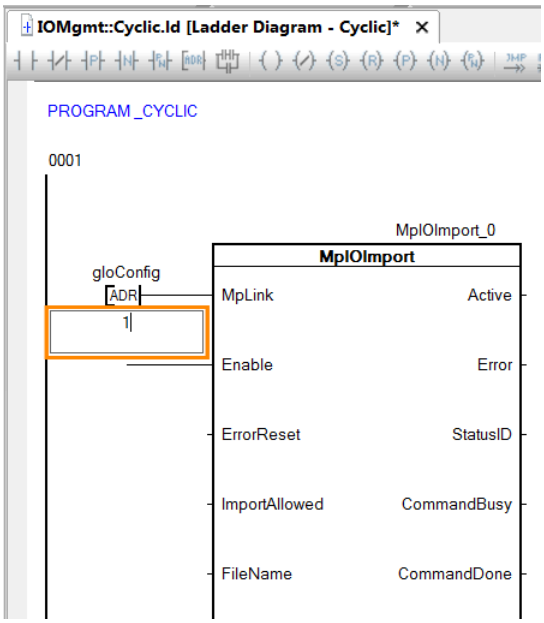
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gIOConfig erstellt haben.



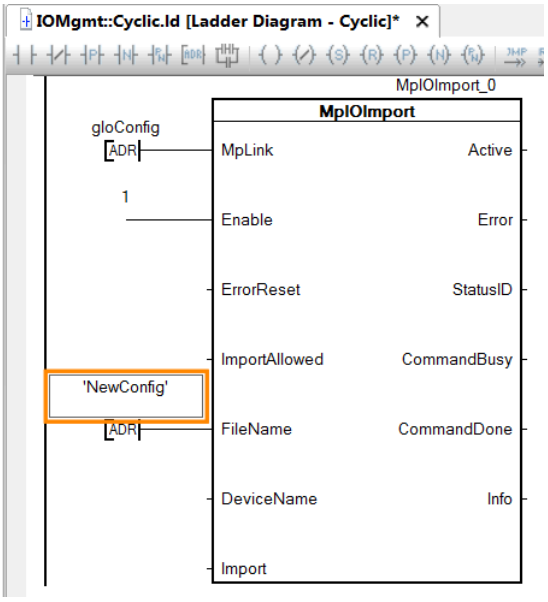
Aktivieren des Funktionsbausteins

Der Funktionsbaustein wird durch den Eingang „Enable“ aktiviert.



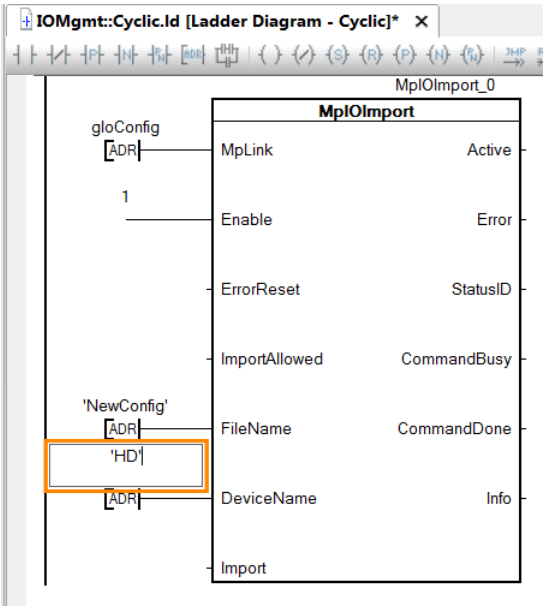
Zu importierende Hardwarekonfiguration definieren

Mithilfe von „FileName“ kann festgelegt werden wie der Name der zu importierenden Datei lautet.



Speichermedium festlegen

Über „DeviceName“ wird definiert auf welchem Speichermedium die Import-Datei liegt. Das Speichermedium wurde zuvor in der Konfiguration der Hardware erstellt.

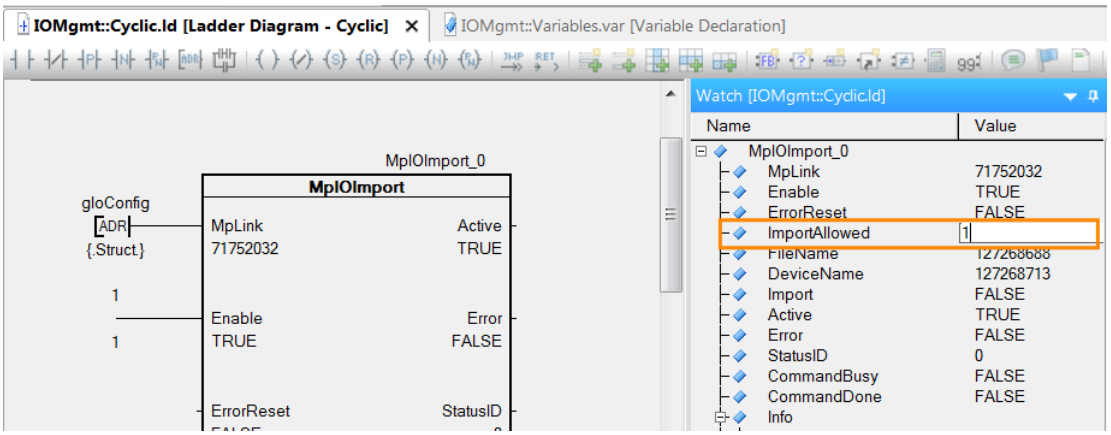


4.2.1.8 Dateistruktur erzeugen

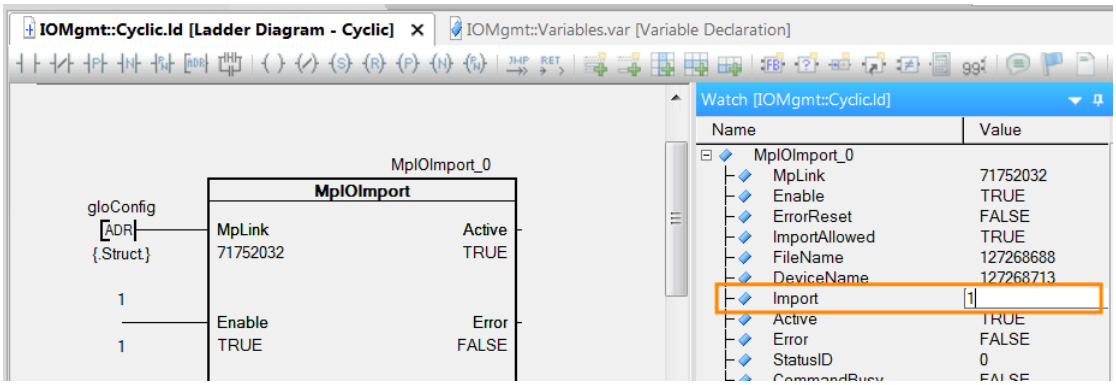
Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

4.2.1.9 Testen des Programms

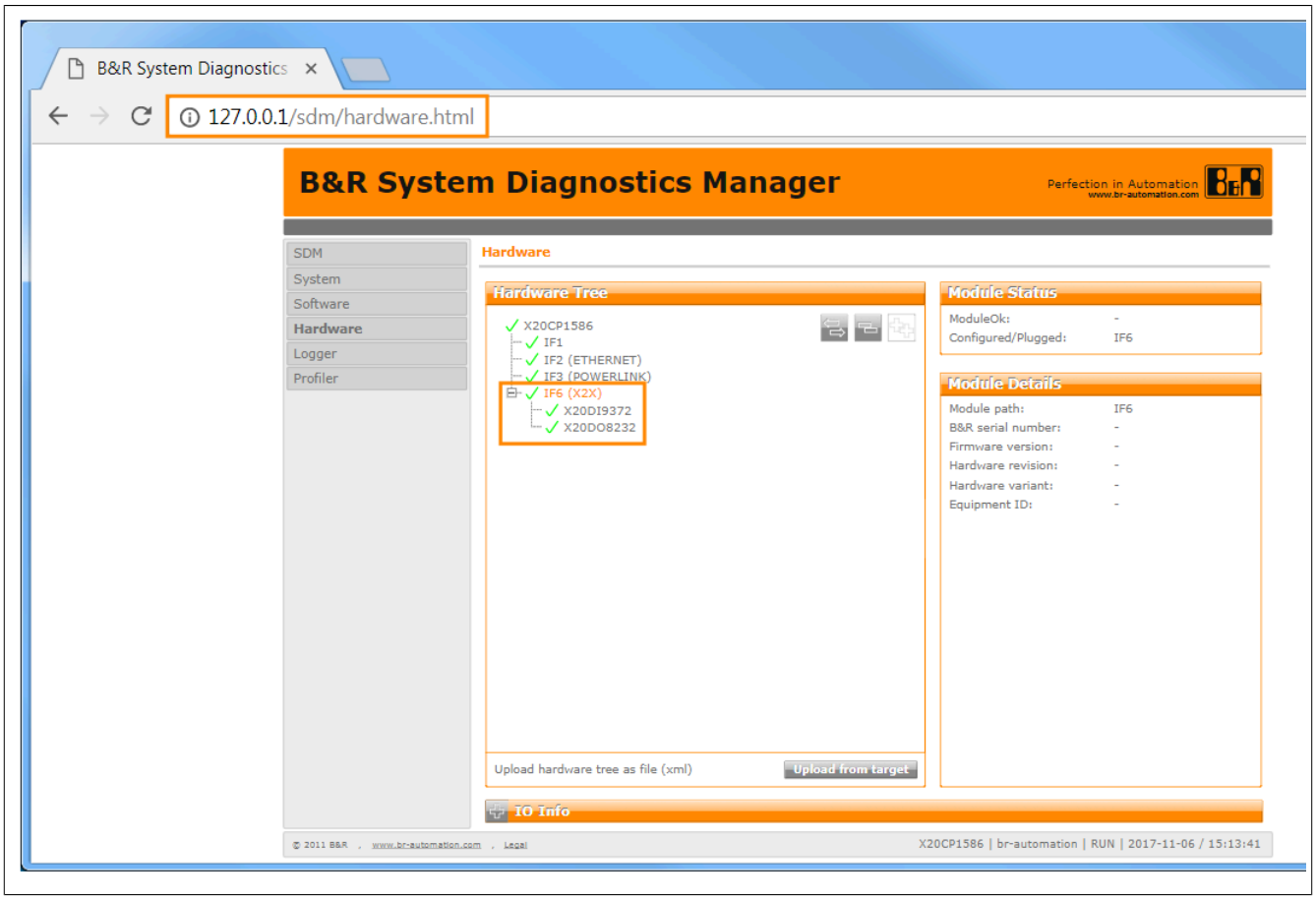
Nach einem Download der Änderungen kann das Programm getestet werden. Es wird der Monitor-Modus aktiviert. Der Funktionsbaustein MplOImport wurde in das Watch-Fenster eingefügt. Über „ImportAllowed“ muss das Importieren von Hardwarekonfigurationen erlaubt werden.



Mithilfe des Import-Befehls kann die Konfiguration „NewConfig“ importiert werden. Nach einem Neustart der Steuerung ist die Hardwarekonfiguration importiert.



Dies kann mithilfe des Systemdiagnose Manager überprüft werden. Es wird die vorher importierte Hardwarekonfiguration verwendet.



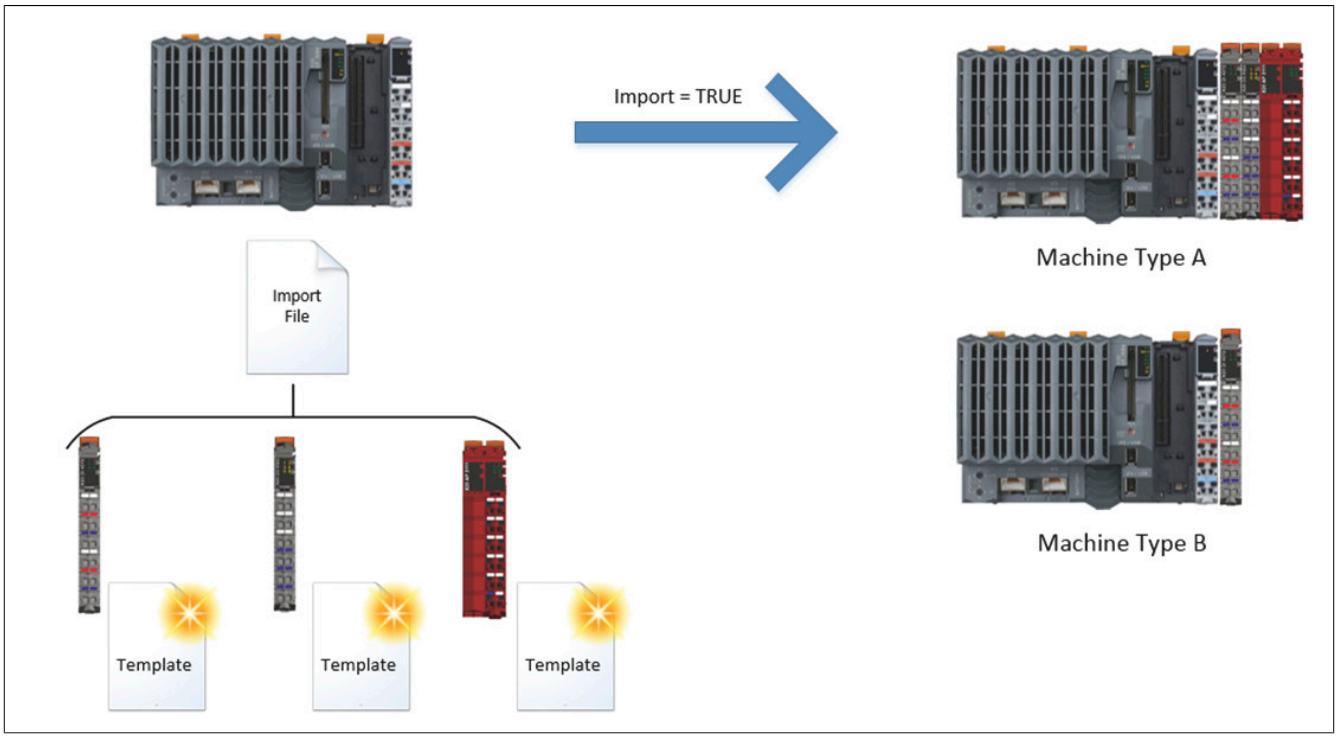
4.2.2 Hardware Konfiguration zur Laufzeit über Template-Dateien verwalten

In diesem Abschnitt wird erklärt, wie man mithilfe von MplOImport und Template-Dateien die Hardwarekonfiguration zur Laufzeit verwalten kann.

Im Abschnitt Hardwarekonfiguration zur Laufzeit verwalten wurde die Hardware-Konfiguration aus Automation Studio mit allen möglichen Modulen auf die Steuerung übertragen. Über die Import-Datei wurde dann entschieden, welche Module an welcher Stelle verwendet werden sollen.

Template-Dateien bieten eine weitere, flexiblere Variante um Hardware-Module zu referenzieren. Dabei befindet sich im Projekt nur das gewünschte Zielsystem, in unserem Fall eine X20 Steuerung, ohne weitere Module. Zur Laufzeit sollen nun verschiedene Module importiert werden.

Dies wird mithilfe von Template-Dateien realisiert. Dadurch kann die Hardwarekonfiguration zur Laufzeit beliebig geändert werden. Das bedeutet auch die Import-Datei kann beliebig geändert und angepasst werden. Dadurch können verschiedene Maschinentypen realisiert werden.



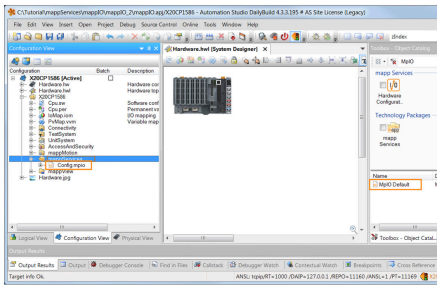
4.2.2.1 Projekt erstellen

Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

4.2.2.2 mapp Komponente einfügen

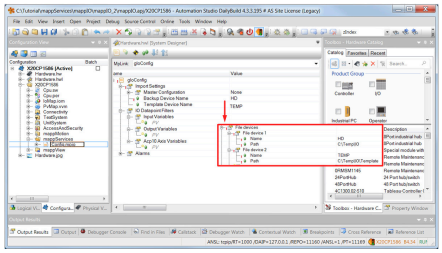
Konfiguration MpIO einfügen

Um die Hardwarekonfiguration zur Laufzeit zu verändern wird die MpIO Konfiguration eingefügt.



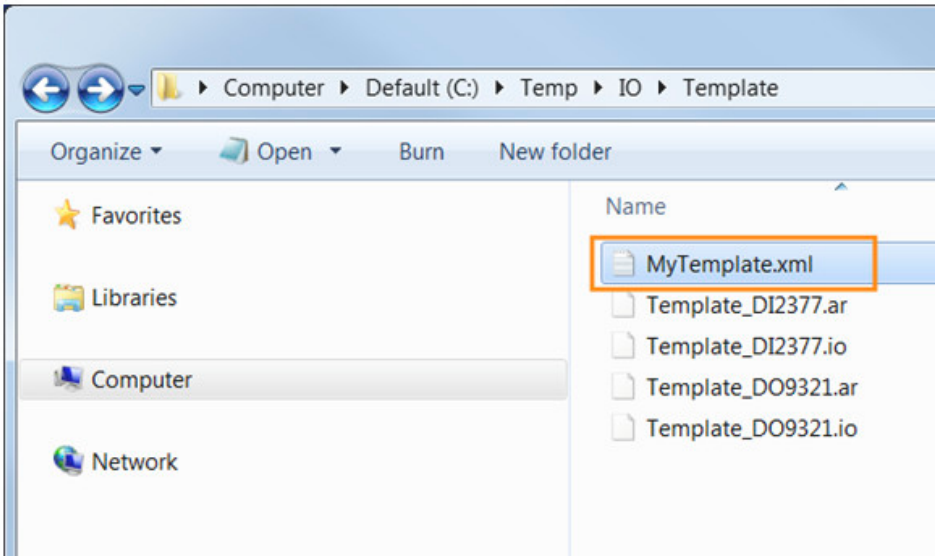
Hier befindet sich der mapp Link, welcher die mapp Komponente in der „world of mapp“ eindeutig identifiziert. Der mapp Link stellt die Verbindung zwischen Programmierung und Konfiguration her.

Unter „Backup Device Name“ wird ein Speichermedium angegeben auf dem während eines Imports Backup-Dateien der Original-Konfiguration gespeichert werden. Um Template Dateien importieren zu können, muss ein „Template“ Speichermedium definiert werden. Die beiden Speichermedien wurden bereits in der Konfiguration des Zielsystems angegeben.



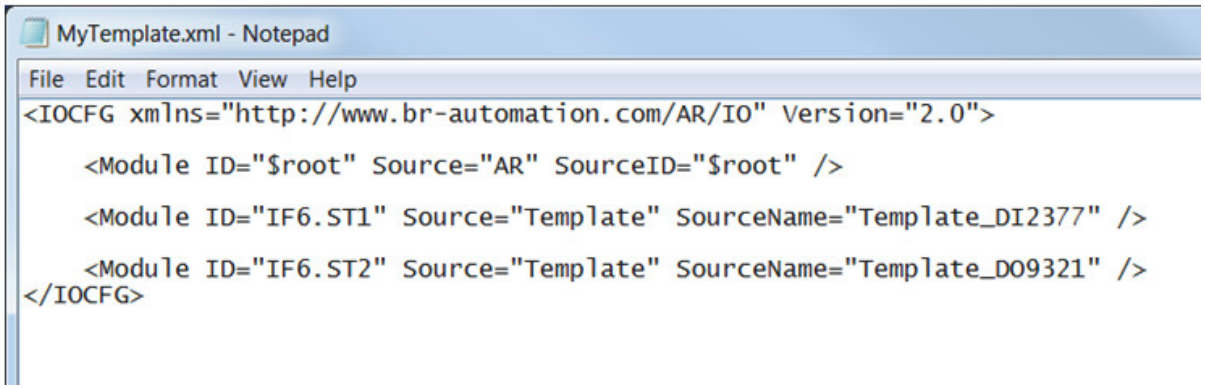
4.2.2.3 Import- und Template-Dateien vorbereiten

Als nächstes wird die zu importierende Hardwarekonfiguration definiert. Die Import-Datei und die Template-Dateien wurde dafür bereits erstellt:



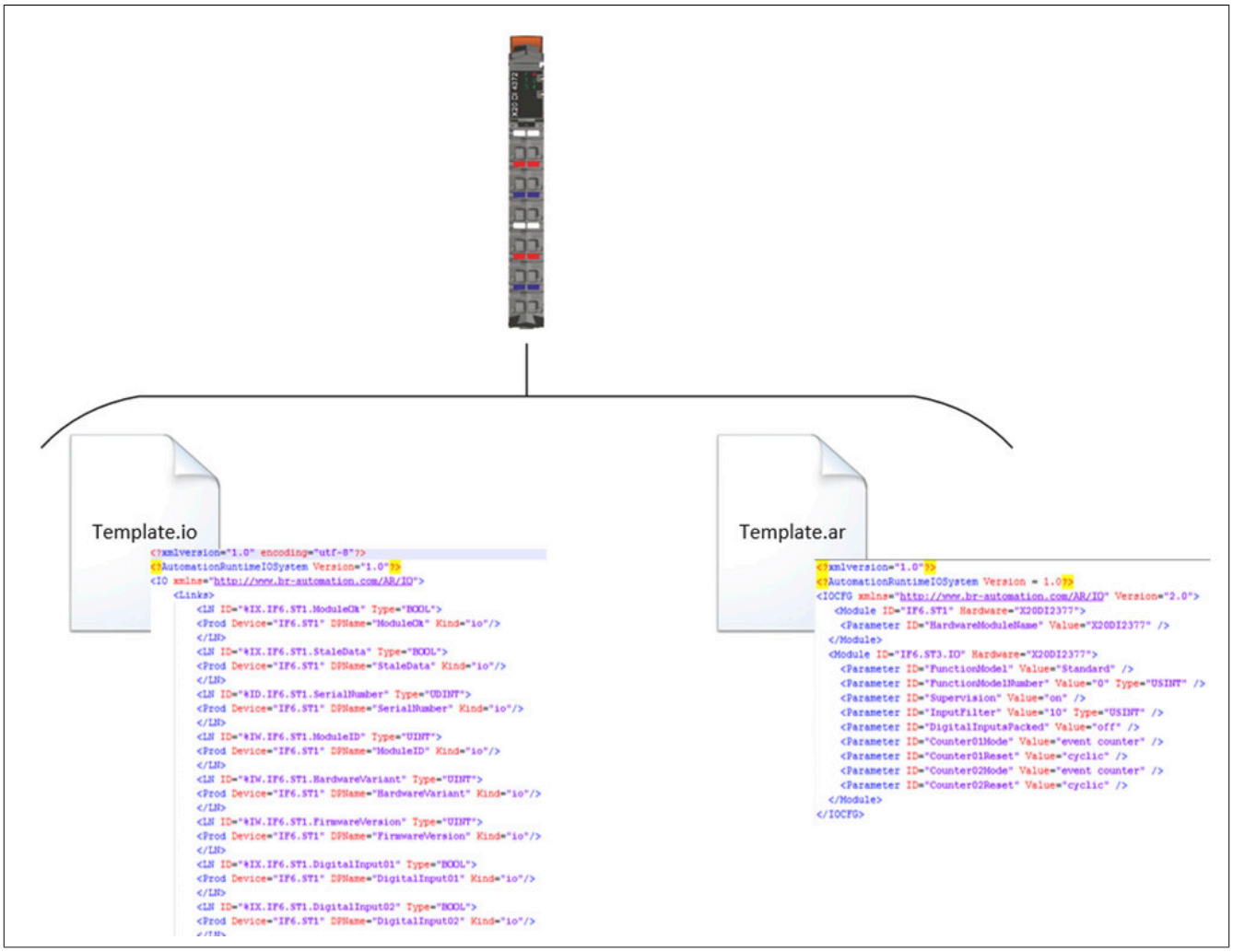
Import-Datei

Über den Module-Knoten mit der ID "$root" wird die CPU aus der aktuellen Konfiguration übernommen. Anschließend wird definiert welche Module an welchem Steckplatz importiert werden sollen. Über Sourcename wird der Name der Template-Datei angegeben.

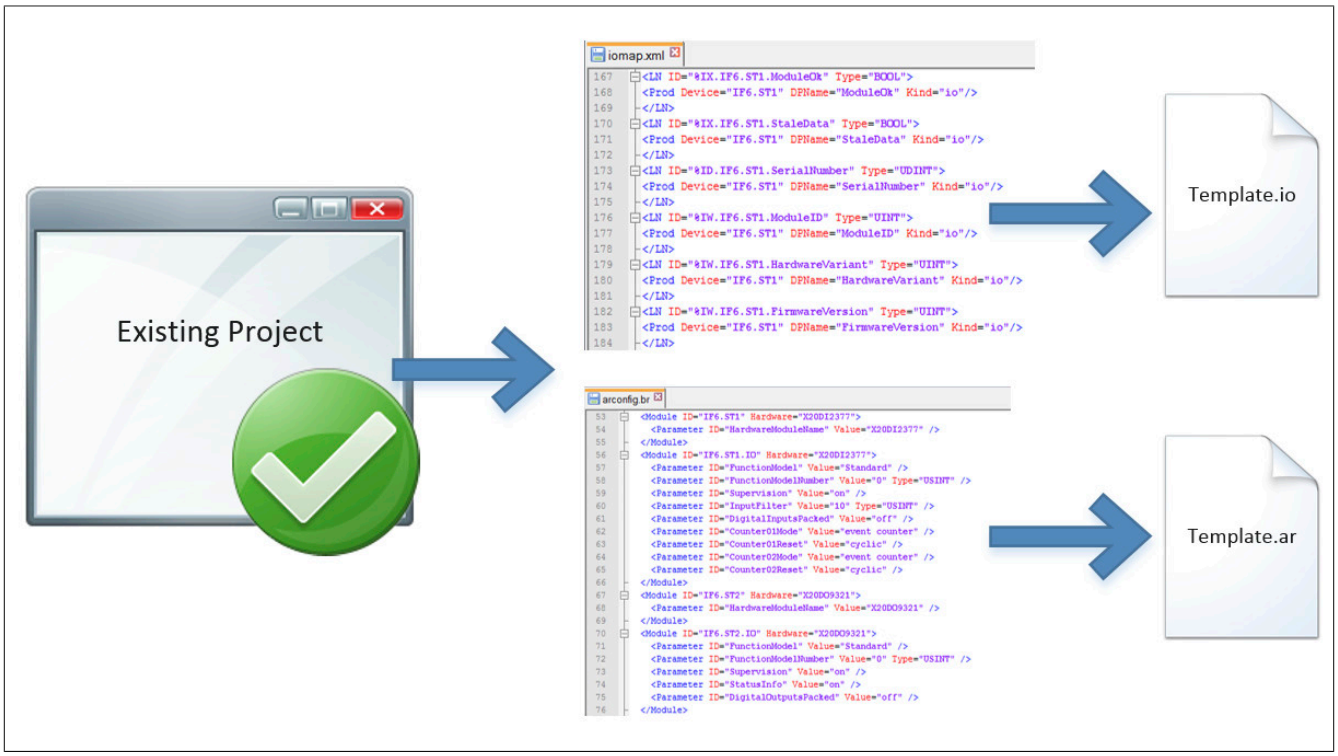


Template-Dateien

Pro Modul müssen zwei Template Dateien erstellt werden. Die .ar und die .io Datei. Wichtig hierbei ist, dass die Dateien den gleichen Namen besitzen, sie unterscheiden sich nur in der Endung. In der .ar Datei befinden sich die Eigenschaften eines Moduls. In der .io Datei befindet sich die IO Zuordnung eines Moduls.

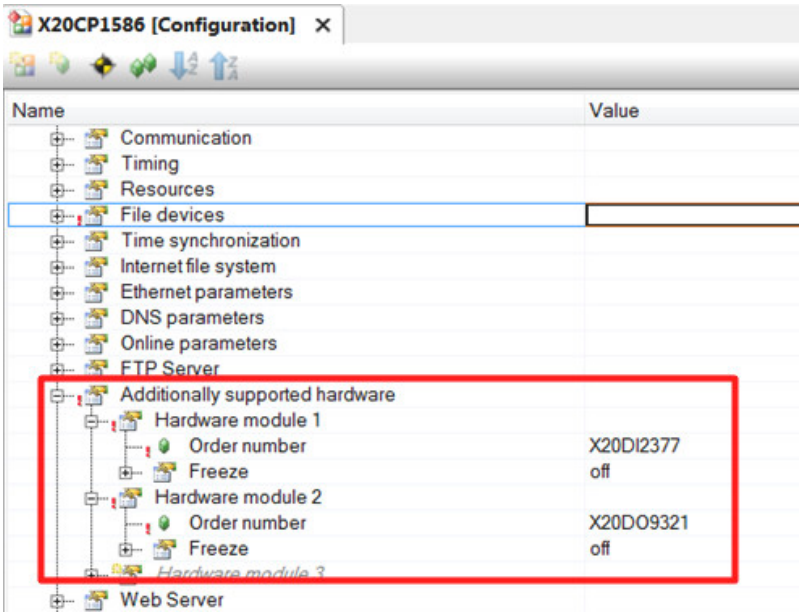


Um nun die Eigenschaften und die IO Zuordnung eines Moduls zu bekommen, muss in einer bestehenden Konfiguration die arconfig.br Datei bzw. die iomap.br Datei geöffnet werden. Die Information des jeweiligen Moduls wird anschließend in die .ar bzw. .io Datei kopiert. Das bedeutet die Information wird von einem existierenden Projekt verwendet, welches die gewünschten Module beinhaltet. Nur auf diese Art erhält man die .io und .ar Information. An welchen Steckplatz sich das Modul im existierenden Projekt befindet ist egal. Für mehr Information zu den Template-Dateien siehe hier.



Hier sieht man, dass ein Teil der .arconfig Datei für die Template Datei eines Moduls verwendet wird:

Um sicherzustellen, dass auch die Firmware für Module, welche über Template Dateien referenziert werden auf der Steuerung ist, wird die Funktionalität „Additionally supported hardware“ von Automation Studio verwendet. Diese Option wird in der Konfiguration des Zielsystems definiert.



4.2.2.4 Programm hinzufügen

Der nächste Schritt ist das Einfügen eines Kontaktplan-Programms. Für mehr Information siehe hier.

4.2.2.5 Funktionsbaustein einfügen und parametrieren

Als nächstes wird der Funktionsbaustein MpIOImport eingefügt und parametriert. Wie der Funktionsbaustein eingefügt wird ist hier erklärt. Wie der Funktionsbaustein parametriert werden muss ist im Abschnitt Parametrieren von MpIOImport erklärt.

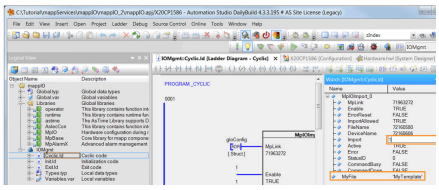
4.2.2.6 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

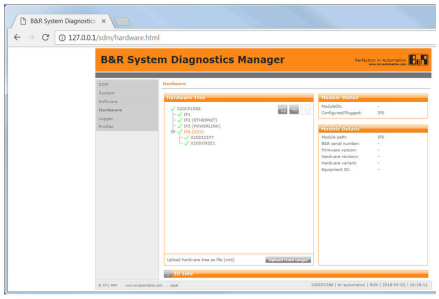
4.2.2.7 Testen des Programms

Nach einem Download der Änderungen kann das Programm getestet werden. Der Monitor-Modus wird aktiviert.

Der Funktionsbaustein MpIOImport wurde in das Watch-Fenster eingefügt. Über „ImportAllowed“ muss das Importieren von Hardwarekonfigurationen erlaubt werden. Mithilfe des Import-Befehls kann die Konfiguration „MyTemplate“ importiert werden.



Nach einem Neustart der Steuerung ist die Hardwarekonfiguration importiert. Dies kann mithilfe des Systemdiagnose Manager überprüft werden. Es wird die vorher importierte Hardwarekonfiguration verwendet.

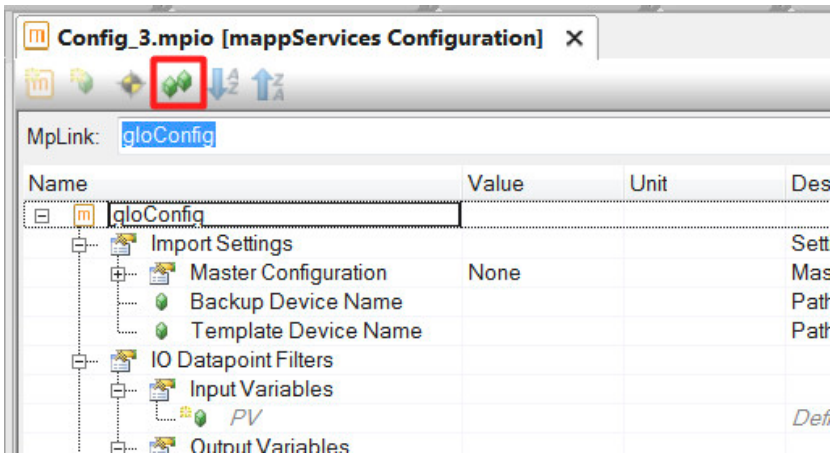


4.3 Konfiguration

4.3.1 MpIO Konfiguration

> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**
>
> **Die geänderten Konfigurationsparameter werden sofort von der Komponente übernommen.**

Um die erweiterten Parameter in der Konfiguration anzeigen zu lassen, muss "Ändere Erweiterte Parameter Sichtbarkeit" aktiviert werden:



Master Configuration

Der Parameter "Master Configuration" bestimmt von welcher Basis ausgegangen wird beim Konfigurieren der Hardware Konfiguration:

- **None:** Wird "None" gewählt definiert die Import-Datei die gesamte Hardware-Konfiguration, angefangen bei der Steuerung
- **Automation Runtime:** In diesem Fall dient die, auf der Steuerung vorhandene, Hardware-Konfiguration als Basis. Über den Parameter "Insert Position" wird definiert an welchem Interface die Module in der Import-Datei eingefügt werden sollen (z.B. "IF6" für das X2X Interface einer X20CP1586).
  Weiters können bestimmte Bereiche der Master-Konfiguration ignoriert werden über den Parameter "Exclude". Die hier definierten Interfaces werden beim Import aus der Hardware-Konfiguration gestrichen.

Preserve PV Mapping

Über diesen Punkt kann sichergestellt werden, dass die Variablen-Zuordnungen (.vvm) beim Importieren unberührt bleiben. Bei `FALSE` kann ein neues PV-Mapping importiert werden.

Remove Unresolved Links

Betrifft nur Safety Module. Wenn ein Safety Modul während einem Import über MpIOImport entfernt wird, werden die Links innerhalb der SafeLogic nicht automatisch aktualisiert, was zu Fehlern führen kann. Mithilfe der "Remove Unresolved Links" Eigenschaft kann definiert werden, ob der Import nicht verwendete oder ungültige Links entfernen bei der SafeLogic suchen und entfernen soll.

Weitere Informationen kann im Abschnitt Umgang mit spezieller Hardware gefunden werden.

Backup Device Name

Mithilfe des "Backup Device Name" Parameters wird angegeben, auf welchem File Device während einem Import Backup-Dateien der Original-Konfiguration gespeichert werden. Das File Device sollte auf der CompactFlash oder Onboard-Flash erstellt werden.

Template Device Name

Mithilfe des "Template Device Name" Parameters wird angegeben, auf welchem File Device während einem Import nach Template Dateien gesucht wird.

Plausibility Checks

Die Plausibilitäts-Checks prüfen, ob die in der Import-Datei angegebenen Prozessvariablen für das I/O Mapping gültig sind, beispielsweise kann geprüft werden ob der Datentyp einer Prozessvariable zum Datentyp auf einem Hardware-Modul passt. Sind alle Plausibilitäts-Checks deaktiviert, wird die Sinnhaftigkeit der Zuordnungen nicht geprüft.

- **Allow Narrowing:** Definiert, ob eine Zuordnung eines kleinen Datentyps (z.B. INT) auf einen größeren Datentypen (z.B. DINT) zugelassen wird. Hat keine Auswirkung falls "Check Types = FALSE" ist
- **Check Types:** Definiert, ob bei einem Import die Datentypen bei einem Mapping übereinstimmen müssen.
- **Check PVs:** Definiert, ob bei einem Import geprüft werden soll, ob die angegebenen Variablen existieren.
- **Check Axis PVs:** Definiert, ob bei einem Import geprüft werden soll, ob die angegebenen Achsreferenzen existieren.

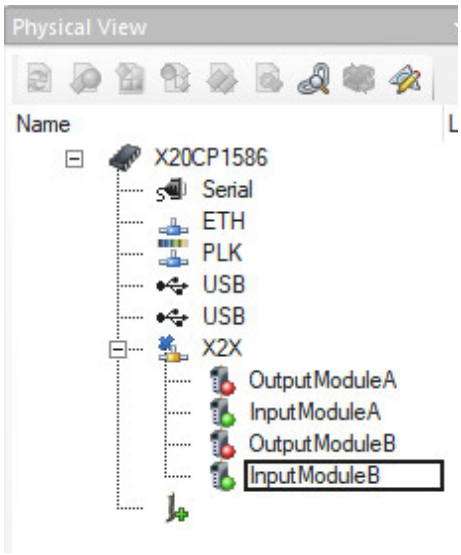
Replace

Mithilfe von "Replace" kann z.B. bei den Namen von Hardware-Modulen ein bestimmter Präfix entfernt werden. Hierbei wird angegeben, in welcher Konfiguration die Aktion ausgeführt werden soll.

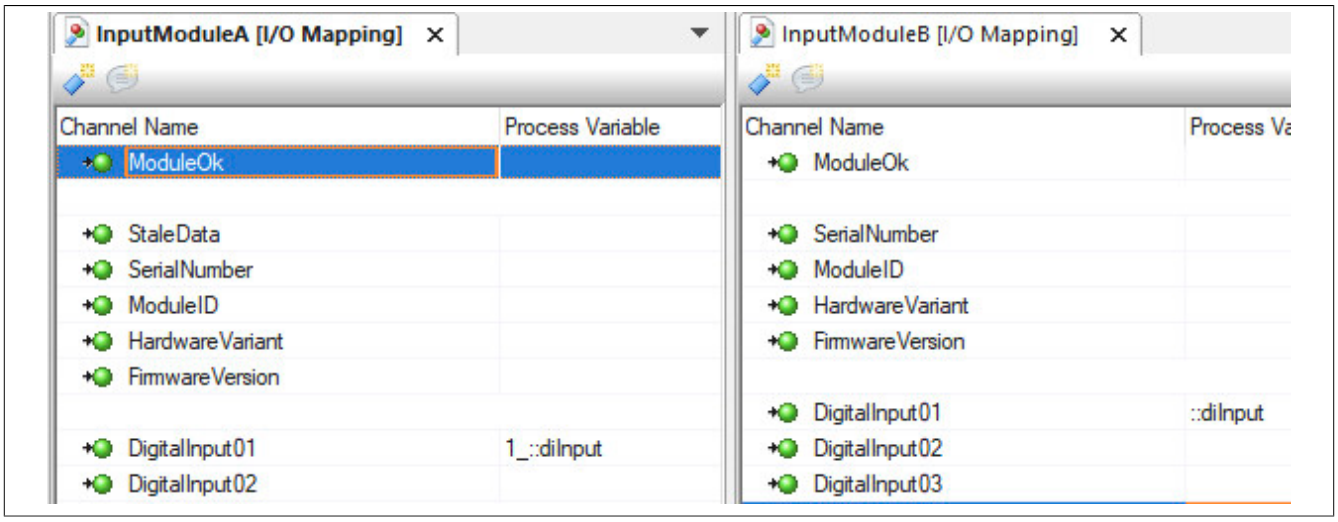
Die Aktion definiert, was in dieser Datei geändert werden soll. "Pattern" definiert z.B. den Präfix der entfernt werden soll.

Beispiel:

Es wird die Hardwarekonfiguration für verschiedene Maschinentypen in AS projektiert. Dabei soll zwischen den Maschinentypen A und Maschinentypen B gewählt werden können. Abhängig davon welcher Maschinentyp gewählt wird, soll dementsprechend nur die benötigten Ein-, und Ausgangsmodule importiert werden.

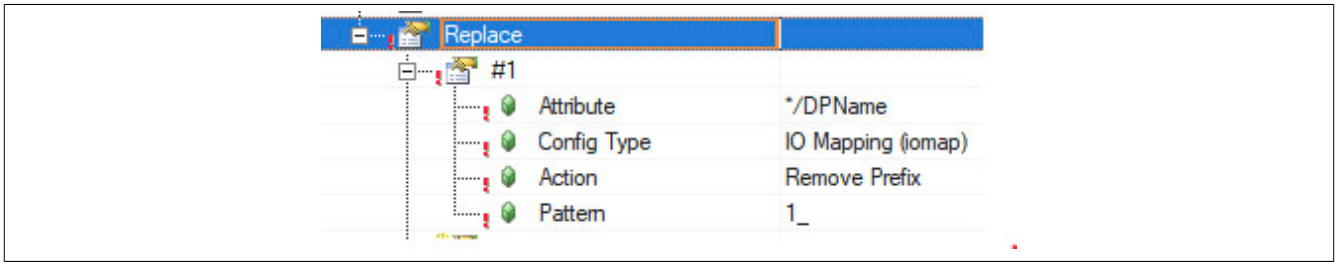


Beide Eingangsmodule sollen dabei die gleiche Prozess Variable verwenden. Wird die gleiche PV in beiden Modulen gemappt, erscheint während des Kompilier-Vorgangs eine Fehlermeldung welche angibt, dass eine PV nur einmal an einem Eingangsmodul gemappt werden darf. Um dies zu umgehen, wird in einem Modul der Präfix "1_" angegeben:



Dadurch kann die gleiche PV mehrmals gemappt werden.

Damit während des Import der gewünschten Hardwarekonfiguration (Maschinentyp A oder Maschinentyp B) die korrekte PV verwendet wird, muss der Präfix in der Konfiguration gelöscht werden. Dazu kann die Funktion "Replace" verwendet werden.



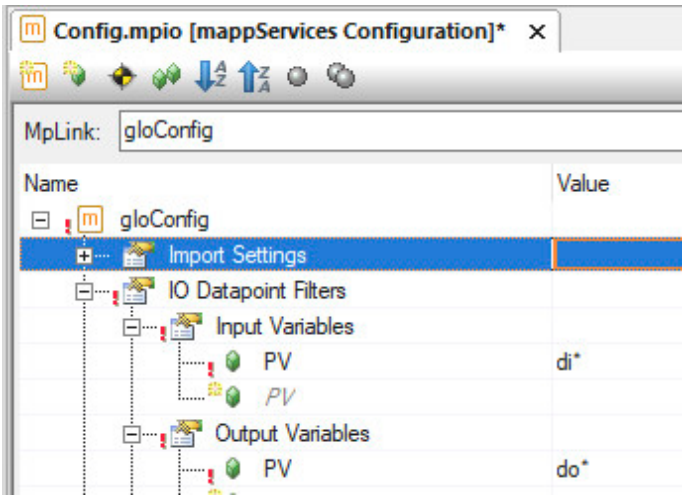
Der Präfix wird dabei automatisch gelöscht.

Dadurch kann in der Import-Datei die Prozess-Variable normal angegeben werden. Es muss auf keinen Präfix geachtet werden. Die Import-Datei kann beispielsweise folgendermaßen aussehen:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />

    <Module ID="IF6.ST1" Source="AR" SourceName="InputModuleA" />
    <Module ID="IF6.ST1" Source="AR" SourceName="OutputModuleA" />
</IOCFG>
```

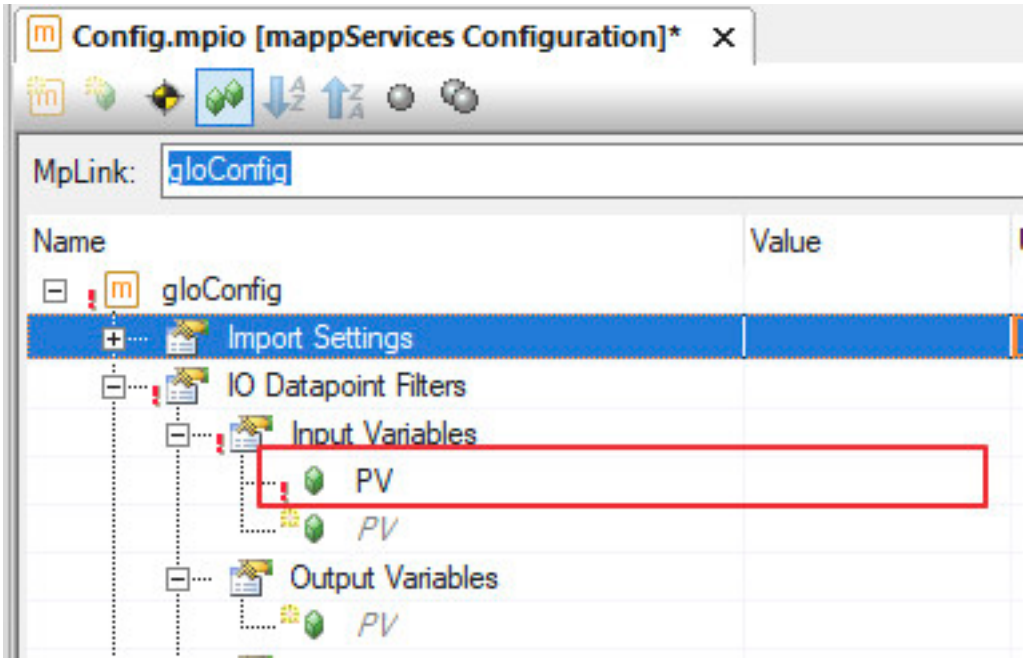
IO Datapoint Filters

Über den Punkt "IO Datapoint Filters" kann definiert werden, welche Variablen für eine I/O Zuordnung erlaubt sind. Wird beispielsweise definiert, dass für "Input Varibales" nur Variablen mit dem Präfix *'di'* oder *'ai'* erlaubt sind, prüft MpIO beim Import ob die Import-Datei dieser Regel folgt. Die Angabe erfolgt dabei mit dem Asteriks (*), was beispielsweise folgendermaßen aussehen kann:



Bei einer eingängigen Variable mit unterschiedlichem Präfix wird ein Fehler ausgegeben.

Wird bei der PV kein Wert angegeben, so kann auch kein IO-Mapping durchgeführt werden, da nach einer Prozessvariable mit keinem Namen gesucht wird.



4.4 Anwendungsfälle

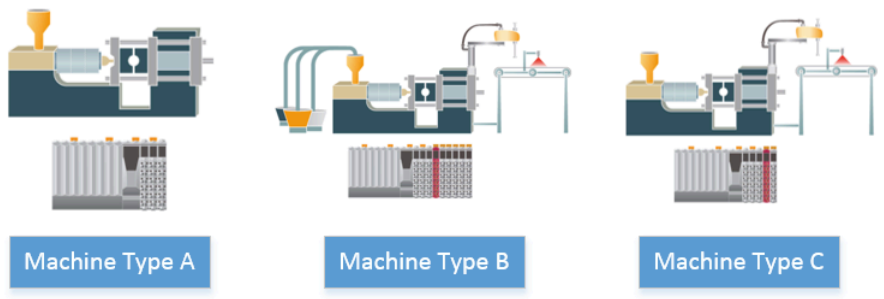
4.4.1 Auswählen der Hardware-Konfiguration zur Laufzeit

Anforderung

Ein Maschinenhersteller verkauft eine Maschine mit 3 unterschiedlichen Ausbaustufen, Maschinentyp A, B und C. Beim Kauf entscheidet der Kunde welche Variante er braucht. Es soll aber jederzeit möglich sein nachzurüsten und direkt über die Visualisierung die Ausbaustufe zu wechseln. Die Applikation passt sich an die jeweilige Ausbaustufe an.

Es soll möglich sein während der Laufzeit zu entscheiden, welcher Maschinentyp aktuell verwendet werden soll. Das heißt je nachdem welche Maschine gekauft wurde, kann die entsprechende Hardwarekonfiguration eingestellt werden.

Der Maschinenbauer möchte nur ein Projekt mit einem Hardwaresetup warten.



Lösung

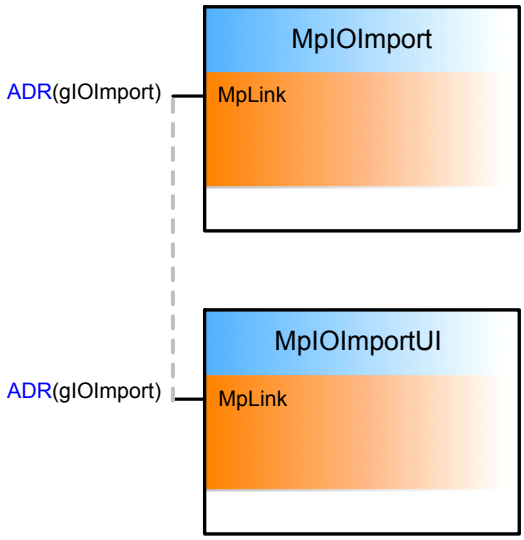
Grundidee

- Ein Projekt mit einer Konfiguration, welche die gesamte Hardware für alle Maschinentypen enthält.
- Importieren der jeweiligen Ausbaustufe mithilfe einer XML-Datei.
- Anpassen der Applikation mithilfe der Bibliothek SYS_Lib.

Komponentenliste

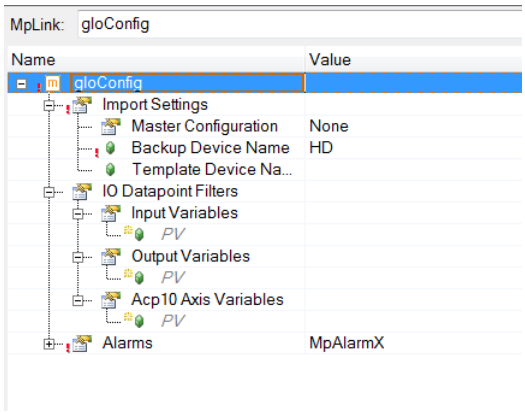
- ***MpIOImport*** (Eigener MpLink): Verwaltet den Import der neuen Hardware-Konfiguration.
- ***MpIOImportUI*** (MpLink von MpIOImport): Beinhaltet alle nötigen Parameter und Strukturen um eine Verbindung zu einer Visualisierung herzustellen und eine Hardwarekonfiguration zu importieren.

Verbindungsschema



Konfiguration

In der Konfiguration wird ein "Backup Device Name" angegeben. In diesem Speichermedium, welches in der Konfiguration des Zielsystem festgelegt wird, werden die Originaldateien der Konfiguration abgespeichert. Für mehr Information zur Konfiguration siehe MpIO Konfiguration.



Verwendung der mapp Komponenten

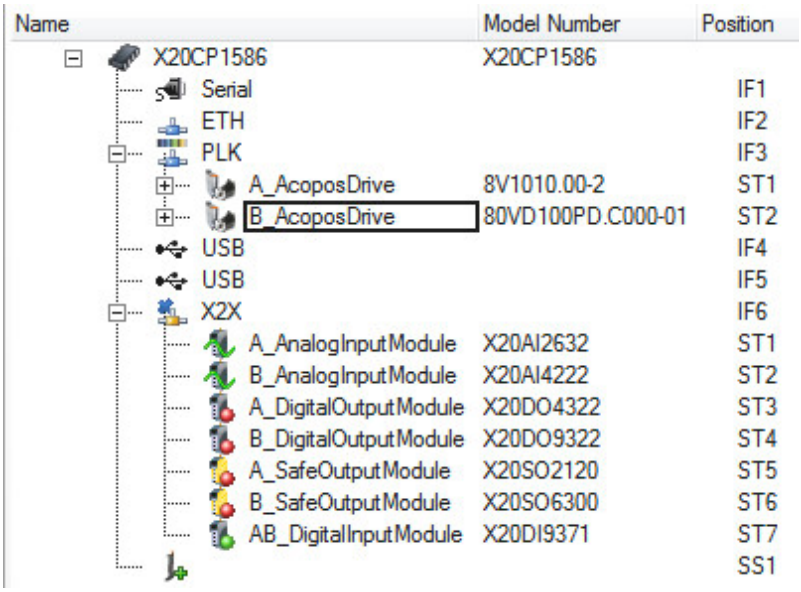
Es wird der Funktionsbaustein **MpIOImport** zum Verwalten der Hardware-Konfiguration eingefügt.

Anschließend wird **MpIOImport** parametriert. Dafür benötigt werden "FileDevice", "FileName" sowie "ImportAllowed" und "Import".

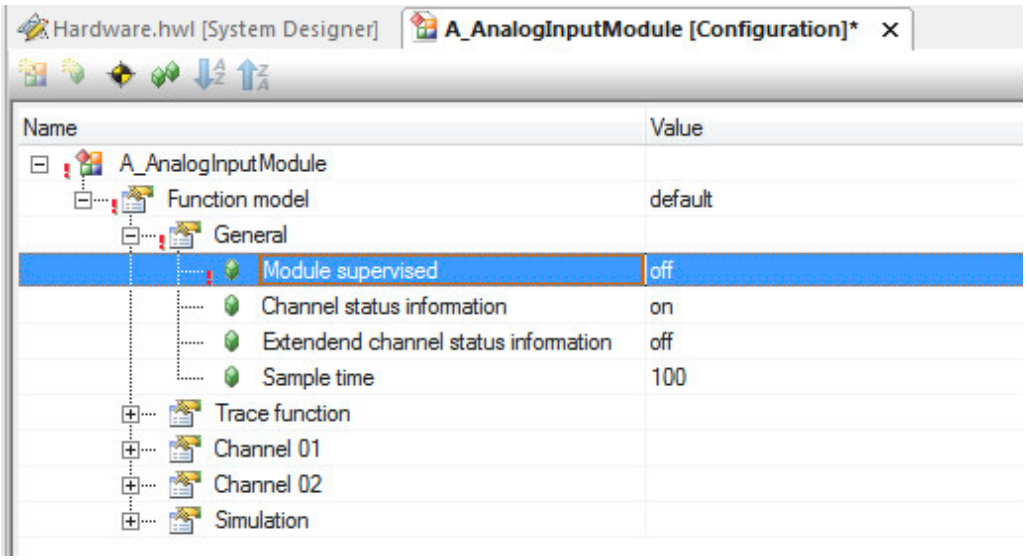
Um die Hardwarekonfiguration zur Laufzeit über die Visualisierung zu ändern wird der Funktionsbaustein **MpIOImportUI** eingefügt und parametriert.

Je nachdme welcher Maschinentyp verwendet wird, wird die Applikation mithilfe der SYS_Lib Bibliothek angepasst.

Um die jeweilige Hardware-Konfiguration mit mapp IO importieren zu können, werden im Automation Studio alle Module eingefügt die für die unterschiedlichen Maschinentypen benötigt werden. Für mehr Information siehe hier.



Hierbei ist zu beachten, dass "ModuleSupervised" auf "off" zu stellen ist! Ansonsten startet die Steuerung im Service-Modus wenn die Hardware-Module nicht am angegebenen Steckplatz vorhanden sind.



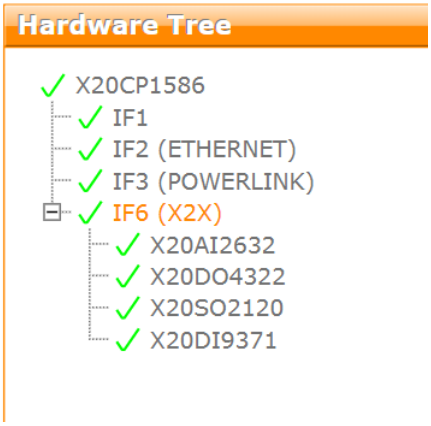
Die Import-Datei für den Maschinentyp A könnte folgendermaßen aussehen:

```
<IOCFG xmlns="http://www.br-automation.com/AR/IO" Version="2.0">

    <Module ID="$root" Source="AR" SourceID="$root" />

    <Module ID="IF6.ST1" Source="AR" SourceName="A_AnalogInputModule" >
        <Parameter ID="Supervision" Value="on" />
        <LN ID="AnalogInput01" DPName="Program:aiInput1" />
        <LN ID="AnalogInput02" DPName="Program:aiInput2" />
    </Module>

    <Module ID="IF6.ST2" Source="AR" SourceName="A_DigitalOutputModule" >
        <Parameter ID="Supervision" Value="on" />
        <LN ID="DigitalOutput01" DPName="Program:doOutput1" />
        <LN ID="DigitalOutput02" DPName="Program:doOutput2" />
    </Module>

    <Module ID="IF6.ST3" Source="AR" SourceName="A_SafeOutputModule" >
        <Parameter ID="Supervision" Value="on" />
    </Module>

    <Module ID="IF6.ST4" Source="AR" SourceName="AB_DigitalInputModule" >
        <Parameter ID="Supervision" Value="on" />
    </Module>

</IOCFG>
```

Für mehr Information zum Aufbau einer Import-Datei siehe hier.

Diese Importdatei wird auf die Steuerung in das Speichermedium "HD" übertragen:

Mithilfe von MpIOImport wird diese Datei importiert. Sobald der Import abgeschlossen ist, müssen alle Module am angegebenen Steckplatz vorhanden sein, da mithilfe der Importdatei "ModuleSupervised" wieder aktiviert wird. Die finale Hardware-Konfiguration für Maschinentyp A könnte zum Beispiel folgendermaßen aussehen:



4.5 Diagnose

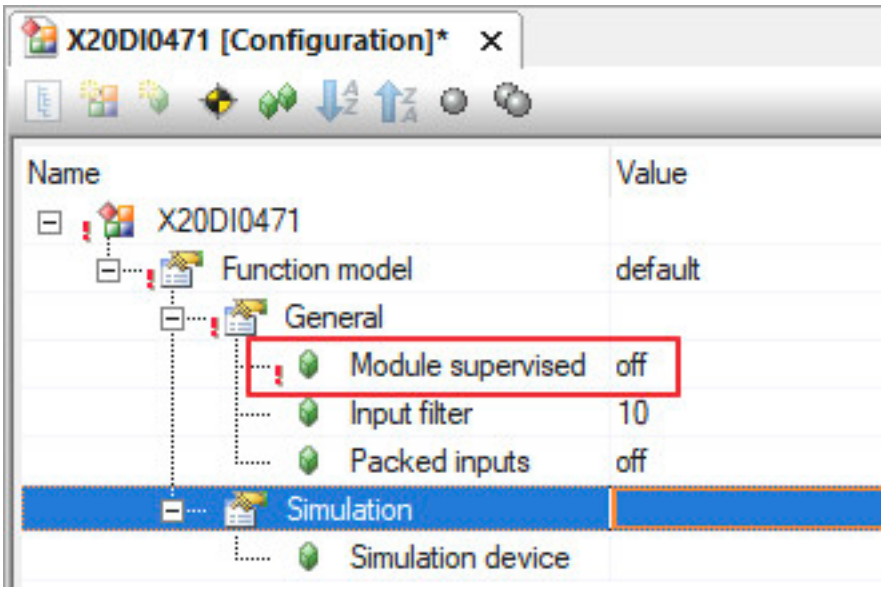
Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

4.5.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp IO auftreten können.

Meine Steuerung startet im Service-Modus

Beim Verändern der Hardwarekonfiguration muss darauf geachtet werden, dass die Modulüberwachung bei den verwendeten Modulen ausgeschaltet ist. Standardmäßig ist die Überwachung aktiviert. Fehlt das Modul, wird ein Service Modus ausgelöst.



Meine Motion-Elemente funktionieren nicht wie angenommen

Bei der Verwendung von Antriebstechnik in Kombination mit mapp IO müssen folgende Punkte beachtet werden:

- mapp IO kann nur in Kombination mit ACP10/ARNC0 verwendet werden! Aktuell wird mapp Motion nicht unterstützt.
- CNC-Kanäle können nicht konfiguriert werden.
- SDC Achsen können nicht konfiguriert werden.
- ACOPOS P3 kann im Projekt verwendet werden. Beim importieren der Hardware darf sich aber die Position des ACOPOS P3 nicht ändern.
- Bei der Verwendung von Template Dateien können Generische Interface Karten nicht verwendet werden.

4.6 Bibliotheken

4.6.1 Datentypen und Enumeratoren

4.6.1.1 Datentypen

4.6.1.1.1 MpIODiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	MpIOStatusIDType	StatusID Diagnose Struktur.

4.6.1.1.2 MpIOImportInfoType

Mit diesem Datentyp wird für den Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
ImportFileActive	BOOL	Die Konfiguration wurde importiert. Die Hardware-Konfiguration wurde verändert.
Diag	MpIODiagType	Eine Diagnosestruktur für die StatusID.

4.6.1.1.3 MpIOImportUIConnectType

Mithilfe dieses Datentyps wird eine Verbindung zwischen der MpIOImportUI Komponente und der Visual Components 4 Visualisierung erstellt.

Parameter	Datentyp	Beschreibung
DefaultLayerStatus	UINT	Statusdatenpunkt für den Default Layer der Visualisierungsseite, auf der die Liste der möglichen Importdateien angezeigt wird **VC4 Anbindung:** StatusDatapoint von *Layer*
Status	MpIOImportUIStatusEnum	Aktuelle Operation
FileList	MpIOUIFileListType	Beinhaltet eine Liste aller für den Import verfügbaren Dateien sowie eine Navigation durch die Liste
MessageBox	MpIOUIMessageBoxType	Bedienung der Dialogfenster
Import	BOOL	Befehl zum Starten eines Imports

4.6.1.1.4 MpIOImportUISetupType

Zusätzliche Konfigurationsmöglichkeiten um die Visualisierung zu beschreiben.

Parameter	Datentyp	Beschreibung
FileListSize	UINT	Zahl an Dateien die auf einer Seite der Liste auf der Visualisierung dargestellt werden können. Es können maximal 20 Dateien angezeigt werden.
FileListScrollWindow	USINT	Zeigt an, wie viele Einträge der Liste beim Auf- oder Abscrollen im Vorhinein angezeigt werden.

4.6.1.1.5 MpIOInfoType

Mit diesem Datentyp wird für den Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpIODiagType	Eine Diagnosestruktur für die StatusID.

4.6.1.1.6 MpIOStatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	MpIOErrorEnum	Fehlerkode des Funktionsbausteins.
Severity	MpComSeveritiesEnum	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler).
Code	UINT	Code der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden.

4.6.1.1.7 MpIOUIFileListType

Dieser Datentyp enthält eine Liste an Dateien und die zugehörige Navigation.

Parameter	Datentyp	Beschreibung
Names	`ARRAY[0..19] von STRING[100]`	Liste aller für den Import verfügbaren Dateien.
Scroll	`MpIOUIScrollSelectType`	Enthält alle Datenpunkte welche auf der Visualisierung zum Scrollen in der Liste verwendet werden.
Refresh	`BOOL`	Befehl zum neu Laden der Liste.

4.6.1.1.8 MpIOUIMessageBoxType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern.

Parameter	Datentyp	Beschreibung
LayerStatus	`UINT`	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Type	`MpIOUIMessageEnum`	Art des Dialogfensters.
StatusID	`DINT`	Aktuelle StatusID zum Anzeigen. **VC4 Anbindung:** Datapoint von *Numeric*
ErrorNumber	`UINT`	Aktuelle Fehlernummer zum Anzeigen (gleicht dem Inhalt von "StatusID") **VC4 Anbindung:** Datapoint von *Numeric*
Confirm	`BOOL`	Bestätigen der Operation. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	`BOOL`	Abbrechen der Operation. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1

4.6.1.1.9 MpIOUIScrollSelectType

Dieser Datentyp enthält alle notwendigen Datenpunkte, um in einer Liste zu scrollen.

Parameter	Datentyp	Beschreibung
MaxSelection	`UINT`	Index des letzten Eintrages in der Liste. **VC4 Anbindung:** MaxDatapoint von *Listbox*
SelectedIndex	`UINT`	Index des aktuell ausgewählten Eintrages in der Liste. **VC4 Anbindung:** IndexDatapoint von *Listbox*
PageUp	`BOOL`	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. Die Größe der Seite wird über den Parameter "FileListSize" der Struktur MpIOImportUISetupType definiert. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepUp	`BOOL`	Wählt vorhergehenden Eintrag in der Liste aus. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepDown	`BOOL`	Wählt nächsten Eintrag in der Liste aus. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0

Parameter	Datentyp	Beschreibung
PageDown	BOOL	Springt an das Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. Die Größe der Seite wird über den Parameter "FileListSize" der Struktur MpIOImportUISetupType definiert. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
RangeStart	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von *Scale*
RangeEnd	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von *Scale*
ScrollStatus	UINT	Statusdatenpunkt zum Umschalten der Sichtbarkeit der Scroll-Icons.

4.6.1.2 Enumeratoren

4.6.1.2.1 MpIOUIMessageEnum

Dieser Aufzählungsdatentyp definiert die Meldungen welche in einer Dialogbox angezeigt werden können.

Aufzählung	Beschreibung
mpIO_UI_MSG_NONE	Die Dialogbox wird nicht angezeigt.
mpIO_UI_MSG_ERROR	Die Dialogbox zeigt einen Fehler an.
mpIO_UI_MSG_CONFIRM_IMPORT	Die Dialogbox zeigt einen Bestätigungsdialog für den Import an.
mpIO_UI_MSG_CONFIRM_REBOOT	Die Dialogbox zeigt einen Bestätigungsdialog für den Import mit anschließendem Warmstart an.

4.6.1.2.2 MpIOImportUIStatusEnum

Dieser Aufzählungsdatentyp gibt Auskunft über die aktuelle Aktivität der Komponente.

Aufzählung	Beschreibung
mpIO_IMPORTUI_STATUS_IDLE	Aktuell ist kein Prozess aktiv.
mpIO_IMPORTUI_STATUS_REFRESH	Die Dateiliste wird neu geladen.
mpIO_IMPORTUI_STATUS_BUILD	Die Import-Daten werden von der Import-Datei erstellt.
mpIO_IMPORTUI_STATUS_DIALOG	Ein Dialog ist aktiv. Warten auf Benutzereingabe.
mpIO_IMPORTUI_STATUS_DISPOSE	Import-Daten werden verworfen (Nach Abbruch durch Benutzer).
mpIO_IMPORTUI_STATUS_IMPORT	Import der Hardware-Konfiguration wird ausgeführt.
mpIO_IMPORTUI_STATUS_ERROR	Während der letzten Operation ist ein Fehler aufgetreten.

4.6.2 Funktionsbausteine

Unterstützt ab	Name	Beschreibung
5.0.0	MpIOImport	Mithilfe dieses Funktionsbausteins kann eine neue Hardware-Konfiguration importiert werden.
5.0.0	MpIOImportUI	Dieser Funktionsbaustein stellt ein Benutzerinterface für den Import zur Verfügung.

4.6.2.1 MpIOImport

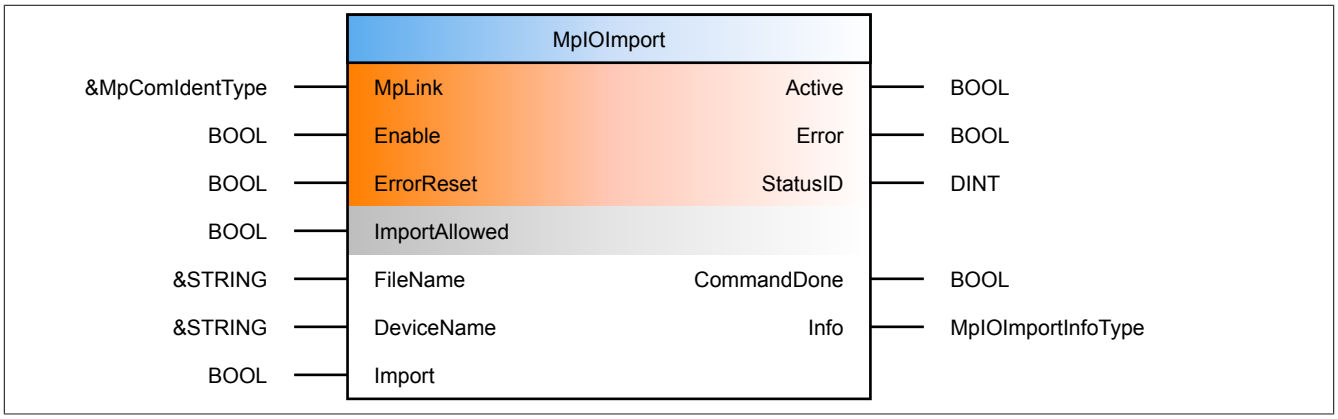
Mithilfe dieses Funktionsbausteins kann eine neue Hardware-Konfiguration importiert werden.

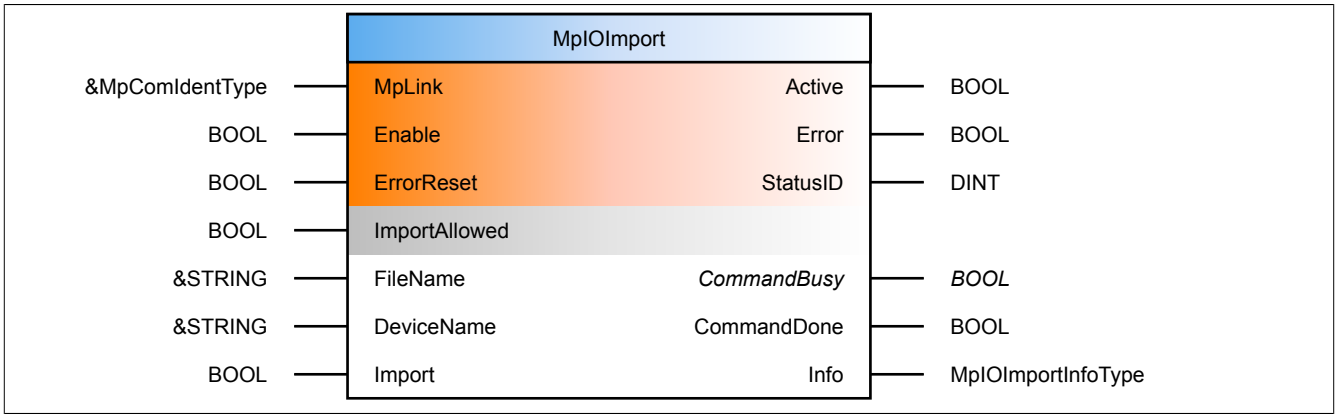
Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp IO (GER/ENG) zu finden. (Zugangsdaten)

Auf unserem Youtube Kanal sind viele Tutorials zum Thema zu finden.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpIO Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	ImportAllowed	`BOOL`	Definiert, ob ein Import ausgeführt werden darf.
IN	FileName	`Zeiger auf STRING[255]`	Name der zu importierenden Datei.
IN	DeviceName	`Zeiger auf STRING[50]`	Name des FileDevice, auf welchem die Datei liegt.
IN	Import	`BOOL`	Befehl zum Importieren der Konfiguration.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpIOImportInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

4.6.2.1.1 Beschreibung

Dieser Funktionsbaustein benötigt den MpLink einer MpIO Konfiguration.

Der MpIOImport Funktionsbaustein verwaltet den Import von Hardware-Konfigurationen. Zusätzlich zu veränderten Hardware-Konfigurationen kann auch das NC-Mapping oder das IO-Mapping über eine neue Konfiguration importiert werden. Durch "ImportAllowed" können zwei Funktionen durchgeführt werden:

Importieren einer neuen Hardware-Konfiguration

Über den Eingang "Import" wird die Hardware-Konfiguration, welche über den Eingang "FileName" definiert wird, importiert. Der Eingang "ImportAllowed" wird als zusätzlicher Sicherheitsmechanismus genutzt, um zu verhindern, dass eine Konfiguration zum falschen Zeitpunkt importiert wird. Um eine neue Konfiguration zu importieren muss also "Import = `TRUE`" sowie "ImportAllowed = `TRUE`" sein. Bei "CommandDone = `TRUE`" wurde das Kommando fertig ausgeführt. Um ein neues Kommando zu starten, muss der vorherige Befehl auf `FALSE` gesetzt werden. Der Funktionsbaustein ist nun bereit einen neuen Befehl auszuführen.

Überprüfung der Hardware-Konfiguration

Bei der Aktivierung von mapp IO ("Enable = `TRUE`") wird geprüft, ob die Hardware-Konfiguration vom Backup File Device und die aus der ROM geladene Hardwarekonfiguration, identisch sind. Ist das nicht der Fall, dann wurde die Hardware-Konfiguration extern (= nicht durch mapp IO) verändert. Es kann nun die neue Hardware-Konfiguration weiter verwendet werden oder es wird die zuletzt über mapp IO importierte Konfiguration erneut aktiviert. Diese Entscheidung wird über den Parameter "ImportAllowed" gesteuert. Für mehr Information dazu siehe Abschnitt Verwalten der Hardware-Konfiguration.

Import-Datei

Die Import-Datei ist kompatibel mit arconfig-/ iomap-/ und ncmapping-Format, welches von der AR verwendet wird. Daher ist es möglich, die Daten, welche von Automation Studio generiert wurden, zu importieren.

Zusätzlich dazu können existierende Daten von der originalen AR Konfiguration oder von Template Dateien referenziert werden. Während eines Imports werden alle diese Dateien kombiniert, um die Hardware-Konfiguration zu erstellen.

Weitere Informationen zum Aufbau der Importdatei können hier gefunden werden.

Plausibilitätsprüfungen während dem Import

Während des Imports kann die Zuordnung zwischen Prozessvariable und Hardware-Modul auf bestimmte Merkmale geprüft werden ob. Im speziellen kann geprüft werden, ob...

- ...die angegebenen Prozessvariablen auf der Steuerung existieren.
- ... die Datentypen der Prozessvariablen mit den Datentypen der Ein- / Ausgänge übereinstimmen.
- ... eine PV auf mehrere Eingänge verbunden wurde.
- ... ein Modul, welches in der Importdatei angegeben wurde, nicht existiert.
- ... ein Modul an einer nicht existierenden Position oder an einer bereits belegten Position eingefügt wurde.

Wenn eine dieser Prüfungen fehlschlägt, werden die Daten trotzdem soweit möglich importiert und ein Fehler wird am Funktionsbaustein ausgegeben.

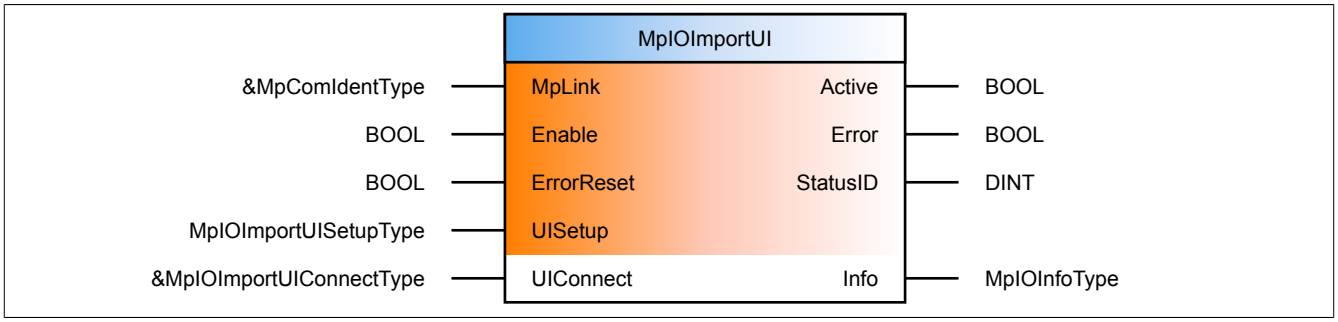
> **Wenn der Funktionsbaustein während des Projektdownloads aufgerufen wird, existieren manche PVs vielleicht noch nicht auf der Steuerung. Der Funktionsbaustein wird dann einen Fehler anzeigen.**

Weitere Informationen sind im Abschnitt MpIO Konfiguration zu finden.

4.6.2.2 MpIOImportUI

Dieser Funktionsbaustein stellt ein Benutzerinterface für den Import zur Verfügung.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpIO Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.

I/O	Parameter	Datentyp	Beschreibung
IN	UISetup	MpIOImportUISetupType	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
IN	UIConnect	Zeiger auf MpIOImportUIConnectType	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
OUT	Active	BOOL	FB ist aktiv.
OUT	Error	BOOL	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	DINT	Statusinformation.
OUT	Info	MpIOInfoType	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

4.6.2.2.1 Beschreibung

Der MpLink für diesen Funktionsbaustein ist derselbe, der auch am MpIOImport Funktionsbaustein verwendet wird. Die Komponente stellt eine Verbindung zwischen MpIOImport und einer Visualisierung dar. Um MpIOImportUI zu verwenden, muss eine MpIOImport Komponente aktiv sein.

Pro MpIOImport kann nur ein MpIOImportUI verwendet werden. Es ist nicht möglich mehrere MpIOImportUI-Instanzen zu verwenden.

MpIOImportUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

- **DefaultLayerStatus**: Zeigt an, ob der Layer mit der Liste der möglichen Import-Dateien angezeigt wird.
- **Status**: Aktuelle Operation.
- **FileList**: Beinhaltet eine Liste aller für den Import verfügbaren Dateien.
- **MessageBox**: In der Struktur MpIOUIMessageBoxType können die unterschiedlichen Dialogfenster für MpIO konfiguriert werden.
- **Import**: Über dieses Kommando kann ein Import gestartet werden.

MpIOImportIOSetupType

In dieser Struktur wird über "FileListSize" definiert, wieviele Import-Dateien auf einer Seite der Visualisierung angezeigt werden. Über "FileListScrollWindow" wird entschieden, wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden.

4.6.3 Statusnummern

4.6.3.1 -1064120312: Import gesperrt

Beschreibung:

Der Import Befehl ist gesperrt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen des Eingangs "ImportAllowed" an MpIOImport.

Konstante

`mpIO_ERR_IMPORT_NOT_ALLOWED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport

4.6.3.2 -1064120314: MpIOImport ist noch nicht aktiv

Beschreibung:

Der Funktionsbaustein MpIOImport ist nicht aktiv.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Aktivieren des MpIOImport Funktionsbausteins.

Konstante

`mpIO_ERR_MISSING_IMPORT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImportUI

4.6.3.3 -1064120315: Fehlender Wert an UIConnect

Beschreibung:

An "UIConnect" wurde NULL angehängt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Eingang "UIConnect" vergessen

Konstante

`mpIO_ERR_MISSING_UICONNECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImportUI

4.6.3.4 -1064120316: Fehler beim Erstellen der Backup-Dateien

Beschreibung:

Beim Erstellen der Backup-Dateien ist ein Fehler aufgetreten. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- In der MpIO Konfiguration angegebene "Backup Device Name" überprüfen.

Konstante

`mpIO_ERR_CREATE_BACKUP`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport

4.6.3.5 -1064120317: Fehler beim Aktivieren der importierten Hardware Konfiguration

Beschreibung:

Beim Aktivieren der Importieren auf der Steuerung ist ein Fehler aufgetreten. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob die in der Import-Datei angegebenen Daten gültig sind.

Konstante

`mpIO_ERR_ACTIVATE_IMPORT_DATA`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport

4.6.3.6 -1064120318: Fehler beim Erstellen der Importdaten

Beschreibung:

Beim Erstellen der Importdaten ist ein Fehler aufgetreten. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen der Importdatei

Konstante

`mpIO_ERR_BUILD_IMPORT_DATA`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport

4.6.3.7 -1064120319: Ungültiger Dateiname

Beschreibung:

Der Dateiname am Eingang "FileName" ist ungültig. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Am Eingang "FileName" ist nichts angehängt
- Die angehängte Datei existiert nicht
- Keine Zugriffsberechtigung

Konstante

`mpIO_ERR_INVALID_FILE_NAME`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport

4.6.3.8 -1064120320: Ungültiges File Device

Beschreibung:

Das angegebene File Device am Eingang "DeviceName" ist ungültig. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Am Eingang "DeviceName" ist nichts angehängt
- Das angehängte File Device existiert nicht
- Es konnte keine Verbindung zum gewünschten File Device erstellt werden

Konstante

`mpIO_ERR_INVALID_FILE_DEV`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport

4.6.3.9 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die Mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport

4.6.3.10 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsblocks ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsblöcken)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport
- MpIOImportUI

4.6.3.11 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport
- MpIOImportUI

4.6.3.12 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport
- MpIOImportUI

4.6.3.13 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsblock überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport
- MpIOImportUI

4.6.3.14 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsblocks nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImport
- MpIOImportUI

4.6.3.15 -1064293112: Ungültiger Import-Modus

Beschreibung:

Für den Import wurde an {2:ID} ein ungültiger Modus definiert.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Der angegebene Modus ist für diese Konfiguration ungültig.

Zusätzliche Informationen:

- {2:ID}: Die Id des betroffenen "Module"- oder "LN"-Knotens in der Import-Datei.

4.6.3.16 -1064293113: Fehler beim Kopieren der Moduldaten

Beschreibung:

Beim Kopieren der Import-Daten zum Modul {2:ModuleID} ist ein Fehler aufgetreten.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Zusätzliche Informationen:

- {2:ModuleID}: Id des betroffenen Moduls.

4.6.3.17 -1064293114: NC-Zuordnung fehlt

Beschreibung:

Die angegebene NC-Zuordnung fehlt in der Master-Konfiguration. Es ist nicht möglich, ein NC-Mapping auf diesen Kanal {2:ChannelNumber} zu erstellen.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob der angegebene Kanal am Modul existiert.

Zusätzliche Informationen:

- {2:ChannelNumber}: Nummer des betroffenen Kanals am Modul.

4.6.3.18 -1064293115: Ungültiges Interface

Beschreibung:

Das Modul {2:Module} wurde an ein ungültiges Interface verbunden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob das Modul mit dem Interface verbunden werden kann.

Zusätzliche Informationen:

- {2:Module}: Id des betroffenen Moduls.

4.6.3.19 -1064293116: Achsenreferenz ungültig

Beschreibung:

Es wurde eine ungültige Achsenreferenz {2:ID} in einem NC-Mapping definiert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob der Name der Achsreferenz in der Import-Datei stimmt.

Zusätzliche Informationen:

- {1:ErrorNumber}: Grund des Fehlers.
- {2:ID}: Id der betroffenen Achsreferenz.

4.6.3.20 -1064293118: Nicht aufgelöste "Cross-Links" gefunden

Beschreibung:

In der Import-Datei wurden nicht aufgelöste "Cross-Links" gefunden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Aktivieren des Konfigurationsparameters "Remove Unresolved Links".

4.6.3.21 -1064293119: Fehler beim Ändern der Konfiguration

Beschreibung:

Beim Ändern der Konfiguration {2:AR|IO|NC} ist ein Fehler aufgetreten.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob der angegebene Konfigurationsparameter geändert werden kann.

Zusätzliche Informationen:

- {1:ErrorNumber}: Grund des Fehlers.
- {2:AR|IO|NC}: Zeigt, an welcher Konfiguration der Fehler aufgetreten ist.

4.6.3.22 -1064293120: Variable mehrfach verbunden

Beschreibung:

Die PV {2:PvName} wurde auf mehrere Eingänge verbunden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Eine PV darf nicht auf mehrere Eingänge verbunden werden

Zusätzliche Informationen:

- {2:PvName}: Name der betroffenen Prozessvariable.

4.6.3.23 -1064293121: Ungültiges Modul

Beschreibung:

Das in der Import-Datei angegebene Modul existiert nicht in der Master-Konfiguration.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob das Modul in der Master-Konfiguration existiert.
- Überprüfen, ob das Modul in der Import-Datei richtig angegeben ist.

4.6.3.24 -1064293122: Fehler beim Kopieren der Import-Datei

Beschreibung:

Beim Kopieren der Import-Datei {2:FileName} zum Backup-Speichermedium ist ein Fehler aufgetreten.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob genug Speicher vorhanden ist.
- Überprüfen, ob der Zugriff auf das Speichermedium erlaubt / möglich ist.

Zusätzliche Informationen:

- {1:ErrorNumber}: Grund des Fehlers.
- {2:FileName}: Name der betroffenen Datei.

4.6.3.25 -1064293123: Knoten fehlerhaft definiert

Beschreibung:

Ein Knoten in der Import-Datei ist fehlerhaft definiert.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen der Import-Datei.

Zusätzliche Informationen:

- {1:ErrorNumber}: Grund des Fehlers.
- {2:ID}: Die ID des betroffenen "Module"-Knotens.

4.6.3.26 -1064293124: PV passt nicht in Filter

Beschreibung:

Die angegebene PV {2:PvName} kann nicht verlinkt werden, da sie keinen in der Konfiguration als Filter angegebenen Präfix enthält.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob der Filter richtig konfiguriert ist.
- Überprüfen, ob der Name der PV richtig ist.

Zusätzliche Informationen:

- {2:PvName}: Name der betroffenen Prozessvariable.

4.6.3.27 -1064293125: PV existiert nicht

Beschreibung:

Die für das IO-Mapping angegebene PV {2:PvName} existiert nicht auf der Steuerung.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Wenn bereits im INIT-Teil des Tasks ein Import gestartet wird, kann es sein, dass manche PVs noch nicht existieren.
- Überprüfen, ob die angegebene PV auf der Steuerung existiert.

Zusätzliche Informationen:

- {2:PvName}: Angegebener Name der betroffenen Prozessvariable.

4.6.3.28 -1064293126: Ungültiger Datentyp

Beschreibung:

Der Datentyp {3:PvType} der Prozessvariable {2:PvName} stimmt nicht mit dem Datentyp des IO-Datenpunktes {4:IoType} überein.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob der Datentyp der PV mit dem des IO-Datenpunktes übereinstimmt.

Zusätzliche Informationen:

- {2:PvName}: Name der betroffenen Prozessvariable.
- {3:PvType}: Der Datentyp der betroffenen Prozessvariable.
- {4:IoType}: Der Datentyp des IO-Datenpunktes.

4.6.3.29 -1064293127: Datentyp unbekannt

Beschreibung:

Der in der Import-Datei für die Prozessvariable {2:PvName} angegebene Datentyp {3:Type} konnte nicht erkannt werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob der angegebene Datentyp bekannt ist.

Zusätzliche Informationen:

- {2:PvName}: Der Name der betroffenen Prozessvariable.
- {3:Type}: Der Datentyp, welcher nicht erkannt wurde.

4.6.3.30 -1064293128: Ungültiger Dateiinhalt

Beschreibung:

Der Inhalt der Import-Datei {2:FileName} vom Speichermedium {3:DeviceName} ist ungültig. Für den Import benötigte Knoten konnten nicht gefunden werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob die Struktur der Import-Datei der benötigten Struktur entspricht.

Zusätzliche Informationen:

- {2:FileName}: Name der betroffenen Datei.
- {3:DeviceName}: Name des Speichermediums, auf welchem sich die Datei befindet.

4.6.3.31 -1064293129: Datenpunkt fehlt

Beschreibung:

Der angegebene Datenpunkt {2:DatapointID} fehlt für das Modul. Es ist nicht möglich, ein IO-Mapping auf diesen Datenpunkt zu erstellen.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Sicherstellen, dass der angegebene Datenpunkt für das Modul existiert.

Zusätzliche Informationen:

- {2:DatapointID}: Id des gewünschten Datenpunktes.

4.6.3.32 -1064293130: Konflikt beim Importieren

Beschreibung:

Beim Importieren der Daten {2:ImportSource} ist ein Konflikt aufgetreten. Unveränderbare Eigenschaften können nicht überschrieben werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob unveränderbare Konfigurations-Eigenschaften überschrieben werden.

Zusätzliche Informationen:

- {2:ImportSource}: Zeigt, ob die NC-Zuordnungstabelle oder die IO-Zuordnungstabelle betroffen ist.
- {3:NodeID}: Der in der Import-Datei von dem Fehler betroffenen Knoten.
- {4:AffectedSubNode}: Der betroffenen Kind-Knoten.

4.6.3.33 -1064293131: Fehler beim Verlinken des Moduls

Beschreibung:

Beim Verlinken des Moduls {2:ModuleID} aus der Master-Konfiguration mit {3:ModuleSourceID} {4:SourceID} ist ein Fehler aufgetreten.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob das angegebene Modul in der Master-Konfiguration existiert.

Zusätzliche Informationen:

- {2:ModuleID}: Id des betroffenen Moduls.
- {3:ModuleSourceID}: Der angegebene "SourceName" des zu verlinkenden Moduls.
- {4:SourceID}: Die angegebene "SourceID" des zu verlinkenden Moduls.

4.6.3.34 -1064293132: Modul existiert bereits

Beschreibung:

Das angegebene Modul {2:ModuleID} wurde bereits verlinkt und kann kein zweites Mal verwendet werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Entfernen einer Verlinkung des Moduls.
- Ändern der Master-Konfiguration zu "None".

Zusätzliche Informationen:

- {2:ModuleID}: Id des betroffenen Moduls.

4.6.3.35 -1064293133: SourceID fehlt

Beschreibung:

Für das einzufügende Modul {2:ModuleID} wurde keine "SourceID" angegeben.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob eine ID angegeben wurde.

Zusätzliche Informationen:

- {2:ModuleID}: Die ID des einzufügenden Moduls.

4.6.3.36 -1064293134: Modul wurde nicht gefunden

Beschreibung:

Das Modul {2:ModuleID} konnte in der angegebenen Quelle {3:SourceID} nicht gefunden werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob das Modul in der ursprünglichen AR-Konfiguration vorhanden ist.
- Überprüfen, ob für das Modul Template-Dateien vorhanden sind.

Zusätzliche Informationen:

- {1:ErrorNumber}: Grund des Fehlers.
- {2:ModuleID}: Das einzufügende Modul.
- {3:SourceID:}: Die angegeben Position des Moduls in der ursprünglichen AR-Konfiguration.

4.6.3.37 -1064293135: Import-Quelle ungültig

Beschreibung:

Die angegebene "SourceID" {2:SourceID} konnte nicht in der AR-Konfiguration gefunden werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob als "Source" entweder "AR" oder "Template" angegeben wurde.

Zusätzliche Informationen:

- {2:SourceID}: Die Position des zu importierenden Moduls.
- {3:ModuleID}: Die Modul-ID des zu imoprtierenden Moduls.
- {4:Offset}: Die Position in der Import-Datei, an welcher der Fehler aufgetreten ist.

4.6.3.38 -1064293136: Attribut "ID" fehlt

Beschreibung:

Für ein Modul wurde keine "ID" definiert.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Definieren einer ID für das Modul

Zusätzliche Informationen:

- {2:DebugOffset}: Position, an der der Fehler aufgetreten ist.

4.6.3.39 -1064293137: AR-Modul nicht gefunden

Beschreibung:

Das in der Import-Datei angegebene Modul {2:ModuleName} kann nicht in der AR-Konfiguration gefunden werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Einfügen des Moduls in der AR-Konfiguration.
- Verwenden von Template-Dateien.

Zusätzliche Informationen:

- {2:ModuleName}: Name des betroffenen Moduls.

4.6.3.40 -1064293138: Fehler beim Parsen der Import-Datei

Beschreibung:

Beim Parsen der XML-Daten aus der Import-Datei {2:FileName} vom Speichermedium {3:DeviceName} ist an der Position {5:OffsetInFile} folgender Fehler aufgetreten: {4:ErrorDescription}.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob die Syntax der XML-Daten korrekt ist.

Zusätzliche Informationen:

- {2:FileName}: Name der betroffenen Import-Datei.
- {3:DeviceName}: Speichermedium, auf welchem sich die Import-Datei befindet.
- {4:ErrorDescription}: Beschreibung des Fehlers in der Import-Datei.
- {5:OffsetInFile}: Zeigt an, an welcher Position in der Import-Datei der Fehler aufgetreten ist.

4.6.3.41 -1064293139: Fehler beim Lesen der Datei

Beschreibung:

Beim Lesen der Import-Datei {2:FileName} vom Speichermedium {3:DeviceName} ist ein Fehler aufgetreten.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob der Zugriff auf die Datei erlaubt ist.

Zusätzliche Informationen:

- {1:ErrorCause}: Grund des Fehlers.
- {2:FileName}: Dateiname der betroffenen Import-Datei.
- {3:DeviceName}: Speichermedium, auf welchem sich die Import-Datei befindet.

4.6.3.42 -1064293140: Import-Datei nicht gefunden

Beschreibung:

Die Import-Datei (.ar/.xml) konnte nicht gefunden werden.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen, ob die Import-Datei existiert.

4.6.3.43 1083190531: Neustart erforderlich

Beschreibung:

Zum Aktivieren der Hardware-Konfiguration wird ein Neustart der Steuerung benötigt.

Ursache / Abhilfe:

- Zum Aktivieren der Hardware-Konfiguration wird ein Neustart benötigt.

4.6.3.44 1083363335: Warte auf Import FB

Beschreibung:

Der Funktionsbaustein kann erst aktiv werden, wenn MpIOImport aktiv ist. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird diese Information an dem Ausgang "StatusID" anzeigen.

Ursache / Abhilfe:

- MpIOImport aktivieren.

Konstante

`mpIO_INF_WAIT_IMPORT_FB`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpIOImportUI

4.6.4 Alarme

4.6.4.1 mpIO_ALM_IMPORT_FAILED: Importieren der Hardware-Konfiguration fehlgeschlagen

Beschreibung:

Das Importieren der Hardware-Konfiguration ist fehlgeschlagen. Weitere Details siehe Logger.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Der Inhalt der Import-Datei ist fehlerhaft.

4.6.4.2 mpIO_ALM_IMPORT_DONE: Import erfolgreich

Beschreibung:

Der Import der neuen Hardware-Konfiguration war erfolgreich.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Import einer Hardware-Konfiguration erfolgreich ausgeführt.

4.6.4.3 mpIO_ALM_CONFIG_CHANGED: Hardware-Konfiguration wurde verändert

Beschreibung:

Die Hardware-Konfiguration wurde (extern) verändert. Weitere Details siehe Logger.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Ein Download mit unterschiedlicher Hardware-Konfiguration wurde über Automation Studio durchgeführt
- Ein Download mit unterschiedlicher Hardware-Konfiguration wurde über das Runtime Utility Center durchgeführt
- Die Hardware-Konfiguration wurde über Bibliotheken wie AsIOMMan oder AsIOAcc manipuliert

5 mapp Skyline: Maschinenlinie visualisieren

mapp Skyline wird in Kombination mit dem Skyline-Widget verwendet. Mithilfe dieser Komponente kann eine Maschinenlinie in einer Visualisierung grafisch dargestellt werden.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Skyline (GER/ENG) zu finden. (Zugangsdaten)

5.1 Konzept

Allgemeines

- Als Applikationist möchte ich schnell und einfach meine Maschinenlinie in der Applikation sowie für die Visualisierung konfigurieren

Maschinenlinie visualisieren

- Als Endnutzer möchte ich meine Maschinenlinie visualisieren
- Als Endnutzer möchte ich die Zustände einzelner Maschinen der Maschinenlinie auf einen Blick sehen
- Als Endnutzer möchte ich bei Warnungen, Stillständen oder möglichen Servicarbeiten an der Maschinenlinie benachrichtigt werden

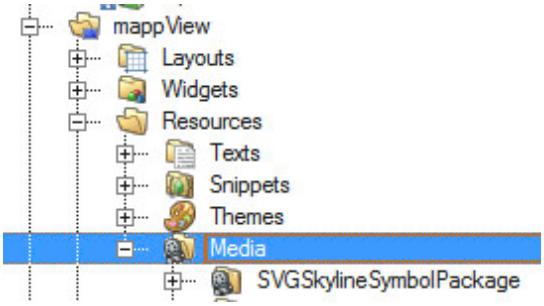
5.1.1 Grundkonzept

Ziel von mapp Skyline ist es, eine Maschinenlinie visuell darzustellen. Dabei können innerhalb der Linie verschiedene Benachrichtigungen (z.B.:Informationen, Warnungen) für den Maschinenbenutzer oder auch Maschinenzustände (z.B.:Ein-, oder Ausgeschaltet) grafisch dargestellt werden.

In diesem Abschnitt werden alle verwendeten Begriffe und Definitionen aufgelistet und erklärt.

Auf unserem B&R Tutorial Portal sind Tutorials zum Thema mapp Skyline (GER/ENG) zu finden. (Zugangsdaten)

In der von mapp View bereitsgestellten SVG Skyline Bibliothek sind Bilder enthalten, welche für die Skyline verwendet werden können. Die Bibliothek ist über die Toolbox verfügbar und kann im mapp View Paket in der Logical View unter Resources -> Media abgelegt werden.



Skyline

Als Skyline wird die anzuzeigende Maschinenline bezeichnet.

Der Aufbau wird in der MpSkyline Konfiguration definiert. Um die Skyline in einer mapp View Visualisierung darzustellen, wird das Skyline-Widget verwendet.

Abbildung 12: Beispiel einer Maschinenlinie

Modul-Typ

Für eine Skyline müssen Modul-Typen festgelegt werden. Das bedeutet es muss definiert werden aus welchen Maschinentypen die Maschinenlinie besteht. Ein Maschinentyp kann zum Beispiel eine Drucker-Einheit ("PrinterUnit") oder eine Stapel-Einheit ("StackerUnit") sein.

Abbildung 13: PrinterUnit

Abbildung 14: StackerUnit

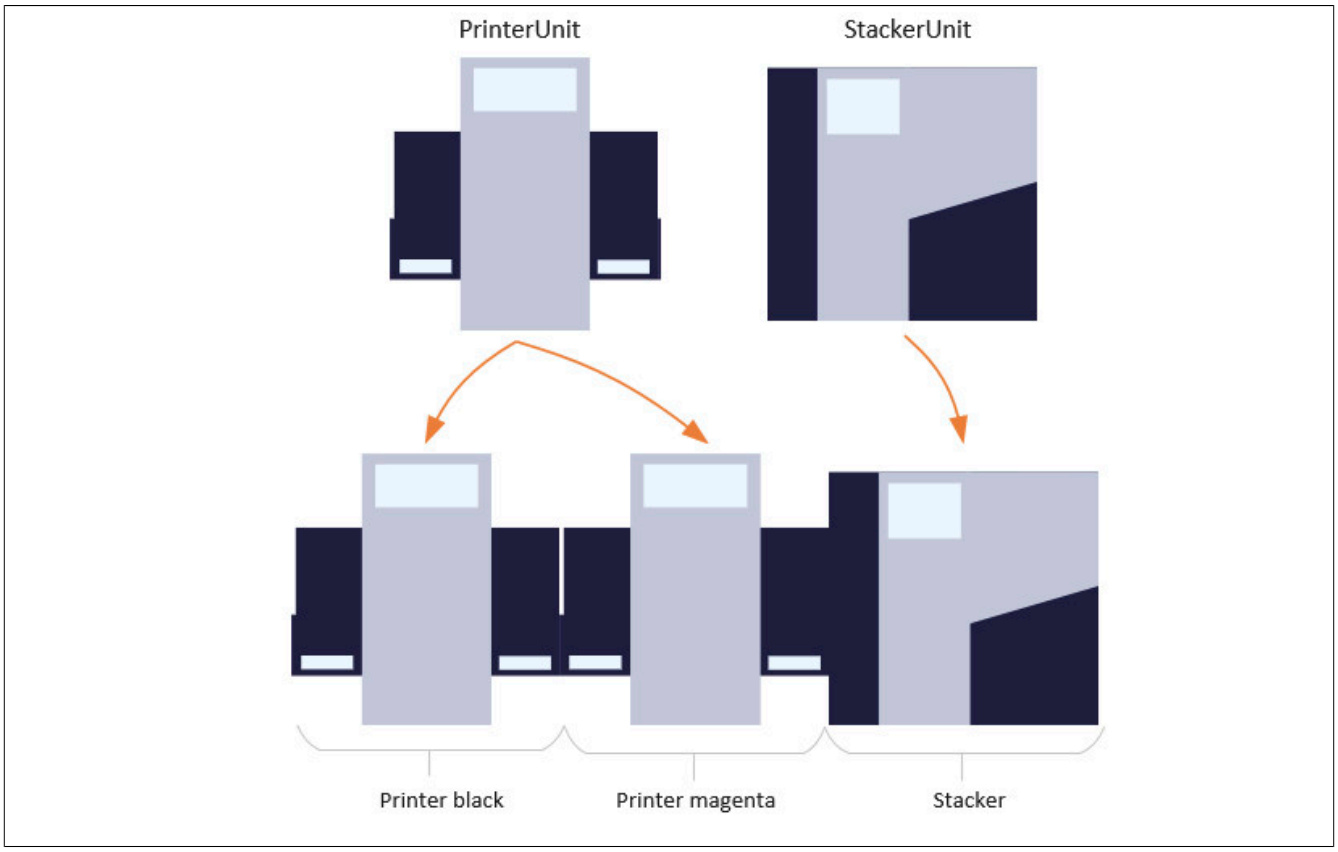
Die Modul-Typen werden in der MpSkyline Konfiguration unter "Module types" definiert.

Modul

Die Skyline wird mithilfe von Modulen erstellt. Einem Modul muss immer ein Modul-Typ zugewiesen werden. Dabei können ein oder mehrere Module vom gleichen Typen existieren.

Die Reihenfolge der Module, welche in der MpSkyline Konfiguration unter "Machine skyline" festgelegt wird, bestimmt das Aussehen der Skyline.

Im folgenden Beispiel besteht die Skyline aus zwei Modul-Typen, der "PrinterUnit" und der "StackerUnit".



Modul-Zustand

Ein Modul kann unterschiedliche Zustände annehmen. Es kann aber immer nur ein Zustand zu einem Zeitpunkt aktiv sein. Der Modul-Zustand kann zum Beispiel anzeigen, ob das Modul ein oder ausgeschalten ist oder ob sich die Maschine im Fehlerzustand befindet.

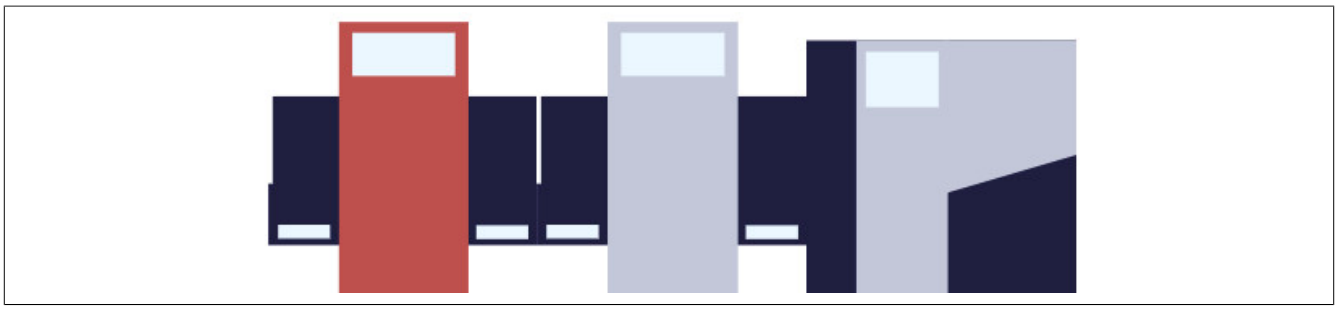


Abbildung 15: Printer black ist im Fehlerzustand

Welche Modul-Zustände existieren, wird in der MpSkyline Konfiguration unter "Module states collection" festgelegt. Pro Modul-Typ kann ein Zustandsmodell ausgewählt werden. Standardmäßig ist der erste festgelegte Zustand in der Konfiguration aktiv.

Um einen Zustand für ein Modul zu aktivieren, wird die Funktion MpSkylineSetModuleState verwendet. Wie die verschiedenen Zustände grafisch dargestellt werden, ist im Abschnitt Interaktion mit dem Widget des Skyline-Widgets erklärt.

Benachrichtigung

Jedes Modul kann ein oder mehrere Benachrichtigungen anzeigen, welche den Benutzer zum Beispiel auf Servicarbeiten, Warnungen oder Informationen hinweisen.

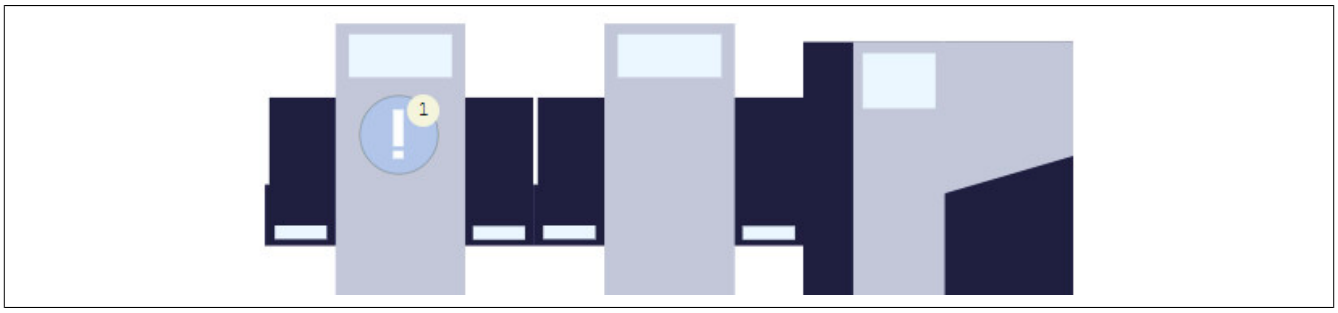


Abbildung 16: Printer black zeigt eine Information an

Die Benachrichtigungen werden in der MpSkyline Konfiguration unter "Notification collection" angegeben. Pro Modul-Typ kann eine Benachrichtigungsliste festgelegt werden.

Um eine Benachrichtigung für einen Modul-Typen zu aktivieren bzw. zu deaktivieren, werden die Funktionen MpSkylineSetNotification und MpSkylineResetNotification verwendet.

Wie eine Benachrichtigung im Skyline-Widgets dargestellt wird, ist im Abschnitt Interaktion mit dem Widget erklärt.

Optionen

Pro Modul-Typ können verschiedene Optionen erstellt werden. Das bedeutet für ein Modul kann nicht nur ein Zustand, sondern auch verschiedene Optionen festgelegt werden. Diese Optionen können dafür verwendet werden, um in einem Modul weitere Funktionalitäten ein-, oder auszuschalten. Eine Option kann zum Beispiel für einen Heizvorgang ("Heating") oder Kühlevorgang ("Cooling") eines Moduls verwendet werden.

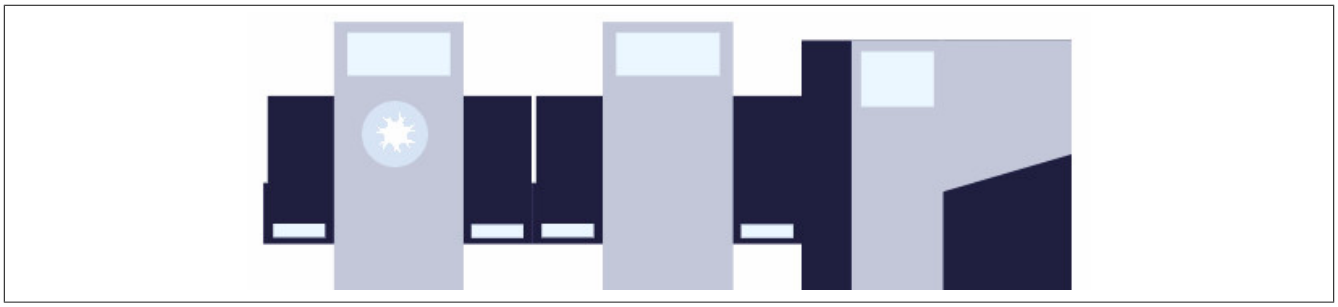


Abbildung 17: Option "Cooling" ist für Printer black aktiv

Die Optionen werden in der MpSkyline Konfiguration angegeben.

Eine Option kann verschiedene Zustände annehmen. Die Options-Zustände werden in der MpSkyline Konfiguration unter "Option state collection" definiert. Das bedeutet, die Option "Cooling" kann zum Beispiel den Zustand "On" oder "Off" besitzen.

Um einen Zustand für eine Option zu verändern, wird die Funktion MpSkylineSetOptionState verwendet. Um zu überprüfen, welcher Zustand aktuell für eine Option aktiv ist, wird die Funktion MpSkylineCheckOptionState verwendet.

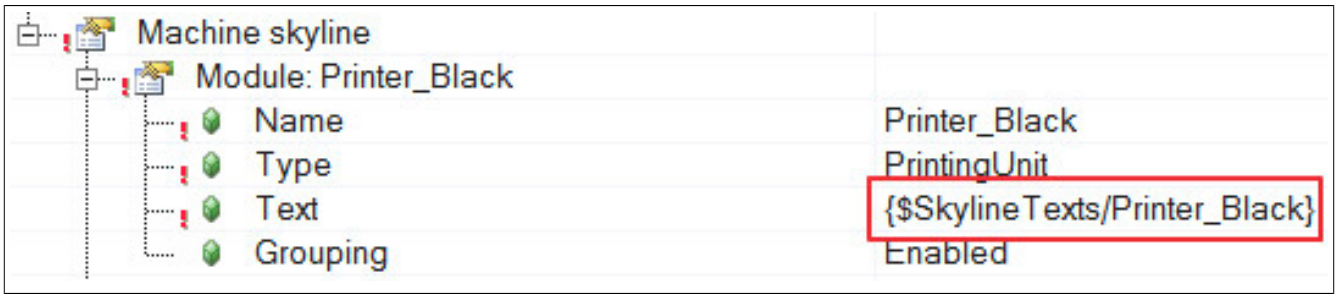
Wie eine Option im Skyline-Widgets dargestellt wird, ist im Abschnitt Interaktion mit dem Widget erklärt.

5.1.2 Texte lokalisiert über das Textsystem angeben

Auf unserem B&R Tutorial Portal wird über das Tutorial "Lokalisierbare Texte in mapp" (GER/ENG) erklärt, wie das Textsystem in Kombination mit mapp Services verwendet wird. (Zugangsdaten)

Texte können in der Skyline statisch oder lokalisiert über das Textsystem angegeben werden.

Um Texte zu lokalisieren wird das Automation Studio Textsystem verwendet. Dabei wird der Text nicht direkt angegeben, sondern es wird zum Textsystem verlinkt. Das könnte folgendermaßen aussehen:



Vorab wurde in der Logical View bereits eine Datei mit den eigentlichen Texten eingefügt. Diese Datei enthält den Text in jeder Projektsprache. Die Verbindung zwischen dem Textsystem und mapp Skyline wird über den Namespace und die TextID des Textes hergestellt. Den Namespace kann man innerhalb der Datei definieren. Die Text ID wird dann für jeden zu lokalisierenden Text in der Datei definiert. Namespace sowie Text ID werden in geschwungenen Klammern angegeben. Die Verbindung zum Textsystem wird über das $ - Symbol definiert: {$Namespace/TextID}

Abb.: Defined project languages

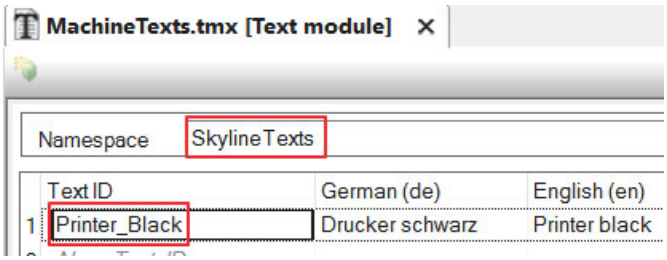
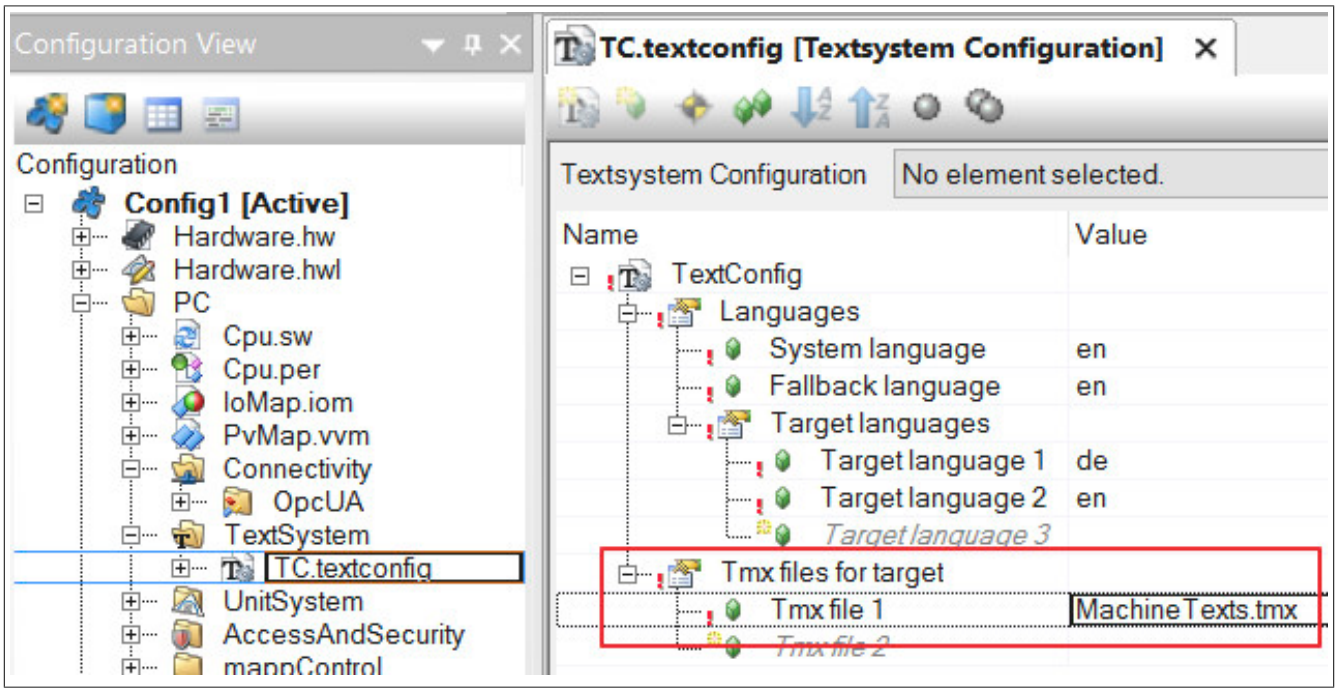


Abb.: Configured Text and Namespace

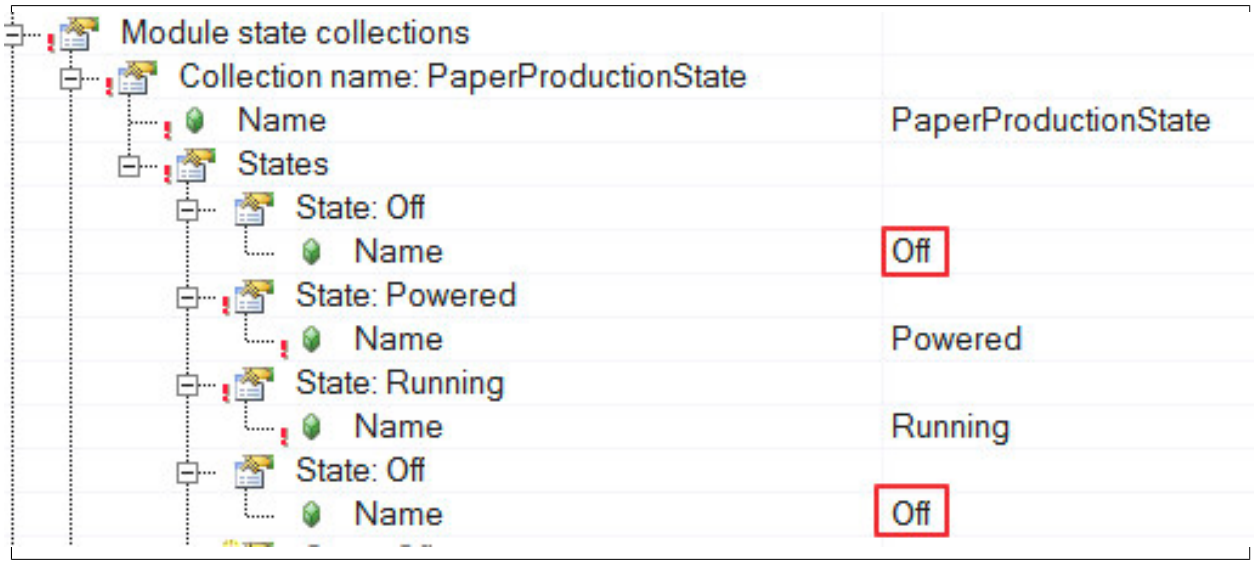
Wichtig hierbei ist, dass die .tmx Datei auch in der Textkonfiguration eingefügt wird:



5.1.3 Rahmenbedingungen

Folgende Rahmenbedinungen sind in mapp Skyline zu beachten:

- Es kann nur eine MpSkyline Konfiguration auf dem Zielsystem vorhanden sein
- Bei der Angabe von Zuständen, Benachrichtigungen und Optionen, kann für eine Liste der gleiche Name vergeben werden.



Dabei gilt es aber zu beachten, dass zur Laufzeit nur das erste in der jeweiligen Liste verwendet werden kann.

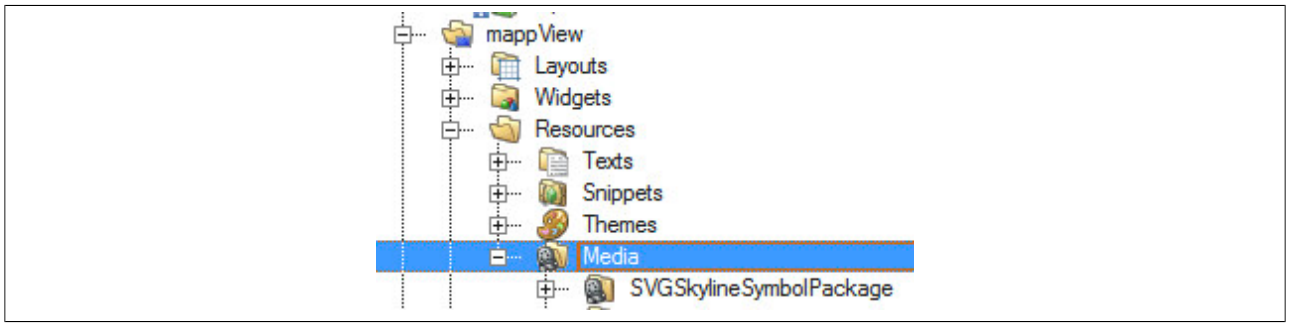
5.2 Getting Started

5.2.1 Maschinenlinie visualisieren

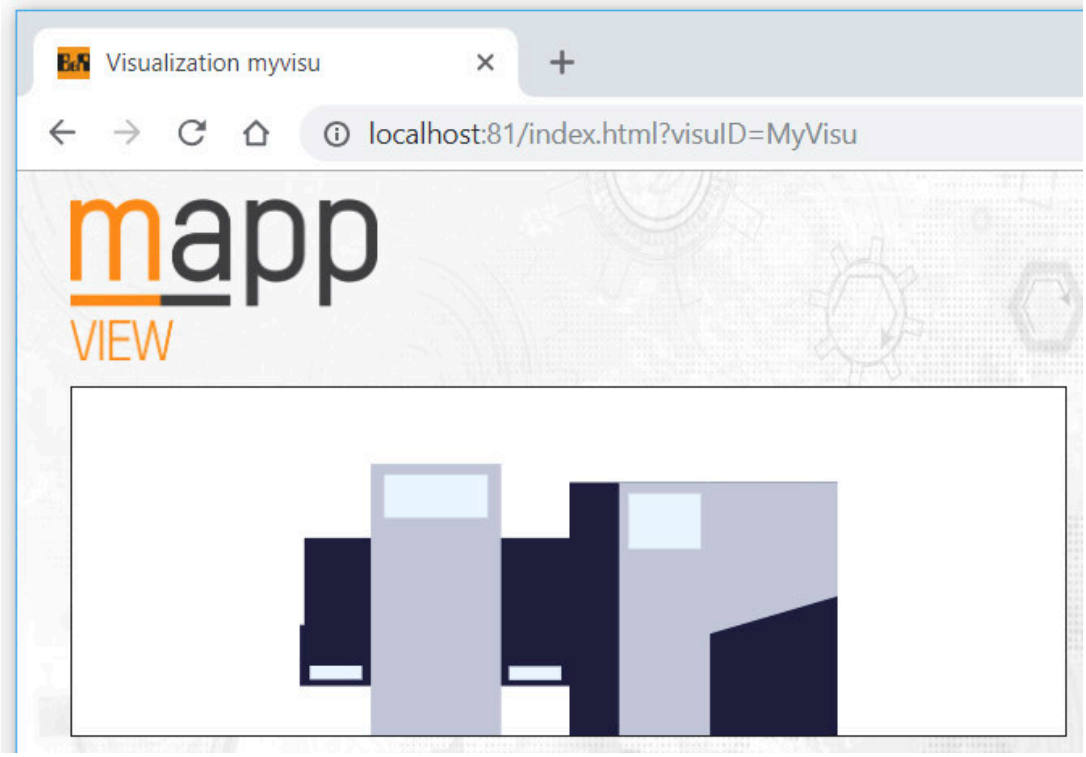
In diesem Abschnitt wird Schritt für Schritt erklärt, wie mithilfe von mapp Skyline und dem Skyline-Widget eine Maschinenlinie visualisiert werden kann.

In diesem Abschnitt werden Begriffe und Definitionen verwendet, welche im Grundkonzept von mapp Skyline erklärt werden.

In der von mapp View bereitsgestellten SVG Skyline Bibliothek sind Bilder enthalten, welche für die Skyline verwendet werden können. Die Bibliothek ist über die Toolbox verfügbar und kann im mapp View Paket in der Logical View unter Resources -> Media abgelegt werden.



Ziel ist es eine Druckermaschinen-Linie in einer mapp View Visualisierung anzeigen zu lassen.



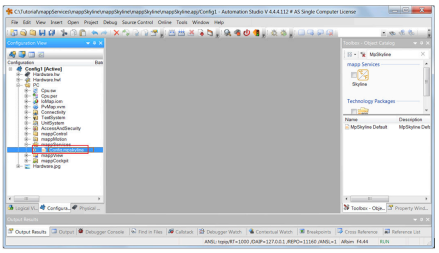
5.2.1.1 Projekt erstellen

Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

5.2.1.2 mapp Komponente einfügen

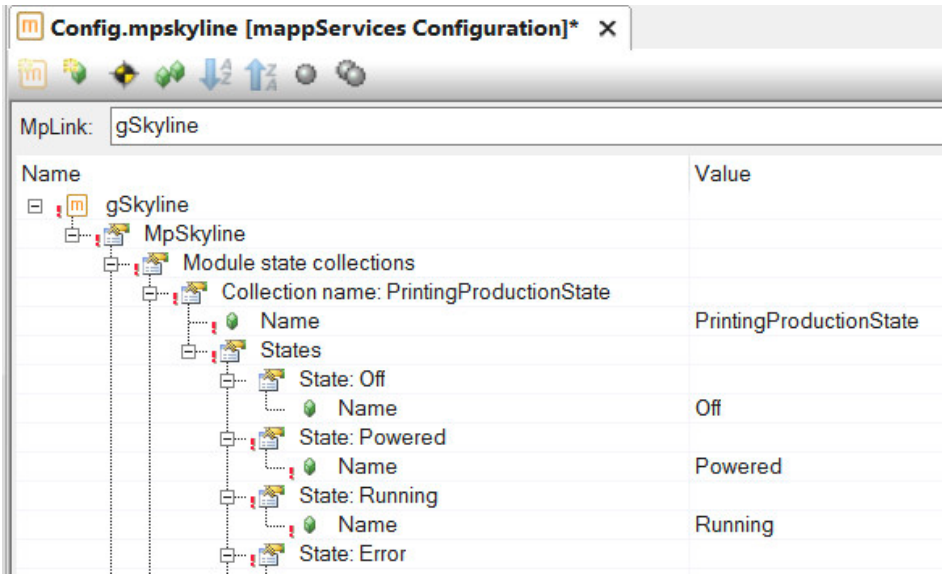
Konfiguration MpSkyline einfügen

Um die Skyline einer Maschinenlinie zu erstellen, wird die MpSkyline Konfiguration eingefügt.

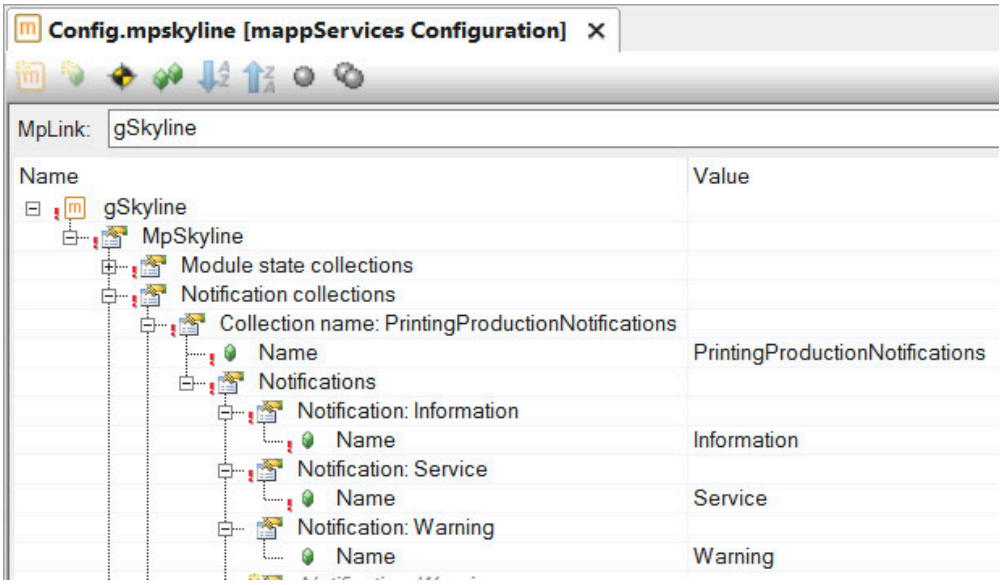


Hier befindet sich der mapp Link, welcher die mapp Komponente in der "world of mapp" eindeutig identifiziert. Der mapp Link stellt die Verbindung zwischen Programmierung und Konfiguration her.

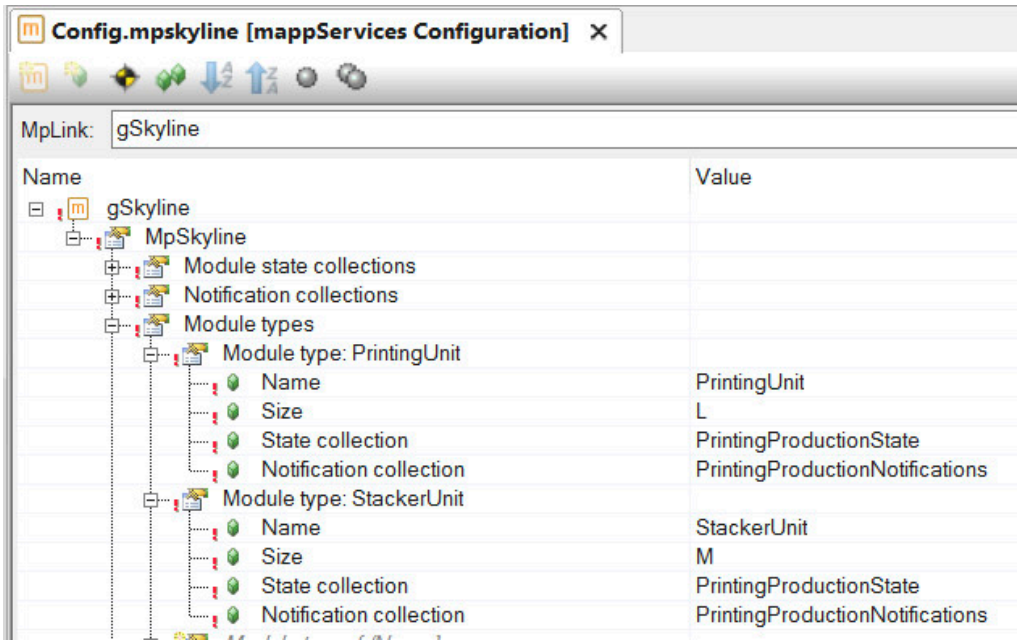
Unter "Module state collections" wird das Zustands-Modul "PrintingProductionState" angelegt. Es sollen die Zustände Off, Powered, Running und Error für die Maschinenlinie existieren.



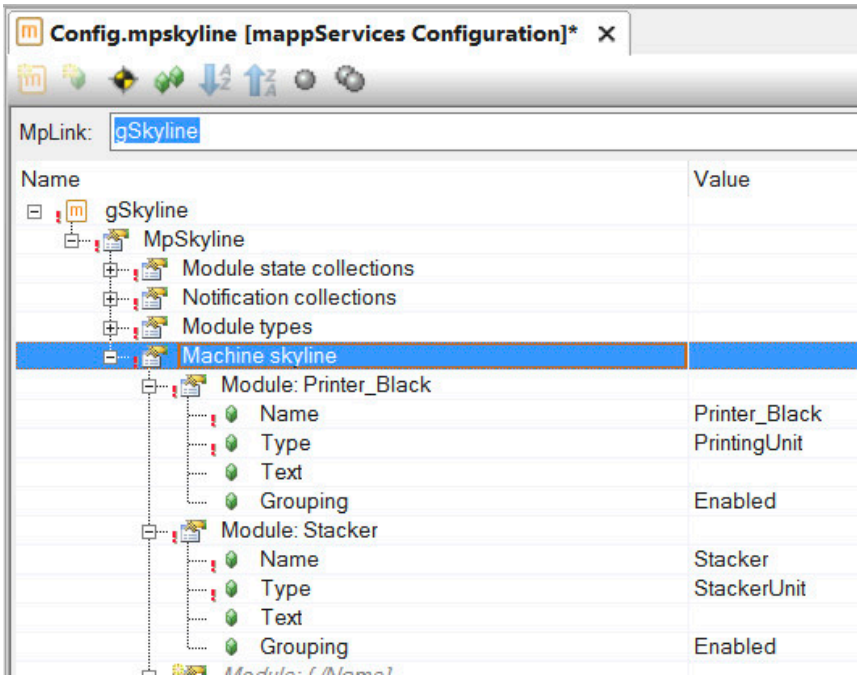
Danach werden die Benachrichtigungen unter "Notification collections" definiert. Dabei wird die Liste "PrintingProductionNotifications" angelegt, welche aus den Benachrichtigungen "Information", "Service" und "Warning" besteht.



Nun werden die unterschiedlichen Modul-Typen festgelegt. Es existiert der Typ "PrintingUnit" und "StackerUnit". PrintingUnit hat dabei die Größe L und StackerUnit M. Die Höhe und Breite dieser Größen wird später im Skyline-Widget festgelegt. Es wird die zuvor erstellte Benachrichtigung-Liste und Zustands-Modul mit dem Modul-Typen verknüpft.



Final wird die Maschinenlinie aus den einzelnen Modulen unter "Machine skyline" zusammengestellt. Es wird das Modul "Printer_black" vom Typen "PrintingUnit" und das Modul "Stacker" vom Typen "StackerUnit" definiert. Die Konfiguration ist nun abgeschlossen.



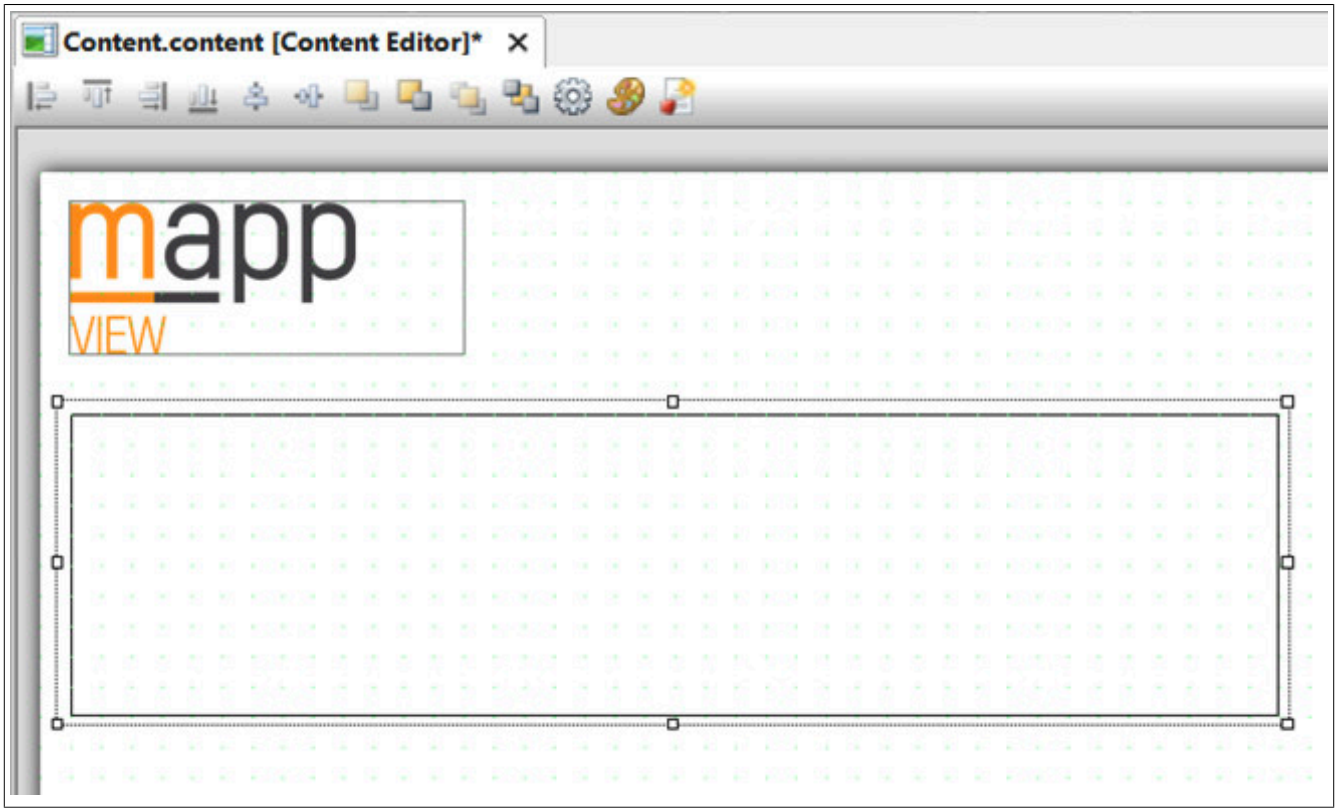
5.2.1.3 mapp View Visualisierung

Im nächsten Schritt wird eine mapp View Visualisierung hinzugefügt. Für mehr Information siehe hier.

Die Visualisierung kann folgendermaßen aussehen:

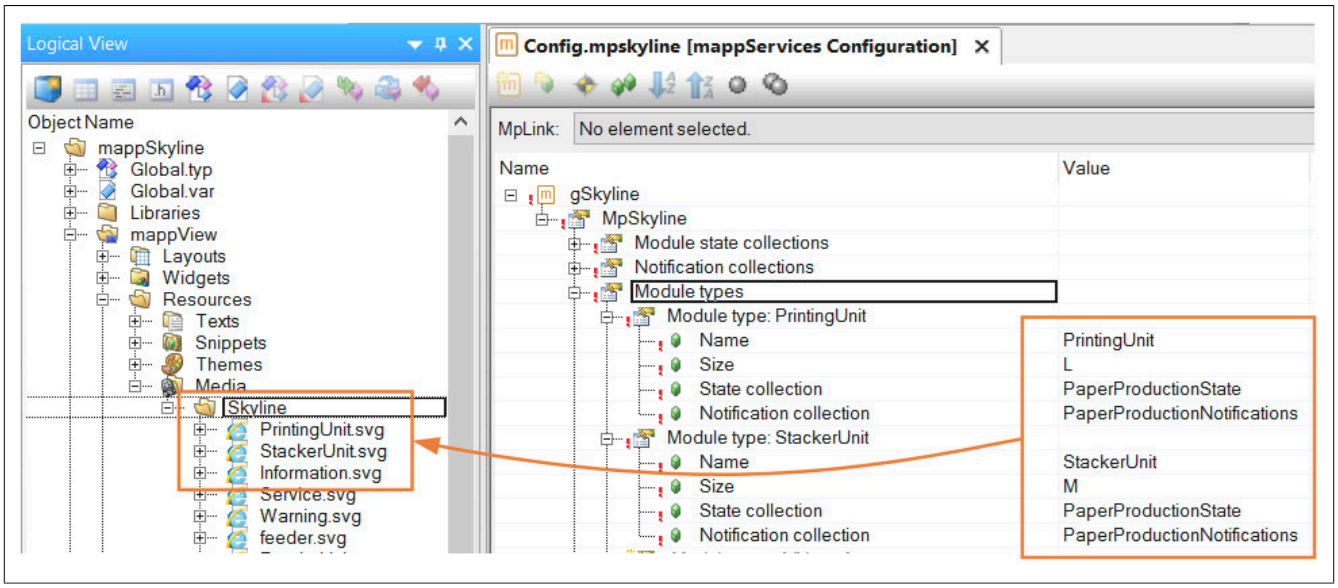
5.2.1.4 Widget einfügen und konfigurieren

Im nächsten Schritt wird in AS zum Content navigiert, welcher die Skyline anzeigen soll. Per Drag&Drop wird das Skyline-Widget eingefügt und auf dem Content platziert.



Um eine Verbindung zu mapp Skyline herzustellen, muss der mapp Link gSkyline an die Eigenschaft *mpLink* angebunden werden.

An der Eigenschaft *imagePrefix* wird der Pfad zu den Bildern, welche für die Modul-Typen verwendet werden sollen, angegeben. Der Bildname muss identisch mit dem Namen des Modul-Typen sein. Wichtig hierbei ist, dass nur SVG-Bilder unterstützt werden.



Über die Eigenschaften minXS, maxXS, minS, maxS etc. werden die minimalen und maximalen Größen für die Modul-Typen definiert.

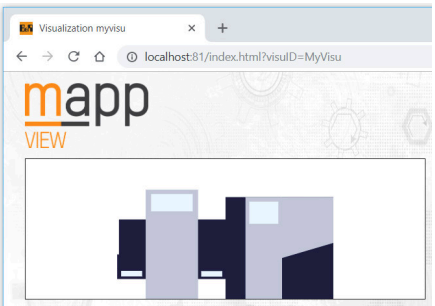
Zuletzt muss die Bibliothek MpServer eingefügt werden. Somit sind alle notwendigen Schritte, um die Skyline über die Visualisierung darzustellen, abgeschlossen.

5.2.1.5 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

5.2.1.6 Testen des Programms

Nach einem Download der Änderungen zur Steuerung kann die Visualisierung über den Webbrowser geöffnet werden. Die vorher konfigurierte Skyline ist nun in der Visualisierung sichtbar.



Die zuvor konfigurierten Zustände können mithilfe der Funktion MpSkylineSetModuleState aktiviert werden. Um eine Benachrichtigung für ein Modul zu aktivieren bzw. zu deaktivieren, wird die Funktion MpSkylineSetNotification bzw. MpSkylineResetNotification verwendet.

Wie die Skyline zur Laufzeit erweitert werden kann, wird im Anwendungsfall "Skyline zur Laufzeit erweitern" erklärt.

Auf unserem B&R Tutorial Portal wird über das Tutorial mapp Skyline Part 3 (GER/ENG) erklärt, wie zur Laufzeit die Skyline geändert werden kann. (Zugangsdaten)

5.3 Konfiguration

5.3.1 MpSkyline Konfiguration

> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**
>
> **Die geänderten Konfigurationsparameter werden sofort von der Komponente übernommen.**

In diesem Abschnitt werden Begrifflichkeiten verwendet, welche im Abschnitt Grundkonzept zu finden sind.

Zustände

Unter "Module state collections" können Zustands-Modelle für die Modul-Typen erstellt werden.

In einem Zustands-Modul werden die gewünschten Zustände definiert. Dabei kann aus einigen vordefinierten Zuständen gewählt, oder eigene Zustände hinzugefügt werden.

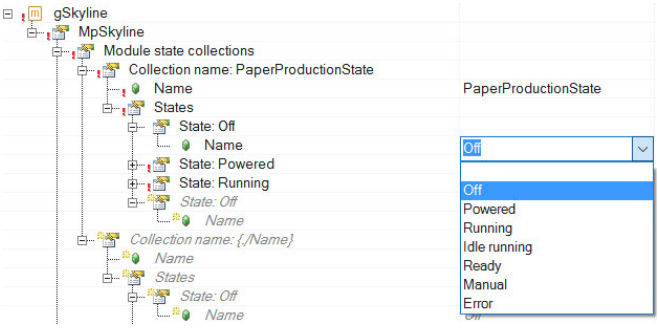


Abbildung 18: Vordefinierte Zustände

Abbildung 19: Selbstdefinierter Zustand

Wie eine Zustandsänderung im Skyline-Widget sichtbar wird, ist im Abschnitt Interaktion mit dem Widget erklärt.

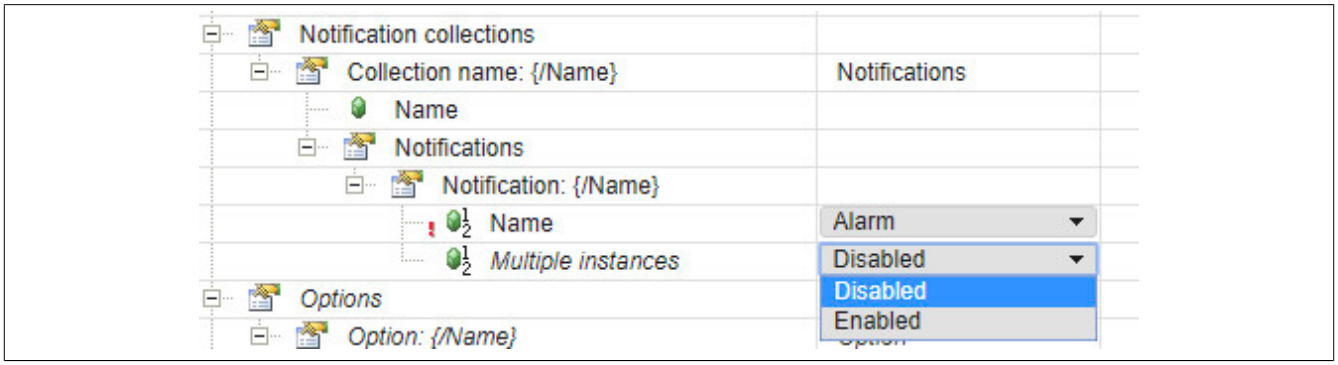
Benachrichtigungen

Unter "Notification collections" können Benachrichtigungs-Listen für die Modul-Typen angelegt werden.

In einer Benachrichtigungs-Liste werden die gewünschten Benachrichtigungen definiert. Dabei kann aus einigen vordefinierten gewählt, oder eigene Benachrichtigungen hinzugefügt werden.

Wie eine Benachrichtigung im Skyline-Widget angezeigt wird, ist im Abschnitt Interaktion mit dem Widget erklärt.

Werden die erweiterten Parameter in der Konfiguration aktiviert, so kann für eine Benachrichtigung festgelegt werden, ob diese einmalig oder mehrmals angezeigt werden kann.



Wird "Multiple instances" auf "Enabled" gesetzt, so kann die gleiche Benachrichtigung mithilfe MpSkylineSetNotification mehrmals gesetzt werden. Dies kann am Skyline-Widget folgendermaßen aussehen:

Modul-Typen

Unter "Module types" werden die unterschiedlichen Modul-Typen definiert. Dabei muss unter "Name" ein eindeutiger Name für den Typ festgelegt werden. Wie groß der Modul-Typ im Skyline-Widget angezeigt werden soll, wird über "Size" definiert. Dabei kann zwischen XS, S, M oder L gewählt werden. Die Größe dieser Parameter wird dabei über die Eigenschaften "minXS", "maxXS", "minM", "maxM" usw. am Skyline-Widget definiert.

Welches Zustands-Modul und welche Benachrichtigungs-Liste für den Modul-Typen verwendet werden soll, wird über "State collection" und "Notification collection" angegeben.

Wie ein Modul-Typ im Skyline-Widget angezeigt wird, ist im Abschnitt Interaktion mit dem Widget erklärt.

Skyline

Unter "Machine skyline" wird definiert, wie die Skyline aussehen soll. Dabei werden der Reihe nach die gewünschten Module angegeben.

Pro Modul muss ein eindeutiger Name definiert werden. Unter "Type" wird der gewünschte Modul-Typ ausgewählt. Unter "Text" kann ein statischer oder auch ein Text lokalisiert über das Textsystem angegeben werden. Dieser Text wird am Skyline-Widget angezeigt. Wie der Text im Modul positioniert und das Aussehen definiert werden kann ist im Abschnitt Interaktion mit dem Widget bzw. in Positionierung von Texten und Symbolen erklärt.

Pro Modul ist es auch möglich ein Modul-Symbol anzeigen zu lassen. Dies wird am Skyline-Widget definiert. Für mehr Information siehe hier.

Mithilfe von "Grouping" kann bestimmt werden ob Module, welche vom gleichen Typen sind, für eine bessere Anzeige am Skyline-Widget gruppiert werden können oder nicht. Dies kann zum Beispiel folgendermaßen aussehen:

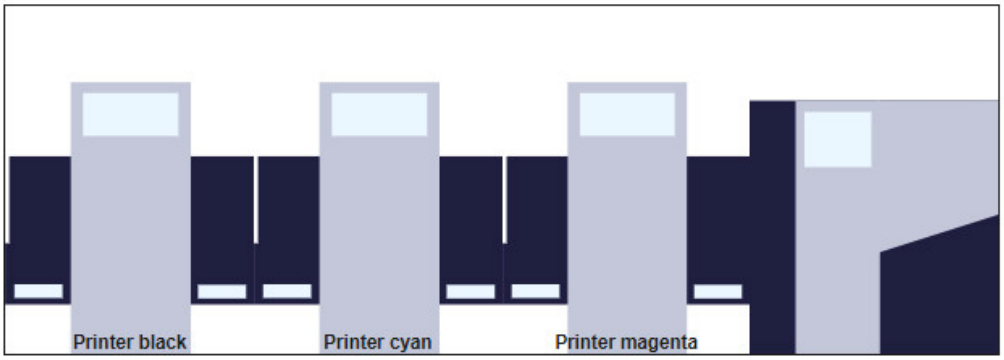


Abbildung 20: Skyline ohne Gruppierung der Module

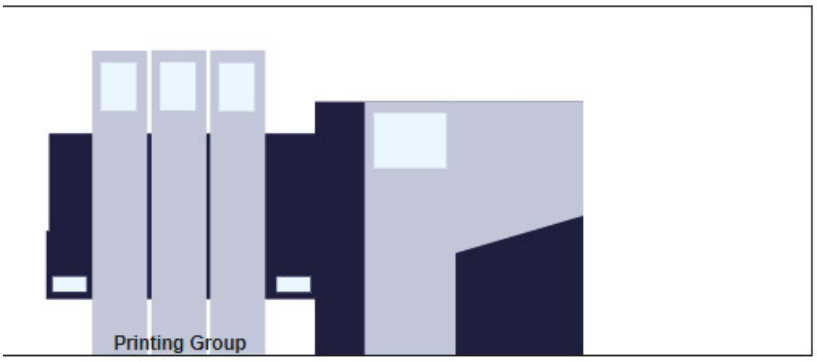
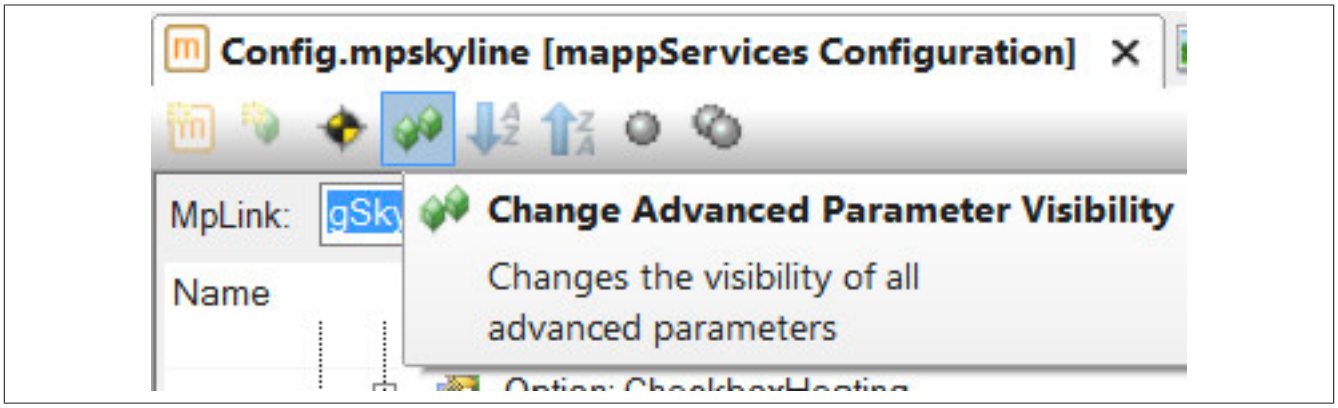


Abbildung 21: Skyline mit Gruppierung der Module

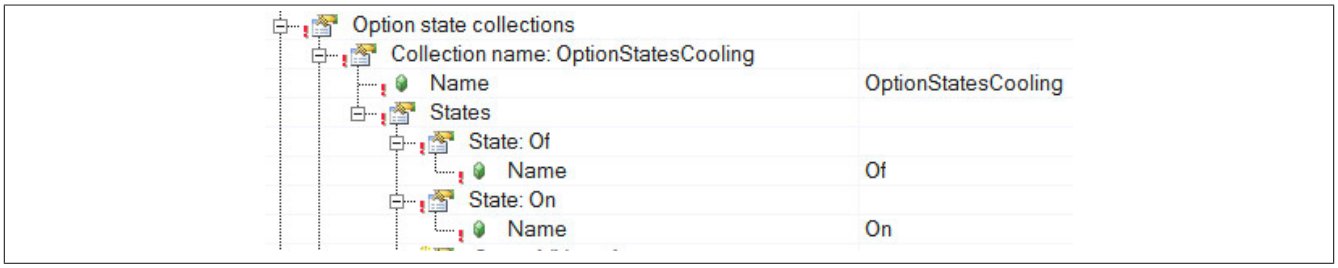
Für mehr Information dazu siehe Abschnitt "Positionieren der Module / Gruppieren" in Interaktion mit dem Widget.

Optionen

Um Optionen für Modul-Typen zu erstellen, müssen die erweiterten Eigenschaften in der Konfiguration aktiviert sein.

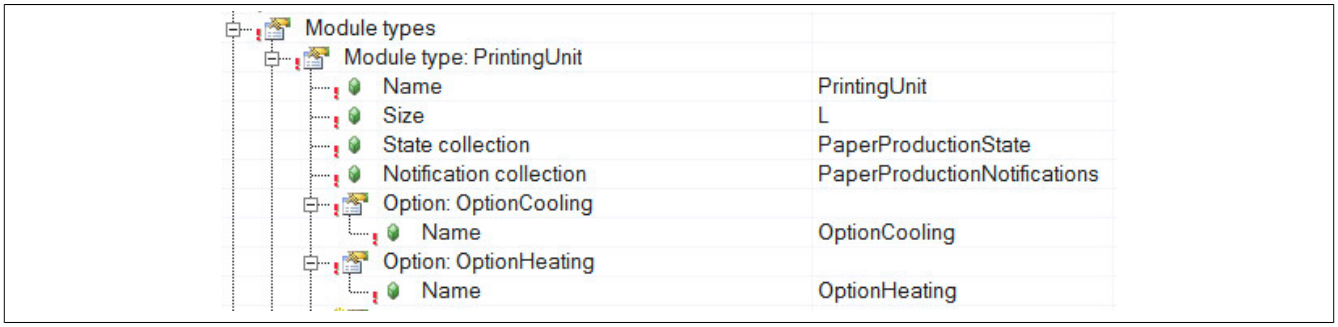


Zuerst muss für die Option eine Zustands-Modul unter "Option state collections" erstellt werden. Dies funktioniert nach dem gleichen Prinzip wie unter "Zustände" erklärt wurde. Das bedeutet, eine Option kann verschiedene Zustände annehmen. Zum Beispiel kann die Option "Cooling" den Zustand "On" oder "Off" besitzen.



Die Option selbst wird unter "Options" definiert. Dabei muss der Option ein eindeutiger Name unter "Names" angegeben werden. Unter "State collection" wird das gewünschte Zustands-Modul angegeben, welche zuvor unter "Option state collections" erstellt wurde.

Zum Schluss wird die Option einem Modul-Typen zugewiesen. Es können auch mehrere verschiedene Optionen angegeben werden.



5.4 Anwendungsfälle

5.4.1 Skyline zur Laufzeit erweitern

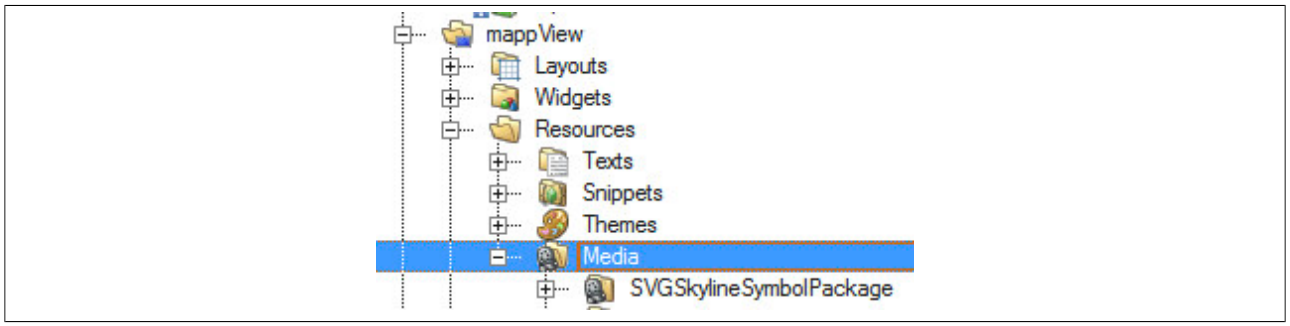
Anforderung

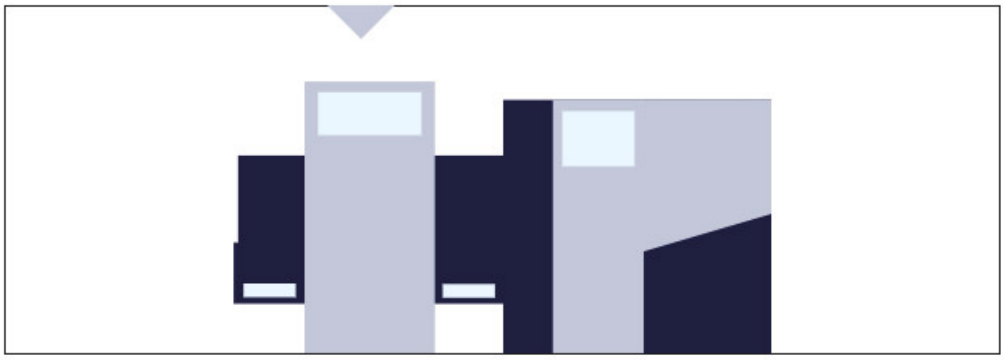
Eine bereits bestehende Drucker-Maschinenlinie soll erweitert werden. Aktuell besteht die Maschinenline aus einem Printing-Modul und einem Stacker-Modul. Nun soll ein weiteres Printing-Modul ("Printer cyan") zur Laufzeit hinzugefügt werden.

Die in der Visualisierung angzeigte Skyline soll ebenfalls aktualisiert werden, damit alle bestehenden Module vorhanden sind.

Auf unserem B&R Tutorial Portal wird über das Tutorial mapp Skyline Part 3 (GER/ENG) erklärt, wie zur Laufzeit die Skyline geändert werden kann. (Zugangsdaten)

In der von mapp View bereitsgestellten SVG Skyline Bibliothek sind Bilder enthalten, welche für die Skyline verwendet werden können. Die Bibliothek ist über die Toolbox verfügbar und kann im mapp View Paket in der Logical View unter Resources -> Media abgelegt werden.





Vorbereiten der Konfiguration

Als erstes muss die bereits vorhandene Konfiguration durch den Funktionsbaustein MpComConfigManager exportiert werden. Die Datei wird dabei auf einem Speichermedium, wie zum Beispiel einen USB-Stick, abgespeichert. Die exportierte Datei kann anschließend mithilfe eines Texteditors bearbeitet werden.

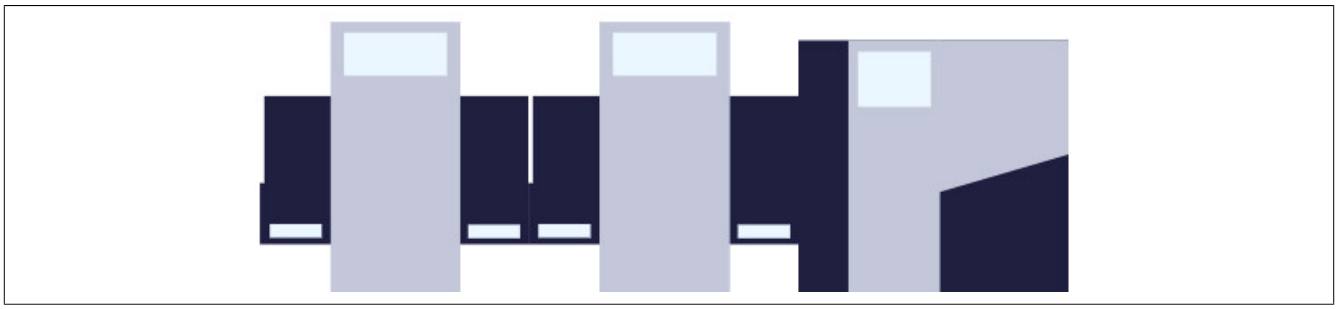
Es wird ein zustätzliches Modul "Printer_Cyan" unter "Modules" eingefügt.

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
    <Version ID="1" />
    <Element ID="gSkyline">
        <Group ID="MpSkyline">
            <Group ID="ModuleStateCollections">
            <Group ID="ModuleTypes">
            <Group ID="Modules">
                <Group ID="[0]">
                    <Property ID="Grouping" DataType="DINT" Value="1" />
                    <Property ID="Name" DataType="STRING" Value="Printer_Black" />
                    <Property ID="Text" DataType="STRING" Value="" />
                    <Property ID="Type" DataType="STRING" Value="PrintingUnit" />
                </Group>
                <Group ID="[1]">
                    <Property ID="Grouping" DataType="DINT" Value="1" />
                    <Property ID="Name" DataType="STRING" Value="Printer_Cyan" />
                    <Property ID="Text" DataType="STRING" Value="" />
                    <Property ID="Type" DataType="STRING" Value="PrintingUnit" />
                </Group>
                <Group ID="[2]">
                    <Property ID="Grouping" DataType="DINT" Value="1" />
                    <Property ID="Name" DataType="STRING" Value="Stacker" />
                    <Property ID="Text" DataType="STRING" Value="" />
                    <Property ID="Type" DataType="STRING" Value="StackerUnit" />
                </Group>
            </Group>
            <Group ID="NotificationCollections">
            <Group ID="OptionStateCollections">
```

Die Group-ID muss entsprechend angepasst werden. Jedes Modul braucht eine eigene ID, welche nicht mehrfach vorkommen darf. Die Konfiguration wird gespeichert und kann verwendet werden.

Konfiguration zur Laufzeit aktualisieren

Über den Funktionsbaustein MpComConfigManager wird die zuvor bearbeitete Konfiguration importiert. Nachdem die Visualisierung aktualisiert wurde, wird die neue Skyline angezeigt.



5.5 Diagnose

Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

5.5.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp Skyline auftreten können.

Allgemein

Alle Ereignisse und Informationen werden im Logger aufgelistet. Tritt ein Fehler in der Kommunikation zwischen dem Skyline-Widget und mapp Skyline auf, so kann die Fehlernummer auch über die Aktion "OnError" ausgelesen werden.

Die Skyline wird im Webbrowser nicht angezeigt

Erscheint beim Aufrufen der Visualisierung im Webbrowser ein leeres Skyline-Widget, so kann dies folgende Möglichkeiten haben:

- An der Eigenschaft *mpLink* im Skyline-Widget wurde nicht der mapp Link einer MpSkyline Konfiguration angegeben.
- Die Bibliothek MpServer wurde nicht auf das Zielsystem übertragen. Für mehr Information dazu siehe mapp Kommunikation.
- Es wurden keine Größen für die Eigenschaften *minXS*, *maxXS*, *minS*, *maxS* usw. im Skyline-Widget definiert.
- An der Eigenschaft *pointerImage* im Skyline-Widget muss ein SVG-Bild angegeben werden.

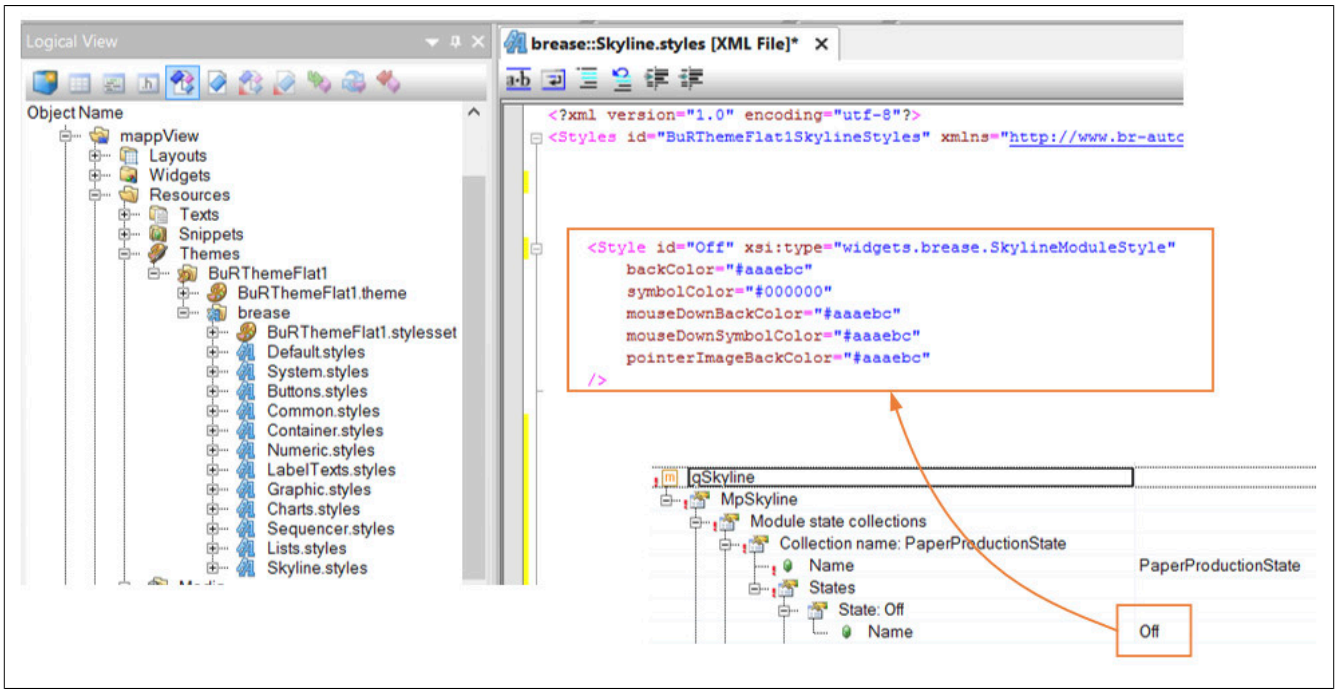
Styleänderungen werden in der Skyline nicht angezeigt

Mithilfe des SkylineModuleStyle-Widget und SkylineNotificationStyle-Widget kann definiert werden, wie das Modul und das Benachrichtigungs-Symbol farblich angezeigt werden soll. Wie diese verwendet werden, ist im Abschnitt Interaktion mit dem Widget erklärt.

Werden die angegebenen Styleänderungen nicht übernommen, so kann dies zwei Ursachen haben:

Style-ID stimmt nicht mit dem Namen des Zustands oder mit der Benachrichtigungs-ID überein

Bei der Verwendung des SkylineModuleStyle-Widget und SkylineNotificationStyle-Widget muss die Style-ID mit dem gewünschten Zustand bzw. der Benachrichtigungs-ID übereinstimmen. Diese sind in der MpSkyline Konfiguration zu finden.



SVG-Image enthält bereits das "fill" Attribut

Bei der Verwendung von SVG-Images in mapp View Visualisierungen gilt zu beachten, dass nahezu jedes SVG-Element eine Farbe annehmen kann, indem das zugehörige "fill" Attribut gesetzt wird. Soll nun über einen Style die Farbe angepasst werden, so darf das SVG-Element das "fill" Attribut nicht besitzen.

Für mehr Information zu SVG-Images in mapp View Visualisierungen siehe hier.

Meine zweite Skyline wird nicht angezeigt

Aktuell ist es nicht möglich mehere Skylines zu konfigurieren. Pro Zielsystem kann eine MpSkyline Konfiguration verwendet werden und dadurch nur eine Skyline angezeigt werden.

5.6 Bibliotheken

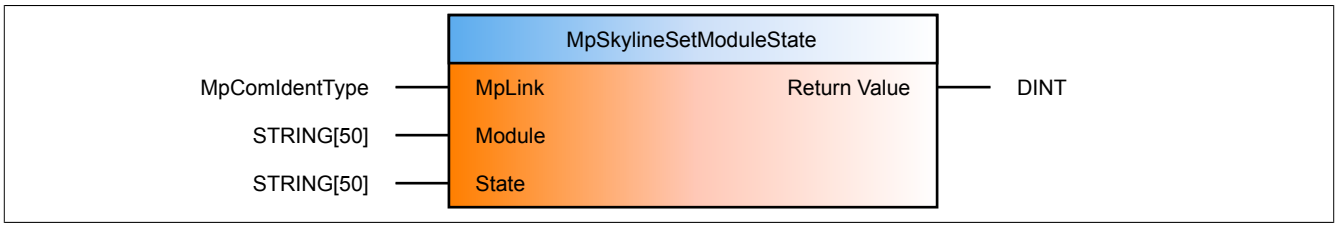
5.6.1 Funktionsbausteine

Untersützt ab	Name	Beschreibung
5.6.0	MpSkylineSetModuleState	Durch MpSkylineSetModuleState kann ein Zustand für ein Modul festgelegt werden.
5.6.0	MpSkylineSetNotification	Mithilfe von MpSkylineSetNotification kann eine Benachrichtigung für ein Modul aktiviert werden.
5.6.0	MpSkylineResetNotification	Mithilfe dieser Funktion kann eine Benachrichtigung deaktiviert werden.
5.6.0	MpSkylineSetOptionState	Mithilfe von MpSkylineSetOptionState kann ein Options-Zustand geändert werden.
5.6.0	MpSkylineCheckOptionState	Mit MpSkylineCheckOptionState kann der aktuelle Zustand für eine Option überprüft werden.

5.6.1.1 MpSkylineSetModuleState

Durch MpSkylineSetModuleState kann ein Zustand für ein Modul festgelegt werden.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp (MpLink einer MpSkyline Konfiguration)
IN	Module	`STRING[50]`	Angabe des Modulnamens
IN	State	`STRING[50]`	Angabe des Zustands
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt Design der Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

5.6.1.1.1 Beschreibung

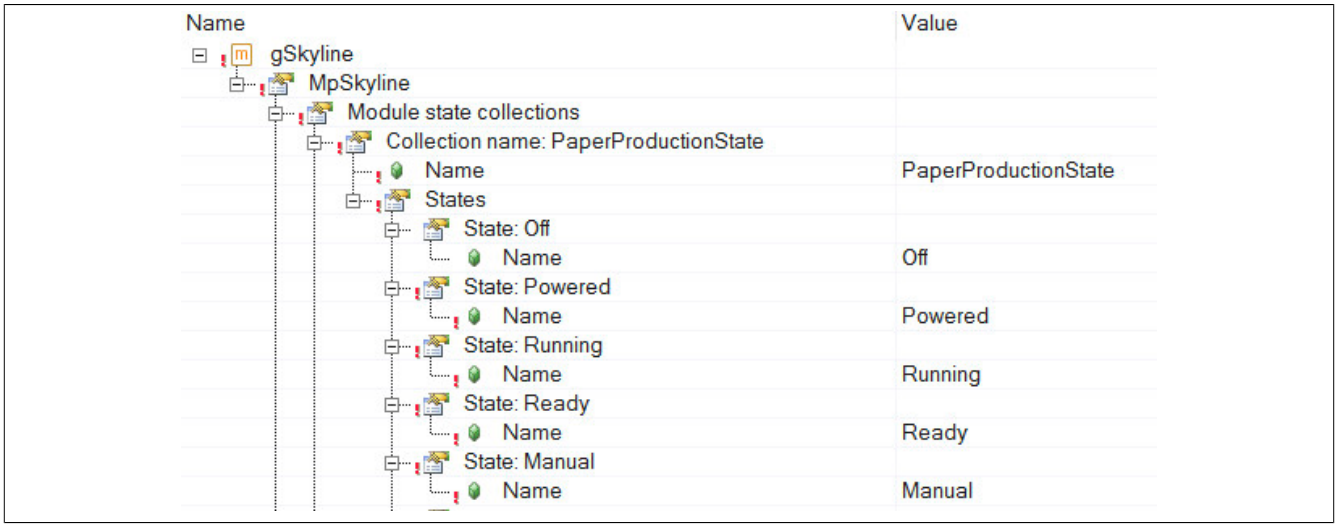
MpSkylineSetModuleState verwendet den MpLink einer MpSkyline Konfiguration.

Mithilfe der Funktion kann ein Zustand eines Moduls aktiviert werden.

Dazu wird am Übergabeparameter "Module" der gewünschte Modulname und an "State" der zu aktivierende Zustand definiert.

Der Modulname kann in der MpSkyline Konfiguration im Abschnitt "Machine skyline" entnommen werden.

Welche Zustände aktiviert werden können, ist in der MpSkyline Konfiguration im Abschnitt "Module state collections" definiert.



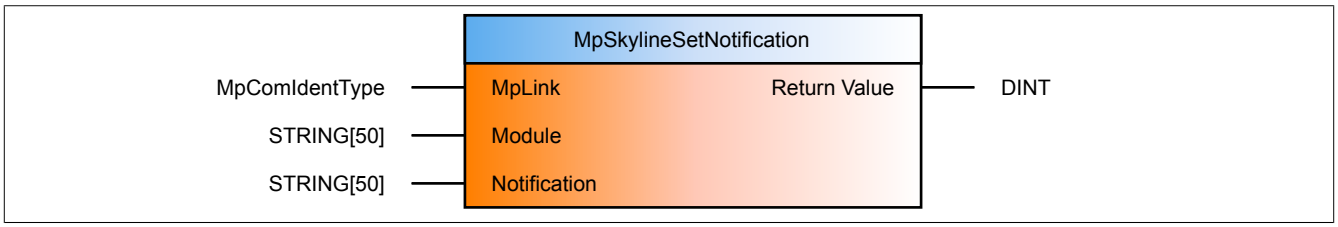
Standardmäßig ist der erste festgelegte Zustand in der Konfiguration aktiv. Es kann immer nur ein Zustand in einem Modul aktiv sein. Das heißt sobald ein neuer Zustand mithilfe der Funktion aktiviert wird, wird der vorhergehende Zustand ersetzt.

"Return value" liefert den Rückgabewert der Funktion. Bei "0" wurde die Funktion erfolgreich ausgeführt. Kann die Funktion nicht ausgführt werden, wird eine Fehlernummer angzeigt. Die Fehlernummer, sowie deren Ursache und Abhilfe, kann hier gefunden werden.

5.6.1.2 MpSkylineSetNotification

Mithilfe von MpSkylineSetNotification kann eine Benachrichtigung für ein Modul aktiviert werden.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp (MpLink einer MpSkyline Konfiguration)
IN	Module	`STRING[50]`	Angabe des Modulnamens
IN	Notification	`STRING[50]`	Name der Benachrichtigung
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt Design der Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

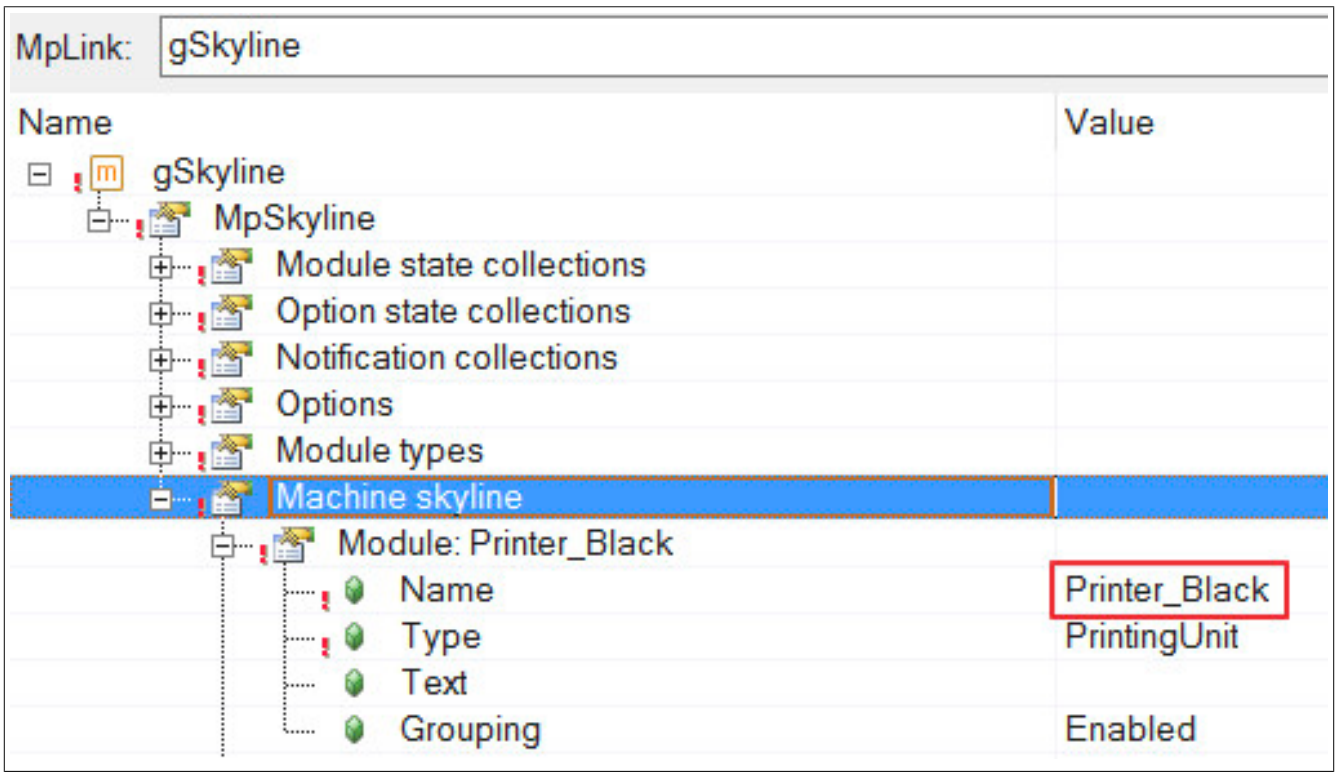
5.6.1.2.1 Beschreibung

Die Funktion verwendet den MpLink einer MpSkyline Konfiguration.

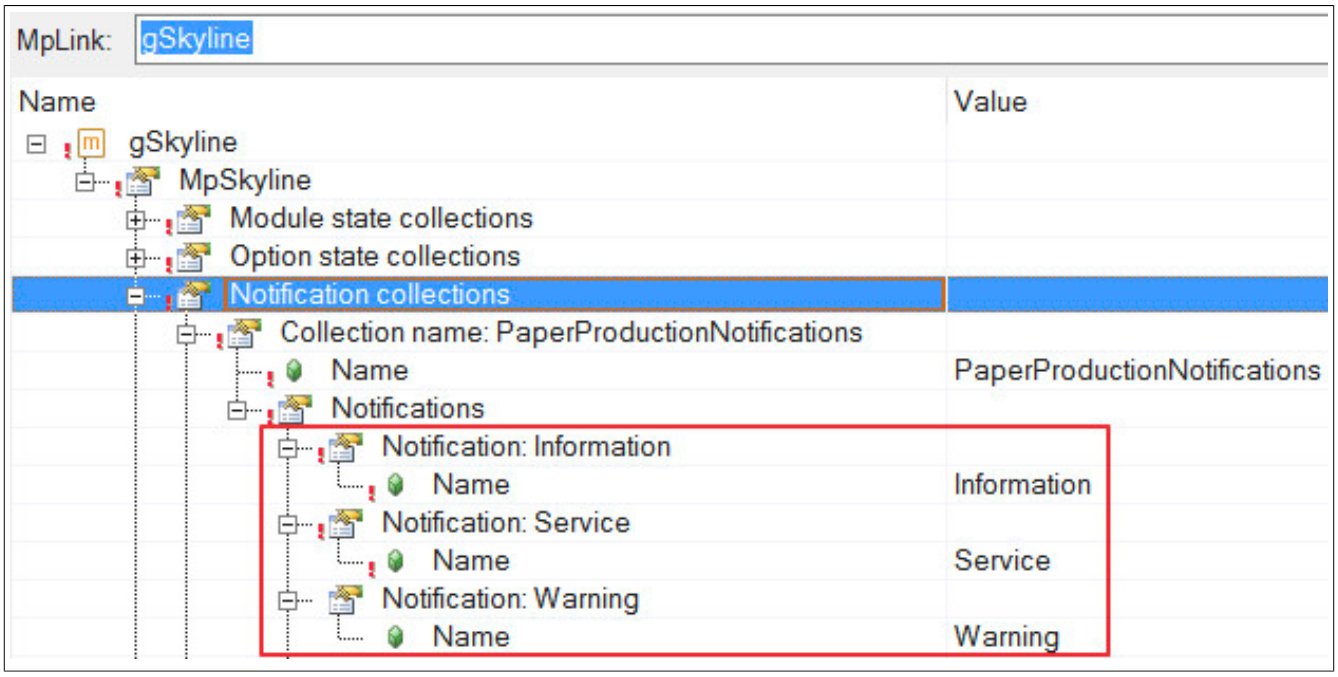
Mithilfe von MpSkylineSetNotification kann eine Benachrichtigung in einem Modul aktiviert werden.

Aktivieren einer Benachrichtigung

Dabei muss der gewünschte Modulname am Übergabeparameter "Module" angegeben werden. Dies ist jener Name, welche in der MpSkyline Konfiguration unter "Machine skyline" festgelegt wurde.



An "Notification" wird der Name der gewünschte Benachrichtigung angegeben. Diese wird in der MpSkyline Konfiguration im Abschnitt "Notification collection" definiert.



Sobald die Funktion aufgerufen wird, wird die Benachrichtigung im Modul aktiviert. Die gleiche Benachrichtigung, wie zum Beispiel "Information", kann nur einmal pro Modul aktiviert werden.

"Return value" liefert den Rückgabewert der Funktion. Bei "0" wurde die Funktion erfolgreich ausgeführt. Kann die Funktion nicht ausgführt werden, wird eine Fehlernummer angzeigt. Die Fehlernummer, sowie deren Ursache und Abhilfe, kann hier gefunden werden.

Mehrere Benachrichtigungen für ein Modul aktivieren

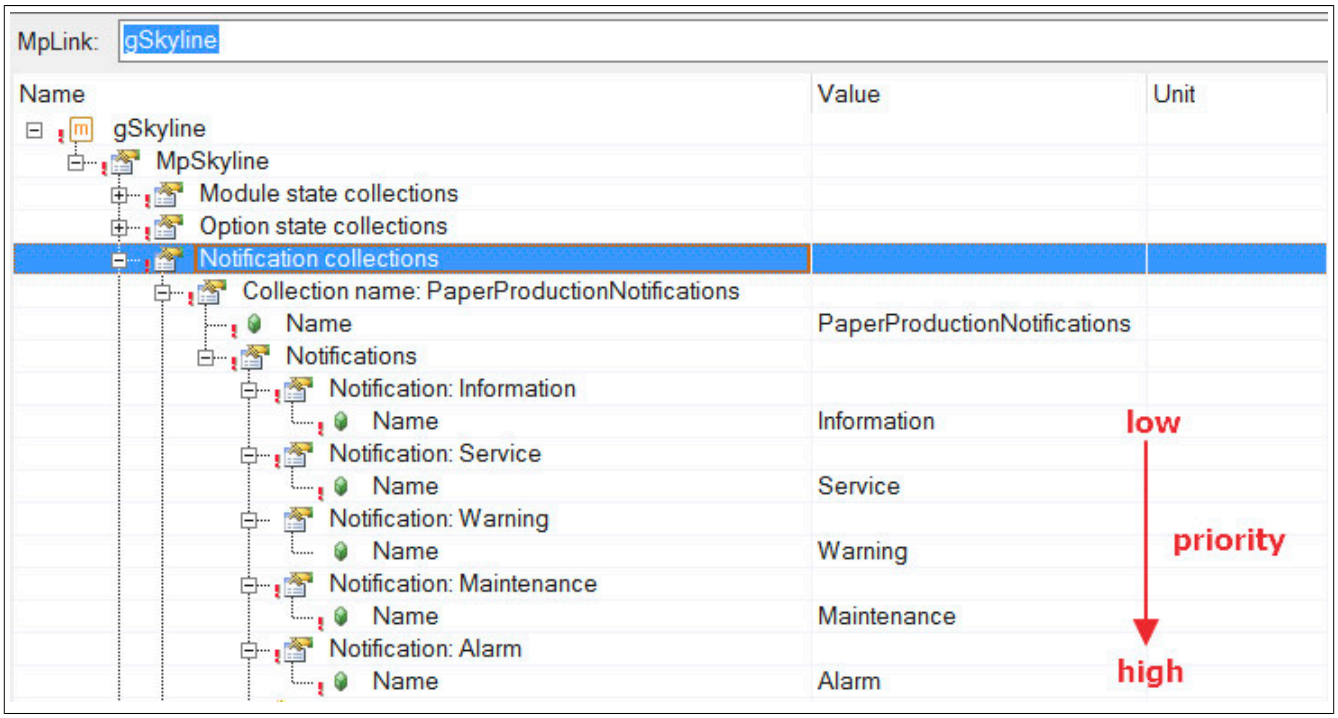
Für ein Modul können mehrere Benachrichtigungen aktiviert werden. Dabei muss nur ein neuer Benachrichtigungs-Name an "Notification" angegeben und die Funktion aufgerufen werden. Wie viele Benachrichtigungen aktuell für ein Modul aktiv sind, wird am Skyline-Widget angezeigt.

Abbildung 22: Eine aktive Benachrichtigung

Abbildung 23: Zwei aktive Benachrichtigungen

Welches Symbol dabei für welche Benachrichtigung verwendet wird, ist im Abschnitt Interaktion mit dem Widget zu finden. Das bedeutet pro Benachrichtigung kann ein eigenes Symbol angezeigt werden.

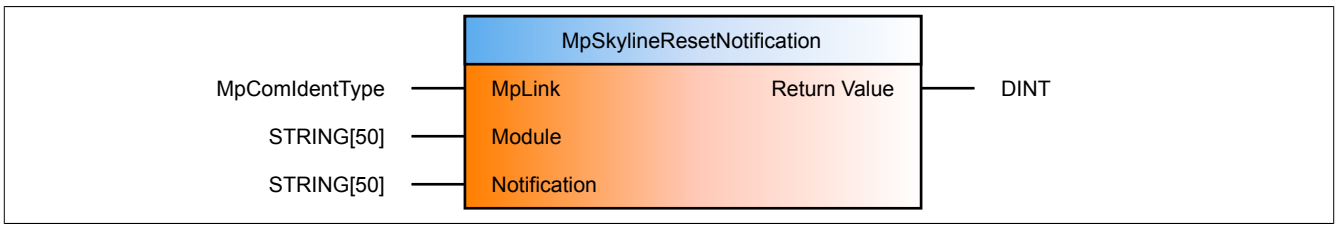
Welches Benachrichtigungs-Symbol dabei zuerst angezeigt wird, hängt von der Priorität der Benachrichtigung ab. Diese wird in der Konfiguration bestimmt. Die erste Benachrichtigung in der Liste hat die niedrigste und die letzte die höchste Priorität.



5.6.1.3 MpSkylineResetNotification

Mithilfe dieser Funktion kann eine Benachrichtigung deaktiviert werden.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp (MpLink einer MpSkyline Konfiguration)
IN	Module	`STRING[50]`	Angabe des Modulnamens
IN	Notification	`STRING[50]`	Name der Benachrichtigung
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt Design der Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

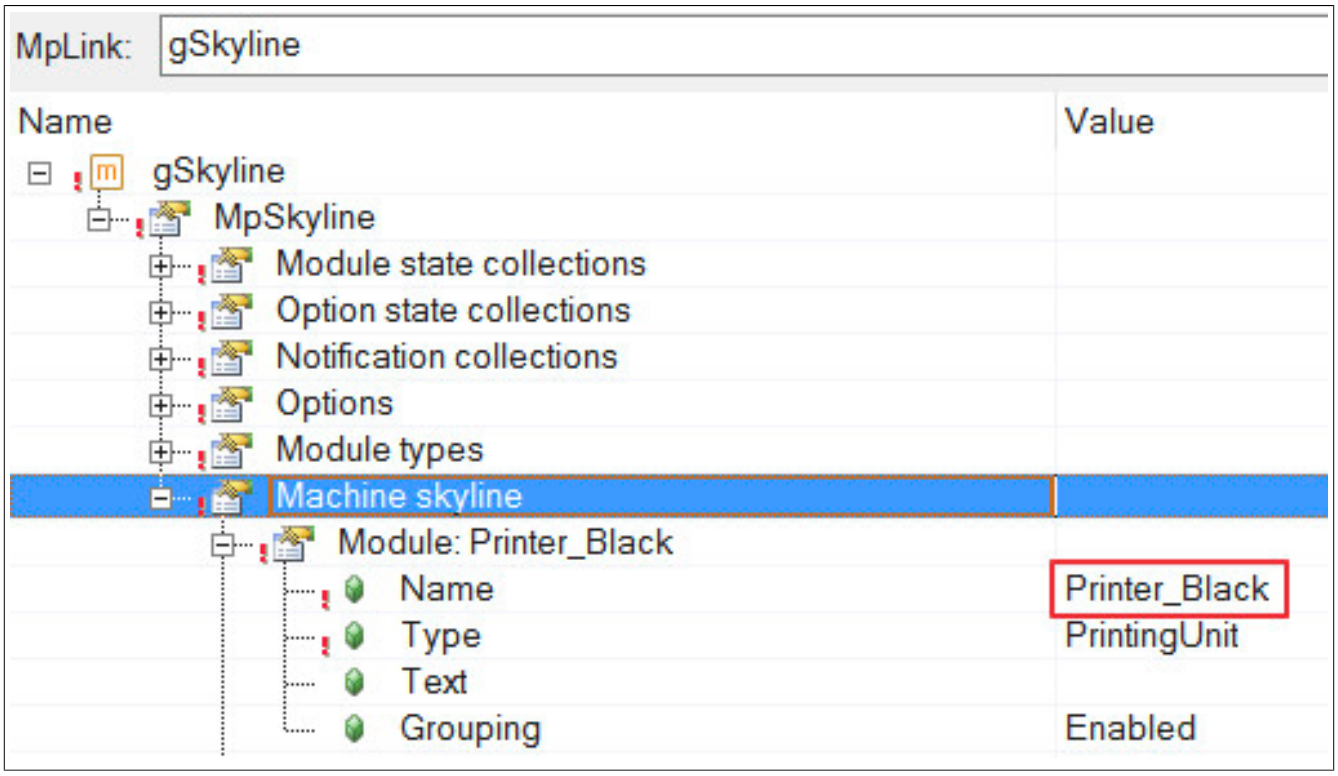
5.6.1.3.1 Beschreibung

Diese Funktion verwendet den MpLink einer MpSkyline Konfiguration.

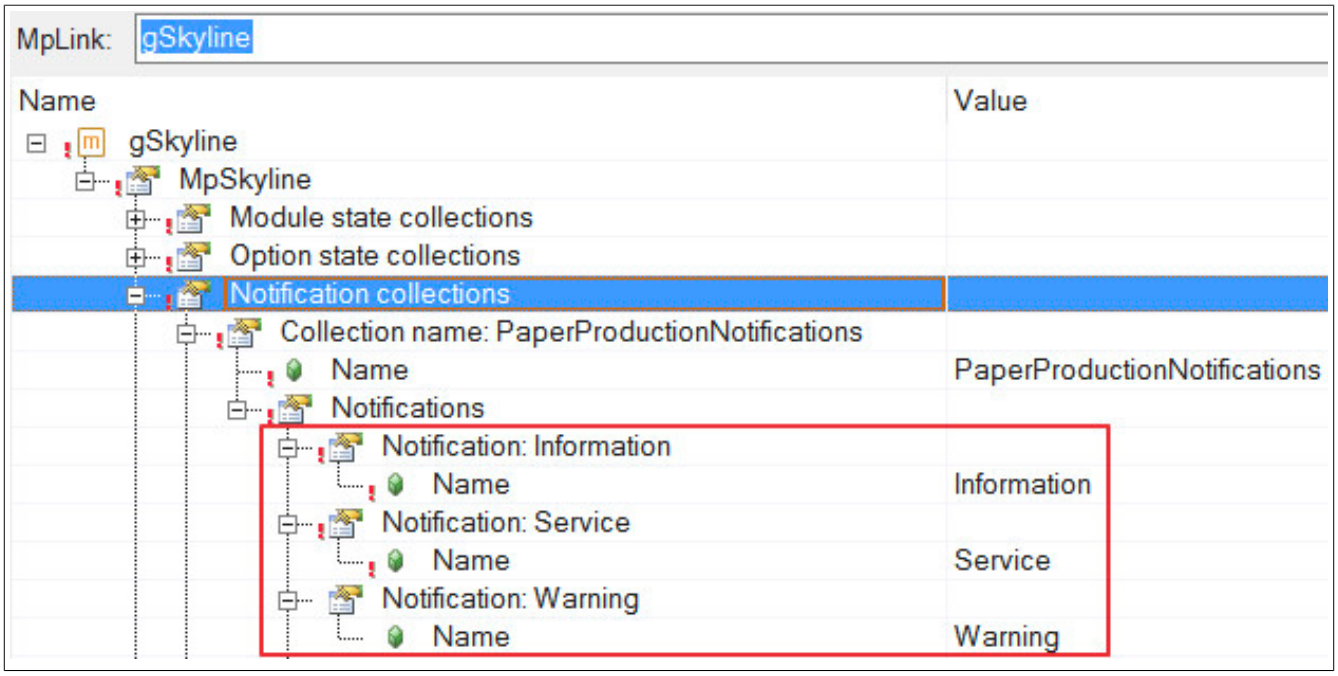
Mithilfe von MpSkylineResetNotification können Benachrichtigungen in einem Modul, welche zuvor mit MpSkylineSetNotification aktiviert wurden, deaktiviert werden.

Deaktivieren einer Benachrichtigung

Dabei muss der gewünschte Modulname am Übergabeparameter "Module" angegeben werden. Dies ist jener Name, welche in der MpSkyline Konfiguration unter "Machine skyline" festgelegt wurde.



An "Notification" wird der Name der gewünschte Benachrichtigung angegeben. Diese wird in der MpSkyline Konfiguration im Abschnitt "Notification collection" definiert.



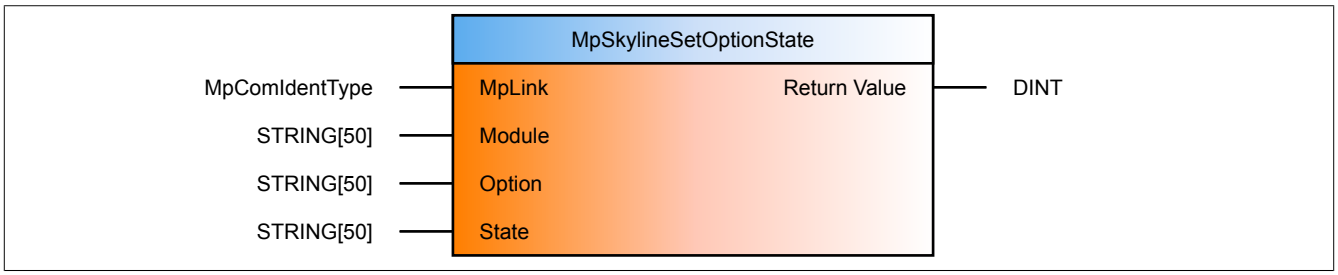
Sobald die Funktion aufgerufen wird, wird die gewünschte Benachrichtigung deaktiviert.

"Return value" liefert den Rückgabewert der Funktion. Bei "0" wurde die Funktion erfolgreich ausgeführt. Kann die Funktion nicht ausgführt werden, wird eine Fehlernummer angzeigt. Die Fehlernummer, sowie deren Ursache und Abhilfe, kann hier gefunden werden.

5.6.1.4 MpSkylineSetOptionState

Mithilfe von MpSkylineSetOptionState kann ein Options-Zustand geändert werden.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp (MpLink einer MpSkyline Konfiguration)
IN	Module	`STRING[50]`	Angabe des Modulnamens
IN	Option	`STRING[50]`	Angabe der Options-ID
IN	State	`STRING[50]`	Angabe des Zustands
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt Design der Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

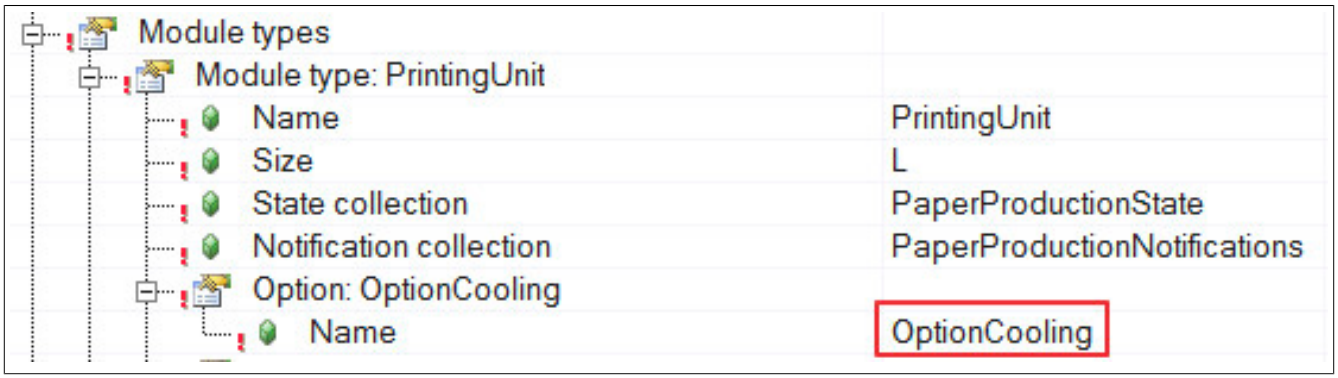
5.6.1.4.1 Beschreibung

Diese Funktion verwendet den MpLink einer MpSkyline Konfiguration.

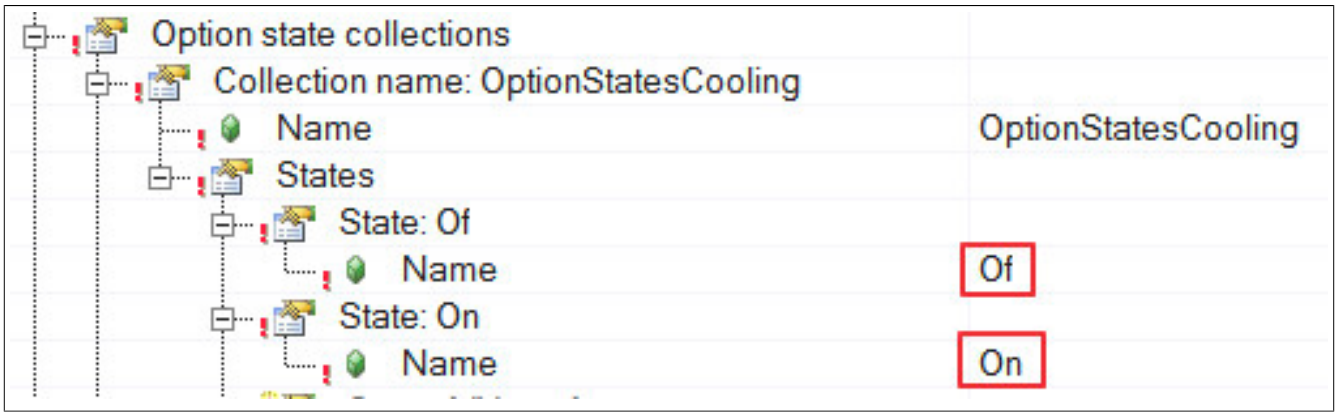
Mit MpSkylineSetOptionstate kann ein Zustand für eine Option gesetzt werden.

An "Modul" muss der gewünschte Modulname angegeben werden. Dieser ist in der MpSkyline Konfiguration unter "Machine skyline" zu finden.

"Option" definiert die Option welche für das angegebene Modul definiert wurde. Diese ist in der MpSkyline Konfiguration unter "Module types" zu finden.



"State" definiert den Zustand der für eine Option aktiviert werden soll. Die Zustände für eine Option werden in der MpSkyline Konfiguration unter "Option state collections" angelegt.



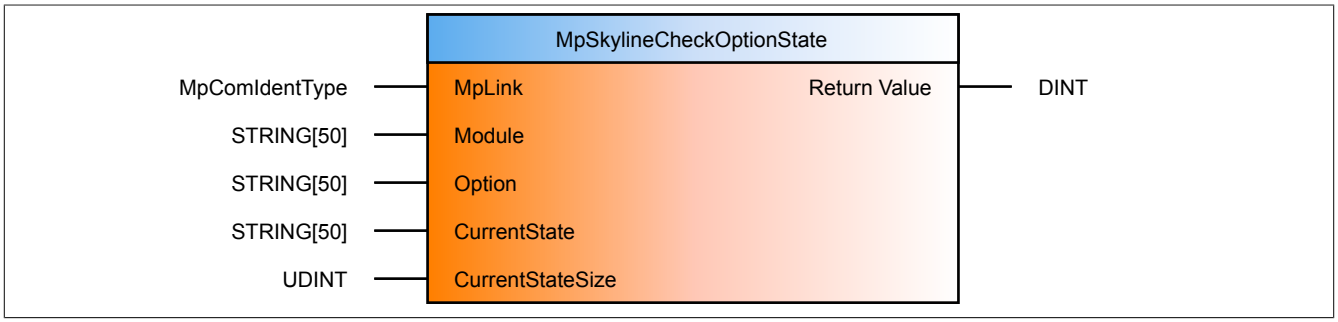
Standardmäßig ist immer der erste Zustand in der Options-Zustands-Liste aktiv.

"Return value" liefert den Rückgabewert der Funktion. Bei "0" wurde die Funktion erfolgreich ausgeführt. Kann die Funktion nicht ausgführt werden, wird eine Fehlernummer angzeigt. Die Fehlernummer, sowie deren Ursache und Abhilfe, kann hier gefunden werden.

5.6.1.5 MpSkylineCheckOptionState

Mit MpSkylineCheckOptionState kann der aktuelle Zustand für eine Option überprüft werden.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp (MpLink einer MpSkyline Konfiguration)
IN	Module	`STRING[50]`	Angabe des Modulnamens
IN	Option	`STRING[50]`	Angabe der Options-ID
IN	CurrentState	`STRING[50]`	Angabe einer Variable, welche nach ausführen der Funktion die Options-Zustands-ID enthält
IN	CurrentStateSize	`UDINT`	Definiert die Größe der "State" Variable.
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt Design der Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

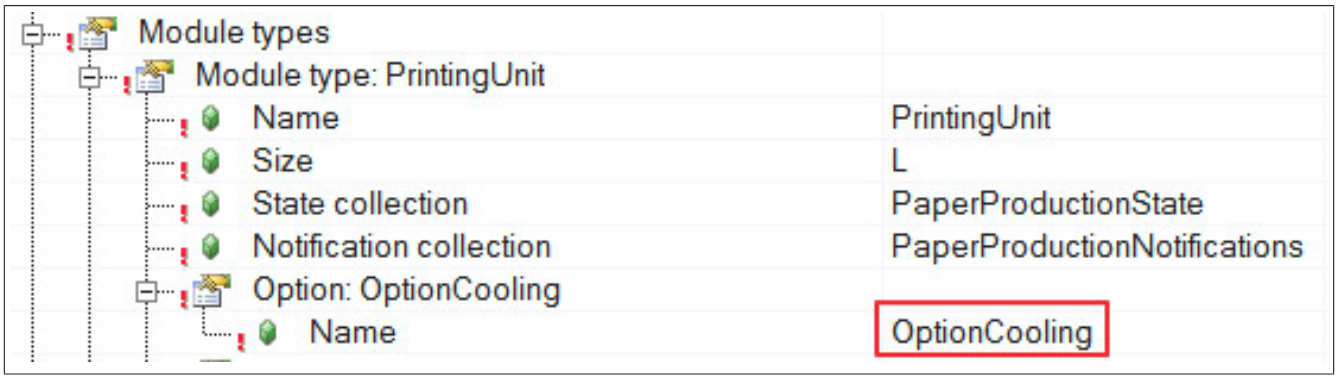
5.6.1.5.1 Beschreibung

Diese Funktion verwendet den MpLink einer MpSkyline Konfiguration.

Mit MpSkylineCheckOptionState kann der aktuell aktive Zustand für eine Option überprüft werden.

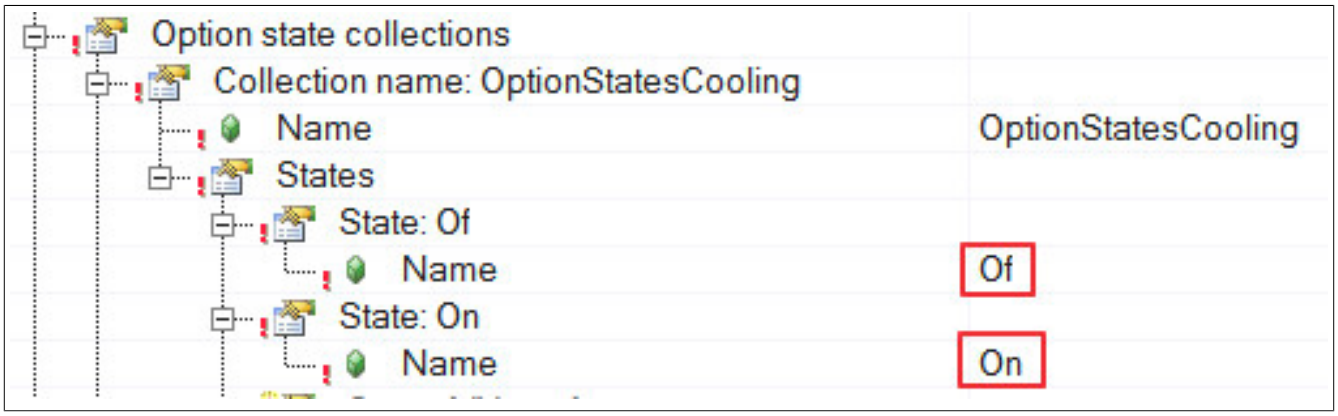
An "Modul" muss der gewünschte Modulname angegeben werden. Dieser ist in der MpSkyline Konfiguration unter "Machine skyline" zu finden.

"Option" definiert die Option welche für das angegebene Modul definiert wurde. Diese ist in der MpSkyline Konfiguration unter "Module types" zu finden.



Die Variable, welche an "CurrentState" angegeben wird, enthält nach Ausführen der Funktion, den aktuell aktiven Zustand der ausgewählten Option. Die Größe der Variable wird dabei an "CurrentStateSize" angegeben.

Die Zustände für eine Option werden in der MpSkyline Konfiguration unter "Option state collections" angelegt.



Standardmäßig ist immer der erste Zustand in der Options-Zustands-Liste aktiv.

"Return value" liefert den Rückgabewert der Funktion. Bei "0" wurde die Funktion erfolgreich ausgeführt. Kann die Funktion nicht ausgführt werden, wird eine Fehlernummer angzeigt. Die Fehlernummer, sowie deren Ursache und Abhilfe, kann hier gefunden werden.

5.6.2 Statusnummern

5.6.2.1 -1064126464: Zustand in der Option nicht verfügbar

Beschreibung:

Der Options-Zustand {2:Option state} existiert nicht in der Options-Zustands-Liste {3:Option state collection}

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Zusätzliche Information

- {2:Option state}: Name des Options-Zustands
- {3:Option state collection}: Name der Options-Zustands-Liste

Ursache / Abhilfe:

- Kontrollieren, ob alle Übergabeparameter richtig an MpSkylineSetOptionState angegeben wurden
- Überprüfen der Options-Zustände und Listen in der MpSkyline Konfiguration

Konstante

`mpSKYLINE_ERR_OPTION_STATE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpSkylineSetOptionState

5.6.2.2 -1064126463: Option für Modul nicht verfügbar

Beschreibung:

Die Option {2:Option} welche für das Modul {3:Modul} angegeben wurde, existiert nicht.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Zusätzliche Information

- {2:Option}: Angegebene Option
- {3:Modul}: Angegebenes Modul

Ursache / Abhilfe:

- Kontrollieren, ob alle Übergabeparameter richtig an MpSkylineSetOptionState bzw. MpSkylineSetOptionS-tate angegeben wurden
- Überprüfen der Optionen in der MpSkyline Konfiguration

Konstante

`mpSKYLINE_ERR_OPTION_COLLECTION`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpSkylineCheckOptionState
- MpSkylineSetOptionState

5.6.2.3 -1064126461: Ungültige Größe

Beschreibung:

Die Variable "StateSize" wurde nicht richtig angegeben.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Kontrollieren, ob alle Übergabeparameter richtig an MpSkylineCheckOptionState angegeben wurden
- Parameter "StateSize" darf nicht 0 sein

Konstante

`mpSKYLINE_ERR_SIZE_STATE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpSkylineCheckOptionState

5.6.2.4 -1064126460: Modul existiert nicht

Beschreibung:

Das angegebene Modul {2:Modul} existiert nicht.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Zusätzliche Information

- {2:Modul}: Name des Moduls

Ursache / Abhilfe:

- Kontrollieren, ob alle Übergabeparameter richtig angegeben wurden
- Überprüfen der Module in der MpSkyline Konfiguration

Konstante

`mpSKYLINE_ERR_MODULE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpSkylineCheckOptionState
- MpSkylineResetNotification
- MpSkylineSetModuleState
- MpSkylineSetNotification
- MpSkylineSetOptionState

5.6.2.5 -1064126459: Modul-Zustand existiert nicht

Beschreibung:

Der Modul-Zustand {2:Module state} existiert nicht im Zustands-Modul {3:Module state collection}

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen der Übergabeparameter an MpSkylineSetModuleState
- Überprüfen der Modul-Zustände in der MpSkyline Konfiguration

Zusätzliche Information

- {2:Module state}: Angegebener Modul-Zustand
- {3:Module state collection}: Zustands-Modul

Konstante

`mpSKYLINE_ERR_MODULE_STATE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpSkylineSetModuleState

5.6.2.6 -1064126458: Benachrichtigung existiert nicht

Beschreibung:

Die Benachrichtigung {2:Notification} existiert nicht innerhalb der Benachrichtigungsliste {3:Notification collection}.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Zusätzliche Information

- {2:Notification}: Angegebene Benachrichtigung
- {3:Notification collection}: Benachrichtigungs-Liste

Ursache / Abhilfe:

- Überprüfen der Übergabeparameter an MpSkylineSetNotification bzw. MpSkylineResetNotification
- Überprüfen der Benachrichtigungen in der MpSkyline Konfiguration

Konstante

`mpSKYLINE_ERR_NOTIFICATION`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpSkylineResetNotification
- MpSkylineSetNotification

5.6.2.7 -1064126457: MpServer Bibliothek fehlt

Beschreibung:

Die MpServer Bibliothek fehlt.

Ursache / Abhilfe:

- Einfügen der MpServer Bibliothek

5.6.2.8 -2137868280: Zu kleine Größe definiert

Beschreibung:

Die aktuell angegebene Größe an "StateSize" ist zu klein um den kompletten Zustand anzuzeigen.

Anstatt {2:Cut off current option state} wird {3:Complete current option state} angezeigt

Zusätzliche Information

- {2:Cut off current option state}: Nicht vollständig angzeigter Zustand
- {3:Complete current option state}: Aktueller Zustand

Ursache / Abhilfe:

- Anpassen der "StateSize" Variable

Konstante

`mpSKYLINE_WRN_SIZE_TOO_SMALL`

Diese Funktionsblöcke / Funktionen können diesen Fehler melden:

- MpSkylineCheckOptionState

5.6.2.9 1083357193: Konfiguration erfolgreich geändert

Beschreibung:

Die Konfiguration wurde erfolgreich geändert.

5.6.2.10 1083357194: Erfolgreich initialisiert

Beschreibung:

mapp Skyline hat sich erfolgreich initialisiert und kann verwendet werden.

5.6.3 Alarme

5.6.3.1 mpSKYLINE_ALM_WRONG_CONFIG: Ungültige Konfiguration

Beschreibung:

Ungültige MpSkyline Konfiguration.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Kontrolle der Konfiguration
- Für mehr Information siehe Logger

Verhalten

Edge Alarm

6 mapp J1939: SAE J1939 Netzwerkprotokoll

mapp J1939 folgt dem Netzwerkprotokoll SAE J1939. SAE J1939 beschreibt die Kommunikation auf einem CAN-Bus in Nutzfahrzeugen zur Übermittlung von Diagnosedaten (z.B. Motordrehzahl, Temperatur) und Steuerungs-informationen.

Die Komponente ermöglicht es über das Netzwerkprotokoll Daten zu schicken und zu empfangen. Genauere Infos über das Netzwerkprotokoll SAE J1939 können hier gefunden werden:

- SAE J1939 auf Wikipedia

6.1 Konzept

Allgemein

- Als OEM möchte ich schnell und einfach das Kommunikationsprotokoll SAE J1939 verwenden
- Als Endnutzer möchte ich in der Lage sein, die Motorsollwerte für Drehzahl/Drehmoment zu ändern
- Als Endnutzer möchte ich Werte der Abgasnachbehandlung überwachen
- Als Servicetechniker möchte ich mir Diagnosedaten der Abgasnachbehandlung anzeigen lassen

6.1.1 Informationen senden und empfangen

Mithilfe der mapp Komponente können über eine CAN-Schnittstelle, welche das Netzwerkprotokoll J1939 verwendet, Daten gesendet und empfangen werden. Dabei kann es sich zum Beispiel um Diagnosedaten oder Steuerungsinformationen handeln. Wie die J1939 CAN-Nachricht aufgebaut ist, kann hier gefunden werden. Wie das Netzwerkprotokoll J1939 aufgebaut ist und funktioniert ist hier zu finden:

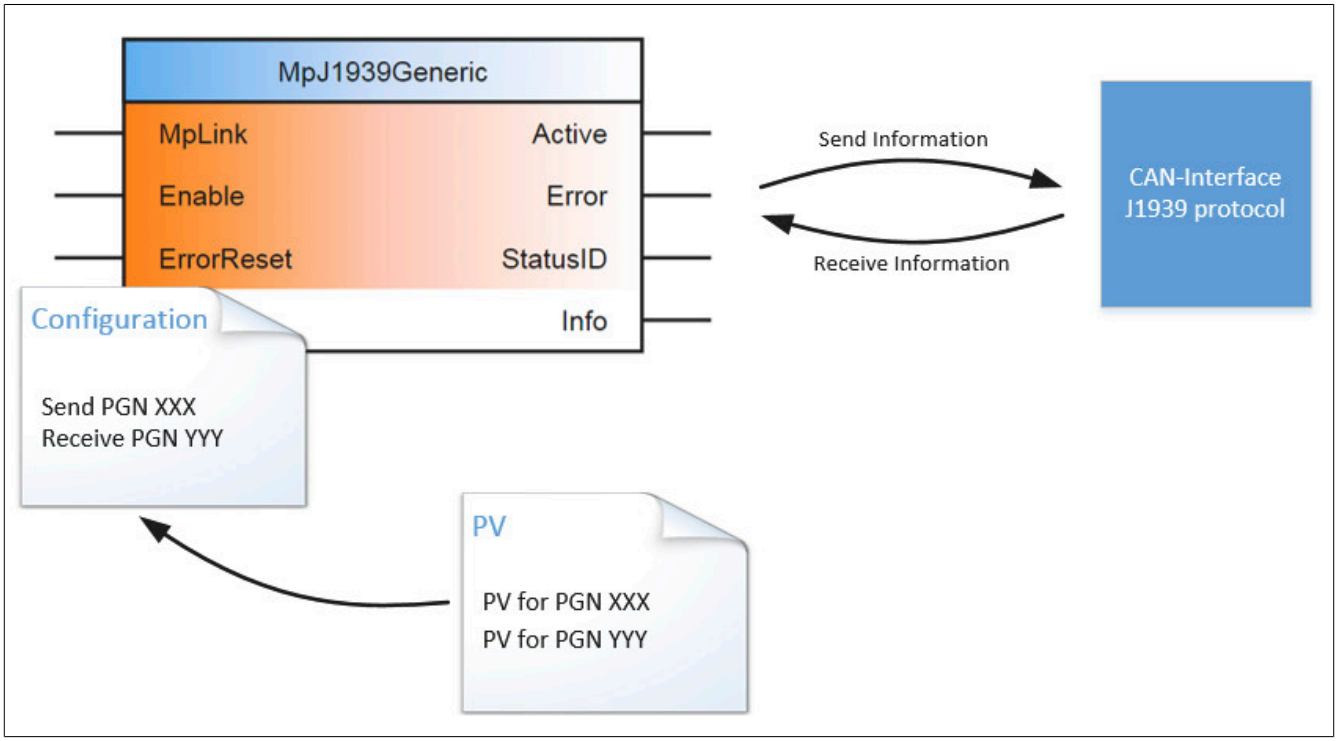
- SAE J1939 auf Wikipedia

Mithilfe des Funktionsbaustein MpJ1939Generic werden Daten gesendet und empfangen.

Welche Information übertragen wird, wird über die sogennante Parametergruppennummer (PGN) bestimmt. Welche PGNs gesendet oder empfangen werden sollen, wird in der MpJ1939Generic Konfiguration festgelegt. In der Konfiguration steht ein Teil der PGNs zur Verfügung. Manche PGNs können zyklisch abgefragt werden, andere PGNs stehen auf Abfrage zur Verfügung.

PGN welche über die Konfiguration definiert werden

Für jede PGN die verwendet werden soll, muss eine Prozessvariable von einem bestimmten Datentypen angelegt werden. Diese Prozessvariable enthält die gewünschten Informationen welche gesendet oder empfangen werden sollen. Für mehr Information siehe Arbeiten mit PGNs in der Konfiguration.



PGN welche nicht über die Konfiguration definiert werden

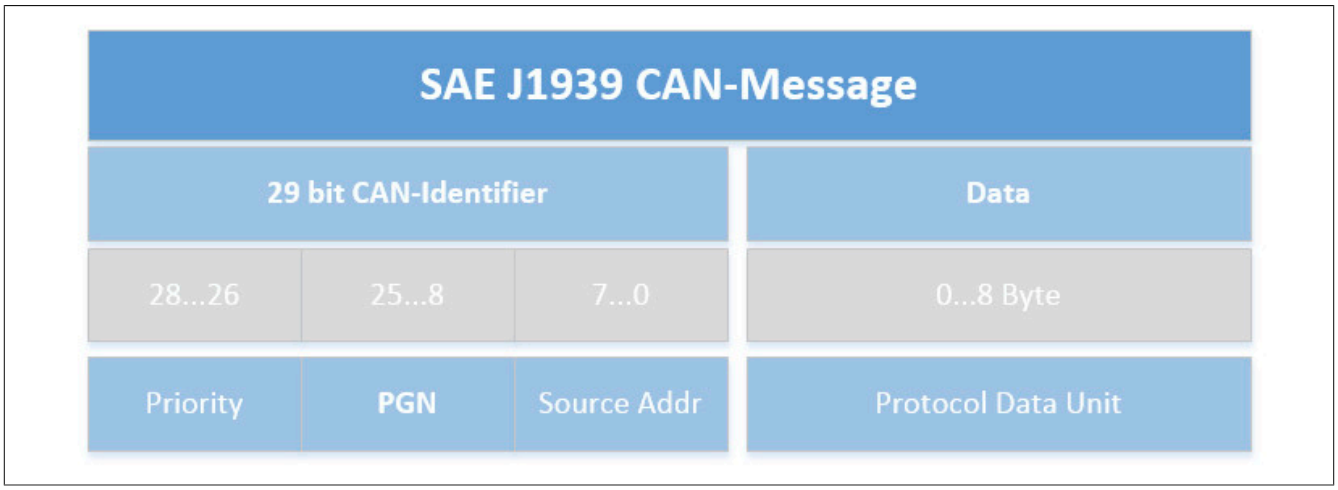
Aufgrund der Vielfalt an möglichen PGNs sind nicht alle direkt in der Konfiguration aufgelegt. Für jene PGNs welche nicht in der Konfiguration zu finden sind, können die Funktionsbausteine MpJ1939Receive, MpJ1939Transmit oder MpJ1939SpecificRequest verwendet werden.

Für zyklisch lesbare/sendbare PGNs können die Funktionsbausteine MpJ1939Receive und MpJ1939Transmit verwendet werden. Für jene PGNs die nur einmalig abgefragt werden können, wird der Funktionsbaustein MpJ1939SpecificRequest verwendet. Ob eine PGN zyklisch oder einmalig abgefragt wird, kann aus dem J1939 Standard entnommen werden.Für mehr Information siehe hier.

Dabei muss die gewünschte PGN am Funktionsbaustein angegeben werden. Die Verwaltung der Daten liegt in dem Fall beim Verwender. Für mehr Information siehe hier.

6.1.2 Aufbau der J1939 CAN-Nachricht

Die J1939 CAN-Nachricht ist folgendermaßen aufgebaut:



Der CAN-Identifier besteht aus 3 Teilen. Der Priorität, der Parametergruppennummer (PGN) und der Quelladresse. Für mehr Information zur PGN siehe hier. Die Priorität einer PGN kann aus dem J1939 Standard entnommen werden. Für weitere Informationen sie SAE J1939 auf Wikipedia.

Auflösen eines CAN-Identifiers

Ein CAN-Identifier kann zum Beispiel *18FF72E0* sein. Um herauszufinden welche PGN und Priorität der CAN-Identifier hat kann folgendermaßen vorgegangen werden:

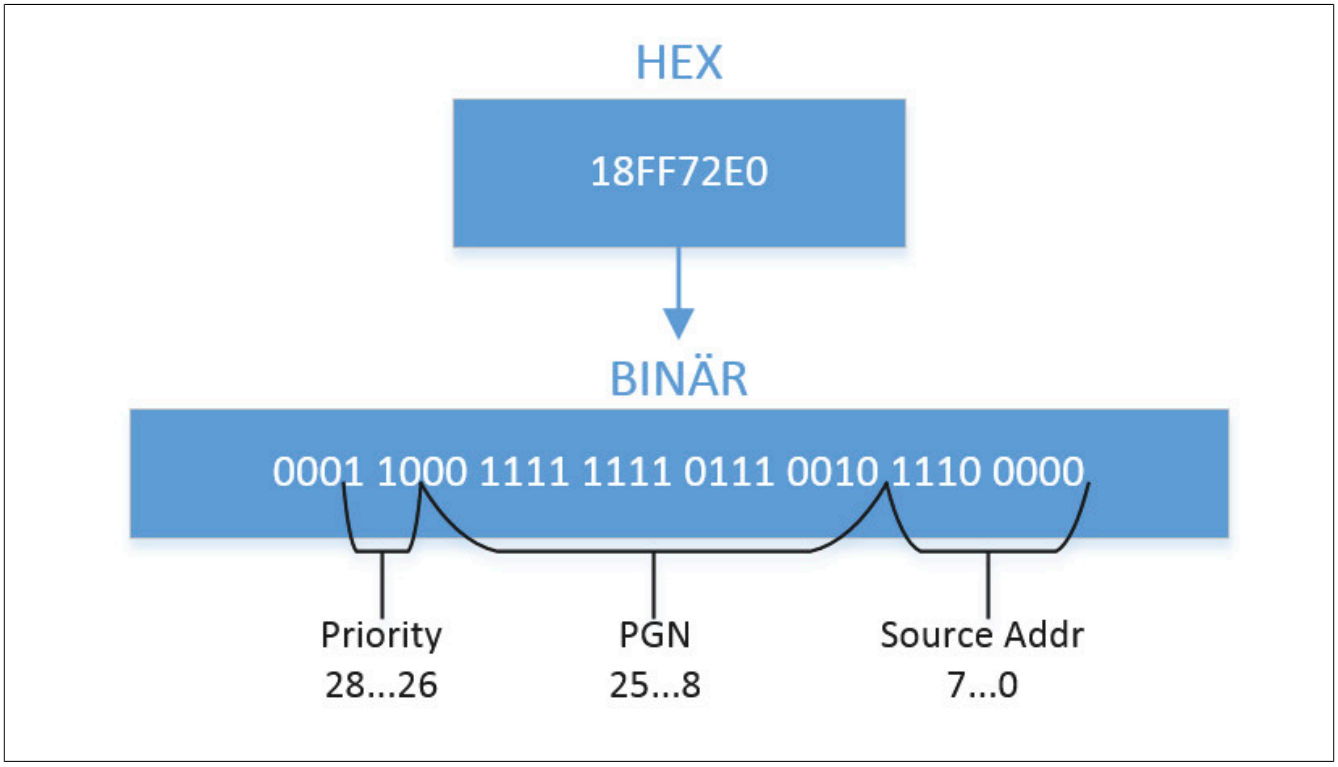
1. Umrechnen

Der CAN-Identifier wird im HEX-Format angegeben. Um die PGN Nummer und die Priorität herauszufinden, muss das HEX-Format in Binär-Format umgewandelt werden.

18FF72E0 in Binär ist 0001 1000 1111 1111 0111 0010 1110 0000.

2. PGN und Priorität bestimmen

Aus der Binär-Zahl kann die PGN und die Priorität und auch die Quelladresse wieder im HEX-Format bestimmt werden.



- **Priorität:** 6
- **PGN:** FF72
- **Quelladresse:** E0

6.1.3 Timing

PGNs müssen, damit Information gesendet oder empfangen werden kann, in einer bestimmten Zykluszeit empfangen bzw. gesendet werden.

Die PGN DM01 muss zum Beispiel mindestens alle 1000ms empfangen bzw. gesendet werden. Wird die PGN in einer langsameren Zykluszeit wie zum Beispiel 3000ms empfangen, so wird die PGN zwar empfangen, am Status der PGN ist aber ersichtlich, dass die PGN in der falschen Zykluszeit (mpJ1939_PGN_ERROR_CYCLE_TIME) emfpangen wurde.

Wird die PGN DM01 in einer schnelleren Zykluszeit wie zum Beispiel 10ms abgefragt, so ist dies kein Problem. Es entsteht kein Fehler beim Empfangen. Die PGN DM01 wird dabei trotzdem erst in 1000ms empfangen/gesendet.

Wird die PGN DM01 über den Funktionsbaustein MpJ1939Generic empfangen, so muss sich dieser Funktionsbaustein in einer Taskklasse von <=1000ms befinden.

Den Status der empfangenen/ gesendeten PGN kann am Status Parameter der bereitgestellten Variable abgelesen werden.

Sollen nun verschiedenen PGNs empfangen/ gsendet werden, so können diese in einer Taskklasse aufgerufen werden, welche die schnellste Zykluszeit für eine PGN beinhaltet. Werden also die PGNs TSC1, DM1 und CCVS1 verwendet, so können alle in der Zykluszeit 10ms abgefragt werden, welche für die PGN TSC1 notwendig ist. Wichtig hierbei ist, dass zum Beispiel die PGN DM1 dabei trotzdem automatisch nur alle 1000ms empfangen/gesendet wird, da das die Zykluszeit der PGN ist.

Für die PGNs welche in der MpJ1939Generic Konfiguration definiert werden können, sind folgende Zykluszeiten einzuhalten:

PGN	Zykluszeit [ms]
TSC1	10
DM1	1000
AMB	1000
CCVS1	100
EC1	5000
EEC1	20
EEC2	50
EEC3	250
EFL_P1	500
ET1	1000
IC1	500
LFE	100
SHUTDN	1000
VEP1	1000
HOURS	PGN wird in vorgegebenen Zyklus angefragt.
CI	PGN wird in vorgegebenen Zyklus angefragt.
LFC	PGN wird in vorgegebenen Zyklus angefragt.
SOFT	PGN wird in vorgegebenen Zyklus angefragt.
CM1	1000
AT1IG1	50
AT1IG2	500
AT1IMG	500
AT1OG1	50
AT1OG2	500
AT1T1I	1000
DPFC1	1000
AT1S1	1000
DM01	1000

Für PGNs welche nicht in der MpJ1939Generic Konfiguration bereitgestellt werden, kann die Information der Zykluszeit aus dem J1939 Standard entnommen werden. Für mehr Information siehe hier.

6.1.4 Speichermanagement

Mithilfe der Funktionsbausteine MpJ1939Receive, MpJ1939Transmit und MpJ1939SpecificRequest kann eine PGN gesendet oder empfangen werden, welche nicht über die Konfiguration verfügbar ist.

> **Um mit PGNs zu arbeiten welche nicht über die Konfiguration verfügbar sind, empfiehlt es sich die benötigte Information aus dem J1939 Standard zu entnehmen. Für mehr Information siehe hier.**

Die gewünschte PGN wird dabei über den Eingangsparameter "PGN" definiert. Dabei wird die numerische Kennzahl der PGN angegeben. Welchen numerischen Wert eine PGN identifiziert, kann aus dem J1939 Standard entnommen werden. Um die Information einer PGN verwalten zu können gibt es zwei verschiedene Möglichkeiten.

- PGN anhand PV verwalten
- PGN im Speicher verwalten

Je nachdem wie die Signale in der PGN liegen, kann zwischen den beiden Varianten entschieden werden. Die Signale einer PGN haben immer ein Startbit und eine maximale Länge.

> **Es gilt zu beachten, dass eine PGN Grenzwerte, Skalierungen sowie einen Offset haben kann. Diese müssen bei der Weiterverarbeitung der PGN berücksichtigt werden. Welche Grenzwerte, Offset oder Skalierung eine PGN hat kann in einer J1939 Datenbank entnommen werden, welche hier heruntergeladen werden kann.**

Hat man zum Beispiel eine PGN welche die Signale "Test_A" und "Test_B" beinhalten, so haben diese Signale ein Startbit und eine definierte Länge.

Für die Variante PGN anhand PV verwalten müssen die Signale direkt aufeinander folgen. Es gibt keine Bits zwischen den Signalen:

Signal	Startbit	Länge
Test_A	0	8
Test_B	8	16

Für die Variante PGN im Speicher verwalten folgen die Signale nicht direkt aufeinander. Im unteren Beispiel hat Signal "Test_A" eine Länge von 8 Bit. Signal "Test_B" fängt aber erst an der Bitstelle 16 an.

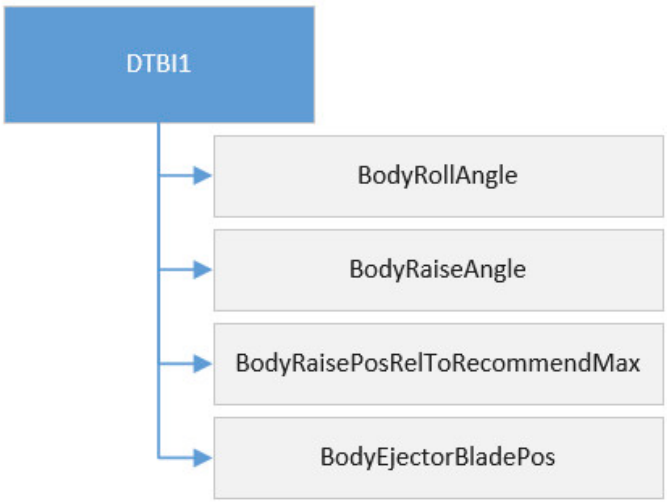
Signal	Startbit	Länge
Test_A	0	8
Test_B	16	8

6.1.4.1 PGN anhand PV verwalten

In diesem Abschnitt wird erklärt, wie die Information einer PGN anhand einer PV verwaltet werden kann.

Beispiel

Der numerische Wert der PGN DTBI1 ist zum Beispiel 61710 (0xF10E). Die PGN DTBI1 kann folgende Signale liefern bzw. senden:



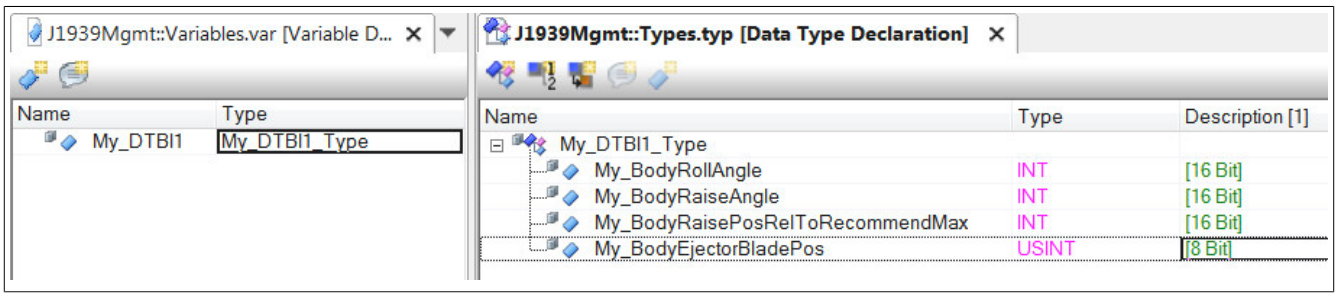
Diese Signale haben eine definierte Länge. Das Signal BodyRollAngle hat zum Beispiel eine Länge von 16 Bit.

Über den Eingangsparameter "Data" der Funktionsbausteine muss ein Buffer bereitgestellt werden, der die Werte des PGN DTBI1 empfangen und anzeigen kann.

Um die PGN DTBI1 zu empfangen müsste eine Struktur erstellt werden welche exakt die Längen der Signale enthält. Dabei muss genau bekannt sein wie das PGN DTBI1 aufgebaut ist. Jedes Signal hat ein Start-Bit und eine fixe Bit-Länge. Das heißt es ist genau festgelegt ab welchen Bit das Signal startet und ab welchen Bit es endet. Für PGN DTBI1 sieht das folgendermaßen aus:

Signal	Startbit	Länge
BodyRollAngle	0	16 Bit
BodyRaiseAngle	16	16 Bit
BodyRaisePosRelToRecommendMax	32	16 Bit
BodyEjectorBladePos	48	8 Bit

Um die Information dieses PGN zu erhalten, muss nun eine Struktur erstellt werden, welche exakt die gleiche Bitlänge hat.
Dies kann zum Beispiel folgendermaßen aussehen:



Es wurde die Variable "My_DTBI1" erstellt welche vom benutzerdefinierten Datentypen "My_DTBI1_Type" ist. Dieser Datentyp enthält Variablen welche der gleichen Bit-Länge vom PGN DTBI1 entspricht.

> **Das Alignment des Benutzerdefinierten Datentypen muss genau beachtet werden. Dies muss genau mit der gewünschten PGN übereinstimmen.**
>
> **Der Benutzerdefinierte Datentyp darf keine Füllbytes enthalten!**

6.1.4.2 PGN im Speicher verwalten

In diesem Abschnitt wird erklärt, wie die Information einer PGN im Speicher verwaltet werden kann.

Beispiel

Die PGN CCVS3 hat den numerischen Wert 64732 (0xFCDC) . Die PGN CCVS3 kann folgende Signale liefern bzw. senden:

Diese Signale haben eine definierte Länge. Das Signal CruiseCtrlSpeed hat zum Beispiel eine Länge von 16 Bit.

Über den Eingangsparameter "Data" der Funktionsbausteine MpJ1939Receive , MpJ1939SpecificRequest und MpJ1939Transmit muss ein Buffer bereitgestellt werden, der die Werte des PGN CCVS3 empfangen und anzeigen kann.

Um die PGN CCVS3 zu empfangen müsste eine Variable erstellt werden welche die Information von der PGN empfangen kann.

Dabei muss genau bekannt sein wie das PGN CCVS3 aufgebaut ist. Jedes Signal hat ein Start-Bit und eine fixe Bit-Länge. Das heißt es ist genau festgelegt ab welchen Bit das Signal startet und ab welchen Bit es endet. Die Signale der PGN folgen nicht alle direkt aufeinander. Bei manchen Signalen gibt es dazwischen unbenützte Bits. Für PGN CCVS3 sieht das folgendermaßen aus:

Signal	Startbit	Länge [Bit]
AdptiveCruiseCtrlReadinessStatus	0	2
CruiseCtrlSystemCmdState	2	3
PrdctvCrsCtrlStSpeedOffsetStatus	5	2
SrcAddrssOfCntrllngDvcFrDsblngCr	8	8
SrcAddrssOfCntrllngDvcFrPsngCrsC	16	8
AEBSReadinessState	24	2
DriverCancellationStatus	26	2
CruiseCtrlSetSpeed	32	16
CruiseCtrlSpeed	48	16

Um die Informationen zu empfangen wird, zum Beispiel, die Variable "DataBuffer" erstellt, welche vom Datentypen "BYTE[0..19]" ist.

Die Variable "DataBuffer" wird später am Eingang "Data" des zum Beispiel MpJ1939Receive Funktionsbausteins angegeben.

Soll nun zum Beispiel der Wert des Signals "SrcAddrssOfCntrllngDvcFrPsngCrsC" evaluiert werden, so muss auf "DataBuffer[2]" zugegriffen werden. Soll auf das Signal "CruiseCtrlSystemCmdState" zugegriffen werden, so kann dies durch einen Bitweisen Zugriff erfolgen. Dies kann zum Beispiel durch "DataBuffer[0].2" realisiert werden.

6.1.5 Rahmenbedingungen

In diesem Abschnitt werden Rahmenbedingungen von mapp J1939 aufgelistet.

- Das J1939 Protokoll basiert auf "CAN2.0B - Extended frame format" mit 29 Identifizierungs-Bits
- Hardware welche von CAN2.0B unterstützt wird, kann auch von mapp J1939 verwendet werden
- mapp J1939 kann in Simulation nicht verwendet werden.
- Module, welche die Bibliothek "CAN_Lib" nicht unterstützten, können nicht mit mapp J1939 verwendet werden. Dabei kann auch das Funktionsmodell des Moduls entscheidend sein. Das Modul X20CS1070 unterstützt beispielsweise im Modus "Flat" die Bibliothek "CAN_lib" nicht. Bei der Verwendung von "Stream" bzw. "Cyclicstream" wird "CAN_lib" allerdings unterstützt!

6.2 Getting Started

6.2.1 Motordaten empfangen

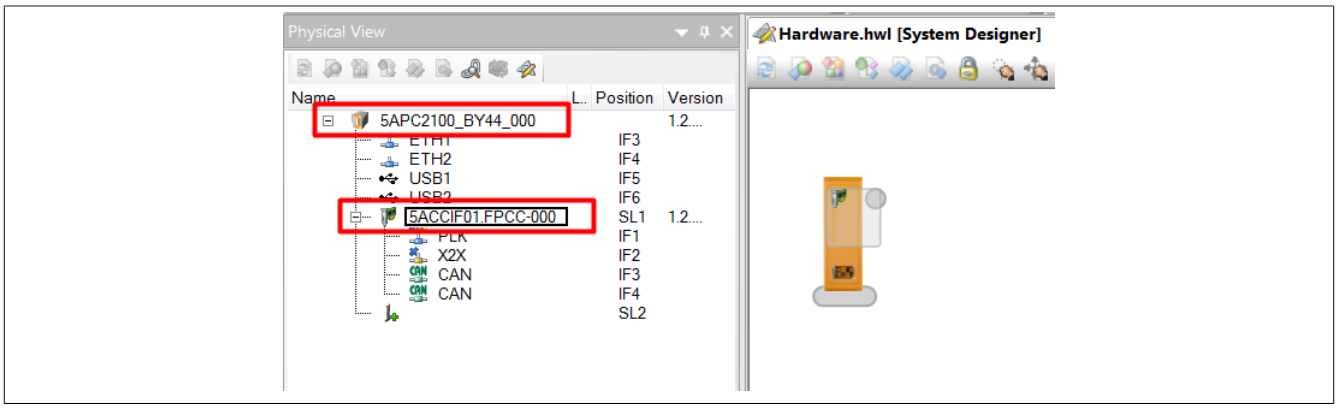
In diesem Abschnitt wird erklärt, wie man mithilfe von MpJ1939Generic Motordaten empfangen kann.

6.2.1.1 Projekt erstellen

Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

Hardwarekonfiguration

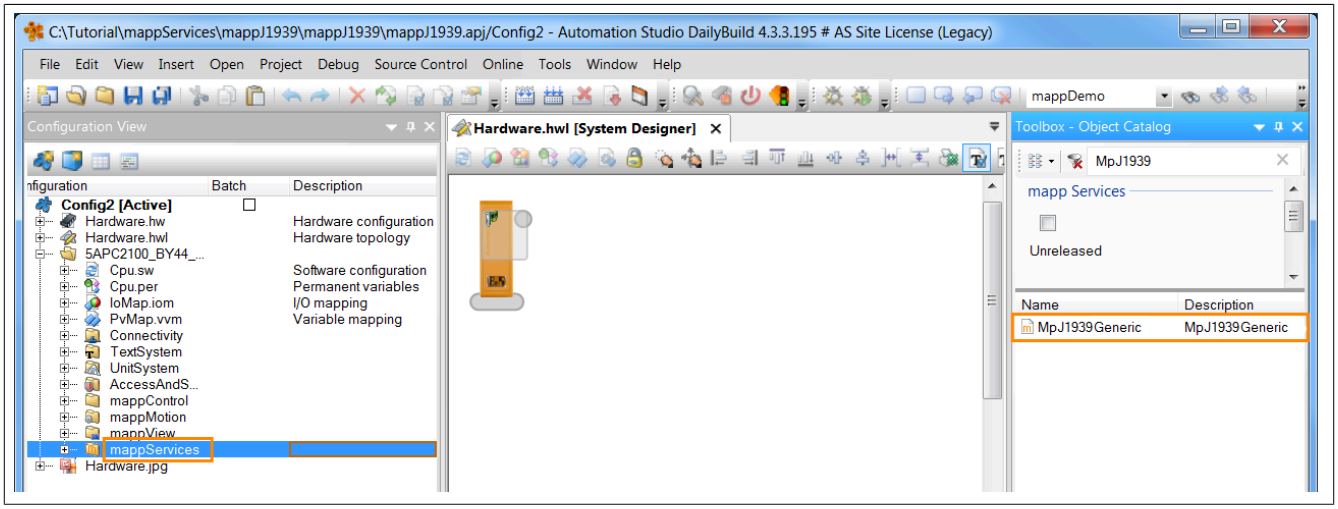
Als zu verwendende Hardware kann zum Beispiel der APC2100.BY44.0000 eingefügt werden. Damit die CAN-Schnittstelle verwendet werden kann, wird die Einsteck-Karte 5ACCIF01.FPCC-0000 hinzugefügt.



6.2.1.2 mapp Komponente einfügen

Konfiguration MpJ1939Generic einfügen

Um Motordaten empfangen zu können, wird die Konfiguration MpJ1939Generic eingefügt.



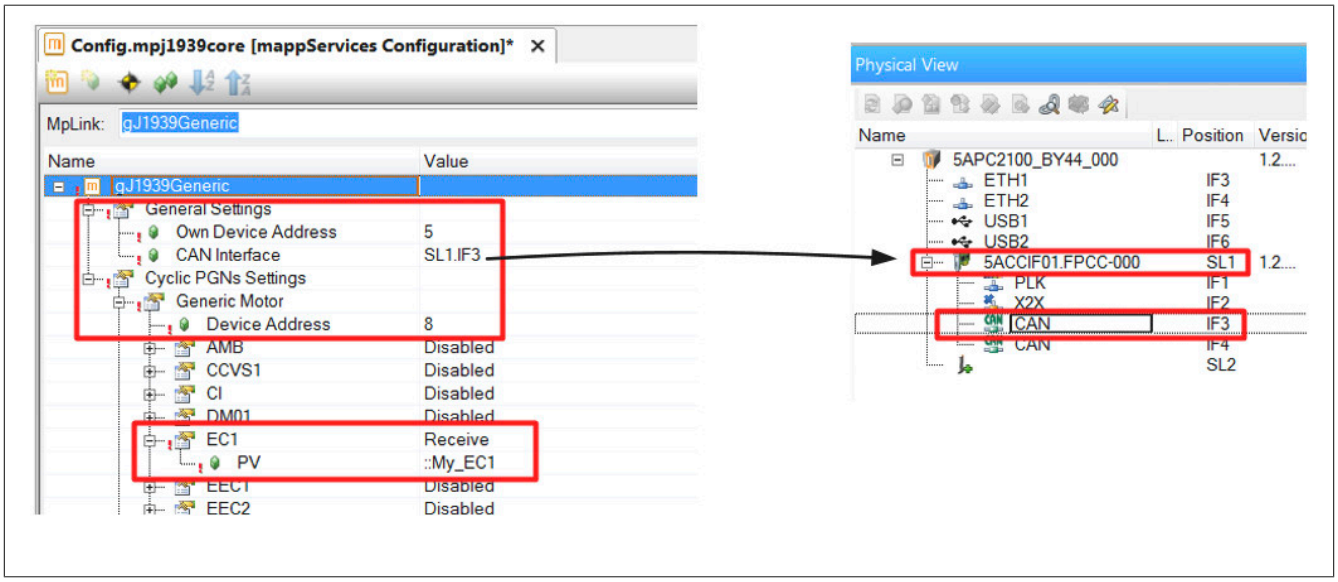
Hier befindet sich der mapp Link, welcher die mapp Komponente in der „world of mapp“ eindeutig identifiziert. Der mapp Link stellt die Verbindung zwischen Programmierung und Konfiguration her.

Die CAN-Schnittstelle wird am Parameter "CAN Interface angegeben". Diese Information kann aus der Physical View entnommen werden.

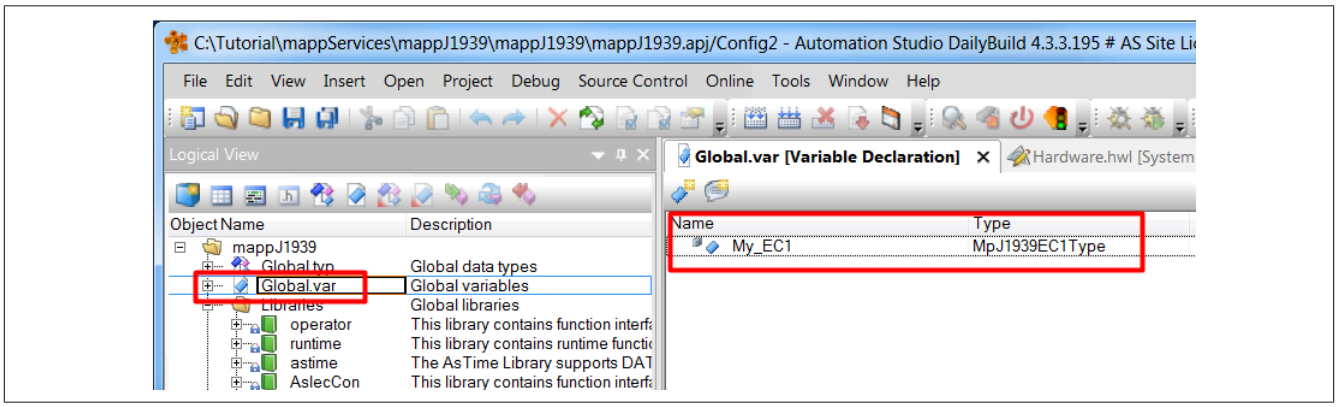
Unter "Own Device Address" muss die Adresse angegeben werden, welche für die eigene Identifizierung im CAN-Netzwerk dient. Unter "Device Adress" muss die Gegenstelle im CAN-Netzwerk angegeben werden. Diese wird beim Aufbau des CAN-Netzwerks festgelegt.

Die unten angezeigten PGNs können empfangen oder gesendet werden. Es soll die PGN EC1 empfangen werden.

Damit die Information empfangen werden kann, muss eine Prozessvariable angegeben werden. Es wird die Prozessvariable My_EC1 vom Datentypen MpJ1939EC1Type angegeben.



Die Prozessvariable wurde in der globalen Variablendeklarationsdatei definiert:

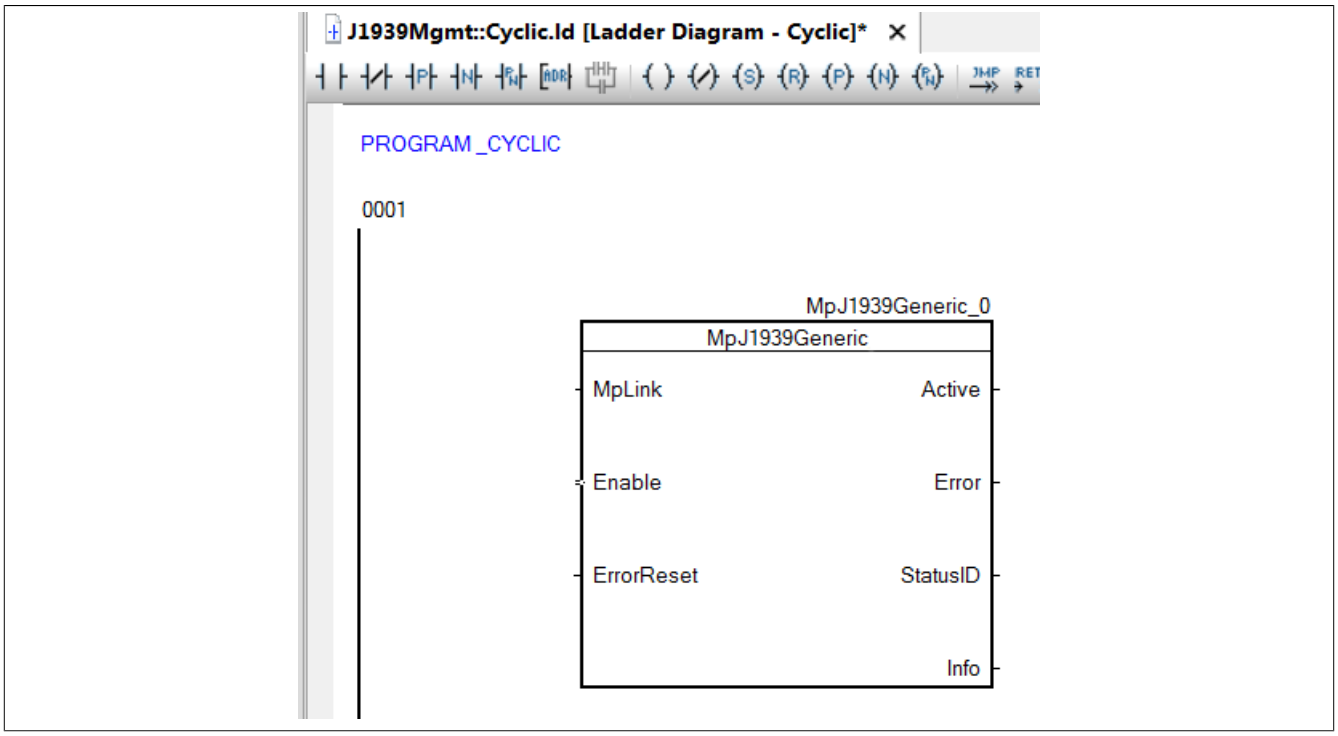


6.2.1.3 Programm hinzufügen

Der nächste Schritt ist das Einfügen eines Kontaktplan-Programms. Für mehr Information siehe hier.

6.2.1.4 MpJ1939Generic einfügen

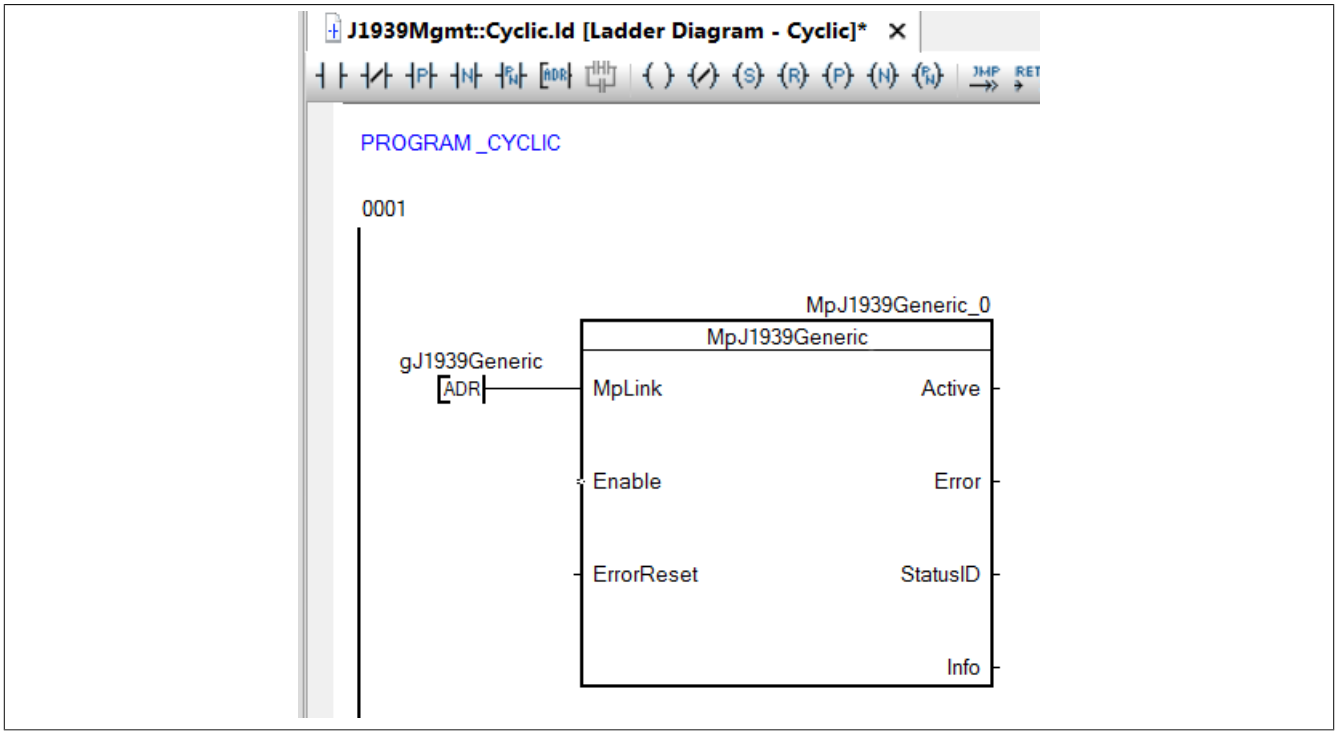
Es wird der Funktionsbaustein MpJ1939Generic eingefügt.



6.2.1.5 Parametrieren von MpJ1939Generic

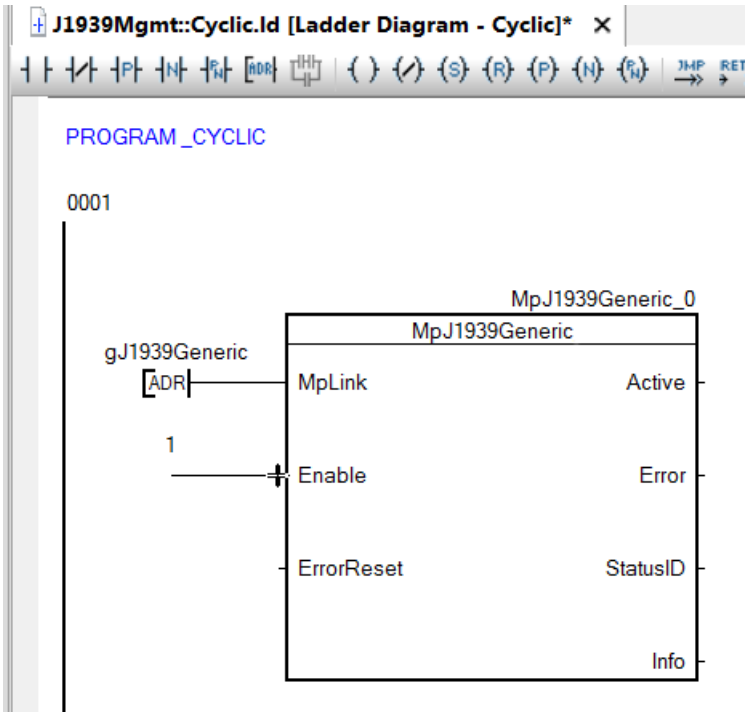
Verbinden des MpLinks

Mit dem Eingang MpLink stellen wir den Bezug zur Konfiguration her, welche wir zuvor mit dem Namen gJ1939Generic erstellt haben.



Aktivieren des Funktionsbausteins

Der Funktionsbaustein wird durch den Eingang „Enable“ aktiviert.



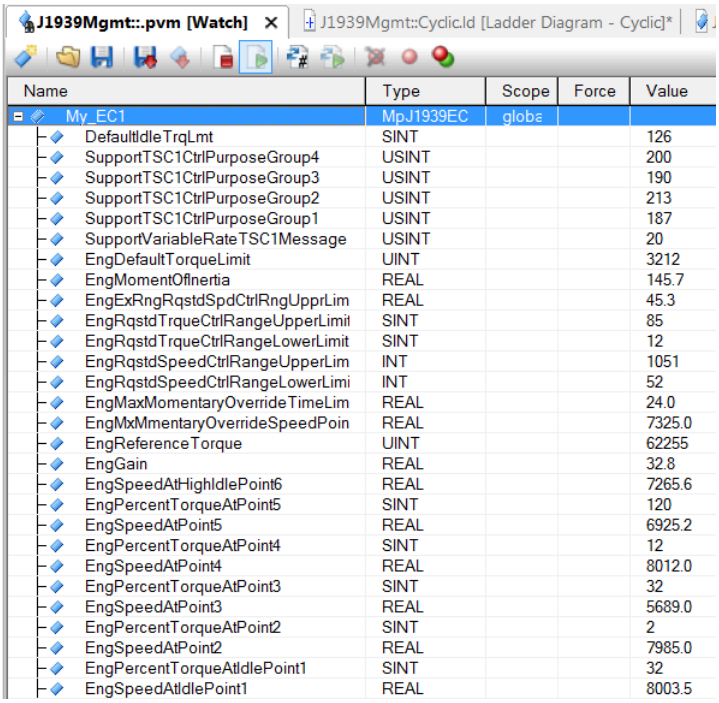
6.2.1.6 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

6.2.1.7 Testen des Programms

Nach einem Download der Änderungen kann das Programm getestet werden.

Ist an der CAN-Schnittstelle ein Motor verbunden, so wird die in der Konfiguration festgelegte PGN EC1 mithilfe von MpJ1939Generic automatisch empfangen und die Werte werden an der Variable My_EC1 angezeigt.



Name	Type	Scope	Force	Value
My_EC1	MpJ1939EC	globa		
DefaultIdleTrqLmt	SINT			126
SupportTSC1CtrlPurposeGroup4	USINT			200
SupportTSC1CtrlPurposeGroup3	USINT			190
SupportTSC1CtrlPurposeGroup2	USINT			213
SupportTSC1CtrlPurposeGroup1	USINT			187
SupportVariableRateTSC1Message	USINT			20
EngDefaultTorqueLimit	UINT			3212
EngMomentOfInertia	REAL			145.7
EngExRngRqstdSpdCtrlRngUpprLim	REAL			45.3
EngRqstdTrqueCtrlRangeUpperLimit	SINT			85
EngRqstdTrqueCtrlRangeLowerLimit	SINT			12
EngRqstdSpeedCtrlRangeUpperLim	INT			1051
EngRqstdSpeedCtrlRangeLowerLimi	INT			52
EngMaxMomentaryOverrideTimeLim	REAL			24.0
EngMxMmentaryOverrideSpeedPoin	REAL			7325.0
EngReferenceTorque	UINT			62255
EngGain	REAL			32.8
EngSpeedAtHighIdlePoint6	REAL			7265.6
EngPercentTorqueAtPoint5	SINT			120
EngSpeedAtPoint5	REAL			6925.2
EngPercentTorqueAtPoint4	SINT			12
EngSpeedAtPoint4	REAL			8012.0
EngPercentTorqueAtPoint3	SINT			32
EngSpeedAtPoint3	REAL			5689.0
EngPercentTorqueAtPoint2	SINT			2
EngSpeedAtPoint2	REAL			7985.0
EngPercentTorqueAtIdlePoint1	SINT			32
EngSpeedAtIdlePoint1	REAL			8003.5

6.3 Konfiguration

6.3.1 MpJ1939Generic Konfiguration

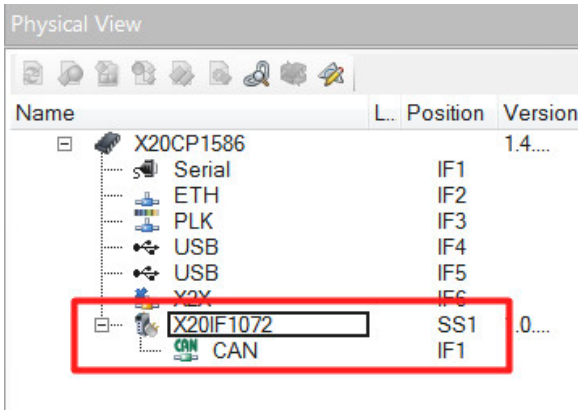
> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**
>
> **Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein MpJ1939Generic de-, und wieder aktiviert werden.**

Allgemeine Einstellungen

In den allgemeinen Einstellungen werden Informationen der CAN-Schnittstelle angegeben.

Unter "Own Device Address" muss die Adresse angegeben werden, welche für die eigene Identifizierung im CAN-Netzwerk dient.

Unter "CAN Interface" wird die Schnittstelle des CAN-Interface angegeben. Diese kann zum Beispiel SS1.IF1 sein.



PGN Einstellungen

Unter "Device Adress" muss die Gegenstelle im CAN-Netzwerk angegeben werden. Diese wird beim Aufbau des CAN-Netzwerks festgelegt.

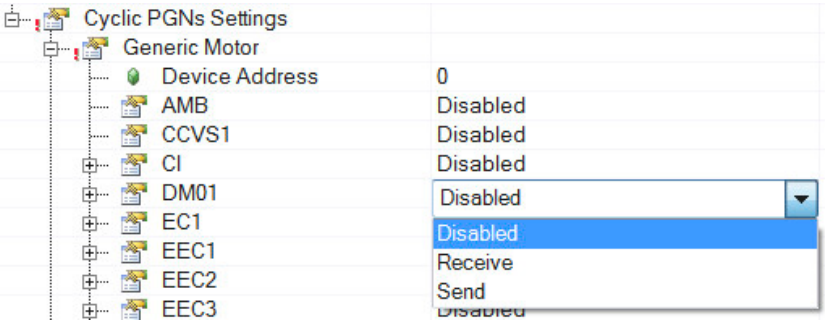
PGNs sind sogenannte Parametergruppennummern. Über diese kann Information geschickt und empfangen werden. Für mehr Information siehe Informationen senden und empfangen.

Wie mit den PGNs in der Konfiguration gearbeitet wird ist hier zu finden.

6.3.1.1 Arbeiten mit PGNs in der Konfiguration

In der MpJ1939Generic Konfiguration kann definiert werden, welche PGN aktiviert werden soll und welche nicht.

Wird eine PGN aktiviert, so kann definiert werden ob die PGN gesendet("Send") oder empfangen("Receive") werden soll.



Kann eine PGN nicht zyklisch gelesen oder gesendet werden, so erscheint die Auswahl "Request".

Zum Schluss muss eine Prozessvariable angegeben werden, welche die Informationen empfängt bzw. schickt. Der Datentyp der Prozessvariable unterscheidet sich von PGN zu PGN.

Wie die Daten richtig gesendet oder empfangen werden, ist im Abschnitt Timing zu finden.

In der folgenden Tabelle wird erklärt, welcher Datentyp für welche PGN verwendet werden muss.

PGN	Zu verwendender Datentyp
AMB	MpJ1939AMBType
CCVS1	MpJ1939CCVS1Type
CI	MpJ1939CIType
DM01	MpJ1939DM01Type
EC1	MpJ1939EC1Type
EEC1	MpJ1939EEC1Type
EEC2	MpJ1939EEC2Type
EEC3	MpJ1939EEC3Type
EFL_P1	MpJ1939EFL_P1Type
ET1	MpJ1939ET1Type
HOURS	MpJ1939HOURSType
IC1	MpJ1939IC1Type
LFC1	MpJ1939LFC1Type
LFE1	MpJ1939LFE1Type
SHUTDN	MpJ1939SHUTDNType
SOFT	MpJ1939SOFTType
TSC1	MpJ1939TSC1Type
VEP1	MpJ1939VEP1Type
AT1T1I1	MpJ1939AT1T1I1Type
AT1IG1	MpJ1939AT1IG1Type
AT1IG2	MpJ1939AT1IG2Type
AT1IMG	MpJ1939AT1IMGType
AT1OG1	MpJ1939AT1OG1Type
AT1OG2	MpJ1939AT1OG2Type
AT1S1	MpJ1939AT1S1Type
CM1	MpJ1939CM1Type
DPFC1	MpJ1939DPFC1Type

6.3.2 MpJ1939Receive Konfiguration

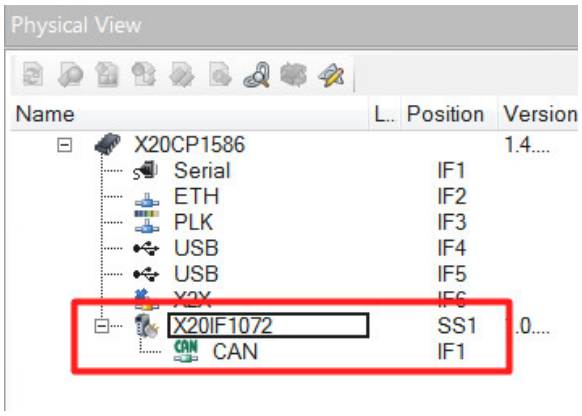
Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein MpJ1939Receive de-, und wieder aktiviert werden.

Allgemeine Einstellungen

Unter "Own Device Address" muss die Adresse angegeben werden, welche für die eigene Identifizierung im CAN-Netzwerk dient.

Unter "CAN Interface" wird der Pfad der CAN-Schnittstelle angegeben. Diese kann zum Beispiel SS1.IF1 sein



Unter "Source Adress" muss die Gegenstelle im CAN-Netzwerk angegeben werden. Diese wird beim Aufbau des CAN-Netzwerks festgelegt.

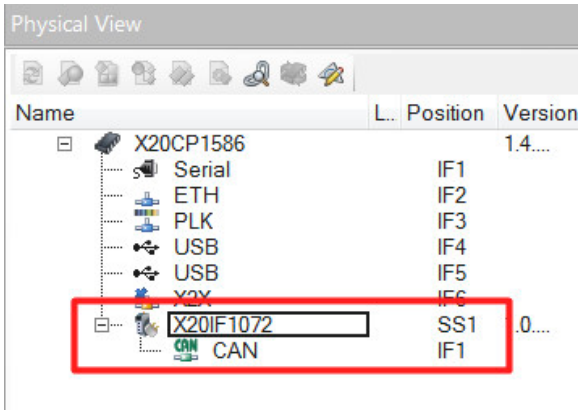
6.3.3 MpJ1939SpecificRequest Konfiguration

> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**
>
> **Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein MpJ1939SpecificRequest de-, und wieder aktiviert werden.**

Allgemeine Einstellung

Unter "Own Device Address" muss die Adresse angegeben werden, welche für die eigene Identifizierung im CAN-Netzwerk dient.

Unter "CAN Interface" wird der Pfad der CAN-Schnittstelle angegeben. Diese kann zum Beispiel SS1.IF1 sein



Unter "Source Adress" muss die Gegenstelle im CAN-Netzwerk angegeben werden. Diese wird beim Aufbau des CAN-Netzwerks festgelegt.

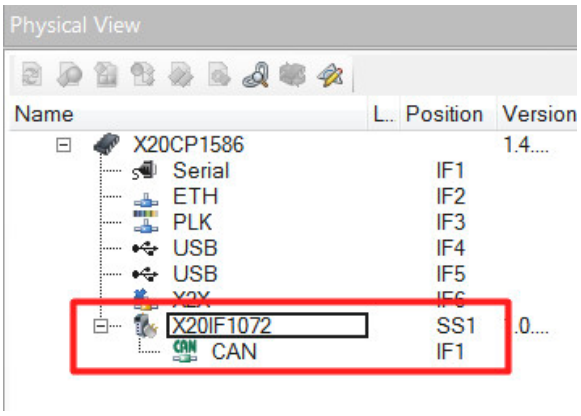
6.3.4 MpJ1939Transmit Konfiguration

> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**
>
> **Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein MpJ1939Transmit de-, und wieder aktiviert werden.**

Allgemeine Einstellungen

Unter "Own device address" muss die Knotennummer des verwendeten Geräts angegeben werden.

Unter "CAN Interface" wird der Pfad der CAN-Schnittstelle angegeben. Diese kann zum Beispiel SS1.IF1 sein



Unter "Destination Adress" muss die Gegenstelle im CAN-Netzwerk angegeben werden. Diese wird beim Aufbau des CAN-Netzwerks festgelegt.

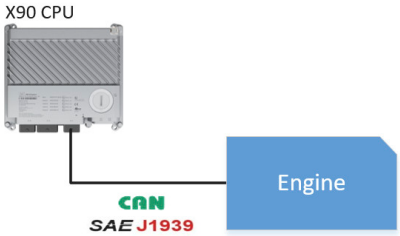
6.4 Anwendungsfälle

6.4.1 Kommunikation mit Motorsteuergerät

Anforderung

In einem Bagger ist eine X90 Steuerung verbaut, die als Steuergerät für den kompletten Bagger fungiert.

Diese kommuniziert über die CAN-Schnittstelle und das J1939 Protokoll mit dem Motorsteuergerät. Da die X90 Steuerung die zentrale Logik des Baggers darstellt muss sie sowohl Sollwerte wie Solldrehzahl und maximales Moment an das Motorsteuergerät senden können, als auch Istwerte wie aktuelle Drehzahl oder Öltemperatur zurücklesen können.

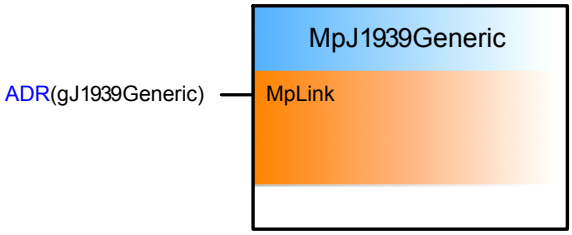


Lösung

Funktionsbausteinliste

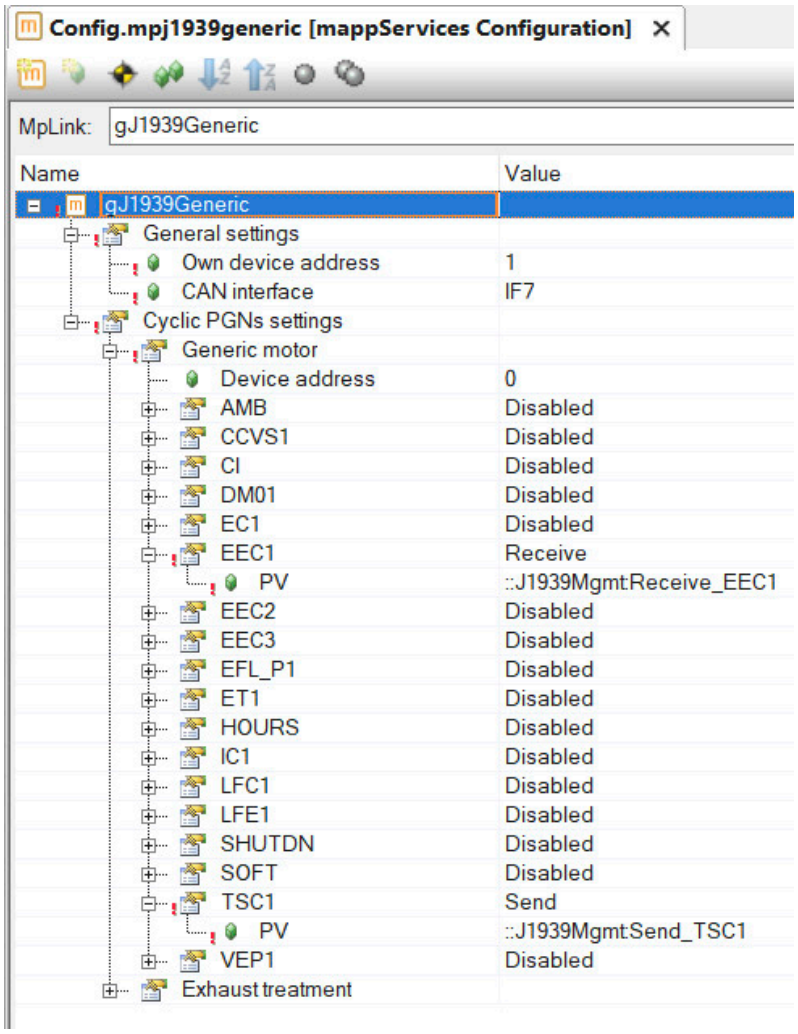
- ***MpJ1939Generic*** (Eigener MpLink): Mithilfe dieses Funktionsbausteins können PGNs über das J1939 Protokoll empfangen und gesendet werden.

Verbindungsschema

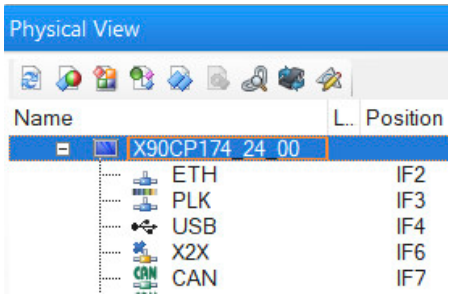


Konfiguration

Um über die CAN-Schnittstelle und das J1939 Protokoll mit dem Motorsteuergerät zu kommunizieren, wird die MpJ1939Generic Konfiguration eingefügt.

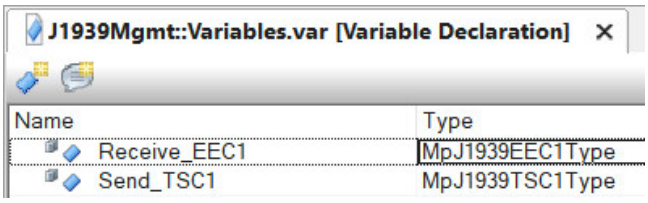


Unter "Own Device Address" muss die Adresse angegeben werden, welche für die eigene Identifizierung im CAN-Netzwerk dient. Es wurde 1 festgelegt. Unter "CAN Interface" wird die Schnittstelle des CAN-Interface angegeben. In diesem Fall wird IF7 definiert.



Unter "Device Adress" muss die Adresse des Motorsteuergeräts im CAN-Netzwerk angegeben werden. Es wurde 0 definiert.

Über die PGNs können Informationen geschickt und empfangen werden. Es sollen die Sollwerte des Motorsteuergeräts, wie zum Beispiel Solldrehzahl und maximales Moment bestimmt werden. Dies kann über die PGN TSC1 definiert werden. Um die Istwerte des Motorsteuergeräts zu empfangen, wird die PGN EEC1 verwendet. Es wird dementsprechend "Receive" und "Send" eingestellt. Im Anschluss müssen PVs angegeben werden welche die Informationen empfangen bzw. senden. Die PVs müssen dementsprechend den richtigen Datentypen besitzen. Für mehr Information siehe hier.



Verwendung der mapp Komponenten

Es wird der Funktionsbaustein **MpJ1939Generic** eingefügt und parametriert. Die in der Konfiguration definierten PGNs werden automatisch gesendet oder empfangen.

Um eine PGN richtig zu senden bzw. zu empfangen muss das Timing beachtet werden. Sollen verschiedenen PGNs empfangen/ gsendet werden, so können diese in einer Taskklasse aufgerufen werden, welche die schnellste Zykluszeit für eine PGN beinhaltet.

TSC1 wird in einer Zykluszeit von 10ms und EEC1 in 20ms abgefragt. Aus diesem Grund wird der Funktionsbaustein **MpJ1939Generic** in einem Task aufgerufen, welcher sich in einer Taskklasse von 10ms befindet.

Die an der Variable "Send_TSC1" definierten Werte werden automatisch an das verbundene Motorsteuergerät gesendet. Die Ist-Werte des Geräts werden an der Variable "Receive_EEC1" angezeigt. Der Funktionsbaustein **MpJ1939Generic** regelt das senden und empfangen der PGNs automatisch ohne ein zusätzliches Kommando zu starten.

Haben Signale in PGNs Skalierungen oder einen Offset so werden diese beim Senden und Empfangen der PGN automatisch angepasst. Das Signal EngRequestedSpeed_SpeedLimit in der PGN TSC1 hat zum Beispiel einen Faktor von 0,125. Dieser Faktor wird beim Senden und Empfangen der PGN automatisch berücksichtigt!

6.5 Diagnose

Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

6.5.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp J1939 auftreten können.

Meine Standard PGN wird nicht richtig empfangen

Unter Standard PGN werden die PGNs bezeichnet, welche über die MpJ1939Generic Konfiguration verfügbar sind.

Diese PGNs folgen einem festgelegten J1939 Protokoll. Wurde zum Beispiel ein weiterer Fehlercode in einer PGN, beispielsweise DM01 hinzugefügt, so muss die PGN mithilfe von MpJ1939Receive extrahiert werden.

Welche Module können in Kombination mit mapp J1939 verwendet werden?

Es werden nur Module untersützt, welche die Bibliothek "CAN_Lib" unterstützen.

Dabei kann auch das Funktionsmodell des Moduls entscheidend sein. Das Modul X20CS1070 unterstützt beispielsweise im Modus "Flat" die Bibliothek "CAN_Lib" nicht. Bei der Verwendung von "Stream" bzw. "Cyclicstream" wird "CAN_Lib" allerdings unterstützt!

6.6 Bibliotheken

6.6.1 Datentypen und Enumeratoren

6.6.1.1 Datentypen

6.6.1.1.1 MpJ1939AMBType

Dieser Datentyp beschreibt die PGN AMB (Umgebungsbedingungen).

Parameter	Datentyp	Beschreibung
RoadSurfaceTemp	`REAL`	Temperatur der Straßenoberfläche, über der das Fahrzeug fährt.
EngAirIntakeTemp	`INT`	Temperatur der Luft, die in das Luftansaugsystem des Fahrzeugs eintritt.
AmbientAirTemp	`REAL`	Temperatur der umgebenden Luft.
CabInteriorTemp	`REAL`	Temperatur der Luft in der Fahrzeugkabine.
BarometricPress	`REAL`	Absoluter Luftdruck in der Atmosphäre.
Status	`MpJ1939PGNStatusEnum`	Status der Parametergruppennummer.

6.6.1.1.2 MpJ1939AT1IG1Type

Dieser Datentyp beschreibt die PGN AT1IG1 (Nachbehandlung 1/Ansauggas 1).

Parameter	Datentyp	Beschreibung
NOxSnsr1SelfDiagStatus	MpJ1939NOS1SDSEnum	Status des NOx-Sensor ATI1 (Nachbehandlungs-Einspeicherbank 1).
O2Snsr1PreliminaryFMI	INT	Wert des Nachbehandlungs-Ansaugsauerstoffsensor.
SCRInNOxSnsr1PreliminaryFMI	INT	Wert des Nachbehandlungs-Einlass-NOx-Sensor.
Snsr1HtrCtrl	MpJ1939S1HCEnum	Zeigt den Heizstatus während des Aufwärmvorgangs an. Nach Erhalt eines Einschaltbefehls wird der Gassensor gemäß einem vom Hersteller definierten Profil hochgefahren.
Snsr1HtrPreliminaryFMI	INT	Wert der in der Heizung des Einlass-Abgassensors, entweder NOx oder O2, gemessen wurde.
WRPercentO21ReadingStable	MpJ1939WRPO21RSEnum	Zeigt an, ob der prozentuale Sauerstoffmesswert des Nachbehandlungs-Einlassgassensors, entweder NOx oder O2, stabil ist, wie von der Steuerungssoftware des Herstellers in der Abgasbank 1 bestimmt.
SCRInNOx1ReadingStable	MpJ1939SCRINO1RSEnum	Zeigt an, ob der NOx-Messwert des Nachbehandlungs-Einlass-NOx-Sensors stabil ist, wie von der Steuerungssoftware des Herstellers in der Abgasbank 1 bestimmt.
Snsr1AtTemp	MpJ1939S1ATEnum	Zeigt an, ob das Heizelement des Nachbehandlungs-Ansauggassensors, entweder NOx oder O2, für genaue Messungen in der Abgasbank 1 innerhalb des vom Hersteller angegebenen Bereichs liegt.
Snsr1PwrInRange	MpJ1939S1PIREnum	Zeigt an, ob die dem Nachbehandlungs-Einlassgassensor zugeführte Leistung, entweder NOx oder O2, innerhalb der Herstellerspezifikation in Abgasbank 1 liegt.
PercentO21	REAL	Der tatsächliche Oxidationsfaktor des Gases im Abgasstrom.
SCRInNOx1	REAL	Die Menge an kombiniertem NO und NO2 in dem Abgas, die in das Nachbehandlungssystem eintritt.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.3 MpJ1939AT1IG2Type

Dieser Datentyp beschreibt die PGN AT1IG2 (Nachbehandlung 1/Ansauggas 2).

Parameter	Datentyp	Beschreibung
DPFInTempPreliminaryFMI	INT	Wert, der vom Einlasstemperatursensor des Dieselpartikelfilters gemessen wurde.
ExhTemp1PreliminaryFMI	INT	Wert, der vom Sensor für die Abgastemperatur 1 gemessen wurde.
DPFInTemp	REAL	Temperatur von Nebenprodukten der Motorverbrennung, die in den Dieselpartikelfilter in der Abgasbank 1 eintreten.
ExhTemp1	REAL	Wert der vom Abgastemperatursensor gemessen wurde.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.4 MpJ1939AT1IMGType

Dieser Datentyp beschreibt PGN AT1IMG (Nachbehandlung 1/Zwischengas).

Parameter	Datentyp	Beschreibung
DPFIntmdTempPreliminaryFMI	INT	Wert der vom Zwischentemperatursensor des Dieselpartikelfilters gemessen wurde.
DPFDiffPressPreliminaryFMI	INT	Wert der vom Differenzdrucksensor des Dieselpartikelfilters gemessen wurde.
ExhTemp2PreliminaryFMI	INT	Wert der von der Sensorsteuerungssoftware des Herstellers in Abgasbank 1 im Sensor für Abgastemperatur 2 gemessen wurde.

Parameter	Datentyp	Beschreibung
DPFDiffPress	REAL	Abgasdifferenzdruck, gemessen zwischen Ansaug- und Abgastemperatur eines Dieselpartikelfilters in der Abgasbank 1.
DPFIntmdTemp	REAL	Temperatur der Verbrennungsnebenprodukte der Maschine in der Mitte des Dieselpartikelfilters in der Abgasbank 1.
ExhTemp2	REAL	Wert der vom Abgastemperatursensor gemessen wurde.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.5 MpJ1939AT1OG1Type

Dieser Datentyp beschreibt die PGN AT1OG1 (Nachbehandlung 1/Ausgangsgas 1).

Parameter	Datentyp	Beschreibung
O2Snsr1PreliminaryFMI	INT	Wert der vom Sauerstoffsensor des Nachbehandlungsauslasses gemessen wurde.
NOxSnsr1SelfDiagStatus	MpJ1939NOS1SDSEnum	Status des NOx-Sensor ATO1 (Nachbehandlungs-Ausgangsbank 1)
NOxSnsr1PreliminaryFMI	INT	Wert der vom NOx-Sensor des Nachbehandlungsauslasses gemessen wurde.
Snsr1HtrCtrl	MpJ1939S1HCEnum	Zeigt den Heizstatus während des Aufwärmvorgangs an. Nach Erhalt eines Einschaltbefehls wird der Gassensor gemäß einem vom Hersteller definierten Profil hochgefahren.
Snsr1HtrPreliminaryFMI	INT	Gibt den Wert, der in der Heizung des Auslass-Abgassensors, entweder NOx oder O2, gemessen wurde, an.
WRPercentO21ReadingStable	MpJ1939WRPO21RSEnum	Gibt an, ob der prozentuale Sauerstoffmesswert des Nachbehandlungsauslassgassensors, entweder NOx oder O2, stabil ist, wie von der Steuerungssoftware des Herstellers in der Abgasbank 1 bestimmt.
NOx1ReadingStable	MpJ1939NO1RSEnum	Gibt an, ob der NOx-Messwert des Nachbehandlungsauslass-NOx-Sensors stabil ist, wie von der Steuerungssoftware des Herstellers in der Abgasbank 1 bestimmt.
Snsr1AtTemp	MpJ1939S1ATEnum	Gibt an, ob das Heizelement des Nachbehandlungsauslassgassensors, entweder NOx oder O2, für genaue Messungen in der Abgasbank 1 innerhalb des vom Hersteller angegebenen Bereichs liegt.
Snsr1PwrInRange	MpJ1939S1PIREnum	Gibt an, ob die an den Nachbehandlungs-Auslassgassensor gelieferte Leistung, entweder NOx oder O2, innerhalb der Herstellerspezifikation in Abgasbank 1 liegt.
PercentO21	REAL	Der tatsächliche Oxidationsfaktor (Prozent Sauerstoff) des Gases im Abgasstrom.
NOx1	REAL	Die Menge an kombiniertem NO und NO2 in dem Abgas, die das Nachbehandlungssystem verlässt.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.6 MpJ1939AT1OG2Type

Dieser Datentyp beschreibt die PGN AT1OG2 (Nachbehandlung 1/Ausgangsgas 2).

Parameter	Datentyp	Beschreibung
DPFOutExhTempPreliminaryFMI	INT	Wert der vom Sensor für die Auslasstemperatur des Dieselpartikelfilters gemessen wurde.
ExhTemp3PreliminaryFMI	INT	Wert der vom Abgastemperatursensor 3 gemessen wurde.
DPFOutTemp	REAL	Temperatur der Motorverbrennungsnebenprodukte, die das Dieselpartikelfilterabgas in der Abgasbank 1 verlassen.
ExhTemp3	REAL	Wert der vom Abgastemperatursensor gemessen wurde.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.7 MpJ1939AT1S1Type

Dieser Datentyp beschreibt die PGN AT1S1 (Nachbehandlung 1/Service 1).

Parameter	Datentyp	Beschreibung
DPFSootLoadRegenThreshold	REAL	Dieser Parameter gibt den Wert an, der zuerst die DPF-Regeneration in der Nachbehandlung 1 verursacht.
DPFTimeSinceLastActiveRegen	INT	Gibt die Zeit seit dem letzten aktiven Regenerationsereignis des Dieselpartikelfilters 1 an.
DPFAshLoadPercent	INT	Gibt den Aschelastprozentsatz des Dieselpartikelfilters 1 an.
DPFSootLoadPercent	INT	Zeigt den Rußbeladungsprozentsatz des Dieselpartikelfilters 1 an.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.8 MpJ1939AT1T1I1Type

Dieser Datentyp beschreibt PGN AT1T1I1 (Nachbehandlung 1/Dieselabgasflüssigkeitstank 1/Information 1).

Parameter	Datentyp	Beschreibung
Volume	REAL	Anzahl der Softwareidentifikationsbezeichner, die in der Softwareidentifikationsparametergruppe dargestellt sind.
Temp1	INT	Software-Identifikation eines elektronischen Moduls.
Lvl	REAL	Gibt das Level der Dieselabgasfluidspiegelhöhe in mm im Dieselabgasfluidvorratsbehälter für das Nachbehandlungssystem 1 an.
LvlVolPreliminaryFMI	SINT	Wert der vom Füllstandssensor des Dieselabgastanks gemessen wurde.
DEFTankLowLvlInd	MpJ1939DEFTLLIEnum	Gibt die gewünschte Leuchtkraft des Fahrerwarnindikators für Dieselabgase an.
TempPreliminaryFMI	SINT	Gibt die Dieselabgastank-Temperatur für das Nachbehandlungssystem 1 an.
SCROpIndSev	MpJ1939MpJ1939SCROISEnum	Gibt den Status des Bedieneraufforderungssystems für Anomalien mit dem SCR-System an.
Htr	REAL	Prozentsatz der Erwärmung, die auf die Nachbehandlungsvorrichtung für Dieselabgasfluidtanks angewendet wird.
HtrPreliminaryFMI	SINT	Wert, der für das Heizgerät des Dieselabgas-Flüssigkeitstanks für das Nachbehandlungssystem 1 gemessen wurde.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.9 MpJ1939CCVS1Type

Dieser Datentyp beschreibt die PGN CCVS1 (Geschwindigkeitsregelung / Fahrzeuggeschwindigkeit 1).

Parameter	Datentyp	Beschreibung
EngShutdownOverrideSwitch	MpJ1939ESOSEnum	Schaltsignal, das die Position des Übersteuerungsschalters der Motorabschaltung anzeigt. Diese Schalterfunktion ermöglicht es dem Bediener, eine bevorstehende Motorabschaltung zu überschreiben.
EngIdleDecrementSwitch	MpJ1939ETMSEnum	Schaltsignal, das die Position des Motordiagnosetestmodusschalters anzeigt.
EngIdleDecrementSwitch	MpJ1939EIDSEnum	Schaltsignal, das die Position des Leerlauf-Dekrementierschalters anzeigt.
EngIdleIncrementSwitch	MpJ1939EIISEnum	Schaltsignal, das die Position des Leerlaufinkrementschalters anzeigt.
CruiseCtrlStates	MpJ1939CCSEnum	Zustand des Tempomats.
PTOGovernorState	MpJ1939PTOGSEnum	Dieser Parameter wird verwendet, um den aktuellen Zustand oder die Betriebsart des Zapfwellenreglers anzuzeigen.
CruiseCtrlSetSpeed	INT	Geschwindigkeit des Tempomats.
CruiseCtrlAccelerateSwitch	MpJ1939CCASEnum	Gibt an, ob der Tempomat sich in der Position "Beschleunigen" befindet.
CruiseCtrlResumeSwitch	MpJ1939CCRSEnum	Gibt an ob sich der Tempomat in der Position "Fortsetzen" befindet.
CruiseCtrlCoastSwitch	MpJ1939CCCSEnum	Gibt an ob sich der Tempomat in der Position "Verzögerung" befindet.
CruiseCtrlSetSwitch	MpJ1939CCSSEnum	Gibt an ob sich der Tempomat in der Position "Gesetzt" befindet.

Parameter	Datentyp	Beschreibung
ClutchSwitch	MpJ1939CSEnum	Gibt an ob das Kupplungspedal gedrückt wird.
BrakeSwitch	MpJ1939BSEnum	Gibt an ob das Bremspedal betätigt wird.
CruiseCtrlEnableSwitch	MpJ1939CCESEnum	Gibt an ob es ermöglicht werden soll den Tempomat zu verändern.
CruiseCtrlActive	MpJ1939CCAEnum	Gibt an ob der Tempomat aktiviert ist oder nicht.
WheelBasedVehicleSpeed	REAL	Geschwindigkeit des Fahrzeugs, berechnet aus der Rad- oder Heckwellengeschwindigkeit.
ParkBrakeReleaseInhibitRq	MpJ1939PBRIREnum	Die Parkbremsfreigabesperranforderung gibt an, ob die Parkbremse betätigt bleibt.
CruiseCtrlPauseSwitch	MpJ1939CCPSEnum	Gibt die Position des Tempomat-Unterbrechungsschalters an.
ParkingBrakeSwitch	MpJ1939PBSEnum	Gibt den Zustand der Parkbremse an.
TwoSpeedAxleSwitch	MpJ1939TSASEnum	Gibt den Status des aktuellen Achsbereichs an.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.10 MpJ1939CIType

Dieser Datentyp beschreibt die PGN CI (Komponentenidentifikation).

Parameter	Datentyp	Beschreibung
UnitNumber	INT	Gibt die Gerätenummer für das Netzteil des Fahrzeugs an.
SerialNumber	INT	Seriennummer der Komponente.
Model	INT	Modell der Komponente.
Make	INT	Nummer der Komponente welche nach den Richtlinien des "American Trucking Association Vehicle Maintenance Reporting Standard (ATA/VMRS)" erstellt wurde.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.11 MpJ1939CM1Type

Dieser Datentyp beschreibt die PGN CM1 (Kabinenmeldung 1).

Parameter	Datentyp	Beschreibung
SelectedMaxVehicleSpeedLimit	INT	Gibt die vom Benutzer gewählte maximale Fahrzeuggeschwindigkeit an.
RqCabZoneHeating	MpJ1939RCZHEnum	Gibt den Status der Kabinenheizung an.
RqEngZoneHeating	MpJ1939REZHEnum	Gibt den Status der Motorzonenheizung an.
AuxHeaterModeRq	MpJ1939AHMREnum	Gibt den Status der Zusatzheizung an.
EngAutomaticStartEnableSwitch	MpJ1939EASESEnum	Gibt den Status des Leerlaufmanagementsystems an.
AtomaticGearShiftingEnableSwitch	MpJ1939AGSESEnum	Gibt den Status der automatischen Gangschaltung an.
AFTRegenForceSw	MpJ1939AFTRFSEnum	Gibt den Zustand des Schalters an, welcher die Nachbehandlungsregeneration erzwingt.
AFTRegenInhSw	MpJ1939AFTRISEnum	Gibt den Zustand des Schalter an, welcher die Nachbehandlungsregeneration verhindert.
VhclLmtngSpdGovernorEnableSwitch	MpJ1939VLSGESEnum	Gibt an, ob der VLSG ("Vehicle Limiting Speed Governor") eingeschaltet ist.
VhclLmtngSpdGvrnrIncrementSwitch	MpJ1939VLSGISEnum	Schaltsignal, das den VLSG (Vehicle Limiting Speed Governor) erhöht.
VhclLmtngSpdGvrnrDecrementSwitch	MpJ1939VLSGDSEnum	Schaltsignal, das den VLSG (Vehicle Limiting Speed Governor) verringert.
ParkBrakeCmd	MpJ1939PBCEnum	Gibt an, ob die Parkbremse durch ein sekundäres Gerät gesteuert wird.
SeatBeltSwitch	MpJ1939SBSEnum	Gibt den Zustand des Schalter an, der verwendet wird, um festzustellen ob der Sicherheitsgurt angeschnallt ist.
OperatorSeatDirectionSwitch	MpJ1939OSDSEnum	Gibt an, ob sich der Fahrersitz in der vorderen Fahrposition befindet.
BatteryMainSwitchHoldRq	MpJ1939BMSHREnum	Gibt an, ob der Hauptschalter der Batterie gehalten werden muss.

Parameter	Datentyp	Beschreibung
AuxHeaterCoolantPumpRq	MpJ1939AHCPREnum	Gibt an, ob die Kühlwasserpumpe der Zusatzheizung aktiviert werden soll.
CabInteriorTempCmd	REAL	Parameter zum Steuern einer bestimmten Kabineninnentemperatur.
RequestedPercentFanSpeed	REAL	Lüfterdrehzahl des ersten oder einzigen Lüfters im System als Verhältnis des tatsächlichen Lüfterantriebs (aktuelle Drehzahl) zum voll belegten Lüfterantrieb (maximale Lüfterdrehzahl).
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.12 MpJ1939DM01Type

Dieser Datentyp beschreibt die PGN DM01 (Aktive Diagnose-Fehlercodes) .

Parameter	Datentyp	Beschreibung
FlashRedStopLamp	MpJ1939FRSLEnum	Bietet die Möglichkeit die Rote Bremsleuchte blinken zu lassen.
FlashProtectLamp	MpJ1939FPLEnum	Bietet die Möglichkeit die Motorschutzlampe blinken zu lassen.
FlashMalfuncIndicatorLamp	MpJ1939FMILEnum	Bietet die Möglichkeit die Störungsanzeigelampe blinken zu lassen.
FlashAmberWarningLamp	MpJ1939FAWLEnum	Bietet die Möglichkeit die gelbe Warnlampe blinken zu lassen.
MalfunctionIndicatorLampStatus	MpJ1939MILSEnum	Status der Störungsanzeigelampe.
AmberWarningLampStatus	MpJ1939AWLSEnum	Status der gelben Warnlampe. Bei einem Fehler muss das Fahrzeug nicht sofort gestoppt werden.
RedStopLampState	MpJ1939RSLSEnum	Status der roten Bremsleuchte. Bei einem Fehler muss das Fahrzeug sofort gestoppt werden.
ProtectLampStatus	MpJ1939PLSEnum	Status der Motorschutzlampe.
DTC5	UDINT	Diagnose-Fehlercode 5
DTC4	UDINT	Diagnose-Fehlercode 4
DTC3	UDINT	Diagnose-Fehlercode 3
DTC2	UDINT	Diagnose-Fehlercode 2
DTC1	UDINT	Diagnose-Fehlercode 1
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.13 MpJ1939DPFC1Type

Dieser Datentyp beschreibt die PGN DPFC1 (Dieselpartikelfilter - Steuerung 1).

Parameter	Datentyp	Beschreibung
DslPrtcltFltrActvRgnrtnInhbt5629	MpJ1939DPFARI5629Enum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters aufgrund des niedrigen Abgasdrucks an.
AFT1DPFCondNotMetForActiveRegen	MpJ1939AFT1DPFCNMFAREnum	Gibt an, ob die Regeneration des Dieselpartikelfilters 1 bei den aktuellen Motorbetriebsbedingungen gehemmt ist oder nicht.
HydrocarbonDoserPurgingEnable	MpJ1939HDPEEnum	Definiert die Spülfreigabe des Kohlenwasserstoffdosierers.
DslPrtcltFltrActvRgnrtnFrcdSttus	MpJ1939DPFARFSEnum	Zeigt den erzwungenen Ausführungsstatus der Regeneration des Dieselpartikelfilters an.
ExhaustSystemHighTempLampCmd	MpJ1939ESHTLCEnum	Gibt des Status der Abgassystem-Hochtemperaturlampe an.
DslPrtcltFltrAtmtcActvRgnrtnIntt	MpJ1939DPFAARIEnum	Zeigt die Konfiguration der automatischen Regenerierung des Dieselpartikelfilters an.
DslPrtcltFltrActvRgnrtnInhbt3717	MpJ1939DPFARI3717Enum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da die Fahrzeuggeschwindigkeit kleiner als die zulässige Fahrzeuggeschwindigkeit ist.
DslPrtcltFltrActvRgnrtnInhbtdDTE	MpJ1939DPFARIDTEEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da der Motor nicht aufgewärmt wird.
DslPrtcltFltrActvRgnrtnInhbt3715	MpJ1939DPFARI3715Enum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters aufgrund einer permanenten Systemsperre an.
DslPrtcltFltrActvRgnrtnInhbtdDTT	MpJ1939DPFARIDTTEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters aufgrund einer vorübergehenden Systemsperre an.

Parameter	Datentyp	Beschreibung
DslPrtcltFltrActvRgnrtnInhbt3713	MpJ1939DPFARI3713Enum	Zeigt den Status der aktiven Regenerationshemmung des Dieselpartikelfilters aufgrund eines System-Timeouts an.
DslPrtcltFltrActvRgnrtnInhbt3712	MpJ1939DPFARI3712Enum	Zeigt den Status der aktiven Regenerationshemmung des Dieselpartikelfilters an, der auf einen aktiven Systemfehler zurückzuführen ist.
DslPrtcltFltrActvRgnrtnInhbtdDTL	MpJ1939DPFARIDTLEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da die Abgastemperatur zu niedrig ist.
DslPrtcltFltrActvRgnrtnInhbt3710	MpJ1939DPFARI3710Enum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da die Parkbremse nicht eingestellt ist.
DslPrtcltFltrActvRgnrtnInhbtdDTV	MpJ1939DPFARIDTVEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da die Fahrzeuggeschwindigkeit über einem zulässigen Grenzwert liegt.
DslPrtcltFltrActvRgnrtnInhbtdDTO	MpJ1939DPFARIDTOEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da das Getriebe nicht im Leerlauf ist.
DslPrtcltFltrActvRgnrtnInhbtdDTA	MpJ1939DPFARIDTAEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da das Gaspedal im Leerlauf ist.
DslPrtcltFltrActvRgnrtnInhbtdDTP	MpJ1939DPFARIDTPEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da die Zapfwelle aktiv ist.
DslPrtcltFltrActvRgnrtnInhbtdDTS	MpJ1939DPFARIDTSEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da die Betriebsbremse aktiviert ist.
DslPrtcltFltrActvRgnrtnInhbtdDTC	MpJ1939DPFARIDTCEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, da die Kupplung ausgerückt ist.
DslPrtcltFltrActvRgnrtnInhbtdDTI	MpJ1939DPFARIDTIEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, der durch den Dieselpartikelfilter-Regenerationsunterdrückungsschalter verursacht wird.
DslPrtcltFltrActvRgnrtnInhbtdStt	MpJ1939DPFARISEnum	Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an.
AFTDPFStatus	MpJ1939AFTDPFSEnum	Zeigt den Zustand des Regenerationsbedarfs und der Dringlichkeit des Dieselpartikelfilters an.
AFTDPFActiveRegenStatus	MpJ1939AFTDPFARSEnum	Zeigt den Zustand der aktiven Regeneration des Dieselpartikelfilters an.
AFTDPFPassiveRegenStatus	MpJ1939AFTDPFPRSEnum	Zeigt den Zustand der passiven Regeneration des Dieselpartikelfilters an.
DieselParticulateFilterLampCmd	MpJ1939DPFLCEnum	Befehl zum Steuern der Dieselpartikelfilterlampe.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.14 MpJ1939EC1Type

Dieser Datentyp beschreibt die PGN EC1 (Motorkonfiguration 1).

Parameter	Datentyp	Beschreibung
SupportTSC1CtrlPurposeGroup4	INT	Dieser Parameter gibt an, welche TSC1-Steuerungszwecke in Gruppe 4 unterstützt werden.
SupportTSC1CtrlPurposeGroup3	INT	Dieser Parameter gibt an, welche TSC1-Steuerungszwecke in Gruppe 3 unterstützt werden.
SupportTSC1CtrlPurposeGroup2	INT	Dieser Parameter gibt an, welche TSC1-Steuerungszwecke in Gruppe 2 unterstützt werden.
SupportTSC1CtrlPurposeGroup1	INT	Dieser Parameter gibt an, welche TSC1-Steuerungszwecke in Gruppe 1 unterstützt werden.
SupportVariableRateTSC1Message	INT	Dieser Parameter gibt an, welche TSC1-Übertragungsraten von der Motor-ECU zusätzlich zu der erforderlichen 10-ms-Übertragungsrate für temporäre Antriebsstrangsteuerzwecke unterstützt werden.

Parameter	Datentyp	Beschreibung
EngDefaultTorqueLimit	DINT	Gibt den Begleitparameter zu Übertragungsdrehmomentgrenze an.
EngMomentOfInertia	REAL	Trägheitsmoment des Motors.
EngExRngRqstdSpdCtrlRngUpprLimit	REAL	Die maximale Motordrehzahl.
EngRqstdTrqueCtrlRangeUpperLimit	INT	Das maximale Motordrehmoment, das der Motor zulässt, wenn er in einem Drehmomentsteuerungs- / Grenzmodus arbeitet.
EngRqstdTrqueCtrlRangeLowerLimit	INT	Das minimale Motordrehmoment, das der Motor zulässt, wenn er in einem Drehmomentsteuerungs- / Grenzmodus arbeitet.
EngRqstdSpeedCtrlRangeUpperLimit	INT	Die maximale Motordrehzahl unabhängig von der Last, die der Motor zulässt, wenn er in einem Drehzahlregelungs- / Grenzmodus arbeitet.
EngMaxMomentaryOverrideTimeLimit	INT	Die minimale Motordrehzahl, die der Motor zulässt, wenn er in einem Geschwindigkeitsregelungs- / Grenzmodus arbeitet.
EngMxMmentaryOverrideSpeedPoint7	REAL	Die maximale Motordrehzahl über dem Leerlauf, die von der Motorsteuerung während einer momentanen hohen Leerlaufüberbrückung zugelassen wird.
EngReferenceTorque	REAL	Dieser Parameter ist der 100% Referenzwert für alle definierten angegebenen Motordrehmomentparameter. Er ist nur einmal definiert und ändert sich nicht, wenn ein anderes Motordrehmomentkennfeld gültig wird.
EngGain	DINT	Wert des Endspeed-Regler.
EngSpeedAtHighIdlePoint6	REAL	Motordrehzahl des hohen Leerlaufs (Punkt 6) des Motordrehmomentkennfelds.
EngPercentTorqueAtPoint5	REAL	Die Drehmomentgrenze, die das verfügbare Motordrehmoment angibt, das vom Motor an den Punkten 5 des Motorkennfelds bereitgestellt werden kann.
EngSpeedAtPoint5	INT	Motordrehzahl von Punkt 5 des Motordrehmomentkennfelds.
EngPercentTorqueAtPoint4	REAL	Die Drehmomentgrenze, die das verfügbare Motordrehmoment angibt, das vom Motor an den Punkten 4 des Motorkennfelds bereitgestellt werden kann.
EngSpeedAtPoint4	INT	Motordrehzahl von Punkt 4 des Motordrehmomentkennfelds.
EngPercentTorqueAtPoint3	INT	Die Drehmomentgrenze, die das verfügbare Motordrehmoment angibt, das vom Motor bei Punkt 3 Motorkennfelds bereitgestellt werden kann.
EngSpeedAtPoint3	REAL	Motordrehzahl von Punkt 3 des Motordrehmomentkennfelds.
EngPercentTorqueAtPoint2	INT	Die Drehmomentgrenze, die das verfügbare Motordrehmoment angibt, das vom Motor bei Punkt 2 des Motorkennfelds bereitgestellt werden kann.
EngSpeedAtPoint2	REAL	Motordrehzahl von Punkt 2 des Motordrehmomentkennfelds.
EngPercentTorqueAtIdlePoint1	INT	Die Drehmomentgrenze gibt das verfügbare Motordrehmoment an, das der Motor im Leerlauf liefern kann.
EngSpeedAtIdlePoint1	REAL	Stationäre niedrige Leerlaufdrehzahl des Motors, die Einflüsse aufgrund der Motortemperatur (nach dem Einschalten) und anderer stationärer Änderungen (Kalibrierungsversätze, Sensorausfälle usw.) enthält.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.15 MpJ1939EEC1Type

Dieser Datentyp beschreibt die PGN EEC1 (Elektronische Motorsteuerung 1).

Parameter	Datentyp	Beschreibung
EngDemandPercentTorque	SINT	Die geforderte Drehmomentabgabe des Motors durch alle dynamischen internen Eingänge.
EngStarterMode	MpJ1939ESMEnum	Gibt an, warum ein Start nicht funktioniert hat.
SrcAddrssOfCntrlIngDvcForEngCtrl	USINT	Die Quelladresse des SAE J1939-Geräts, das den Motor derzeit steuert.
EngSpeed	REAL	Tatsächliche Motordrehzahl, die über einen minimalen Kurbelwellenwinkel von 720 Grad geteilt durch die Anzahl der Zylinder berechnet wird.
ActualEngPercentTorque	SINT	Das berechnete Ausgangsdrehmoment des Motors. Die Daten werden in angegebenem Drehmoment in Prozent des Referenzmotordrehmoments übertragen.
DriversDemandEngPercentTorque	SINT	Die angeforderte Drehmomentausgabe des Motors durch den Fahrer.
ActlEngPrcntTorqueHighResolution	REAL	Dieser Parameter zeigt ein zusätzliches Drehmoment in Prozent des Referenzmotordrehmoments an.
EngTorqueMode	SINT	Zustandssignal, das anzeigt, welcher Motordrehmomentmodus gerade das Drehmoment erzeugt, begrenzt oder regelt.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.16 MpJ1939EEC2Type

Dieser Datentyp beschreibt die PGN EEC2 (Elektronische Motorsteuerung 2).

Parameter	Datentyp	Beschreibung
EstPumpingPercentTorque	SINT	Das berechnete Drehmoment gibt die geschätzte Größe des Drehmomentverlustes aufgrund des Motorlüftungssystems an.
ActlMaxAvailableEngPercentTorque	REAL	Dies ist die maximale Menge an Drehmoment, die der Motor sofort als Prozentsatz des Referenzmotordrehmoments abgeben kann.
SCRThermalManagementActive	MpJ1939SCRTMAEnum	Gibt an, ob die Abgastemperaturen für die Regeneration des SCR-Nachbehandlungssystems oder für die Vorbereitung der Regeneration des SCR-Nachbehandlungssystems erhöht wurden.
DPFThermalManagementActive	MpJ1939DPFTMAEnum	Zeigt an, dass die Abgastemperaturen für die Regeneration des Dieselpartikelfilter-Nachbehandlungssystems oder für die Vorbereitung der Regeneration des Dieselpartikel-Nachbehandlungssystems erhöht wurden.
MmntaryEngMaxPowerEnableFeedback	MpJ1939MEMPEFEnum	Gibt an, ob die maximale Motorleistung momentan aktiviert wird.
VhcleAccelerationRateLimitStatus	MpJ1939VARLSEnum	Status des Systems, mit dem die maximale Fahrzeugvorwärtsbeschleunigung begrenzt wird.
AccelPedalPos2	REAL	Das Verhältnis der Ist-Position der zweiten analogen Motordrehzahl- / Drehmomentanforderungs-Eingabevorrichtung (wie etwa eines Gaspedals oder eines Gashebels) zu der maximalen Position der Eingabevorrichtung.
RemoteAccelPedalPos	REAL	Das Verhältnis der tatsächlichen Position der entfernten analogen Motordrehzahl- / Drehmomentanforderungs-Eingabevorrichtung (wie etwa eines Gaspedals oder eines Gashebels) zu der maximalen Position der Eingabevorrichtung.
EngPercentLoadAtCurrentSpeed	SINT	Das Verhältnis des tatsächlichen Motor-Prozent-Drehmoments (angezeigt) zum maximalen angezeigten Drehmoment, das bei der gegenwärtigen Motordrehzahl verfügbar ist, wird während des Motorbremsens auf ein Null-Drehmoment begrenzt.

Parameter	Datentyp	Beschreibung
AccelPedalPos1	REAL	Das Verhältnis der tatsächlichen Position der analogen Motordrehzahl- / Drehmomentanforderungs-Eingabevorrichtung (wie etwa eines Gaspedals oder eines Gashebels) zu der maximalen Position der Eingabevorrichtung.
AccelPedal2LowIdleSwitch	MpJ1939AP2LISEnum	Schaltsignal, das den Zustand des unteren Leerlaufschalters des Gaspedals 2 anzeigt.
RoadSpeedLimitStatus	MpJ1939RSLiSEnum	Status des Systems zur Begrenzung der maximalen Fahrzeuggeschwindigkeit.
AccelPedalKickdownSwitch	MpJ1939APKSEnum	Schaltsignal, das anzeigt, ob der Gaspedal-Kickdown-Schalter geöffnet oder geschlossen ist.
AccelPedal1LowIdleSwitch	MpJ1939AP1LISEnum	Schaltsignal, das den Zustand des niedrigen Leerlaufschalters des Gaspedals 1 anzeigt.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.17 MpJ1939EEC3Type

Dieser Datentyp beschreibt PGN EEC3 (Elektronische Motorsteuerung 3).

Parameter	Datentyp	Beschreibung
AFT2ExhDewPoint	MpJ1939AFT2EDPEnum	Gibt an, ob die Temperatur auf der Auslassseite der Nachbehandlung den Taupunkt überschritten hat.
AFT2InDewPoint	MpJ1939AFT2IDPEnum	Gibt an, ob die Temperatur auf der Einlassseite des Nachbehandlungssystems den Taupunkt überschritten hat.
AFT1ExhDewPoint	MpJ1939AFT1EDPEnum	Gibt an, ob die Temperatur auf der Auslassseite der Nachbehandlung den Taupunkt überschritten hat.
AFT1InDewPoint	MpJ1939AFT1IDPEnum	Gibt an, ob die Temperatur auf der Einlassseite des Nachbehandlungssystems den Taupunkt überschritten hat.
AFT1ExhGasMFR	REAL	Gemessene Abgasmasse vor dem Nachbehandlungssystem in Abgasbank 1 und 2.
EstEngPrsiticLossesPercentTorque	INT	Das berechnete Drehmoment gibt den geschätzten Betrag des Drehmomentverlustes aufgrund von Verschleißteilen an, wie z. B. Kühlgebläse, Luftkompressor, Klimaanlage usw. Es wird als Prozentsatz des Motorreferenzdrehmoments ausgedrückt.
EngnsDsrdOprtngSpdAsymmtryAdjstm	INT	Dieses Byte wird in den Übertragungsgangauswahlroutinen verwendet und zeigt die Vorliebe des Motors für niedrigere gegenüber höheren Motordrehzahlen an, falls seine gewünschte Drehzahl nicht erreichbar sein sollte.
EnginesDesiredOperatingSpeed	REAL	Eine Anzeige des Motors für die optimale Betriebsdrehzahl des Motors für die aktuellen vorhandenen Bedingungen.
NominalFrictionPercentTorque	INT	Das Drehmoment gibt an, wie viel Drehmoment vom Grundmotor selbst benötigt wird, und zwar um den Verlustdrehmoment des Zubehörs.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.18 MpJ1939EFL_P1Type

Dieser Datentyp beschreibt PGN EFL_P1 (Motorölstand / Druck 1).

Parameter	Datentyp	Beschreibung
EngCoolantLevel	REAL	Verhältnis des Flüssigkeitsvolumens im Motorkühlsystem zum gesamten Kühlsystemvolumen.
EngCoolantPress	INT	Gibt den Messdruck der Flüssigkeit im Motorkühlsystem an.
EngCrankcasePress	REAL	Erste Instanz des Messdrucks im Kurbelgehäuse des Motors.
EngOilPress	INT	Druck des Öls im Motorschmiersystem. Wird von der Ölpumpe bereitgestellt.

Parameter	Datentyp	Beschreibung
EngOilLevel	REAL	Verhältnis des aktuellen Volumens des Motoröls zum maximal erforderlichen Volumen.
EngExCrankcaseBlowbyPress	REAL	Gibt den Kurbelgehäusedurchblasdruck an.
EngFuelDeliveryPress	INT	Tankdruck des Kraftstoffs im System, wie er von der Versorgungspumpe an die Einspritzpumpe geliefert wird.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.19 MpJ1939ET1Type

Dieser Datentyp beschreibt PGN ET1 (Motortemperatur 1).

Parameter	Datentyp	Beschreibung
EngIntercoolerThermostatOpening	REAL	Position des Thermostats. Wird verwendet um die Temperatur des Motorladeluftkühlers zu regulieren.
EngIntercoolerTemp	INT	Die Temperatur der Flüssigkeit, die im Ladeluftkühler zu finden ist.
EngTurboOilTemp	REAL	Temperatur des Turboladerschmierstoffs.
EngOilTemp1	REAL	Temperatur des Motorschmiermittels.
EngFuelTemp1	INT	Temperatur des Kraftstoffs.
EngCoolantTemp	INT	Die Temperatur der Flüssigkeit, die im Motorkühlsystem zu finden ist.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.20 MpJ1939HOURSType

Dieser Datentyp beschreibt PGN HOURS (Motorstunden).

Parameter	Datentyp	Beschreibung
EngTotalRevolutions	REAL	Anzahl von Umdrehungen der Motorkurbelwelle während ihres Betriebs.
EngTotalHoursOfOperation	REAL	Betriebszeit des Motors.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.21 MpJ1939IC1Type

Dieser Datentyp beschreibt PGN IC1 (Ansaug - / Abgasbedingungen 1).

Parameter	Datentyp	Beschreibung
EngCoolantFilterDiffPress	REAL	Gibt die Änderung des Kühlmitteldrucks an.
EngExhaustGasTemp	REAL	Temperatur der Verbrennungsnebenprodukte, die den Motor verlassen.
EngAirFilter1DiffPress	REAL	Gibt die Änderung des Motorlufsystemdrucks an.
EngAirIntakePress	INT	Absoluter Luftdruck am Eingangsanschluss zum Ansaugkrümmer oder zum Luftkasten.
EngIntakeManifold1Temp	INT	Temperatur der Vorverbrennungsluft im Ansaugkrümmer des Motorluftversorgungssystems.
EngIntakeManifold1Press	INT	Ergebniss der Messdruckmessung des Ansaugkrümmers.
AFT1DPFInPress	REAL	Abgasdruck als Folge der Partikelansammlung auf dem Filter im Abgasstrom.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.22 MpJ1939LFC1Type

Dieser Datentyp beschreibt PGN LFC1 (Kraftstoffverbrauch (Flüssigkeit) 1).

Parameter	Datentyp	Beschreibung
EngTotalFuelUsed	REAL	Kraftstoffmenge, die während des Fahrzeugbetriebs verbraucht wird.
EngTripFuel	REAL	Kraftstoff, der während der gesamten oder einer Teilstrecke verbraucht wird.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.23 MpJ1939LFE1Type

Dieser Datentyp beschreibt PGN LFE1 (Kraftstoffverbrauch 1).

Parameter	Datentyp	Beschreibung
EngThrottleValve2Pos	REAL	Gibt die erfasste Positionsrückmeldung eines Sensors an, welcher den Wert eines Kraftstoff/Luft-Gemisches misst.
EngThrottleValve1Pos	REAL	Gibt die erfasste Positionsrückmeldung eines Sensors an, welcher den Wert eines Kraftstoff/Luft-Gemisches misst.
EngAverageFuelEconomy	REAL	Durchschnitt des momentanen Kraftstoffverbrauchs.
EngInstantaneousFuelEconomy	REAL	Aktuelle Kraftstoffersparnis bei aktueller Fahrzeuggeschwindigkeit.
EngFuelRate	REAL	Treibstoffverbrauch pro Zeiteinheit.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.24 MpJ1939SHUTDNType

Dieser Datentyp beschreibt PGN SHUTDN (Stilllegung).

Parameter	Datentyp	Beschreibung
ClntLvlEngPrtctionShutdownStatus	MpJ1939CLEPSSEnum	Gibt die Konfiguration der Motorschutzabschaltung des Kühlmittelstands an.
PTOShutdownHasShutdownEng	MpJ1939PTOSHSEEnum	Statussignal, das angibt, ob der Motor durch das PTO-Abschaltsystem abgeschaltet wurde oder nicht.
EngAirShutoffStatus	MpJ1939EASSEnum	Zustandssignal, das die tatsächlich gemessene Position der Luftabschaltung anzeigt.
EngOverspeedTest	MpJ1939EOTEnum	Gibt den Status des Motorüberdrehzahltestsignal, wie es durch das berichtende ECM gemessen wird.
EngAirShutoffCmdStatus	MpJ1939EASCSEnum	Zeigt an, wenn der Air Shutoff-Treiberausgang angesteuert wird.
EngAlarmOutputCmdStatus	MpJ1939EAOCSEnum	Zeigt an, wann der Alarmtreiberausgang angesteuert wird.
EngAlarmAcknowledge	MpJ1939EAAEnum	Zeigt an, ob die Quittierung des Motoralarms aktiv ist oder nicht.
EngProtectionSystemConfig	MpJ1939EPSCEnum	Parameter, der die Konfiguration des Motorabschaltsystems angibt.
EngProtectionSystemTimerState	MpJ1939EPSTSEnum	Gibt den Modus des Timer-Systems des Motorschutzsystems an.
EngProtectionSystemTimerOverride	MpJ1939EPSTOEnum	Gibt den Status der Übersteuerungsfunktion des Motorschutzsystem-Timers an.
EngPrtctnSystmApprachingShutdown	MpJ1939EPSASEnum	Statussignal, das anzeigt, ob die Motorabschaltung unmittelbar bevorsteht.
EngPrtectionSystemHasShutdownEng	MpJ1939EPSHSEEnum	Statussignal, das anzeigt, ob das Motorschutzsystem den Motor abgeschaltet hat oder nicht.
EngWaitToStartLamp	MpJ1939EWTSLEnum	Zeigt an, ob der Motor zu kalt zum starten ist oder nicht.
RefrigerantHighPressSwitch	MpJ1939RHPSEnum	Gibt die Stellung des Hochdruckschalters im Kühlmittelkreislauf einer Klimaanlage an.
RefrigerantLowPressSwitch	MpJ1939RLPSEnum	Gibt die Stellung des Niederdruckschalters im Kühlmittelkreislauf einer Klimaanlage an.
A_CHighPressFanSwitch	MpJ1939ACHPFSEnum	Schaltsignal, das anzeigt, ob der Druck im Kühlmittelkreislauf einer Klimaanlage zu hoch ist und der Ventilator eingeschaltet werden muss.
EngIdleShutdownTimerFunction	MpJ1939EISTFEnum	Parameter, der die Konfiguration des Leerlaufabschalt-Zeitgebersystems angibt.
EngIdleShutdownTimerState	MpJ1939EISTSEnum	Statussignal, das den aktuellen Betriebsmodus des Leerlaufabschalt-Zeitgebersystems anzeigt.
EngIdleShutdownTimerOverride	MpJ1939EISTOEnum	Statussignal, das den Status der Übersteuerungsfunktion des Leerlaufabschalt-Zeitgebersystems anzeigt.
EngIdleShutdownDriverAlertMode	MpJ1939EISDAMEnum	Statussignal, das den Status des Fahrerwarnmodus des Leerlaufabschalt-Zeitgebersystems anzeigt.
EngIdleShutdownHasShutdownEng	MpJ1939EISHSEEnum	Statussignal, das identifiziert, ob der Motor durch das Leerlaufabschalt-Zeitgebersystem abgeschaltet wurde oder nicht.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.25 MpJ1939SOFTType

Dieser Datentyp beschreibt PGN SOFT (Software Identifikation).

Parameter	Datentyp	Beschreibung
NmbrOfSftwreIdentificationFields	INT	Anzahl der Softwareidentifikationsbezeichner, die in der Softwareidentifikationsparametergruppe dargestellt sind.
SoftwareIdentification	INT	Software-Identifikation eines elektronischen Moduls.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.26 MpJ1939TSC1Type

Dieser Datentyp beschreibt PGN TSC1 (Drehmoment- / Drehzahlregelung 1).

Parameter	Datentyp	Beschreibung
MessageChecksum	INT	Nachrichtenprüfsumme. Die Nachrichtenprüfsumme wird verwendet, um den Signalpfad von dem sendenden Gerät zu dem empfangenden Gerät zu verifizieren.
MessageCounter	INT	Nachrichtenzähler. Der Nachrichtenzähler wird verwendet, um Situationen zu erfassen, bei denen die übertragende ECU-Fehlfunktion die gleiche Zeitspanne wiederholt.
EngRequestedTorqueHighResolution	MpJ1939ERTHREnum	Dieser Parameter zeigt ein zusätzliches Drehmoment in Prozent des Referenzmotordrehmoments an.
TSC1CtrlPurpose	MpJ1939TSC1CPEnum	Gibt an welchen Steuermodus das sendende Gerät verwendet, um den TSC1-Befehl zu generieren.
TSC1TransRate	MpJ1939TSC1TREnum	Gibt die Übertragungsrate an mit der das sendende Gerät die TSC1-Nachricht überträgt.
EngRequestedTorque_TorqueLimit	INT	Parameter, der dem Motor in der Drehmoment- / Drehzahlsteuerungsmeldung zum Steuern oder Begrenzen des Ausgangsdrehmoments bereitgestellt wird.
EngRequestedSpeed_SpeedLimit	REAL	Parameter, die dem Motor von externen Quellen in der Drehmoment- / Drehzahlregelungsmeldung bereitgestellt werden.
OverrideCtrlModePriority	MpJ1939OCMPEnum	Definiert die Priorität des Override-Steuerungsmodus, der in der Drehmomentmeldung empfangen wird.
EngRequestedSpeedCtrlConditions	MpJ1939ERSCCEnum	Der Parameter teilt dem Motorsteuerungssystem die Reglerkennlinien mit, die während der Geschwindigkeitssteuerung verwendet werden soll.
EngOverrideCtrlMode	MpJ1939EOCMEnum	Der Override-Steuermodus definiert, welche Art von Befehl verwendet wird.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.27 MpJ1939VEP1Type

Dieser Datentyp beschreibt die PGN VEP1 (Fahrzeugstrom 1).

Parameter	Datentyp	Beschreibung
KeyswitchBatteryPotential	REAL	Am Eingang der elektronischen Steuereinheit gemessenes Batteriepotential, das über einen Schlüsselschalter oder eine ähnliche Schaltvorrichtung zugeführt wird.
BatteryPotential_PowerInput1	REAL	Dieser Parameter misst die erste Quelle des Batteriepotentials, gemessen am Eingang der ECU / des Aktors usw., die von einer oder mehreren Batterien kommt, unabhängig vom Abstand zwischen der Komponente und der Batterie.
ChargingSystemPotential	REAL	Elektrisches Potential gemessen am Ladesystemausgang.
AltCurrent	INT	Wert des elektrischen Stromflusses von der Lichtmaschine.
NetBatteryCurrent	INT	Netto-Stromfluss in / aus der ersten Batterie oder dem ersten Batteriesatz, der zum Starten des Motors, zum Anzünden und zum Zünden (SLI) verwendet wird.
Enable	BOOL	Aktivieren der Fahrzeugelektrischen Leistung.
Status	MpJ1939PGNStatusEnum	Status der Parametergruppennummer.

6.6.1.1.28 MpJ1939GenericInfoType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
Diag	MpJ1939DiagType	Eine Diagnosestruktur für die StatusID.

6.6.1.1.29 MpJ1939ReceiveInfoType

Mit diesem Datentyp wird für den Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpJ1939DiagType	Eine Diagnosestruktur für die StatusID.
ReceivedBytes	UINT	Anzahl der erhaltenen Bytes.

6.6.1.1.30 MpJ1939SpecificRequestInfoType

Mit diesem Datentyp wird für den Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpJ1939DiagType	Eine Diagnosestruktur für die StatusID.
ReceivedBytes	UINT	Anzahl der erhaltenen Bytes.

6.6.1.1.31 MpJ1939TransmitInfoType

Mit diesem Datentyp wird für den Funktionsbaustein zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpJ1939DiagType	Eine Diagnosestruktur für die StatusID.

6.6.1.1.32 MpJ1939StatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	MpJ1939ErrorEnum	Fehlerkode des Funktionsbausteins.
Severity	MpComSeveritiesEnum	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler).
Code	UINT	Code der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden.

6.6.1.1.33 MpJ1939DiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	MpJ1939StatusIDType	StatusID Diagnose Struktur.

6.6.1.2 Enumeratoren

6.6.1.2.1 MpJ1939ACHPFSEnum

Dieser Aufzählungsdatentyp zeig an, ob der Druck im Kühlmittelkreislauf einer Klimaanlage zu hoch ist und der Ventilator eingeschaltet werden muss.

Aufzählung	Beschreibung
mpJ1939_ACHPFS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_ACHPFS_ERROR	Fehlerzustand.
mpJ1939_ACHPFS_PRESSURE_HIGH	Druck ist zu hoch.
mpJ1939_ACHPFS_PRESSURE_NORMAL	Druck ist normal.

6.6.1.2.2 MpJ1939AFT1DPFCNMFAREnum

Dieser Aufzählungsdatentyp gibt den Zustand der Dieselpartikelfilter 1-Regeneration bei den aktuellen Motorbetriebsbedingungen an.

Aufzählung	Beschreibung
mpJ1939_AFT1DPFCNMFAR_NOT_AVAILA	Status nicht verfügbar.
mpJ1939_AFT1DPFCNMFAR_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_AFT1DPFCNMFAR_INHIBIT	Gibt an, dass die aktive Dieselpartikelfilter -Regeneration gehemmt ist.
mpJ1939_AFT1DPFCNMFAR_NOT_INHIBI	Gibt an, dass die aktive Dieselpartikelfilter -Regeneration nicht gehemmt ist.

6.6.1.2.3 MpJ1939AFT1EDPEnum

Dieser Aufzählungsdatentyp gibt an, ob die Temperatur auf der Auslassseite der Nachbehandlung den Taupunkt überschritten hat.

Aufzählung	Beschreibung
mpJ1939_AFT1EDP_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFT1EDP_ERROR	Fehlerzustand.
mpJ1939_AFT1EDP_EXCEEDED	Taupunkt überschritten.
mpJ1939_AFT1EDP_NOT_EXCEED	Taupunkt ist nicht überschritten.

6.6.1.2.4 MpJ1939AFT1IDPEnum

Dieser Aufzählungsdatentyp gibt an, ob die Temperatur auf der Einlassseite des Nachbehandlungssystems den Taupunkt überschritten hat.

Aufzählung	Beschreibung
mpJ1939_AFT1IDP_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFT1IDP_ERROR	Fehlerzustand.
mpJ1939_AFT1IDP_EXCEEDED	Taupunkt ist überschritten.
mpJ1939_AFT1IDP_NOT_EXCEED	Taupunkt ist nicht überschritten.

6.6.1.2.5 MpJ1939AFT2EDPEnum

Dieser Aufzählungsdatentyp gibt an ob die Temperatur auf der Auslassseite der Nachbehandlung den Taupunkt überschritten hat.

Aufzählung	Beschreibung
mpJ1939_AFT2EDP_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFT2EDP_ERROR	Fehlerzustand.
mpJ1939_AFT2EDP_EXCEEDED	Taupunkt wurde überschritten.
mpJ1939_AFT2EDP_NOT_EXCEED	Taupunkt ist nicht überschritten.

6.6.1.2.6 MpJ1939AFT2IDPEnum

Dieser Aufzählungsdatentyp gibt an, ob die Temperatur auf der Einlassseite des Nachbehandlungssystems den Taupunkt überschritten hat.

Aufzählung	Beschreibung
mpJ1939_AFT2IDP_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFT2IDP_ERROR	Fehlerzustand.
mpJ1939_AFT2IDP_EXCEEDED	Taupunkt ist überschritten.
mpJ1939_AFT2IDP_NOT_EXCEED	Taupunkt ist nicht überschritten.

6.6.1.2.7 MpJ1939AFTDPFARSEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regeneration des Dieselpartikelfilters (AFTDPFARS) an.

Aufzählung	Beschreibung
mpJ1939_AFTDPFARS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFTDPFARS_NEEDED	AFTDPFARS wird benötigt.
mpJ1939_AFTDPFARS_ACTIVE	AFTDPFARS aktiv.
mpJ1939_AFTDPFARS_NOT_ACTIVE	AFTDPFARS nicht aktiv.

6.6.1.2.8 MpJ1939AFTDPFPRSEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der passiven Regeneration des Dieselpartikelfilters (AFTDPFPRS) an.

Aufzählung	Beschreibung
mpJ1939_AFTDPFPRS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFTDPFPRS_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_AFTDPFPRS_ACTIVE	AFTDPFPRS aktiv.
mpJ1939_AFTDPFPRS_NOT_ACTIVE	AFTDPFPRS nicht aktiv.

6.6.1.2.9 MpJ1939AFTDPFSEnum

Dieser Aufzählungsdatentyp zeigt den Zustand des Regenerationsbedarfs und der Dringlichkeit des Dieselpartikelfilters (AFTDPFS) an.

Aufzählung	Beschreibung
mpJ1939_AFTDPFS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFTDPFS_RESERVED6	Reserviert für SAE-Zuordnung.
mpJ1939_AFTDPFS_RESERVED5	Reserviert für SAE-Zuordnung.
mpJ1939_AFTDPFS_RESERVED4	Reserviert für SAE-Zuordnung.
mpJ1939_AFTDPFS_HIGHEST_LEVEL	AFTDPFS wird benötigt (höchstes Level)
mpJ1939_AFTDPFS_MODERATE_LEVEL	AFTDPFS wird benötigt (mittleres Level)
mpJ1939_AFTDPFS_LOWEST_LEVEL	AFTDPFS wird benötigt (niedriges Level)
mpJ1939_AFTDPFS_NOT_NEEDED	AFTDPFS wird nicht benötigt.

6.6.1.2.10 MpJ1939AFTRFSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Schalters an, welcher die Nachbehandlungsregeneration erzwingt.

Aufzählung	Beschreibung
mpJ1939_AFTRFS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFTRFS_ERROR	Fehlerzustand.
mpJ1939_AFTRFS_ACTIVE	Schalter ist aktiv.
mpJ1939_AFTRFS_NOT_ACTIVE	Schalter ist deaktiviert.

6.6.1.2.11 MpJ1939AFTRISEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Schalter an, welcher die Nachbehandlungsregeneration verhindert.

Aufzählung	Beschreibung
mpJ1939_AFTRIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AFTRIS_ERROR	Fehlerzustand.
mpJ1939_AFTRIS_ACTIVE	Schalter ist aktiv.
mpJ1939_AFTRIS_NOT_ACTIVE	Schalter ist deaktiviert.

6.6.1.2.12 MpJ1939AGSESEnum

Dieser Aufzählungsdatentyp gibt den Status der automatischen Gangschaltung an.

Aufzählung	Beschreibung
mpJ1939_AGSES_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AGSES_ERROR	Fehlerzustand.
mpJ1939_AGSES_AUTO_ENABLED	Gangschaltung ist aktiviert.
mpJ1939_AGSES_AUTO_DISABLED	Gangschaltung ist deaktiviert.

6.6.1.2.13 MpJ1939AHCPREnum

Dieser Aufzählungsdatentyp gibt an, ob die Kühlwasserpumpe der Zusatzheizung aktiviert werden soll.

Aufzählung	Beschreibung
mpJ1939_AHCPR_DONT_CARE	Kühlwasserpumpe wartet.
mpJ1939_AHCPR_RESERVED	Reserviert.
mpJ1939_AHCPR_ACTIVATE	Kühlwasserpumpe muss aktiviert werden.
mpJ1939_AHCPR_DEACTIVATE	Kühlwasserpumpe muss deaktiviert werden.

6.6.1.2.14 MpJ1939AHMREnum

Dieser Aufzählungsdatentyp gibt den Status der Zusatzheizung an.

Aufzählung	Beschreibung
mpJ1939_AHMR_DONT_CARE	Zusatzheizung wartet.
mpJ1939_AHMR_RESERVED	Reserviert.
mpJ1939_AHMR_HEATER_PUMP_UPKEEP	Heizungspumpe läuft.
mpJ1939_AHMR_NORMAL_MODE	Zusatzheizung ist im "Normalen-Modus".
mpJ1939_AHMR_ECONOMY_MODE	Zusatzheizung ist im "Economy-Modus".
mpJ1939_AHMR_OFF_HAZARDOUS	Zusatzheizung ist ausgeschaltet aufgrund von gefährlichen Materialien.
mpJ1939_AHMR_DEACTIVATE_AUX	Zusatzheizung deaktiviert.

6.6.1.2.15 MpJ1939AP1LISEnum

Dieser Aufzählungsdatentyp gibt den Zustand des niedrigen Leerlaufschalters des Gaspedals 1 anzeigt.

Aufzählung	Beschreibung
mpJ1939_AP1LIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AP1LIS_ERROR	Fehlerzustand.
mpJ1939_AP1LIS_IN_LOW_IDLE	Leerlaufschalter befindet sich im niedrigen Leerlaufzustand.
mpJ1939_AP1LIS_NOT_IN_LOW_IDLE	Leerlaufschalter befindet sich nicht im niedrigen Leerlaufzustand.

6.6.1.2.16 MpJ1939AP2LISEnum

Dieser Aufzählungsdatentyp gibt den Zustand des unteren Leerlaufschalters des Gaspedals 2 an.

Aufzählung	Beschreibung
mpJ1939_AP2LIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AP2LIS_ERROR	Fehlerzustand.
mpJ1939_AP2LIS_IN_LOW_IDLE	Leerlaufschalters befindet sich im niedrigen Leerlaufzustand.
mpJ1939_AP2LIS_NOT_IN_LOW_IDLE	Leerlaufschalters befindet sich nicht im niedrigen Leerlaufzustand.

6.6.1.2.17 MpJ1939APKSEnum

Dieser Aufzählungsdatentyp gibt an ob der Gaspedal-Kickdown-Schalter geöffnet oder geschlossen ist.

Aufzählung	Beschreibung
mpJ1939_APKS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_APKS_ERROR	Fehlerzustand.
mpJ1939_APKS_KICKDOWN_ACTIVE	Gaspedal-Kickdown-Schalter aktiv.
mpJ1939_APKS_KICKDOWN_PASSIVE	Gaspedal-Kickdown-Schalter nicht aktiv.

6.6.1.2.18 MpJ1939AWLSEnum

Dieser Aufzählungsdatentyp zeigt den Status der gelben Warnlampe an.

Aufzählung	Beschreibung
mpJ1939_AWLS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_AWLS_RESERVED	Gelbe Warnlampe ist reserviert.
mpJ1939_AWLS_LAMP_ON	Gelbe Warnlampe ist an.
mpJ1939_AWLS_LAMP_OFF	Gelbe Warnlampe ist ausgeschaltet.

6.6.1.2.19 MpJ1939BMSHREnum

Dieser Aufzählungsdatentyp gibt an, ob der Hauptschalter der Batterie gehalten werden muss.

Aufzählung	Beschreibung
mpJ1939_BMSHR_DONT_CARE	Schalter wartet.
mpJ1939_BMSHR_UNDEFINED	Schalter befindet sich im undefinierten Zustand.
mpJ1939_BMSHR_HOLD_BATTERY	Schalter muss betätigt werden.
mpJ1939_BMSHR_RELEASE_BATTERY	Schalter muss nicht betätigt werden.

6.6.1.2.20 MpJ1939BSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Bremspedal an.

Aufzählung	Beschreibung
mpJ1939_BS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_BS_ERROR	Fehlerzustand.
mpJ1939_BS_DEPRESSED	Bremspedal ist gedrückt.
mpJ1939_BS_RELEASED	Bremspedal ist nicht gedrückt.

6.6.1.2.21 MpJ1939CCAEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Tempomats an.

Aufzählung	Beschreibung
mpJ1939_CCA_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_CCA_ERROR	Fehlerzustand.
mpJ1939_CCA_ON	Tempomat ist eingeschaltet.
mpJ1939_CCA_OFF	Tempomat ist ausgeschaltet.

6.6.1.2.22 MpJ1939CCASEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Tempomat-Schalters an.

Aufzählung	Beschreibung
mpJ1939_CCAS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_CCAS_ERROR	Fehlerzustand.
mpJ1939_CCAS_IN_POS_ACCEL	Schalter befindet sich in der "Beschleunigen" Position.
mpJ1939_CCAS_NOT_IN_POS_ACCEL	Schalter befindet sich nicht in der "Beschleunigen" Position.

6.6.1.2.23 MpJ1939CCCSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Tempomat-Schalters an.

Aufzählung	Beschreibung
mpJ1939_CCCS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_CCCS_ERROR	Fehlerzustand.
mpJ1939_CCCS_IN_POS_COAST	Schalter in Position "Verzögerung".
mpJ1939_CCCS_NOT_IN_POS_COAST	Schalter nicht in Position "Verzögerung".

6.6.1.2.24 MpJ1939CCDCEnum

Dieser Aufzählungsdatentyp gibt an, ob der Tempomat von einer anderen Quelle als den Fahrer gesteuert werden soll.

Aufzählung	Beschreibung
mpJ1939_CCDC_DONT_CARE	Tempomat wartet.
mpJ1939_CCDC_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_CCDC_NOT_ALLOWED	Tempomat wird nicht von einer anderen Quelle gesteuert.
mpJ1939_CCDC_IS_ALLOWED	Tempomat wird von einer anderen Quelle gesteuert.

6.6.1.2.25 MpJ1939CCESEnum

Dieser Aufzählungsdatentyp gibt an wie der Tempomat verändert werden kann.

Aufzählung	Beschreibung
mpJ1939_CCES_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_CCES_ERROR	Fehlerzustand.
mpJ1939_CCES_ENABLED	Tempomat kann verändert werden.
mpJ1939_CCES_DISABLED	Veränderung des Tempomats deaktiviert.

6.6.1.2.26 MpJ1939CCPCEnum

Dieser Aufzählungsdatentyp gibt an, ob die Möglichkeit den Tempomat von einer anderen Quelle als den Fahrer steuern zu lassen, vorübergehend deaktiviert werden soll.

Aufzählung	Beschreibung
mpJ1939_CCPC_DONT_CARE	Tempomat wartet.
mpJ1939_CCPC_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_CCPC_NOT_ALLOWED	Möglichkeit, Tempomat vorübergehend von anderer Quelle zu steuern ist nicht deaktiviert.
mpJ1939_CCPC_ALLOWED	Möglichkeit, Tempomat vorübergehend von anderer Quelle zu steuern ist deaktiviert.

6.6.1.2.27 MpJ1939CCPSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Tempomat-Unterbrechungsschalters an.

Aufzählung	Beschreibung
mpJ1939_CCPS_TAKE_NO_ACTION	Tempomat-Unterbrechungsschalters
mpJ1939_CCPS_ERROR_INDICATOR	Fehlerzustand.
mpJ1939_CCPS_ON	Tempomat-Unterbrechungsschalters ist eingeschalten.
mpJ1939_CCPS_OFF	Tempomat-Unterbrechungsschalters ist ausgeschalten.

6.6.1.2.28 MpJ1939CCRCEnum

Dieser Aufzählungsdatentyp gibt an, ob der Tempomat von einer anderen Quelle als den Fahrer gesteuert werden soll.

Aufzählung	Beschreibung
mpJ1939_CCRC_DONT_CARE	Tempomat wartet.
mpJ1939_CCRC_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_CCRC_REQUESTED	Tempomat wird von einer anderer Quelle gesteuert.
mpJ1939_CCRC_NOT_REQUESTED	Tempomat wird nicht von einer anderer Quelle gesteuert.

6.6.1.2.29 MpJ1939CCRSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Tempomat-Schalters an.

Aufzählung	Beschreibung
mpJ1939_CCRS_NOT_AVAILABLE	Schalter ist nicht verfügbar.
mpJ1939_CCRS_ERROR	Schalter ist im Fehlerzustand.
mpJ1939_CCRS_IN_POS_RESUME	Schalter befindet sich in der Position "Fortsetzen".
mpJ1939_CCRS_NOT_IN_POS_RESUME	Schalter befindet sich nicht in der Position "Fortsetzen".

6.6.1.2.30 MpJ1939CCSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Tempomats an.

Aufzählung	Beschreibung
mpJ1939_CCS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_CCS_ACCELERATOR_OVERRIDE	Tempomat ist in Beschleunigungsüberschreibung.
mpJ1939_CCS_SET	Tempomat in Position "Gesetzt".
mpJ1939_CCS_RESUME	Tempomat in Position "Fortfahren".
mpJ1939_CCS_DECELERATE	Tempomat in Position "Verzögern".
mpJ1939_CCS_ACCELERATE	Tempomat in Position "Beschleunigen".
mpJ1939_CCS_HOLD	Tempomat in Position "Halten".
mpJ1939_CCS_OFF	Tempomat ist ausgeschaltet.

6.6.1.2.31 MpJ1939CCSSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Tempomat-Schalters an.

Aufzählung	Beschreibung
mpJ1939_CCSS_NOT_AVAILABLE	Schalter ist nicht verfügbar.
mpJ1939_CCSS_ERROR	Schalter ist im Fehlerzustand.
mpJ1939_CCSS_IN_POS_SET	Schalter ist in Position "Gesetzt".
mpJ1939_CCSS_NOT_IN_POS_SET	Schalter ist nicht in Position "Gesetzt".

6.6.1.2.32 MpJ1939CLEPSSEnum

Dieser Aufzählungsdatentyp gibt die Konfiguration der Motorschutzabschaltung des Kühlmittelstands an.

Aufzählung	Beschreibung
mpJ1939_CLEPSS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_CLEPSS_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_CLEPSS_ENABLED	Motorschutzabschaltung des Kühlmittelstands aktiviert.
mpJ1939_CLEPSS_DISABLED	Motorschutzabschaltung des Kühlmittelstands nicht aktiviert.

6.6.1.2.33 MpJ1939CSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Kupplungspedals an.

Aufzählung	Beschreibung
mpJ1939_CS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_CS_ERROR	Fehlerzustand.
mpJ1939_CS_DEPRESSED	Kupplungspedal ist gedrückt.
mpJ1939_CS_RELEASED	Kupplungspedal ist nicht gedrückt.

6.6.1.2.34 MpJ1939DPFAARIEnum

Dieser Aufzählungsdatentyp gibt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFAARI) an, da die Fahrzeuggeschwindigkeit kleiner als die zulässige Fahrzeuggeschwindigkeit ist.

Aufzählung	Beschreibung
mpJ1939_DPFAARI_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFAARI_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFAARI_ENABLED	DPFAARI aktiviert.
mpJ1939_DPFAARI_NOT_ENABLED	DPFAARI nicht aktiviert.

6.6.1.2.35 MpJ1939DPFARFSEnum

Dieser Aufzählungsdatentyp gibt den Ausführungsstatus der Regeneration des Diesespartikelfilters (DPFARFS) an.

Aufzählung	Beschreibung
mpJ1939_DPFARFS_RESERVED3	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARFS_SERVICE_TOOL	DPFARFS aktiviert durch Service Tool
mpJ1939_DPFARFS_SWITCH	DPFARFS aktiviert durch Schalter.

Aufzählung	Beschreibung
mpJ1939_DPFARFS_NOT_ACTIVE	DPFARFS nicht aktiviert.
mpJ1939_DPFARFS_NOT_AVAILABLE	DPFARFS nicht verfügbar.
mpJ1939_DPFARFS_RESERVED6	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARFS_RESERVED5	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARFS_RESERVED4	Reserviert für SAE-Zuordnung.

6.6.1.2.36 MpJ1939DPFARI3710Enum

Dieser Aufzählungsdatentyp Zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARI3710) an, da die Parkbremse nicht eingestellt ist.

Aufzählung	Beschreibung
mpJ1939_DPFARI3710_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARI3710_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARI3710_INHIBITED	DPFARI3710 ist gehemmt.
mpJ1939_DPFARI3710_NOT_INHIBITED	DPFARI3710 ist nicht gehemmt.

6.6.1.2.37 MpJ1939DPFARI3712Enum

Dieser Aufzählungsdatentyp zeigt den Status der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARI3712) an, der auf einen aktiven Systemfehler zurückzuführen ist.

Aufzählung	Beschreibung
mpJ1939_DPFARI3712_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARI3712_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARI3712_INHIBITED	DPFARI3712 ist gehemmt.
mpJ1939_DPFARI3712_NOT_INHIBITED	DPFARI3712 ist nicht gehemmt.

6.6.1.2.38 MpJ1939DPFARI3713Enum

Dieser Aufzählungsdatentyp zeigt den Status der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARI3713) aufgrund eines System-Timeouts an.

Aufzählung	Beschreibung
mpJ1939_DPFARI3713_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARI3713_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARI3713_INHIBITED	DPFARI3713 ist gehemmt.
mpJ1939_DPFARI3713_NOT_INHIBITED	DPFARI3713 ist nicht gehemmt.

6.6.1.2.39 MpJ1939DPFARI3715Enum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARI3715) aufgrund einer permanenten Systemsperre an.

Aufzählung	Beschreibung
mpJ1939_DPFARI3715_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARI3715_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARI3715_INHIBITED	DPFARI3715 gehemmt.
mpJ1939_DPFARI3715_NOT_INHIBITED	DPFARI3715 nicht gehemmt.

6.6.1.2.40 MpJ1939DPFARI3717Enum

Dieser Aufzählungsdatentyp gibt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARI3717) an, da die Fahrzeuggeschwindigkeit kleiner als die zulässige Fahrzeuggeschwindigkeit ist.

Aufzählung	Beschreibung
mpJ1939_DPFARI3717_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARI3717_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARI3717_INHIBITED	DPFARI3717 aktiv.
mpJ1939_DPFARI3717_NOT_INHIBITED	DPFARI3717 nicht aktiv.

6.6.1.2.41 MpJ1939DPFARI5629Enum

Dieser Aufzählungsdatentyp gibt des Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters aufgrund des niedrigen Abgasdrucks (DPFARI) an.

Aufzählung	Beschreibung
mpJ1939_DPFARI5629_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARI5629_RESERVED	DPFARI ist reserviert.
mpJ1939_DPFARI5629_INHIBIT	DPFARI gehemmt.
mpJ1939_DPFARI5629_NOT_INHIBIT	DPFARI nicht gehemmt.

6.6.1.2.42 MpJ1939DPFARIDTAEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTA) an, da das Gaspedal im Leerlauf ist.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTA_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTA_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTA_INHIBITED	DPFARIDTA ist gehemmt.
mpJ1939_DPFARIDTA_NOT_INHIBITED	DPFARIDTA ist nicht gehemmt.

6.6.1.2.43 MpJ1939DPFARIDTCEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTC) an, da die Kupplung ausgerückt ist.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTC_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTC_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTC_INHIBITED	DPFARIDTC ist gehemmt.
mpJ1939_DPFARIDTC_NOT_INHIBITED	DPFARIDTC ist nicht gehemmt.

6.6.1.2.44 MpJ1939DPFARIDTEEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTE) an, da der Motor nicht aufgewärmt wird.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTE_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTE_RESERVED	Für SAE-Zuordnung reserviert.
mpJ1939_DPFARIDTE_INHIBITED	DPFARIDTE gehemmt.
mpJ1939_DPFARIDTE_NOT_INHIBITED	DPFARIDTE ist nicht gehemmt.

6.6.1.2.45 MpJ1939DPFARIDTIEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters an, der durch den Dieselpartikelfilter-Regenerationsunterdrückungsschalter (DPFARIDTI) verursacht wird.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTI_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTI_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTI_INHIBITED	DPFARIDTI ist gehemmt.
mpJ1939_DPFARIDTI_NOT_INHIBITED	DPFARIDTI ist nicht gehemmt.

6.6.1.2.46 MpJ1939DPFARIDTLEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTL) an, da die Abgastemperatur zu niedrig ist.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTL_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTL_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTL_INHIBITED	DPFARIDTL ist gehemmt.
mpJ1939_DPFARIDTL_NOT_INHIBITED	DPFARIDTL ist nicht gehemmt.

6.6.1.2.47 MpJ1939DPFARIDTOEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTO) an, da das Getriebe nicht im Leerlauf ist.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTO_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTO_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTO_INHIBITED	DPFARIDTO ist gehemmt.
mpJ1939_DPFARIDTO_NOT_INHIBITED	DPFARIDTO ist nicht gehemmt.

6.6.1.2.48 MpJ1939DPFARIDTPEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTP) an, da die Zapfwelle aktiv ist.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTP_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTP_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTP_INHIBITED	DPFARIDTP ist gehemmt.
mpJ1939_DPFARIDTP_NOT_INHIBITED	DPFARIDTP ist nicht gehemmt.

6.6.1.2.49 MpJ1939DPFARIDTSEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTS) an, da die Betriebsbremse aktiviert ist.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTS_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTS_INHIBITED	DPFARIDTS ist gehemmt.
mpJ1939_DPFARIDTS_NOT_INHIBITED	DPFARIDTS ist nicht gehemmt.

6.6.1.2.50 MpJ1939DPFARIDTTEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTT) aufgrund einer vorübergehenden Systemsperre an.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTT_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTT_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTT_INHIBITED	DPFARIDTT ist gehemmt.
mpJ1939_DPFARIDTT_NOT_INHIBITED	DPFARIDTT ist nicht gehemmt.

6.6.1.2.51 MpJ1939DPFARIDTVEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIDTV) an, da die Fahrzeuggeschwindigkeit über einem zulässigen Grenzwert liegt.

Aufzählung	Beschreibung
mpJ1939_DPFARIDTV_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIDTV_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIDTV_INHIBITED	DPFARIDTV ist gehemmt.
mpJ1939_DPFARIDTV_NOT_INHIBITED	DPFARIDTV ist nicht gehemmt.

6.6.1.2.52 MpJ1939DPFARISEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der aktiven Regenerationshemmung des Dieselpartikelfilters (DPFARIS) an.

Aufzählung	Beschreibung
mpJ1939_DPFARIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFARIS_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFARIS_INHIBITED	DPFARIS ist gehemmt.
mpJ1939_DPFARIS_NOT_INHIBITED	DPFARIS ist nicht gehemmt.

6.6.1.2.53 MpJ1939DPFLCEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der Dieselpartikelfilterlampe an.

Aufzählung	Beschreibung
mpJ1939_DPFLC_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DPFLC_RESERVED6	Reserviert für SAE-Zuordnung.
mpJ1939_DPFLC_RESERVED5	Reserviert für SAE-Zuordnung.
mpJ1939_DPFLC_ON_FAST_BLINK	Dieselpartikelfilterlampe blinkt.
mpJ1939_DPFLC_RESERVED3	Reserviert für SAE-Zuordnung.
mpJ1939_DPFLC_RESERVED2	Reserviert für SAE-Zuordnung.
mpJ1939_DPFLC_ON_SOLID	Dieselpartikelfilterlampe leuchtet.
mpJ1939_DPFLC_OFF	Dieselpartikelfilterlampe ist ausgeschaltet.

6.6.1.2.54 MpJ1939DPFTMAEnum

Dieser Aufzählungsdatentyp zeigt den Zustand der Abgastemperaturen für die Regeneration des Dieselpartikelfilter-Nachbehandlungssystems oder für die Vorbereitung der Regeneration des Dieselpartikel-Nachbehandlungssystems (DPFTMA) an.

Aufzählung	Beschreibung
mpJ1939_DPFTMA_DONT_CARE	DPFTMA wartet.
mpJ1939_DPFTMA_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_DPFTMA_ACTIVE	DPFTMA aktiv.
mpJ1939_DPFTMA_NOT_ACTIVE	DPFTMA nicht aktiv.

6.6.1.2.55 MpJ1939EAAEnum

Dieser Aufzählungsdatentyp zeigt an, ob die Quittierung des Motoralarms aktiv ist oder nicht.

Aufzählung	Beschreibung
mpJ1939_EAA_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EAA_ERROR	Quittierung des Motoralarms befindet sich im Fehlerzustand.
mpJ1939_EAA_ACTIVE	Quittierung des Motoralarms aktiv.
mpJ1939_EAA_NOT_ACTIVE	Quittierung des Motoralarms nicht aktiv.

6.6.1.2.56 MpJ1939EAOCSEnum

Dieser Aufzählungsdatentyp zeigt an, wann der Alarmtreiberausgang angesteuert wird.

Aufzählung	Beschreibung
mpJ1939_EAOCS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EAOCS_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_EAOCS_ACTIVE	Alarmtreiberausgang aktiv.
mpJ1939_EAOCS_NOT_ACTIVE	Alarmtreiberausgang nicht aktiv.

6.6.1.2.57 MpJ1939EASCSEnum

Dieser Aufzählungsdatentyp zeigt an, wenn der Air Shutoff-Treiberausgang angesteuert wird.

Aufzählung	Beschreibung
mpJ1939_EASCS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EASCS_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_EASCS_ENABLED	Air Shutoff-Treiberausgang aktiviert.
mpJ1939_EASCS_DISABLED	Air Shutoff-Treiberausgang nicht aktiviert.

6.6.1.2.58 MpJ1939EASESEnum

Dieser Aufzählungsdatentyp gibt den Status des Leerlaufmanagementsystems an.

Aufzählung	Beschreibung
mpJ1939_EASES_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EASES_ERROR	Fehlerzustand.
mpJ1939_EASES_ON_STATE	Leerlaufmanagementsystems ist eingeschaltet.
mpJ1939_EASES_OFF_STATE	Leerlaufmanagementsystems ist ausgeschaltet.

6.6.1.2.59 MpJ1939EASSEnum

Dieser Aufzählungsdatentyp ob die tatsächlich gemessene Position der Luftabschaltung anzeigt wird oder nicht.

Aufzählung	Beschreibung
mpJ1939_EASS_ERROR	Fehlerzustand.
mpJ1939_EASS_RESTRICTED	Position ist nicht erlaubt.
mpJ1939_EASS_ALLOWED	Position ist erlaubt.
mpJ1939_EASS_NOT_AVAILABLE	Position nicht verfügbar.

6.6.1.2.60 MpJ1939EIDSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Leerlauf-Dekrementierschalters an.

Aufzählung	Beschreibung
mpJ1939_EIDS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EIDS_ERROR	Fehlerzustand.
mpJ1939_EIDS_ON	Schalter ist eingeschaltet.
mpJ1939_EIDS_OFF	Schalter ist ausgeschaltet.

6.6.1.2.61 MpJ1939EIISEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Leerlaufinkrementschalters an.

Aufzählung	Beschreibung
mpJ1939_EIIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EIIS_ERROR	Fehlerzustand.
mpJ1939_EIIS_ON	Schalter ist eingeschaltet.
mpJ1939_EIIS_OFF	Schalter ist ausgeschaltet.

6.6.1.2.62 MpJ1939EISDAMEnum

Dieser Aufzählungsdatentyp zeigt den Status des Fahrerwarnmodus des Leerlaufabschalt-Zeitgebersystems an.

Aufzählung	Beschreibung
mpJ1939_EISDAM_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EISDAM_ERROR	Fehlerzustand.
mpJ1939_EISDAM_ACTIVE	Fahrerwarnmodus des Leerlaufabschalt-Zeitgebersystems ist aktiv.
mpJ1939_EISDAM_INACTIVE	Fahrerwarnmodus des Leerlaufabschalt-Zeitgebersystems ist nicht aktiv.

6.6.1.2.63 MpJ1939EISHSEEnum

Dieser Aufzählungsdatentyp gibt an, ob der Motor durch das Leerlaufabschalt-Zeitgebersystem abgeschaltet wurde oder nicht.

Aufzählung	Beschreibung
mpJ1939_EISHSE_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EISHSE_ERROR	Fehlerzustand.
mpJ1939_EISHSE_YES	Motor wurde abgeschaltet.
mpJ1939_EISHSE_NO	Motor wurde nicht abgeschaltet.

6.6.1.2.64 MpJ1939EISTFEnum

Dieser Aufzählungsdatentyp gibt die Konfiguration des Leerlaufabschalt-Zeitgebersystems an.

Aufzählung	Beschreibung
mpJ1939_EISTF_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EISTF_ERROR	Fehlerzustand.
mpJ1939_EISTF_ENABLED	Leerlaufabschalt-Zeitgebersystem ist eingeschaltet.
mpJ1939_EISTF_DISABLED	Leerlaufabschalt-Zeitgebersystem ist ausgeschaltet.

6.6.1.2.65 MpJ1939EISTOEnum

Dieser Aufzählungsdatentyp zeigt den Status der Übersteuerungsfunktion des Leerlaufabschalt-Zeitgebersystems an.

Aufzählung	Beschreibung
mpJ1939_EISTO_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EISTO_ERROR	Fehlerzustand.
mpJ1939_EISTO_ACTIVE	Übersteuerungsfunktion des Leerlaufabschalt-Zeitgebersystems ist aktiv.
mpJ1939_EISTO_INACTIVE	Übersteuerungsfunktion des Leerlaufabschalt-Zeitgebersystems ist nicht aktiv.

6.6.1.2.66 MpJ1939EISTSEnum

Dieser Aufzählungsdatentyp zeigt den aktuellen Betriebsmodus des Leerlaufabschalt-Zeitgebersystems an.

Aufzählung	Beschreibung
mpJ1939_EISTS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EISTS_ERROR	Fehlerzustand.
mpJ1939_EISTS_ACTIVE	Leerlaufabschalt-Zeitgebersystems aktiv.
mpJ1939_EISTS_INACTIVE	Leerlaufabschalt-Zeitgebersystems nicht aktiv.

6.6.1.2.67 MpJ1939EOCMEnum

Dieser Aufzählungsdatentyp definiert welche Art von Befehl für den Override-Steuermodus verwendet wird.

Aufzählung	Beschreibung
mpJ1939_EOCM_SPEED	Drehzahl- / Drehmomentbegrenzung
mpJ1939_EOCM_TORQUE_CONTROL	Drehmomentregelung
mpJ1939_EOCM_SPEED_CONTROL	Geschwindigkeitskontrolle
mpJ1939_EOCM_OVERRIDE_DISABLED	Override-Steuermodus deaktiviert.

6.6.1.2.68 MpJ1939EOTEnum

Dieser Aufzählungsdatentyp gibt den Status des Motorüberdrehzahltestsignal, wie es durch das berichtende ECM gemessen wird.

Aufzählung	Beschreibung
mpJ1939_EOT_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EOT_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_EOT_ACTIVE	Motorüberdrehzahltestsignal aktiv.
mpJ1939_EOT_NOT_ACTIVE	Motorüberdrehzahltestsignal nicht aktiv.

6.6.1.2.69 MpJ1939EPSASEnum

Dieser Aufzählungsdatentyp anzeigt an, ob die Motorabschaltung unmittelbar bevorsteht.

Aufzählung	Beschreibung
mpJ1939_EPSAS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EPSAS_ERROR	Fehlerzustand.
mpJ1939_EPSAS_APPROACHING	Motorabschaltung steht unmittelbar bevor.
mpJ1939_EPSAS_NOT_APPROACHING	Motorabschaltung steht nicht unmittelbar bevor.

6.6.1.2.70 MpJ1939EPSCEnum

Dieser Aufzählungsdatentyp gibt die Konfiguration des Motorabschaltsystems an.

Aufzählung	Beschreibung
mpJ1939_EPSC_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EPSC_ERROR	Fehlerzustand.
mpJ1939_EPSC_ENABLED	Motorabschaltsystems ist aktiviert.
mpJ1939_EPSC_DISABLED	Motorabschaltsystems ist deaktiviert.

6.6.1.2.71 MpJ1939EPSHSEEnum

Dieser Aufzählungsdatentyp zeigt an, ob das Motorschutzsystem den Motor abgeschaltet hat oder nicht.

Aufzählung	Beschreibung
mpJ1939_EPSHSE_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EPSHSE_ERROR	Fehlerzustand.
mpJ1939_EPSHSE_YES	Motorschutzsystem hat den Motor abgeschaltet.
mpJ1939_EPSHSE_NO	Motorschutzsystem hat den Motor nicht abgeschaltet.

6.6.1.2.72 MpJ1939EPSTOEnum

Dieser Aufzählungsdatentyp gibt den Status der Übersteuerungsfunktion des Motorschutzsystem-Timers an.

Aufzählung	Beschreibung
mpJ1939_EPSTO_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EPSTO_ERROR	Fehlerzustand.
mpJ1939_EPSTO_ACTIVE	Übersteuerungsfunktion aktiv.
mpJ1939_EPSTO_INACTIVE	Übersteuerungsfunktion nicht aktiv.

6.6.1.2.73 MpJ1939EPSTSEnum

Dieser Aufzählungsdatentyp gibt den Modus des Timer-Systems des Motorschutzsystems an.

Aufzählung	Beschreibung
mpJ1939_EPSTS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EPSTS_ERROR	Fehlerzustand.
mpJ1939_EPSTS_ACTIVE	Timer-Systems aktiv.
mpJ1939_EPSTS_INACTIVE	Timer-Systems nicht aktiv.

6.6.1.2.74 MpJ1939ERSCCEnum

Dieser Aufzählungsdatentyp definiert die Reglerkennlinien, die während der Geschwindigkeitssteuerung (ERSCC) verwendet werden soll.

Aufzählung	Beschreibung
mpJ1939_ERSCC_ENGAGED_AND3	Antriebsstrang ist aktiviert und im Sperrzustand 2.
mpJ1939_ERSCC_ENGAGED_AND2	Antriebsstrang ist aktiviert und im Sperrzustand 1.
mpJ1939_ERSCC_DISENGA_NONLOCKUP1	Antriebsstrang ist nicht aktiviert und befindet sich nicht im Sperrzustand 1.
mpJ1939_ERSCC_DISENGA_NONLOCKUP2	Antriebsstrang ist nicht aktiviert und befindet sich nicht im Sperrzustand 2.

6.6.1.2.75 MpJ1939ERTHREnum

Dieser Aufzählungsdatentyp gibt an welches zusätzliche Drehmoment in Prozent des Referenzmotordrehmoments verwendet werden soll.

Aufzählung	Beschreibung
mpJ1939_ERTHR_0875	Es wird 0,875% verwendet.
mpJ1939_ERTHR_0125	Es wird 0,125% verwendet.
mpJ1939_ERTHR_0000	Drehmoment soll nicht erhöht werden.

6.6.1.2.76 MpJ1939ESHTLCEnum

Dieser Aufzählungsdatentyp gibt den Status der Abgassystem-Hochtemperaturlampe an.

Aufzählung	Beschreibung
mpJ1939_ESHTLC_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_ESHTLC_RESERVED6	Reserviert für SAE-Zuordnung.
mpJ1939_ESHTLC_RESERVED5	Reserviert für SAE-Zuordnung.
mpJ1939_ESHTLC_RESERVED4	Reserviert für SAE-Zuordnung.
mpJ1939_ESHTLC_RESERVED3	Reserviert für SAE-Zuordnung.
mpJ1939_ESHTLC_RESERVED2	Reserviert für SAE-Zuordnung.
mpJ1939_ESHTLC_ON	Lampe ist eingeschaltet.
mpJ1939_ESHTLC_OFF	Lampe ist ausgeschaltet.

6.6.1.2.77 MpJ1939ESMEnum

Dieser Aufzählungsdatentyp gibt an, warum ein Start nicht funktioniert hat.

Aufzählung	Beschreibung
mpJ1939_ESM_INHIBIT_OVERTEMP	Starter gesperrt wegen Überhitzung.
mpJ1939_ESM_INHIBIT_IMMOBILIZER	Starter aufgrund von Wegfahrsperre gesperrt.
mpJ1939_ESM_INHIBIT_ENGAGED	Starter gesperrt aufgrund eines eingerückten Antriebsstrangs oder einer anderen Getriebehemmung.
mpJ1939_ESM_INHIBIT_ENG_NOT_RDY	Starter gesperrt da Motor noch nicht bereit ist.
mpJ1939_ESM_INHIBIT_ENG_RUNNING	Starter gesperrt wegen laufendem Motor
mpJ1939_ESM_FINISHED	Start wurde durchgeführt.
mpJ1939_ESM_ACTIVE_GEAR_ENGAGED	Starter aktiv, Gang ist eingelegt.
mpJ1939_ESM_ACTIVE_GEAR_NOT_ENGA	Starter aktiv, Gang ist nicht eingelegt.
mpJ1939_ESM_NOT_REQUESTED	Starter wurde nicht angefragt.
mpJ1939_ESM_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_ESM_ERROR	Fehlerzustand.
mpJ1939_ESM_ERROR_LEGACY	Starter befindet sich in einem alten Fehlerzustand.
mpJ1939_ESM_INHIBITED	Starter ist gesperrt.
mpJ1939_ESM_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_ESM_INHIBIT_ACTIVE_INDUC	Starter gesperrt durch aktiven SCR-Anstoß.
mpJ1939_ESM_INHIBIT_AIR_SHT_VALV	Starter gesperrt, da das Ansaugluftabsperrventil aktiv ist.

6.6.1.2.78 MpJ1939ESOSEnum

Dieser Aufzählungsdatentyp gibt des Zustand des Schalters an.

Aufzählung	Beschreibung
mpJ1939_ESOS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_ESOS_ERROR	Fehlerzustand.
mpJ1939_ESOS_ON	Schalter ist eingeschaltet.
mpJ1939_ESOS_OFF	Schalter ist ausgeschaltet.

6.6.1.2.79 MpJ1939ETMSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Motordiagnosetestmodusschalters an.

Aufzählung	Beschreibung
mpJ1939_ETMS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_ETMS_ERROR	Fehlerzustand.
mpJ1939_ETMS_ON	Schalter ist eingeschaltet.
mpJ1939_ETMS_OFF	Schalter ist ausgeschaltet.

6.6.1.2.80 MpJ1939EWTSLEnum

Dieser Aufzählungsdatentyp zeigt an, ob der Motor zu kalt zum starten ist oder nicht.

Aufzählung	Beschreibung
mpJ1939_EWTSL_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_EWTSL_ERROR	Fehlerzustand.
mpJ1939_EWTSL_ON	Motor kann gestartet werden.
mpJ1939_EWTSL_OFF	Motor kann nicht gestartet werden.

6.6.1.2.81 MpJ1939FAWLEnum

Dieser Aufzählungsdatentyp definiert das Verhalten der gelben Warnlampe.

Aufzählung	Beschreibung
mpJ1939_FAWL_UNAVAILABLE	Gelbe Warnlampe blinkt nicht.
mpJ1939_FAWL_RESERVED	Gelbe Warnlampe reserviert.
mpJ1939_FAWL_FAST_FLASH	Gelbe Warnlampe blinkt mit >=2Hz (50% Einschaltdauer)
mpJ1939_FAWL_SLOW_FLASH	Gelbe Warnlampe blinkt mit 1Hz (50% Einschaltdauer)

6.6.1.2.82 MpJ1939FMILEnum

Dieser Aufzählungsdatentyp definiert das Verhalten der Störungsanzeigelampe.

Aufzählung	Beschreibung
mpJ1939_FMIL_UNAVAILABLE	Störungsanzeigelampe blinkt nicht.
mpJ1939_FMIL_CLASS_C_DTC	Störungsanzeigelampe wird in der "Class C DTC" betrieben.
mpJ1939_FMIL_FAST_FLASH	Störungsanzeigelampe blinkt mit >=2Hz (50% Einschaltdauer)
mpJ1939_FMIL_SLOW_FLASH	Störungsanzeigelampe blinkt mit 1Hz (50% Einschaltdauer)

6.6.1.2.83 MpJ1939FPLEnum

Dieser Aufzählungsdatentyp definiert das Verhalten der Motorschutzlampe.

Aufzählung	Beschreibung
mpJ1939_FPL_UNAVAILABLE	Motorschutzlampe blinkt nicht.
mpJ1939_FPL_RESERVED	Motorschutzlampe ist reserviert.
mpJ1939_FPL_FAST_FLASH	Motorschutzlampe blinkt mit >=2Hz (50% Einschaltdauer)
mpJ1939_FPL_SLOW_FLASH	Motorschutzlampe blinkt mit 1Hz (50% Einschaltdauer)

6.6.1.2.84 MpJ1939FRSLEnum

Dieser Aufzählungsdatentyp definiert das Verhalten der rotem Bremsleuchte.

Aufzählung	Beschreibung
mpJ1939_FRSL_UNAVAILABLE	Rote Bremsleuchte nicht vorhanden
mpJ1939_FRSL_RESERVED	Rote Bremsleuchte ist reserviert
mpJ1939_FRSL_FAST_FLASH	Rote Bremsleuchte blinkt mit >=2Hz (50% Einschaltdauer)
mpJ1939_FRSL_SLOW_FLASH	Rote Bremsleuchte blinkt mit 1Hz (50% Einschaltdauer)

6.6.1.2.85 MpJ1939HDPEEnum

Dieser Aufzählungsdatentyp definiert die Spülfreigabe des Kohlenwasserstoffdosierers (HDPE).

Aufzählung	Beschreibung
mpJ1939_HDPE_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_HDPE_URGENT	Spülung aktiviert (dringend).
mpJ1939_HDPE_LESS_URGENT	Spülung aktiviert (weniger dringend).
mpJ1939_HDPE_NOT_ENABLED	HDPE nicht eingeschaltet.

6.6.1.2.86 MpJ1939MEMPEFEnum

Dieser Aufzählungsdatentyp gibt an, ob die maximale Motorleistung momentan aktiviert wird oder nicht.

Aufzählung	Beschreibung
mpJ1939_MEMPEF_DONT_CARE	Motorleistung wartet.
mpJ1939_MEMPEF_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_MEMPEF_SUPPORTED	Motorleistung aktiviert.
mpJ1939_MEMPEF_DISABLED	Motorleistung nicht aktiviert.

6.6.1.2.87 MpJ1939MILSEnum

Dieser Aufzählungsdatentyp zeigt den Status der Störungsanzeigelampe.

Aufzählung	Beschreibung
mpJ1939_MILS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_MILS_SHORT_FOR_WWH_OBD	Störungsanzeigelampe nicht WWH-OBD konform.
mpJ1939_MILS_LAMP_ON	Störungsanzeigelampe ist eingeschaltet.
mpJ1939_MILS_LAMP_OFF	Störungsanzeigelampe ist ausgeschaltet.

6.6.1.2.88 MpJ1939NO1RSEnum

Dieser Aufzählungsdatentyp gibt den Status des NOx-Sensors an.

Aufzählung	Beschreibung
mpJ1939_NO1RS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_NO1RS_ERROR	Fehlerzustand.
mpJ1939_NO1RS_STABLE	Der gemessene Wert des Sensors ist stabil.
mpJ1939_NO1RS_NOT_STABLE	Der gemessene Wert des Sensors ist nicht stabil.

6.6.1.2.89 MpJ1939NOS1SDSEnum

Dieser Aufzählungsdatentyp gibt den Status des NOx Sensors an.

Aufzählung	Beschreibung
mpJ1939_NOS1SDS_NOT_SUPPOR	Sensor wird nicht unterstützt
mpJ1939_NOS1SDS_NOT_POSSIBLE	Überprüfung durch Sensor nicht möglich.
mpJ1939_NOS1SDS_ABORTED	Überprüfung des Sensors abgebrochen.
mpJ1939_NOS1SDS_COMPLETE	Überprüfung des Sensors abgeschlossen
mpJ1939_NOS1SDS_ACTIVE	Sensor aktiv.
mpJ1939_NOS1SDS_NOT_ACTIVE	Sensor nicht aktiv.

6.6.1.2.90 MpJ1939OCMPEnum

Dieser Aufzählungsdatentyp definiert die Priorität des Override-Steuerungsmodus, der in der Drehmomentmeldung empfangen wird.

Aufzählung	Beschreibung
mpJ1939_OCMP_LOW_PRIORITY	Niedrige Priorität.
mpJ1939_OCMP_MEDIUM_PRIORITY	Mittlere Priorität.
mpJ1939_OCMP_HIGH_PRIORITY	Hohe Priorität.
mpJ1939_OCMP_HIGHEST_PRIORITY	Sehr hohe Priorität.

6.6.1.2.91 MpJ1939OSDSEnum

Dieser Aufzählungsdatentyp gibt an, ob sich der Fahrersitz in der vorderen Fahrposition befindet.

Aufzählung	Beschreibung
mpJ1939_OSDS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_OSDS_ERROR	Fehlerzustand.
mpJ1939_OSDS_FACING_FORWARD	Fahrersitz befindet sich an der vorderen Position.
mpJ1939_OSDS_NOT_FACING_FORWARD	Fahrersitz befindet sich nicht an der vorderen Position.

6.6.1.2.92 MpJ1939PBCEnum

Dieser Aufzählungsdatentyp gibt an, ob die Parkbremse durch ein sekundäres Gerät gesteuert wird.

Aufzählung	Beschreibung
mpJ1939_PBC_DONT_CARE	Parkbremse wartet.
mpJ1939_PBC_SAE_RESERVED	Für SAE Zuweisung reserviert.
mpJ1939_PBC_REQUESTED	Parkbremse wird angefragt.
mpJ1939_PBC_NOT_REQUESTED	Parkbremse wird nicht angefragt.

6.6.1.2.93 MpJ1939PBRIREnum

Dieser Aufzählungsdatentyp gibt den Zustand der Parkbremsfreigabesperranforderung (PBRIR) an.

Aufzählung	Beschreibung
mpJ1939_PBRIR_UNAVAILABLE	PBRIR nicht verfügbar.
mpJ1939_PBRIR_SAE_RESERVED	Für SAE Zuweisung reserviert.
mpJ1939_PBRIR_REQUESTED	PBRIR wurde angefragt.
mpJ1939_PBRIR_NOT_REQUESTED	PBRIR wurde nicht angefragt.

6.6.1.2.94 MpJ1939PBSEnum

Dieser Aufzählungsdatentyp gibt den Zustand der Parkbremse an.

Aufzählung	Beschreibung
mpJ1939_PBS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_PBS_ERROR	Fehlerzustand.
mpJ1939_PBS_SET	Parkbremse ist aktiviert.
mpJ1939_PBS_NOT_SET	Parkbremse ist nicht aktiviert.

6.6.1.2.95 MpJ1939PGNStatusEnum

Dieser Aufzählungsdatentyp gibt den Status der Parametergruppennummer (PGN) an.

Aufzählung	Beschreibung
mpJ1939_PGN_IDLE	Aktuell ist keine PGN aktiv.
mpJ1939_PGN_NEW_DATA	Daten von PGN sind verfügbar.
mpJ1939_PGN_ERROR_CYCLE_TIME	PGN wurde in falscher Zykluszeit aufgerufen.
mpJ1939_PGN_ERROR_LIMITS	Das Limit wurde überschritten
mpJ1939_PGN_ERROR_TRANSMIT	Gesendete PGN fehlerhaft.
mpJ1939_PGN_ERROR_RECEIVE	Empfangene PGN fehlerhaft.
mpJ1939_PGN_NOT_ACTIVE	PGN ist nicht aktiv.

6.6.1.2.96 MpJ1939PLSEnum

Dieser Aufzählungsdatentyp gibt den Status der Motorschutzlampe an.

Aufzählung	Beschreibung
mpJ1939_PLS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_PLS_RESERVED	Reserviert.
mpJ1939_PLS_LAMP_ON	Motorschutzlampe ist an.
mpJ1939_PLS_LAMP_OFF	Motorschutzlampe ist aus.

6.6.1.2.97 MpJ1939PTOGSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Zapfwellenreglers an.

Aufzählung	Beschreibung
mpJ1939_PTOGS_NOT_AVAILABL	Status nicht verfügbar.
mpJ1939_PTOGS_SPEED_MEMORY_2	Zapfwellenreglers ist auf Geschwindigkeitspeicher 2 eingestellt.
mpJ1939_PTOGS_SPEED_MEMORY_1	Zapfwellenreglers ist auf Geschwindigkeitspeicher1 eingestellt.
mpJ1939_PTOGS_SET_SPEED_8	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 8 eingestellt.
mpJ1939_PTOGS_SET_SPEED_7	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 7 eingestellt.
mpJ1939_PTOGS_SET_SPEED_6	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 6 eingestellt.
mpJ1939_PTOGS_SET_SPEED_5	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 5 eingestellt.
mpJ1939_PTOGS_SET_SPEED_4	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 4 eingestellt.
mpJ1939_PTOGS_SET_SPEED_3	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 3 eingestellt.
mpJ1939_PTOGS_SET_SPEED_2	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 2 eingestellt.
mpJ1939_PTOGS_SET_SPEED_1	Zapfwellenregler ist auf die voreingestellte Geschwindigkeit 1 eingestellt.
mpJ1939_PTOGS_ACCEL_OVERRIDE	Zapfwellenreglers ist auf "Beschleunigungsüberschreibung".
mpJ1939_PTOGS_ACCELERATE	Zapfwellenreglers ist auf "Beschleunigung".
mpJ1939_PTOGS_RESUME	Zapfwellenreglers ist auf "Fortsetzen".
mpJ1939_PTOGS_DECELERATE	Zapfwellenreglers ist auf "Verlangsamen".
mpJ1939_PTOGS_SET	Zapfwellenreglers ist auf "Set".
mpJ1939_PTOGS_REMOTE_STANDBY	Zapfwellenreglers ist auf "Remote-Standby".
mpJ1939_PTOGS_STANDBY	Zapfwellenreglers ist auf "Standby".
mpJ1939_PTOGS_REMOTE_HOLD	Zapfwellenreglers ist auf "Remote-Halten".
mpJ1939_PTOGS_HOLD	Zapfwellenreglers ist auf "Halten".
mpJ1939_PTOGS_OFF	Zapfwellenreglers ist ausgeschaltet.

6.6.1.2.98 MpJ1939PTOSHSEEnum

Dieser Aufzählungsdatentyp gibt an, ob der Motor durch das PTO-Abschaltsystem abgeschaltet wurde oder nicht.

Aufzählung	Beschreibung
mpJ1939_PTOSHSE_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_PTOSHSE_ERROR	Fehlerzustand.
mpJ1939_PTOSHSE_YES	Motor wurde abgeschaltet.
mpJ1939_PTOSHSE_NO	Motor wurde nicht abgeschaltet.

6.6.1.2.99 MpJ1939RCZHEnum

Dieser Aufzählungsdatentyp gibt den Status der Kabinenheizung an.

Aufzählung	Beschreibung
mpJ1939_RCZH_DONT_CARE	Kabinenheizung wartet.
mpJ1939_RCZH_RESERVED	Reserviert.
mpJ1939_RCZH_HEAT_CAB_ZONE	Kabinenheizung heizt.
mpJ1939_RCZH_DO_NOT_HEAT	Kabinenheizung heizt nicht.

6.6.1.2.100 MpJ1939REZHEnum

Dieser Aufzählungsdatentyp bibt den Status der Motorzonenheizung an.

Aufzählung	Beschreibung
mpJ1939_REZH_DONT_CARE	Motorheizung wartet.
mpJ1939_REZH_RESERVED	Reserviert.
mpJ1939_REZH_HEAT_ENGINE_ZONE	Motorheizung heizt.
mpJ1939_REZH_DO_NOT_HEAT	Motorheizung heizt nicht.

6.6.1.2.101 MpJ1939RHPSEnum

Dieser Aufzählungsdatentyp gibt die Stellung des Hochdruckschalters im Kühlmittelkreislauf einer Klimaanlage an.

Aufzählung	Beschreibung
mpJ1939_RHPS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_RHPS_ERROR	Fehlerzustand.
mpJ1939_RHPS_PRESSURE_TOO_HIGH	Druck ist zu hoch.
mpJ1939_RHPS_PRESSURE_NORMAL	Druck ist normal.

6.6.1.2.102 MpJ1939RLPSEnum

Dieser Aufzählungsdatentyp gibt die Stellung des Niederdruckschalters im Kühlmittelkreislauf einer Klimaanlage an.

Aufzählung	Beschreibung
mpJ1939_RLPS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_RLPS_ERROR	Fehlerzustand.
mpJ1939_RLPS_PRESSURE_TOO_LOW	Druck zu niedrig.
mpJ1939_RLPS_PRESSURE_NORMAL	Druck ist normal.

6.6.1.2.103 MpJ1939RSLiSEnum

Dieser Aufzählungsdatentyp gibt den Status des Systems zur Begrenzung der maximalen Fahrzeuggeschwindigkeit an.

Aufzählung	Beschreibung
mpJ1939_RSLIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_RSLIS_ERROR	Fehlerzustand.
mpJ1939_RSLIS_NOT_ACTIVE	Begrenzung nicht aktiv.
mpJ1939_RSLIS_ACTIVE	Begrenzung ist aktiv.

6.6.1.2.104 MpJ1939RSLSEnum

Dieser Aufzählungsdatentyp zeigt den Status der roten Bremsleuchte an.

Aufzählung	Beschreibung
mpJ1939_RSLS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_RSLS_RESERVED	Rote Bremsleuchte ist reserviert.
mpJ1939_RSLS_LAMP_ON	Rote Bremsleuchte ist an.
mpJ1939_RSLS_LAMP_OFF	Rote Bremsleuchte ist aus.

6.6.1.2.105 MpJ1939S1ATEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Heizelements an.

Aufzählung	Beschreibung
mpJ1939_S1AT_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_S1AT_ERROR	Fehlerzustand.
mpJ1939_S1AT_IN_RANGE	Wert des Heizelements im erlaubten Bereich.
mpJ1939_S1AT_NOT_IN_RANGE	Wert des Heizelements ist nicht im erlaubten Bereich.

6.6.1.2.106 MpJ1939S1HCEnum

Dieser Aufzählungsdatentyp gibt des Status des Aufwärmvorgangs an.

Aufzählung	Beschreibung
mpJ1939_S1HC_HEATER_OFF	Heizung ist aus.
mpJ1939_S1HC_PREHEAT_1	Vorheizung Stufe 1.
mpJ1939_S1HC_PREHEAT_2	Vorheizung Stufe 2.
mpJ1939_S1HC_AUTOMATIC	Automatische Heizung.

6.6.1.2.107 MpJ1939S1PIREnum

Dieser Aufzählungsdatentyp gibt den Zustand des Nachbehandlungs-Einlassgassensors an.

Aufzählung	Beschreibung
mpJ1939_S1PIR_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_S1PIR_ERROR	Fehlerzustand.
mpJ1939_S1PIR_IN_RANGE	Wert im erlaubten Bereich.
mpJ1939_S1PIR_NOT_IN_RANGE	Wert nicht im erlaubten Bereich.

6.6.1.2.108 MpJ1939SBSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Schalter an, der verwendet wird, um festzustellen ob der Sicherheitsgurt angeschnallt ist.

Aufzählung	Beschreibung
mpJ1939_SBS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_SBS_ERROR	Fehlerzustand.
mpJ1939_SBS_OK	Sicherheitsgurt ist angeschnallt.
mpJ1939_SBS_NOT_BUCKLED	Sicherheitsgurt ist nicht angeschnallt.

6.6.1.2.109 MpJ1939SCRINO1RSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des NOx-Messwerts an.

Aufzählung	Beschreibung
mpJ1939_SCRINO1RS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_SCRINO1RS_ERROR	Fehlerzustand.
mpJ1939_SCRINO1RS_STABLE	Messwert stabil.
mpJ1939_SCRINO1RS_NOT_STABLE	Messwert nicht stabil.

6.6.1.2.110 MpJ1939SCRTMAEnum

Dieser Aufzählungsdatentyp gibt an, ob die Abgastemperaturen für die Regeneration des SCR-Nachbehandlungssystems oder für die Vorbereitung der Regeneration des SCR-Nachbehandlungssystems (SCRTMA) erhöht wurden.

Aufzählung	Beschreibung
mpJ1939_SCRTMA_DONT_CARE	SCRTMA wartet.
mpJ1939_SCRTMA_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_SCRTMA_ACTIVE	SCRTMA aktiv.
mpJ1939_SCRTMA_NOT_ACTIVE	SCRTMA nicht aktiv.

6.6.1.2.111 MpJ1939TSASEnum

Dieser Aufzählungsdatentyp gibt den Zustand des aktuellen Achsbereichs an.

Aufzählung	Beschreibung
mpJ1939_TSAS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_TSAS_ERROR	Fehlerzustand.
mpJ1939_TSAS_HIGH_SPEED_RANGE	Achsbereich befindet sich im hohen Geschwindigkeitsbereich.
mpJ1939_TSAS_LOW_SPEED_RANGE	Achsbereich befindet sich im niedrigen Geschwindigkeitsbereich.

6.6.1.2.112 MpJ1939TSC1CPEnum

Dieser Aufzählungsdatentyp gibt an welchen Steuermodus das sendende Gerät verwendet, um den TSC1-Befehl zu generieren.

Aufzählung	Beschreibung
mpJ1939_TSC1CP_P32_TEMP_PWR_CTRL	Modus "Temporäre Antriebssteuerung".
mpJ1939_TSC1CP_P5_ENG_PROTECTION	Modus "Motorschutz".
mpJ1939_TSC1CP_P4_ROAD_SPEED_GVN	Modus "Geschwindigkeitsbegrenzer".
mpJ1939_TSC1CP_P3_PTO_GOVERNOR	Modus "Zapfwellenregler".
mpJ1939_TSC1CP_P2_CRUISE_CONTROL	Modus "Tempomat".
mpJ1939_TSC1CP_P1_ACCEL_PEDAL	Modus "Gaspedal".

6.6.1.2.113 MpJ1939TSC1TREnum

Dieser Aufzählungsdatentyp gibt die Übertragungsrate an mit der das sendende Gerät die TSC1-Nachricht überträgt.

Aufzählung	Beschreibung
mpJ1939_TSC1TR_100_MS	100ms Übertragungsrate
mpJ1939_TSC1TR_250_MS	250ms Übertragungsrate
mpJ1939_TSC1TR_500_MS	500ms Übertragungsrate
mpJ1939_TSC1TR_750_MS	750ms Übertragungsrate
mpJ1939_TSC1TR_1000_MS	1000ms Übertragungsrate
mpJ1939_TSC1TR_STANDARD_10_MS	10ms Übertragungsrate
mpJ1939_TSC1TR_20_MS	20ms Übertragungsrate
mpJ1939_TSC1TR_50_MS	50ms Übertragungsrate

6.6.1.2.114 MpJ1939VARLSEnum

Dieser Aufzählungsdatentyp gibt den Status des Systems, mit dem die maximale Fahrzeugvorwärtsbeschleunigung begrenzt wird.

Aufzählung	Beschreibung
mpJ1939_VARLS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_VARLS_RESERVED	Reserviert für SAE-Zuordnung.
mpJ1939_VARLS_LIMIT_ACTIVE	Begrenzung aktiv.
mpJ1939_VARLS_LIMIT_NOT_ACTIVE	Begrenzung nicht aktiv.

6.6.1.2.115 MpJ1939VLSGDSEnum

Dieser Aufzählungsdatentyp gibt an ob der Fahrzeugbegrenzungsgeschwindigkeitsregler (VLSG) verringert ist.

Aufzählung	Beschreibung
mpJ1939_VLSGDS_ERROR	Status nicht verfügbar.
mpJ1939_VLSGDS_ON_STATE	VLSG ist verringert.
mpJ1939_VLSGDS_OFF_STATE	VLSG ist nicht verringert.
mpJ1939_VLSGDS_NOT_AVAILABLE	VLSG ist nicht verfügbar.

6.6.1.2.116 MpJ1939VLSGESEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Fahrzeugbegrenzungsgeschwindigkeitsregler (VLSGS) an.

Aufzählung	Beschreibung
mpJ1939_VLSGES_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_VLSGES_ERROR	Fehlerzustand.
mpJ1939_VLSGES_SWITCHED_ENABLED	VLSGS ist aktiviert.
mpJ1939_VLSGES_SWITCH_DISABLED	VLSGS ist deaktiviert.

6.6.1.2.117 MpJ1939VLSGISEnum

Dieser Aufzählungsdatentyp gibt an ob der Fahrzeugbegrenzungsgeschwindigkeitsregler (VLSG) erhöht ist.

Aufzählung	Beschreibung
mpJ1939_VLSGIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_VLSGIS_ERROR	Fehlerzustand.
mpJ1939_VLSGIS_ON_STATE	VLSG ist erhöht.
mpJ1939_VLSGIS_OFF_STATE	VLSG ist nicht erhöht.

6.6.1.2.118 MpJ1939WRPO21RSEnum

Dieser Aufzählungsdatentyp gibt den Zustand des Sauerstoffmesswerts an.

Aufzählung	Beschreibung
mpJ1939_WRPO21RS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_WRPO21RS_ERROR	Fehlerzustand.
mpJ1939_WRPO21RS_STABLE	Sauerstoffmesswert ist im stabilen Zustand.
mpJ1939_WRPO21RS_NOT_STABLE	Sauerstoffmesswert ist im nicht stabilen Zustand.

6.6.1.2.119 MpJ1939ACHPFSEnum

Dieser Aufzählungsdatentyp zeig an, ob der Druck im Kühlmittelkreislauf einer Klimaanlage zu hoch ist und der Ventilator eingeschaltet werden muss.

Aufzählung	Beschreibung
mpJ1939_ACHPFS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_ACHPFS_ERROR	Fehlerzustand.
mpJ1939_ACHPFS_PRESSURE_HIGH	Druck ist zu hoch.
mpJ1939_ACHPFS_PRESSURE_NORMAL	Druck ist normal.

6.6.1.2.120 MpJ1939DEFTLLIEnum

Dieser Aufzählungsdatentyp gibt den Status des Bedieneraufforderungssystems für Anomalien mit dem SCR-System an.

Aufzählung	Beschreibung
mpJ1939_DEFTLLI_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_DEFTLLI_RESERVED6	Reserviert für SAE Zuweisung.
mpJ1939_DEFTLLI_RESERVED5	Reserviert für SAE Zuweisung.
mpJ1939_DEFTLLI_ON_FAST_BLINK	Bedieneraufforderungssystems blinkt.
mpJ1939_DEFTLLI_RESERVED3	Reserviert für SAE Zuweisung.
mpJ1939_DEFTLLI_RESERVED2	Reserviert für SAE Zuweisung.
mpJ1939_DEFTLLI_ON_SOLID	Bedieneraufforderungssystems leuchtet.
mpJ1939_DEFTLLI_OFF	Bedieneraufforderungssystems ist aus.

6.6.1.2.121 MpJ1939MpJ1939SCROISEnum

Dieser Aufzählungsdatentyp gibt den Status der Leuchtkraft des Fahrerwarnindikators für Dieselabgase an.

Aufzählung	Beschreibung
mpJ1939_SCROIS_NOT_AVAILABLE	Status nicht verfügbar.
mpJ1939_SCROIS_TEMP_OVERRIDE	Leuchte leuchtet.
mpJ1939_SCROIS_INDUCEMENT_LEVEL5	Reserviert für SAE Zuweisung.
mpJ1939_SCROIS_INDUCEMENT_LEVEL4	Reserviert für SAE Zuweisung.
mpJ1939_SCROIS_INDUCEMENT_LEVEL3	Leuchte blinkt.
mpJ1939_SCROIS_INDUCEMENT_LEVEL2	Reserviert für SAE Zuweisung.
mpJ1939_SCROIS_INDUCEMENT_LEVEL1	Reserviert für SAE Zuweisung.
mpJ1939_SCROIS_DRIVER_WARNING	Leuchte im Fehlerzustand.

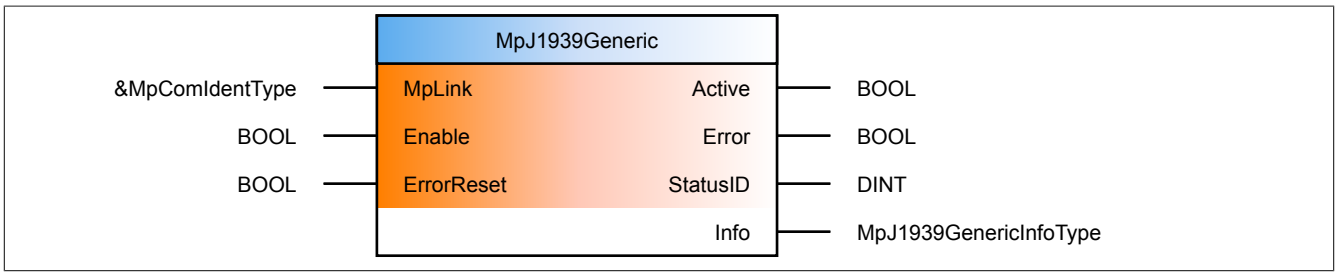
6.6.2 Funktionsbausteine

Unterstützt ab	Name
5.2.0	MpJ1939Generic
5.2.0	MpJ1939Receive
5.2.0	MpJ1939SpecificRequest
5.2.0	MpJ1939Transmit

6.6.2.1 MpJ1939Generic

Mithilfe dieses Funktionsbausteins können PGNs empfangen und gesendet werden.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpJ1939Generic Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpJ1939GenericInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

6.6.2.1.1 Beschreibung

Dieser Funktionsbaustein verwendet den MpLink einer MpJ1939Generic Konfiguration.

Mithilfe von MpJ1939Generic werden die in der MpJ1939Generic Konfiguration festgelegten Parametergruppennummern (PGN) an der angegebenen CAN-Schnittstelle gesendet und empfangen. Die CAN-Schnittstelle wird in der MpJ1939Generic Konfiguration unter "CAN Interface" festgelegt. Wie die Daten gesendet und empfangen werden, wird im Abschnitt Informationen senden und empfangen erklärt.

Manche PGNs müssen in bestimmten Zykluszeiten gesendet werden. Für mehr Information siehe Timing.

Die Info-Stuktur zeigt Diagnoseinformationen des Funktionsbausteins an.

Haben Signale in PGNs Skalierungen oder einen Offset so werden diese beim Senden und Empfangen der PGN automatisch angepasst. Das Signal EngRequestedSpeed_SpeedLimit in der PGN TSC1 hat zum Beispiel einen Faktor von 0,125. Dieser Faktor wird beim Senden und Empfangen der PGN automatisch berücksichtigt.

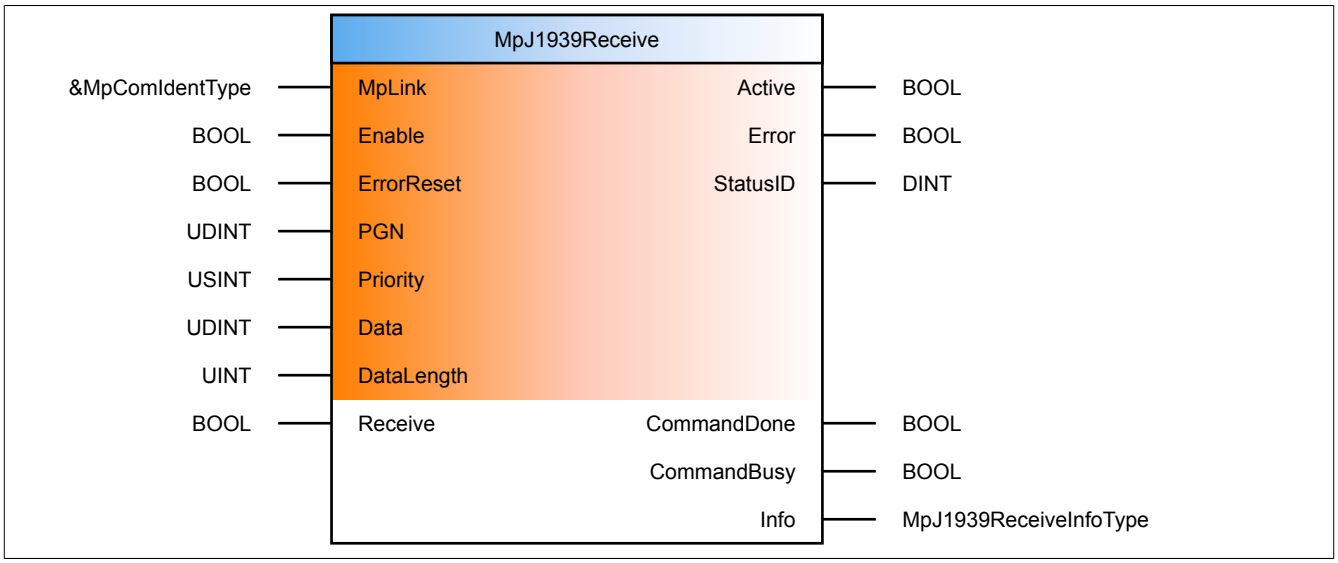
Eine PGN kann auch minimal und maximal Werte haben. Diese werden auch automatisch berücksichtigt. Wird eine PGN gesendet in der ein Wert den maximalen Wert der PGN überschreiten würde, so wird der Fehler Wert außerhalb der Grenzen ausgelöst und der Wert kann nicht geschickt werden.

Ob und welche Skalierung oder Offset eine PGN hat, kann aus einer PGN-Datenbank entnommen werden.

6.6.2.2 MpJ1939Receive

Mithilfe dieses Funktionsbausteins kann eine angegebene PGN empfangen werden.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpJ1939Receive Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	PGN	`UDINT`	Angabe des numerischen Wertes der PGN.
IN	Priority	`USINT`	Priorität der PGN.
IN	Data	`UDINT`	Angabe der Adresse des Datenbuffers für die PGN. Für mehr Information siehe Speichermanagement.
IN	DataLength	`UINT`	Definiert die Größe des Datenbuffers "Data". Wird in Byte angegeben.
IN	Receive	`BOOL`	Kommando um PGN zu empfangen.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	CommandBusy	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	Info	`MpJ1939ReceiveInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

6.6.2.2.1 Beschreibung

Der Funktionsbaustein verwendet den MpLink einer MpJ1939Receive Konfiguration.

Mithilfe dieses Funktionsbausteins kann die Information einer lesbaren PGN empfangen werden. Für mehr Information siehe Informationen senden und empfangen.

Durch den Funktionsbaustein wird die am Eingangsparameter "PGN" angegebene PGN abgefragt. Dabei muss der numerische Wert der PGN angegeben werden. An "Data" muss der Buffer für die PGN bereitgestellt werden. Wie dieser Buffer erstellt werden muss, ist im Abschnitt Speichermanagement erklärt.

Am Eingang "Priority" muss die Priorität der PGN angegeben werden. Diese kann für jede PGN unterschiedlich sein. Die Priorität einer PGN ist in einer J1939 Datenbank zu finden. Für die PGN DTBI1 ist die Priorität zum Beispiel 5.

Die angefragte Information wird durch eine positive Flanke am Eingangsparameter "Receive=`TRUE`" am Parameter "Data" angezeigt. Ob das Kommando erfolgreich war, ist am Ausgang "CommandDone = `TRUE`" ersichtlich. Das bedeutet, durch jede positive Flanke an "Receive" wird erneut die Information an der angegebenen PGN abgefragt.

Es gilt zu beachten, dass PGNs in einer bestimmten Zykluszeit abgefragt werden müssen. Für mehr Information siehe Timing.

Am Eingangsparameter "DataLength" wird die Größe des Datenbuffers "Data" angegeben.

Wurde die Information einer PGN erhalten und es soll eine neue PGN angefragt werden, so muss der Funktionsbaustein über "Enable = `FALSE`" deaktiviert und über "Enable = `TRUE`" neu aktiviert werden. Nach einer neuen Aktivierung des Funktionsbausteins kann eine neue PGN angegeben werden.

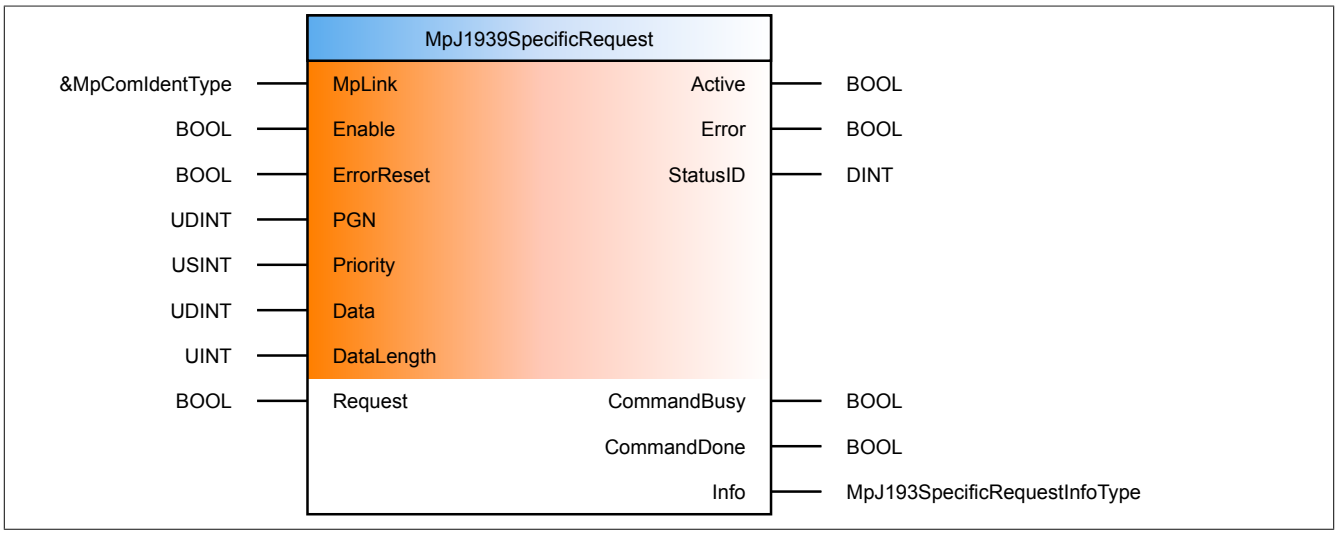
Es gilt zu beachten, dass eine PGN Grenzwerte, Skalierungen sowie einen Offset haben kann. Diese müssen bei der Weiterverarbeitung der PGN berücksichtigt werden. Welche Grenzwerte, Offset oder Skalierung eine PGN hat kann in einer J1939 Datenbank entnommen werden, welche hier heruntergeladen werden kann.

Wurde das Kommando "Receive" erfolgreich ausgeführt, so ist die Anzahl der erhaltenen Bytes am Parameter "ReceivedBytes" der Info-Struktur ersichtlich.

6.6.2.3 MpJ1939SpecificRequest

Mithilfe dieses Funktionsbausteins kann eine PGN angefragt werden, welche nicht zyklisch gelesen werden kann.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpJ1939SpecificRequest Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	PGN	`UDINT`	Angabe des numerischen Wertes der PGN.
IN	Priority	`USINT`	Priorität der PGN.
IN	Data	`UDINT`	Angabe der Adresse des Datenbuffers für die PGN. Für mehr Information siehe Speichermanagement.
IN	DataLength	`UINT`	Definiert die Größe des Datenbuffers "Data". Wird in Byte angegeben.
IN	Request	`BOOL`	Kommando um Anfrage zu starten.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	CommandBusy	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpJ1939SpecificRequestInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

6.6.2.3.1 Beschreibung

Der Funktionsbaustein verwendet den MpLink einer MpJ1939SpecificRequest Konfiguration.

Mithilfe dieses Funktionsbausteins kann eine PGN abgefragt werden, welche nicht zyklisch gelesen werden kann. Für mehr Information siehe Informationen senden und empfangen.

Durch den Funktionsbaustein wird die am Eingangsparameter "PGN" angegebene PGN abgefragt. Dabei muss der numerische Wert der PGN angegeben werden. Der numerische Wert einer PGN kann in einer J1939 Datenbank entnommen werden. Diese kann zum Beispiel hier heruntergeladen werden. An "Data" muss der Buffer für die PGN bereitgestellt werden. Wie dieser Buffer erstellt werden muss, ist im Abschnitt Speichermanagement erklärt.

Am Eingang "Priority" muss die Priorität der PGN angegeben werden. Diese kann für jede PGN unterschiedlich sein. Die Priorität einer PGN ist in einer J1939 Datenbank zu finden. Für die PGN DTBI1 ist die Priorität zum Beispiel 5.

Die angefragte Information wird durch das Kommando "Request = `TRUE`" am Parameter "Data" angezeigt. Ob das Kommando erfolgreich war, ist am Ausgang "CommandDone = `TRUE`" ersichtlich.

Am Eingangsparameter "DataLength" wird die Größe des Datenbuffers "Data" angegeben.

Wurde die Information einer PGN erhalten und es soll eine neue PGN angefragt werden, so muss der Funktionsbaustein über "Enable = `FALSE`" deaktiviert und über "Enable = `TRUE`" neu aktiviert werden. Nach einer neuen Aktivierung des Funktionsbausteins kann eine neue PGN angegeben werden.

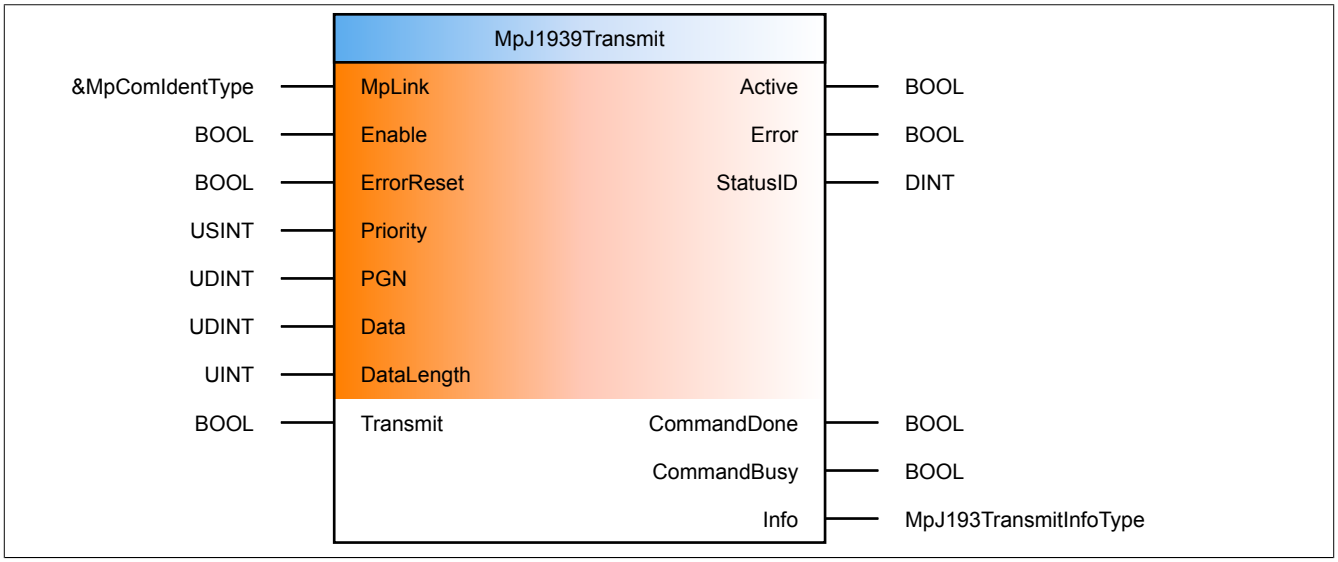
Es gilt zu beachten, dass eine PGN Grenzwerte, Skalierungen sowie einen Offset haben kann. Diese müssen bei der Weiterverarbeitung der PGN berücksichtigt werden. Welche Grenzwerte, Offset oder Skalierung eine PGN hat kann in einer J1939 Datenbank entnommen werden, welche hier heruntergeladen werden kann.

Wurde das Kommando "Request" erfolgreich ausgeführt, so ist die Anzahl der erhaltenen Bytes am Parameter "ReceivedBytes" der Info-Struktur ersichtlich.

6.6.2.4 MpJ1939Transmit

Mithilfe dieses Funktionsbausteins kann eine PGN gesendet werden.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpJ1939Transmit Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	PGN	`UDINT`	Angabe des numerischen Wertes der PGN.
IN	Priority	`USINT`	Priorität der PGN.
IN	Data	`UDINT`	Angabe der Adresse des Datenbuffers für die PGN. Für mehr Information siehe Speichermanagement.
IN	DataLength	`UINT`	Definiert die Größe des Datenbuffers "Data". Wird in Byte angegeben.
IN	Transmit	`BOOL`	Kommando um PGN zu senden.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	CommandBusy	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpJ1939TransmitInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

6.6.2.4.1 Beschreibung

Dieser Funktionsbaustein verwendet den MpLink einer MpJ1939Transmit Konfiguration.

Mithilfe dieses Funktionsbausteins kann eine lesbare PGN gesendet werden. Für mehr Information siehe Informationen senden und empfangen.

Durch den Funktionsbaustein wird die am Eingangsparameter "PGN" angegebene PGN abgefragt. Dabei muss der numerische Wert der PGN angegeben werden. Der numerische Wert einer PGN kann in einer J1939 Datenbank entnommen werden. Diese kann zum Beispiel hier heruntergeladen werden. An "Data" muss der Buffer für die PGN bereitgestellt werden. Wie dieser Buffer erstellt werden muss, ist im Abschnitt Speichermanagement erklärt.

Am Eingang "Priority" muss die Priorität der PGN angegeben werden. Diese kann für jede PGN unterschiedlich sein. Die Priorität einer PGN ist in einer J1939 Datenbank zu finden. Für die PGN DTBI1 ist die Priorität zum Beispiel 5.

Die Information welche am Parameter "Data" angegeben wurde, wird durch eine positive Flanke am Eingangsparameter "Transmit" gesendet. Um die gewünschte PGN erneut zu senden, ist eine weitere positive Flanke an "Transmit" erforderlich. Das erfolgreiche Senden wird durch "CommandDone = `TRUE`" bestätigt.

Es gilt zu beachten, dass PGNs in einer bestimmten Zykluszeit gesendet werden müssen. Für mehr Information siehe Timing.

Am Eingangsparameter "DataLength" wird die Größe des Datenbuffers "Data" angegeben.

Wurde die PGN gesendet und es soll eine neue PGN gesendet werden, so muss der Funktionsbaustein über "Enable = `FALSE`" deaktiviert und über "Enable = `TRUE`" neu aktiviert werden. Nach einer neuen Aktivierung des Funktionsbausteins kann eine neue PGN gesendet werden.

Es gilt zu beachten, dass eine PGN Grenzwerte, Skalierungen sowie einen Offset haben kann. Diese müssen bei der Weiterverarbeitung der PGN berücksichtigt werden. Welche Grenzwerte, Offset oder Skalierung eine PGN hat kann in einer J1939 Datenbank entnommen werden, welche hier heruntergeladen werden kann.

6.6.3 Statusnummern

6.6.3.1 -1064171516: Ungültiger Datenbuffer

Beschreibung:

Der Eingangsparameter "Data" ist nicht verbunden bzw. ein NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Parameter an "Data" verbinden

Konstante

`mpJ1939_ERR_INVALID_DATA_BUFFER`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.2 -1064171517: J1939 Kommunikation fehlgeschlagen

Beschreibung:

Ein Fehler während der Kommunikation oder der zyklischen Abarbeitung der PGNs ist aufgetreten. Für weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfung der CAN-Schnittstelle.
- Überprüfung des bereitgestellten Datenpuffers "Data". Für mehr Information siehe Speichermanagement.
- Für mehr Information siehe Logger.

Konstante

`mpJ1939_ERR_COMMUNICATION`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.3 -1064171518: Simulation nicht erlaubt

Beschreibung:

Die Komponente kann nicht in einer simulierten Umgebung verwendet werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Verwenden von echter Hardware.

Konstante

`mpJ1939_ERR_ARSIM_NOT_ALLOWED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.4 -1064171519: Fehler beim Öffnen der CAN-Schnittstelle

Beschreibung:

Beim Versuch die CAN-Schnittstelle zu Öffnen, ist ein Fehler aufgetreten. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen des CAN-Pfads der in der MpJ1939Generic Konfiguration angegeben wurde.

Konstante

`mpJ1939_ERR_INVALID_CAN_IF`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.5 -1064171520: Wert außerhalb der Grenzen

Beschreibung:

Der Wert eines PGN-Signals ist außerhalb der definierten Grenzen.

Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Maximale und minimale Werte eines PGN-Signals überprüfen. Für mehr Information siehe Speichermanagement.

Konstante

`mpJ1939_ERR_LIMITS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic

6.6.3.6 -1064239091: Ungültige Konfiguration

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Konfiguration ist beschädigt
- Die Funktion ist in der Konfiguration nicht aktiviert

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.7 -1064239094: Fehler beim Laden der Konfiguration

Beschreibung:

Fehler beim Laden der Konfigurationsdatei (in die PV). Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Konfigurationsdatei beschädigt
- Unterschied zwischen dem Inhalt der Konfigurationsdatei und der PV (z.B. unterschiedliche Datentypen, Daten außerhalb der gültigen Limits, ...)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.8 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.9 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.10 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.11 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.12 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.13 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpJ1939Generic
- MpJ1939Receive
- MpJ1939SpecificRequest
- MpJ1939Transmit

6.6.3.14 -1064290619: Ungültige PGN Konfiguration

Beschreibung:

Die angegebene Konfiguration für {2:PGN} ist ungültig. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen der PGN.
- PGN existiert nicht.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.15 -1064290621: CAN Schnittstelle nicht definiert

Beschreibung:

Es wurde kein Pfad zur CAN Schnittstelle definiert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Pfad zur CAN Schnittstelle in den verwendeten Konfigurationen definieren.

6.6.3.16 -1064290622: PGN-Anfrage fehlgeschlagen

Beschreibung:

PGN {2:PGN} wird nicht vom Gerät zur Verfügung gestellt. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen ob gewünschte PGN in der richtigen Zykluszeit abgefragt wurde. Für mehr Information siehe Timing.
- Überprüfen der PGN Nummer.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.17 -1064290623: Empfangen der Daten abgebrochen

Beschreibung:

Das Empfangen der Daten für {2:PGN} wurde abgebrochen. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Der Empfang der Daten wurde vom Sender abgebrochen.
- PGN nochmal anfordern.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.18 -1064290624: Ungültige Zieladresse

Beschreibung:

Die für {2:PGN} angegebene Zieladresse ist ungültig. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppennummer.

6.6.3.19 -1064290625: Zeitüberschreitung beim Empfangen

Beschreibung:

Beim Empfangen der Daten für {2:PGN} wurde eine Zeitüberschreitung ausgelöst. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Gewünschte PGN nochmal empfangen.
- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.20 -1064290626: Zeitüberschreitung beim Übertragen

Beschreibung:

Beim Übertragen der Daten für {2:PGN} wurde eine Zeitüberschreitung ausgelöst. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Gewünschte PGN nochmal senden.
- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe

6.6.3.21 -1064290627: Empfangen fehlgeschlagen

Beschreibung:

Beim Empfangen der Daten für {2:PGN} ist ein Fehler aufgetreten. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.22 -1064290630: Ungültiger Zeiger für Dateninfo

Beschreibung:

Ungültiger Zeiger für Dateninformation. PGN kann nicht geschickt oder empfangen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

6.6.3.23 -1064290632: Übertragung fehlgeschlagen

Beschreibung:

Die Datenübertragung für {2:PGN} ist fehlgeschlagen. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.24 -1064290634: Ungültiger Datenzeiger

Beschreibung:

Der angehängte Datenzeiger ist ungültig für {2:PGN}. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.25 -1064290635: Ungültige PGN

Beschreibung:

Die angegebene PGN {2:PGN} liegt außerhalb des gültigen Bereichs. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen der angegebenen PGN-Nummer.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.26 -1064290636: Ungültige Datenlänge

Beschreibung:

Die angegebene Datenlänge für {2:PGN} liegt außerhalb des erlaubten Bereichs. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.27 -1064290637: Ungültige Priorität

Beschreibung:

Die angegebene Priorität liegt außerhalb des gültigen Bereichs für {2:PGN}. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die PGN-Priorität muss <= 7 sein.
- Die Priorität einer PGN kann aus dem J1939 Standard entnommen werden.

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

6.6.3.28 -1064290639: Kein CAN Sender angegeben

Beschreibung:

Es wurde kein CAN-Sender angegeben. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

6.6.3.29 -1064290620: Ungültiges PV-Mapping

Beschreibung:

Das angegebene PV-Mapping für {2:PGN} ist ungültig. Weitere Details siehe Logger.

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen der PV in der MpJ1939Generic Konfiguration
- Die PV welche für die gewünschte PGN verwendet werden soll enthält falschen Datentypen

Zusätzliche Informationen:

- {2:PGN}: Nummer der betroffenen Parametergruppe.

Konstante

`mpJ1939_ERR_INVALID_PV_MAPPING`

7 mapp File: Dateiverwaltungssystem

mapp File stellt ein Dateiverwaltungssystem zur Verfügung, sowie eine Verbindung zur Visualisierung um die Dateien anzuzeigen.

7.1 Konzept

Allgemein

- Als Applikationist möchte ich wenig Aufwand in den Dateiexplorer als Basisfunktion einer Maschine investieren

Dateibrowser visualisieren

- Als Maschinenbediener möchte ich beliebig durch meine Dateistruktur browsen können, sowie Ordner erzeugen, löschen, ...
- Als Maschinenbediener möchte ich Dateien von einem Speichermedium auf ein anderes kopieren

7.1.1 Visualisierungstyp

mapp File kann auf zwei verschiedenen Arten verwendet werden. Mit einer mapp View Visualisierung oder mit einer VC4 Visualisierung.

mapp View

Wird als Visualisierungstyp mapp View verwendet, so kann die MpFileManager Konfiguration und/oder die MpFileManagerUI Konfiguration verwendet werden. Das bedeutet, es können zwei verschiedene Arten von Dateibrowsersystemen verwendet werden.

Dabei gilt es folgendes zu beachten:

MpFileManager Konfiguration

Diese Konfiguration ist die Basis für alle Widgets, welche Ordner und Dateien bearbeiten. Beispielsweise wird sie benötigt, um das MotionPad-Widget (Editieren von G-Code Programmen) oder das TextPad-Widget in Betrieb zu nehmen. Wie beispielsweise das MotionPad-Widget in Betrieb genommen wird, ist hier erklärt.

Es gilt zu beachten, dass die MpFileManager Konfiguration nicht in Kombination mit den Funktionsbausteinen MpFileManagerUI und MpFileManagerConfig verwendet werden kann! Die Konfiguration stellt direkt eine Verbindung mit dem gewünschten Widget her.

Wie ein Dateiverwaltungssystem in mapp View erstellt werden kann, ist im Getting Started Texteditor für CNC Programme in mapp View erstellen zu finden.

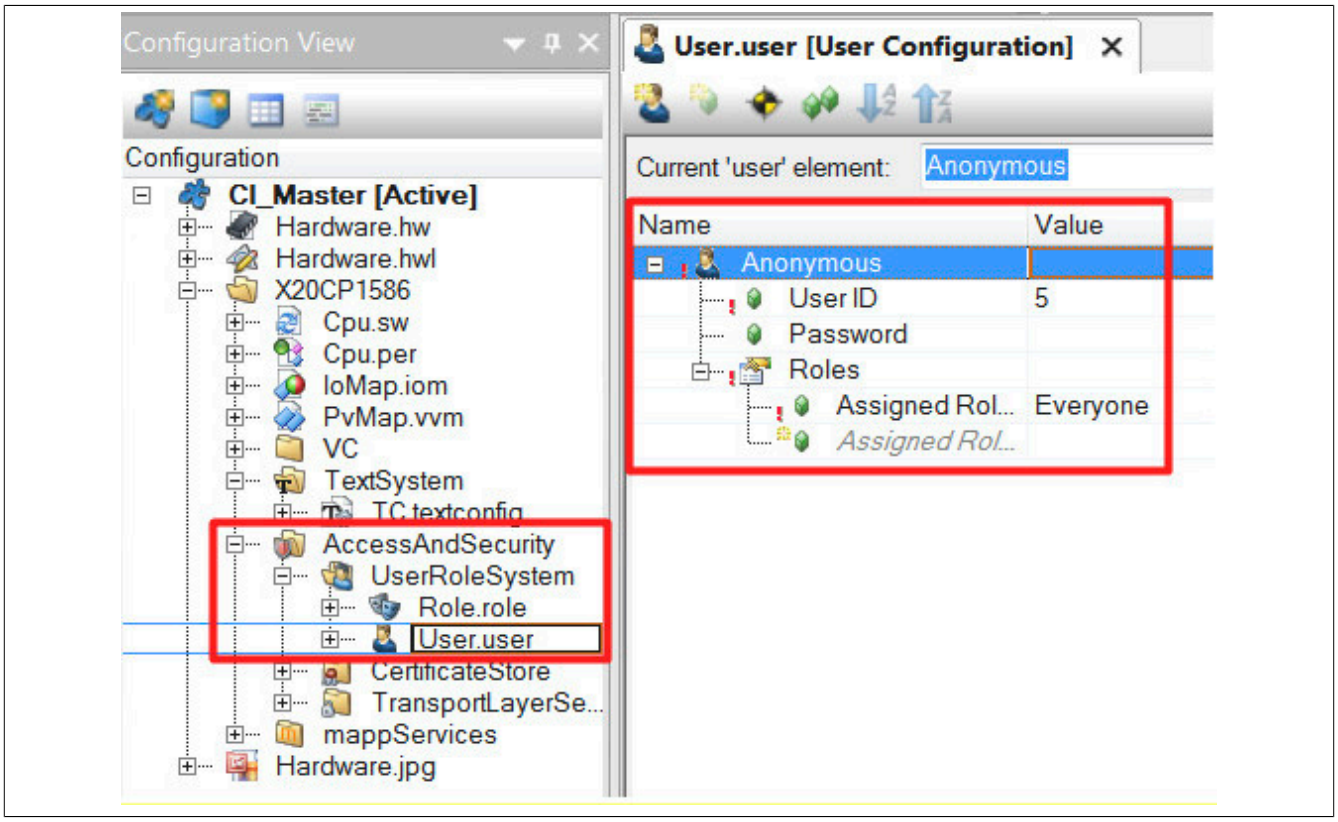
MpFileManagerUI Konfiguration

Bei der Verwendung der MpFileManagerUI Konfiguration muss, damit ein Dateiverwaltungssystem erstellt werden kann, der Funktionsbaustein MpFileManagerUI verwendet werden. Mithilfe der UIConnect-Struktur des Funktionsbausteins, kann eine Verbindung zu verschiednene Widgets aufgebaut werden. Es können beispielsweise die Dateinamen mit einem Table-Widget angezeigt werden und vieles mehr.

> **Beide Dateiverwaltungssysteme (MpFileManager und MpFileManagerUI Konfiguration) können innerhalb eines Projekt verwendet werden.**

mapp View Visualisierung

Es gilt zu beachten, dass bei der Verwendung von mapp View der Benutzer "Anonymous" mit der Rolle "Everyone" benötigt wird. Diese sind standardmäßig im Benutzer-Rollen-System von Automation Studio enthalten und sollten nicht verändert werden.



VC4

Wird als Visualisierungstyp VC4 ausgewählt, so kann nur die MpFileManagerUI Konfiguration in Kombination mit dem MpFileManagerUI Funktionsbaustein verwendet werden um ein Dateibrowser-System zu implementieren.

Wie ein Dateiverwaltungssystem in VC4 erstellt werden kann, ist im Getting Started Datei Explorer mit VC4 mapp Vorlage zu finden.

Die MpFileManager Konfiguration kann nicht in einer VC4-Visualisierung verwendet werden!

7.1.2 Auditing des Dateibrowsers

Beim Verwenden der MpFileManager Konfiguration können Ereignisse mithilfe von mapp Audit augezeichnet werden. Für mehr Information zur Konfiguration siehe hier.

> **Zugriffe außerhalb der mapp View Visualsierung (e.g. FileIO) werden momentan noch nicht überwacht und generieren daher auch keine Ereignisse.**

Das heißt alle Ordner-, und Datei-Änderungen welche in der mapp View Visualisierung (z.B. MotionPad-Widget) durchgeführt werden, werden aufgezeichnet.

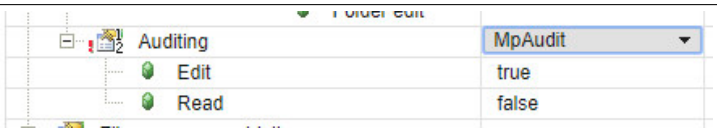
Für mehr Information wie ein Eintrag erstellt werden kann, siehe Abschnitt Definieren des Ereignistexts.

Folgende Ereignisse können auftreten:

Dateibrowser Ereignis

Durch diesen Ereignistyp kann aufgzeichnet werden, ob auf einen Ordner/Datei zugegriffen und/oder gelesen wurde.

Um die Ereignisse aufzuzeichnen, muss in der MpFileManager Konfiguration die Option "Auditing" freigeschaltet sein:



Ereignis ID 113	Konfiguration MpFileManager Konfiguration
Ereignisse	
idx	Liefert den Index des aktuell generierten Ereignisses. Der Wert wird bei jedem neuen Ereignis inkrementiert. Das erste generierte Ereignis in der Ereignisliste hat somit den Wert 1, dass fünfzigste Ereignis hat den Index 50, das hundertste den Index 100 und so weiter.
path	Datei-, oder Ordner-Pfad
file	Gibt an ob es sich um einen Ordner oder um eine Datei handelt. Bei `TRUE` handelt es sich um eine Datei. Bei `FALSE` um einen Ordner.
act	Aktion, welche in einer mapp Komponente ausgeführt wurde. Die ID für die Aktion wird zurükgegeben (z.B. 1 für "Ordner (oder Dokument) wure erstellt"). Für mehr Infomation siehe folgende Tabelle:

Ereignis	Identifikationsnummer (wird über %act zurückgegeben)
Ordner/Datei wurde erstellt	1
Ordner/Datei wurde überschrieben	2
Ordner/Datei wurde hinzugefügt	3
Ordner/Datei wurde gelöscht	4
Ordner/Datei wurde umbenannt	5
Ordner/Datei wurde gelesen	6
Ordner/Datei wurde kopiert	7
Ordner/Datei wurde ausgeschnitten	8
Ordner/Datei wurde zuerst kopiert und wieder eingefügt. Ein neuer Ordner/Datei wurde erstellt	9
Ordner/Datei wurde zuerst kopiert und wieder eingefügt. Der vorhandene Ordner/Datei wurde überschrieben	10
Ordner/Datei wurde gesperrt	11
Ordner/Datei wurde entsperrt	12

7.1.3 Rahmenbedingungen

In diesem Abschnitt werden die Rahmenbedingungen für mapp File aufgelistet.

Verwendung des Dateisystems

mapp File kann auf zwei verschiedenen Arten verwendet werden. Mit einer VC4-Visualisierung oder mit einer mapp View Visualisierung. Für mehr Information siehe hier.

7.1.4 Konfiguration während der Laufzeit

Die Konfigurations Parameter können zur Laufzeit über MpFileManagerConfig verändert werden. Dabei ist für die unterschiedlichen Parameter jedoch zu beachten in welchem Zustand sich die Komponente befinden muss um eine Änderung zuzulassen. Folgende Zustände sind möglich:

- : Dieses Symbol bedeutet, dass dieser Parameter in jedem Zustand geändert werden kann

Jeder Parameter der Konfigurations Struktur (MpFileManagerConfigType) ist mit einem dieser Symbole ausgestattet.

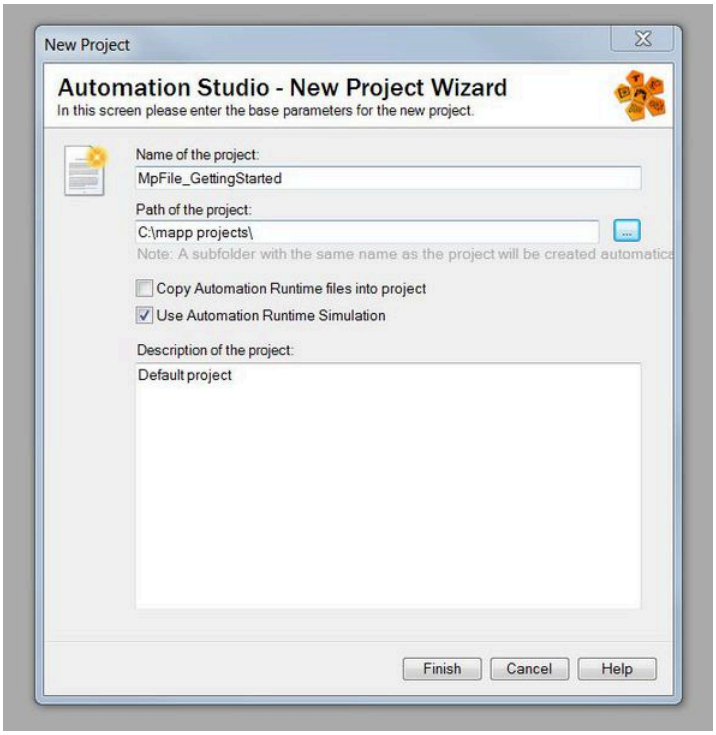
7.2 Getting Started

7.2.1 Datei Explorer mit VC4 mapp Vorlage

In diesem Abschnitt wird Schritt für Schritt beschrieben wie man mithilfe von MpFileManagerUI einen Datei Explorer erstellt und ihn mit der VC4 mapp Vorlage verwendet.

7.2.1.1 Projekt erstellen

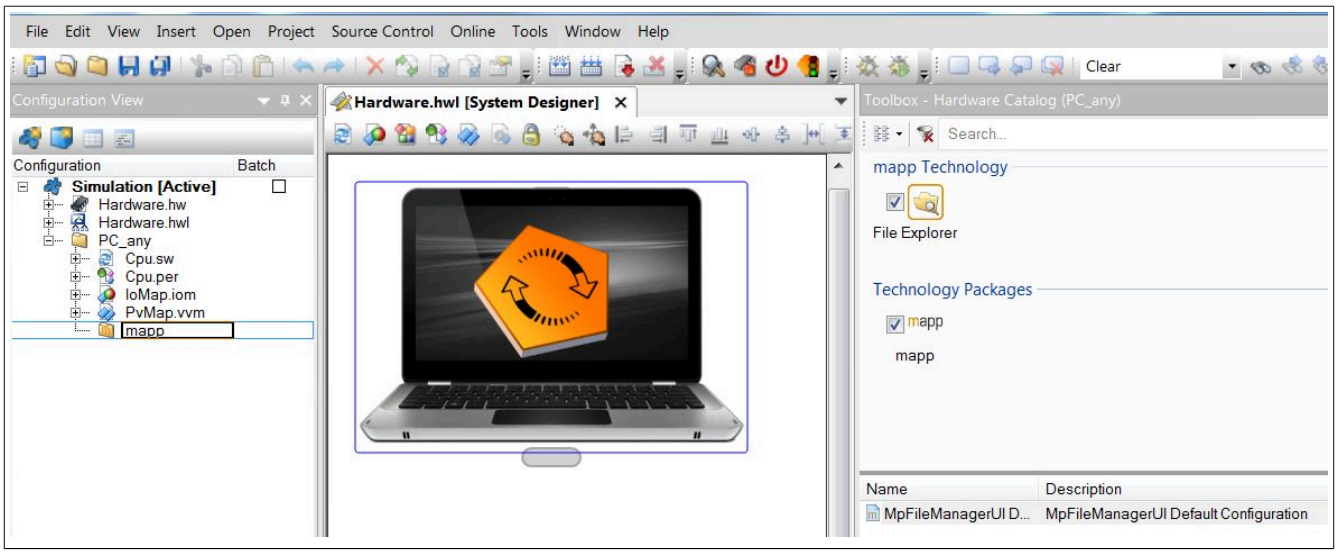
Zuerst muss ein neues Projekt über Automation Studio erzeugt werden.



7.2.1.2 mapp Komponente Einfügen

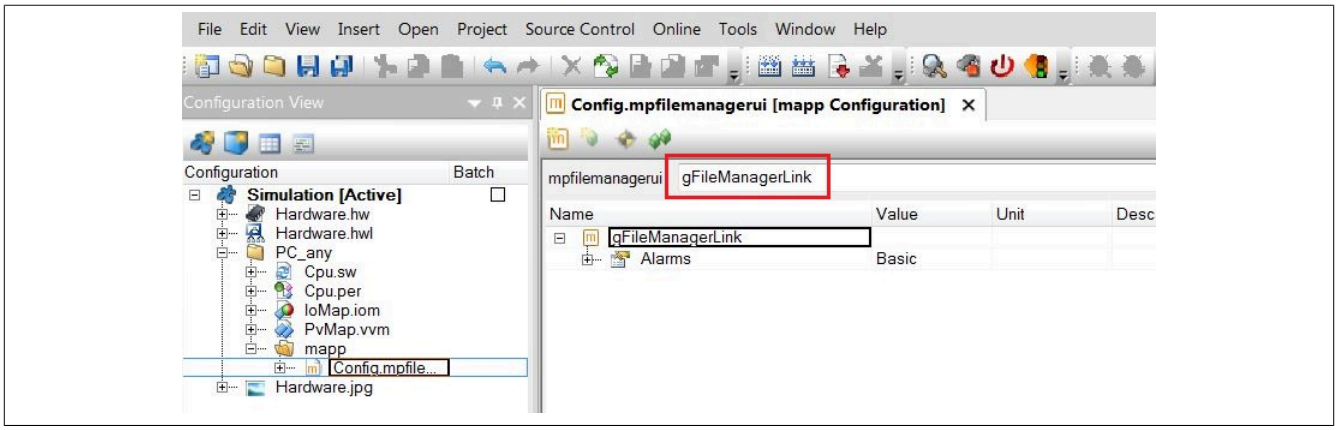
Konfiguration für MpFile einfügen

Beim nächsten Schritt wird über die Toolbox die gewünschte mapp Komponente ausgewählt und die Konfiguration dann per Drag&Drop im "mapp Services" Ordner eingefügt.



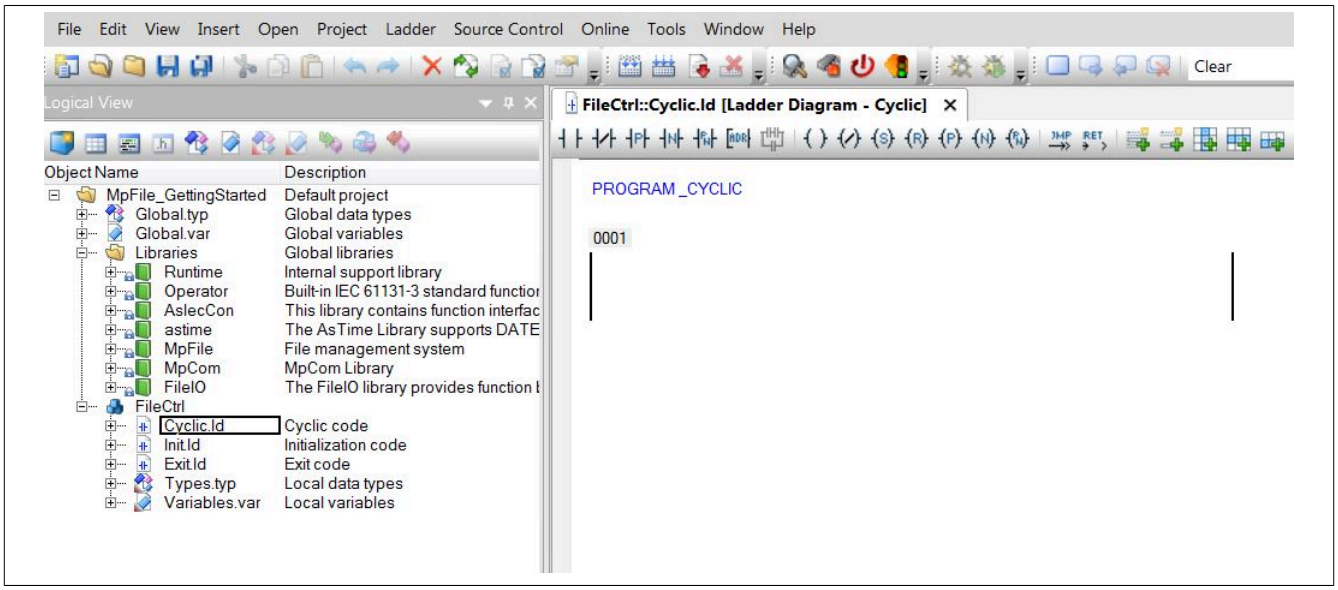
MpLink benennen

Beim Einfügen der Konfiguration wurde der MpLink automatisch erstellt. Wenn die Konfiguration geöffnet wird, ist der MpLink ganz oben zu finden.



7.2.1.3 Programm hinzufügen

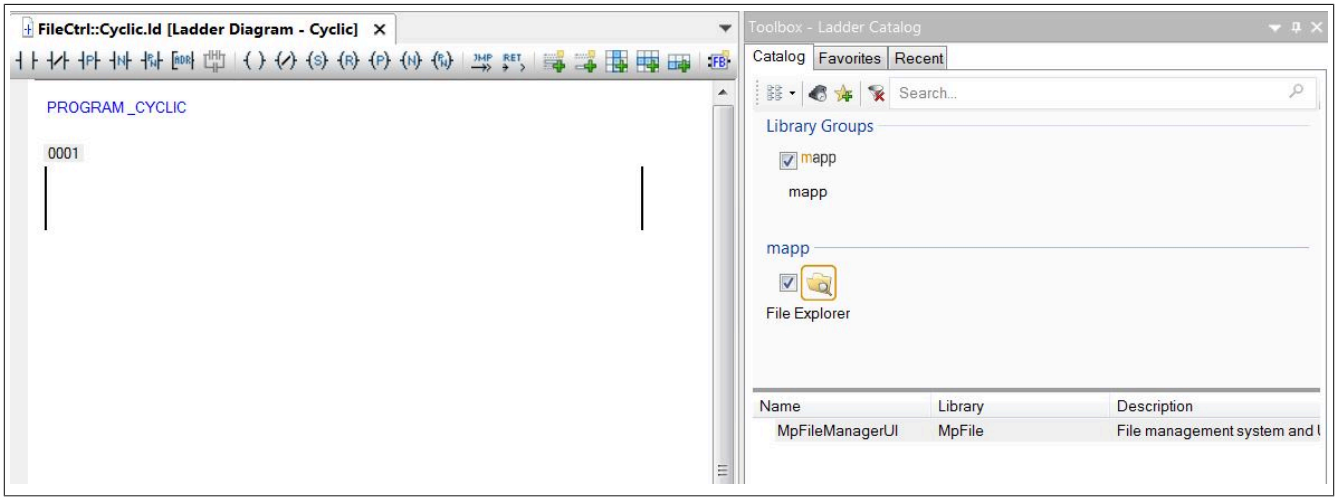
Der nächste Schritt ist das Einfügen eines Programms. In diesem Fall wird ein Kontaktplan Programm eingefügt. Jede andere Programmiersprache kann genauso verwendet werden.



7.2.1.4 MpFileManagerUI im Programm einfügen

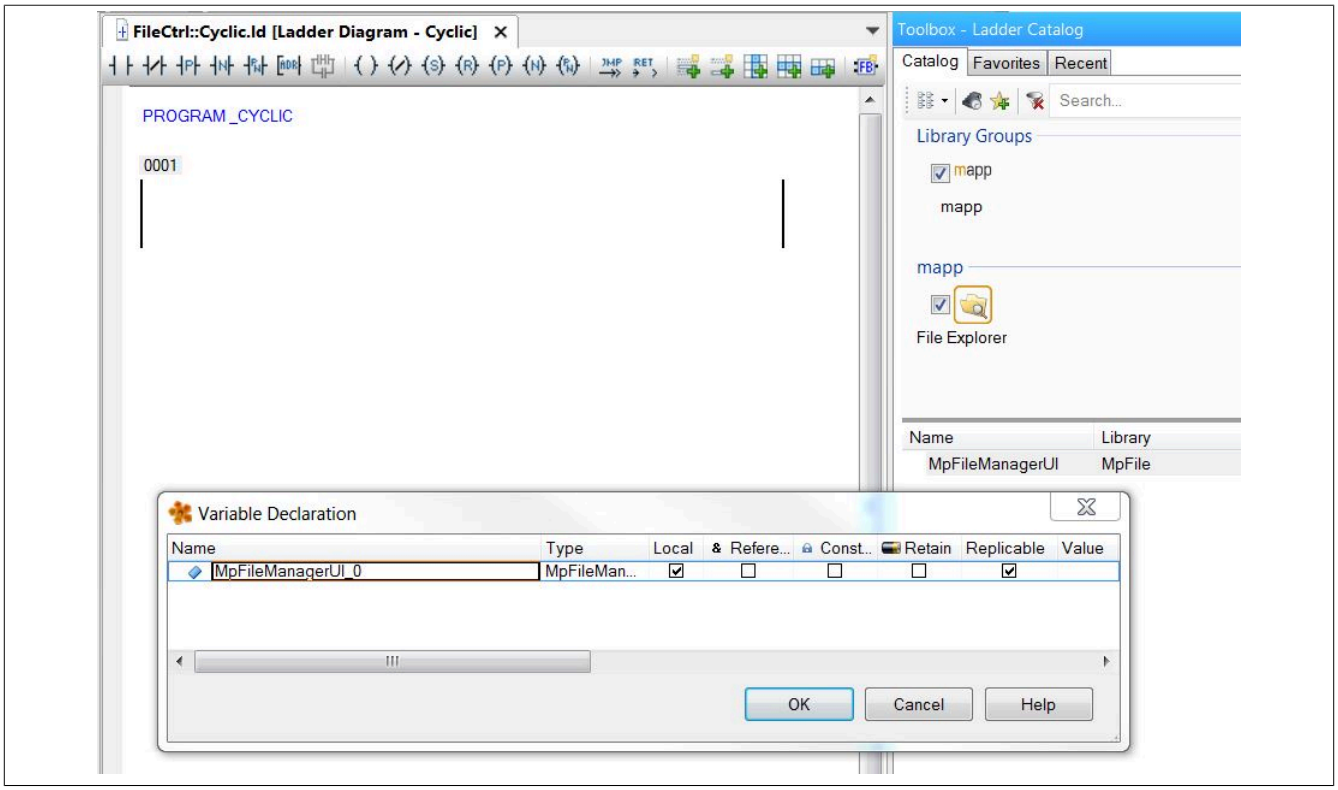
MpFile im Kontaktplan Katalog suchen

Im Katalog in Kontaktplan kann die Checkbox mapp ausgewählt werden. Danach werden darunter alle verfügbaren mapp Komponenten angezeigt, welche im Projekt sind. In unserem Fall zählt der *File Explorer* dazu.



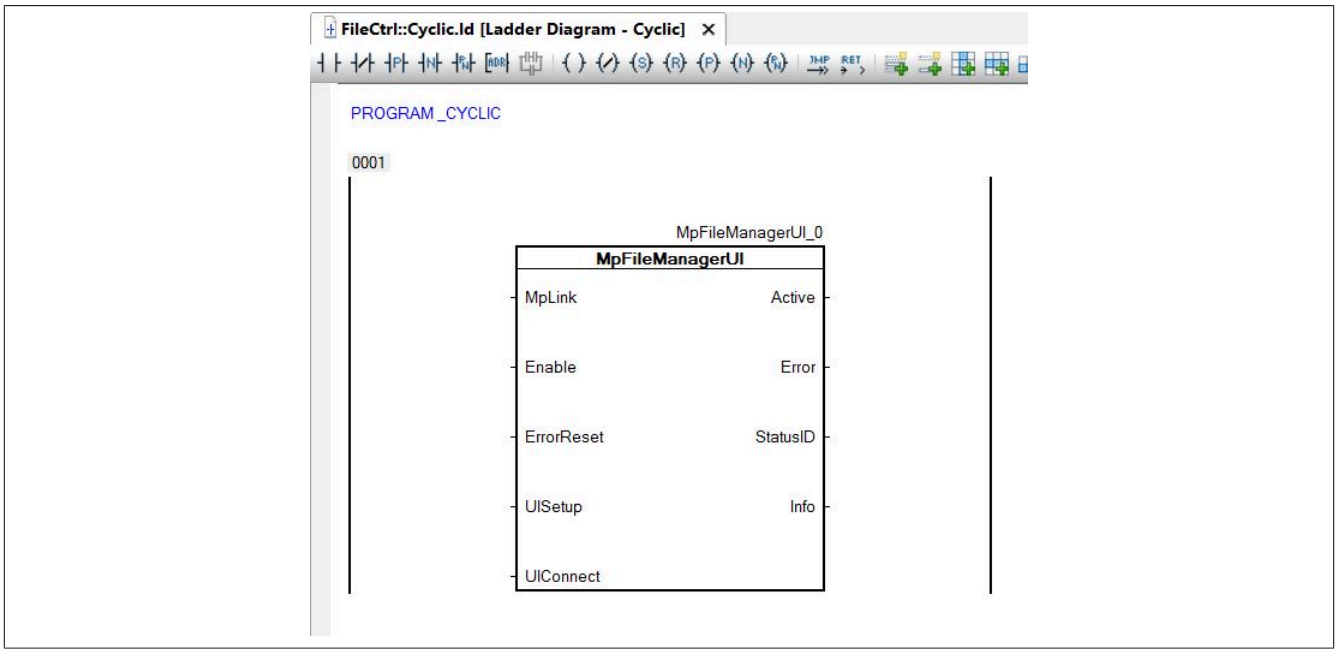
Funktionsbaustein auswählen

Über Drag & Drop wird der Funktionsbaustein MpFileManagerUI in das Programm gezogen.



Funktionsbaustein einfügen

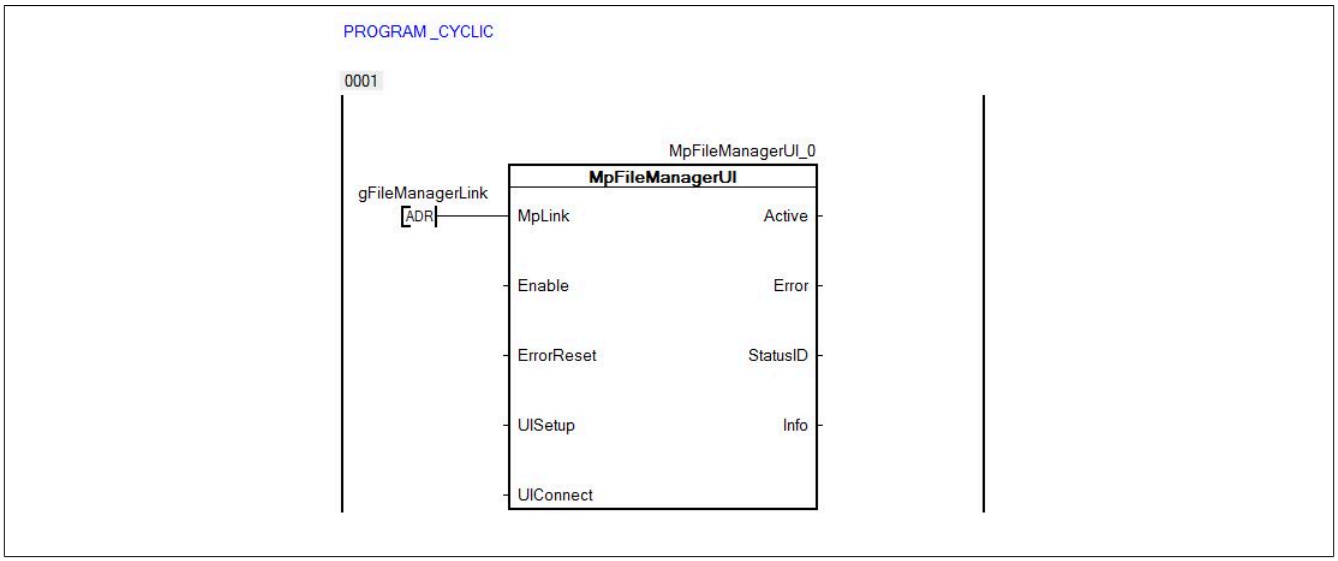
Nachdem wir den Funktionsbaustein ins Programm gezogen haben, wird eine Funktionsbaustein-Instanz erzeugt.



7.2.1.5 Parametrieren von MpFileManagerUI

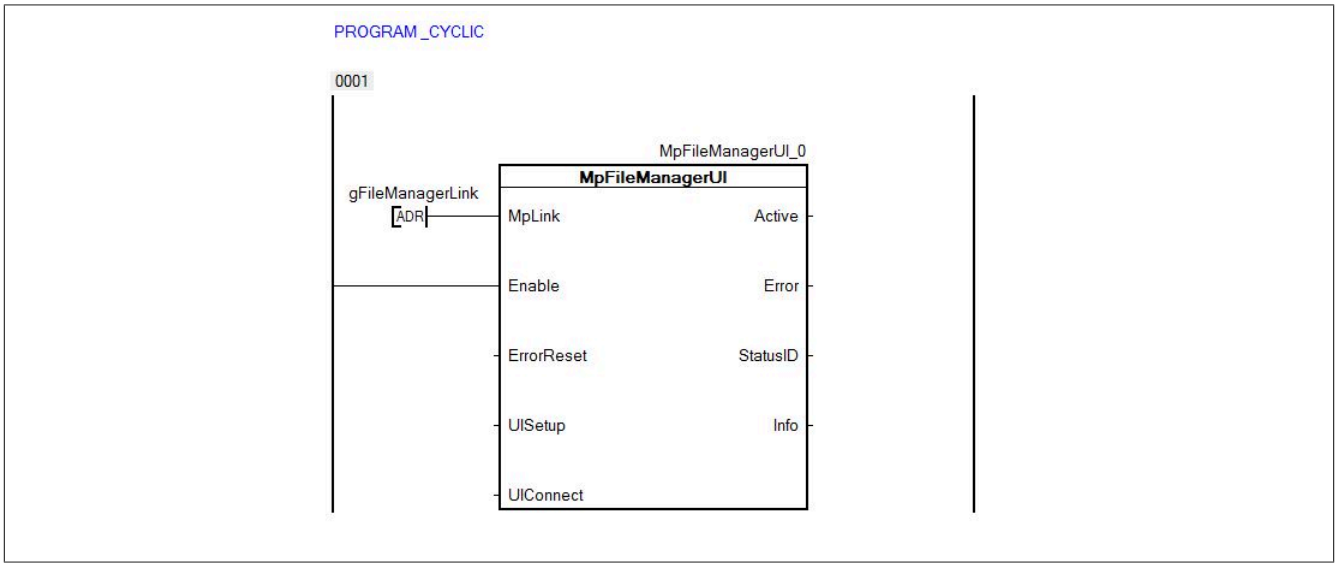
Verbinden des MpLinks

Zuerst wird der Eingang "MpLink" mit dem in der Configuration View bereits erstellten MpLink verbunden. Die Adresse wird an den Funktionsbaustein übergeben.



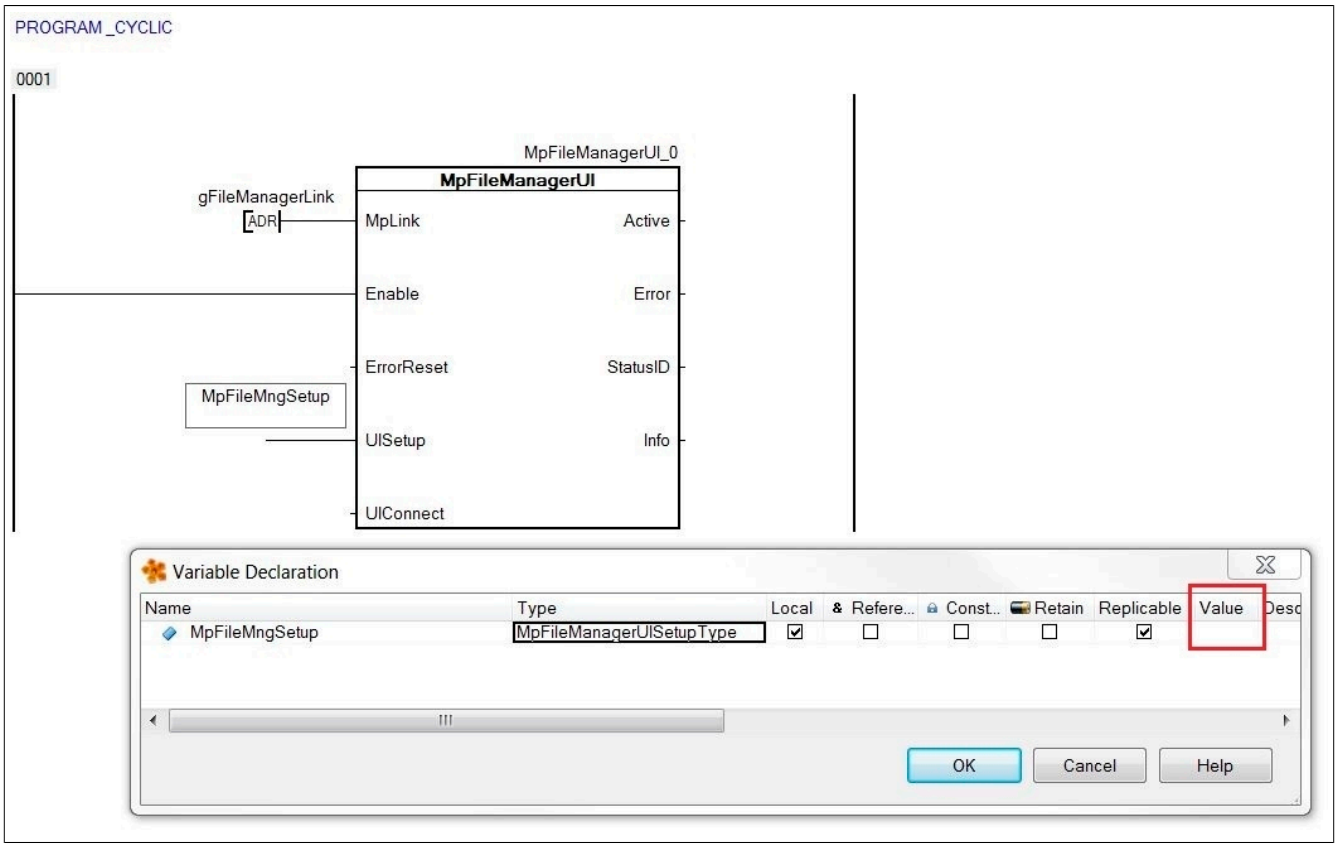
Aktivieren der Komponente

Durch Setzen des "Enable" Eingangs wird die Komponente aktiviert. Das wird über den Ausgang "Active" angezeigt.



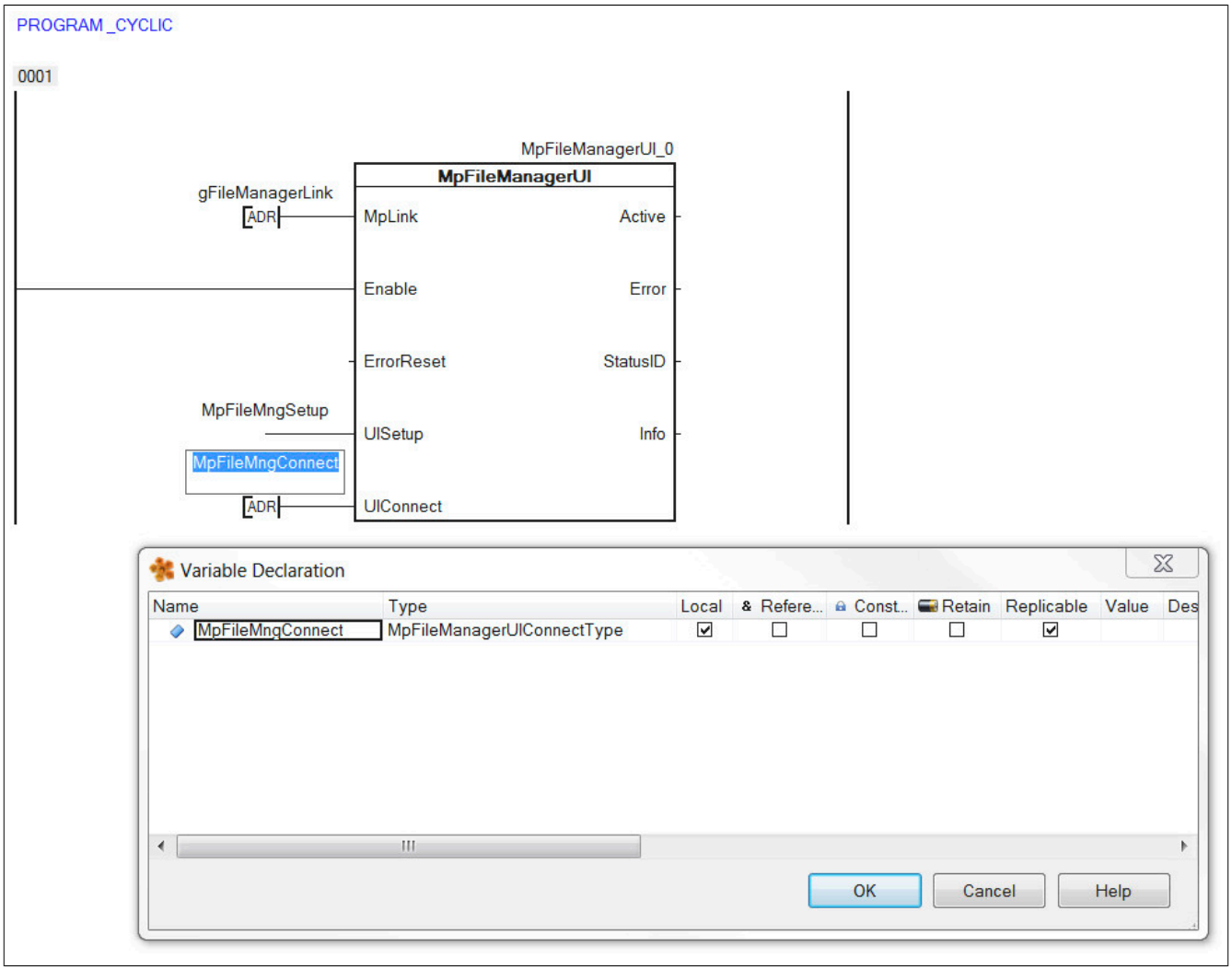
UISetup bestimmen

Nun muss eine UISetup Variable erstellt werden. Wichtig hierbei ist, dass das Feld "Value" leer ist.



UIConnect bestimmen

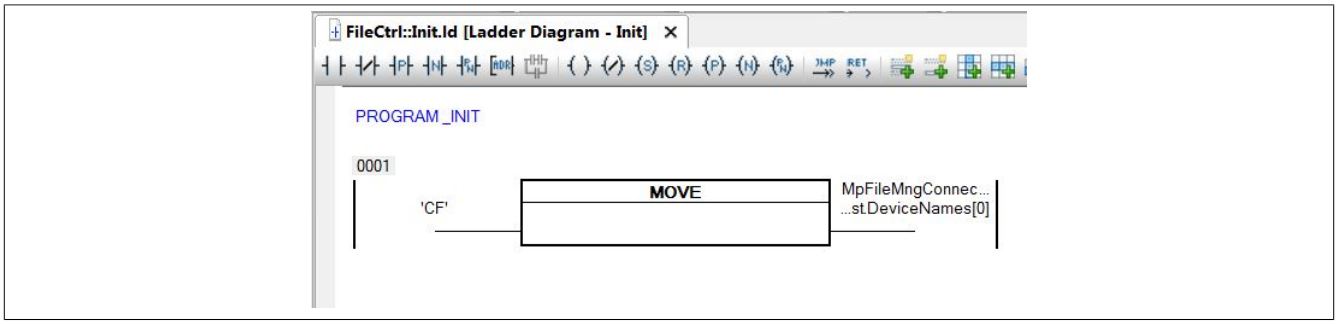
Im nächsten Schritt muss eine UIConnect Variable erstellt werden. Diese wird als Adresse übergeben.



Speichermedium bestimmen

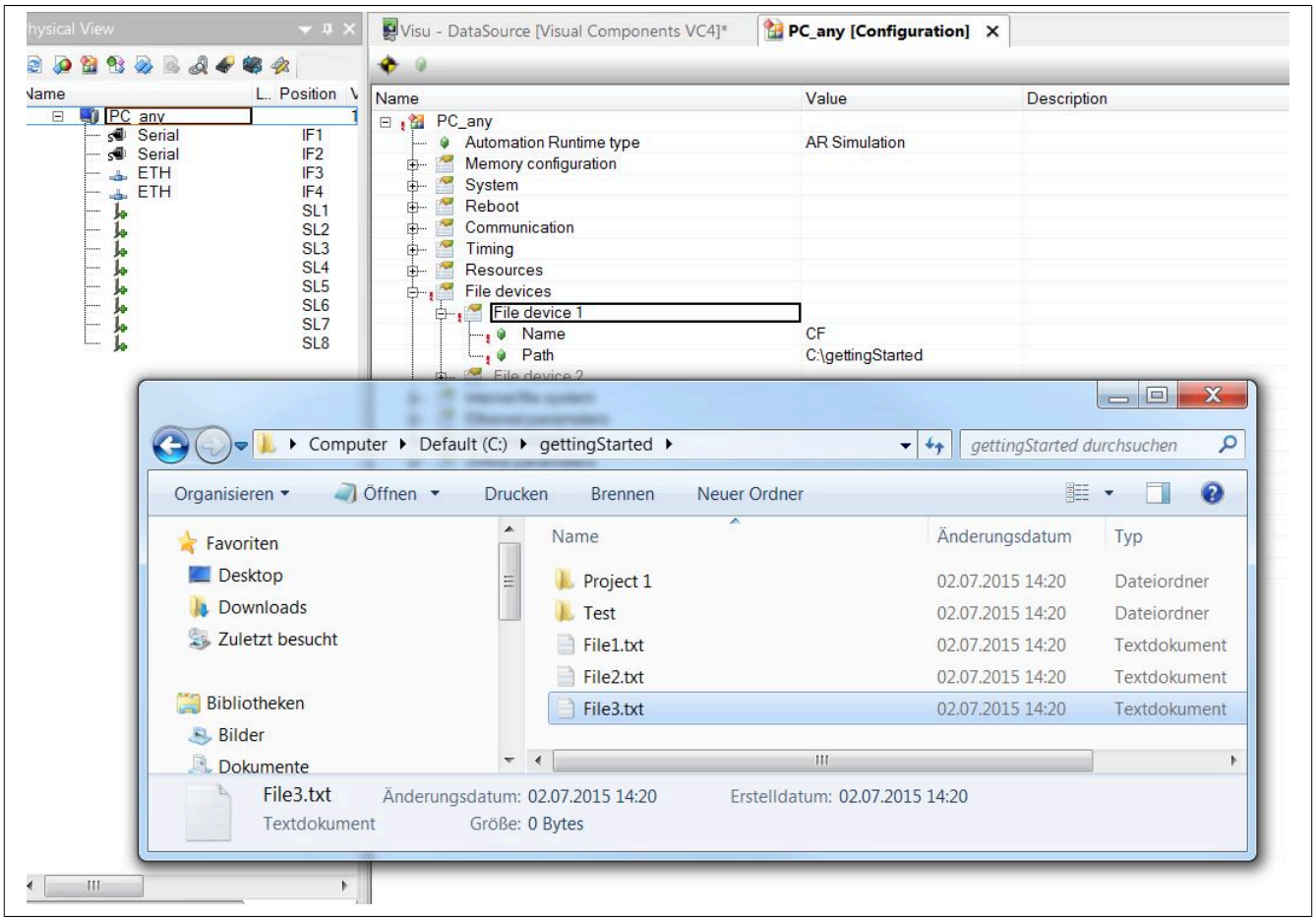
Im Speichemedium werden die Dateien die im Datei Explorer erstellt werden angezeigt.

Das Speichermedium für den Datei Explorer wird über "MpFileMngConnect.DeviceList.DeviceNames[x]" im Init Programm bestimmt. Insgesamt können 10 Speichermedien genutzt werden.



7.2.1.6 Speichermedium erstellen

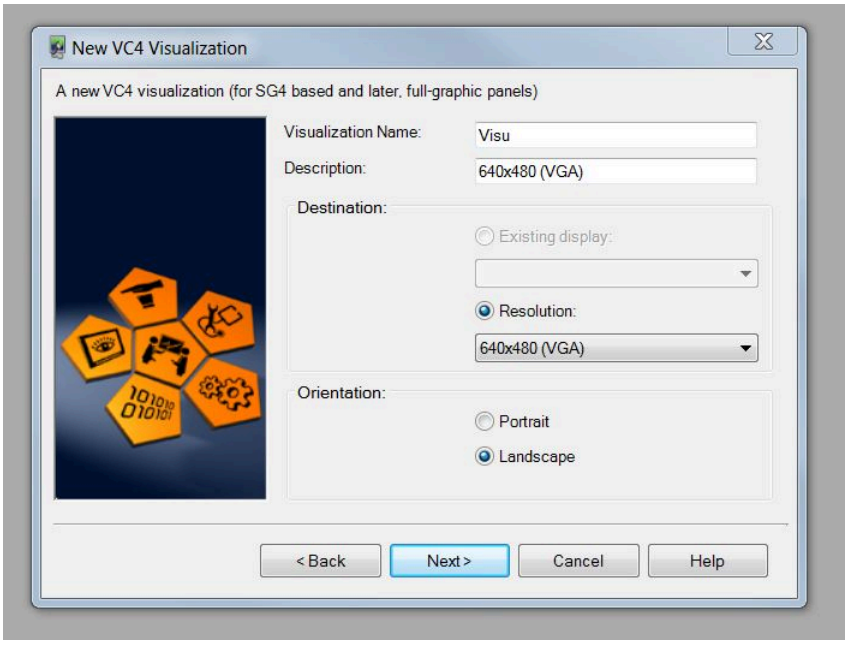
Als nächstes muss ein Speichermedium ("CF") erstellt werden. Dazu muss ein Ordner auf Windows ausgewählt werden / erstellt werden. Danach muss in der Automation Runtime für diesen Ordner ein Speichermedium (File Device) erstellt werden.



7.2.1.7 Visualisierung erstellen

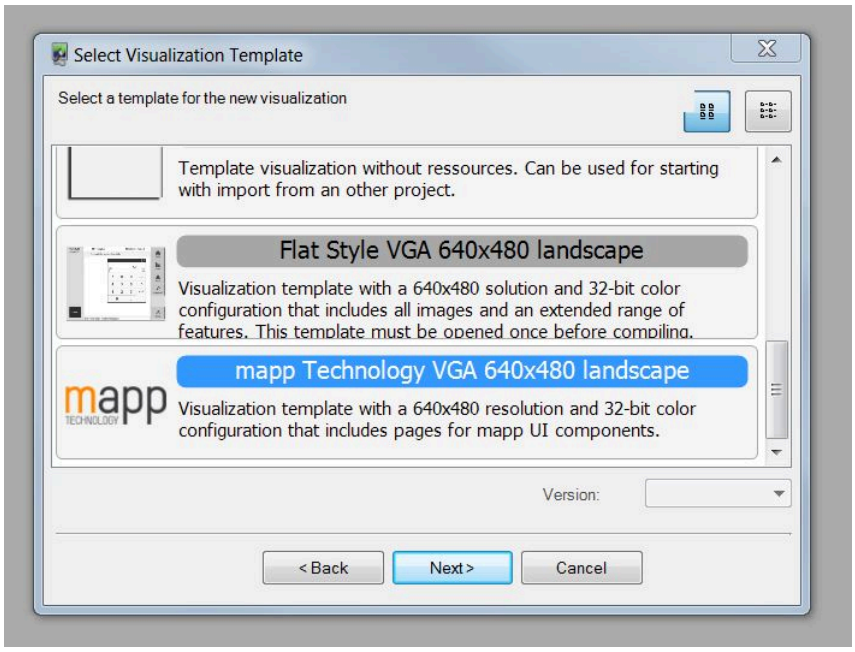
Visualisierung einfügen

Um den Datei Explorer anzeigen zu können, wird eine VC4 Visualisierung erstellt.



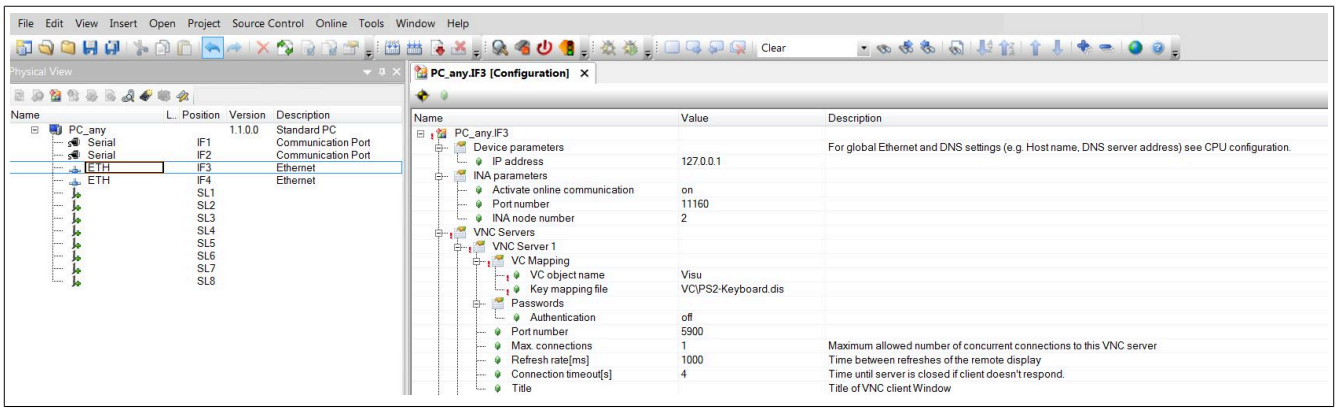
Vorlage auswählen

Damit die mapp Visualisierung angezeigt werden kann, muss die Vorlage "mapp Technology VGA 640x480 landscape" ausgewählt werden.



VNC Server für die Visualisierung konfigurieren

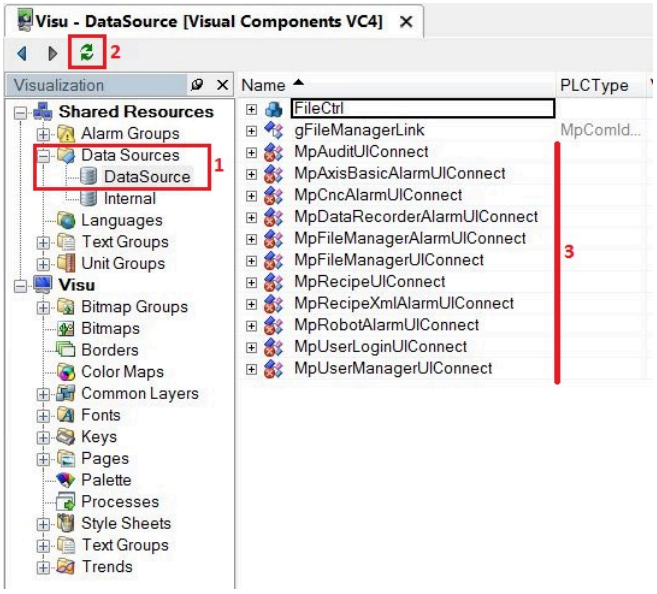
Danach wird die Visualisierung mit einem VNC Server verbunden.



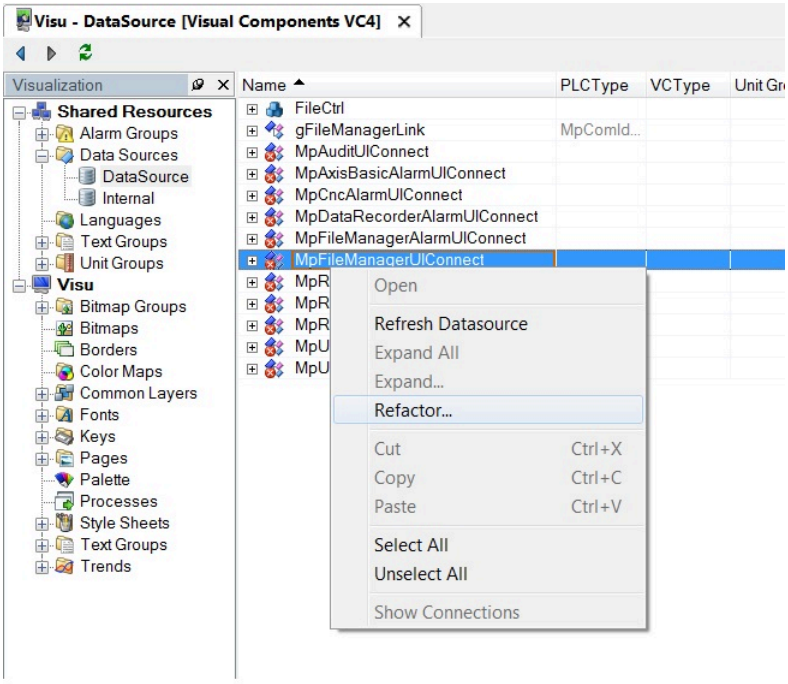
7.2.1.8 Anbinden der Vorlage an das eigene Projekt

Erstellen der Verbindung zwischen Task und Visualisierung

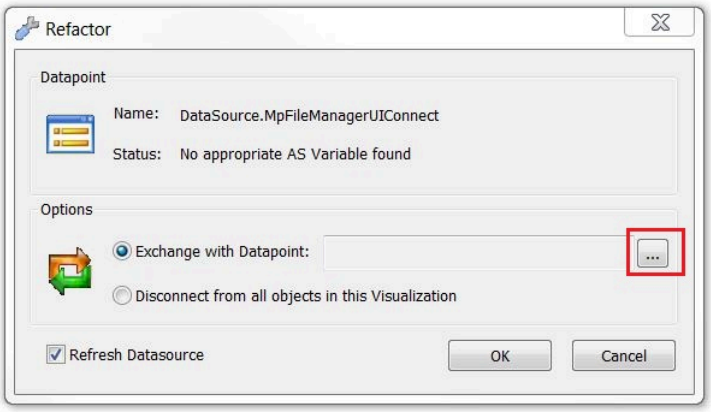
Nach dem Öffnen der Visualisierung wird zunächst die Datenquelle ausgewählt[1] und aktualisiert[2]. Nach dem aktualisieren wird sichtbar welche Variablen in der Visualisierung verbunden aber im Projekt nicht vorhanden sind[3].

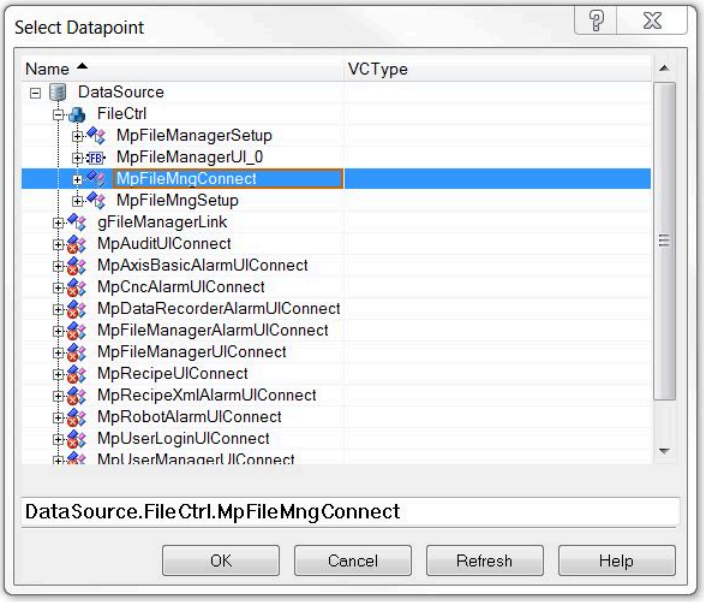


Aus dieser Liste wird die Variable ausgewählt welche mit der eigenen ersetzt werden soll. Mit einem Rechtsklick auf "MpFileManagerUI" wählt man den Eintrag „Refactor“ aus.

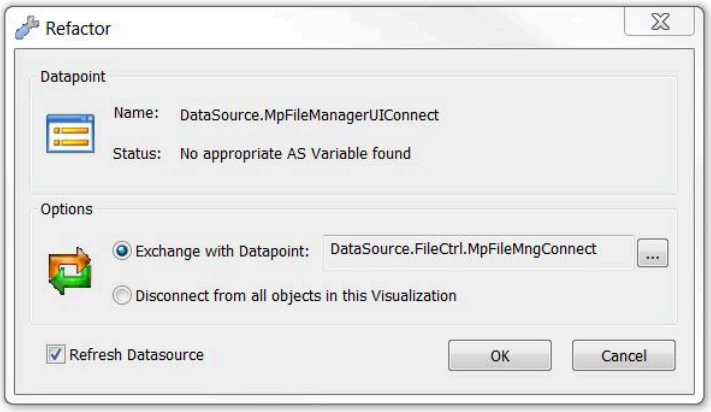


Jetzt wird über das Dialogfenster, welches sich öffnet, die eigene UIConnect Struktur "MpFileMngConnect" ausgewählt.

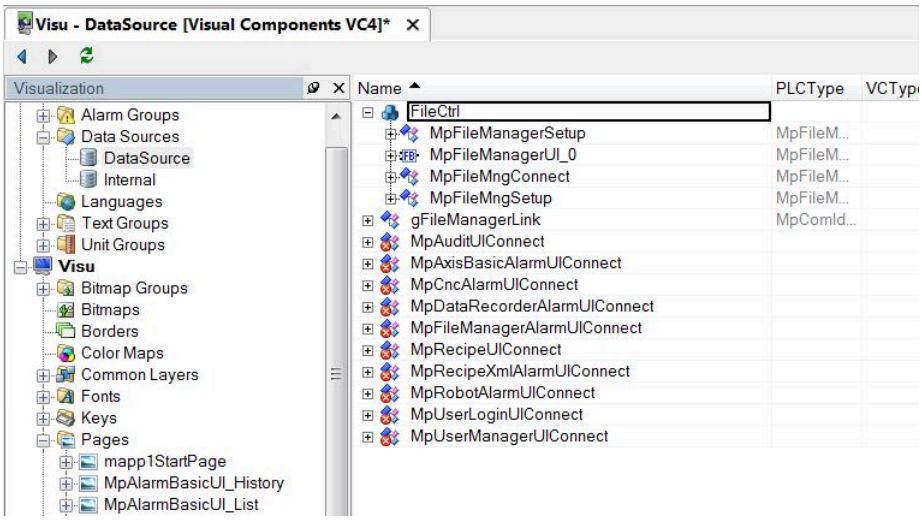




Danach wird der Austausch bestätigt. Alle Verbindungen werden jetzt automatisch erstellt.



Nach dem "Refactoring" wird die Datenquelle wieder aktualisiert. Jetzt sollte die vorhergehende Verbindung "MpFileManagerUI" nicht mehr vorhanden sein.



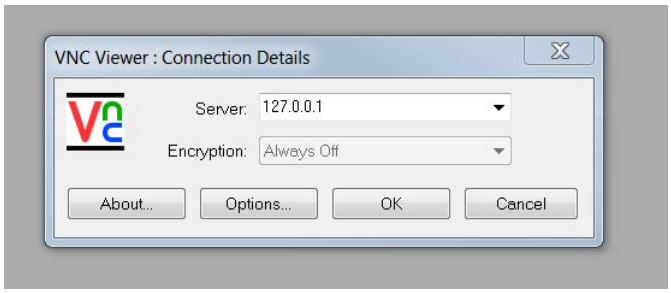
7.2.1.9 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

7.2.1.10 Visualisierung testen

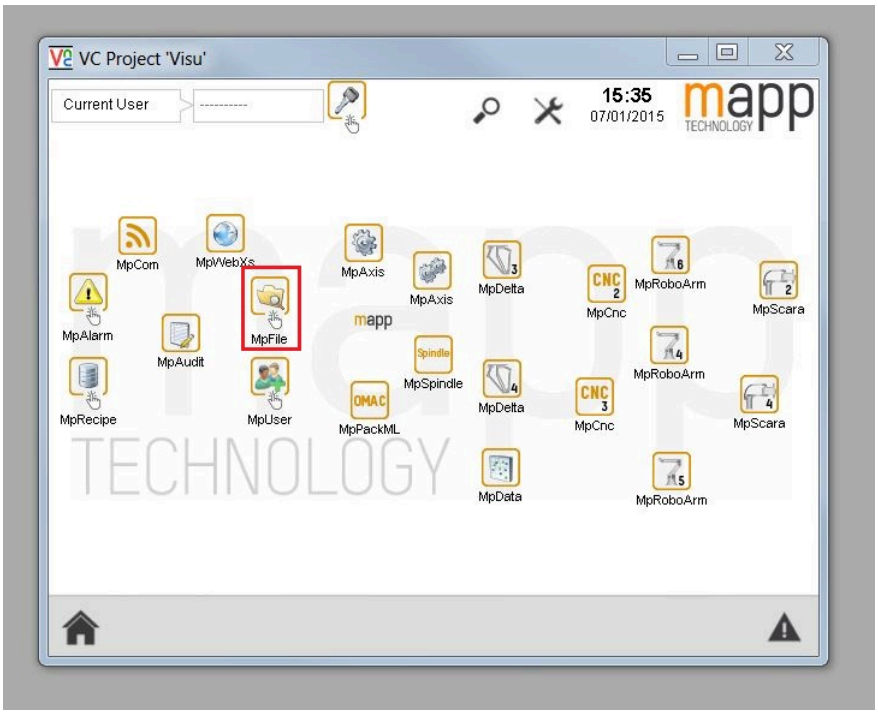
VNC Viewer öffnen

Die Visualisierung kann über den VNC Viewer geöffnet werden.



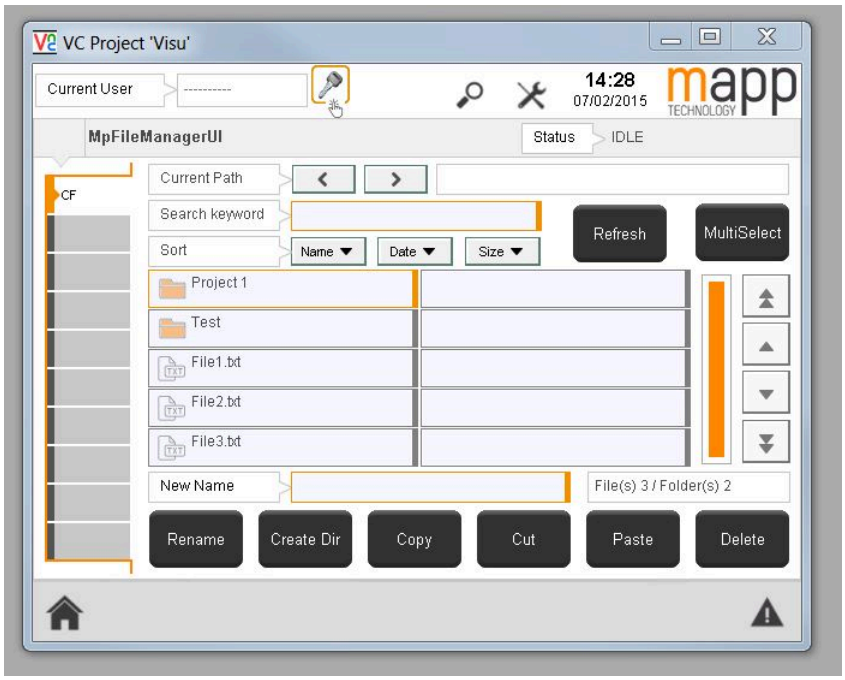
Startseite der Visualisierung

Hat man sich erfolgreich mit der Visualisierung durch den VNC Viewer verbunden, wird die Startseite angezeigt. Durch klicken des Buttons "MpFile" erscheint der Datei Explorer.



Datei Explorer

Der Datei Explorer ist nun funktionsfähig. Alle benötigten Datenpunkte sind bereits verbunden. Es können Order erstellt, gelöscht etc. werden.



7.2.2 Texteditor für CNC Programme in mapp View erstellen

In diesem Abschnitt wird Schritt für Schritt erklärt wie man mithilfe der MpFileManager Konfiguration und dem MotionPad-Widget einen Texteditor für CNC Programme erstellen kann.

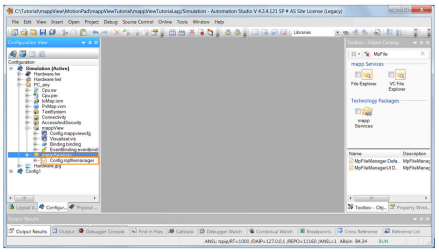
7.2.2.1 Projekt erstellen

Zuerst muss ein neues Projekt über Automation Studio erzeugt werden. Für mehr Information siehe hier.

7.2.2.2 mapp Komponente einfügen

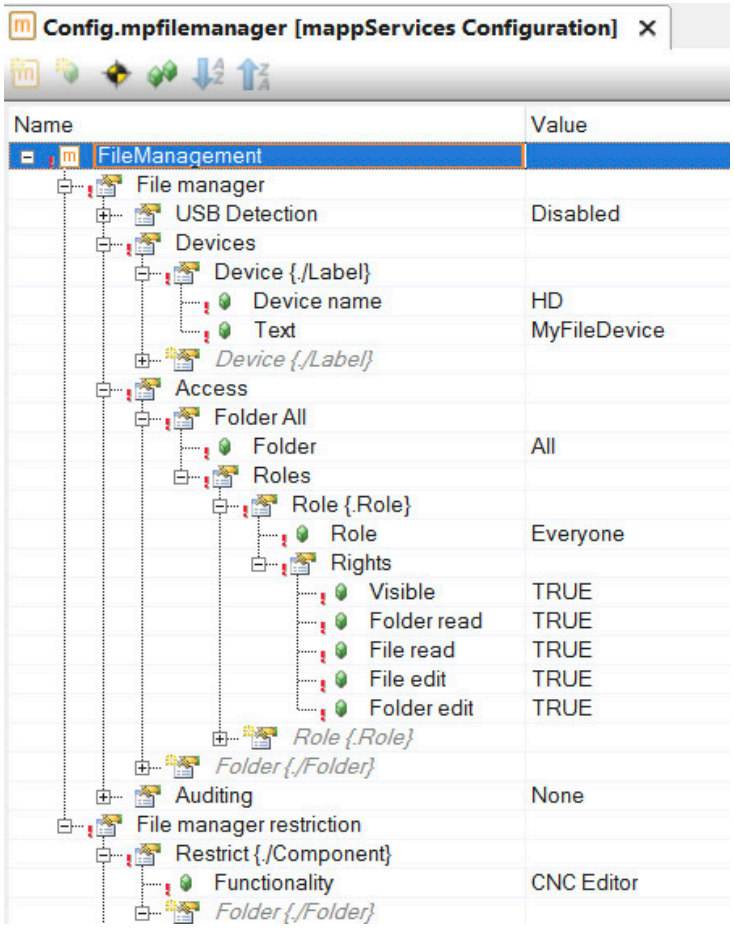
Konfiguration MpFileManager einfügen

Damit CNC Programme später geladen und auch abgespeichert werden können, muss die MpFileManager Konfiguration eingefügt werden.



Unter "Devices" muss das Speichermedium angegeben werden, von dem aus Programme geladen oder gespeichert werden können. Dies wurde zuvor in der Konfiguration des Zielsystems mit dem Namen "HD" erstellt. Unter welchen Namen das Speichermedium in der Visualisierung erscheint, wird mithilfe von "Name" bestimmt. Es wird "MyFileDevice" angegeben.

Unter Access wird bestimmt welche Rollen auf welche Speichermedien Zugriff haben. Mithilfe von "All" haben alle angegebenen Rollen unter "Role" Zugriff auf die unter "Devices" angegebenen Speichermedien. Unter "Role" wird die Rolle "Everyone" angegeben, somit hat jeder Zugriff auf das Speichermedium. Zum Schluss muss noch die Funktionalität festgelegt werden. Es wird "CNCEditor" ausgewählt.



7.2.2.3 mapp View Visualisierung hinzufügen

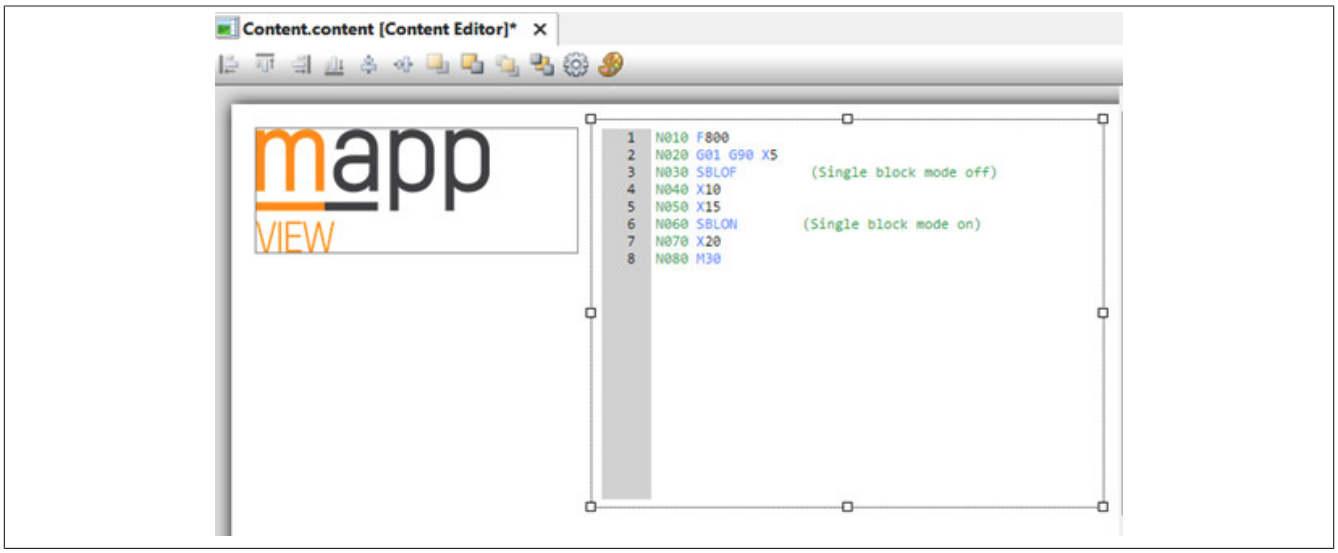
Im nächsten Schritt wird eine mapp View Visualisierung hinzugefügt. Für mehr Information siehe hier.

Die Visualisierung kann folgendermaßen aussehen:

7.2.2.4 Widget einfügen und konfigurieren

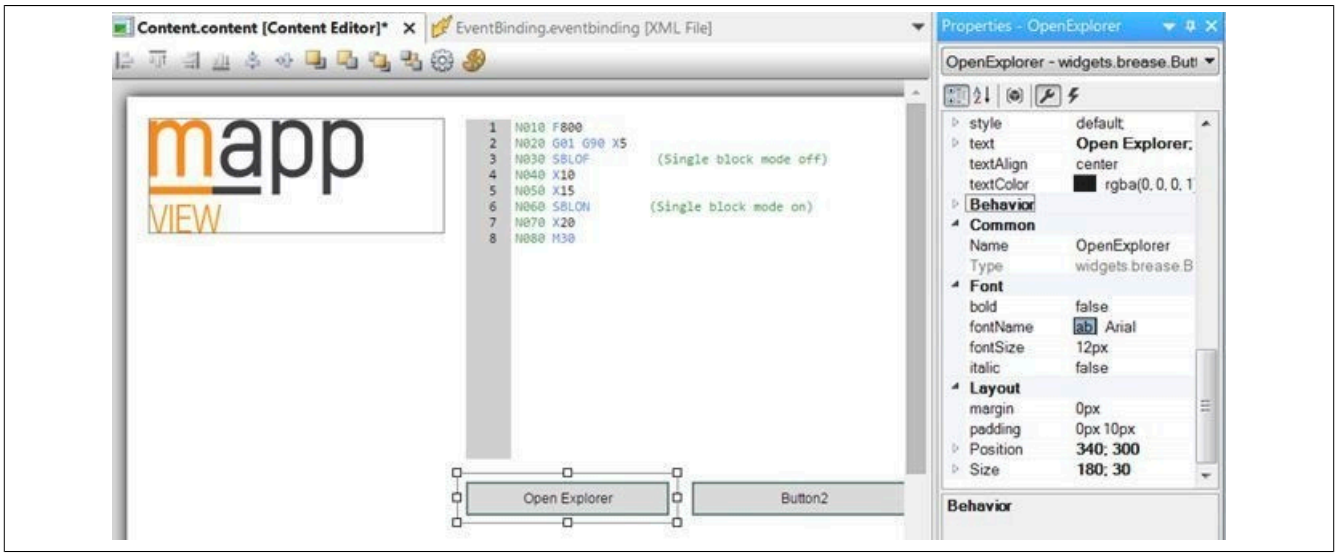
Als nächstes wird in Automation Studio zu dem Content navigiert, welcher den CNC Editor beinhalten soll.

Es wird das MotionPad-Widget eingefügt. Über dieses Widget können CNC Programme angezeigt werden.



Um Programme zu laden und zu speichern werden zwei Buttons eingefügt.

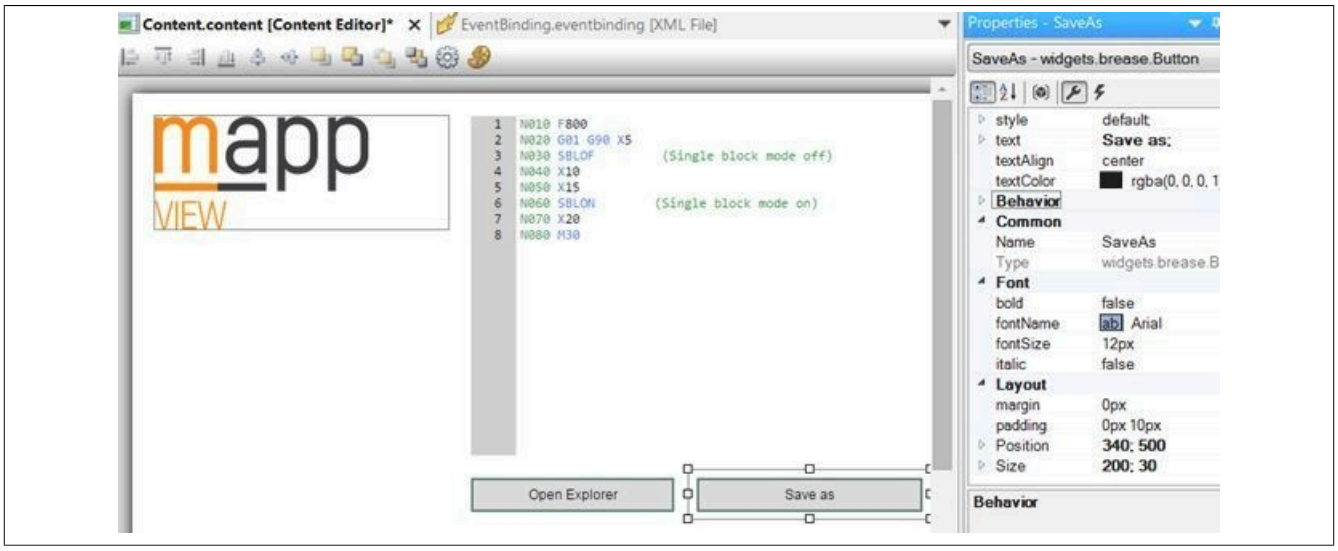
Der erste Button wird verwendet um ein CNC Programm zu laden. Wird auf den Button geklickt, soll die Aktion "OpenFileExplorer" ausgelöst werden. Dadurch wird ein Dateibrowser geöffnet, durch welchen das gewünschte Programm ausgewählt und geladen werden kann.



Das Event-Binding kann dabei folgendermaßen aussehen:

```
<EventBinding>
    <Source xsi:type="widgets.brease.Button.Event" contentRefId="MyContent" widgetRefId="OpenExplorer" event="Clic
        <EventHandler>
            <Action>
                <Target xsi:type="widgets.brease.MotionPad.Action" contentRefId="MyContent" widgetRefId="MotionPad
                    <Method xsi:type="widgets.brease.MotionPad.Action.OpenFileExplorer" />
                </Target>
            </Action>
        </EventHandler>
</EventBinding>
```

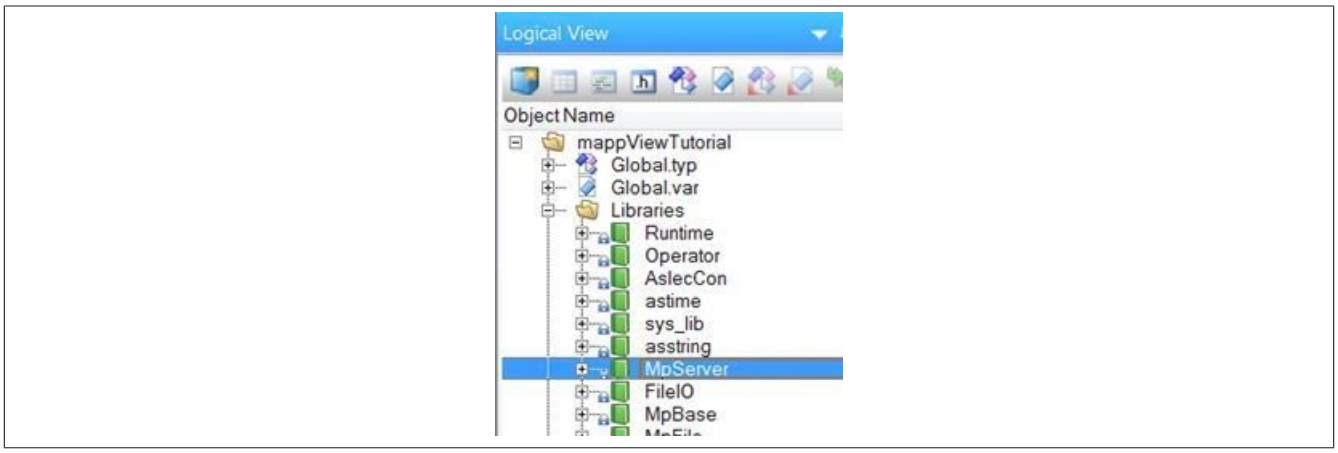
Mithilfe des zweiten Button sollen Programme abgespeichert werden. Aus diesem Grund wird, bei einem Klick auf den Button, die Aktion "SaveAs" ausgelöst.



Das Event-Binding kann dabei folgendermaßen aussehen:

```
<EventBinding>
    <Source xsi:type="widgets.brease.Button.Event" contentRefId="MyContent" widgetRefId="SaveAs" event="Click" />
        <EventHandler>
            <Action>
                <Target xsi:type="widgets.brease.MotionPad.Action" contentRefId="MyContent" widgetRefId="MotionPa
                    <Method xsi:type="widgets.brease.MotionPad.Action.SaveAs" />
                </Target>
            </Action>
        </EventHandler>
</EventBinding>
```

Zuletzt muss die Bibliothek MpServer eingefügt werden. Somit sind alle notwendigen Schritte, um die Ereignisse über die Visualisierung darzustellen, abgeschlossen.



7.2.2.5 Dateistruktur erzeugen

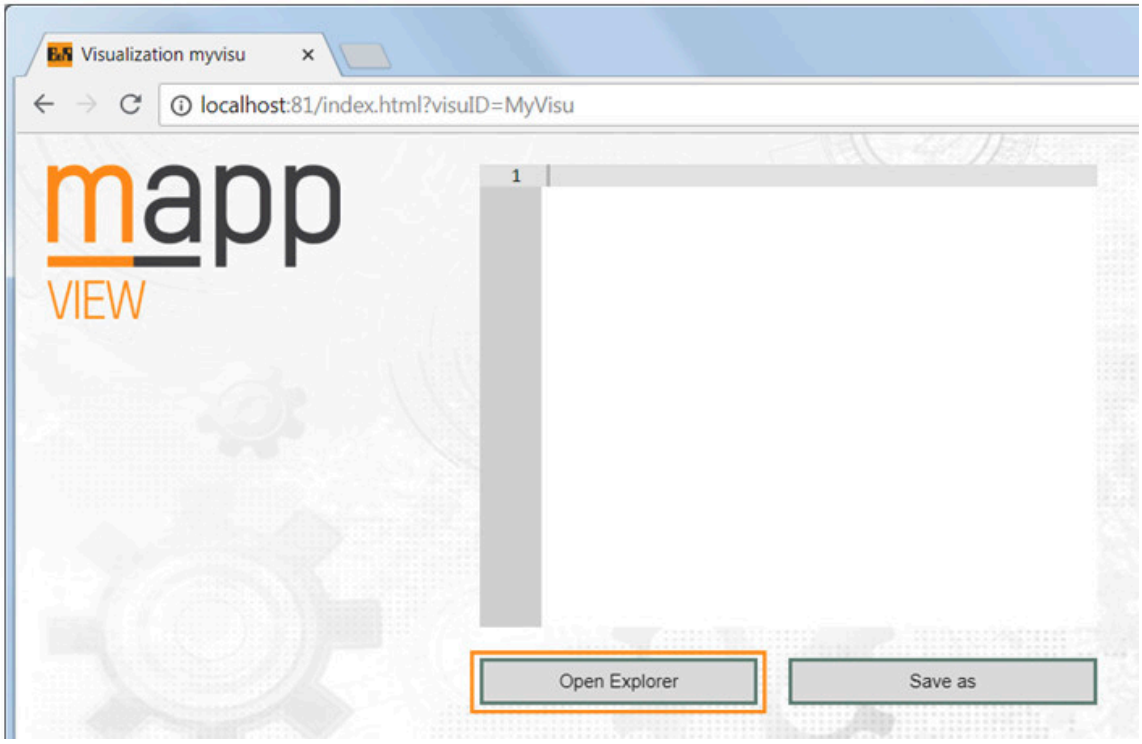
Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

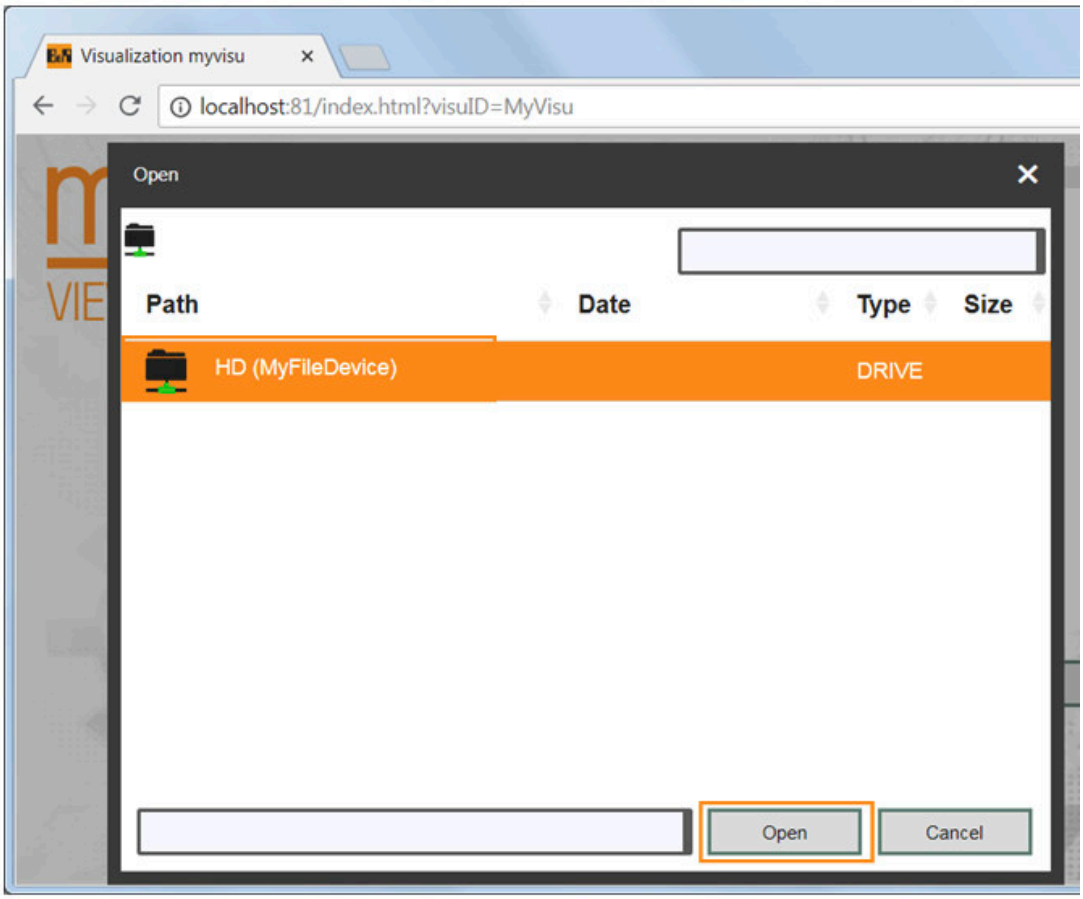
7.2.2.6 Testen des Programms

Nach einem Download der Änderungen zur Steuerung kann die Visualisierung über den Webbrowser geöffnet werden.

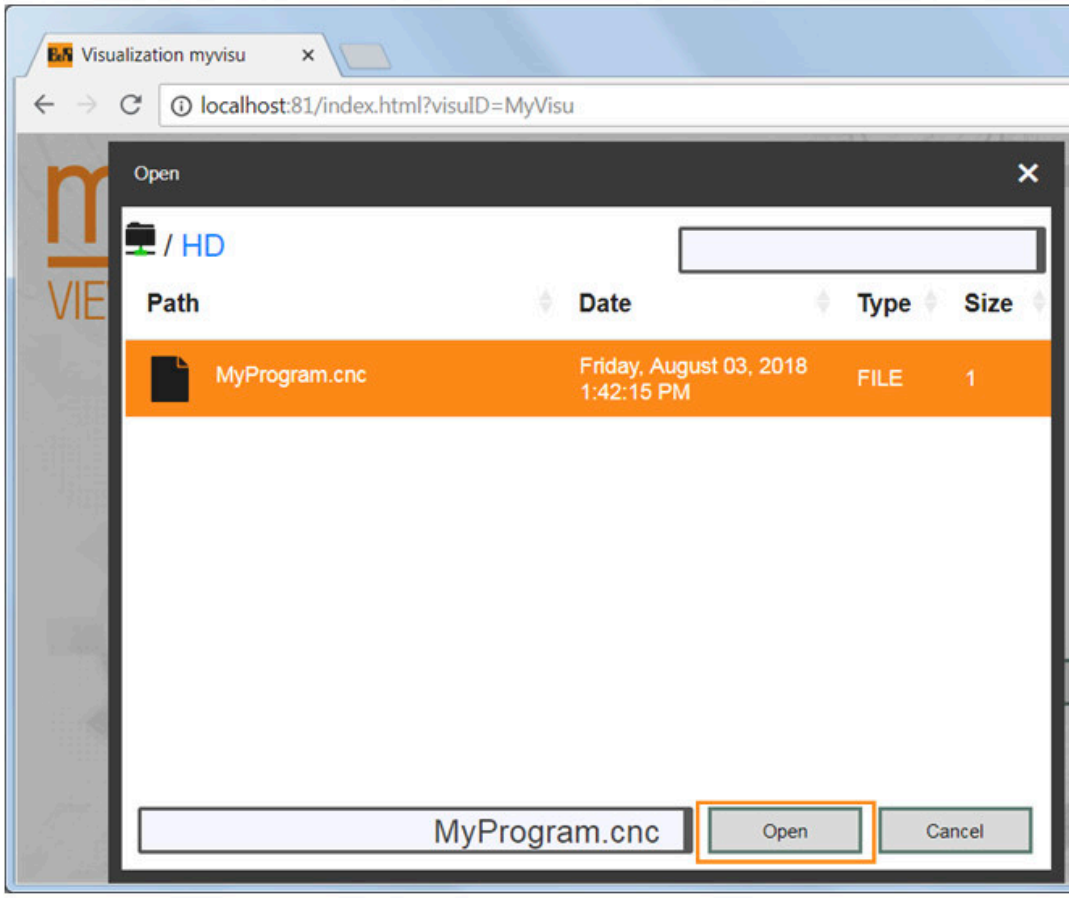
Auf dem vorher angegebenen Speichermedium befindet sich ein CNC Programm welches zuerst geladen und anschließend bearbeitet werden soll.

Dazu wird zuerst über den "Open Explorer" Button der Dateiexplorer geöffnet und das Speichermedium "MyFile-Device" selektiert.

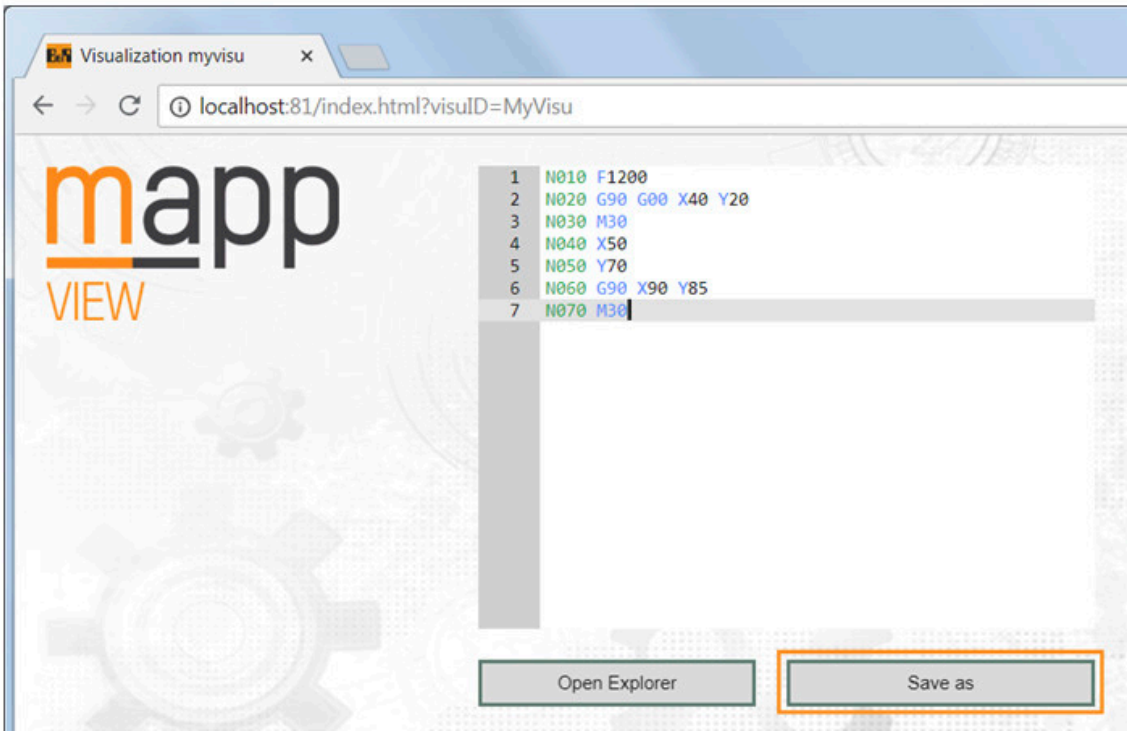




Das Programm "MyProgram.cnc" wird ausgewählt und geöffnet.



Das Programm wird nun im Browser angezeigt und kann bearbeitet werden. Die Änderungen werden mithilfe von "Save as" abgespeichert.



7.3 Konfiguration

7.3.1 MpFileManagerUI Konfiguration

Diese Konfiguration kann nur in Kombination mit MpFileManagerUI bzw. MpFileManagerConfig verwendet werden! Für mehr Information siehe hier.

Die Konfiguration kann zur Laufzeit mithilfe des Funktionsbausteins MpFileManagerConfig angepasst werden.

Über diesen Paramter "Customized file type" ist es dem Benutzer möglich seine eigenen Dateiendungen zu definieren und dafür dann in der Visualisierung ein Icon anzuzeigen. Die Konfiguration der Endung hier korrespondiert mit MpFileManagerUIItemTypeEnum. Wird eine neue Dateiendung eingegeben erhält diese den Wert mpFILE_ITEM_TYPE_USERX in der Enumeration.

7.3.2 MpFileManager Konfiguration

Alle hier konfigurierten Speichermedien/Einschränkungen etc. gelten für das gesamte Projekt. Aus diesem Grund darf die Konfiguration nur einmal eingefügt werden.

Diese Konfiguration kann nur in Kombination mit einer mapp View Visualisierung verwendet werden. Für mehr Information siehe hier!

Bei der Verwendung des Dateibrowsers in einer mapp View Visualisierung (z.B. am MotionPad-Widget) wird, sobald ein Fehler auftritt, die Fehlernummer über die Aktion "OnError" sichtbar.

Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein de-, und wieder aktiviert werden.

Die geänderten Konfigurationsparameter werden sofort von der Komponente übernommen.

Speichermedien

Unter "USB Detection" und "Devices" können Speichermedien angegeben werden. Ob das Speichermedium vorhanden ist wird erst zur Laufzeit während eines Zugriffs, wie beispielsweise das Öffnen einer Datei, geprüft. Das bedeutet auch während der Laufzeit kann das hier angegebene Speichermedium erst erstellt werden.

Im Dateibrowser einer mapp View Visualisierung (z.B. am MotionPad-Widget) erscheint eine Liste mit den hier angegebenen Speichermedien. Es kann ein Speichermedium ausgewählt und damit gearbeitet (Dateien hinzufügen, löschen etc.) werden.

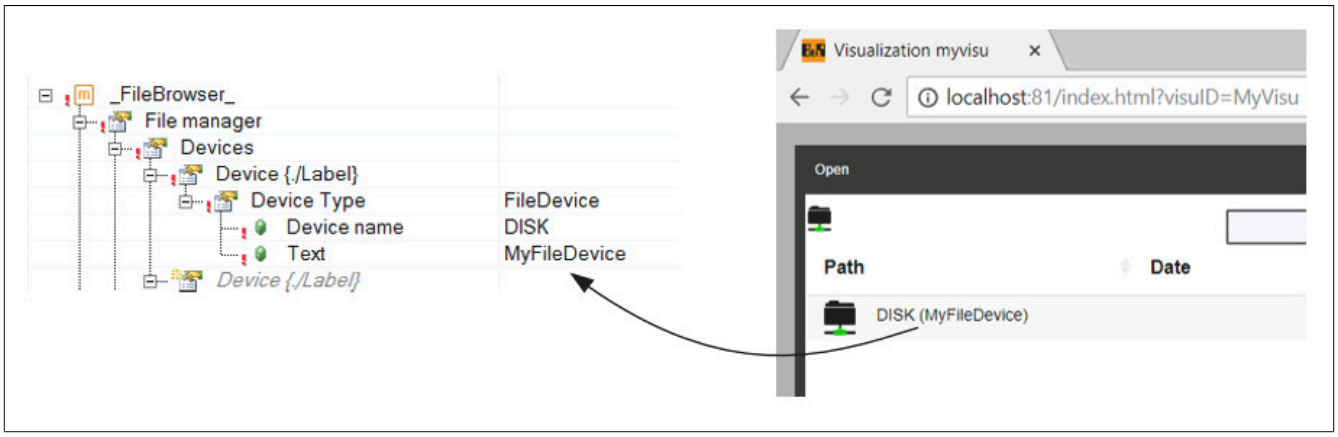
USB detection

Sollen USB-Sticks verwendet werden, so kann "USB Detection" aktiviert werden. Dadurch können ein oder mehrere USB-Sticks am Zielsystem verwendet werden. Die USB-Sticks werden automatisch erkannt. Die Sticks werden im Dateibrowser einer mapp View Visualisierung angezeigt und kann verwendet werden. Dabei wird der Name des USB-Sticks angezeigt. Der Name, welcher angezeigt wird, ist Herstellerspezifisch. Der USB Stick kann auch erst während der Laufzeit gesteckt werden.

mapp File prüft ob ein USB-Stick vorhanden ist oder nicht. Wird der USB-Stick am Zielsystem erst zu einem späteren Zeitpunkt gesteckt, so wird er, sobald er erkannt wurde, automatisch im Dateibrowser einer mapp View Visualisierung (z.B. am MotionPad-Widget) angezeigt.

Devices

Bei der Verwendung von "Devices", kann ein Speichermedium angegeben werden, welches bereits in der Konfiguration des Zielsystems definiert oder zur Laufzeit erstellt wird. Unter welchen Namen das Speichermedium in der Visualisierung erscheint, wird mithilfe von "Name" bestimmt. mapp File prüft die Gültigkeit des Speichermediums. Wird dieses zum Beispiel falsch angegeben (statt HD wird HC verwendet), so wird das Speichermedium im Dateibrowser der mapp View Visualisierung nicht angezeigt.



Zugriffsrechte

Unter "Access" können die Zugriffsrechte auf Ordner Rollenbasierend angegeben werden. Für mehr Information zum Benutzer-Rollensystem siehe hier. Unter "Folder" kann ein Ordner angegeben werden, oder es wird "All" ausgewählt.

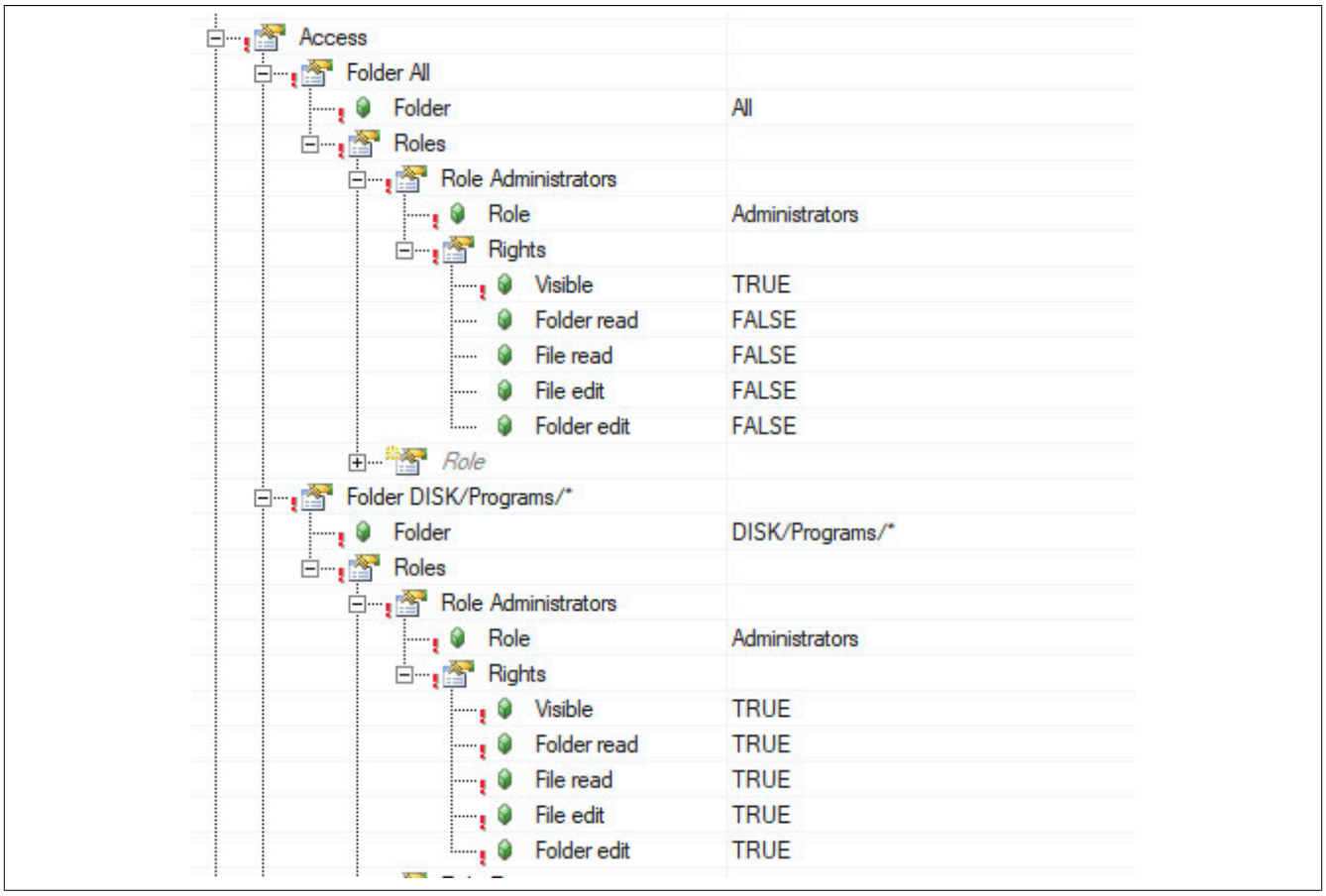
Mithilfe von "All" haben alle angegebenen Rollen Zugriff auf die unter "Devices" angegebenen Speichermedien.

Wird ein Ordner angegeben, so haben die Rollen nur Zugriff auf diesen Ordner. Die Angabe dazu kann zum Beispiel folgendermaßen aussehen: "DISK\Programs*"
Dadurch hat die Rolle Zugriffsrechte auf alle Ordner welche sich im Ordner "Programs" befinden. "DISK" ist dabei das Speichermedium, welches in der Konfiguration des Zielsystems angegeben wurde.

Unter "Roles", kann eine im Benutzer-Rollensystem angelegte Rolle ausgewählt werden. Anschließend kann definiert werden, welche Zugriffsrechte die Rolle besitzt. Folgendes kann festgelegt werden:

- **Visible:** Bei "Visible = `TRUE`" sind die Ordner für die Rolle sichtbar
- **Folder read:** Bei "Folder read = `TRUE`" kann die Rolle auf Ordner zugreifen und navigieren
- **File read:** Bei "File read = `TRUE`" kann die Rolle Dateien lesen
- **File edit:** Bei "File edit = `TRUE`" kann die Rolle Dateien editieren
- **Folder edit:** Bei "Folder edit = `TRUE`" kann die Rolle Ordner editieren und zum Beispiel umbenennen

Es gilt zu beachten, dass nur bei der Angabe einer Zugriffsberechtigung von "DISK\Programs*" keine Ordner angezeigt werden können, da keine Rolle die Erlaubnis hat auf Ordner "DISK" zuzugreifen. Aus diesem Grund muss auch eine Zugriffsberechtigung auf "DISK" gegeben werden. Die angegebenen Zugriffsberechtigungen werden mit einer "OR" Verknüpfung verbunden. Eine korrekte Angabe könnte folgendermaßen aussehen:



In diesem Fall hat die Rolle "Administrator" ein "Visible" Recht auf alle Ordner innerhalb des Speichermediums "DISK". Dateien und Ordner ansehen sowie bearbeiten kann die Rolle "Administrator" aber nur die Dateien im Ordner "Programs".

Auditing

Über "Auditing" kann festgelegt werden, ob Ereignisse in mapp File von mapp Audit aufgezeichnet werden sollen. Für mehr Information siehe Auditing des Dateibrowsers.

Es kann das Editieren ("Edit") und/oder Lesen ("Read") von Ordner und/oder Dateien aufgezeichnet werden.

Funktionsbezogene-Einschränkungen

Für unter "Folder" angegebene Ordner können Einschränkungen für verschieden Nutzer der Dateiverwaltung (z.B. MotionPad-Widget) definiert werden. Dabei muss der Pfad des Ordners angegeben werden. Die Rollenbasierten-Rechte gelten dabei immer noch!

CNC Editor: Bei der Verwendung des MotionPad-Widget wird CNC Editor ausgewählt.

Text Editor: Bei der Verwendung des TextPad-Widget wird Text Editor ausgewählt.

Folgende Einschränkungen können durchgeführt werden:

- **Browse up:** Bei "Browse up = `TRUE`" wird der Zugriff auf übergeordnete Ordner erlaubt
- **Browse down:** Bei "Browse down = `TRUE`" wird der Zugriff auf Unterordner erlaubt
- **Create folders:** Bei "Create folders = `TRUE`" wird das Erstellen weiterer Ordner erlaubt
- **Read only:** Bei "Read only= `TRUE`" kann auf die Inhalte nur lesend zugegriffen werden

7.4 Anwendungsfälle

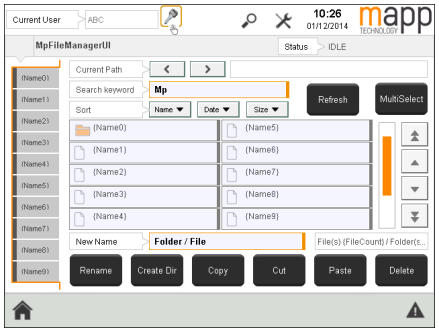
In diesem Abschnitt werden mögliche Anwendungsfälle der MpFile Komponenten beschrieben.

7.4.1 Dateiverwaltungssystem für unterschiedliche Speichermedien

Anforderung

Aufgabe ist es, ein Dateiverwaltungssystem auf der Steuerung zu realisieren.

Es soll möglich sein, verschiedene Datenquellen wie CF, USB und FTP zu verwalten. Weiterhin sollten grundlegende Dateisystem Funktionalitäten (Kopieren, Umbenennen, usw.) und die VC4 Visualisierungsanbindung an das mapp Dateiverwaltungssystem implementiert werden.

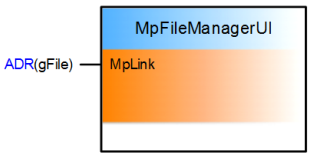


Lösung

Komponentenliste

- ***MpFileManagerUI*** (Eigener MpLink): Dateiverwaltungssystem mit VC4 Anbindung

Verbindungsschema



Verwendung der mapp Komponenten

MpFileManagerUI bietet Basis Dateisystemfunktionen wie aktualisieren, umbenennen, kopieren, ausschneiden, einfügen und so weiter. Es können Filter und Sortierreihenfolge eingestellt werden.

Der Inhalt der ausgewählten Datenquelle (Dateien / Ordner) werden in der Struktur MpFileManagerUIFileListType zusammengefasst. Die Liste mit den Speichermedien (FileDevices) findet man unter MpFileManagerUIDeviceListType. Die unterschiedlichen Speichermedien müssen hier manuell eingegeben werden.

Mit dem UIConnect.Status wird der jeweilige Zustand des **MpFileManagerUI** sichtbar. Im Ruhestand steht dieser auf mpFILE_UI_STATUS_IDLE. Während eines Befehls befindet sich der Status in entsprechendem Zustand. Dieser wird nach der Abarbeitung des Befehls wieder auf mpFILE_UI_STATUS_IDLE zurückgesetzt.

7.5 Diagnose

Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

7.5.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp File auftreten können.

Funktionsbaustein funktioniert nicht mit MpFileManager Konfiguration

mapp File kann auf zwei verschiedene Arten verwendet werden, mehr Information dazu kann hier gefunden werden.

Bei der Verwendung des Funktionsbausteins MpFileManagerUI oder MpFileManagerConfig kann nur die MpFileManagerUI Konfiguration verwendet werden.

Die MpFileManager Konfiguration kann nur in Kombination mit Widgets, welche Dateien bearbeiten oder abspeichern, wie das MotionPad-Widget, verwendet werden.

7.6 Bibliotheken

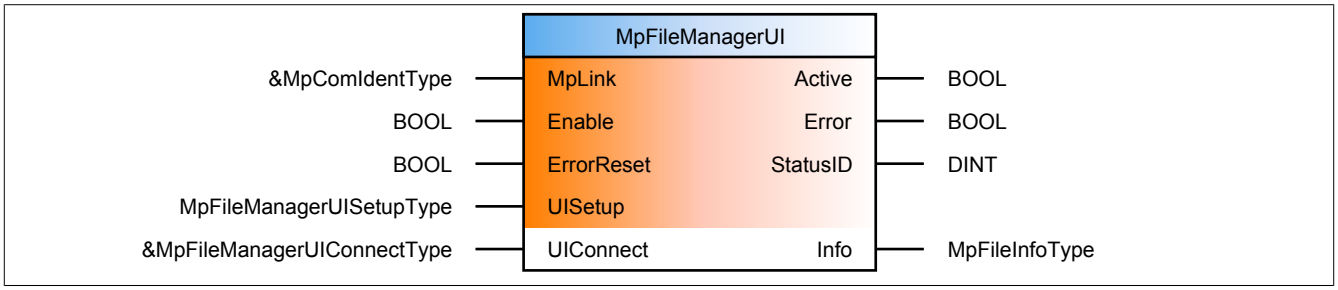
7.6.1 Funktionsbausteine

Unterstützt ab	Name	Beschreibung
5.0.0	MpFileManagerUI	Dieser Funktionsbaustein stellt ein Dateiverwaltungssystem zur Verfügung, sowie eine Verbindung zu Visual Components 4.
5.0.0	MpFileManagerConfig	Dieser Funktionsbaustein konfiguriert den Dateibrowser.

7.6.1.1 MpFileManagerUI

Dieser Funktionsbaustein stellt ein Dateiverwaltungssystem zur Verfügung, sowie eine Verbindung zu Visual Components 4 um die Dateien anzuzeigen.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpFileManagerUI Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	UISetup	`MpFileManagerUISetupType`	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
IN	UIConnect	`Zeiger auf MpFileManagerUIConnectType`	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpFileInfoType`	Zusätzliche Funktionsbausteins-Information.

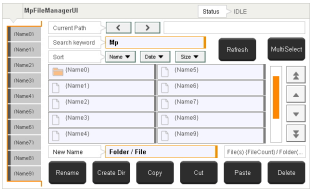
mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

7.6.1.1.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpFileManagerUI Konfiguration verwendet.

Es wird empfohlen am Zielsystem nur eine Instanz von MpFileManagerUI zu verwenden.



Initialisierung des Dateiverwaltungssystems

Vor dem Aktivieren des Funktionsblocks müssen die zwei Strukturen MpFileManagerUISetupType und MpFileManagerUIConnectType konfiguriert werden.

Beim Arbeiten auf Linux-Systemen kann auf keine Systemordner mit Zugangsbeschränkung zugegriffen werden. Dies kann zum Beispiel der "lost+found" Ordner sein. Wird versucht auf so einen Ordner zuzugreifen, wird der Fehler "Ungültiges File Device" ausgelöst. Betroffen sind alle Linux Geräte mit EXT2/3/4 Dateisystemen zb. MP71xx oder T30.

```
C:\Users\kratzern>dir \\10.43.37.3\usershare
 Volume in drive \\10.43.37.3\usershare is usershare
 Volume Serial Number is 9325-608F

 Directory of \\10.43.37.3\usershare

12/12/2017  07:42 PM    <DIR>          .
01/19/2018  02:01 PM    <DIR>          ..
12/12/2017  07:27 PM    <DIR>          TestDir
01/01/1970  02:00 AM    <DIR>          lost+found
               0 File(s)              0 bytes
               4 Dir(s)     473,661,440 bytes free
```

MpFileManagerUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

- **Status:** Der "Status" Parameter gibt Auskunft über die aktuelle Aktivität der Rezeptverwaltung
- **DeviceList:** Über die Struktur MpFileManagerUIDeviceListType wird das Speichermedium (File Device) ausgewählt, auf welches der Dateimanager zugreift. Diese Liste muss selbst verwaltet werden. Das bedeutet die Einträge in der Liste "DeviceNames" werden nicht vom Dateimanager erstellt, sondern es müssen gültige File Devices eingetragen werden
- **File:** Unter der Struktur MpFileManagerUIFileType ist die Liste mit den Dateien und Ordnern im aktuellen Pfad zu finden. Die Liste kann sortiert und gefiltert werden, um die Suche nach einer Datei / einem Ordner zu vereinfachen. Das Navigieren durch die Ordner und das Erstellen neuer Ordner ist auch über diese Struktur möglich. Weiter sind Standardfunktionen wie Kopieren, Ausschneiden, Einfügen, Löschen oder das Umbennen einer Datei / eines Ordners über diese Struktur möglich. Bei Angabe der Endung einer Datei oder des Filters muss nicht auf Groß- /Kleinschreibung geachtet werden. Beispielsweise kann eine Datei als Endung .txt .TxT oder .TXT besitzen, bei allen Varianten wird die Datei als *mpFILE_ITEM_TYPE_TXT* abgespeichert.
- **MessageBox:** In der Struktur MpFileManagerUIMessageBoxType können die unterschiedlichen Dialogfenster für MpFile konfiguriert werden
- **DefaultLayerStatus:** Solange eines der Dialogfenster aktiv ist, kann der dahinterliegende Layer über diesen Datenpunkt gesperrt werden

Suchfunktion

Mit dem Parameter "Filter" im Datentyp MpFileManagerUIFileType kann nach bestimmten Dateien gesucht werden. Es ist möglich in der Suchfunktion das Platzhaltersymbol "*" zu verwenden. Wird als Filter, zum Beispiel "*Test*.xml" verwendet, können die Dateien "Test01.xml", "MyTest.xml" oder "MyText01.xml" angezeigt werden. Bei Nichtbenützung des Platzhalters "*" wird direkt nach der Datei "Test.xml" gesucht.

Wird kein Filter angegeben oder nur das Symbol "*", werden alle Dateien und Ordner angezeigt die vorhanden sind.

MpFileManagerUISetupType

In dieser Struktur wird über "FileListSize" definiert, wieviele Einträge der Dateiliste auf einer Seite der Visualisierung angezeigt werden. Über "FileListScrollWindow" wird entscheiden wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden.

Fehlernummern

Tritt bei der Verwendung des Dateibrowsersystems ein Fehler auf, so wird der Fehler in der MessageBox-Struktur MpFileManagerUIMessageBoxType unter "ErrorNumber" angezeigt. Folgende Fehler können auftreten:

- 8192: Fehlender Wert an UIConnect
- 8193: Befehl nicht erlaubt
- 8194: Kein Element zum Einfügen
- 8195: Nichts ausgewählt
- 8196: Ordner existiert bereits

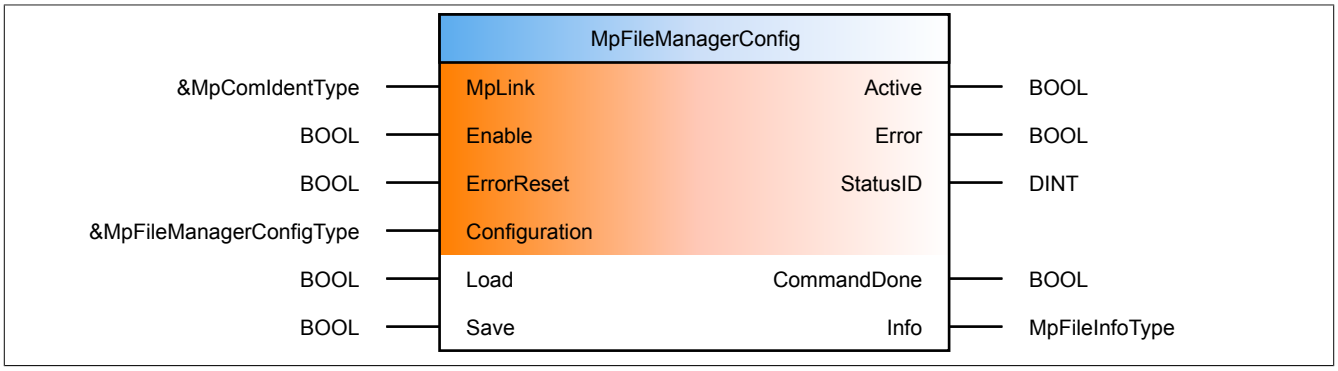
- 8197: Ungültiges File Device
- 8198: Kein neuer Name eingegeben
- 8199: Ungültiger Name
- 8200: Einfügen nicht erlaubt
- 8201: Allgemeiner Fehler

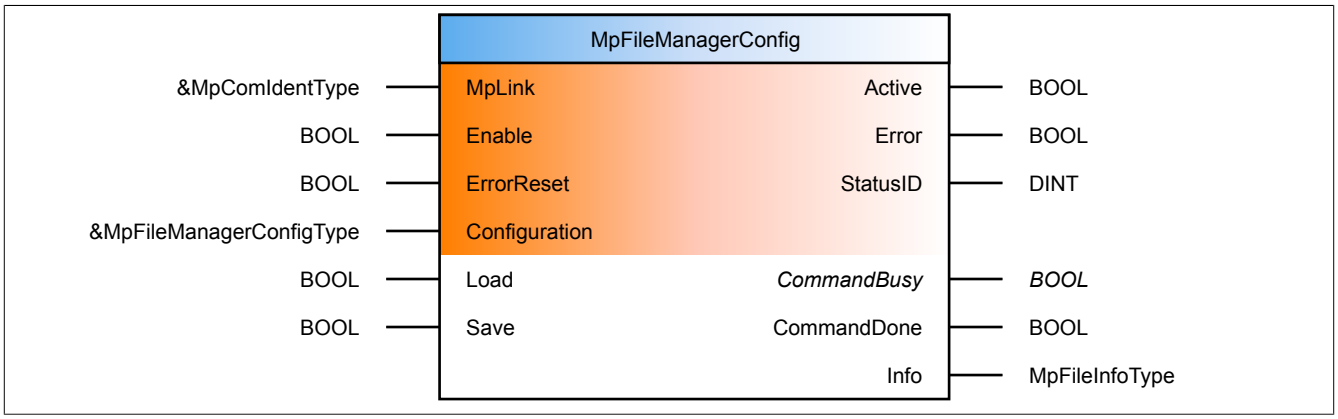
7.6.1.2 MpFileManagerConfig

Dieser Funktionsbaustein konfiguriert den Dateibrowser.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpFileManager Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Configuration	`Zeiger auf MpFileManagerConfigType`	Struktur, in der die Konfiguration festgelegt wird.
IN	Load	`BOOL`	Lädt die Konfiguration der Komponente.
IN	Save	`BOOL`	Speichert die Konfiguration der Komponente.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpFileInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

7.6.1.2.1 Beschreibung

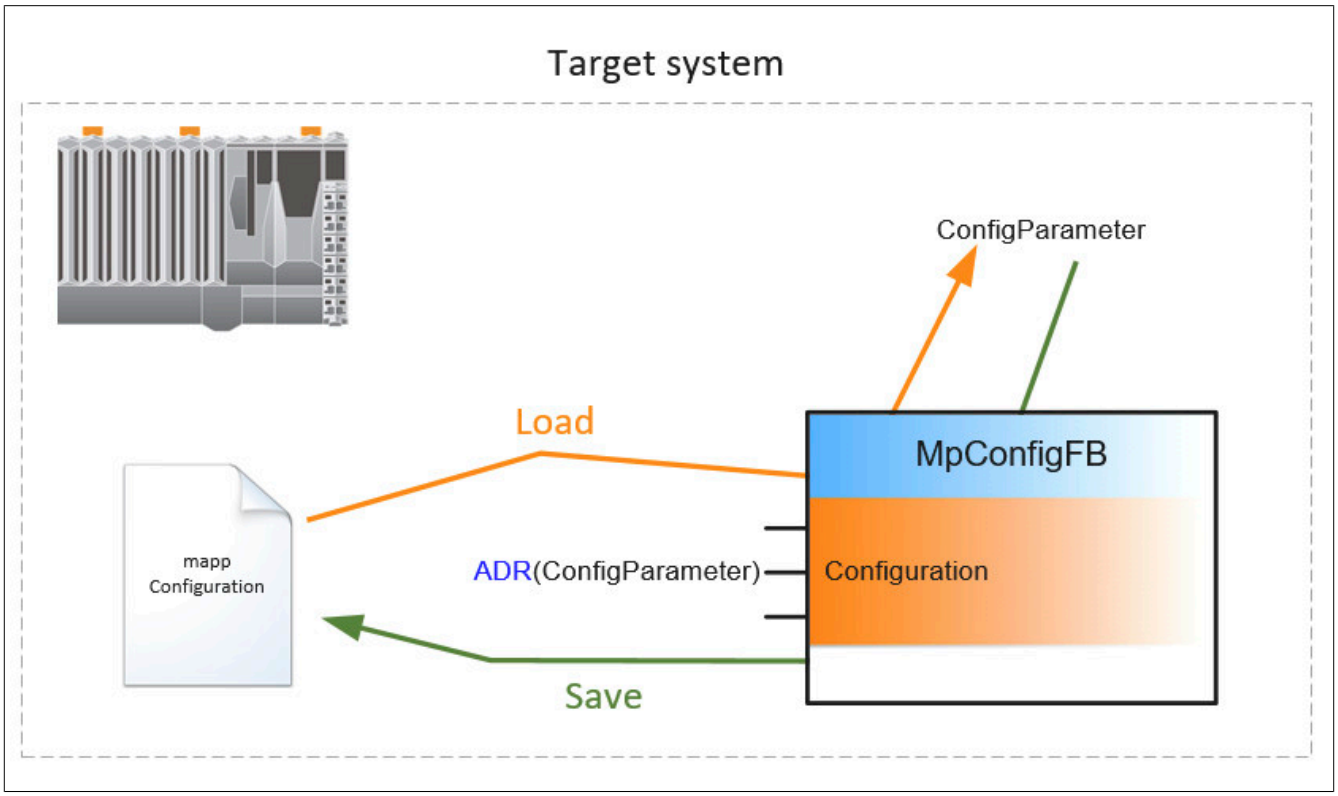
Für diese Komponente wird derselbe MpLink wie für MpFileManagerUI verwendet.

Mithilfe des Funktionsbausteins sind Änderungen der Konfiguration zur Laufzeit möglich.

Der Parameter "Configuration" ermöglicht es die Konfiguration, bzw. einen Teil der Konfiguration zu bearbeiten.

Der Befehl "Load" wird verwendet, um die aktuell konfigurierten Parameter auszulesen. Dabei wird die aktuell aktive Konfiguration der Steuerung in die PV am Eingang "Configuration" geschrieben. Die Daten in der Struktur werden dabei überschrieben. Danach können die Werte beliebig editiert werden.

Über den Befehl "Save" werden die Werte aus der "Configuration" Struktur auf die aktive Konfiguration übernommen und, wenn möglich, direkt verwendet.



Um die Konfigurations-Parameter in der Appliaktion korrekt anzupassen, wird folgendermaßen vorgegangen:

1. Befehl "Load = `TRUE`"

Die aktuelle Konfiguration wird vom Zielsystem in die Konfigurations-Struktur geladen. Der Vorgang ist mit "CommandDone = TRUE" abgeschlossen.

2. Befehl "Load = `FALSE`"

Rücksetzen des "Load" Befehls. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

3. Parametrieren der Konfigurationsstruktur

Die Parameter der Konfigurationsstruktur können nun wie gewünscht angepasst werden.

4. Befehl "Save = `TRUE`" setzen

Die geänderte Konfigurationsstruktur wird auf das Zielsystem übertragen. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

5. Befehl "Save = `FALSE`" setzen

Rücksetzen des "Save" Befehls. Der Vorgang ist mit "CommandDone = `FALSE`" abgeschlossen. Um zu kontrollieren ob die Parameter erfolgreich geändert worden sind, können die Schritte 1 und 2 erneut ausgeführt werden.

Konfigurieren der MpFileManagerConfigType Struktur

Über die Struktur MpFileManagerConfigType können Benutzerdefinierte Dateitypen während der Laufzeit hinzugefügt werden.

7.6.2 Datentypen und Enumeratoren

7.6.2.1 Datentypen

7.6.2.1.1 MpFileDiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	`MpFileStatusIDType`	StatusID Diagnose Struktur

7.6.2.1.2 MpFileInfoType

Mit diesem Datentyp wird für die MpFile Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	`MpFileDiagType`	Diagnosestruktur für die StatusID

7.6.2.1.3 MpFileManagerUIConnectType

Dieser Datentyp enthält alle notwendigen Informationen, um eine Verbindung zur Visualisierung herzustellen.

Parameter	Datentyp	Beschreibung
Status	`MpFileManagerUIStatusEnum`	Aktuelle Operation
DeviceList	`MpFileManagerUIDeviceListType`	Liste der Speichermedien (File Device) auf die der Dateimanager Zugriff hat
File	`MpFileManagerUIFileType`	Enthält Dateiliste, sowie zusätzliche Informationen über Dateien und Ordner und mögliche Befehle
MessageBox	`MpFileManagerUIMessageBoxType`	Bedienung der Dialogfenster
DefaultLayerStatus	`UINT`	Statusdatenpunkt für den Default Layer der Visualisierungsseite, auf der der Dateimanager angezeigt wird **VC4 Anbindung:** StatusDatapoint von *Layer*

7.6.2.1.4 MpFileManagerConfigType

Über diesen Datentypen wird der Dateibrowser konfiguriert. Die hier angegebenen Dateitypen können sofort verwendet werden.

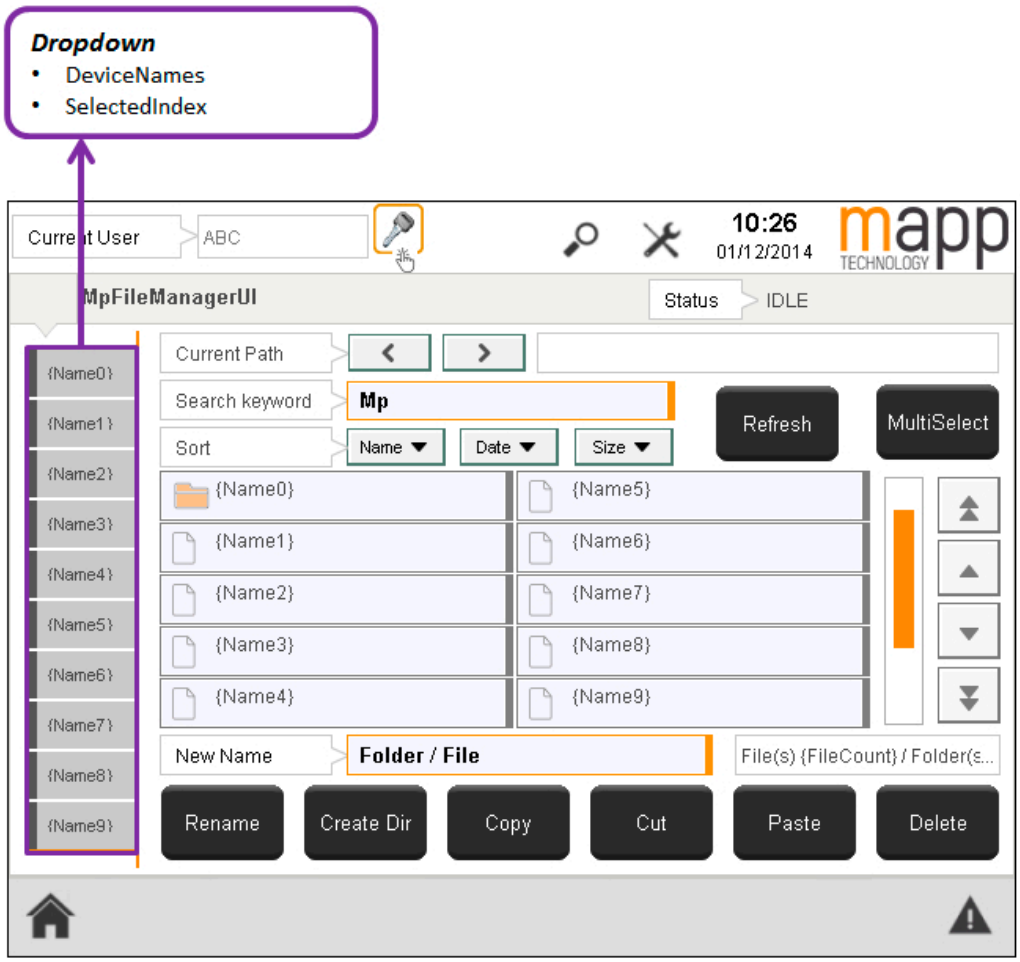
Wie das Konfigurationsupdate der einzelnen Parameter zur Laufzeit durchgeführt wird, ist im Abschnitt Konfiguration während der Laufzeit erklärt.

Parameter	Datentyp	Beschreibung	Konfigurationsupdate
CustomizedFileType	`ARRAY[0..9] von STRING[5]`	Benutzerdefinierter Dateityp	

7.6.2.1.5 MpFileManagerUIDeviceListType

Liste mit möglichen Speichermedien (File Device) und die dazugehörige Navigation.

Parameter	Datentyp	Beschreibung
DeviceNames	`ARRAY[0..9] von STRING[50]`	Liste aller verfügbaren Speichermedien **VC4 Anbindung:** TextDatapoint von *Dropdown*
SelectedIndex	`UINT`	Index des aktuell ausgewählten Eintrages in der Liste **VC4 Anbindung:** IndexDatapoint von *Dropdown*

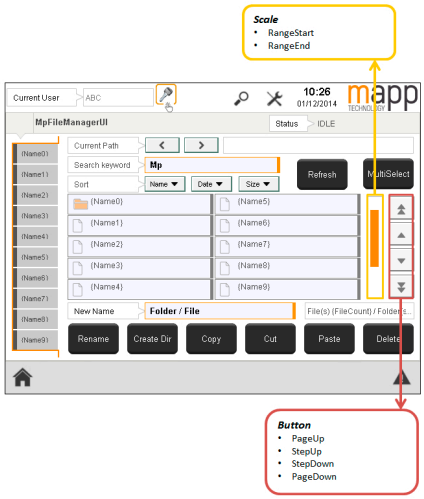


7.6.2.1.6 MpFileManagerUIFileListType

Liste mit möglichen Dateien/Ordnern und die dazugehörige Navigation.

Parameter	Datentyp	Beschreibung
Items	`ARRAY[0..49] von MpFileManagerUIItemType`	Liste aller Dateien und Ordner des aktuellen Pfades
PageUp	`BOOL`	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. Die Größe der Seite wird über den Parameter "FileListSize" der Struktur MpFileManagerUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
PageDown	`BOOL`	Springt ans Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. Die Größe der Seite wird über den Parameter "FileListSize" der Struktur MpFileManagerUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0

Parameter	Datentyp	Beschreibung
StepUp	BOOL	Wählt vorhergehenden Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepDown	BOOL	Wählt nächsten Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
RangeStart	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von *Scale*
RangeEnd	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von *Scale*

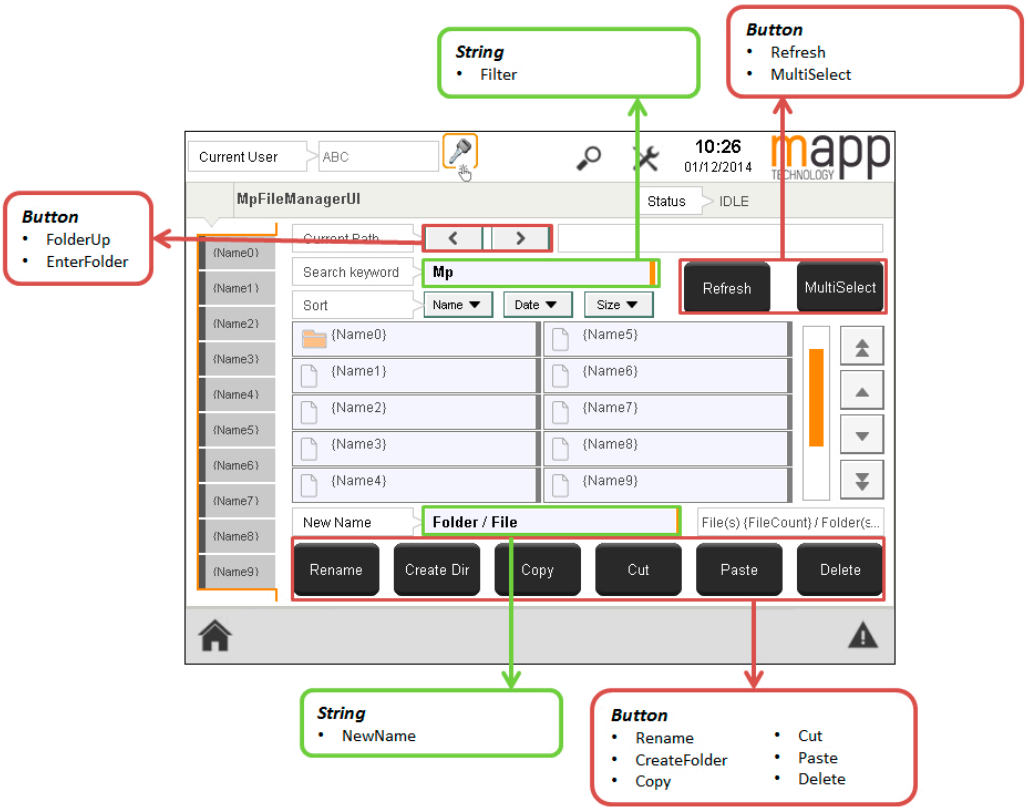


7.6.2.1.7 MpFileManagerUIFileType

Dieser Datentyp enthält alle Parameter, welche das Verwalten der Dateien betreffen.

Parameter	Datentyp	Beschreibung
List	MpFileManagerUIFileListType	Liste aller Ordner und Dateien des aktuellen Pfades, sowie die Navigation für die Liste
Filter	STRING[255]	Filter, mit dem Dateien aussortiert werden können. Es ist möglich in der Filterfunktion das Platzhaltersymbol "*" zu verwenden. Für mehr Information zur Filterfunktion siehe Abschnitt "Suchfunktion" in der Beschreibung von MpFileManagerUI **VC4 Anbindung:** Datapoint von *String*
SortOrder	MpFileManagerUISortOrderEnum	Verantwortlich für die Reihenfolge der Dateien in der Liste
Refresh	BOOL	Auffrischen der Dateiliste im aktuellen Ordner **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
PathInfo	MpFileManagerUIPathInfoType	Informationen über die Daten des aktuellen Pfades
EnterFolder	BOOL	Einen Ordner tiefer **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0

Parameter	Datentyp	Beschreibung
FolderUp	`BOOL`	Einen Ordner zurück **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
MultiSelect	`BOOL`	Erlaube das Auswählen mehrerer Elemente gemeinsam **VC4 Anbindung:** Datapoint von *Button* / Type: ToggleDatapoint / SetValue: 1 / ResetValue: 0
Delete	`BOOL`	Löschen einer Datei / eines Ordners **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
Copy	`BOOL`	Kopieren einer Datei / eines Ordners. Bei Kopieren in denselben Ordner, in dem auch die Quelle liegt, wird am Namen der kopierten Datei *(1), (2)* angehängt **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
Cut	`BOOL`	Ausschneiden einer Datei / eines Ordners **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
Paste	`BOOL`	Einfügen einer Datei / eines Ordners **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
NewName	`STRING[255]`	Der neue Name einer Datei / eines Ordners wenn "Rename" oder "CreateFolder" Befehl benutzt wird Die Datei kann auch in einem Ordner auf dem angegebenen Speichermdium abgelegt werden. Dabei muss der Ordnername mitangegeben werden, wie zum Beispiel "Ordnername/FileName" **VC4 Anbindung:** Datapoint von *String* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
Rename	`BOOL`	Umbenennen einer Datei / eines Ordners **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
CreateFolder	`BOOL`	Erstellen eines Ordners **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0



7.6.2.1.8 MpFileManagerUIItemType

Beschreibt eine Datei / einen Ordner genauer.

Parameter	Datentyp	Beschreibung
Name	`STRING[255]`	Dateiname / Ordnername **VC4 Anbindung:** TextIndexOffset von *Button* (Textgruppe für Dateimanager) - siehe Info
Size	`UDINT`	Dateigröße [Bytes] **VC4 Anbindung:** Datapoint von *Numeric*
LastModified	`DATE_AND_TIME`	Datum, an dem zuletzt mit dieser Datei / diesem Ordner gearbeitet wurde **VC4 Anbindung:** Datapoint von *DateTime*
ItemType	`MpFileManagerUIItemTypeEnum`	Dateityp (csv, jpg, ...) **VC4 Anbindung:** BitmapIndexDatapoint von *Button* (BitmapGroup für DateiFormate) - siehe Dateiformat anzeigen
IsSelected	`BOOL`	Datei / Ordner ist ausgewählt **VC4 Anbindung:** Datapoint von *Button* / Type: ToggleDatapoint / SetValue: 1 / ResetValue: 0
IsFolder	`BOOL`	Element ist ein Ordner

none

Info:

Für die Dateiauswahl werden Buttons mit ToggleDatapoint Aktion benutzt. Welchen Name eine Datei besitzt, wird über ein Textschnipsel in einer Textgruppe definiert. Die Größe der Textgruppe ergibt sich aus der Anzahl der Dateien, die auf einmal angezeigt werden sollen. Jedem Element in der Textgruppe ist ein Textschnipsel mit unterschiedlichem Index zugewiesen. (*Textschnipsel[0]* verbunden mit *FileManUIConnect.File.List.Items[0].Name*)

Name	Typ	Details
Name0	String	Datapoint(Name)
Name1	String	Datapoint(Name)
Name2	String	Datapoint(Name)
Name3	String	Datapoint(Name)
Name4	String	Datapoint(Name)
Name5	String	Datapoint(Name)

Abbildung 24: Textschnipsel

Index	English	German
0	{Name0}	{Name0}
1	{Name1}	{Name1}
2	{Name2}	{Name2}
3	{Name3}	{Name3}
4	{Name4}	{Name4}
5	{Name5}	{Name5}

Abbildung 25: Textgruppe Dateimanager

none

Dateiformat anzeigen:

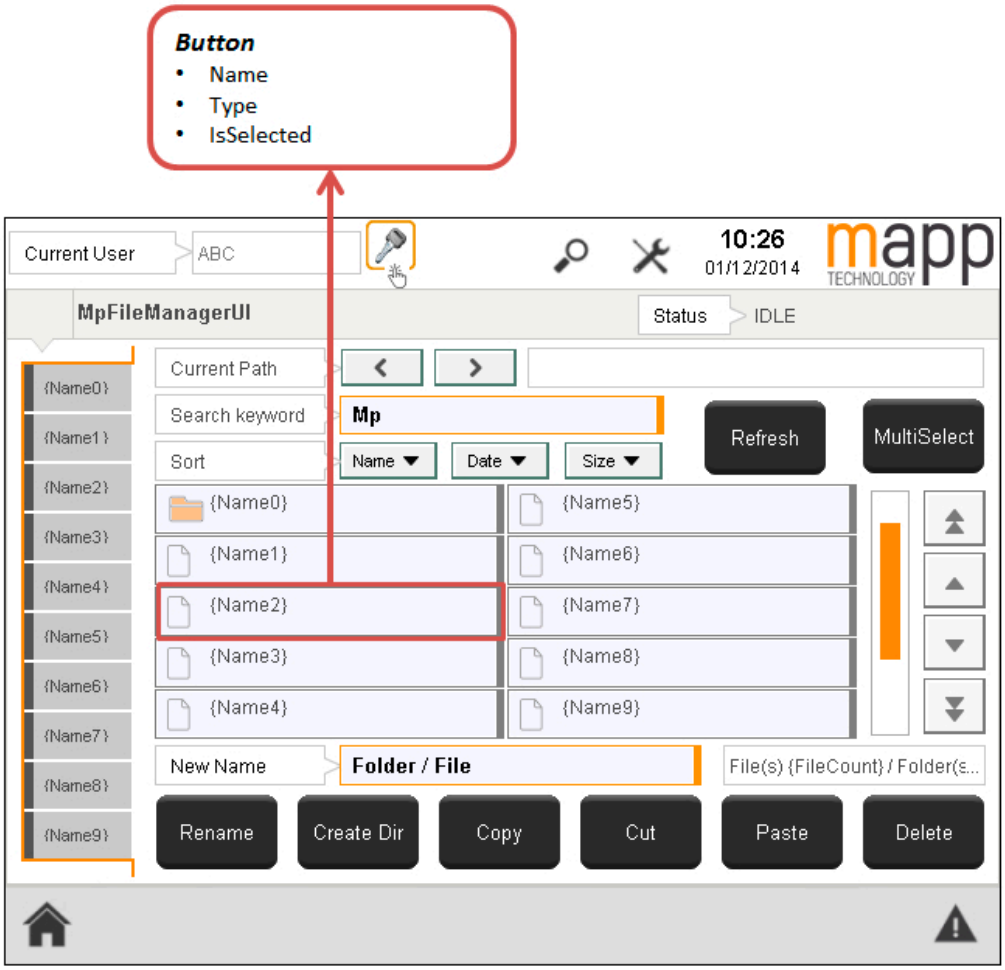
Um unterscheiden zu können, ob eine Datei ein Ordner oder eine Datei ist, wird der Parameter Type verwendet. Zusätzlich kann über diesen Enumerator festgestellt werden, um welches Dateiformat es sich handelt (CSV, XML, JPG, usw.).

Dazu wird eine BitmapGroup erstellt, welche für jedes Dateiformat ein Bild enthält. Welchen Index dieses Bild in der Bitmapgroup besitzt, ergibt sich aus dem Enumerator `MpFileManagerUIItemTypeEnum`.

Zuletzt wird diese BitmapGroup als Eigenschaft von dem Button angegeben (BitmapSource) und der Parameter Type als BitmapIndexOffset verbunden

Index	Bitmap
0	<None>
1	MpFile_FOLDER
2	MpFile_FILE
3	MpFile_TXT
4	MpFile_HTML
5	MpFile_DOC
6	MpFile_XLS
7	MpFile_XML
8	MpFile_CSV
9	MpFile_JPG
10	MpFile_BMP
99	<None>

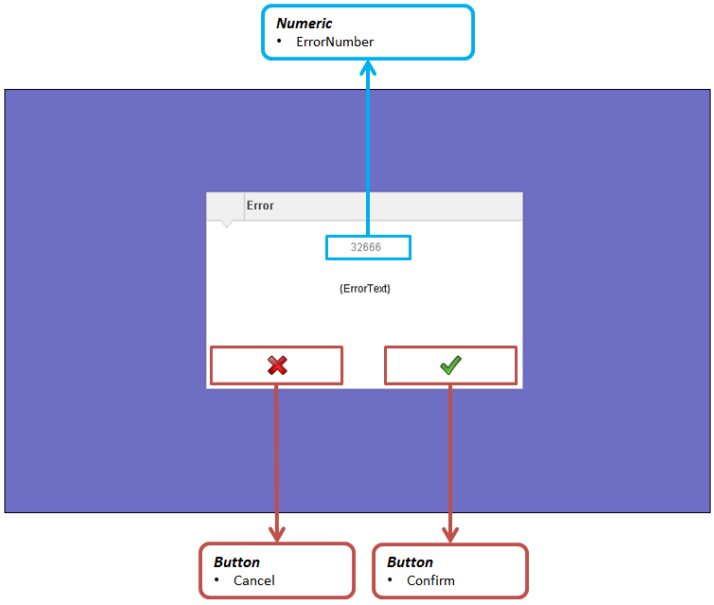
Abbildung 26: Bitmap-Group Dateiformate



7.6.2.1.9 MpFileManagerUIMessageBoxType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern.

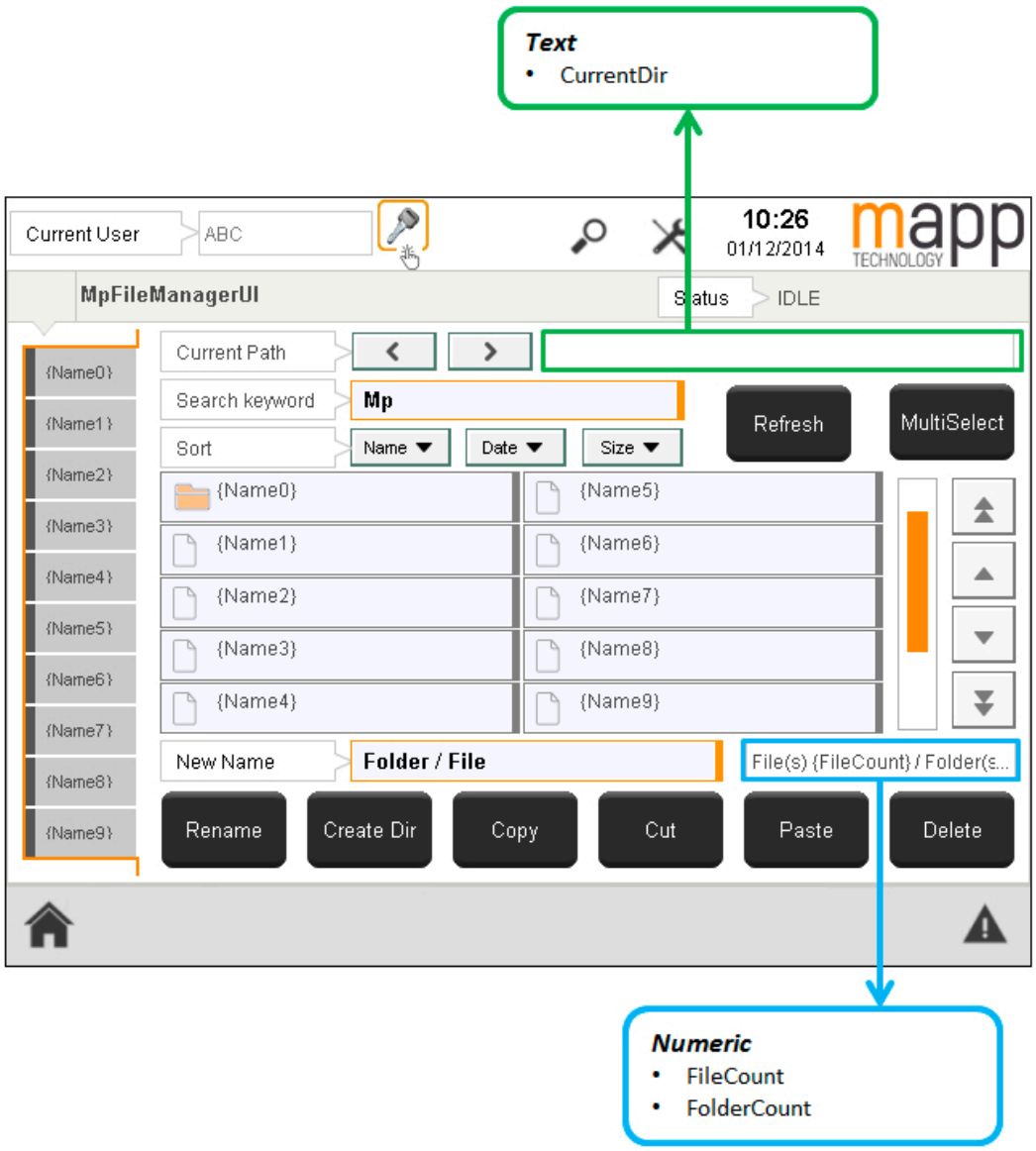
Parameter	Datentyp	Beschreibung
LayerStatus	`UINT`	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Type	`MpFileManagerUIMessageEnum`	Art des Dialogfensters
ErrorNumber	`UINT`	Aktuelle Fehlernummer zum Anzeigen. Eine detaillierte Beschreibung der Fehlernummern ist hier zu finden. **VC4 Anbindung:** Datapoint von *Numeric*
Confirm	`BOOL`	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	`BOOL`	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1



7.6.2.1.10 MpFileManagerUIPathInfoType

Zusätzliche Informationen, die auf der Visualisierung angezeigt werden können.

Parameter	Datentyp	Beschreibung
CurrentDir	`STRING[255]`	Aktueller Ordnername **VC4 Anbindung:** TextDatapoint von *Text*
FileCount	`UDINT`	Anzahl an Dateien im aktuellen Ordner **VC4 Anbindung:** Datapoint von *Numeric*
FolderCount	`UDINT`	Anzahl an Ordnern im aktuellen Ordner **VC4 Anbindung:** Datapoint von *Numeric*

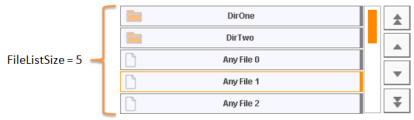


7.6.2.1.11 MpFileManagerUISetupType

Zusätzliche Konfigurationsmöglichkeiten, um die Visualisierung zu beschreiben.

Parameter	Datentyp	Wert	Beschreibung
FileListSize	`UINT`	10	Zahl an Dateien die auf einer Seite der Liste auf der Visualisierung dargestellt werden können Es können maximal 50 Dateien bzw. Ordner angezeigt werden
FileListScrollWindow	`USINT`	1	Zeigt an, wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden

FileListSize



FileListSize gibt an, wie viele Dateien auf einer Seite der VC4 Visualisierung angezeigt werden. Die Anzahl der Buttons auf der Visualisierung, die für das Anzeigen von Dateien verwendet werden, muss gleich der FileListSize sein. (siehe Bild oben)

FileListScrollWindow

FileListScrollWindow gibt an, wie viele Einträge beim Scrollen "vorrausschauend" angezeigt werden sollen. In diesem Beispiel wurde FileListScrollWindow = 1 definiert. Befindet sich nun die aktuelle Auswahl an 4. Stelle der Liste, so wird bei erneutem Drücken des StepDown Buttons der nächste Eintrag (Any File 2) ausgewählt und vorrauschschauend der nächste Eintrag (Any File 3) angezeigt.

7.6.2.1.12 MpFileStatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	`MpFileErrorEnum`	Fehlerkode der mapp Komponente
Severity	`MpComSeveritiesEnum`	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler)
Code	`UINT`	Kode der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden

7.6.2.2 Enumeratoren

7.6.2.2.1 MpFileManagerUIItemTypeEnum

Dieser Aufzählungsdatentyp beschreibt, welchen Typ eine Datei hat.

Aufzählung	Beschreibung
mpFILE_ITEM_TYPE_NONE	Keine Typ Definition
mpFILE_ITEM_TYPE_FOLDER	Das Element ist ein Ordner
mpFILE_ITEM_TYPE_FILE	Das Element ist eine Datei
mpFILE_ITEM_TYPE_TXT	Das Element is eine Textdatei
mpFILE_ITEM_TYPE_HTML	Das Element is eine HTML Datei
mpFILE_ITEM_TYPE_DOC	Das Element ist eine Word Datei (.doc)
mpFILE_ITEM_TYPE_XLS	Das Element ist eine Excel Datei (.xls)
mpFILE_ITEM_TYPE_XML	Das Element ist eine XML Datei (.xml)
mpFILE_ITEM_TYPE_CSV	Das Element ist eine CSV Datei (.csv)
mpFILE_ITEM_TYPE_JPG	Das Element ist eine Bild Datei (.jpg)
mpFILE_ITEM_TYPE_BMP	Das Element ist eine Bild Datei (.bmp)
mpFILE_ITEM_TYPE_USER1	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER2	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER3	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER4	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER5	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER6	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER7	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER8	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER9	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_USER10	Das Element ist eine benutzedefinierte Datei (kann über die Konfiguration festgelegt werden)
mpFILE_ITEM_TYPE_UNKNOWN	Der Dateityp ist unbekannt

7.6.2.2.2 MpFileManagerUIMessageEnum

Dieser Aufzählungsdatentyp entscheidet über die Art des Dialogfensters.

Aufzählung	Beschreibung
mpFILE_MSG_OK	Dialogfenster, dass zur Bestätigung der Eingabe auffordert
mpFILE_MSG_CONFIRM_DELETE	Dialogfenster, dass eine Bestätigung beim Löschen fordert
mpFILE_MSG_CONFIRM_OVERWRITE	Dialogfenster, dass eine Bestätigung beim Überschreiben fordert
mpFILE_MSG_BUSY	Dialogfenster, dass eine anzeigt, dass aktuell eine andere Aktion ausgeführt wird

7.6.2.2.3 MpFileManagerUISortOrderEnum

Dieser Aufzählungsdatentyp ist eine Vorlage für alle weiteren mapp Komponenten Aufzählungsdatentypen.

Aufzählung	Beschreibung
mpFILE_SORT_BY_NAME_ASC	Sortiere in ansteigender (alphabetischer) Reihenfolge
mpFILE_SORT_BY_NAME_DESC	Sortiere in absteigender (alphabetischer) Reihenfolge
mpFILE_SORT_BY_SIZE_ASC	Sortiere in ansteigender Reihenfolge über die Dateigröße
mpFILE_SORT_BY_SIZE_DES	Sortiere in absteigender Reihenfolge über die Dateigröße
mpFILE_SORT_BY_MOD_TIME_ASC	Sortiere in ansteigender Reihenfolge über den Zugriffszeitpunkt
mpFILE_SORT_BY_MOD_TIME_DESC	Sortiere in absteigender Reihenfolge über den Zugriffszeitpunkt

7.6.2.2.4 MpFileManagerUIStatusEnum

Dieser Aufzählungsdatentyp gibt Auskunft über die aktuelle Aktivität der Komponente.

Aufzählung	Beschreibung
mpFILE_UI_STATUS_IDLE	Aktuell ist kein Prozess aktiv
mpFILE_UI_STATUS_REFRESH	Aktualisieren der Elemente
mpFILE_UI_STATUS_CHANGE_PAGE	Seitenwechsel aktiv
mpFILE_UI_STATUS_CHANGE_DIR	Ordnerwechsel aktiv
mpFILE_UI_STATUS_CHANGE_DEVICE	Speichermedium (FileDevice) wird gewechselt
mpFILE_UI_STATUS_CREATE	Erstellen einer Datei / eines Ordners aktiv
mpFILE_UI_STATUS_RENAME	Umbenennen einer Datei / eines Ordners aktiv
mpFILE_UI_STATUS_DELETE	Löschen einer Datei / eines Ordners aktiv
mpFILE_UI_STATUS_COPY	Kopieren einer Datei / eines Ordners aktiv
mpFILE_UI_STATUS_CUT	Ausschneiden einer Datei / eines Ordners aktiv
mpFILE_UI_STATUS_PASTE	Einfügen einer Datei / eines Ordners aktiv
mpFILE_UI_STATUS_ERROR	Fehler aktiv

7.6.3 Statusnummern

7.6.3.1 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

Konstante:

`mpFILE_ERR_ACTIVATION`

7.6.3.2 -1064329102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

Konstante:

mpFILE_ERR_MPLINK_NULL

7.6.3.3 -1064329101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

Konstante:

`mpFILE_ERR_MPLINK_INVALID`

7.6.3.4 -1064329100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

Konstante:

`mpFILE_ERR_MPLINK_CHANGED`

7.6.3.5 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

Konstante:

`mpFILE_ERR_MPLINK_CORRUPT`

7.6.3.6 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

Konstante:

`mpFILE_ERR_MPLINK_IN_USE`

7.6.3.7 -1064239091: Ungültige Konfiguration

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Konfiguration ist beschädigt
- Die Funktion ist in der Konfiguration nicht aktiviert

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

Konstante:

`mpFILE_ERR_CONFIG_INVALID`

7.6.3.8 -1064165376: Fehlender Wert an UIConnect

Beschreibung:

An "UIConnect" wurde NULL angehängt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Eingang "UIConnect" vergessen

Konstante:

`mpFILE_ERR_MISSING_UICONNECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.9 -1064165375: Befehl nicht erlaubt

Beschreibung:

Im Multiselect Modus ist dieser Befehl nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Multiselect ausschalten
- Befehl wechseln

Konstante:

`mpFILE_ERR_CMD_NOT_ALLOWED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.10 -1064165374: Kein Element zum Einfügen

Beschreibung:

Es wurde kein Element ausgewählt, das eingefügt werden könnte.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Element kopieren oder ausschneiden, bevor der "Paste" Befehl ausgelöst wird

Konstante:

`mpFILE_ERR_NOTHING_TO_PASTE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.11 -1064165373: Nichts ausgewählt

Beschreibung:

Es wurde nichts ausgewählt für eine Operation.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Zuerst Datei oder Ordner auswählen

Konstante:

`mpFILE_ERR_NOTHING_SELECTED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.12 -1064165372: Ordner existiert bereits

Beschreibung:

Beim Erstellen oder Umbenennen eines Ordners ist ein Fehler aufgetreten, da der neue Name bereits in Verwendung ist.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Der neue Name ("NewName" in MpFileManagerUIFileType) existiert bereits in diesem Pfad

Konstante:

`mpFILE_ERR_DIR_ALREADY_EXISTS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.13 -1064165371: Ungültiges File Device

Beschreibung:

Das gewählte Speichermedium (File Device) ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Verbindung zum Speichermedium (FileDevice) überprüfen ("DeviceList" und "DeviceIndex" in der MpFileManagerUIConnectType Struktur)
- Beim Arbeiten auf Linux-Systemen kann auf keine Systemordner mit Zugangsbeschränkung zugegriffen werden. Dies kann zum Beispiel der "lost+found" Ordner sein.

```
C:\Users\kratzern>dir \\10.43.37.3\usershare
 Volume in drive \\10.43.37.3\usershare is usershare
 Volume Serial Number is 9325-608F

 Directory of \\10.43.37.3\usershare

12/12/2017  07:42 PM    <DIR>          .
01/19/2018  02:01 PM    <DIR>          ..
12/12/2017  07:27 PM    <DIR>          TestDir
01/01/1970  02:00 AM    <DIR>          lost+found
               0 File(s)              0 bytes
               4 Dir(s)     473,661,440 bytes free
```

- Es können nur Dateien/Ordner verwendet werden, welche ASCII-Zeichen enthalten. Andere Zeichen für Dateien/Ordner werden nicht unterstützt.

Konstante:

`mpFILE_ERR_INVALID_FILE_DEV`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.14 -1064165370: Kein neuer Name eingegeben

Beschreibung:

Beim Erstellen oder Umbenennen einer Datei / eines Ordners wurde kein neuer Name gefunden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Der neue Name ("NewName" in MpFileManagerUIFileType) ist leer

Konstante:

`mpFILE_ERR_NAME_EMPTY`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.15 -1064165369: Ungültiger Name

Beschreibung:

Der Name der Datei oder des Ordners ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Folgende Sonderzeichen sind im Namen nicht erlaubt: \ / : * ? " < > |
- Datei mit gewähltem Namen existiert bereits

Konstante:

`mpFILE_ERR_INVALID_NAME`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.16 -1064165368: Einfügen nicht erlaubt

Beschreibung:

In diesem Pfad ist das Einfügen nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Einfügen in nicht erlaubtem Pfad (gesperrt, existiert bereits, ...)
- Pfad überprüfen

Konstante:

`mpFILE_ERR_PASTE_NOT_ALLOWED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.17 -1064165367: Allgemeiner Fehler

Beschreibung:

Allgemeiner Fehler im Filesystem. Für mehr Information siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Siehe Logger für mehr Informationen

Konstante:

`mpFILE_ERR_FILE_SYSTEM`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpFileManagerUI

7.6.3.18 -1064165358: Keine Rechte

Beschreibung:

Der Benutzer {2: UserName} hat nicht die Rechte um auf den Pfad {3: Path} zuzugreifen

Ursache / Abhilfe:

- Überprüfen ob in der MpFileManager Konfigurationdas Recht vergeben wurde um auf einen Pfad zuzugreifen

7.6.3.19 -1064165357: Kein Zugriff auf gesperrte Datei

Beschreibung:

Auf eine gesperrte Datei kann nicht zugegriffen werden

Ursache / Abhilfe:

- Datei wird von anderen Programm verwendet
- Warten bis das Datei nicht mehr gesperrt ist

7.6.3.20 -1064165356: Datei kann nicht kopiert werden

Beschreibung:

Die Datei im Pfad {3:SourcePath} kann nicht nach {4:DestPath} kopiert werden. Fehlerursache: {2:ErrorNumber}

Zusätzliche Information

- {2:ErrorNumber}: Fehlerursache
- {3:SourcePath}: Quellpfad
- {4:DestPath}: Zielpfad

Ursache / Abhilfe:

- Prüfen ob Dateien bzw. Ziel-Pfad noch vorhanden ist

7.6.3.21 -1064165355: Datei kann nicht gelöscht werden

Beschreibung:

Die Datei im Pfad {3:Path} kann nicht gelöscht werden. Fehlerursache {2:ErrorNumber}

Zusätzliche Information

- {3:Path}: Pfad
- {2:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Prüfen ob Datei noch vorhanden oder defekt ist

7.6.3.22 -1064165354: Ungültiger Pfad

Beschreibung:

Es wurde ein ungültiger Pfad {2:PathName} angegeben

Zusätzliche Information

- {2:PathName}: Pfad

Ursache / Abhilfe:

- Angegebenen Pfad überprüfen

7.6.3.23 -1064165353: Ungültiger Parameter

Beschreibung:

Es wurde ein ungültiger Parameter angegeben

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

7.6.3.24 -1064165352: Datei kann nicht umbenannt werden

Beschreibung:

Datei im Pfad {3: Path} konnte nicht umbenannt werden zu {4:NewName}. Fehlerursache: {2: ErrorNumber}

Zusätzliche Information

- {3:Path}: Pfad
- {2:ErrorNumber}: Fehlerursache
- {4:NewName}: Neuer Dateiname

Ursache / Abhilfe:

- Überprüfen ob Dateien und Pfade vorhanden sind

7.6.3.25 -1064165351: Datei kann nicht erstellt werden

Beschreibung:

Die Datei in {3: Path} konnte nicht erstellt werden. Fehlerursache: {2: ErrorNumber}

Zusätzliche Information

- {3:Path}: Pfad
- {2:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen ob Dateien und Pfade vorhanden sind

7.6.3.26 -1064165350: Datei kann nicht gelesen werden

Beschreibung:

Die Datei in {3: Path} konnte nicht gelesen werden. Fehlerursache: {2: ErrorNumber}

Zusätzliche Information

- {3:Path}: Pfad
- {2:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen ob Dateien und Pfade vorhanden sind

7.6.3.27 -1064165349: Datei kann nicht bearbeitet werden

Beschreibung:

Die Datei in {3: Path} kann nicht bearbeitet werden. Fehlerursache: {2: ErrorNumber}

Zusätzliche Information

- {3:Path}: Pfad
- {2:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen ob die benötigten Rechte in der MpFileManager Konfiguration vergeben wurden

7.6.3.28 1083318300: Datei wurde überschrieben

Beschreibung:

Die Datei wurde überschrieben

7.6.3.29 -1064285140: Auschneiden fehlgeschlagen

Beschreibung:

Das Ausschneiden und Einfügen ist fehlgeschlagen. Der Ordner bzw. die Datei existiert bereits.

Ursache / Abhilfe:

- Auswählen eines anderen Ordners bzw. Datei
- Name des Ordners bzw. der Datei ändern

7.6.3.30 -1064285139: Kopieren fehlgeschlagen

Beschreibung:

Das Kopieren und Einfügen ist fehlgeschlagen. Der Ordner bzw. die Datei existiert bereits.

Ursache / Abhilfe:

- Auswählen eines anderen Ordners bzw. Datei
- Name des Ordners bzw. der Datei ändern

7.6.4 Alarme

7.6.4.1 mpFILE_ALM_DEVICE_NOT_FOUND: FileDevice wurde nicht gefunden

Beschreibung:

Aktion fehlgeschlagen, da das FileDevice (Speichermedium) nicht gefunden wurde.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Parameter am Eingang "DeviceList" im Datentyp überprüfen
- Wert am Eingang "DeviceIndex" möglicherweise nicht gültig

Verhalten

Edge Alarm

7.6.4.2 mpFILE_ALM_ALREADY_EXISTING: Datei/Ordner existiert bereits

Beschreibung:

Die Aktion ist fehlgeschlagen, da der Ordner/die Datei bereits existiert.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Ordner/Datei löschen oder umbenennen

Verhalten

Edge Alarm

7.6.4.3 mpFILE_ALM_ERROR_FILE_SYSTEM: Genereller Fehler des Dateisystems

Beschreibung:

Genereller Fehler des Dateisystems. Details sind in den zusätzlichen Informationen am Ausgang des Funktionsblocks zu finden.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Genaue Ursache in den zusätzlichen Infos zu finden (Info Struktur - "Code" Parameter)

Verhalten

Edge Alarm

8 mapp User: Benutzerverwaltung

mapp User dient zum Einrichten eines Benutzermanagements. Die Benutzer werden in Gruppen mit unterschiedlichen Berechtigungen unterteilt und können sich über Benutzernamen und Passwort einloggen.

MpUser liefert viele Funktionen, welche das Umsetzen der Bestimmung 21 CFR Teil 11 erleichtern (z.B. werden Passwörter automatisch nach einem definierten Zeitraum ungültig).

8.1 Konzept

Allgemeines

- Als Applikationist möchte ich mich nicht um Basisfunktionen wie das Ein- und Ausloggen, sowie die Darstellung über eine Visualisierung kümmern müssen
- Als OEM möchte ich, dass meine Maschine die Regulierungen der FDA erfüllt, dabei werde ich von von MpUser unterstützt
- Als Applikationist möchte ich meine Benutzer, sowie Benutzergruppen einfach konfigurieren können
- Als Applikationist möchte ich System Einstellungen für meine Benutzermanagement vornehmen können (z.B. die Definition wie ein Passwort auszusehen hat, nach welchem Zeitraum ein Benutzer automatisch ausgeloggt wird

Export-, Import-Funktion

- Als Endnutzer möchte ich meine aktuellen Benutzer und Rollen an der bestehenden Maschine exportieren um diese in anderen Maschinen wieder zu verwenden können

Benutzermanagement visualisieren

- Als Endnutzer möchte ich meine eigenen Benutzer anlegen können und ihnen bestimmte Rechte zuweisen
- Als Endnutzer möchte ich Zugriff durch Dritte auf meine Maschine verhindern. Nur Berechtigte besitzen die Möglichkeit meine Maschine zu bedienen
- Als Endnutzer möchte ich Mitarbeiter, welche aus meiner Firma ausgeschieden sind, sperren bzw. löschen können
- Als OEM möchte ich einen eigenen Zugang zu meinen Maschinen, um bestimmte Maschinendaten abzufragen bzw. zu verändern
- Als Maschinenbediener möchte ich mich schnell ein- und ausloggen können
- Als Maschinenbediener möchte ich verhindern, dass jemand ohne mein Wissen Maschinenparameter verändert
- Als Schichtleiter möchte ich meine Rezepte erstellen und verändern können. Ich möchte aber nicht, dass meine Maschinenbediener bzw. Servicetechniker Rezepte verändern können
- Als Servicetechniker möchte ich einen eigenen Zugang zu meinen Maschinen, um meine Maschine bei der Wartung manuell zu steuern

8.1.1 Konfiguration während der Laufzeit

Die Konfigurations Parameter können zur Laufzeit über MpUserConfig oder MpUserLoginConfig verändert werden. Dabei ist für die unterschiedlichen Parameter jedoch zu beachten in welchem Zustand sich die Komponente befinden muss um eine Änderung zuzulassen. Folgende Zustände sind möglich:

- : Dieses Symbol bedeutet, dass dieser Parameter in jedem Zustand geändert werden kann

Jeder Parameter der Konfigurations Struktur (MpUserConfigType, MpUserLoginConfigType) ist mit einem dieser Symbole ausgestattet.

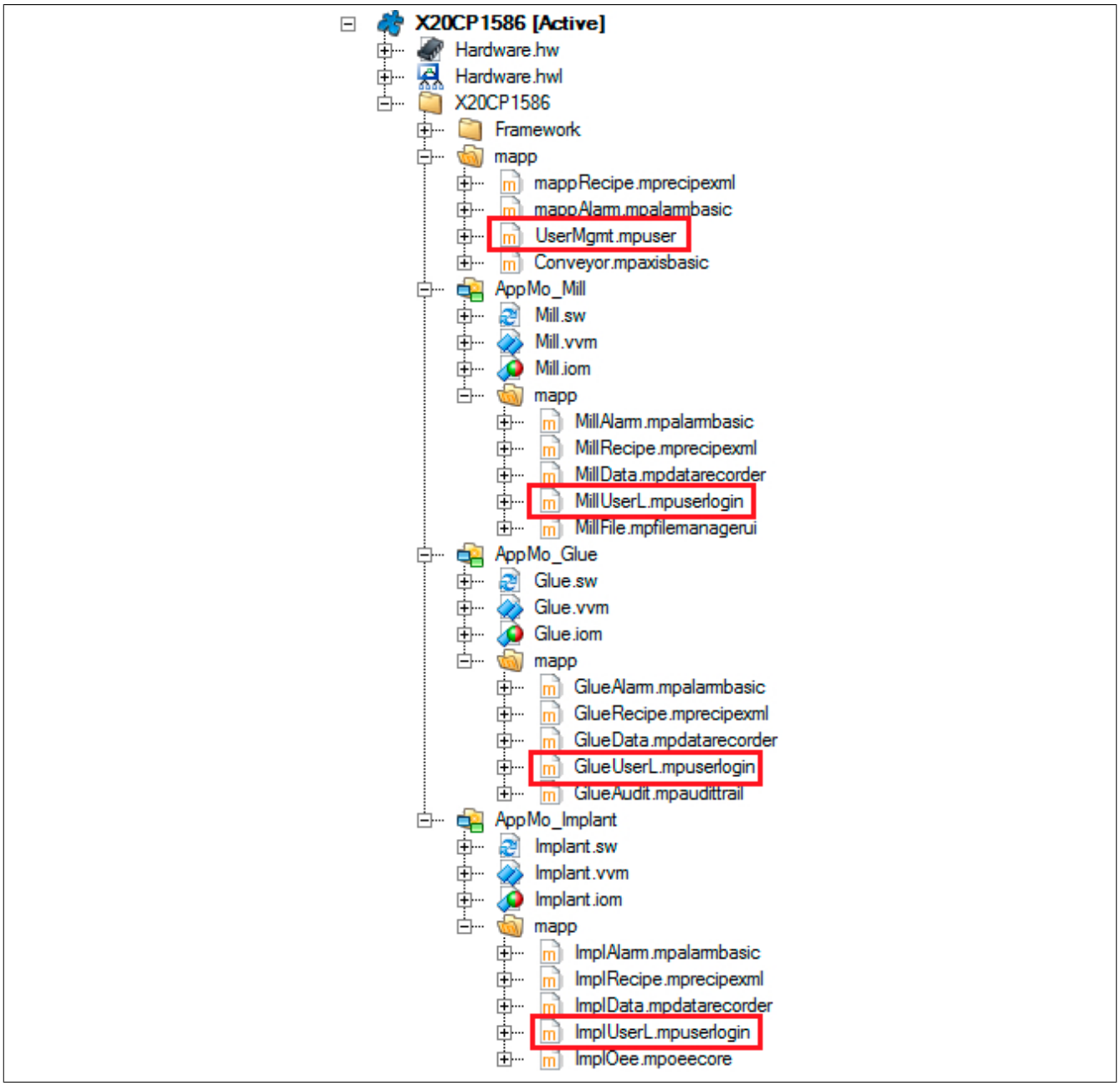
8.1.2 MpUser in einer modularen Applikation

In diesem Abschnitt wird erklärt was man bei der Implementierung von MpUser in modularen Applikationen beachten muss. Allgemeine Infos zu mapp in Applikationsmodulen / einer Maschinengruppe sind Bilden von modularen Applikationen

Verhalten von MpUser in einer modularen Applikation

Die Konfiguration MpUser gilt für die komplette Maschine und sollte daher nicht einer speziellen Maschinengruppe bzw. einem speziellen Applikationsmodul zugewiesen sein. Bei Verwendung von Applikationsmodulen sollte die MpUser Konfiguration im Framework Projekt liegen.

Die MpUserLogin Konfiguration kann mehrfach eingefügt werden. Die Konfiguration beschreibt einen Loginpunkt in der Maschine und kann in den verschiedenen Applikationsmodulen / Maschinengruppen eingefügt werden. Über jeden Loginpunkt kann sich ein Benutzer anmelden und arbeiten. Pro Applikationseinheit / Maschinengruppe ist nur ein Loginpunkt erlaubt.



8.1.3 Rechte, Levels und Funktionen

In diesem Kapitel werden Begriffe erläutert, welche im Benutzermanagement dieser mapp Komponente verwendet werden.

Administrator

Um andere Benutzer zu verwalten (hinzufügen, sperren, löschen, editieren) muss man Administrator des Benutzermanagements sein. Ob ein Benutzer Administrator ist, wird über die Benutzergruppe festgelegt. Welche Rechte der Administrator hat, wird in der MpUser Konfiguration festgelegt.

Levels

Das Level gibt Auskunft über die Zugriffsberechtigungen auf die Maschine. Je höher das Level, desto mehr Berechtigungen besitzt ein Benutzer. Das Level ist in der Benutzergruppe festgelegt. Jeder Benutzer aus einer Gruppe besitzt dasselbe Level. Wenn gewünscht, können die veschiedenen Funktionen der Maschine und der Visualisierung direkt über das Level gesperrt oder freigegeben werden. Folgende Tabelle dient als Beispiel:

Benutzergruppe	Level	Adminrechte	Kommentar
Maschinenbesitzer	100	ja	Kann die Maschine ohne Einschränkungen bedienen und auch die Benutzer verwalten
Prozesstechniker	70	nein	Besitzt hohe Zugriffsrechte, kann jedoch keine Benutzer verwalten, da er keine Adminrechte besitzt
Schichtführer	50	ja	Der Schichtführer kann neue Benutzer anlegen oder editieren. Jedoch nur, falls diese ein niedrigeres Level besitzen
Maschinenbediener	30	nein	Arbeitet täglich mit der Maschine
Wartungstechniker	30	nein	Bedient die Maschine während der Wartung
Besucher	3	nein	Gruppe, die lesenden Zugriff auf die Visualisierung hat, jedoch keine weiteren Funktionen nützen kann

Falls es mehrere Gruppen auf demselben Level gibt, können sich diese durch verschiedene Rechte unterscheiden.

Funktionen und Rechte

Die Maschine, bei der das Benutzermanagement implementiert wird, besitzt unterschiedliche Funktionen. Dazu können Rezeptverwaltung, Prozesssteuerung, Alarmverwaltung und/oder das Ändern der Prozessparameter gehören. Auf diese (und alle anderen) Funktionen können unterschiedliche Rechte vergeben werden:

- **Keine Rechte:** Der Benutzer hat für diese Funktion keine Rechte
- **Lesezugriff:** Ber Benutzer kann Daten einsehen, aber nicht bearbeiten
- **Start- /Stoppzugriff:** Der Benutzer kann eine Funktion Starten und Stoppen, jedoch keinen weiteren Einfluss auf den Prozess nehmen (z.B. Ändern der Prozessparameter)
- **Voller Zugriff:** Der Benutzer hat Lese- und Schreibrechte für eine bestimmte Funktion
- **Undefiniert:** Falls kein spezielles Recht definiert wurde. Der Benutzer hat auch hier keine Rechte für die unterschiedlichen Funktionen

Auf jede Funktion kommt eines dieser Rechte. Die Funktionen können festgelegt werden. Beim obigen Beispiel gibt es zwei Gruppen mit demselben Level. Diese unterscheiden sich aber durch ihre Rechte auf unterschiedliche Funktionen:

Funktion	Maschinenbediener	Wartungstechniker
Automatische Prozesssteuerung	Start-/Stopzugriff	Keine Rechte
Rezeptverwaltung	Keine Rechte	Keine Rechte
Öffnen der Sicherheitstüren	Keine Rechte	Voller Zugriff
Manuelle Prozesssteuerung	Start-/Stopzugriff	Voller Zugriff

Die Funktionen sind Teil der Maschinensoftware und werden nicht über mapp definiert. In der Konfiguration kann beschrieben werden, welches Recht für welche Maschinenfunktion verwendet wird:

Token

Zugriff auf den Benutzeraccount kann auch über Token gegeben werden. Token können sein:

- RFID Chip
- USB Stick

Die Identifikationsnummer wird intern verwaltet. Es kann aber für jeden Token ein Name vergeben werden.

Das Einloggen über einen Token ist noch nicht implementiert.

Sperren eines Benutzers

Das Recht einen Benutzer zu sperren bzw. zu entsperren hat nur der Admin. Wichtig hierbei ist, dass der Admin ein höheres Benutzerlevel hat als der Benutzer.

Andere Benutzer sowie der Admin werden automatisch gesperrt, wenn beim Einloggen das Passwort zu oft falsch eingegeben wurde. Wie oft das Passwort eingegeben werden kann, wird in der Konfiguration über den Parameter "Login attempts" festgelegt.

Der Admin wird automatisch nach einer Stunden wieder entsperrt. Diese Zeit kann auch über die Konfiguration über den Parameter "Admin unlock time" geändert werden. Wird diese Zeit auf 0 gesetzt, so wird der Admin nicht automatisch entsperrt.

Um den Benutzer in dem Fall wieder zu entsperren, kann der Admin das Kommando "Lock = `TRUE`" an der MpUserMgrUIUserType Struktur starten. Über die Dialogbox-Stuktur von MpUserMgrUIConnectType ("MessageBox") kann der Entsperr-Vorgang anschließend bestätigt ("Confirm = `TRUE`") werden.

Ein Benutzer wird auch gesperrt, wenn dessen Account über längere Zeit nicht benutzt wurde.
Ist ein Benutzer gesperrt, so kann nur der Admin diesen Benutzer entsperren. Diese Zeit wird in der Konfiguration über den Parameter "User expiration time" definiert. Bei 0 wird diese Funktion nicht genutzt. Der Account kann immer genutzt werden.

8.1.4 Standardwerte und Einschränkungen

Folgende Standardwerte und Einschränkungen gelten für MpUser:

- Es gibt zwei Standardbenutzergruppen:

Benutzergruppe	Level	Adminrechte	Kommentar
Admin	100	ja	Kann als Systemadministrator direkt verwendet werden. Ist für das Benutzermanagement verantwortlich. Kann Benutzer editieren, sperren, freigeben oder auch löschen.
Operator	10	nein	Hat im Benutzermanagement keine Rechte, bekommt über sein Level Zugang zu bestimmten Maschinenfunktionen.

- Es gibt einen Standardbenutzer:

Benutzername	Passwort	Benutzergruppe
Admin	Admin	Admin

- Es können maximal für 20 unterschiedliche Funktionen Rechte definiert werden
- Es können maximal 5 unterschiedliche Token pro Benutzer vergeben werden (Reserviert für spätere Nutzung)
- Alle Gruppen und Benutzer, welche in der Konfiguration MpUser angelegt werden, werden als Default User bzw Default Gruppe bezeichnet

8.1.5 Elektronische Signatur

Die elektronische Signatur ermöglicht das Zeichnen einzelner **Aktionen**. Möchte der Benutzer eine bestimmte Aktion durchführen, dann öffnet sich auf der Visualisierung ein Dialogfenster, welches Benutzernamen und Passwort fordert. Nach Eingabe der Nutzerdaten wird die Aktion, wenn die Berechtigung vorhanden ist, durchgeführt. Ein Aktion kann beispielsweise sein:

- Parameter über die Visualsierung ändern (z.B. Solltemperatur)
- Reinigungsprozess an der Maschine starten
- Aktuellen Produktionsprozess abbrechen

Damit kann für jede Aktion, welche an der Maschine möglich ist, eine elektronische Signatur verlangt werden. Das Zeichnen wird über ein Ereignis im Audit Trail vermerkt. Das Ereignis wird erzeugt unabhängig davon, ob der Benutzer die entsprechenden Berechtigung besitzt.

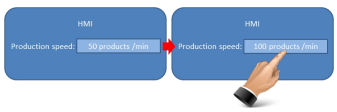
Die Verbindung zur Visualierung erstellt das Benutzermanagement über den Funktionsbaustein MpUserSignatureUI. Über diesen Funktionsbaustein wird die Anzeige des Dialogefensters geregelt. Die Prüfung, ob der Benutzer die Berechtigung für eine Aktion besitzt erfolgt über MpUserSignature. Dieser Funktionsbaustein informiert dann, ob die Signatur erfolgreich war, oder nicht.

Ablauf

Ein möglicher Ablauf beim Zeichnen einer Aktion könnte folgendermaßen aussehen:

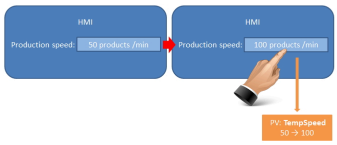
1. Eingabe eines Parameters

Der Benutzer gibt über die Visualisierung einen neuen Wert für die Sollgeschwindigkeit ein. Der Wert der wird von 50 Produkten / Minute auf 100 Produkte / Minute erhöht.



2. Temporäre Lagerung des neuen Parameters

Aufgrund der Eingabe wird eine PV in der Applikation auf den neuen Wert gesetzt. Diese ist eine temporäre PV, welche noch keinen direkten Einfluss auf den Prozess nimmt. Gleichzeitig wird der *Completion Datenpunkt* des Eingabefeldes auf einen bestimmten Wert gesetzt. Dieser Wert entspricht der "ActionID", welche in der MpUserSign Konfiguration festgelegt wurde. Beispielweise ist in der Konfiguration eine Aktion mit der "ActionID" 1 definiert, welche ein Benutzerlevel von 100 voraussetzt. In unserem Beispiel wird als der *Completion Datenpunkt* auf 1 gesetzt.



3. Dialog zur elektronischen Signatur

Erkennt MpUserSignatureUI, dass der *Completion Datenpunkt* ungleich null ist, dann wird der Dialog zur elektronischen Signatur geöffnet.

In diesem Dialog wird der Benutzer aufgefordert seinen Benutzernamen, sowie sein Passwort einzugeben. Optional kann auch ein Kommentar angegeben werden, welcher angibt warum eine bestimmte Aktion durchgeführt werden soll. Der Benutzer hat dann die Wahl den Prozess durch *OK* zu Bestätigen, oder durch *Cancel* abzubrechen.

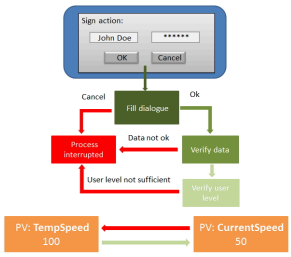
4. Prozess zur Verifizierung der eingegebenen Daten

Nun startet ein Prozess zur Überprüfung der Benutzereingaben. Im Bild unten zeigen die rot markierten Objekte einen unterbrochenen Signier - Prozess an. Das kann mehrere Ursachen haben:

- Der Benutzer drückt *Cancel*, um den Prozess bewusst abzubrechen.

  Wird der Signaturprozess vom Benutzer abgebrochen, so wird der Ausgang "Rejected" an MpUserSignature für einen Zyklus gesetzt
- Der Benutzer hat eine falsche Eingabe gemacht, heißt entweder Benutzername oder Passwort stimmt nicht. In diesem Fall wird auch ein Login-Versuch ("LoginAttempts") verwendet. Das bedeutet, wenn der Benutzername oder das Passwort zu oft falsch eingegeben werden, wird der Benutzer gesperrt. Die Anzahl an Login-Versuchen wird in der MpUser Konfiguration definiert. Werden alle Signatur-Versuche aufgebraucht, so wird der Signatur-Vorgang abgebrochen. Der Signatur-Vorgang kann wieder neu gestartet werden, dabei steht wieder die gleiche Anzahl an Signatur-Versuchen zur Verfügung. Die Login-Versuche werden nicht wieder erhöht.
- Der Benutzer hat nicht das notwendige Benutzerlevel, um die geforderte Aktion durchzuführen, heißt sein Benutzerlevel ist kleiner, als das für die Durchführung der Aktion benötigte Benutzerlevel.

  Schlägt der Signaturprozess fehlt (siehe die beiden letzten Punkte) kann der Vorgang wiederholt werden. Wird der maximal erlaubte Wert an fehlgeschlagenen Versuchen erreicht, dann wird der Ausgang "Rejected" an MpUserSignature für einen Zyklus gesetzt
- In der Applikation wird in diesem Fall der Wert der temporären PV wieder auf den Wert der aktiven Prozessvariable gesetzt. In unserem Beispiel wird der, zur Produktion verwendete, Geschwindigkeitswert von 50 Produkten / Minute auf die temporäre PV geschrieben



Der grüne Zweig zeigt Schritt für Schritt die erfolgreiche Durchführung einer elektronischen Signatur an:

- Zuerst bestätigt der Benutzer, dass er eine Aktion durchführen möchte durch das Drücken der *OK* Taste
- MpUserSignature überprüft nun, ob Benutzername, sowie Passwort stimmen

- Stimmen die Nutzerdaten, wird im nächsten Schritt überprüft, ob der Benutzer die angegebenen Berechtigungen besitzt (Benutzerlevel). In unserem Beispiel wird geprüft, ob der Benutzer mindestens ein Benutzerlevel von 100 besitzt
- Falls alle Punkte gegeben sind war die elektronische Signatur erfolgreich. Der Ausgang "Released" von MpUserSignature wird für einen Zyklus gesetzt
- In der Applikation wird in diesem Fall der Wert der aktiven Prozessvariable auf den Wert der temporären PV gesetzt. In unserem Beispiel wird der, über die Visualisierung eingegebene und signierte, Geschwindigkeitswert von 100 Produkten / Minute auf die zur Produktion verwendete PV geschrieben

8.1.6 Auditing des Benutzermanagements

Für diese Komponente können Ereignisse über MpAudit aufgezeichnet werden. Für mehr Information wie ein Eintrag erstellt werden kann, siehe Abschnitt Definieren des Ereignistexts.

Die Identifikationsnummer kann anhand des Kürzels **%act** in den jeweiligen Eintrag eingebunden werden.

Folgende Ereignisse können für diese mapp Komponenten auftreten:

In diesem Abschnitt werden die verschiedenen Benutzer-Ereignisse aufgelistet:

> **Werden Ereignisse über eine mapp View Visualisierung ausgelöst, können manche Ereignisse nicht einem Benutzer zugeordnet werden. Dies muss, bei der Verwendung des "op" Kürzels beachtet werden. Für weitere Information siehe Abschnitt Ereignisse einem Benutzer zuordnen.**

Benutzermanagement Ereignis

Durch diesen Ereignistyp können Ereignisse des Benutzermanagements, wie zum Beispiel das Löschen eines Benutzers oder Hinzufügen eines Benutzers, aufgezeichnet werden.

Ereignis ID **2**	**Komponenten** **MpUser Komponenten | MpUserX Komponenten**
Ereignisse	
op	Name des Benutzers welcher beim Auftreten des Ereignisses eingeloggt war
ev	Ereignis ID

Tabelle 3: Benutzermanagement Ereignis

Ereignis ID 2	Komponenten MpUser Komponenten | MpUserX Komponenten
idx	Liefert den Index des aktuell generierten Ereignisses. Der Wert wird bei jedem neuen Ereignis inkrementiert. Das erste generierte Ereignis in der Ereignisliste hat somit den Wert 1, dass fünfzigste Ereignis hat den Index 50, das hundertste den Index 100 und so weiter.
map	Name der mapp Komponente, welcher das Ereignis über den MpLink zugeordnet ist
usr	Benutzer, welcher von dem Ereignis betroffen ist (z.B. Benutzer der gelöscht wurde)
act	Aktion, welche in einer mapp Komponente ausgeführt wurde. Die ID für die Aktion wird zurükgegeben (z.B. 10 für "Ungültiger Benutzer-name wurde eingegeben"). Für mehr Informationen siehe folgende Tabelle:

Ereignis	Identifikationsnummer (wird über %act zurückgegeben)	Zusatzinformation
Unbekannte Aktion	0	
Benutzer hat sich eingeloggt	1	• {&op}/[%op] - Benutzer
Benutzer hat sich ausgeloggt	2	• {&op}/[%op] - Benutzer
Benutzer wurde automatisch ausgeloggt	3	• {&op}/[%op] - Benutzer
Benutzer wurde hinzugefügt	4	• {&op}/[%op] - welcher Benutzter hat die Operation durchgeführt • {&usr}/[%usr] - welcher Benutzer wurde hinzugefügt
Benutzer wurde gelöscht	5	• {&op}/[%op] - welcher Benutzter hat die Operation durchgeführt • {&usr}/[%usr] - welcher Benutzer wurde gelöscht
Benutzer wurde gesperrt	6	• {&op}/[%op] - welcher Benutzter hat die Operation durchgeführt • {&usr}/[%usr] - welcher Benutzer wurde gesperrt
Benutzer wurde entsperrt	7	• {&op}/[%op] - welcher Benutzter hat die Operation durchgeführt • {&usr}/[%usr] - welcher Benutzer wurde entsperrt
Benutzer wurde gesperrt, weil er zu oft ein falsches Passwort eingegeben hat	8	• {&op}/[%op] - welcher Benutzter hat die Operation durchgeführt • {&usr}/[%usr] - welcher Benutzer wurde gesperrt
Benutzer hat das Passwort geändert	9	• {&op}/[%op] - Benutzer
Ungültiger Benutzername wurde eingege-ben	10	
Ungültiges Passwort wurde eingegeben	11	
Rolle hinzugefügt	12	• {&op}/[%op] - welcher Benutzter hat die Operation durchgeführt • {&usr}/[%usr] - welche Rolle wurde hinzugefügt
Rolle gelöscht	13	• {&op}/[%op] - welcher Benutzter hat die Operation durchgeführt • {&usr}/[%usr] - welche Rolle wurde gelöscht
Ungültige Anmeldeinformationen wurden eingegeben	14	

Tabelle 3: Benutzermanagement Ereignis

Wie der Ereignistext angegeben werden kann ist in Abschnitt Verwendung des neuen Formats erklärt.

Beispiel neues Format:

- **'{&op} - Aktion: Benutzer {&usr} wurde gelöscht'** ausgewertet: 'Admin Aktion: Benutzer Mike wurde gelöscht' , was bedeutet der Benutzer Admin hat den Benutzer Mike gelöscht

Beispiel altes Format:

- **'[%op] - Aktion: Benutzer [%usr] wurde gelöscht'** ausgewertet: 'Admin Aktion: Benutzer Mike wurde gelöscht' , was bedeutet der Benutzer Admin hat den Benutzer Mike gelöscht

Elektronische Signatur Ereignis

Durch diesen Ereignistyp können Ereignisse rund um die Elektronische Signatur aufgezeichnet werden.

Ereignis ID 3	Komponenten MpUserSignature | MpUserSignatureUI MpUserXSignature | MpUserXSignatureUI
Ereignisse	
cmt	Kommentar, welcher vom Benutzer bei der Signatur angegeben wurde
id	Name der mapp Komponente, welcher das Ereignis über den MpLink zugeordnet ist

Tabelle 4: Elektronische Signatur Ereignis

Ereignis ID 3	Komponenten MpUserSignature | MpUserSignatureUI MpUserXSignature | MpUserXSignatureUI
idx	Liefert den Index des aktuell generierten Ereignisses. Der Wert wird bei jedem neuen Ereignis inkrementiert. Das erste generierte Ereignis in der Ereignisliste hat somit den Wert 1, dass fünfzigste Ereignis hat den Index 50, das hundertste den Index 100 und so weiter.
err	Fehlernummer des Benutzermanagements (kann beispielsweise verwendet werden, um anzuzeigen ob der Benutzername oder das Passwort falsch waren)
act	Liefert die Aktions-ID welche in der Konfiguration für eine Signatur angegeben werden muss
typ	Aktion, welche in einer mapp Komponente ausgeführt wurde. Die ID für die Aktion wird zurükgegeben (z.B. 2 für "Elektronische Signatur wurde abgebrochen"). Für mehr Informationen siehe folgende Tabelle:

Ereignis	Identifikationsnummer (wird über typ zurückgegeben)	Ereignis als Text (wird über typt zurückgegeben)
Elektronische Signatur wurde erfolgreich durchgeführt	0	OK
Elektronische Signatur wurde beendet aufgrund einer falschen Eingabe (Benutzername oder Passwort)	1	FAIL
Elektronische Signatur wurde vom Benutzer abgebrochen	2	ABORT
Elektronische Signatur wurde beendet, da das Limit für falsche Eingaben (siehe Konfiguration des Benutzermanagements) überschritten wurde	3	FATAL

Tabelle 4: Elektronische Signatur Ereignis

Wie der Ereignistext angegeben werden kann ist in Abschnitt Verwendung des neuen Formats erklärt.

Beispiel neues Format:

- **'Signatur ID {&act}: {&typ} ({&typt}) '** ausgewertet: 'Signatur ID 3: Elektronische Signatur wurde erfolgreich durchgeführt (OK)', was bedeutet, dass die Signatur mit der ID 3 erfolgreich durchgeführt wurde

Beispiel altes Format:

- **'Signatur ID [%act]: [%typ] ([%typt] '** ausgewertet: 'Signatur ID 3: Elektronische Signatur wurde erfolgreich durchgeführt (OK)', was bedeutet, dass die Signatur mit der ID 3 erfolgreich durchgeführt wurde

8.1.7 Importieren von extern erstellten Benutzern

In mapp UserX können extern erstellte Benutzer importiert werden.

Folgende Punkte müssen beachtet werden.

Anpassen der Import Datei

Eine zu importierende Datei kann zum Beispiel folgendermaßen aussehen:

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA>
    <Format Type="MpUser" Version="1.00" Check="2339272230" />
    <Groups>
        <Group ID="Admin" Level="100" Admin="true" Check="1236056579">
            <Right ID="0" Access="FullAccess" />
            <Right ID="1" Access="FullAccess" />
            <Right ID="2" Access="FullAccess" />
            <Right ID="3" Access="FullAccess" />
            <Right ID="4" Access="FullAccess" />
        </Group>
        <Group ID="Supervisor" Level="50" Admin="true" Check="2778008081">
            <Right ID="0" Access="FullAccess" />
            <Right ID="1" Access="FullAccess" />
            <Right ID="2" Access="FullAccess" />
            <Right ID="3" Access="FullAccess" />
            <Right ID="4" Access="ViewOnly" />
        </Group>
        <Group ID="Operator" Level="10" Admin="false" Check="3531168924">
            <Right ID="0" Access="FullAccess" />
            <Right ID="1" Access="StartOnly" />
            <Right ID="2" Access="ViewOnly" />
            <Right ID="3" Access="NoAccess" />
            <Right ID="4" Access="NoAccess" />
        </Group>
    </Groups>
    <Users>
        <User ID="Admin" Group="Admin" Name="Default User" Pref="" Pwd="133d4ba6eae6416454f1b10c0df4cb0e" Check="1
```

```
        <User ID="Operator" Group="Operator" Name="Operator User" Pref="" Pwd="06c8f5530383b158ed5f072bb842cd80" C
        <User ID="Visitor" Group="Operator" Name="Unknown user" Pref="" Pwd="9ed0f0162cb321bd778dc48e3be5321e" Che
    </Users>
</DATA>
```

Soll ein externer Benutzer (ExternUser) hinzugefügt werden, so muss die Importdatei erweitert werden. Dabei muss das <User> Tag um den Eintrag *Ext="1"* erweitert werden. Dadurch wird ein externer Benutzer gekennzeichnet.

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA>
    <Format Type="MpUser" Version="1.00" Check="2339272230" />
    <Groups>
        <Group ID="Admin" Level="100" Admin="true" Check="1236056579">
            <Right ID="0" Access="FullAccess" />
            <Right ID="1" Access="FullAccess" />
            <Right ID="2" Access="FullAccess" />
            <Right ID="3" Access="FullAccess" />
            <Right ID="4" Access="FullAccess" />
        </Group>
        <Group ID="Supervisor" Level="50" Admin="true" Check="2778008081">
            <Right ID="0" Access="FullAccess" />
            <Right ID="1" Access="FullAccess" />
            <Right ID="2" Access="FullAccess" />
            <Right ID="3" Access="FullAccess" />
            <Right ID="4" Access="ViewOnly" />
        </Group>
        <Group ID="Operator" Level="10" Admin="false" Check="3531168924">
            <Right ID="0" Access="FullAccess" />
            <Right ID="1" Access="StartOnly" />
            <Right ID="2" Access="ViewOnly" />
            <Right ID="3" Access="NoAccess" />
            <Right ID="4" Access="NoAccess" />
        </Group>
    </Groups>
    <Users>
        <User ID="Admin" Group="Admin" Name="Default User" Pref="" Pwd="133d4ba6eae6416454f1b10c0df4cb0e" Check="
        <User ID="Operator" Group="Operator" Name="Operator User" Pref="" Pwd="06c8f5530383b158ed5f072bb842cd80" C
        <User ID="Visitor" Group="Operator" Name="Unknown user" Pref="" Pwd="9ed0f0162cb321bd778dc48e3be5321e" Che
        <User ID="ExternUser" Ext="1" Group="Admin" Name="ExternUser" Pref="/LANG=0 /UNIT=0" Pwd="Test99" Check="
            <AdditionalData Key="PhoneNumber" Value="123" />
            <AdditionalData Key="eMail" Value="a.b@c.de" />
            <AdditionalData Key="custom" Value="wasd" />
        </User>
    </Users>
</DATA>
```

Anpassen der Konfiguration

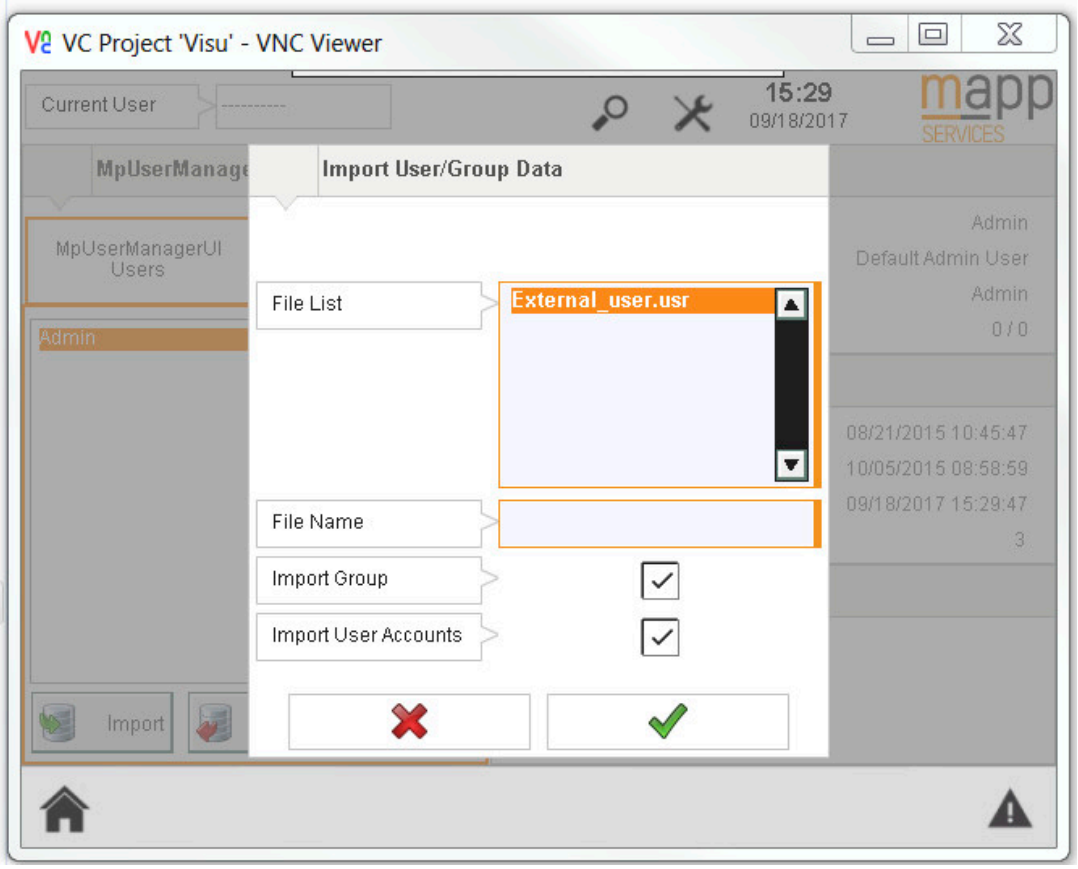
In der MpUser Konfiguration muss die Option "UncheckedImport=`TRUE`" sein. Die Importdatei ist durch eine Prüfsumme geschützt. Ist die Option deaktiviert und die Importdatei wurde um einen Benutzer erweitert, kann die Datei nicht importiert werden.

Wird die Importdatei nicht kontrolliert("UncheckedImport=`TRUE`"), wird die Prüfsumme nicht beachtet und ein extern erstellter Benutzer kann importiert werden.

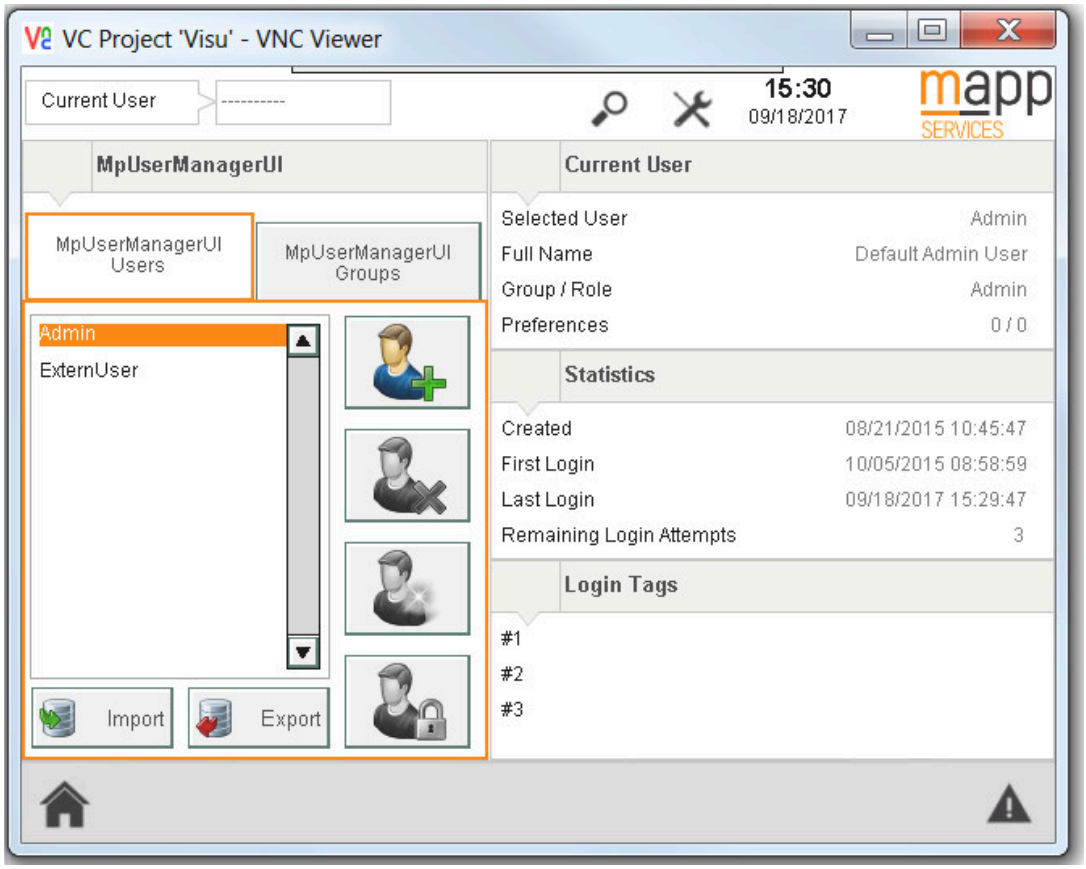
Importieren der Datei

Die Importdatei kann mithilfe von MpUserManagerUI über die Visualisierung importiert werden. Es stehen mehrere verschiedene Import-Modi zur Auswahl, welche in der MpUser Konfiguration festgelegt werden können.

Die gewünschte Datei wird ausgewählt und Importiert:



Der extern angelegte Benutzer wird automatisch angelegt.



8.1.8 Rahmenbedingungen

In diesem Abschnitt werden Rahmenbedingungen von mapp User aufgelistet.

Allgemein

- In einem Projekt kann nur mit mapp User oder mit mapp UserX gearbeitet werden. Nicht mit beidem parallel. Beide Komponenten dienen zum Einrichten eines Benutzermanagements. In mapp User werden die Benutzer in Gruppen mit unterschiedlichen Berechtigungen unterteilt. In mapp UserX werden Rollen und Benutzer über das Automation Studio Benutzer- Rollensystem, damit auch OPC UA konform, erstellt und verwaltet.

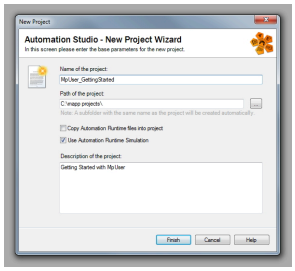
8.2 Getting Started

8.2.1 Einfaches Benutzermanagement

In diesem Abschnitt wird ein einfaches Benutzermanagement mithilfe von MpUser erstellt. Der Benutzer kann ein- und ausloggen. Außerdem wird über die mapp Konfiguration ein neuer Benutzer hinzugefügt.

8.2.1.1 Projekt erstellen

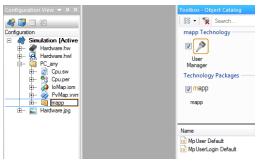
Zuerst muss ein neues Projekt über Automation Studio erzeugt werden.



8.2.1.2 mapp Komponente Einfügen

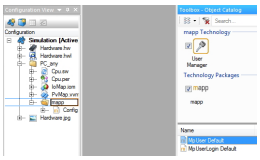
Konfiguration für MpUser einfügen

Zunächst wird über die Toolbox die gewünschte mapp Komponente ausgewählt.



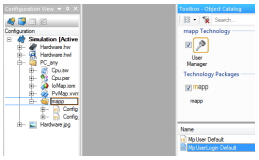
Konfiguration für ein Benutzermanagement Einfügen

Beim nächsten Schritt wird über die Toolbox die gewünschte mapp Komponente ausgewählt und die Konfiguration dann per Drag&Drop im "mapp" Ordner eingefügt. Diese Konfiguration beschreibt das Benutzermanagement allgemein. Alle Parameter in dieser Konfiguration gelten für alle Login - Punkte.



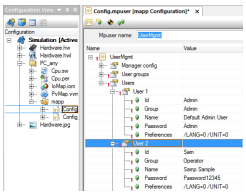
Konfiguration für einen Login Punkt Einfügen

Für das Benutzermanagement ist auch eine Konfiguration für jeden Login Punkt notwendig. Diese werden dann auch einen eigenen MpLink besitzen. Diese wird auch per Drag&Drop über die Toolbox eingefügt.



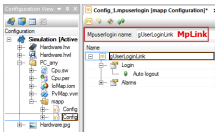
Konfiguration für ein Benutzermanagement Editieren

Die Konfiguration für das Benutzermanagement allgemein iwrd dahingehend editiert, dass ein neuer Benutzer mit Namen *Sam* eingefügt wird.



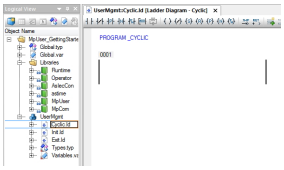
MpLink benennen

Beim Einfügen der Konfiguration, des Login Punktes, wurde der MpLink automatisch erstellt. Wenn die Konfiguration geöffnet wird, ist der MpLink ganz oben zu finden.



8.2.1.3 Programm hinzufügen

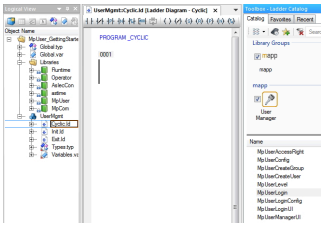
Der nächste Schritt ist das Einfügen eines Programms. In diesem Fall wird ein Kontaktplan Programm eingefügt. Jede andere Programmiersprache kann genauso verwendet werden.



8.2.1.4 MpUserLogin im Programm einfügen

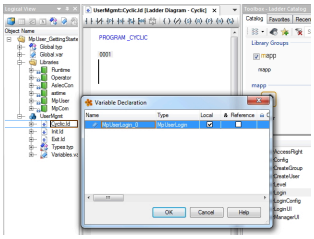
MpUser im Kontaktplan Katalog suchen

Im Katalog in Kontaktplan kann die Checkbox mapp ausgewählt werden. Danach werden darunter alle verfügbaren mapp Komponenten angezeigt, welche im Projekt sind. In unserem Fall zählt das Benutzermanagement dazu.



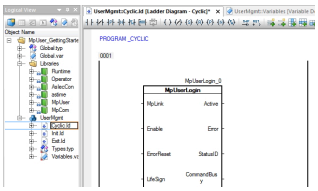
Funktionsbaustein auswählen

Den Kern des Benutzermanagement bildet MpUserLogin. Über Drag & Drop wird der Funktionsbaustein in das Programm gezogen.



Funktionsbaustein einfügen

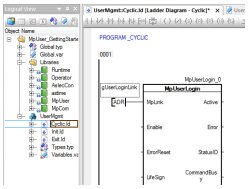
Nachdem wird den Funktionsbaustein ins Programm gezogen haben, wird eine Funktionsbaustein-Instanz erzeugt.



8.2.1.5 Parametrieren von MpUserLogin

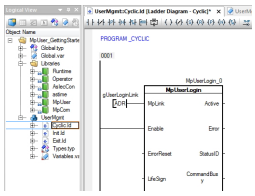
Verbinden des MpLinks

Zuerst wird der Eingang "MpLink" mit dem in der Configuration View bereits erstellten MpLink verbunden. Die Adresse wird an den Funktionsbaustein übergeben.



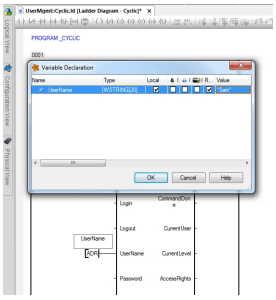
Aktivieren der Komponente

Durch Setzen des "Enable" Eingangs wird die Komponente aktiviert. Das wird über den Ausgang "Active" angezeigt.

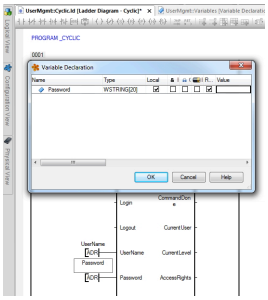


Benutzer und Passwort verbinden

Zwei Variablen für Benutzer und Passwort werden erstellt und mit der mapp Komponente verbunden. Zuerst wird eine Variable für den Benutzer *Sam* erstellt und verbunden.



Weiters muss eine Variable für das Passwort verbunden werden.

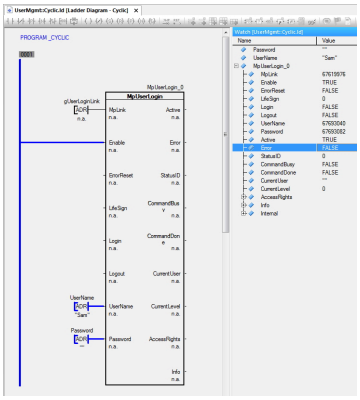


8.2.1.6 Dateistruktur erzeugen

Als nächstes wird eine Dateistruktur für die ARsim erzeugt. Für mehr Information siehe hier.

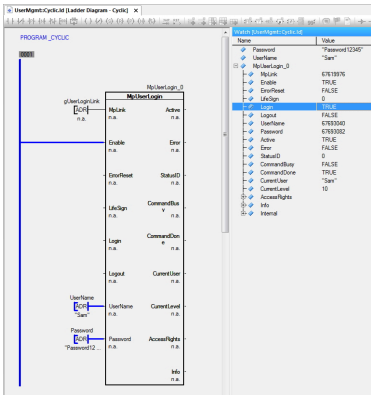
8.2.1.7 Einloggen über Watch Fenster

Sobald die Steuerung im RUN Zustand ist kann beim Funktionsbaustein überprüft werden, ob er aktiv ist. Ist er aktiv, können weitere Befehle abgesetzt werden.



Einloggen

Nach dem Eingeben des Passwortes, kann man sich über den Befehl "Login" Einloggen.



8.3 Konfiguration

8.3.1 MpUser Konfiguration

Diese Konfiguration wird immer benötigt, wenn ein Benutzermanagement über mapp realisiert werden soll. Die Daten in dieser Konfiguration gelten für die komplette Maschine. Deshalb darf nur eine solche Konfiguration eingefügt werden pro Maschine (bzw. pro Maschinenkonfiguration in der Configuration View in Automation Studio). Allle Benutzergruppen und Benutzer, die über diese Konfiguration erstellt werden, gelten an jedem Loging - Punkt (Panel, an welchem man sich über Benutzername und Passwort einloggen kann). Der MpLink dieser Konfiguration wird für keine(n) Funktion bzw. Funktionsbaustein von MpUser verwendet.

Die Konfiguration kann zur Laufzeit über den Funktionsbaustein MpUserConfig angepasst werden.

Die hier in der Konfiguration existierenden Passwörter können verschlüsselt werden. Für mehr Information siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" unter Rahmenbedingungen.

id_Mgr_PasswordChangeReq

Ändert ein Administrator das Passwort eines Benutzers muss dieser sein Passwort beim nächsten Login ändern. Solange der Benutzer das Password nicht ändert hat dieser keine Zugriffsrechte. Um dieses Verhalten zu ermöglichen muss dieser Parameter auf *TRUE* gesetzt werden.

id_Mgr_PasswordHistory

Der Parameter gibt an wie viele Passwörter für jeden User gespeichert werden und nicht wiederverwendet werden dürfen. Der Wert darf zwischen 0 und 5 liegen. Bei 0 ist die Funktion ganz ausgeschaltet.

Allgemeine Einstellungen

Im ersten Abschnitt der Konfiguration können allgemeine Einstellungen festgelegt werden.

Alle Passwort Parameter gelten nur für Benutzer die neu über die Visualisierung mithilfe von MpUserManagerUI angelegt werden. Für Benutzer die über das Konfigurationssystem in Automation Studion, oder über MpUserCreateUser erstellt werden, gelten diese Regeln nicht.

- **Password change interval:** Intervall, in dem das Passwort geändert werden muss
- **User expiration time:** Verfallszeit für die Benutzerkonto Gültigkeit
- **Admin unlock time:** Nach Ablauf der angegebenen Zeit wird der gesperrte Administrator entsperrt. (Eiheit: [sec]) Wird 0 angegeben, wird der Administrator nicht automatisch entsperrt. Die Entsperr-Funktion ist bei 0 nicht verfügbar
- **Password change required:** Passwortänderung beim ersten Login notwendig. Ändert ein Administrator das Passwort eines Benutzers muss dieser sein Passwort beim nächsten Login ändern. Solange der Benutzer das Password nicht ändert hat dieser keine Zugriffsrechte. Um dieses Verhalten zu ermöglichen muss dieser Parameter auf *TRUE* gesetzt werden.
- **Password case:** Passwort muss Klein- und Großbuchstaben enthalten
- **Password alphanumeric:** Passwort muss alphanumerisch sein
- **Password length:** Minimale Passwortlänge. Es muss mindestens ein Zeichen angegeben werden. Es ist nicht erlaubt eine Passwort-Länge von 0 zu haben.
- **Password special charachters:** Bestimmt, ob ein Passwort Sonderzeichen enthalten muss
- **Username length:** Minimale Länge bei der Eingabe eines neuen Benutzernamens
- **Password History Size:** Gibt an wie viele vorherigen Passwörter nicht wiederverwendet werden dürfen Es kann ein Wert zwischen 0 und 5 eingegeben werden Bei der Angabe von 0 wird die Funktion nicht verwendetBei der Angabe von zum Beispiel 2, darf das neu eingegebene Passwort nicht mit den zwei vorherigen Passwörtern übereinstimmen
- **No delete:** Löschen von Benutzern nicht erlaubt
- **Login attempts:** Maximale Anzahl an möglichen Login - Fehlversuchen, bis der Benutzer gesperrt wird. Bei der Eingabe von 0 wird der Benutzer niemals gesperrt. Bei Benutzergruppen mit Adminrechten werden diese maximal eine Stunde gesperrt, bzw. sind Benutzergruppen mit Adminrechten nach einem Warmstart der Maschine nicht mehr gesperrt.
- **Signature attempts:** Maximale Anzahl an möglichen Signatur- Fehlversuchen, bis der Signatur-Vorgang abgebrochen wird. Die Login-Versuche ("LoginAttempts") und die Signatur-Versuche hängen miteinander zusammen. Bei jeder falschen Benutzereingabe während eines Signatur-Versuchs wird auch ein Login-Versuch verwendet. Gibt es 2 Signatur-Versuche und 1 Login-Versuch, so wird der Benutzer bei einem fehlgeschlagenen Signatur-Versuch (Eingabe eines falschen Passworts/Benutzernamens) bereits gesperrt. Wird die Signatur abgebrochen, so bleibt der Login-Versuch erhalten. Für mehr Information siehe hier.
- **Edit users with same user-level:** Bestimmt, ob ein Admin Benutzer mit dem gleichem Benutzerlevel editieren darf
- **User Import-Mode:** Modus für Gruppenimport. Für mehr Information siehe "Import Modi".
- **Group Import Mode:** Modus für Gruppenimport Für mehr Information siehe "Import Modi".
- **Unchecked Import:** Ignorieren der Check-Summe in der Importdatei. Für mehr Information siehe "Import Modi".

Benutzergruppen

Im Abschnitt "User groups" können Benutzergruppen festgelegt werden. Pro Benutzergruppen muss ein eindeutiger Name über "Name" vergeben werden. "Level" bestimmt welches Benutzerlevel die Gruppe haben soll. Über "Administrator" kann definiert werden, ob die Benutzergruppe ein Administrator sein soll oder nicht. Über "Rights" werden die Rechte für die Gruppe festgelegt. Für mehr Information siehe Rechte, Levels und Funktionen.

Benutzer

Im Abschnitt "User" können die einzelnen Benutzer angelegt werden. Pro Benutzer muss ein eindeutiger Name über "UserName" vergeben werden. "GroupName" definiert welcher Benutzergruppe der Benutzer zugeteilt ist. Über "FullName" kann der volle Name des Benutzer angegeben werden. Durch "Password" wird das Passwort des Benutzers bestimmt. "Preferences" legt fest, welche Sprache und Einheit vom Benutzer bevorzugt wird.

Unter "Additional Data" kann für den Benutzer zusätzliche Information hinterlegt werden. Wie zum Beispiel die Telefonnummer (telNr) und die E-Mail Adresse (eMail).

Import Modi

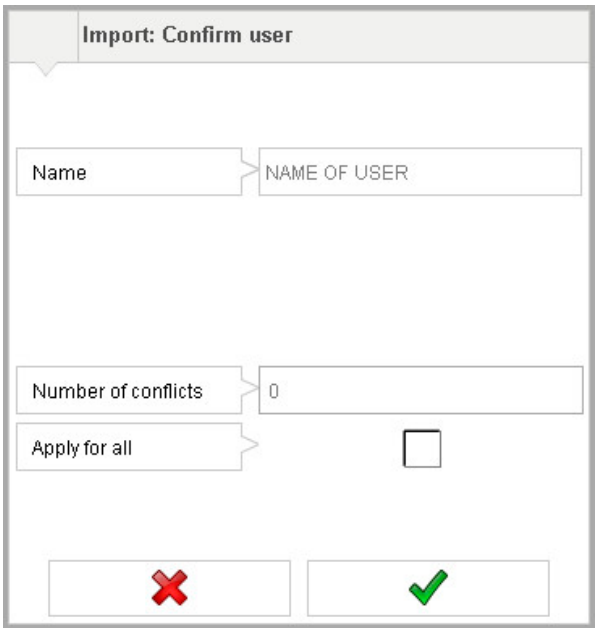
Mithilfe von MpUserManagerUI ist es möglich Benutzerdaten zu importieren bzw. zu exportieren.

In der Konfiguration kann unter dem Abschnitt *User Import Mode* und *Group Import Mode* definiert werden, wie die Daten importiert werden sollen. Dabei gibt es 6 verschiedene Modi:

- **Skip (Do not import):** Es werden keine Daten importiert
- **Ignore Existing Items:** Es werden alle neuen Daten (Benutzer/Benutzergruppen) importiert. Bereits bestehende Daten werden nicht überschrieben.
- **Overwrite Existing Items:** Es werden alle Daten importiert. Bereits bestehende Daten werden überschrieben. Default Benutzer, welche in der Konfiguration MpUser angelegt wurden, werden dabei auch überschrieben und sind somit nicht mehr in der Konfiguration als Default Benutzer vorhanden.
- **Ignore Default Items:** Es werden alle Benutzerdaten importiert. Default Benutzer, welche bereits über die Automation Studio Konfiguration angelegt wurden, werden nicht überschrieben.
- **Overwrite Items Only:** Es werden alle bereits existierenden Daten überschrieben. Neue Daten werden nicht importiert. Default Benutzer, welche in der Konfiguration MpUser angelegt wurden, werden dabei auch überschrieben und sind somit nicht mehr in der Konfiguration als Default Benutzer vorhanden.
- **Remove Existing Items:** Es werden alle Daten importiert. Bereits vorhandene Daten, welche aber nicht in der importierten Datei vorkommen, werden gelöscht.

Wird eine Datei importiert, welcher bestehende Benutzer/Gruppen überschreibt, dann kann dieser Vorgang abgefragt werden. Diese Abfrage erfolgt in Form eines Dialogfensters. Es wird auch angezeigt wieviele Dateien sich in einem Konflikt befinden. Der Überschreibvorgang kann auch für alle Dateien bestätigt werden.

Diese Option wird in der Struktur MpUserMgrUISetupType über die Parameter "ConfirmOverwriteUser" und "ConfirmOverwriteGroup" aktiviert.



8.3.2 MpUserLogin Konfiguration

Die Daten dieser Konfiguration gelten nur für einen Login - Punkt (Panel, an welchem sich jemand über Benutzernamen und Passwort einloggen kann). Ein Login - Punkt bezeichnet einen Station, an welcher der Benutzer die Maschine / einen Teil der Maschine steuern kann. Für jeden Login - Punkt den es für eine Maschine gibt, wird eine dieser Konfigurationen benötigt. Zumindest eine ist notwendig, sonst kann sich niemand einloggen. Bei den unterschiedlichen Funktionen und Funktionsbausteinen, welche MpUser bietet, wird immer mit dem MpLink dieser Konfiguration gearbeitet.

> **Die Konfiguration kann zur Laufzeit über den Funktionsbaustein MpUserLoginConfig angepasst werden.**

Der Parameter "AutoLogout" wird automatisch auf den angebenen Wert zurückgesetzt sobald sich der Eingang "LifeSign" von MpUserLogin erhöht. Verändert sich der Wert an "LifeSign" während der angegebenen Ausloggzeit nicht, wird der Benutzer automatisch ausgeloggt.

8.3.3 MpUserSign Konfiguration

> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein de-, und wieder aktiviert werden.

Die geänderten Konfigurationsparameter werden sofort von der Komponente übernommen.

In der Konfiguration können Einstellungen für die Signatur festgelegt werden.

"User Level" bestimmt welches Benutzerlevel mindestens verwendet werden muss, um die Signatur zu starten.

Im Abschnitt "Actions" können die einzelnen Aktionen festgelegt werden. Pro Aktion muss eine eindeutige ID über "ActionID" angegeben werden. "UserLevel" bestimmt welches Benutzerlevel mindestens für diese Aktion verwendet werden muss. Für mehr Information siehe Elektronische Signatur.

8.4 Anwendungsfälle

In diesem Abschnitt werden mögliche Anwedungsfälle der MpUser Komponenten beschrieben.

8.4.1 Einfache Benutzerverwaltung

Anforderung

Auf der Maschine soll eine einfache Benutzerverwaltung integriert werden. Es sollen 3 Benutzer mit 3 verschiedenen Benutzerebenen (Levels) angelegt werden. Abhängig von der aktuellen Benutzerebene des angemeldeten Benutzers werden dann verschiedenen Funktionen in der Visualisierung freigeschalten / gesperrt.

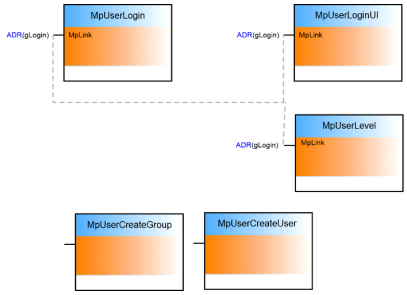
mapp V0.05.x: Das Passwort und die Benutzerebene für den Standard Benutzer "Admin" kann überschrieben werden.

Lösung

Komponentenliste

- ***MpUserCreateGroup*** (kein MpLink): Erstellen einer neuen Benutzergruppe (nur für mapp < V1.00)
- ***MpUserCreateUser*** (kein MpLink): Erstellen eines neuen Benutzers (nur für mapp < V1.00)
- ***MpUserLogin*** (Eigener MpLink): Anmelden / Abmelden eines Benutzers
- ***MpUserLoginUI*** (MpLink von MpUserLogin): Anmelden / Abmelden eines Benutzers über die Visualisierung
- ***MpUserLevel*** (MpLink von MpUserLogin): Abfrage der aktuellen Benutzerebene

Verbindungsschema



Verwendung der mapp Komponenten

mapp V0.05.0: Mit den Komponenten **MpUserCreateGroup** und **MpUserCreateUser** kann man über die Applikation eine Benutzerkonfiguration erstellen. Die Bibliothek selbst erstellt einen Standard-Benutzer "Admin" und 2 Standardgruppen ("Admin" und "Operator"). Diese Komponenten werden verwendet, um die Einstellungen für diese Benutzer / Gruppen zu überschreiben oder neue zu erstellen.

mapp V1.00.0: Die Standard-Benutzerkonfiguration kann direkt im AS-Projekt (Konfigurationssystem AS 4.2) bearbeitet werden. Es besteht keine Notwendigkeit, **MpUserCreateGroup** oder **MpUserCreateUser** in der Applikation zu verwenden.

Der **MpUserLogin** erzeugt einen Anmeldepunkt, an dem sich ein Benutzer mit seinem Benutzernamen (Identifikation) und seinem Passwort anmeldet.

Der **MpUserLoginUI** liefert die entsprechenden Datenpunkte für eine VC4 Visualisierung (z.B. Anmelden, Passwortänderung, usw.)

Mit **MpUserLevel** ist es möglich, die aktuelle Benutzerebene in der Applikation abzufragen. Diese Informationen sind ebenfalls am Ausgang der **MpUserLogin** Komponente verfügbar.

8.4.2 Benutzerrechte

Anforderung

Auf einer Maschine soll jeder Benutzer andere Rechte besitzen. Das heißt für manche Benutzer sind Funktionen der Maschine gesperrt, wie zum Beispiel das ändern von Rezeptdaten, und für andere Benutzer sind diese Funktionen freigeschaltet.

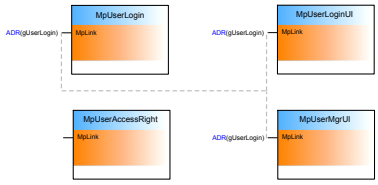
Dazu soll eine Liste der Benutzergruppen mit unterschiedlichen Rechten vorhanden sein. Diese Liste wird bei der Entwicklung der Maschine festgelegt, kann aber auch zur Laufzeit noch weiter verändert werden. Das heißt als Beispiel kann der Administrator den Maschinenbediener weitere Rechte freischalten oder auch sperren. Der Administrator ist nur in der Lage Rechte zu vergeben welche er auch besitzt.

Lösung

Komponentenliste

- ***MpUserLogin*** (Eigener MpLink): Anmelden / Abmelden eines Benutzers
- ***MpUserLoginUI*** (MpLink von MpUserLogin): Anmelden / Abmelden eines Benutzers über die Visualisierung
- ***MpUserManagerUI*** (MpLink von MpUserLogin): Kontrolliert die Benutzerrechte
- ***MpUserAccessRight*** (MpLink von MpUserLogin): Prüfung von Benutzerrechte

Verbindungsschema



Konfiguration

Es werden die Konfigurationen MpUser und MpUserLogin eingefügt. In der Konfiguration MpUser werden die verschiedenen Benutzergruppen angelegt sowie deren Rechte vergeben. Die Benutzerrechte sind an bestimmte Funktionen der Maschine verknüpft. Die Rechteverwaltung kann folgendermaßen aussehen:

Funktion	Rechte	Operator 1	Operator 2	Supervisor
Bewegung der Standard Achse	1	Actuate (Bewegung durchführen)	Actuate (Bewegung durchführen)	Full (Bewegung durchführen + ändern von Achsparametern)
Bewegung der Service Achse	2	None	Actuate (Bewegung durchführen)	Full (Bewegung durchführen + ändern von Achsparametern)
Quittieren kritischer Alarme	3	View (Ansicht der Alarmliste)	View (Ansicht der Alarmliste)	Full (Ansicht der Alarmliste + Quittieren von Alarmen)
Verändern der Rezeptdaten	4	View (Ansicht der Rezeptliste)	Actuate (Ansicht der Rezeptliste + Laden von Rezepten)	Full (Ansicht der Rezeptliste + Laden von Rezepten + Speichern von Rezepten)

User groups		
User Group: Operator1		
Name	Operator1	Unique Name of user-group
Level	10	User level
Administrator	FALSE	Defines whether the group has administrator rights
Rights		Access rights
Access right 1	Actuate	
Access right 2	None	
Access right 3	View	
Access right 4	View	
Access right 5	*Undefined*	
User Group: Operator2		
Name	Operator2	Unique Name of user-group
Level	20	User level
Administrator	FALSE	Defines whether the group has administrator rights
Rights		Access rights
Access right 1	Actuate	
Access right 2	Actuate	
Access right 3	View	
Access right 4	Actuate	
Access right 5	*Undefined*	
User Group: Supervisor		
Name	Supervisor	Unique Name of user-group
Level	100	User level
Administrator	TRUE	Defines whether the group has administrator rights
Rights		Access rights
Access right 1	Full	
Access right 2	Full	
Access right 3	Full	
Access right 4	Full	
Access right 5	*Undefined*	

Verwendung der mapp Komponenten

Es werden die Komponenten **MpUserLogin**, **MpUserLoginUI**, **MpUserManagerUI** und **MpUserAccessRight** eingefügt. Danach werden diese, wie im Punkt Verbindungsschema, miteinander verbunden.

Anschließend werden die Komponenten parametriert.

Der **MpUserLogin** erzeugt einen Anmeldepunkt, an dem sich ein Benutzer mit seinem Benutzernamen und seinem Passwort anmeldet.

MpUserLoginUI und **MpUserManagerUI** liefern die entsprechenden Datenpunkte für eine VC4 Visualisierung (z.B. Anmelden, Passwortänderung, Rechteverwaltung usw.) Um über diese Komponenten eine Verbindung zu einer Visualisierung herzustellen siehe Abschnitt VC4 Vorlagen

Loggt sich nun ein Benutzer ein kontrolliert das mapp Benutzermanagement welche Rechte dieser besitzt und setzt dementsprechend bestimmte Funktionen frei. Handelt es sich um einen Administrator ist dieser in der Lage Rechte zu editieren. Mithilfe von **MpUserAccessRight** kann geprüft werden, welches Recht der aktuell eingeloggte Benutzer für eine bestimmte Funktion besitzt.

8.4.3 Editieren von Benutzern und Gruppen über eine Visualisierung

Anforderung

Es soll eine Benutzerverwaltung in eine Applikation implementiert werden. Dabei sollen mehrere Benutzer vorhanden sein mit jeweils verschiedenen Benutzerrechten. Je nachdem welcher Benutzer angemeldet ist sollen bestimmte Funktionen der Maschine verfügbar oder gesperrt sein. Desweitern sollen neue Gruppen und Benutzer über eine Visualisierung erstellt und editiert werden können.

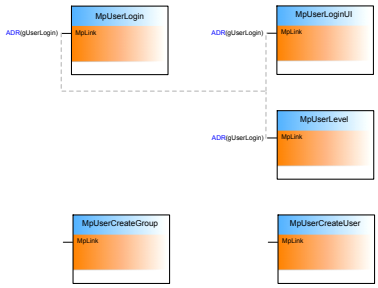
Lösung

Komponentenliste

- ***MpUserCreateGroup*** (MpLink von MpUserLogin): Erstellen einer neuen Gruppe
- ***MpUserCreateUser*** (MpLink von MpUserLogin): Erstellen eines neuen Benutzers
- ***MpUserLogin*** (Eigener MpLink): Anmelden / Abmelden eines Benutzers

- ***MpUserLoginUI*** (MpLink von MpUserLogin): Anmelden / Abmelden eines Benutzers über die Visualisierung
- ***MpUserLevel*** (MpLink von MpUserLogin): Zeigt das Level des angemeldeten Benutzers an

Verbindungsschema



Konfiguration

Es werden die Konfigurationen MpUser und MpUserLogin eingefügt. In der Konfiguration MpUser werden die verschiedenen Benutzergruppen und Benutzer angelegt und deren Rechte vergeben. Die Benutzerrechte sind an bestimmte Funktionen der Maschine verknüpft.

UserMgmt			Settings for user management
General Settings			
Password change inte...	0	s	Interval in which the password must be changed
User expiration time	0	s	Expiration time for the validity of the user account
Password change req...	FALSE		Password change is required on the first user login
Password case	FALSE		Password must have both upper- and lower case letters
Password alphanume...	FALSE		Password must have alpha-numeric characters
Password length	1		Minimum password length
Username length	0		Minimum user-name length
No delete	FALSE		Deletion of users is not allowed
Login attempts	3		Maximum number of failed login attempts until the user is blocked
User groups			
User Group: Administr...			
Name	Administrators		Unique Name of user-group
Level	100		User level
Administrator	TRUE		Defines whether the group has administrator rights
Rights			Access rights
Access right 1	View		
Access right 2	Actuate		
Access right 3	*Undefined*		
User Group: Supervisor			
Name	Supervisor		Unique Name of user-group
Level	50		User level
Administrator	FALSE		Defines whether the group has administrator rights
Rights			Access rights
Access right 1	*Undefined*		
User Group: Operators			
Name	Operators		Unique Name of user-group
Level	10		User level
Administrator	FALSE		Defines whether the group has administrator rights
Rights			Access rights
Access right 1	*Undefined*		

Users		
User: admin		
UserName	admin	Unique identifier (alias) used for the user login
GroupName	Administrators	Group assignment for this user
FullName	Default Admin...	Full (long) name of user
Password	admin	Password
Preferences	/LANG=0 /UNI...	Enter preferences like /LANG=(language-index) or /UNIT=(unit-index)
User: johnd		
UserName	johnd	Unique identifier (alias) used for the user login
GroupName	Supervisor	Group assignment for this user
FullName	John Doe	Full (long) name of user
Password	johnd	Password
Preferences	/LANG=1 /UNI...	Enter preferences like /LANG=(language-index) or /UNIT=(unit-index)
User: janed		
UserName	janed	Unique identifier (alias) used for the user login
GroupName	Operators	Group assignment for this user
FullName	Jane Doe	Full (long) name of user
Password	janed	Password
Preferences	/LANG=0 /UNI...	Enter preferences like /LANG=(language-index) or /UNIT=(unit-index)

Verwendung der mapp Komponenten

Es werden die Komponenten **MpUserLogin**, **MpUserLoginUI**, **MpUserCreateGroup**, **MpUserLevel** und **MpUserCreateUser** eingefügt. Danach werden diese wie im Punkt Verbindungsschema miteinander verbunden.

Anschließend werden die Komponenten parametriert.

Der **MpUserLogin** erzeugt einen Anmeldepunkt, an dem sich ein Benutzer mit seinem Benutzernamen (Identifikation) und seinem Passwort anmeldet.

MpUserLoginUI liefert die entsprechenden Datenpunkte für eine VC4 Visualisierung (z.B. Anmelden, Passwortänderung, usw.) Um durch diese Komponente eine Verbindung zu einer Visualisierung herzustellen siehe Abschnitt VC4 Vorlagen

Loggt sich nun ein Benutzer ein, der die dementsprechenden Rechte besitzt, können neue Gruppen und Benutzer erstellt werden. Dafür werden die Komponenten **MpUserCreateGroup** und **MpUserCreateUser** verwendet. Mithilfe von **MpUserLevel** kann das aktuelle Level des Benutzers angezeigt werden.

8.5 Bibliotheken

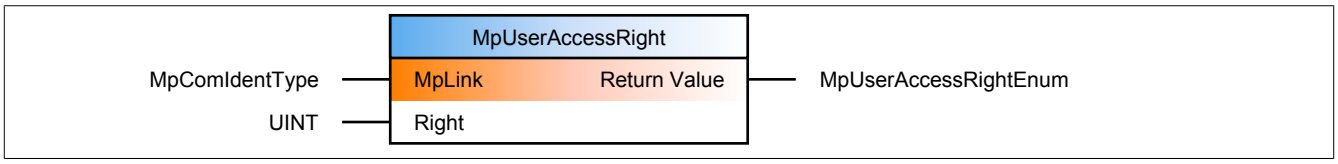
8.5.1 Funktionsbausteine

Unterstützt ab	Name	Beschreibung
5.0.0	MpUserLogin	Dieser Funktionsbaustein ermöglicht das Ein- und Ausloggen von Benutzern (über die Applikation) und er ist für jedes Benutzermanagement notwendig
5.0.0	MpUserManagerUI	Dieser Funktionsbaustein ermöglicht Zugang zum kompletten Benutzermanagement über die Visualisierung
5.0.0	MpUserLoginUI	Dieser Funktionsbaustein ermöglicht das Ein- und Ausloggen von Benutzern (über die Visualisierung)
5.0.0	MpUserCreateUser	Diese Funktion dient zum Hinzufügen eines Benutzers (über die Applikation)
5.0.0	MpUserCreateGroup	Diese Funktion dient zum Hinzufügen einer Gruppe (über die Applikation)
5.0.0	MpUserLevel	Diese Funktion gibt das Level des aktuell eingeloggten Benutzers zurück
5.0.0	MpUserChangePassword	Mithilfe dieser Funktion kann das Passwort eines Benutzers über die Applikation verändert werden.
5.0.0	MpUserCheckPassword	Diese Funktion prüft, ob Benutzername und Passwort übereinstimmen und gültig sind
5.0.0	MpUserAccessRight	Diese Funktion gibt die Rechte des aktuell eingeloggten Benutzers zurück
5.0.0	MpUserConfig	Dieser Funktionsbaustein speichert/lädt die Konfiguration des Benutzermanagements
5.0.0	MpUserLoginConfig	Dieser Funktionsbaustein speichert/lädt die Konfiguration des Login- und Logoutverhaltens
5.0.0	MpUserSignature	Dieser Funktionsbaustein prüft, ob das Zeichnen anhand einer elektronischen Signatur erfolgreich war
5.0.0	MpUserSignatureUI	Dieser Funktionsbaustein ermöglicht eine elektronische Signatur über die Visualisierung
5.2.0	MpUserGetData	Mithilfe dieser Funktion können zusätzliche Informationen eines Benutzers abgefragt werden.

8.5.1.1 MpUserAccessRight

Diese Funktion gibt die Rechte des aktuell eingeloggten Benutzers zurück.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserLogin Konfiguration).
IN	Right	`UINT`	Funktion, für welche das Recht überprüft werden soll.
OUT	ReturnValue	`MpUserAccess-RightEnum`	Benutzerrecht für die hinterfragte Funktion.

mapp Konzept

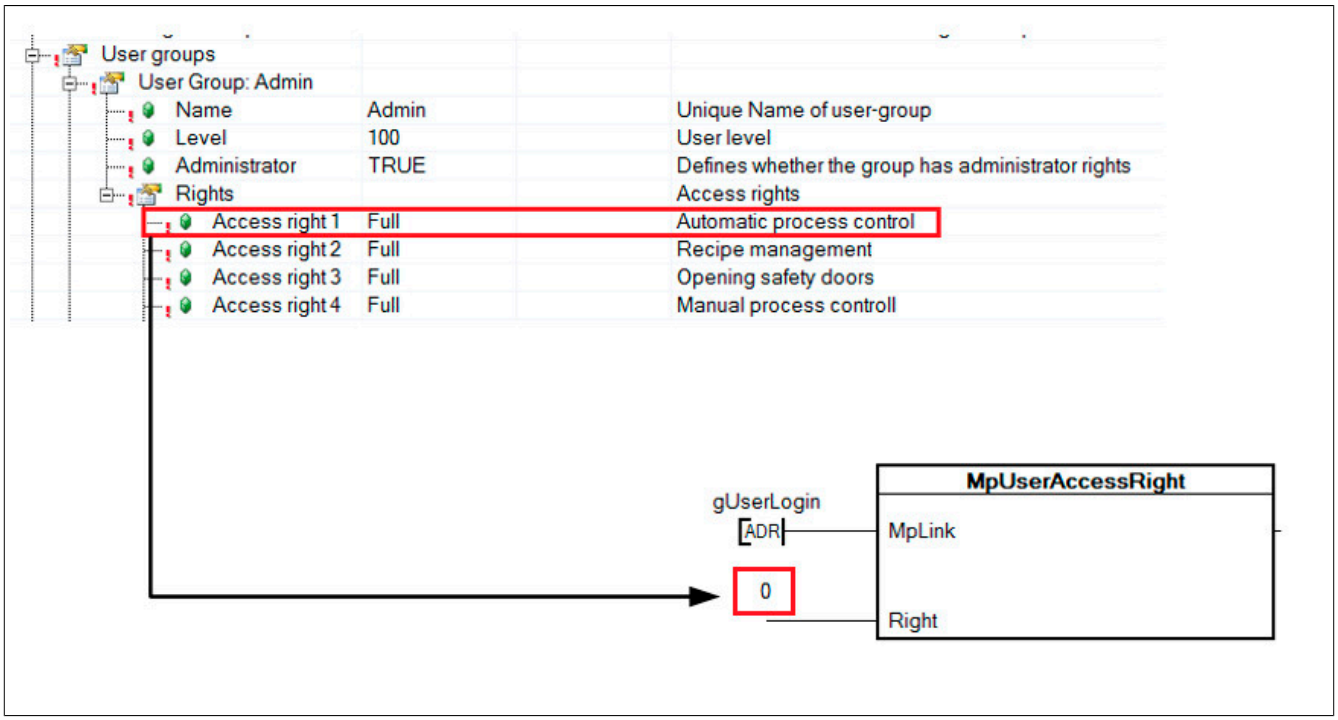
Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.1.1 Beschreibung

Für diese Komponente wird derselbe MpLink verwendet, wie für MpUserLogin.

Über den Parameter "Right" wird die Funktion (siehe Rechte, Levels und Funktionen) angegeben, für welche das Recht überprüft werden soll.

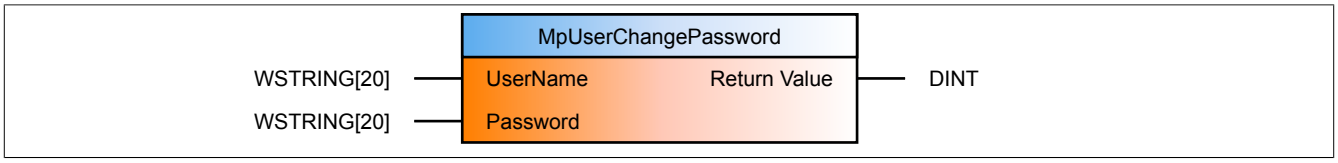
Es ist zu beachten, dass der Eingang "Right" mit dem Index 0 beginnt, aber in der MpUser Konfiguration die Rechtevergabe ab Index 1 beginnt. Soll also das Recht "Automatic process control" geprüft werden, welches in der Konfiguration Recht 1 ist, so muss am Parameter "Right" zur Überprüfung der Index 0 angegeben werden.



8.5.1.2 MpUserChangePassword

Mithilfe dieser Funktion kann das Passwort eines Benutzers über die Applikation verändert werden.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	UserName	`WSTRING[20]`	Benutzer, für welchen das Passwort angepasst werden soll.
IN	Password	`WSTRING[20]`	Neues Passwort.
OUT	ReturnValue	`DINT`	Status Information der Funktion.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

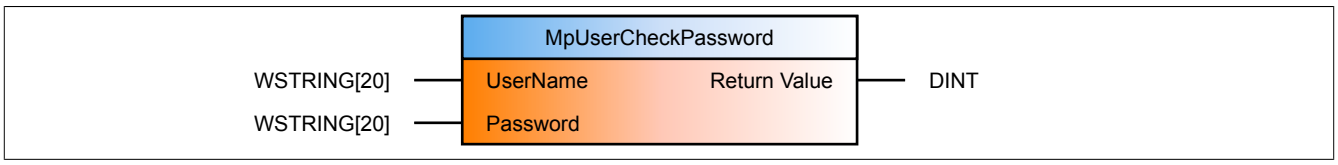
8.5.1.2.1 Beschreibung

Über "UserName" wird definiert für welchen Benutzer das Passwort verändert werden soll. Das neue Passwort wird über "Password" an die Funktion übergeben. Ein Ereignis wird generiert, welches über MpAudit ausgewertet werden kann.

8.5.1.3 MpUserCheckPassword

Diese Funktion prüft, ob Benutzername und Passwort übereinstimmen und gültig sind.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	UserName	`WSTRING[20]`	Benutzername, für welchen das Passwort geprüft werden soll.
IN	Password	`WSTRING[20]`	Zu überprüfendes Passwort.
OUT	ReturnValue	`DINT`	Status Information der Funktion.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.3.1 Beschreibung

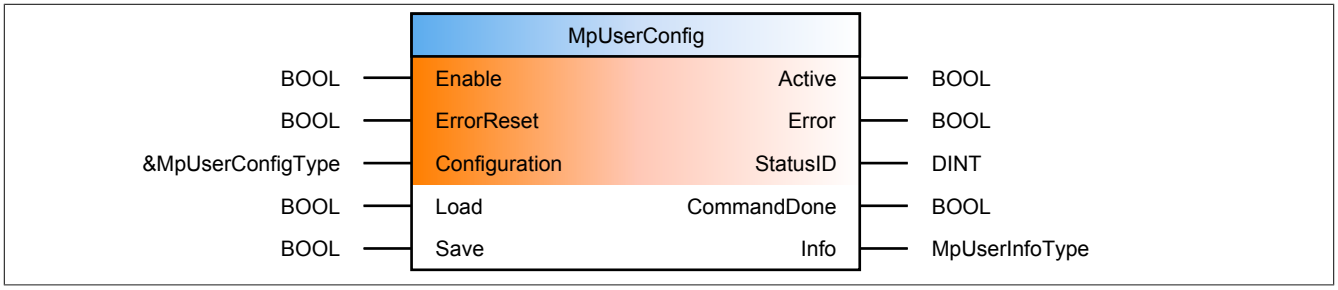
Über den Rückgabewert wird angezeigt, ob der angegebene Benutzername und das angegebene Passwort übereinstimmen.

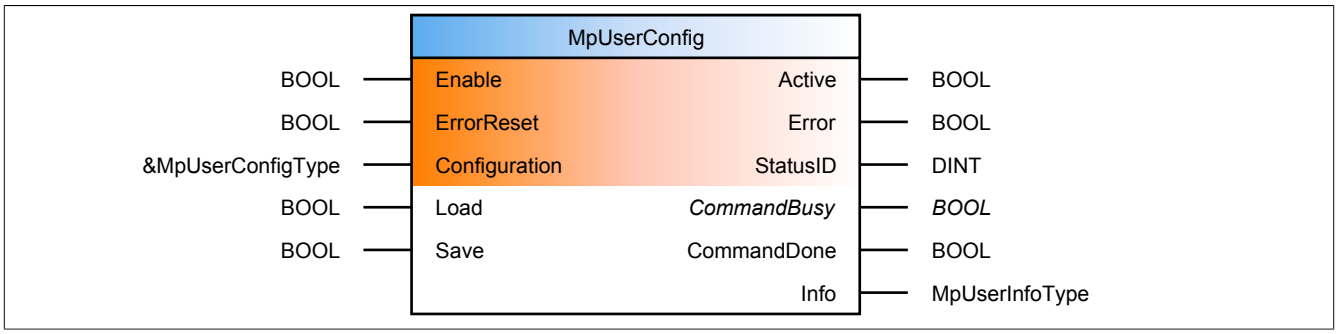
8.5.1.4 MpUserConfig

Dieser Funktionsbaustein speichert/lädt die Konfiguration des Benutzermanagements.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	Enable	BOOL	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	BOOL	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Configuration	Zeiger auf MpUserConfigType	Struktur, in der die Konfiguration festgelegt wird.
IN	Load	BOOL	Lädt die Konfiguration der Komponente.
IN	Save	BOOL	Speichert die Konfiguration der Komponente.
OUT	Active	BOOL	FB ist aktiv.
OUT	Error	BOOL	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	DINT	Statusinformation.
OUT	*CommandBusy*	BOOL	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	BOOL	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	MpUserInfoType	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

Weiters wird im Abschnitt Konfigurieren statt Programmieren erklärt, wie man mit der Konfiguration arbeitet.

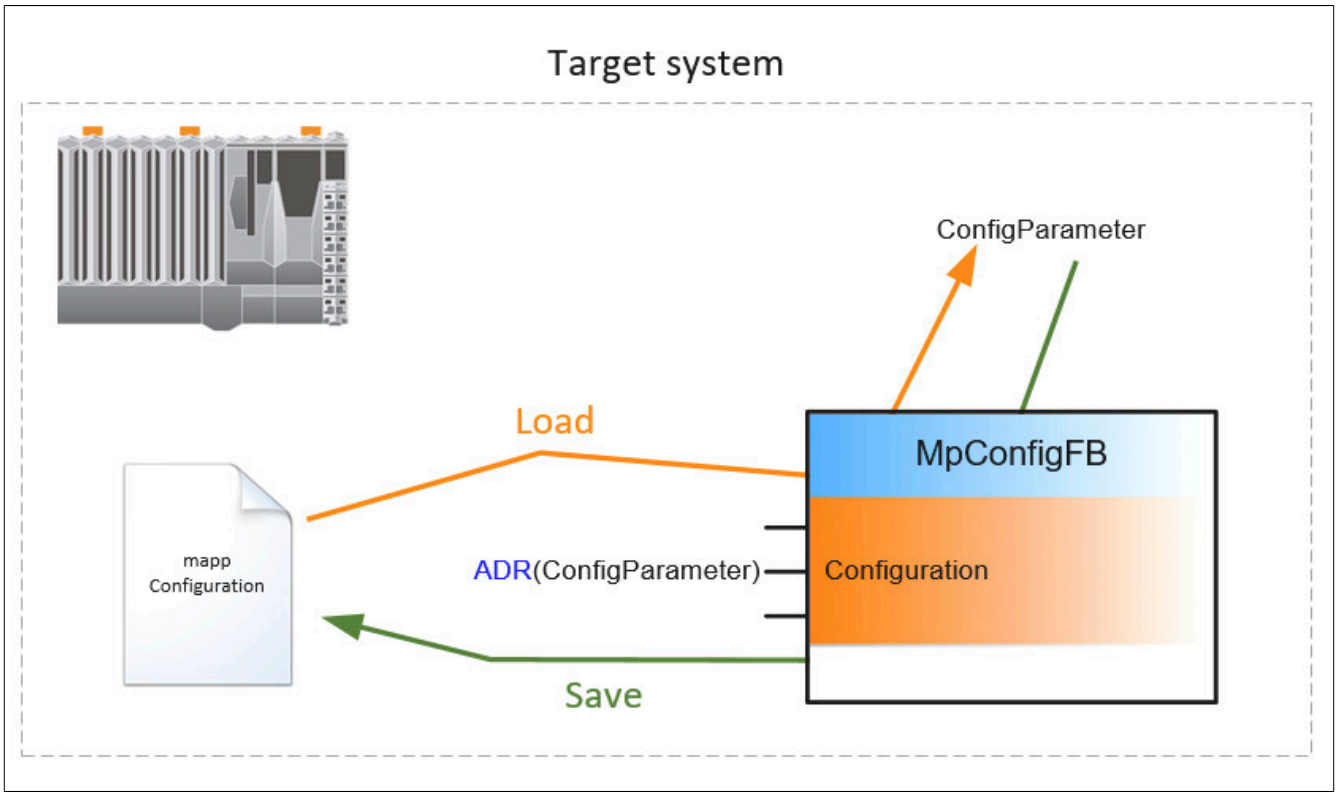
8.5.1.4.1 Beschreibung

Mithilfe des Funktionsbausteins sind Änderungen der Konfiguration zur Laufzeit möglich.

Der Parameter "Configuration" ermöglicht es die Konfiguration, bzw. einen Teil der Konfiguration zu bearbeiten.

Der Befehl "Load" wird verwendet, um die aktuell konfigurierten Parameter auszulesen. Dabei wird die aktuell aktive Konfiguration der Steuerung in die PV am Eingang "Configuration" geschrieben. Die Daten in der Struktur werden dabei überschrieben. Danach können die Werte beliebig editiert werden.

Über den Befehl "Save" werden die Werte aus der "Configuration" Struktur auf die aktive Konfiguration übernommen und, wenn möglich, direkt verwendet.



Um die Konfigurations-Parameter in der Appliaktion korrekt anzupassen, wird folgendermaßen vorgegangen:

1. Befehl "Load = `TRUE`"

Die aktuelle Konfiguration wird vom Zielsystem in die Konfigurations-Struktur geladen. Der Vorgang ist mit "CommandDone = TRUE" abgeschlossen.

2. Befehl "Load = `FALSE`"

Rücksetzen des "Load" Befehls. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

3. Parametrieren der Konfigurationsstruktur

Die Parameter der Konfigurationsstruktur können nun wie gewünscht angepasst werden.

4. Befehl "Save = `TRUE`" setzen

Die geänderte Konfigurationsstruktur wird auf das Zielsystem übertragen. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

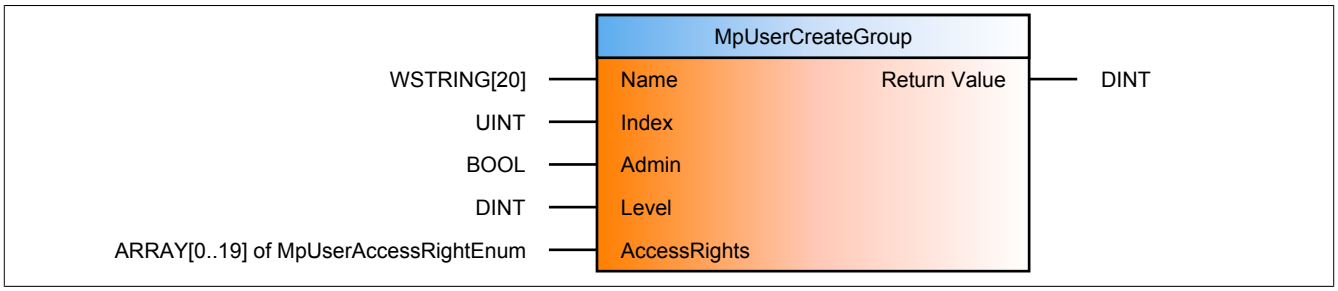
5. Befehl "Save = `FALSE`" setzen

Rücksetzen des "Save" Befehls. Der Vorgang ist mit "CommandDone = `FALSE`" abgeschlossen. Um zu kontrollieren ob die Parameter erfolgreich geändert worden sind, können die Schritte 1 und 2 erneut ausgeführt werden.

8.5.1.5 MpUserCreateGroup

Diese Funktion dient zum Hinzufügen einer Gruppe (über die Applikation).

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	Name	WSTRING[20]	Name der Benutzergruppe.
IN	Index	UINT	Index der Benutzergruppe (muss in aufeinanderfolgender Reihenfolge sein).
IN	Admin	BOOL	Legt fest, ob die Gruppe Adminrechte besitzt.
IN	Level	DINT	Level der Benutzergruppe.
IN	AccessRights	ARRAY[0..19] von MpUserAccess-RightEnum	Liste, welche die Rechte für einzelne Funktionen definiert.
OUT	ReturnValue	DINT	Gibt Status Information zurück.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.5.1 Beschreibung

Mithilfe dieser Funktion kann eine neue Benutzergruppe ins Benutzermanagement hinzugefügt werden. Die Rechte für Maschinenfunktionen werden über den Eingang "AccessRights" vergeben. Jede Funktions ist ein Element in einem Array. Die Größe dieses Arrays entspricht der Anzahl an möglichen Funktionen und erreicht maximal 20. Für jede Funktion muss ein Recht vergeben werden. Die Rechte werden über die Enumeration MpUserAccess-RightEnum definiert. Folgende Tabelle zeigt eine mögliche Benutzergruppe:

Funktion	Rezeptverwaltung	Prozesssteuerung	Alarmsteuerung	Prozessparameter	Benutzermanage-ment
Element des Arrays	0	1	2	3	4
Recht	mpUSER_AC-CESS_VIEW	mpUSER_ACCESS_AC-TUATE	mpUSER_AC-CESS_VIEW	mpUSER_AC-CESS_VIEW	mpUSER_AC-CESS_NONE

In dieser Tabelle wird ein Array mit fünf Elementen des Typs MpUserAccessRightEnum gezeigt. Es wird eine Benutzergruppe erstellt, die für die obigen Funktionen unterschiedliche Rechte besitzt. Welches Element für welche Funktion gilt, muss definiert werden und kann beispielsweise über eine Textgruppe angezeigt werden.

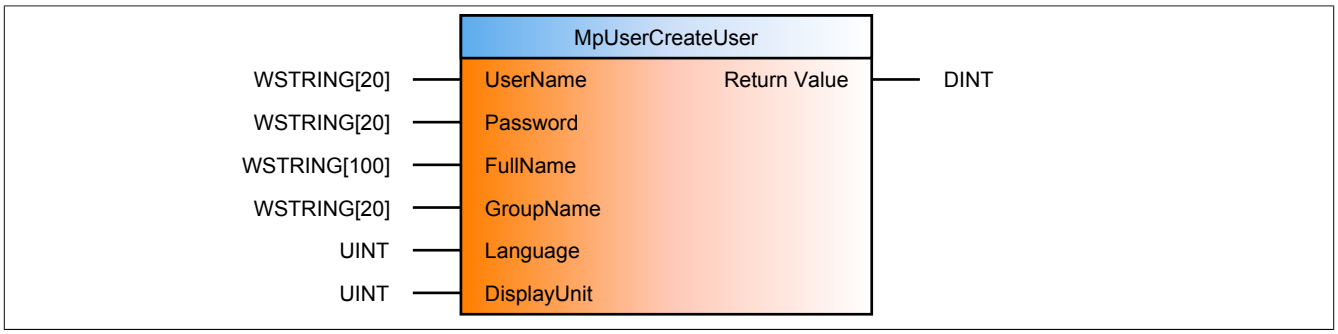
Die Benutzergruppe muss in der INIT Routine hinzugefügt werden. Beim Aufruf der Funktion werden, falls die Benutzergruppe bereits exisitiert, alle Benutzergruppendaten überschrieben. Dafür müssen der Gruppenindex und der Gruppenname richtig angegeben werden. Sind Gruppenname und Gruppenindex neu, so wird eine neue Gruppe mit den angegebenen Daten erstellt. Soabld die Bernutzergruppendaten einmal über die Visualisierung verändert wurden, werden die Daten in der INIT Routine nicht mehr überschrieben.

> **Wird über Automation Studio in der Konfiguration des Benutzermanagements eine Benutzergruppe angelegt, so können mehr als 20 Rechte definiert werden. Diese sind dann nicht mithilfe von MpUser-ManagerUI einsehbar. Die Strukturen MpUserMgrUIGroupInfoType und MpUserMgrUIGroupDlgType sind auf 20 Rechte limitiert. Ungeachtet dessen können die Rechte über MpUserAccessRight abgefragt werden.**

8.5.1.6 MpUserCreateUser

Diese Funktion dient zum Hinzufügen eines Benutzers (über die Applikation).

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	UserName	WSTRING[20]	Benutzername fürs Einloggen.
IN	Password	WSTRING[20]	Passwort fürs Einloggen.
IN	FullName	WSTRING[100]	Vollständiger Name.

I/O	Parameter	Datentyp	Beschreibung
IN	GroupName	`WSTRING[20]`	Gruppe, welcher der Benutzer zugeordnet sein soll.
IN	Language	`UINT`	Bevorzugte Sprache.
IN	DisplayUnit	`UINT`	Bevorzugtes Einheitensystem.
OUT	ReturnValue	`DINT`	Gibt Status Information zurück.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.6.1 Beschreibung

Mithilfe dieser Funktion kann ein neuer initialer Benutzer ins Benutzermanagement hinzugefügt werden. Die Werte für Sprache und Einheitensystem korrespondieren mit den Werten in der VC4 Visualisierung.

Der Benutzer muss in der INIT Routine hinzugefügt werden. Beim Aufruf der Funktion werden, falls der Benutzer bereits exisitiert, alle Benutzerdaten außer der Benutzername ("UserName") überschrieben. Soabld die Bernutzerdaten einmal über die Visualisierung verändert wurden, werden die Daten in der INIT Routine nicht mehr überschrieben.

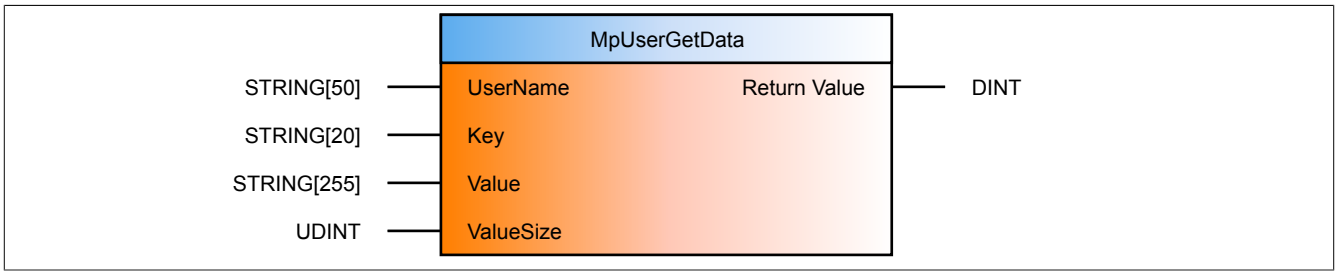
Es gilt zu beachten, dass der Benutzer zur Laufzeit nicht gelöscht sondern nur deaktiviert werden kann.

Benutzer welche zur Laufzeit hinzugefügt werden sollen, werden mithilfe von MpUserManagerUI erstellt. Diese Benutzer können zur Laufzeit auch gelöscht werden.

8.5.1.7 MpUserGetData

Mithilfe dieser Funktion können zusätzliche Informationen eines Benutzers abgefragt werden.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	UserName	`STRING[50]`	Definiert den Benutzernamen für welchen man zusätzliche Information abfragen möchte.
IN	Key	`SRING[20]`	Durch "Key" wird festgelegt, welche Information man vom Benutzer abfragen möchte
IN	Value	`SRING[255]`	Hier wird die abgefragte Information angezeigt.
IN	ValueSize	`UDINT`	Definiert die Buffergröße für den Schreibvorgang an "Value".
OUT	ReturnValue	`DINT`	Rückgabewert der Funktion.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.7.1 Beschreibung

Über den Parameter "UserName" wird der Name eines Benutzers angegeben, für welchen man zusätzliche Information abfragen möchte.

"Key" bestimmt welche Information abgefragt werden soll. Folgende Schlüsselwörter sind möglich:

- **Language:** Liefert die bevorzugte Sprache des Benutzers.
- **DisplayUnit:** Liefert das bevorzugte Einheitensystem des Benutzers.
- **eMail:** Wird "eMail" verwendet, wird die E-Mail-Adresse des Benutzers angezeigt.
- **PhoneNumber:** Durch "telNr" wird die Telefonnummer des Benutzers angezeigt.
- **any:** Durch "any" kann eine, in der Konfiguration festgelegte, zusätzliche Information angezeigt werden, wie zum Beispiel "Team".

Am Parameter "Value" wird die abgefragte Information angezeigt.

Wichtig hierbei ist, dass über "ValueSize" die Buffergröße für den Schreibvorgang an "Value" festgelegt wird.

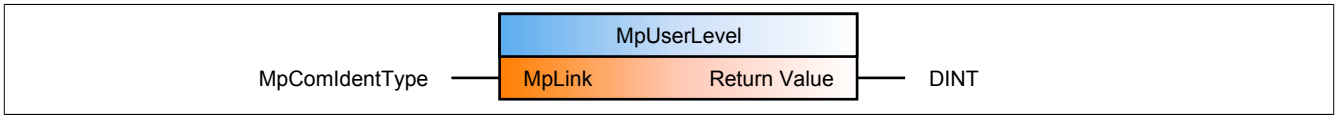
"Return value" liefert den Rückgabewert der Funktion. Dabei können folgende Rückgabewerte auftreten:

- mpUSER_ERR_NO_USERNAME
- mpUSER_ERR_PAR_NULL
- mpUSER_ERR_EMPTY_KEY
- mpUSER_ERR_INFO_NOT_FOUND
- mpUSER_ERR_BUFFER_TOO_SMALL
- mpUSER_WRN_KEY_DUPLICATE

8.5.1.8 MpUserLevel

Diese Funktion gibt das Level des aktuell eingeloggten Benutzers zurück.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserLogin Konfiguration).
OUT	ReturnValue	`DINT`	Gibt das Level des eingeloggten Benutzers zurück.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.8.1 Beschreibung

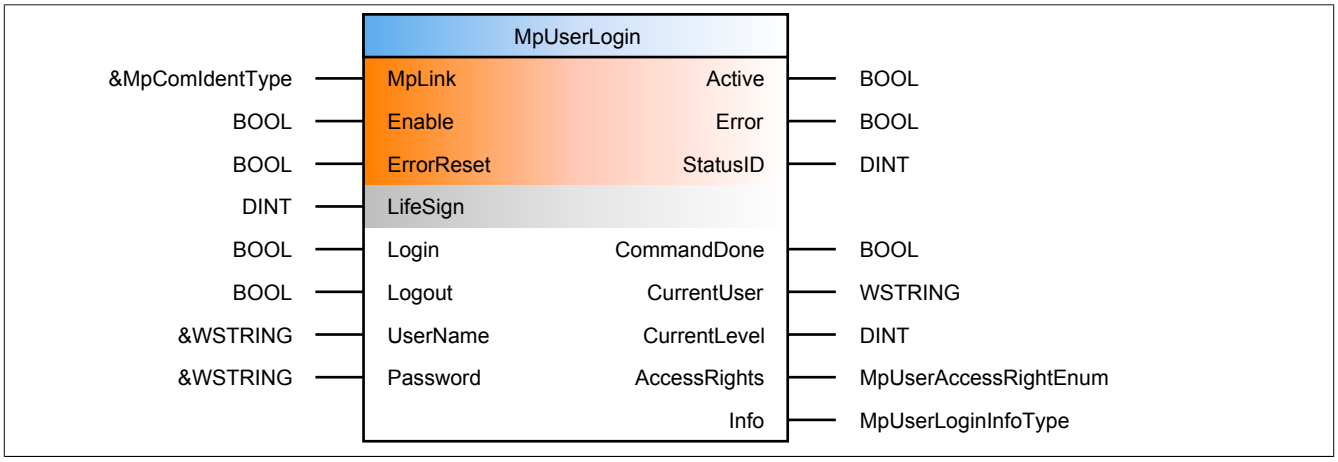
Für diese Komponente wird derselbe MpLink verwendet, wie für MpUserLogin.

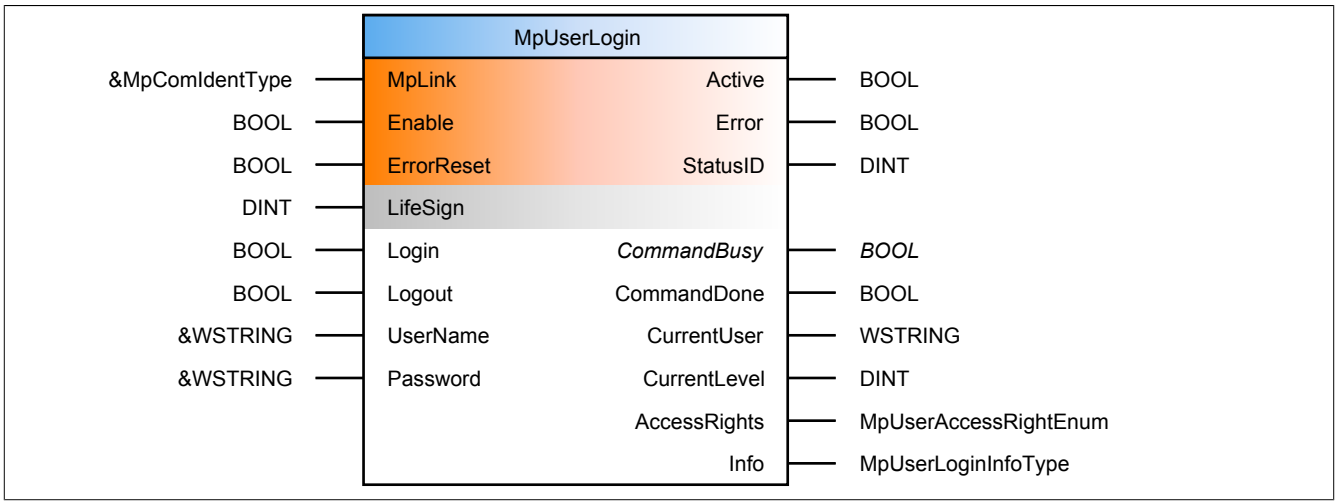
8.5.1.9 MpUserLogin

Dieser Funktionsbaustein ermöglicht das Ein- und Ausloggen von Benutzern (über die Applikation) und er ist für jedes Benutzermanagement notwendig.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserLogin Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	LifeSign	`DINT`	Damit wird festgestellt, ob der eingeloggte Benutzer seit längerer Zeit inaktiv ist. Inaktivzeit wird bei Veränderung dieses Eingangs zurückgesetzt.
IN	Login	`BOOL`	Befehl, um sich einzuloggen.
IN	Logout	`BOOL`	Befehl, um sich auszuloggen.
IN	UserName	`Zeiger auf WSTRING[20]`	Benutzername, mit dem sich eingeloggt werden soll.
IN	Password	`Zeiger auf WSTRING[20]`	Passwort, mit dem sich eingeloggt werden soll.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Zeigt an, dass der Funktionsbaustein im Fehlerzustand ist, oder dass ein Befehl nicht richtig ausgeführt wurde.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	CurrentUser	`WSTRING[40]`	Benutzername des eingeloggten Benutzers.
OUT	CurrentLevel	`DINT`	Level des eingeloggten Benutzers.
OUT	AccessRights	`ARRAY[0..19] von MpUserAccessRightEnum`	Zugriffsrechte des eingeloggten Benutzers.
OUT	Info	`MpUserLoginInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.9.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpUserLogin Konfiguration verwendet.

Dieser Funktionsbaustein wird für **jedes** Benutzermanagement benötigt. Er verwaltet die verschiedenen Benutzergruppen und Benutzer im System. Über die Eingänge "UserName" und "Password" kann man sich einloggen über die Applikation. Dazu werden die Befehle "Login" und "Logout" verwendet. Falls erwünscht, kann der Eingang "LifeSign" verwendet werden, um zu prüfen, ob der Benutzer noch aktiv ist.

Es kann nur ein Benutzer eingeloggt sein. Falls erforderlich können über eine Hierarchie weitere Loginpunkte erstellt werden. Für jede mechatronische Einheit, kann ein Benutzer eingeloggt sein. Dabei werden jedoch die Benutzerdaten, sowie die Benutzergruppendaten zentral abgelegt und sind für alle Loginpunkte dieselben. Eine Hierarchie für mechatronische Einheiten, wird über MpComLinkToParent erzeugt.

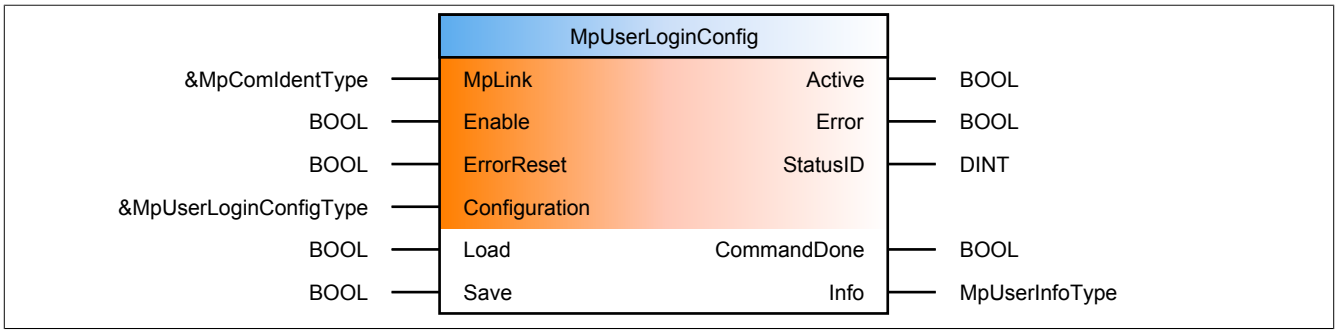
An den Ausgängen "CurrentUser" und CurrentLevel" kann der aktuelle Benutzer identifiziert werden, sowie sein Zugriffslevel. Außerdem sind seine Rechte für verschiedene Funktionen in dem Array "AccessRights" zu finden.

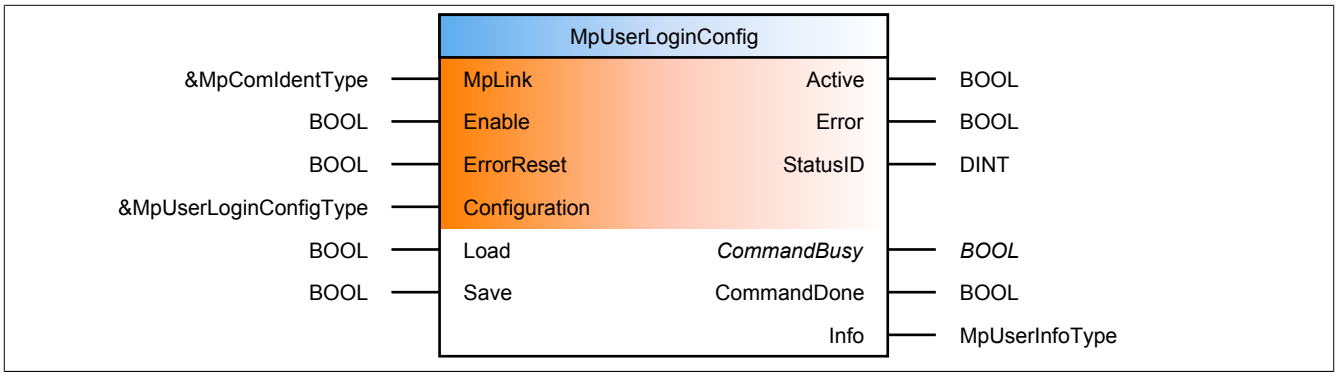
8.5.1.10 MpUserLoginConfig

Dieser Funktionsbaustein speichert/lädt die Konfiguration des Login- und Logoutverhaltens.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserLogin Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Configuration	`Zeiger auf MpUserLoginConfigType`	Struktur, in der die Konfiguration festgelegt wird.
IN	Load	`BOOL`	Lädt die Konfiguration der Komponente.
IN	Save	`BOOL`	Speichert die Konfiguration der Komponente.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpUserInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

Weiters wird im Abschnitt Konfigurieren statt Programmieren erklärt, wie man mit der Konfiguration arbeitet.

8.5.1.10.1 Beschreibung

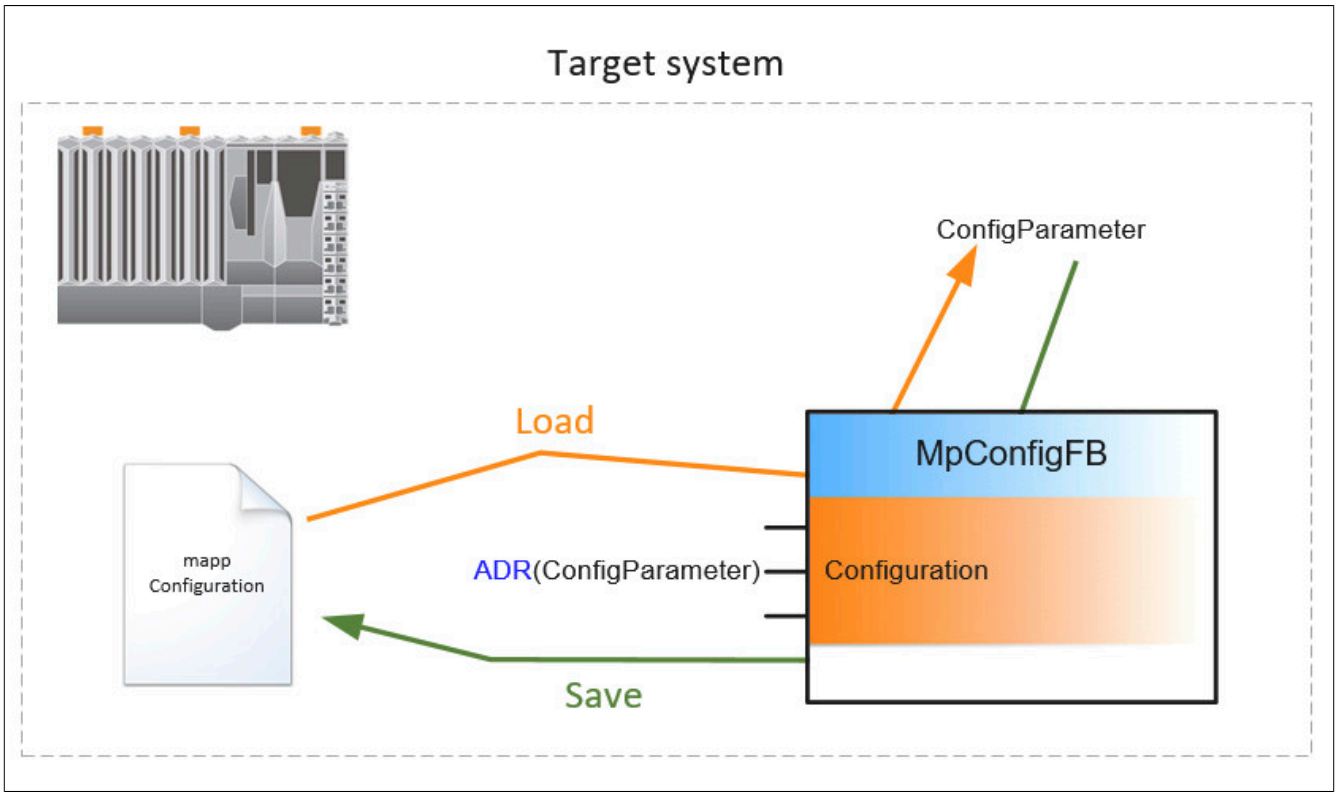
Für diesen Funktionsbaustein wird derselbe MpLink verwendet, wie für MpUserLogin. Für eine große Maschine an der zum Beispiel 3 Personen gleichzeitig an unterschiedlichen Stationen arbeiten können, kann über diesen Funktionsbaustein eine der Stationen konfiguriert werden.

Mithilfe des Funktionsbausteins sind Änderungen der Konfiguration zur Laufzeit möglich.

Der Parameter "Configuration" ermöglicht es die Konfiguration, bzw. einen Teil der Konfiguration zu bearbeiten.

Der Befehl "Load" wird verwendet, um die aktuell konfigurierten Parameter auszulesen. Dabei wird die aktuell aktive Konfiguration der Steuerung in die PV am Eingang "Configuration" geschrieben. Die Daten in der Struktur werden dabei überschrieben. Danach können die Werte beliebig editiert werden.

Über den Befehl "Save" werden die Werte aus der "Configuration" Struktur auf die aktive Konfiguration übernommen und, wenn möglich, direkt verwendet.



Um die Konfigurations-Parameter in der Appliaktion korrekt anzupassen, wird folgendermaßen vorgegangen:

1. Befehl "Load = `TRUE`"

Die aktuelle Konfiguration wird vom Zielsystem in die Konfigurations-Struktur geladen. Der Vorgang ist mit "CommandDone = TRUE" abgeschlossen.

2. Befehl "Load = `FALSE`"

Rücksetzen des "Load" Befehls. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

3. Parametrieren der Konfigurationsstruktur

Die Parameter der Konfigurationsstruktur können nun wie gewünscht angepasst werden.

4. Befehl "Save = `TRUE`" setzen

Die geänderte Konfigurationsstruktur wird auf das Zielsystem übertragen. Der Vorgang ist mit "CommandDone = `TRUE`" abgeschlossen.

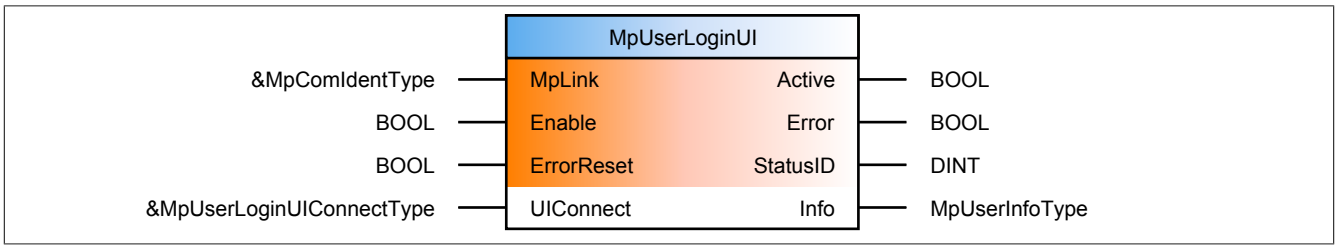
5. Befehl "Save = `FALSE`" setzen

Rücksetzen des "Save" Befehls. Der Vorgang ist mit "CommandDone = `FALSE`" abgeschlossen. Um zu kontrollieren ob die Parameter erfolgreich geändert worden sind, können die Schritte 1 und 2 erneut ausgeführt werden.

8.5.1.11 MpUserLoginUI

Dieser Funktionsbaustein ermöglicht das Ein- und Ausloggen von Benutzern (über die Visualisierung).

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserLogin Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	UIConnect	`Zeiger auf MpUserLoginUIConnectType`	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpUserInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

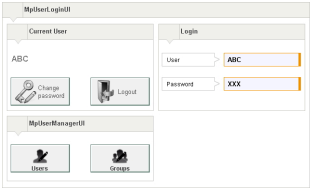
Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.11.1 Beschreibung

Für diese Komponente wird derselbe MpLink verwendet, wie für MpUserLogin.

Diese Komponente stellt eine Verbindung zwischen Benutzerlogin und einer Visual Components 4 Visualisierung dar. Der Austausch der Daten zwischen MpUserLoginUI und der Visualisierung erfolgt über die Struktur MpUserLoginUIConnectType.

Pro Konfiguration kann nur ein MpUserLoginUI verwendet werden. Es ist nicht möglich mehrere MpUserLoginUI-Instanzen zu verwenden.



MpUserLoginUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

- **Status:** Der "Status" Parameter gibt Auskunft über die aktuelle Aktivität der Benutzerverwaltung
- **CurrentUser, DisplayUnit, Language:** Auskunft über aktuell eingeloggten Benutzer und seine bevorzugten Einstellungen in Bezug auf Sprache und Einheitensystem
- **LoggedIn:** Zeigt an ob aktuell ein Benutzer eubgeloggt ist
- **Login:** Über die Struktur MpUserLoginUILoginType sich ein Benutzer ein- und ausloggen kann
- **ChangePassword:** Über die Struktur MpUserLoginUIPwdType kann man das Passwort ändern
- **MessageBox:** In der Struktur MpUserUIMessageBoxType können die unterschiedlichen Dialogfenster für MpUser konfiguriert werden
- **DefaultLayerStatus:** Solange eines der Dialogfenster aktiv ist, kann der dahinterliegende Layer über diesen Datenpunkt gesperrt werden

Fehlernummern

Tritt bei einem Login-Versuch ein Fehler auf, da beispielsweise das Passwort falsch angegeben wurde, so wird der Fehler in der MessageBox-Struktur MpUserUIMessageBoxType unter "ErrorNumber" angezeigt. Folgende Fehler können auftreten:

- 24576: Passwort ungültig
- 24577: Ungültige Konfiguration
- 24578: Benutzer exisitiert bereits
- 24579: Login Token ungültig
- 24582: Benutzergruppe existiert nicht
- 24583: Passwort nicht sicher genug
- 24584: Benutzer gesperrt
- 24585: Passwortänderung erforderlich

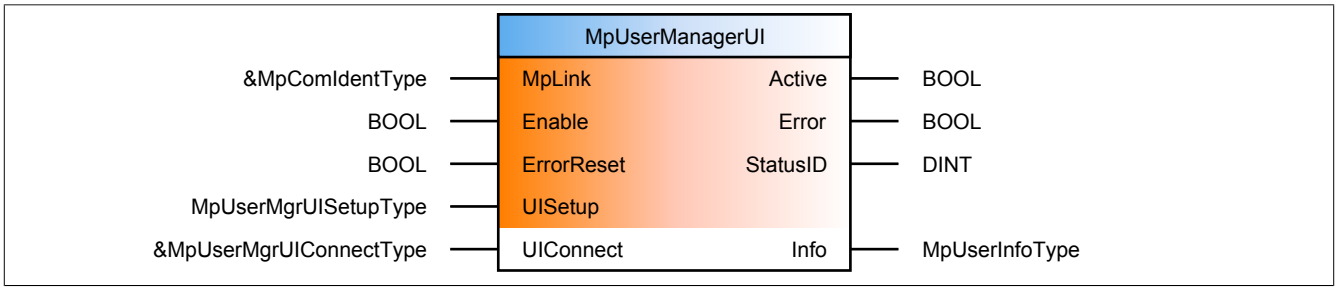
- 24586: Zuviele Login Versuche, Login nicht mehr möglich
- 24589: Kein Benutzer eingeloggt
- 24591: Passwörter identisch
- 24592: Benutzername existiert bereits
- 24597: Benutzer bereits eingeloggt
- 24598: Fehlender Wert an UIConnect
- 24601: MpUserLogin fehlt
- 24602: Passwort zum Bestätigen falsch
- 24616: Passwort wurde bereits verwendet

8.5.1.12 MpUserManagerUI

Dieser Funktionsbaustein ermöglicht Zugang zum kompletten Benutzermanagement über die Visualisierung.

> **Information:**
>
> **Es ist nötig die Bibliothek MpFile auf die Steuerung zu übertragen, um bei der "Import" Funktion dieses Funktionsbausteins eine Liste mit möglichen Dateien anzuzeigen.**

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserLogin Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	UISetup	`MpUserMgrUISetupType`	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
IN	UIConnect	`Zeiger auf MpUserMgrUIConnectType`	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpUserInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

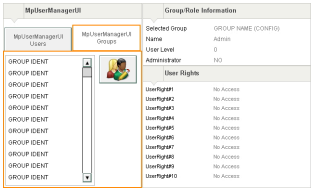
Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

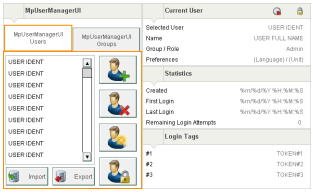
8.5.1.12.1 Beschreibung

Für diese Komponente wird derselbe MpLink verwendet, wie für MpUserLogin.

Diese Komponente stellt eine Verbindung zwischen Benutzermanagement und einer Visual Components 4 Visualisierung dar. Der Austausch der Daten zwischen MpUserManagerUI und der Visualisierung erfolgt über die Struktur MpUserMgrUIConnectType.

Pro MpUserLogin kann nur ein MpUserManagerUI verwendet werden. Es ist nicht möglich mehrere MpUserManagerUI-Instanzen zu verwenden.





MpUserMgrUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

- **Status:** Der "Status" Parameter gibt Auskunft über die aktuelle Aktivität der Benutzerverwaltung
- **User:** In der Struktur MpUserMgrUIUserType sind die Benutzerdaten aufgelegt. In einer Liste können genaue Informationen über einzelne Benutzer abgerufen werden. Es können auch beliebig viele neue Benutzer hinzugefügt werden, bzw. bereits bestehende editiert werden. Bei der Anzahl an neu erstellbaren Nutzern gibt es keine Limitierung.
- **Group:** Über die Struktur MpUserMgrUIGroupType wird eine Liste mit allen Benutzergruppen und deren Berechtigungen angezeigt. Die Benutzergruppen können darin auch editiert werden. Bei der Anzahl an neu erstellbaren Gruppen gibt es keine Limitierung.
- **Export:** Ein Export wird mithilfe der Struktur MpUserMgrUIExportType durchgeführt. Über einen Dialog wird bestimmt, welche Datei exportiert werden soll. Exportiert werden alle Benutzer und Benutzergruppen.
- **Import:** Ein Import wird mithilfe der Struktur MpUserMgrUIImportType durchgeführt. Über einen Dialog wird bestimmt, welche Datei importiert werden soll. Es können in der MpUser Konfiguration verschiedene Import Modi unter "User Import Mode" und "Group Import Mode" ausgewählt werden.
- **MessageBox:** In der Struktur MpUserUIMessageBoxType können die unterschiedlichen Dialogfenster für MpUser konfiguriert werden
- **DefaultLayerStatus:** Solange eines der Dialogfenster aktiv ist, kann der dahinterliegende Layer über diesen Datenpunkt gesperrt werden

> **Es ist nötig die Bibliothek MpFile auf die Steuerung zu übertragen, um bei der "Import" Funktion dieses Funktionsbausteins eine Liste mit möglichen Dateien anzuzeigen.**

MpUserMgrUISetupType

In dieser Struktur wird über "UserListSize" definiert, wieviele Einträge der Benutzerliste auf einer Seite der Visualisierung angezeigt werden. Über "ScrollWindow" wird entscheiden wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden. Über "FileDevice" wird das Speichermedium ausgewählt, auf welchem die Benutzerdaten zum Import liegen, bzw. wohin sie exportiert werden sollen. Über "FileExtension" wird die Endung für Exportdateien festgelegt, bzw. werden in der Importliste nur Dateien mit dieser Endung angezeigt.

Über "ConfirmOverwriteUser" und "ConfirmOverwriteGroup" kann festgelegt werden ob das Überschreiben von Benutzern/Benutzergruppen über einen Dialog bestätigt werden soll.

Fehlernummern

Tritt bei einem Login-Versuch ein Fehler auf, da beispielsweise das Passwort falsch angegeben wurde, so wird der Fehler in der MessageBox-Struktur MpUserUIMessageBoxType unter "ErrorNumber" angezeigt. Folgende Fehler können auftreten:

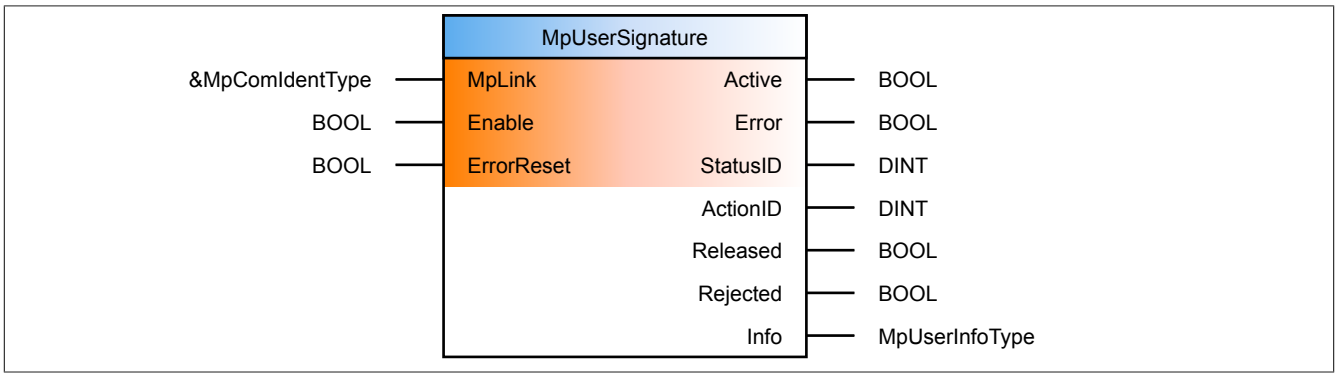
- 24577: Ungültige Konfiguration
- 24578: Benutzer exisitiert bereits
- 24579: Login Token ungültig
- 24580Keine Administratorrechte
- 24581: Ungenügende Benutzerrechte
- 24582: Benutzergruppe existiert nicht
- 24583: Passwort nicht sicher genug

- 24584: Benutzer gesperrt
- 24585: Passwortänderung erforderlich
- 24586: Zuviele Login Versuche, Login nicht mehr möglich
- 24587: Benutzer kat keine Adminrechte
- 24589: Kein Benutzer eingeloggt
- 24591: Passwörter identisch
- 24592: Benutzername existiert bereits
- 24595: Benutzergruppenindex ist ungültig
- 24597: Benutzer bereits eingeloggt
- 24598: Fehlender Wert an UIConnect
- 24599: Ungültiger Benutzername
- 24601: MpUserLogin fehlt
- 24602: Passwort zum Bestätigen falsch
- 24603: Angegebenes Systemrecht existiert nicht
- 24604: Import fehlgeschlagen
- 24605: Fehler beim Laden der Importdatei
- 24606: Fehler beim Speichern der Exportdatei
- 24611: Name des Benutzers ist zu kurz
- 24612: Benutzergruppe exisitiert bereits
- 24613: Benutzergruppe kann nicht gelöscht werden
- 24616: Passwort wurde bereits verwendet
- 24617: Ungültige Import Datei
- 24618: Key wird bereits verwendet
- 24619: Information nicht gefunden

8.5.1.13 MpUserSignature

Dieser Funktionsbaustein prüft, ob das Zeichnen anhand einer elektronischen Signatur erfolgreich war.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserSign Konfiguration)
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	ActionID	`DINT`	Angabe der Aktion, welche signiert werden soll.
OUT	Released	`BOOL`	Wird für einen Zyklus auf `TRUE` gesetzt, wenn die elektronische Signatur erfolgreich durchgeführt wurde.
OUT	Rejected	`BOOL`	Wird für einen Zyklus auf `TRUE` gesetzt, wenn die elektronische Signatur **nicht** erfolgreich durchgeführt wurde.
OUT	Info	`MpUserInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.13.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpUserSign Konfiguration verwendet.

Dieser Funktionsbaustein wird für das elektronische Signieren einer Aktion benötigt. Das Prinzip einer elektronischen Signature ist im Abschnitt Elektronische Signatur erklärt.

Wird über die Visualierung eine elektronische Signatur durchgeführt (Visualisierung ist hierbeit mit MpUserSignatureUI verbunden), so wird an diesem Funktionsbaustein angezeigt, ob der Prozess erfolgreich war oder nicht. Die Ausgängen "Released", bzw. "Rejected" werden jeweils für einen Zyklus gesetzt.

- **Released:** Die elektronische Signatur wurde erfolgreich durchgeführt, der Benutzerlevel des Signierenden ist ausreichend für die geforderte Aktion
- **Rejected:** Die elektronische Signatur wurde nicht erfolgreich durchgeführt, der Benutzer hat den Prozess abgebrochen oder die Anzahl der fehlgeschlagenen Signatur - Versuche hat den maximal erlaubten Wert überschritten

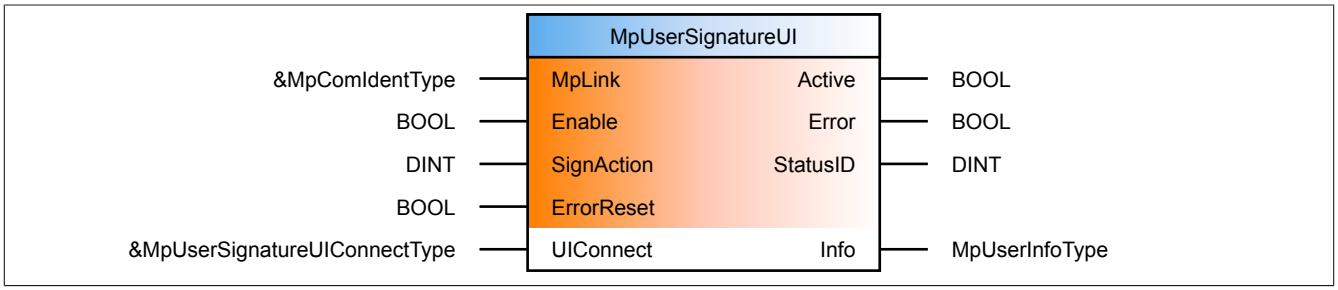
Abhängig von dem gesetzten Ausgang, kann in der Applikation auf die elektronische Signatur reagiert werden, beispielsweise der auf der Visualisierung eingegeben neue Prozesswert von einer temporären Variable auf die eigentliche Prozessvariable übertragen werden.

Der Ausgang "ActionID" gibt an welche Aktion signiert wurde. Die unterschiedlichen Aktionen können in der Konfiguration definiert werden.

8.5.1.14 MpUserSignatureUI

Dieser Funktionsbaustein ermöglicht eine elektronische Signatur über die Visualisierung.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpUserSign Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	SignAction	`DINT`	Löst den Prozess rund um die elektronische Signatur über die Applikation aus.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	UIConnect	`Zeiger auf MpUserSignatureUIConnectType`	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpUserInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

8.5.1.14.1 Beschreibung

Für diese Komponente wird derselbe MpLink verwendet, wie für MpUserSignature

Diese Komponente verwaltet das Dialogfenster zur elektronischen Signatur. Der Austausch der Daten zwischen MpUserSignatureUI und der Visualisierung erfolgt über die Struktur MpUserSignatureUIConnectType.

Pro MpUserSignature kann nur ein MpUserSignatureUI verwendet werden. Es ist nicht möglich mehrere MpUserSignatureUI -Instanzen zu verwenden.

MpUserSignatureUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

- **Status:** Der "Status" Parameter gibt Auskunft über die aktuelle Aktivität der Benutzerverwaltung
- **SignAction:** Dieser Parameter gibt an welche Aktion signiert werden soll. Dieser Parameter muss mit jedem *Completion Datenpunkt eines Visualisierungs-Elements* verbunden werden, welches eine elektronische Signature fordert. Der Wert, auf welchen der *Completion Datenpunkt* gesetzt wird, muss der "ActionID" der jeweiligen Aktion aus der Konfiguration entsprechen
- **Dialog:** Dialogfenster, welches sich öffnet, falls eine elektronische Signature gefordert wird. Es müssen Benutzername und Passwort eingegeben werden, sowie optional ein Kommentar
- **MessageBox:** Unterschiedliche Meldungen an den Benutzer (z.B. fehlerhafter Benutzername oder Passwort wurde angegeben)
- **DefaultLayerStatus:** Solange eines der Dialogfenster aktiv ist, kann der dahinterliegende Layer über diesen Datenpunkt gesperrt werden

Fehlernummern

Tritt bei einem Signatur-Versuch ein Fehler auf, da beispielsweise das Passwort falsch angegeben wurde, so wird der Fehler in der MessageBox-Struktur MpUserUIMessageBoxType unter "ErrorNumber" angezeigt. Folgende Fehler können auftreten:

- 24576: Passwort ungültig
- 24577: Ungültige Konfiguration
- 24581: Ungenügende Benutzerrechte
- 24584: Benutzer gesperrt
- 24585: Passwortänderung erforderlich
- 24598: Fehlender Wert an UIConnect
- 24599: Ungültiger Benutzername
- 24600: Passwort fehlt
- 24614: Signieren fehlgeschlagen
- 24615: Letzte Signatur wurde noch nicht geprüft

8.5.2 Datentypen und Enumeratoren

8.5.2.1 Datentypen

8.5.2.1.1 MpUserConfigType

Über diesen Datentyp wird das Benutzermanagement konfiguriert.

Wie das Konfigurationsupdate der einzelnen Parameter zur Laufzeit durchgeführt wird, ist im Abschnitt Konfiguration während der Laufzeit erklärt.

> **Alle Passwort Parameter gelten nur für Benutzer die neu über die Visualisierung mithilfe von MpUserManagerUI angelegt werden. Für Benutzer die über das Konfigurationssystem in Automation Studion, oder über MpUserCreateUser erstellt werden, gelten diese Regeln nicht.**

Parameter	Datentyp	Wert	Beschreibung	Konfigurationsupdate
NoDelete	BOOL	TRUE	Löschen von Benutzern nicht erlaubt	.
PasswordChangeReq	BOOL	FALSE	Passwortänderung beim ersten Login notwendig	.
PasswordCase	BOOL	FALSE	Passwort muss Klein- und Großbuchstaben enthalten	.
PasswordAlpha	BOOL	TRUE	Passwort muss alphanumerisch sein	.
PasswordLength	UINT	5	Minimale Passwortlänge. Es muss mindestens ein Zeichen angegeben werden. Es ist nicht erlaubt eine Passwort-Länge von 0 zu haben.	.

Parameter	Datentyp	Wert	Beschreibung	Konfigurations-update
LoginAttempts	UINT	3	Maximale Anzahl an möglichen Login - Fehlversuchen, bis der Benutzer gesperrt wird. Bei der Eingabe von 0 wird der Benutzer niemals gesperrt. Bei Benutzergruppen mit Adminrechten werden diese maximal eine Stunde gesperrt, bzw. sind Benutzergruppen mit Adminrechten nach einem Warmstart der Maschine nicht mehr gesperrt.	.
PasswordChangeInterval	DINT	0	Intervall, in dem das Passwort geändert werden muss	.
UserExpirationTime	DINT	0	Verfallszeit für die Benutzerkonto Gültigkeit	.
UserNameLength	UINT	1	Minimale Länge bei der Eingabe eines neuen Benutzernamens	.
SignAttempts	UINT	3	Maximale Anzahl an möglichen Signatur-Fehlversuchen, bis der Signatur-Vorgang abgebrochen wird. Die Login-Versuche ("LoginAttempts") und die Signatur-Versuche hängen miteinander zusammen. Bei jeder falschen Benutzereingabe während eines Signatur-Versuchs wird auch ein Login-Versuch verwendet. Gibt es 2 Signatur-Versuche und 1 Login-Versuch, so wird der Benutzer bei einem fehlgeschlagenen Signatur-Versuch (Eingabe eines falschen Passworts/Benutzernamens) bereits gesperrt. Wird die Signatur abgebrochen, so bleibt der Login-Versuch erhalten. Für mehr Information siehe hier.	.
PasswordHistory	UINT	0	Gibt an wie viele vorherigen Passwörter nicht wiederverwendet werden dürfen Es kann ein Wert zwischen 0 und 5 eingegeben werden Bei der Angabe von 0 wird die Funktion nicht verwendetBei der Angabe von zum Beispiel 2, darf das neu eingegebene Passwort nicht mit den zwei vorherigen Passwörtern übereinstimmen	.
PasswordSpecial	BOOL	FALSE	Bestimmt, ob ein Passwort Sonderzeichen enthalten muss	.
EditSameLevel	BOOL	FALSE	Bestimmt, ob ein Admin Benutzer mit dem gleichem Benutzerlevel editieren darf	.
ImportUser	MpUserImportModeEnum	mpUSER_IMPORT_IGNORE_DEFAULT	Modus für Benutzerimport	.
ImportGroup	MpUserImportModeEnum	mpUSER_IMPORT_OVERWRITE_ONLY	Modus für Gruppenimport	.
ImportUnchecked	BOOL	FALSE	Ignorieren der Check-Summe in der Importdatei	.
AdminUnlockTime	DINT	3600	Nach Ablauf der angegebenen Zeit wird der gesperrte Administrator entsperrt. (Eiheit: [sec]) Wird 0 angegeben, wird der Administrator nicht automatisch entsperrt. Die Entsperr-Funktion ist bei 0 nicht verfügbar	.

8.5.2.1.2 MpUserDiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	MpUserStatusIDType	StatusID Diagnose Struktur

8.5.2.1.3 MpUserInfoType

Mit diesem Datentyp wird für die MpUser Komponenten zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	MpUserDiagType	Eine Diagnosestruktur für die StatusID

8.5.2.1.4 MpUserLoginConfigType

In diesem Datentyp sind Parameter gesammelt, die für jeden Loginpunkt (Zugriffspunkt auf die Maschine) unterschiedlich sein können. Der Parameter "AutoLogout" wird automatisch auf den angegebenen Wert zurückgesetzt sobald sich der Eingang "LifeSign" von MpUserLogin erhöht. Verändert sich der Wert an "LifeSign" während der angegebenen Ausloggzeit nicht, wird der Benutzer automatisch ausgeloggt.

Wie das Konfigurationsupdate der einzelnen Parameter zur Laufzeit durchgeführt wird, ist im Abschnitt Konfiguration während der Laufzeit erklärt.

Parameter	Datentyp	Wert	Beschreibung	Konfigurationsupdate
AutoLogout	DINT	600	Automatische Ausloggzeit (bei inaktivem Benutzer) [s]	.

8.5.2.1.5 MpUserLoginInfoType

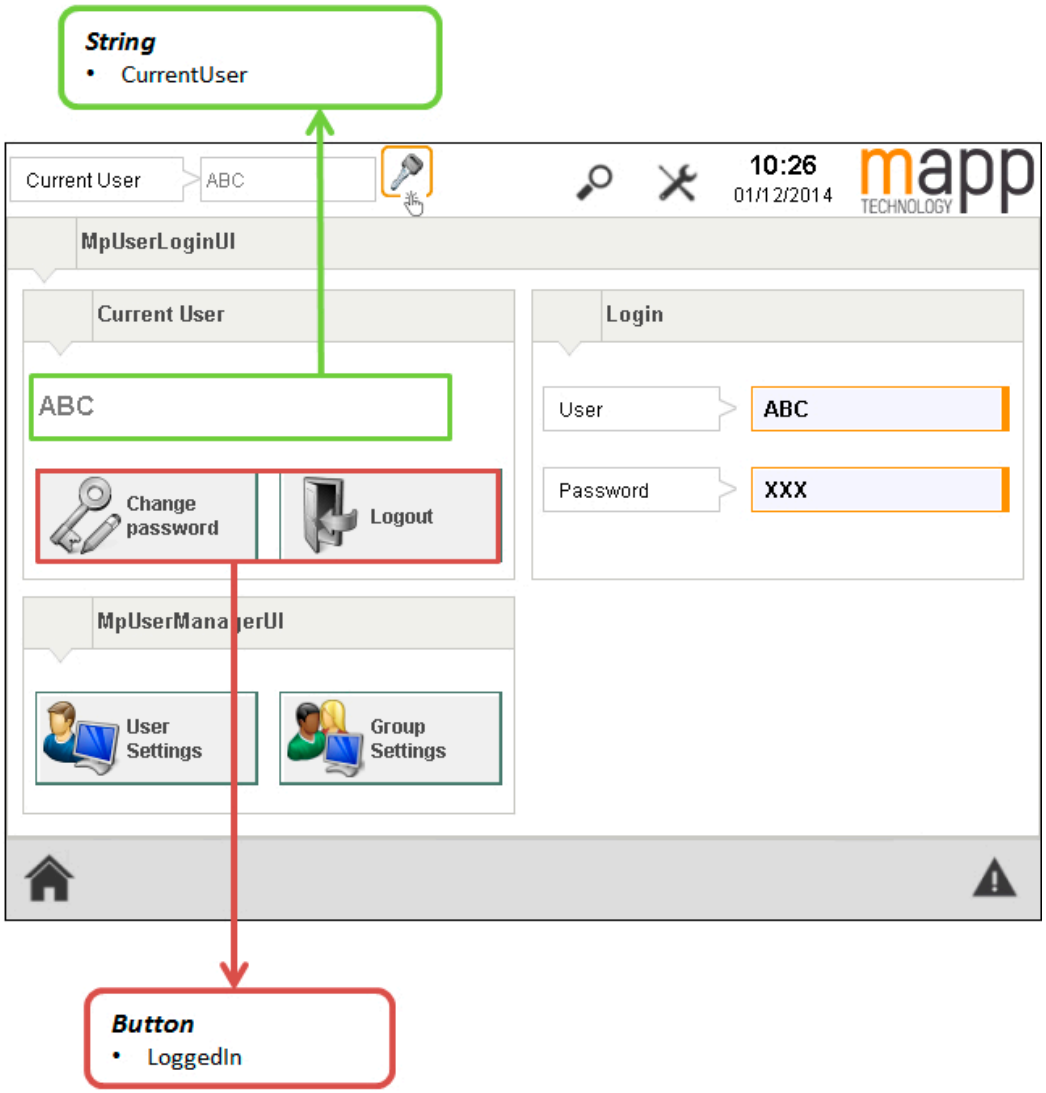
Mit diesem Datentyp wird für die MpUserLogin Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
AutoLogoutRemain	TIME	Verbleibende Zeit bis zum automatischen Ausloggen [s]
Diag	MpUserDiagType	Eine Diagnosestruktur für die StatusID

8.5.2.1.6 MpUserLoginUIConnectType

Mithilfe dieses Datentypes wird eine Verbindung zwischen der MpUserLoginUI Komponente und der Visual Components 4 Visualisierung erstellt. Über diese Struktur kann ein- und ausgeloggt werden.

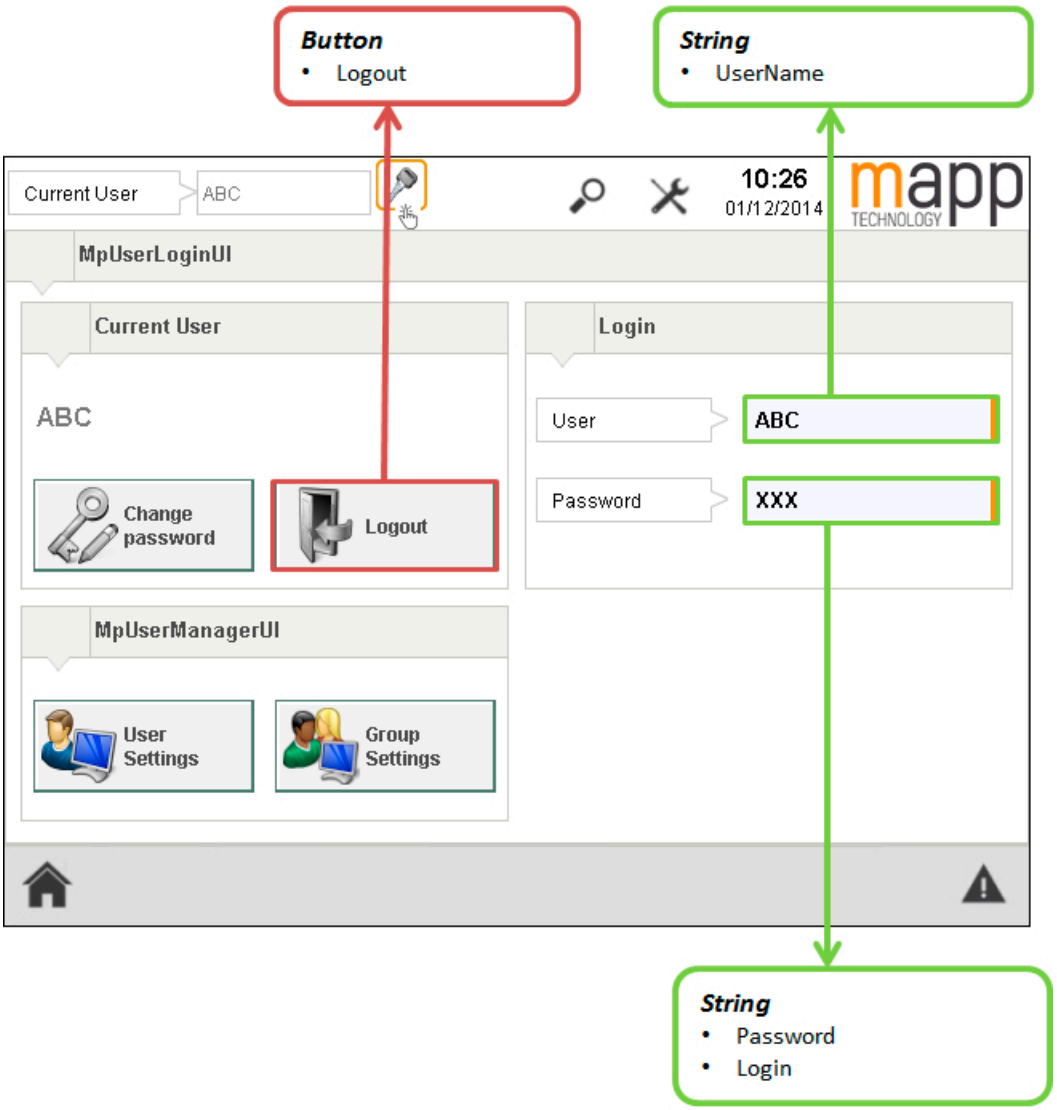
Parameter	Datentyp	Beschreibung
Status	MpUserUIStatusEnum	Aktuelle Operation
CurrentUser	WSTRING[20]	Aktuell eingeloggter Benutzer **VC4 Anbindung:** Datapoint von *String*
Language	UINT	Bevorzugte Sprache des aktuellen (oder letzten) aktiven Benutzers **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Sprachen)
DisplayUnit	UINT	Bevorzugte Einheit des aktuellen (oder letzten) aktiven Benutzers **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Einheiten)
LoggedIn	BOOL	Benutzer ist eingeloggt **VC4 Anbindung:** LockingDatapoint von *Button* / Locking: Datapoint < Level / Level: 1
Login	MpUserLoginUILoginType	Grundlegende Login Informationen
ChangePassword	MpUserLoginUIPwdType	Dient zum Ändern des Passwortes
MessageBox	MpUserUIMessageBoxType	Bedienung der Dialogfenster
DefaultLayerStatus	UINT	Statusdatenpunkt für den Default Layer der Visualisierungsseite, von der sich ein- und ausgeloggt werden kann **VC4 Anbindung:** StatusDatapoint von *Layer*
UserLevel	DINT	Level des aktuell eingeloggten Benutzers



8.5.2.1.7 MpUserLoginUILoginType

Dient zum Ein- und Ausloggen über die Visual Componets 4 Visualisierung.

Parameter	Datentyp	Beschreibung
Login	`BOOL`	Befehl zum Einloggen **VC4 Anbindung:** CompletionDatapoint von *String*
Logout	`BOOL`	Befehl zum Ausloggen **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
UserName	`WSTRING[20]`	Benutzername **VC4 Anbindung:** Datapoint von *String*
Password	`WSTRING[20]`	Passwort **VC4 Anbindung:** Datapoint von *String*

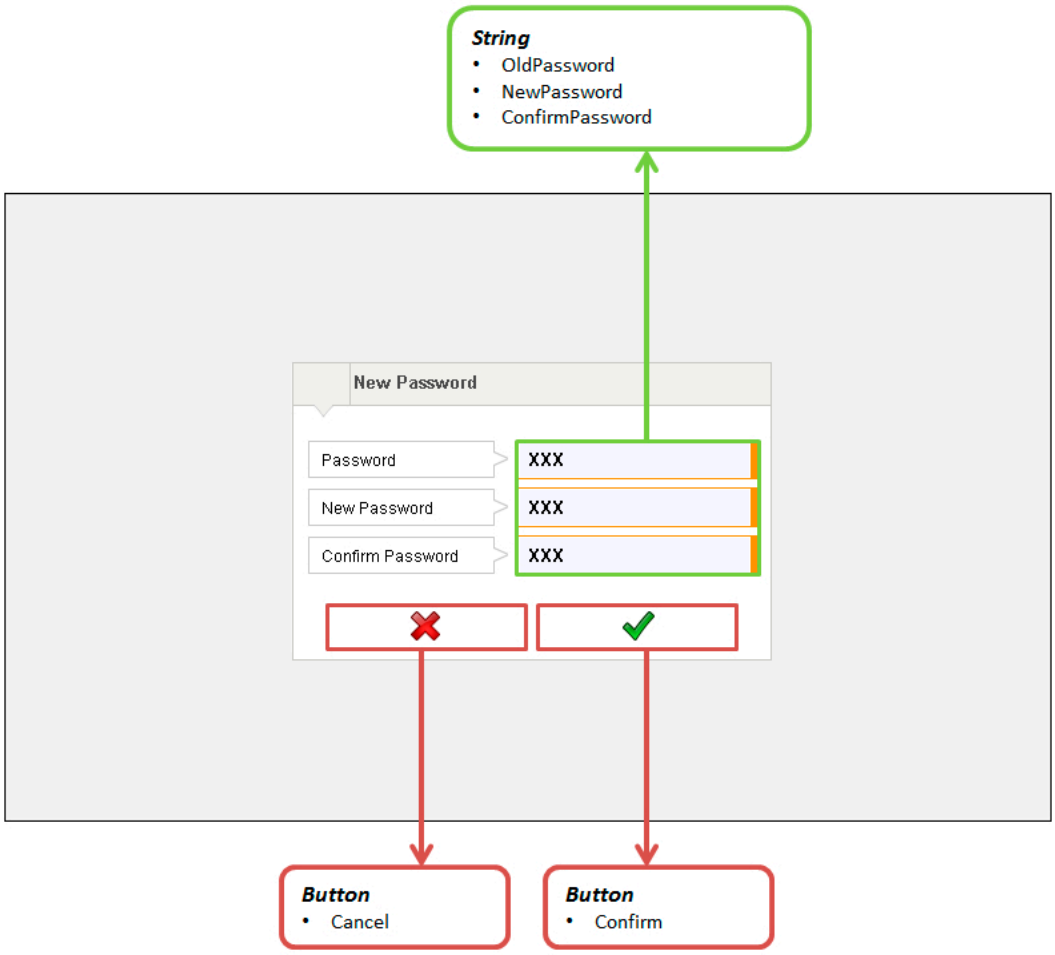


8.5.2.1.8 MpUserLoginUIPwdDlgType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern. Über dieses Dialogfenster wird das Passwort eines Benutzers geändert.

Parameter	Datentyp	Beschreibung
OldPassword	`WSTRING[20]`	Altes Passwort **VC4 Anbindung:** Datapoint von *String*
NewPassword	`WSTRING[20]`	Neues Passwort **VC4 Anbindung:** Datapoint von *String*
NewPasswordOk	`UINT`	Das neue Passwort entspricht den Kriterien für ein Passwort
CriteriaNotMet	`MpUserUIPasswordCriteriaEnum`	Falls das eingegeben Passwort nicht den Kriterien für ein neues Passwort entspricht, wird über diesen Parameter angezeigt was nicht passt. Bei mehreren nicht erfüllten Kriterien, wird immer nur eines über diese Parameter angezeigt
ConfirmPassword	`WSTRING[20]`	Bestätigung des neuen Passwortes **VC4 Anbindung:** Datapoint von *String*

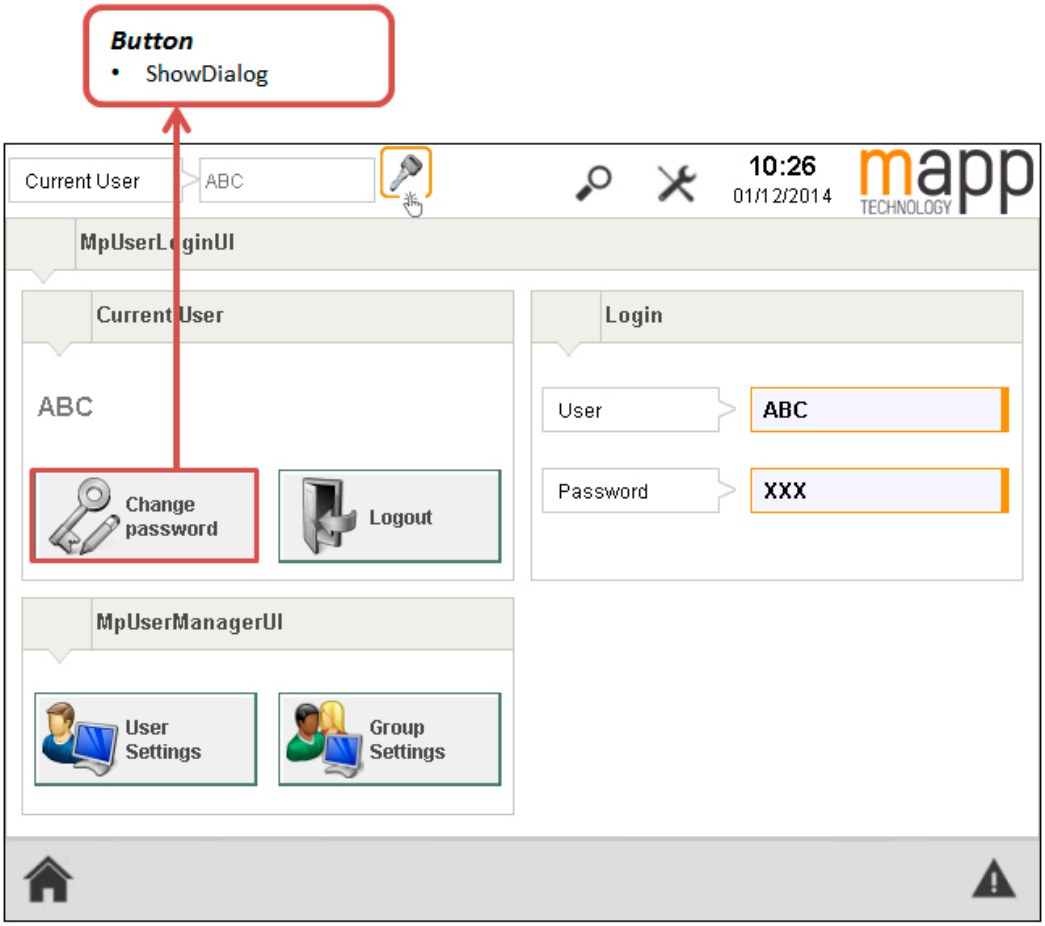
Parameter	Datentyp	Beschreibung
ConfirmPasswordOk	`UINT`	Das Passwort zum Bestätigen der richtigen Eingabe stimmt
LayerStatus	`UINT`	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Confirm	`BOOL`	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	`BOOL`	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1



8.5.2.1.9 MpUserLoginUIPwdType

Vereint Anzeige und Daten für ein Dialogfenster zum Passwort Ändern.

Parameter	Datentyp	Beschreibung
ShowDialog	`BOOL`	Befehl, der das Dialogfenster öffnet **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Dialog	`MpUserLoginUIPwdDlgType`	Dialog zum Passwort Ändern



8.5.2.1.10 MpUserMgrUIAdditionalDataType

Mithilfe dieses Datentypen kann man den Benutzer zusätzliche Information hinzufügen.

Parameter	Datentyp	Beschreibung
Key	`WSTRING[20]`	Kennzeichnung der zusätzlichen Information.
Value	`WSTRING[255]`	Zusätzliche Information des Benutzers.

8.5.2.1.11 MpUserMgrUIConnectType

Dieser Datentyp enthält alle notwendigen Informationen, um eine Verbindung zur Visualisierung herzustellen.

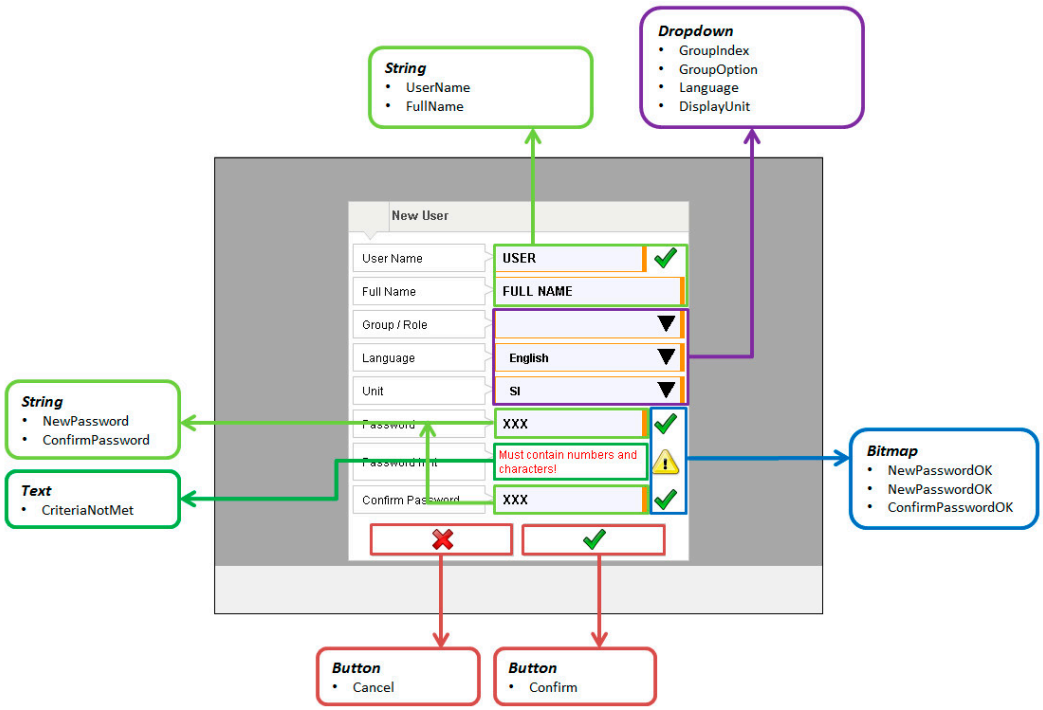
Parameter	Datentyp	Beschreibung
Status	`MpUserUIStatusEnum`	Aktuelle Operation
User	`MpUserMgrUIUserType`	Benutzerinformationen
Group	`MpUserMgrUIGroupType`	Benutzergruppeninformationen
Export	`MpUserMgrUIExportType`	Exportieren der Benutzer und Benutzergruppeneinstellungen
Import	`MpUserMgrUIImportType`	Importieren der Benutzer und Benutzergruppeneinstellungen
MessageBox	`MpUserUIMessageBoxType`	Bedienung der Dialogfenster
DefaultLayerStatus	`UINT`	Statusdatenpunkt für den Default Layer der Visualisierungsseite, auf der das Benutzermanagement angezeigt wird **VC4 Anbindung:** StatusDatapoint von *Layer*

8.5.2.1.12 MpUserMgrUICreateDlgType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern. Über dieses Dialogfenster wird ein neuer Benutzer angelegt bzw. editiert.

Nur ein Benutzer mit Administrator Rechten ist in der Lage Benutzer zu erstellen, welche ebenfalls Administrator Rechte besitzen.

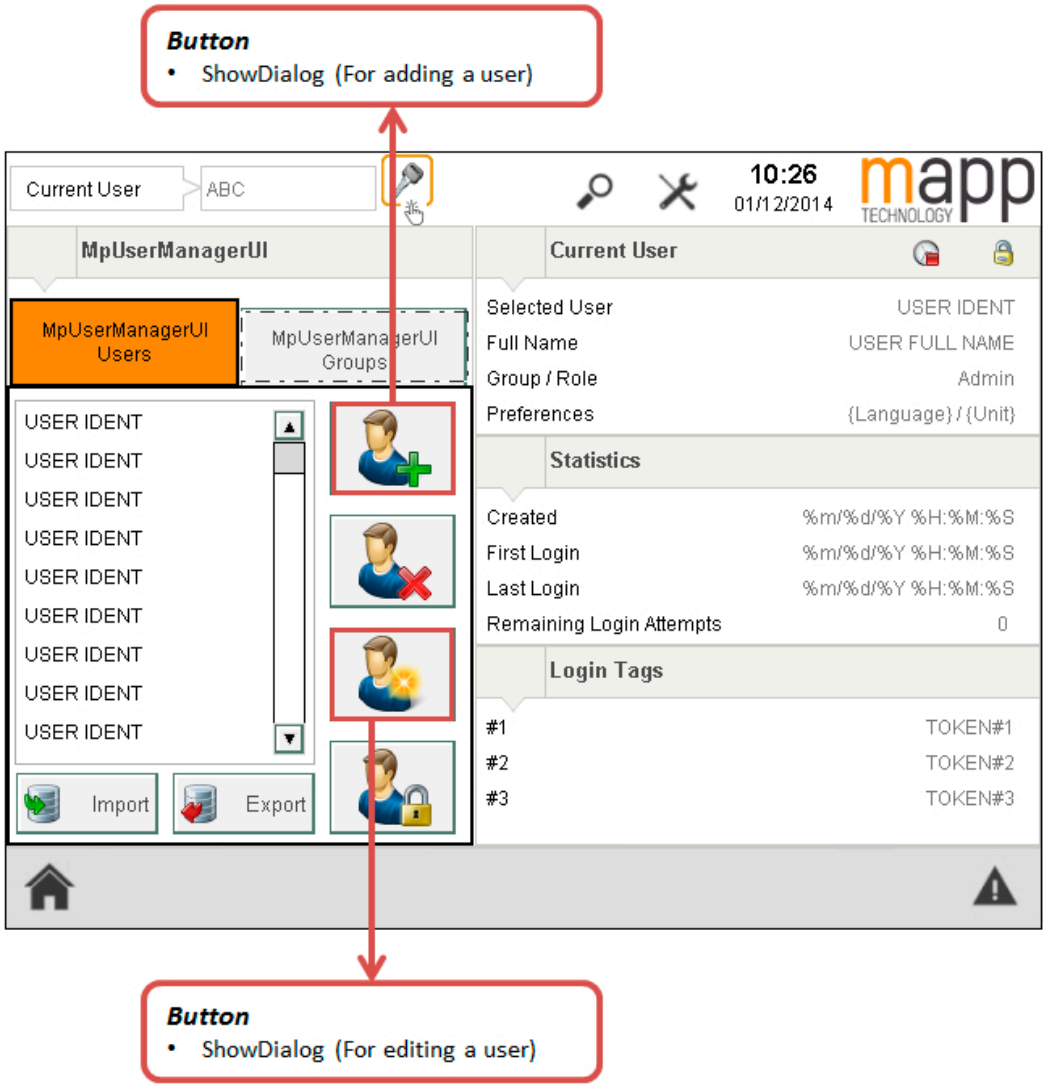
Parameter	Datentyp	Beschreibung
UserName	WSTRING[20]	Name des Benutzers. Wird ein Benutzer editiert so kann der Name nicht verändert werden. Wird ein neuer Benutzer erstellt, so kann der Benutzername hier angegeben werden. **VC4 Anbindung:** Datapoint von *String*
UserNameOk	UINT	Der Benutzername entspricht den Kriterien für einen Benutzernamen
FullName	WSTRING[100]	Voller Name des neuen Benutzers **VC4 Anbindung:** Datapoint von *String*
GroupIndex	UINT	Benutzergruppe, welcher der neue Benutzer zugeordnet werden soll. Wird mit einer Dropdown Liste verbunden, welche eine Textgruppe mit den Benutzergruppen angezeigt **VC4 Anbindung:** IndexDatapoint von *Dropdown* (Textgruppe für Benutzergruppen)
GroupOption	ARRAY[0..9] von UINT	Sortiert Benutzergruppen in der Dropdown Liste aus, welche nicht ausgewählt werden können **VC4 Anbindung:** OptionDatapoint von *Dropdown* (Textgruppe für Benutzergruppen)
NewPassword	WSTRING[20]	Neues Passwort **VC4 Anbindung:** Datapoint von *String*
NewPasswordOk	UINT	Das neue Passwort entspricht den Kriterien für ein Passwort
CriteriaNotMet	MpUserUIPasswordCriteriaEnum	Falls das eingegeben Passwort nicht den Kriterien für ein neues Passwort entspricht, wird über diesen Parameter angezeigt was nicht passt. Bei mehreren nicht erfüllten Kriterien, wird immer nur eines über diese Parameter angezeigt
ConfirmPassword	WSTRING[20]	Bestätigung des Passwortes **VC4 Anbindung:** Datapoint von *String*
ConfirmPasswordOk	UINT	Das Passwort zum Bestätigen der richtigen Eingabe stimmt
Language	UINT	Bevorzugte Sprache des neuen Benutzers **VC4 Anbindung:** IndexDatapoint von *Dropdown* (Textgruppe für Sprachen)
DisplayUnit	UINT	Bevorzugte Einheit des neuen Benutzers **VC4 Anbindung:** IndexDatapoint von *Dropdown* (Textgruppe für Einheiten)
LayerStatus	UINT	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Confirm	BOOL	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	BOOL	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
AdditionalData	ARRAY[0..9] von MpUserMgrUI-AdditionalDataType	Zusätzliche Information des Benutzers



8.5.2.1.13 MpUserMgrUICreateType

Vereint Anzeige und Daten für ein Dialogfenster um einen Benutzer zu erstellen und zu editieren.

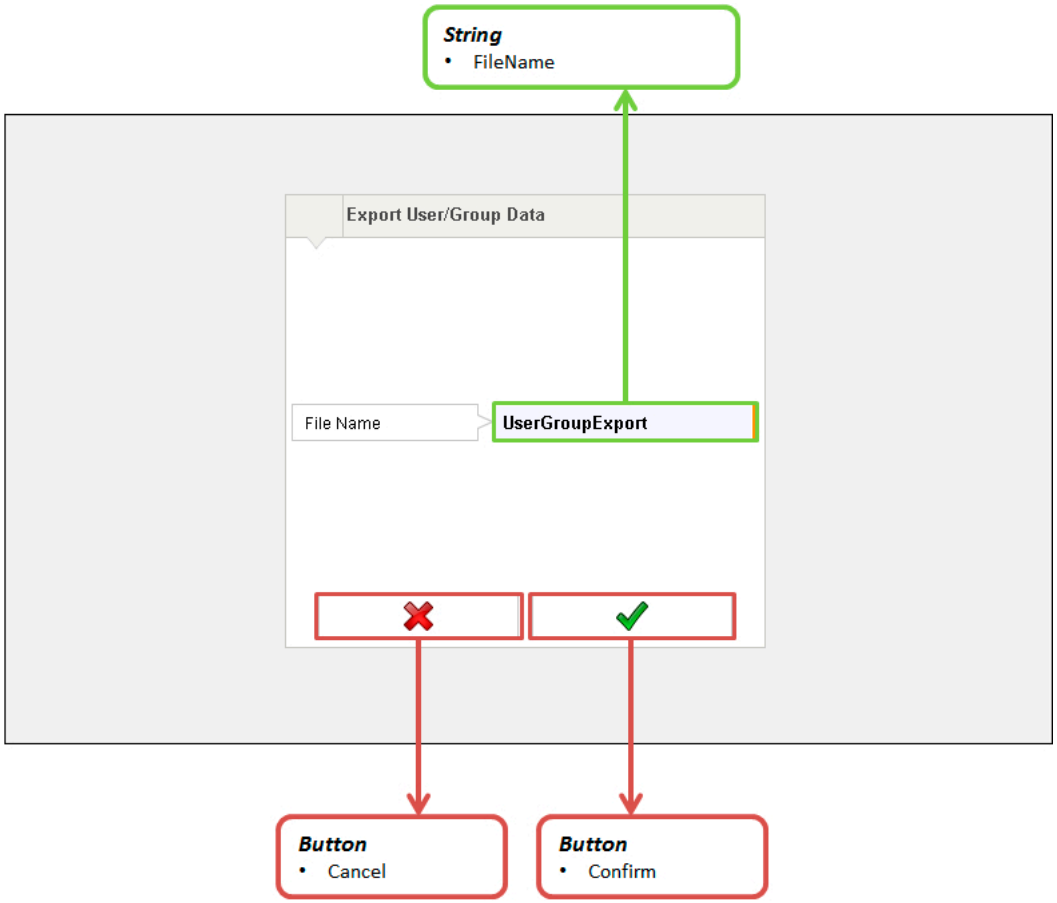
Parameter	Datentyp	Beschreibung
ShowDialog	`BOOL`	Befehl, der das Dialogfenster öffnet **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Lock	`BOOL`	Sperrt die Funktion Benutzer zu Editieren / zu Erstellen **VC4 Anbindung:** LockingDatapoint von *Button* / Locking: Datapoint >= Level / Level: 1
Dialog	`MpUserMgrUICreateDlgType`	Dialog zum Benutzer Erstellen



8.5.2.1.14 MpUserMgrUIExportDlgType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern. Über dieses Dialogfenster wird eine Datei exportiert, welche alle Benutzerdaten und Benutzergruppeneinstellungen enthält.

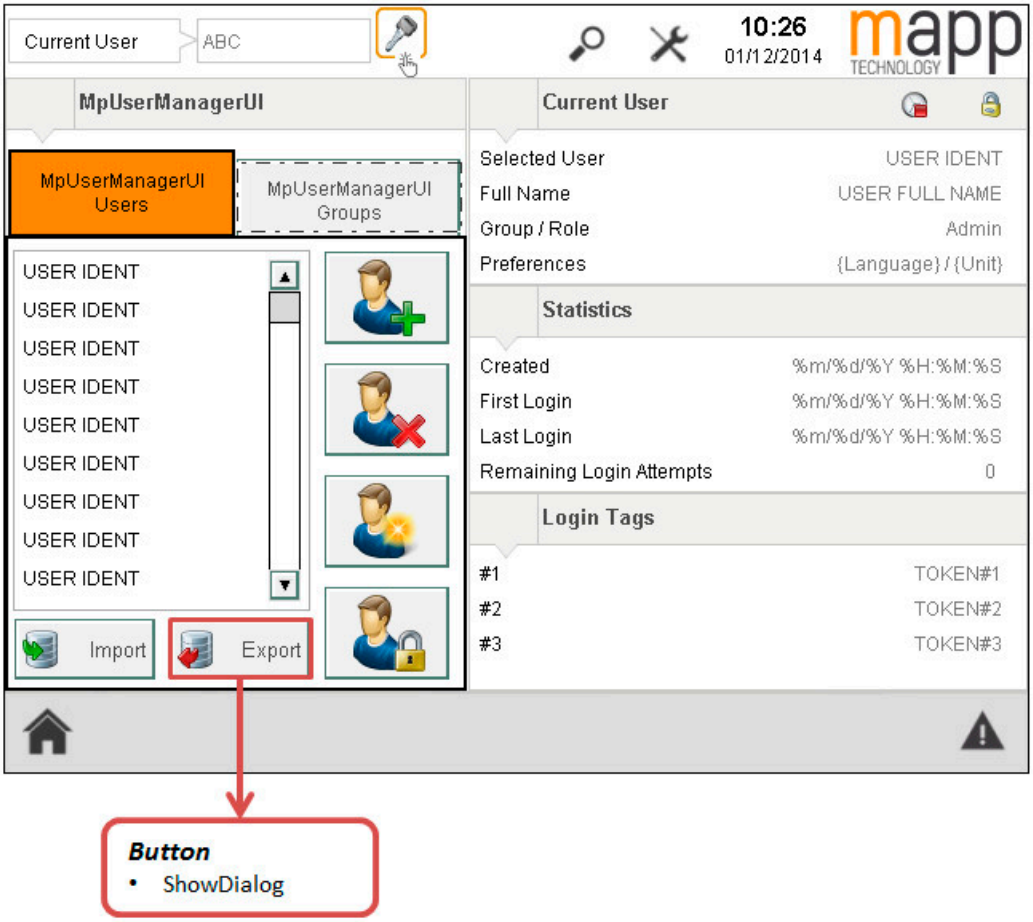
Parameter	Datentyp	Beschreibung
FileName	`STRING[255]`	Name der Datei, welche exportiert werden soll. Die Datei kann auch in einem Ordner auf dem angegebenen Speichermedium abgelegt werden. Dabei muss der Ordername mitangegeben werden, wie zum Beispiel "Ordername/FileName" **VC4 Anbindung:** Datapoint von *Sting*
LayerStatus	`UINT`	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Confirm	`BOOL`	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	`BOOL`	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1



8.5.2.1.15 MpUserMgrUIExportType

Vereint Anzeige und Daten für ein Dialogfenster zum Exportieren der Benutzerdaten.

Parameter	Datentyp	Beschreibung
ShowDialog	`BOOL`	Befehl, der das Dialogfenster öffnet **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Dialog	`MpUserMgrUIExportDlgType`	Dialog zum Exportieren der Benutzer-daten



8.5.2.1.16 MpUserMgrUIGroupCreateDlgType

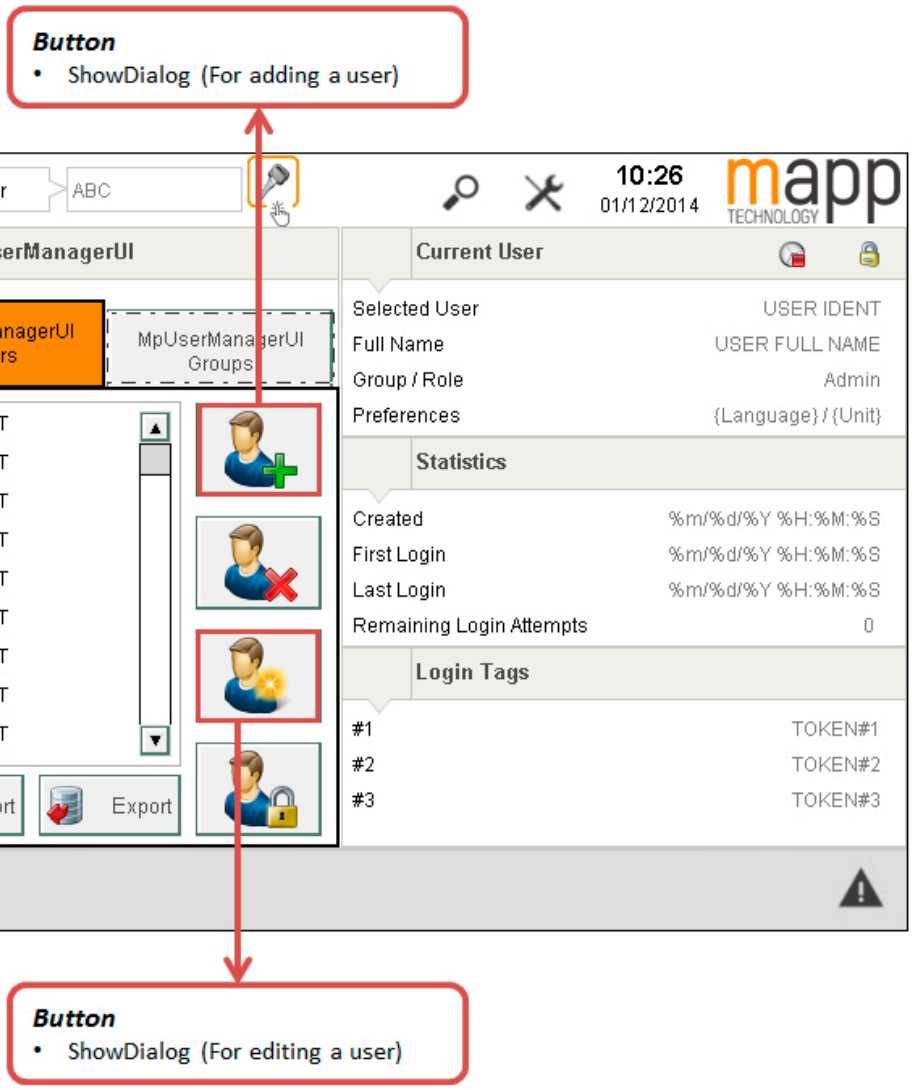
Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern. Über dieses Dialogfenster wird eine neue Benutzergruppe angelegt.

Parameter	Datentyp	Beschreibung
Name	`WSTRING[20]`	Benutzergruppenname **VC4 Anbindung:** Datapoint von *String*
Level	`UINT`	Level dieser Benutzergruppe **VC4 Anbindung:** IndexDatapoint von *Dropdown* (Textgruppe für Benutzergruppen)
LevelLimit	`DINT`	Maximal erlaubtes Level, welches für diese Benutzergruppe eingegeben werden kann (hängt vom aktuell eingeloggten Benutzer ab)
Admin	`BOOL`	Definiert, ob die neue Gruppe Administrator Rechte besitzt
AccessRights	`ARRAY[0..19]` von `MpUserAccessRightEnum`	Bestimmt, welche Rechte die neue Benutzergruppe besitzt **VC4 Anbindung:** OptionDatapoint von *Dropdown* (Textgruppe für Benutzergruppen)
AccessRightsLimit	`ARRAY[0..19]` von `MpUserAccessRightEnum`	Mögliche Rechte, welche für diese Benutzergruppe eingegeben werden kann (hängt vom aktuell eingeloggten Benutzer ab)
LayerStatus	`UINT`	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Confirm	`BOOL`	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	`BOOL`	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1

8.5.2.1.17 MpUserMgrUIGroupCreateType

Vereint Anzeige und Daten für ein Dialogfenster zum Benutzergruppen Erstellen.

Parameter	Datentyp	Beschreibung
ShowDialog	`BOOL`	Befehl, der das Dialogfenster öffnet **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Lock	`BOOL`	Sperrt die Funktion Benutzer zu Editieren / zu Erstellen **VC4 Anbindung:** LockingDatapoint von *Button* / Locking: Datapoint >= Level / Level: 1
Dialog	`MpUserMgrUIGroupCreateDlgType`	Dialog zum Benutzergruppen Erstellen

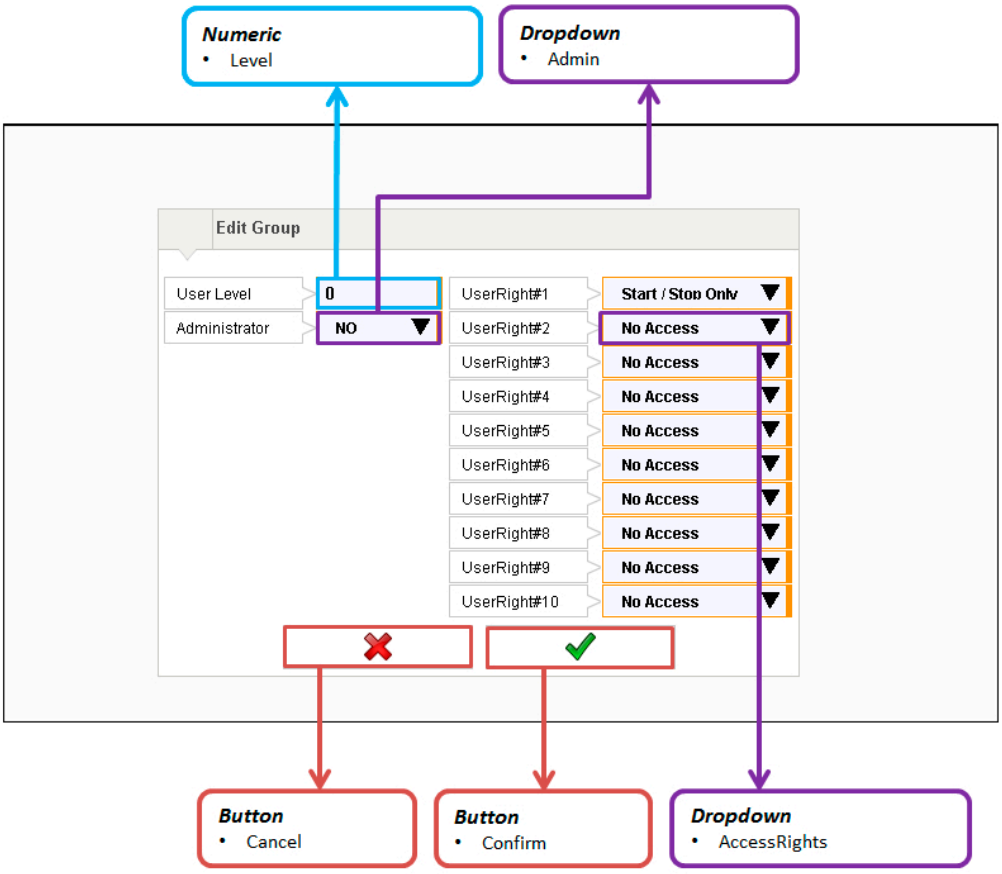


8.5.2.1.18 MpUserMgrUIGroupDlgType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern. Über dieses Dialogfenster wird eine Benutzergruppe editiert.

Parameter	Datentyp	Beschreibung
Name	`WSTRING[20]`	Name der Benutzergruppe **VC4 Anbindung:** Datapoint von *String*
Index	`UINT`	Index der Benutzergruppe **VC4 Anbindung:** Datapoint von *Numeric*
Level	`DINT`	Level der Benutzergruppe **VC4 Anbindung:** Datapoint von *Numeric*
LevelLimit	`DINT`	Maximal erlaubtes Level, das für diese Gruppe gesetzt werden darf
Admin	`BOOL`	Bestimmt, ob diese Gruppe Administratorrechte besitzt **VC4 Anbindung:** IndexDatapoint von *Dropdown* (Textgruppe für Adminrecht)

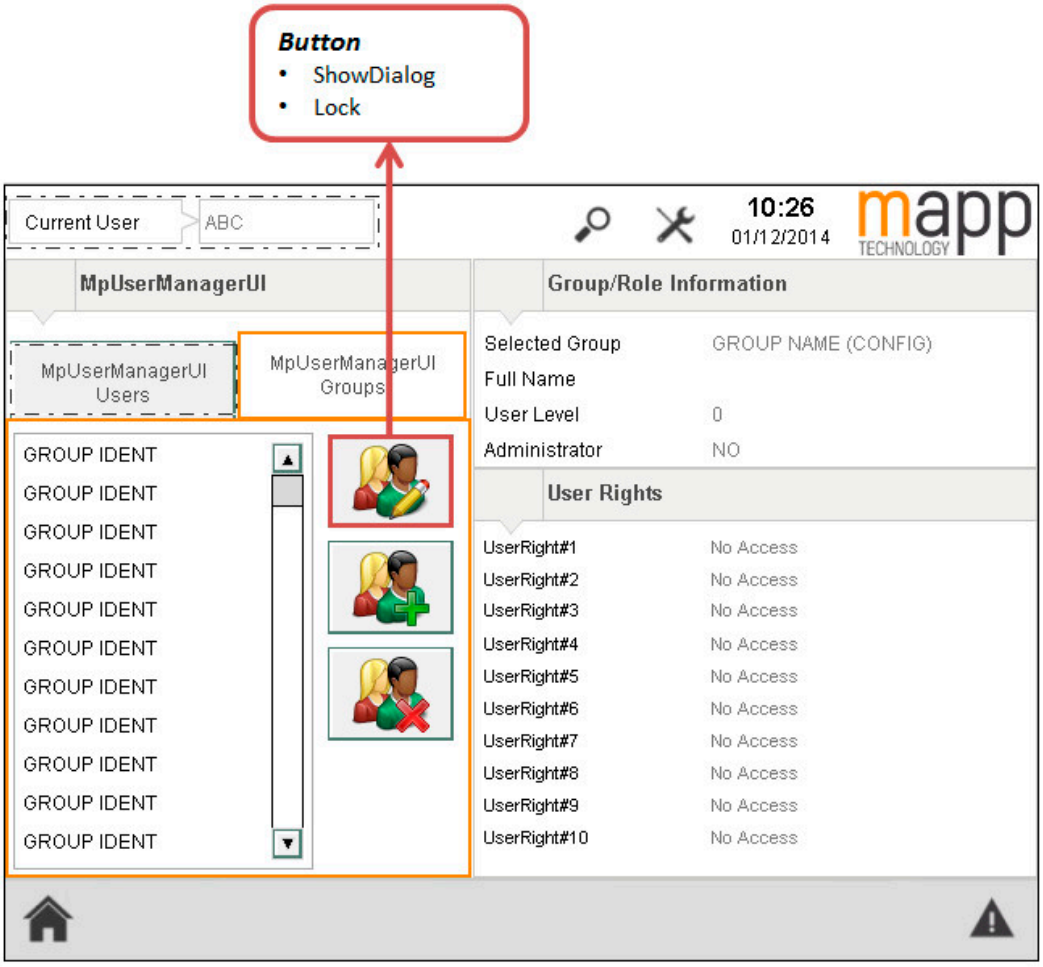
Parameter	Datentyp	Beschreibung
AccessRights	ARRAY[0..19] von MpUserAccessRightEnum	Liste mit den Benutzerrechten für einzelne Aktionen **VC4 Anbindung:** IndexDatapoint von *Dropdown* (Textgruppe für Zugriffsrechte)
AccessRightsLimit	ARRAY[0..19] von MpUserAccessRightEnum	Limits für die Eingabe der Benutzerrechte
LayerStatus	UINT	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Confirm	BOOL	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	BOOL	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1



8.5.2.1.19 MpUserMgrUIGroupEditType

Vereint Anzeige und Daten für ein Dialogfenster zum Editieren einer Benutzergruppe.

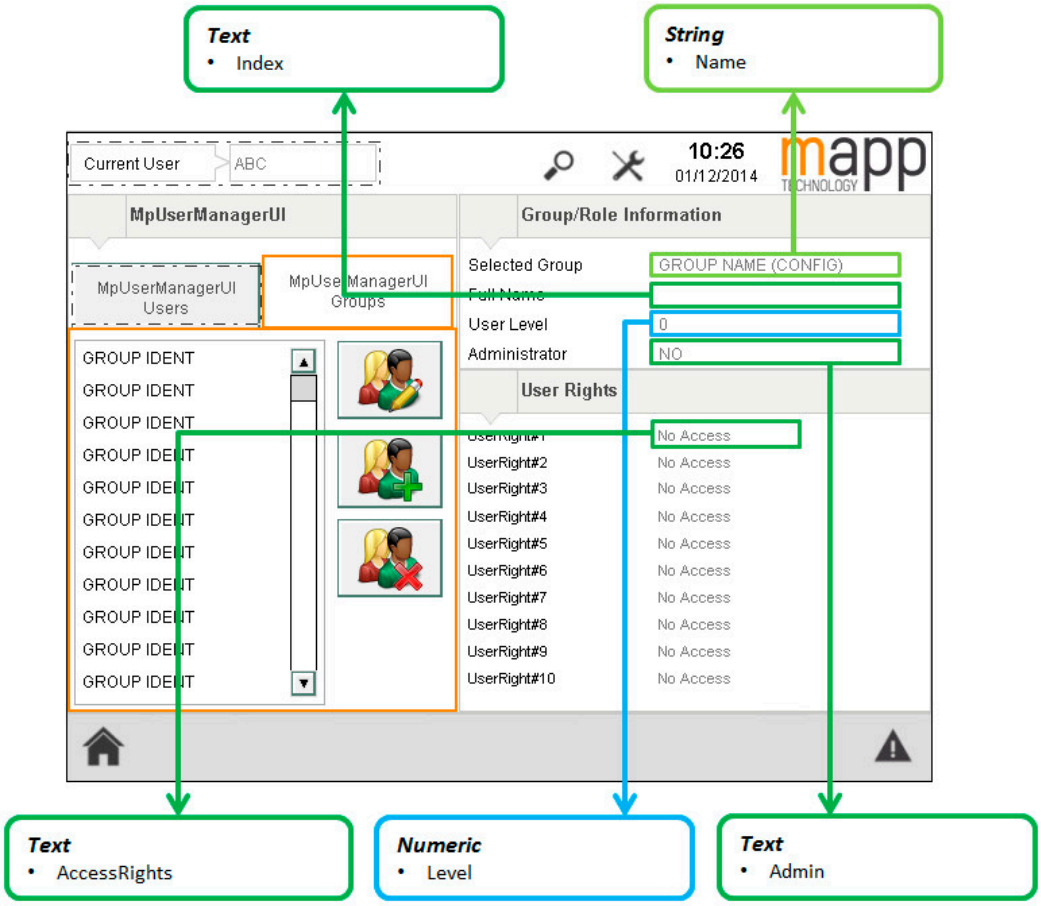
Parameter	Datentyp	Beschreibung
ShowDialog	BOOL	Befehl der das Dialogfenster öffnet **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Lock	BOOL	Sperrt die Funktion Benutzergruppen zu Editieren **VC4 Anbindung:** LockingDatapoint von *Button* / Locking: Datapoint >= Level / Level: 1
Dialog	MpUserMgrUIGroupDlgType	Dialog zum Editieren einer Benutzergruppe



8.5.2.1.20 MpUserMgrUIGroupInfoType

Gibt Auskunft über eine Benutzergruppe.

Parameter	Datentyp	Beschreibung
Name	`WSTRING[20]`	Name der Benutzergruppe **VC4 Anbindung:** Datapoint von *String*
Index	`UINT`	Index der Benutzergruppe **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Benutzergruppen)
Level	`DINT`	Level der Benutzergruppe **VC4 Anbindung:** Datapoint von *Numeric*
Admin	`BOOL`	Zeigt an, ob diese Gruppe Administratorrechte besitzt **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Adminrecht)
AccessRights	`ARRAY[0..19] von MpUserAccessRightEnum`	Liste mit den Benutzerrechten für einzelne Aktionen **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Zugriffsrechte)

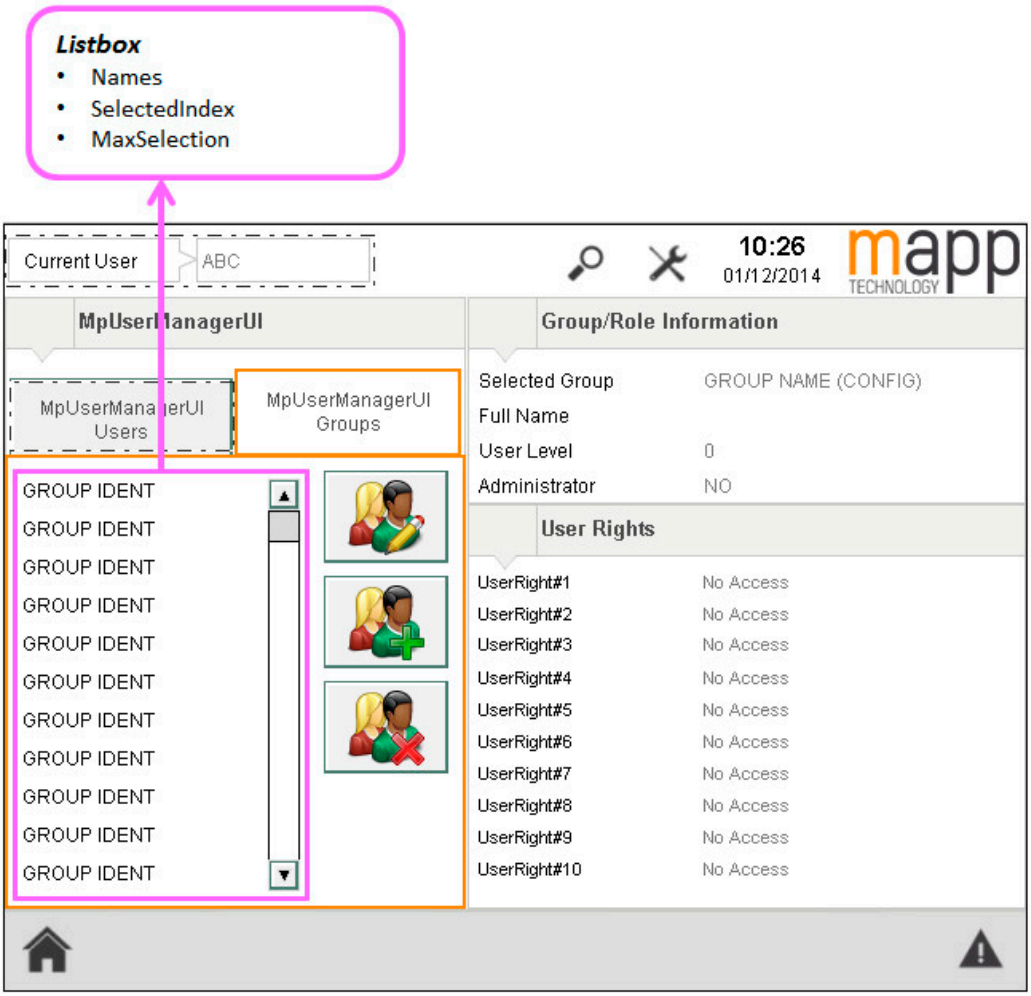


8.5.2.1.21 MpUserMgrUIGroupListType

Liste mit möglichen Benutzergruppen und die dazugehörige Navigation.

Parameter	Datentyp	Beschreibung
Names	`ARRAY[0..9] von WSTRING[20]`	Liste aller verfügbaren Benutzergruppen **VC4 Anbindung:** TextDatapoint von *Listbox*
SelectedIndex	`UINT`	Index des aktuell ausgewählten Eintrages in der Liste **VC4 Anbindung:** IndexDatapoint von *Listbox*
MaxSelection	`UINT`	Index des letzten Eintrages in der Liste **VC4 Anbindung:** MaxDatapoint von *Listbox*
PageUp	`BOOL`	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepUp	`BOOL`	Wählt vorhergehenden Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0

Parameter	Datentyp	Beschreibung
PageDown	BOOL	Springt ans Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepDown	BOOL	Wählt nächsten Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
RangeStart	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von *Scale*
RangeEnd	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von *Scale*



8.5.2.1.22 MpUserMgrUIGroupSelectType

Über diesen Datentyp werden die Namen der Gruppen und die maximale Anzahl an Gruppen angezeigt

Parameter	Datentyp	Beschreibung
Names	ARRAY[0..19 von]WSTRING[20]	Liste mit vorhandenen Gruppen
MaxSelection	UINT	Maximale Anzahl an Gruppen

8.5.2.1.23 MpUserMgrUIGroupType

Mit diesem Datentyp wird eine Liste existierender Benutzergruppen angezeigt.

Parameter	Datentyp	Beschreibung
List	MpUserMgrUIGroupListType	Liste aller Benutzergruppen, sowie die Navigation für die Liste
Info	MpUserMgrUIGroupInfoType	Enthält detailierte Benutzergruppeninformationen
Edit	MpUserMgrUIGroupEditType	Beinhaltet die Daten, welche zum Editieren einer Benutzergruppe notwendig sind
Create	MpUserMgrUIGroupCreateType	Beinhaltet die Daten, welche zum Erstellen einer neuen Benutzergruppe notwendig sind
Remove	BOOL	Löschen der ausgewählten Benutzergruppe
SelectList	MpUserMgrUIGroupSelectType	Auswahl für die Gruppenliste

8.5.2.1.24 MpUserMgrUIImportConfirmType

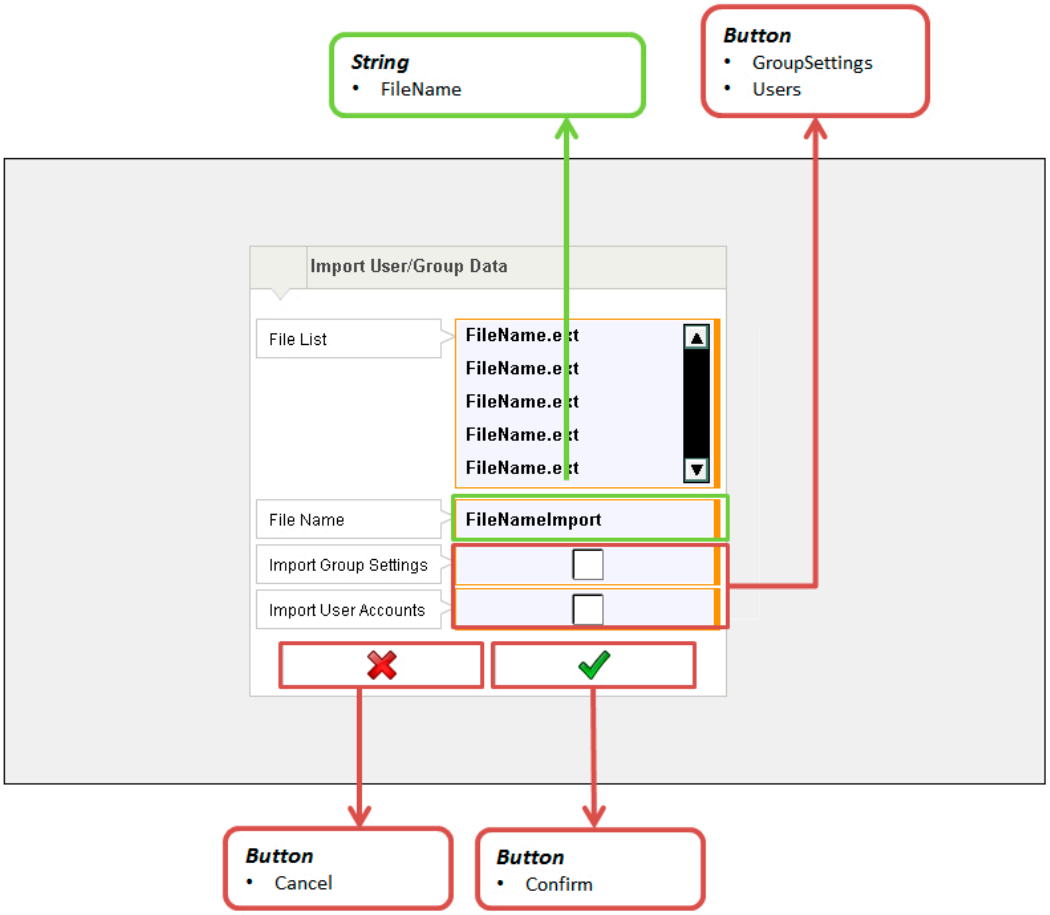
Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern.

Parameter	Datentyp	Beschreibung
LayerStatus	UINT	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Type	MpUserMgrUIImportConfirmEnum	Gibt an ob Benutzer oder Gruppe überschrieben werden sollen
Name	WSTRING[20]	Name des Benutzers oder der Gruppe
NumberOfConflicts	UINT	Anzahl an vorhandenen Konflikten
ApplyForAll	BOOL	Auswahl für alle vorhandenen Konflikte übernehmen
Confirm	BOOL	Bestätigen der Operation **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1
Cancel	BOOL	Abbrechen der Operation **VC4 Anbindung:** Datapoint von Button / Type: SetDatapoint / SetValue: 1

8.5.2.1.25 MpUserMgrUIImportDlgType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern. Über dieses Dialogfenster werden Benutzerdaten und Benutzergruppeneinstellungen importiert. Bei der Auswahl der zu importierenden Datei wird der Name in "FileName" angegeben oder die gewünschte Datei wird aus der Liste ausgewählt.

Parameter	Datentyp	Beschreibung
List	MpUserMgrUIImportListType	Liste aller zum Import verfügbaren Dateien, sowie die Navigation für die Liste
FileName	STRING[255]	Name der Datei, welche importiert werden soll **VC4 Anbindung:** Datapoint von *String*
Users	BOOL	Option, die angibt, ob Benutzerdaten übernommen werden sollen **VC4 Anbindung:** Datapoint von *Button* / Type: ToggleDatapoint / SetValue: 1 / ResetValue: 0
GroupSettings	BOOL	Option, die angibt, ob Benutzergruppeneinstellungen übernommen werden sollen **VC4 Anbindung:** Datapoint von *Button* / Type: ToggleDatapoint / SetValue: 1 / ResetValue: 0
LayerStatus	UINT	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Confirm	BOOL	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	BOOL	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1

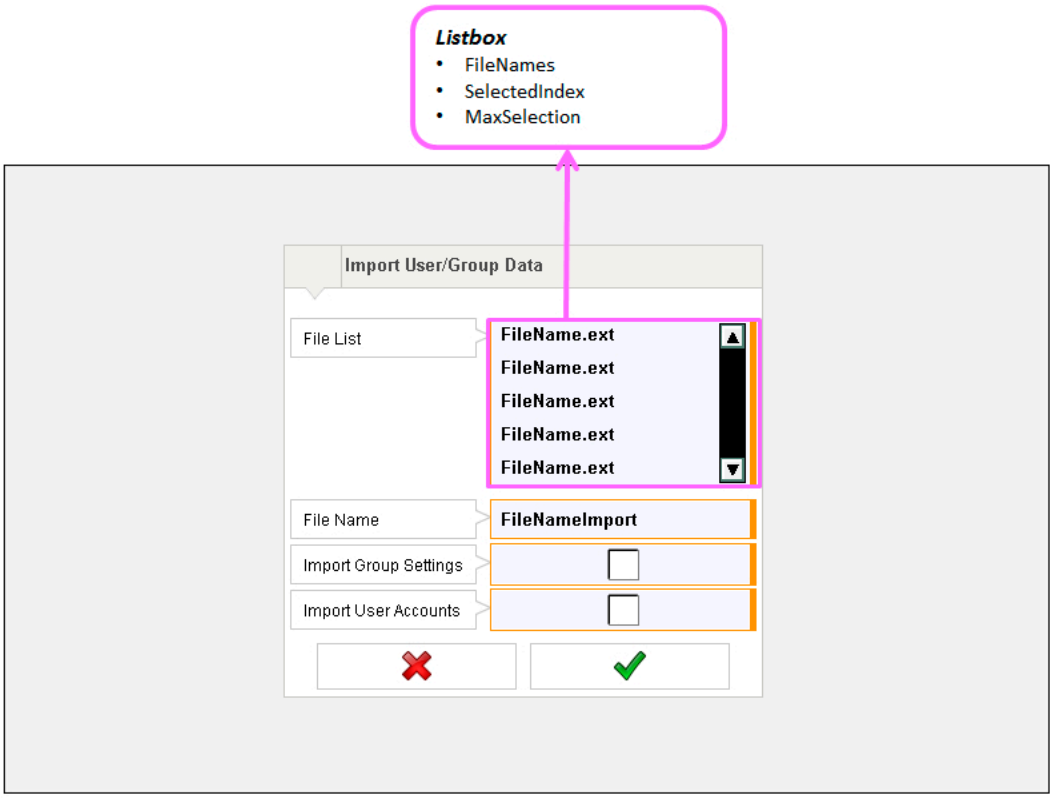


8.5.2.1.26 MpUserMgrUIImportListType

Liste mit, zum Import verfügbaren, Dateien und die dazugehörige Navigation.

> **Es ist nötig die Bibliothek MpFile auf die Steuerung zu übertragen, um bei der "Import" Funktion von MpUserManagerUI eine Liste mit möglichen Dateien anzuzeigen.**

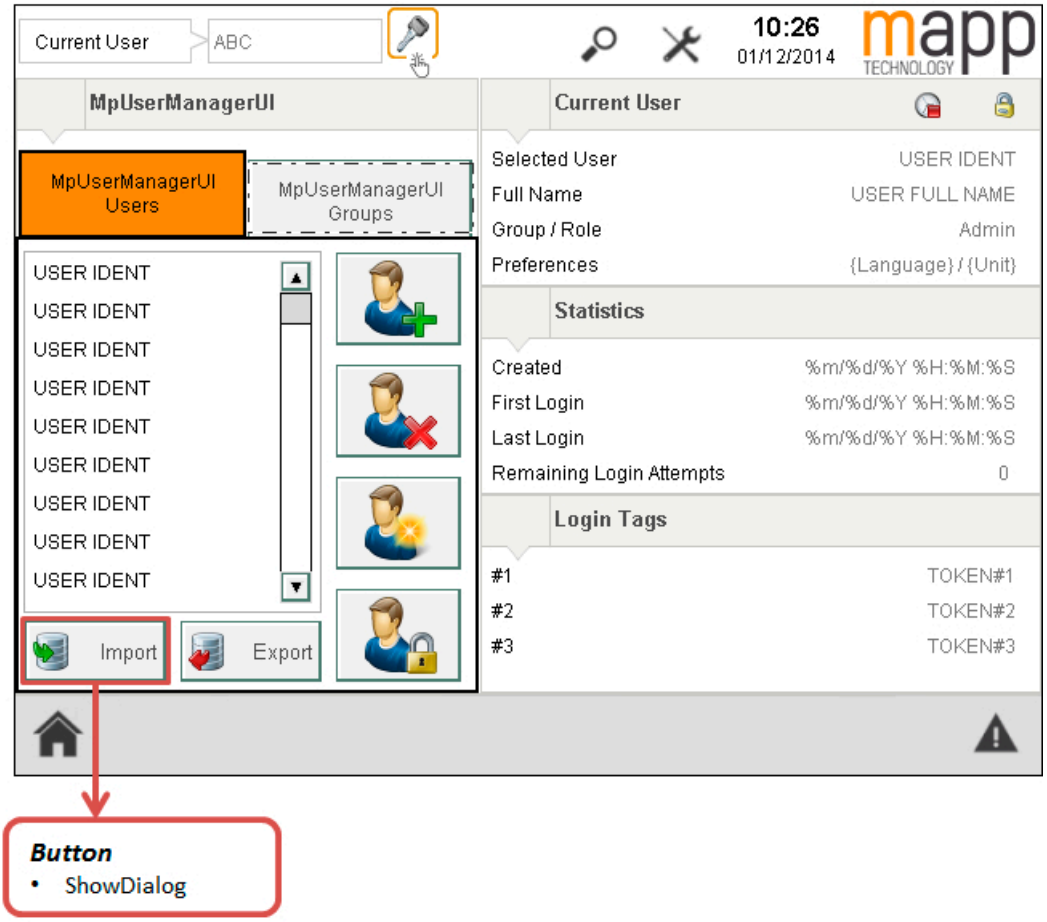
Parameter	Datentyp	Beschreibung
FileNames	`ARRAY[0..4] von STRING[255]`	Liste aller zum Import verfügbaren Dateien **VC4 Anbindung:** TextDatapoint von *Listbox*
SelectedIndex	`UINT`	Index des aktuell ausgewählten Eintrages in der Liste **VC4 Anbindung:** IndexDatapoint von *Listbox*
MaxSelection	`UINT`	Index des letzten Eintrages in der Liste **VC4 Anbindung:** MaxDatapoint von *Listbox*



8.5.2.1.27 MpUserMgrUIImportType

Vereint Anzeige und Daten für ein Dialogfenster zum Importieren der Benutzerdaten.

Parameter	Datentyp	Beschreibung
ShowDialog	`BOOL`	Befehl der das Dialogfenster öffnet **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Dialog	`MpUserMgrUIImportDlgType`	Dialog zum Importieren der Benutzer-daten
ConfirmDialog	`MpUserMgrUIImportConfirmType`	Dialog zum Bestätigen des Imports



8.5.2.1.28 MpUserMgrUISetupConfirmType

Über diesen Datentyp kann definiert werden, wann Bestätigungs Dialoge angezeigt werden sollen.

Parameter	Datentyp	Wert	Beschreibung
ConfirmOverwriteUser	`BOOL`	FALSE	Das überschreiben von Benutzern soll über einen Dialog bestätigt werden
ConfirmOverwriteGroup	`BOOL`	FALSE	Das überschreiben von Gruppen soll über einen Dialog bestätigt werden

8.5.2.1.29 MpUserMgrUISetupType

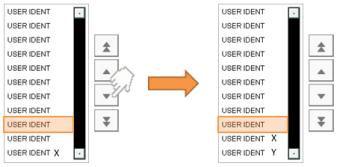
Zusätzliche Konfigurationsmöglichkeiten, um die Visualisierung zu beschreiben.

Parameter	Datentyp	Wert	Beschreibung
UserListSize	`UINT`	20	Zahl an Benutzern, die auf einer Seite der Liste auf der Visualisierung dargestellt werden können Es können maximal 20 Benutzer angezeigt werden
GroupListSize	`UINT`	20	Zahl an Gruppen, die auf einer Seite der Liste auf der Visualisierung dargestellt werden können Es können maximal 20 Gruppen angezeigt werden
ScrollWindow	`USINT`	1	Zeigt an, wieviele Einträge der Liste beim auf- oder abscrollen im Vorhinein angezeigt werden
FileDevice	`STRING[20]`	'HD'	File Device (Speichermedium) zum Exportieren bzw. Importieren der Benutzerdaten
FileExtension	`STRING[20]`	'usr'	Dateiendung für die Export- bzw. Importdateien
Confirmation	`MpUserMgrUISetupConfirmType`		Anzeige der Bestätigungs Dialoge

UserListSize

UserListSize gibt an, wie viele User auf einer Seite einer Listbox im VC4 angezeigt werden. Die Listbox in der Visualisierung zeigt je nach Größe unterschiedlich viele Elemente an. Wie viele Elemente die Listbox anzeigt, kann durch definieren eines Simulationswertes (SimulationValue) in den Listbox Eigenschaften ermittelt werden (siehe Bild oben: "USER IDENT"). Die Anzahl dieser Elemente muss mit dem Wert von UserListSize übereinstimmen.

ScrollWindow

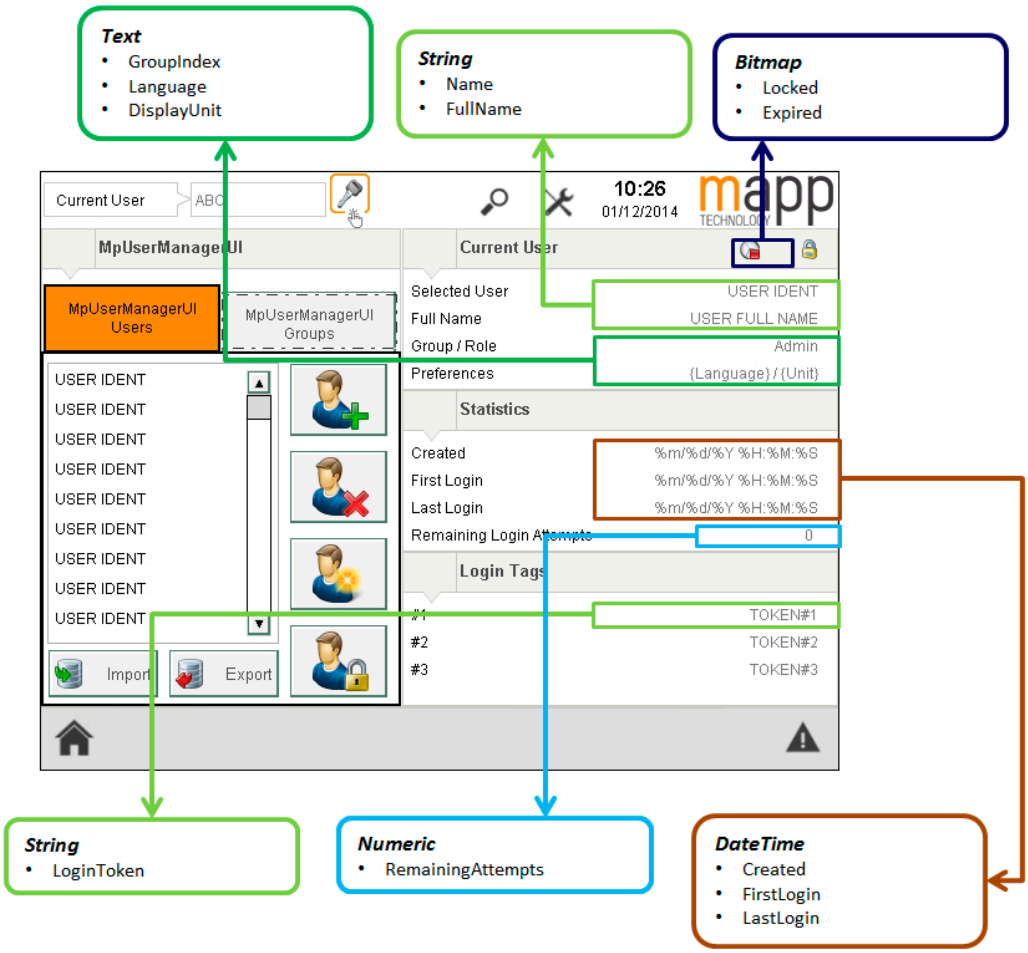


ScrollWindow gibt an, wie viele Einträge beim Scrollen "vorrausschauend" angezeigt werden sollen. In diesem Beispiel wurde ScrollWindow = 2 definiert. Befindet sich nun die aktuelle Auswahl an 3. letzter Stelle in der Liste, so wird bei erneutem Drücken des StepDown Buttons der nächste Eintrag ausgewählt und vorrauschschauend die nächsten 2 Einträge angezeigt.

8.5.2.1.30 MpUserMgrUIUserInfoType

Gibt Auskunft über einen Benutzer.

Parameter	Datentyp	Beschreibung
UserName	`WSTRING[20]`	Benutzername fürs Einloggen **VC4 Anbindung:** Datapoint von *String*
FullName	`WSTRING[100]`	Vollständiger Benutzername **VC4 Anbindung:** Datapoint von *String*
GroupIndex	`UINT`	Gruppe, welcher der Benutzer zugeordnet sein soll **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Benutzergruppen)
Locked	`BOOL`	Benutzer ist gesperrt **VC4 Anbindung:** IndexDatapoint von *Bitmap* (BitmapGroup für Gesperrt)
Expired	`BOOL`	Benutzerpasswort ist abgelaufen **VC4 Anbindung:** IndexDatapoint von *Bitmap* (BitmapGroup für Abgelaufen)
RemainingAttempts	`USINT`	Anzahl an übrigen Loginversuchen, bevor der Benutzer gesperrt wird **VC4 Anbindung:** Datapoint von *Numeric*
Language	`UINT`	Bevorzugte Sprache **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Sprachen)
DisplayUnit	`UINT`	Bevorzugte Einheit **VC4 Anbindung:** IndexDatapoint von *Text* (Textgruppe für Einheiten)
Creation	`DATE_AND_TIME`	Datum und Uhrzeit, wann dieser Benutzer erstellt wurde **VC4 Anbindung:** Datapoint von *DateTime*
FirstLogin	`DATE_AND_TIME`	Datum und Uhrzeit des ersten Logins **VC4 Anbindung:** Datapoint von *DateTime*
LastLogin	`DATE_AND_TIME`	Datum und Uhrzeit des letzen Logins **VC4 Anbindung:** Datapoint von *DateTime*
PasswordExpired	`DATE_AND_TIME`	Gibt Datum und Uhrzeit an, wann das Passwort abgelaufen ist. **VC4 Anbindung:** Datapoint von *DateTime*
AdditionalData	`MpUserMgrUIAdditionalDataType`	Zusätzliche Information des Benutzers.

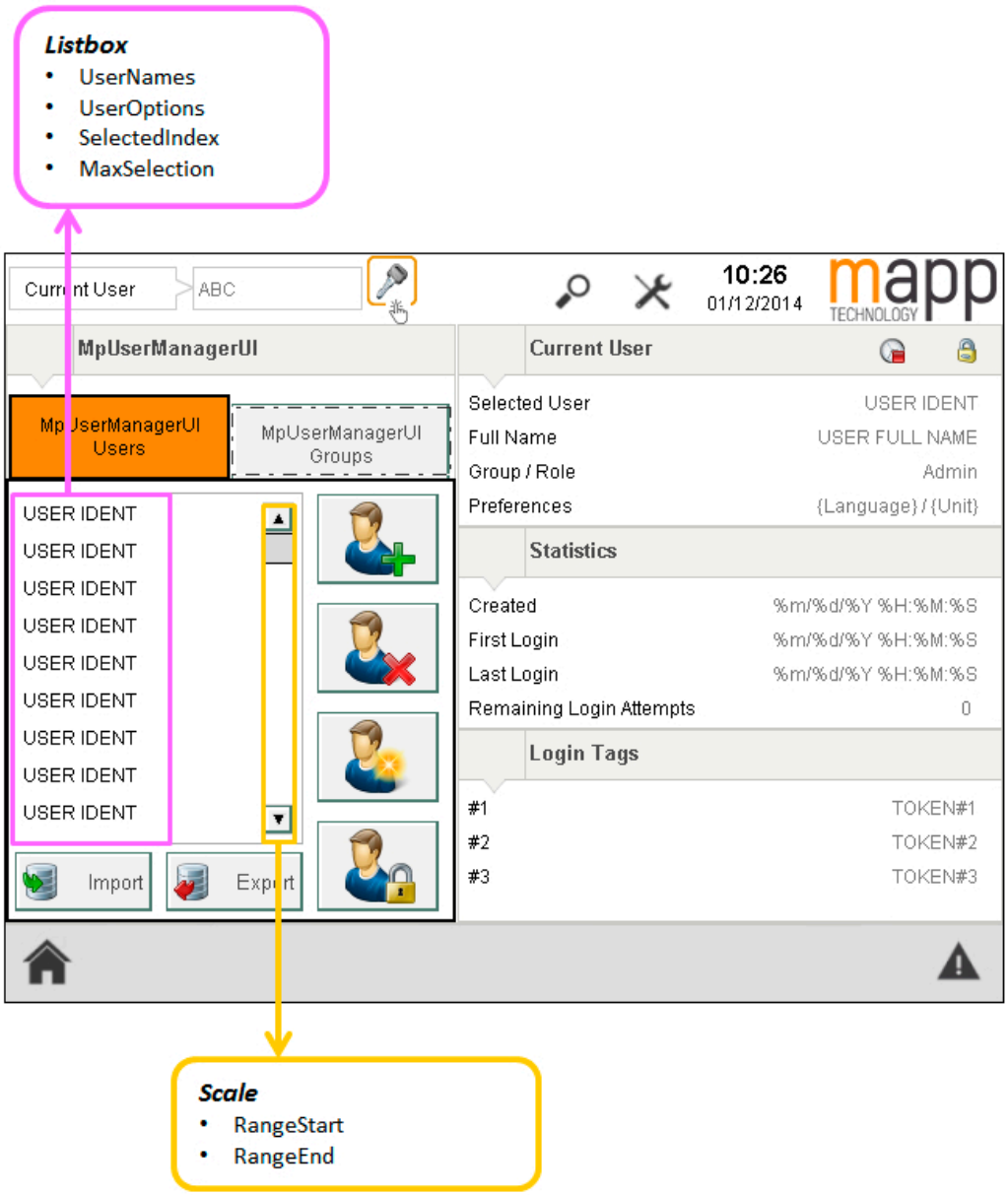


8.5.2.1.31 MpUserMgrUIUserListType

Liste mit möglichen Benutzern und die dazugehörige Navigation.

Parameter	Datentyp	Beschreibung
UserNames	`ARRAY[0..19] von WSTRING[20]`	Liste aller verfügbaren Benutzer **VC4 Anbindung:** TextDatapoint von *Listbox*
UserOptions	`ARRAY[0..19] von USINT`	Steuert die Anzeige der Elemente in der Liste **VC4 Anbindung:** OptionDatapoint von *Listbox*
SelectedIndex	`UINT`	Index des aktuell ausgewählten Eintrages in der Liste **VC4 Anbindung:** IndexDatapoint von *Listbox*
MaxSelection	`UINT`	Index des letzten Eintrages in der Liste **VC4 Anbindung:** MaxDatapoint von *Listbox*
PageUp	`BOOL`	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. Die Größe der Seite wird über den Parameter "UserListSize" der Struktur MpUserMgrUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
PageDown	`BOOL`	Springt ans Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. Die Größe der Seite wird über den Parameter "UserListSize" der Struktur MpUserMgrUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0

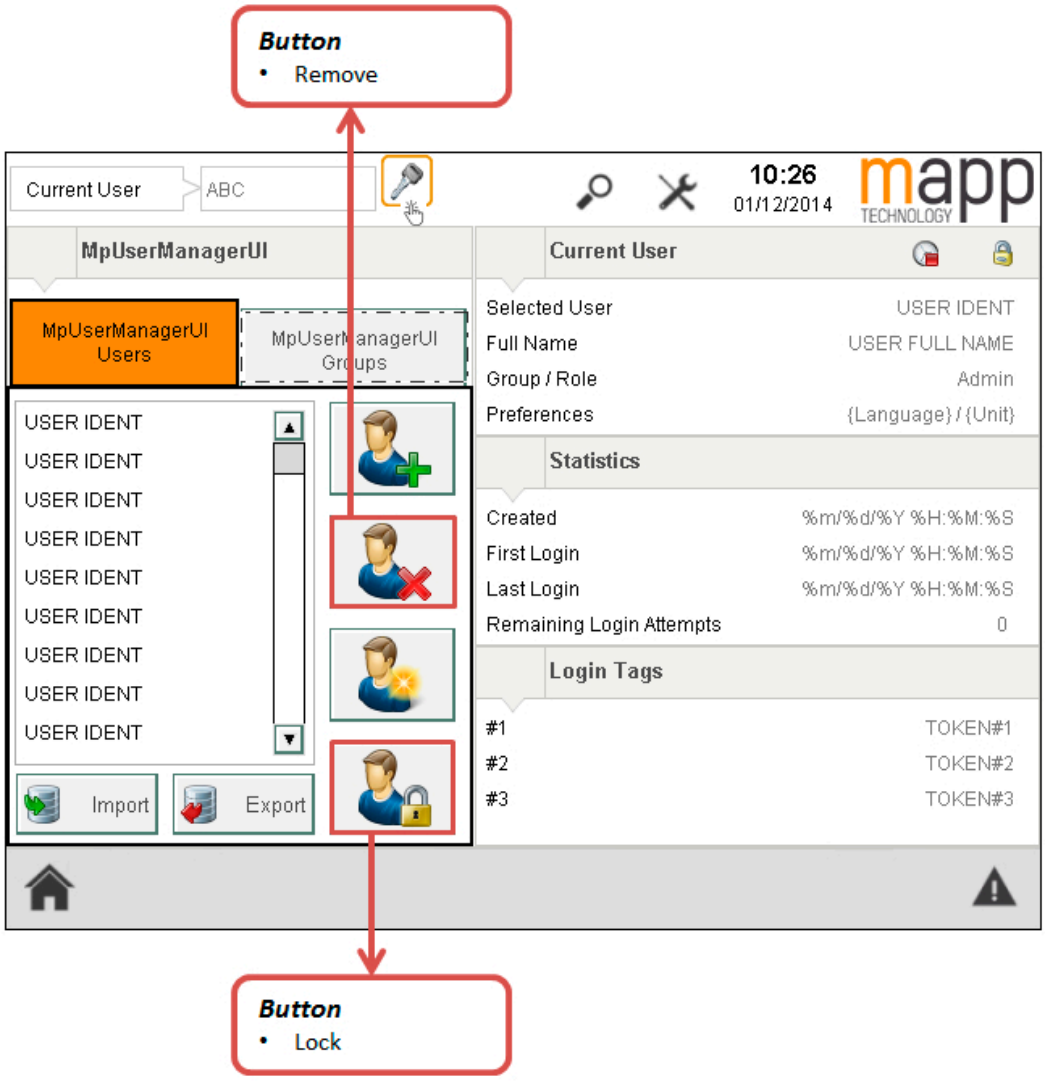
Parameter	Datentyp	Beschreibung
StepUp	BOOL	Wählt vorhergehenden Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepDown	BOOL	Wählt nächsten Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
RangeStart	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von *Scale*
RangeEnd	REAL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von *Scale*



8.5.2.1.32 MpUserMgrUIUserType

Dient zum Verwalten der Benutzer.

Parameter	Datentyp	Beschreibung
List	MpUserMgrUIUserListType	Liste aller Benutzer, sowie die Navigation für die Liste
Info	MpUserMgrUIUserInfoType	Enthält detailierte Benutzerdaten
Create	MpUserMgrUICreateType	Beinhaltet die Daten, welche zum Erstellen eines Benutzers notwendig sind
Edit	MpUserMgrUICreateType	Beinhaltet die Daten, welche zum Editieren eines Benutzers notwendig sind
Remove	BOOL	Löschen des ausgewählten Benutzers. Es können nur Benutzer gelöscht werden, welche auch zur Laufzeit erstellt wurden. Benutzer welche in der MpUser Konfiguration erstellt wurden werden nicht gelöscht sondern deaktiviert. **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Lock	BOOL	Sperren des ausgewählten Benutzers **VC4 Anbindung:** LockingDatapoint von *Button* / Locking: Datapoint >= Level / Level: 1



8.5.2.1.33 MpUserSignatureUIConnectType

Mithilfe dieses Datentypes wird eine Verbindung zwischen der MpUserSignatureUI Komponente und der Visual Components 4 Visualisierung erstellt. Über diese Struktur kann eine elektronische Signatur über die Visualisierung durchgeführt werden.

Parameter	Datentyp	Beschreibung
Status	MpUserUIStatusEnum	Aktuelle Operation
Sign	BOOL	Löst den Prozess rund um eine elektronische Signatur aus. Kann mit *Completion Datapoint* verbunden werden. So wird beispielsweise bei einer Eingabe für einen bestimmten Parameter der Dialog für eine elektronische Signatur geöffnet
Dialog	MpUserSignatureUIDlgType	Dialog, der zur Eingabe von Benutzernamen und Passwort auffordert
MessageBox	MpUserUIMessageBoxType	Bedienung der Dialogfenster
DefaultLayerStatus	UINT	Statusdatenpunkt für den Default Layer der Visualisierungsseite, von der sich ein- und ausgeloggt werden kann **VC4 Anbindung:** StatusDatapoint von *Layer*

8.5.2.1.34 MpUserSignatureUIDlgType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern. Über dieses Dialogfenster wird eine elektronische Signatur durchgeführt.

Parameter	Datentyp	Beschreibung
UserName	WSTRING[20]	Benutzername **VC4 Anbindung:** Datapoint von *String*
Password	WSTRING[20]	Passwort **VC4 Anbindung:** Datapoint von *String*
Comment	WSTRING[100]	Kommentar, welcher dem erzeugten Ereignis im Audit Trail hinzugefügt wird **VC4 Anbindung:** Datapoint von *String*
LayerStatus	UINT	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Confirm	BOOL	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	BOOL	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1

8.5.2.1.35 MpUserStatusIDType

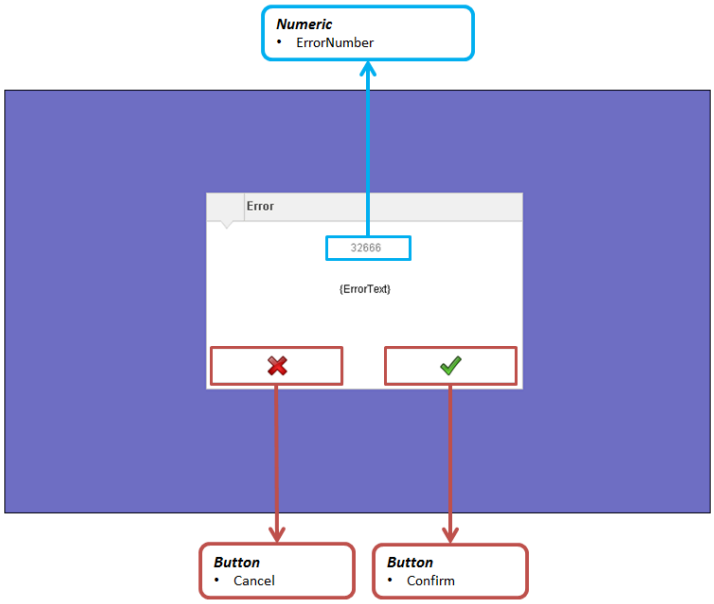
Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	MpUserErrorEnum	Fehlerkode der mapp Komponente
Severity	MpComSeveritiesEnum	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler)
Code	UINT	Kode der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden

8.5.2.1.36 MpUserUIMessageBoxType

Datentyp, der verwendet wird, um die Anzeige eines Dialogfensters zu steuern.

Parameter	Datentyp	Beschreibung
LayerStatus	UINT	Sichtbarkeit des Dialogfensters (Statusdatenpunkt für den Layer des Dialogfensters) **VC4 Anbindung:** StatusDatapoint von *Layer*
Type	MpUserUIMessageEnum	Art des Dialogfensters
ErrorNumber	UINT	Aktuelle Fehlernummer zum Anzeigen. Mehr Information zu den Fehlernummern kann in den Beschreibungen der einzelnen UI-Funktionsbausteine gefunden werden: • MpUserLoginUI • MpUserManagerUI • MpSignatureUI **VC4 Anbindung:** Datapoint von *Numeric*
Confirm	BOOL	Bestätigen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1
Cancel	BOOL	Abbrechen der Operation **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1



8.5.2.2 Enumeratoren

8.5.2.2.1 MpUserAccessRightEnum

Definiert die Zugriffsrechte der einzelnen Benutzer.

Aufzählung	Beschreibung
mpUSER_ACCESS_UNDEFINED	Zugriffsrecht nicht definiert
mpUSER_ACCESS_NONE	Keine Zugriffsrechte
mpUSER_ACCESS_VIEW	Lesezugriff
mpUSER_ACCESS_ACTUATE	Prozesse können gestartet / gestoppt werden, aber nicht editiert.
mpUSER_ACCESS_FULL	Volle Zugriffsrechte (Lese-/Schreibzugriff)

8.5.2.2.2 MpUserImportModeEnum

Dieser Enumerator zeigt die verschiedenen Import Modi

Aufzählung	Beschreibung
mpUSER_IMPORT_SKIP	Es werden keine Daten importiert
mpUSER_IMPORT_IGNORE_EXISITNG	Es werden alle neuen Daten (Benutzer/Benutzergruppen) importiert. Bereits bestehende Daten werden nicht überschrieben
mpUSER_IMPORT_OVERWRITE	Es werden alle Daten importiert. Bereits bestehende Daten werden überschrieben. Default Benutzer, welche in der Konfiguration MpUser angelegt wurden, werden dabei auch überschrieben und sind somit nicht mehr in der Konfiguration als Default Benutzer vorhanden
mpUSER_IMPORT_IGNORE_DEFAULT	Es werden alle Benutzerdaten importiert. Default Benutzer, welche bereits über die Automation Studio Konfiguration angelegt wurden, werden nicht überschrieben
mpUSER_IMPORT_OVERWRITE_ONLY	Es werden alle bereits existierenden Daten überschrieben. Neue Daten werden nicht importiert. Default Benutzer, welche in der Konfiguration MpUser angelegt wurden, werden dabei auch überschrieben und sind somit nicht mehr in der Konfiguration als Default Benutzer vorhanden
mpUSER_IMPORT_REMOVE_EXISTING	Es werden alle Daten importiert. Bereits vorhandene Daten, welche aber nicht in der importierten Datei vorkommen, werden gelöscht

8.5.2.2.3 MpUserMgrUIImportConfirmEnum

Dieser Aufzählungsdatentyp zeigt an, ob alle vorhandenen Benutzer oder Gruppe überschrieben werden sollen.

Aufzählung	Beschreibung
mpUSER_CONFIRM_USER	Überschreiben aller vorhandenen Benutzer
mpUSER_CONFIRM_GROUP	Überschreiben aller vorhandenen Gruppe

8.5.2.2.4 MpUserUIMessageEnum

Dieser Aufzählungsdatentyp zeigt an, wofür das Dialogfenster verwendet werden soll.

Aufzählung	Beschreibung
mpUSER_MSG_NONE	Kein Dialogfenster
mpUSER_MSG_ERROR	Dialog: Fehler
mpUSER_MSG_CONFIRM_DELETE	Dialog: Bestätige das Löschen eines Benutzers
mpUSER_MSG_CONFIRM_LOCK	Dialog: Bestätige das Sperren eines Benutzers
mpUSER_MSG_CONFIRM_UNLOCK	Dialog: Bestätige das Freigeben eines Benutzers
mpUSER_MSG_CONFIRM_GROUP_DELETE	Dialog: Bestätige das Löschen einer Benutzergruppe

8.5.2.2.5 MpUserUIPasswordCriteriaEnum

Dieser Enumerator zeigt an welchen Kriterien ein neues Passwort nicht entspricht.

Aufzählung	Beschreibung
mpUSER_PASSWORD_CRIT_NONE	Das Passwort stimmt mit allen Kriterien für ein neues Passwort überein
mpUSER_PASSWORD_CRIT_LENGTH	Das Passwort ist zu kurz
mpUSER_PASSWORD_CRIT_ALPHA	Das Passwort muss alphanumerische Zeichen enthalten
mpUSER_PASSWORD_CRIT_CASE	Das Passwort muss Groß- und Kleinbuchstaben enthalten
mpUSER_PASSWORD_CRIT_SPECIAL	Passwort muss Sonderzeichen enthalten

8.5.2.2.6 MpUserUIStatusEnum

Dieser Aufzählungsdatentyp gibt Auskunft über die aktuelle Aktivität der Komponente.

Aufzählung	Beschreibung
mpUSER_STATE_IDLE	Aktuell ist kein Prozess aktiv
mpUSER_STATE_WAIT_DLG	Warten auf Bestätigung oder Abbruch
mpUSER_STATE_EXECUTE	Ausführen eines Befehls
mpUSER_STATE_ERROR	Die letzte Operation hat einen Fehler generiert

8.5.3 Statusnummern

8.5.3.1 1083334690: MpUserLogin ist noch nicht aktiv

Beschreibung:

Diese Komponente kann erst aktiv werden, wenn MpUserLogin aktiv ist.

Reaktion:

Der Funktionsbaustein wird diese Information an dem Ausgang "StatusID" anzeigen.

Ursache / Abhilfe:

- MpUserLogin aktivieren

Konstante:

`mpUSER_INF_WAIT_LOGIN_FB`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI

8.5.3.2 -2137980917: Warnung beim Laden der Konfiguration

Beschreibung:

Warnung beim Laden der Konfigurationsdatei (in die PV). Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Konfigurationsdatei beschädigt
- Unterschied zwischen dem Inhalt der Konfigurationsdatei und der PV (z.B. mehr Daten in der Konfigurationsdatei als in der PV)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserConfig
- MpUserLoginConfig

Konstante:

`mpUSER_WRN_CONFIG_LOAD`

8.5.3.3 -2137890798: Passwortänderung erforderlich

Beschreibung:

Der Benutzer muss das Passwort ändern. Einloggen ist weiterhin möglich, weitere Funktionen sind jedoch gesperrt, bis das Passwort geändert wurde.

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Passwort ändern

Konstante:

`mpUSER_WRN_PASSWORD_CHANGE_REQ`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin

8.5.3.4 -2137890774: Key wird bereits verwendet

Beschreibung:

Der Key {2:Key} wird bereits verwendet. Der Wert {3:Value} wird nicht übernommen. {1:ErrorNumber}

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Zusätzliche Information

- {1:ErrorNumber}: Fehlernummer.
- {2:Key}: Key, welcher angegeben wurde.
- {3:Value}: Wert der übergeben wurde.

Ursache / Abhilfe:

- Anderen Key angeben

Konstante:

`mpUSER_WRN_KEY_DUPLICATE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserGetData

8.5.3.5 -1064297940: Fehler beim Lesen der Konfiguration

Beschreibung:

Fehler beim Lesen der Benutzer-Konfiguration für {2:UserName}. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:UserName}: Der Name des aktuellen Benutzers.

Ursache / Abhilfe:

- Überprüfen der Konfiguration

8.5.3.6 -1064297939: Fehler beim Speichern der Konfiguration

Beschreibung:

Fehler beim Speichern der Benutzer-Konfiguration für {2:UserName}. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:UserName}: Name des aktuellen Benutzers.

Ursache / Abhilfe:

- Überprüfen, ob korrekte Daten eingegeben wurden

8.5.3.7 -1064297938: Fehler beim Löschen der Konfiguration

Beschreibung:

Fehler beim Löschen der Benutzer-Konfiguration für {2:UserName}.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:UserName}: Name des aktuellen Benutzers.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

8.5.3.8 -1064297937: Fehler beim Lesen der Konfiguration

Beschreibung:

Fehler beim Lesen der Gruppen-Konfiguration für die Gruppe {2:GroupName}. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:GroupName}: Name der Gruppe.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

8.5.3.9 -1064297936: Fehler beim Speichern der Konfiguration

Beschreibung:

Fehler beim Speichern der Konfiguration für die Gruppe {2:GroupName}. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:GroupName}: Name der Gruppe.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

8.5.3.10 -1064297935: Fehler beim Löschen der Konfiguration

Beschreibung:

Fehler beim Löschen der Konfiguration für die Gruppe {2:GroupName}. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:GroupName}: Name der Gruppe.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

8.5.3.11 -1064297934: Fehler beim Wiederherstellen permanenter Daten

Beschreibung:

Beim Wiederherstellen von permanenten Daten aus der Datei {2:FileName} ist ein Fehler aufgetreten. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:FileName}: Name der Datei.

Ursache / Abhilfe:

- Die Datei wurde eventuell beschädigt.

8.5.3.12 -1064297933: Fehler beim Initialisieren der Konfiguration

Beschreibung:

Beim Initialisieren der Benutzer-Konfiguration ist ein Fehler aufgetreten. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

8.5.3.13 -1064297932: Fehler beim Importieren der Gruppe

Beschreibung:

Beim Importieren der Gruppe {2:GroupName} ist ein Fehler aufgetreten. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:GroupName}: Name der Gruppe.

Ursache / Abhilfe:

- Besteht kein Unterschied der bestehenden Gruppen am Zielsystem und der zu importierenden Gruppe, kann Gruppe nicht importiert werden

8.5.3.14 -1064297931: Fehler beim Importieren des Benutzers

Beschreibung:

Beim Importieren des Benutzers {2:UserName} ist ein Fehler aufgetreten. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:UserName}: Name des Benutzers.

Ursache / Abhilfe:

- Besteht kein Unterschied zu bereits vorhandenen Benutzern am Zielsystem und den zu importierenden Benutzer, kann Benutzer nicht importiert werden

8.5.3.15 -1064297923: Manager konnte nicht installiert werden.

Beschreibung:

Das User-Management konnte nicht registriert werden. Der Fehler kann auftreten wenn die Komponente MpUserX auf dem Zielsystem vorhanden ist. Fehlerursache: {1:ErrorNumber}

Ursache / Abhilfe:

- Auf dem Zielsystem darf MpUserX nicht vorhanden sein

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache

8.5.3.16 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserLoginConfig
- MpUserSignature
- MpUserSignatureUI

Konstante:

`mpUSER_ERR_ACTIVATION`

8.5.3.17 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserLoginConfig
- MpUserSignature
- MpUserSignatureUI

Konstante:

`mpUSER_ERR_MPLINK_NULL`

8.5.3.18 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserLoginConfig
- MpUserSignature
- MpUserSignatureUI

Konstante:

`mpUSER_ERR_MPLINK_INVALID`

8.5.3.19 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserLoginConfig
- MpUserSignature
- MpUserSignatureUI

Konstante:

`mpUSER_ERR_MPLINK_CHANGED`

8.5.3.20 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserSignature
- MpUserSignatureUI
- MpUserLoginConfig

Konstante:

`mpUSER_ERR_MPLINK_CORRUPT`

8.5.3.21 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserLoginConfig

Konstante:

`mpUSER_ERR_MPLINK_IN_USE`

8.5.3.22 -1064239097: Parameter Struktur ist ein Nullzeiger.

Beschreibung:

Der "Parameter" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "Parameter" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserConfig
- MpUserLoginConfig

Konstante:

`mpUSER_ERR_PAR_NULL`

8.5.3.23 -1064239096: Konfigurations Struktur ist ein Nullzeiger

Beschreibung:

Der "Configuration" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "Configuration" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserConfig
- MpUserLoginConfig

Konstante:

`mpUSER_ERR_CONFIG_NULL`

8.5.3.24 -1064239095: Konfigurations Zeiger ist keine PV

Beschreibung:

Der angegebene Zeiger ist kein Zeiger auf eine gültige PV. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "Configuration" Eingang prüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserConfig
- MpUserLoginConfig

Konstante:

`mpUSER_ERR_CONFIG_NO_PV`

8.5.3.25 -1064239094: Fehler beim Laden der Konfiguration

Beschreibung:

Fehler beim Laden der Konfigurationsdatei (in die PV). Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Konfigurationsdatei beschädigt
- Unterschied zwischen dem Inhalt der Konfigurationsdatei und der PV (z.B. unterschiedliche Datentypen, Daten außerhalb der gültigen Limits, ...)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserConfig
- MpUserLoginConfig

Konstante:

`mpUSER_ERR_CONFIG_LOAD`

8.5.3.26 -1064239092: Fehler beim Speichern der Konfiguration

Beschreibung:

Fehler beim Speichern in die Konfigurationsdatei (aus der PV). Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Konfigurationsdatei beschädigt
- Unterschied zwischen dem Inhalt der Konfigurationsdatei und der PV (z.B. unterschiedliche Datentypen, Daten außerhalb der gültigen Limits, ...)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserConfig
- MpUserLoginConfig

Konstante:

`mpUSER_ERR_CONFIG_SAVE`

8.5.3.27 -1064239091: Ungültige Konfiguration

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Konfiguration ist beschädigt
- Die Funktion ist in der Konfiguration nicht aktiviert

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin

Konstante:

`mpUSER_ERR_CONFIG_INVALID`

8.5.3.28 -1064148992: Passwort ungültig

Beschreibung:

Das eingegebene Passwort ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Passwort überprüfen
- Kontrollieren ob weitere Information im Logger verfügbar ist

Konstante:

`mpUSER_ERR_PASSWORD_INCORRECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserSignatureUI

8.5.3.29 -1064148991: Benutzer existiert nicht

Beschreibung:

Der eingegebene Benutzer existiert nicht

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzernamen überprüfen

Konstante:

`mpUSER_ERR_USER_NOT_EXISTING`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserSignatureUI

8.5.3.30 -1064148990: Benutzer exisitiert bereits

Beschreibung:

Der eingegebene Benutzername existiert bereits

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzername überprüfen und / oder ändern

Konstante:

`mpUSER_ERR_USER_EXISTS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI

8.5.3.31 -1064148989: Login Token ungültig

Beschreibung:

Der verwendete Login Token ist ungültig für den angegebenen Benutzer.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Login Token prüfen
- Benutzernamen prüfen

Konstante:

`mpUSER_ERR_LOGIN_ID_INVALID`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI

8.5.3.32 -1064148988: Keine Administratorrechte

Beschreibung:

Benutzer hat keine Administratorrechte.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzer ist kein Administrator
- Unzureichende Benutzerrechte

Konstante:

`mpUSER_ERR_ADMIN_INVALID`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.33 -1064148987: Ungenügende Benutzerrechte

Beschreibung:

Benutzerrechte sind für die gewünschte Operation unzureichend.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzer hat keine Freigabe für die gewünschte Operation
- Benutzer mit zu niedrigem Benutzerlevel hat versucht eine Signatur durchzuführen

Konstante:

`mpUSER_ERR_INSUFFICIENT_RIGHTS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI
- MpUserSignatureUI

8.5.3.34 -1064148986: Benutzergruppe existiert nicht

Beschreibung:

Die angegebene Benutzergruppe existiert nicht.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzergruppe überprüfen

Konstante:

`mpUSER_ERR_GROUP_NOT_PRESENT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI
- MpUserCreateUser

8.5.3.35 -1064148985: Passwort nicht sicher genug

Beschreibung:

Gewähltes Passwort ist nicht sicher genug.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Passwort ändern
- Benötigter Sicherheitsstandard für das Passwort kann über den Funktionsblock MpUserConfig festegelegt werden.

Konstante:

`mpUSER_ERR_PASSWORD_WEAK`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI

8.5.3.36 -1064148984: Benutzer gesperrt

Beschreibung:

Dieser Benutzer ist gesperrt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Admin muss diesen Benutzer wieder freischalten

Konstante:

`mpUSER_ERR_USER_IS_LOCKED`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI
- MpUserSignatureUI

8.5.3.37 -1064148983: Passwortänderung erforderlich

Beschreibung:

Der Benutzer muss das Passwort ändern, davor sind weitere Funktionen nicht möglich.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Passwort ändern

Konstante:

`mpUSER_ERR_PASSWORD_CHANGE_REQ`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI
- MpUserSignatureUI

8.5.3.38 -1064148982: Zuviele Login Versuche, Login nicht mehr möglich

Beschreibung:

Der Benutzer hat zu oft das falsche Passwort verwendet und wurde gesperrt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Admin muss Benutzer wieder freischalten

Konstante:

`mpUSER_ERR_TOO_MANY_LOGINS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI

8.5.3.39 -1064148981: Benutzer kat keine Adminrechte

Beschreibung:

Der eingeloggte Benutzer hat keine Adminrechte.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Admin kontaktieren

Konstante:

`mpUSER_ERR_NO_ADMIN_RIGHTS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.40 -1064148980: Benutzermanagement existiert bereits

Beschreibung:

In dieser mechatronischen Einheit existiert bereits ein Benutzermanagement. Eine weitere Instanz dieses Funktionsblocks ist nicht möglich.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- In eine andere Gruppe wechseln (Hierarchie Konzept)
- Die zusätzliche MpUserLogin Instanz entfernen

Konstante:

`mpUSER_ERR_LOGIN_INST_EXISTS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin

8.5.3.41 -1064148979: Kein Benutzer eingeloggt

Beschreibung:

Es ist kein Benutzer eingeloggt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Mithilfe von Benutzernamen und Passwort einloggen

Konstante:

`mpUSER_ERR_NOT_LOGGED_IN`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI

8.5.3.42 -1064148978: Funktionsindex übersteigt Maximum

Beschreibung:

Der Index für die gewählte Funktion übersteigt den maximalen Index, die Rechte können nicht abgefragt werden (reserviert für spätere Verwendung).

Reaktion:

Der Funktionsblock wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Eingang "RightIndex" überprüfen
- Index für gewählte Funktion zu hoch

8.5.3.43 -1064148977: Passwörter identisch

Beschreibung:

Das neue und das alte Passwort sind identisch.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Passwort ändern

Konstante:

`mpUSER_ERR_PASSWORD_IDENTICAL`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI

8.5.3.44 -1064148976: Benutzername existiert bereits

Beschreibung:

Der gewählte Benutzername für diesen Token existiert bereits

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Anderen Benutzernamen wählen

Konstante:

`mpUSER_ERR_LOGIN_ID_EXISTS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserRFID

8.5.3.45 -1064148975: Benutzer konnte nicht gelöscht werden

Beschreibung:

Der gewählte Benutzer konnte nicht gelöscht werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Weitere Details siehe Logger

Konstante:

`mpUSER_ERR_DELETE_USER`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.46 -1064148973: Benutzergruppenindex ist ungültig

Beschreibung:

Der angegebene Benutzergruppenindex ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzergruppenindex überprüfen

Konstante:

`mpUSER_ERR_GROUP_INDEX_NOT_OK`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI
- MpUserCreateGroup

8.5.3.47 -1064148972: Ungültiger Benutzergruppenname

Beschreibung:

Es wurde ein ungültiger Benutzergruppenname (NULL) übergeben.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzergruppenname ist NULL

Konstante:

`mpUSER_ERR_NO_GROUPNAME`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserCreateGroup
- MpUserCreateUser

8.5.3.48 -1064148971: Benutzer bereits eingeloggt

Beschreibung:

Der angegebene Benutzer ist bereits eingeloggt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzernamen überprüfen

Konstante:

`mpUSER_ERR_LOGGED_IN`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLogin
- MpUserLoginUI
- MpUserManagerUI

8.5.3.49 -1064148970: Fehlender Wert an UIConnect

Beschreibung:

An "UIConnect" wurde NULL angehängt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Eingang "UIConnect" vergessen

Konstante:

`mpUSER_ERR_MISSING_UICONNECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI
- MpUserSignatureUI

8.5.3.50 -1064148969: Ungültiger Benutzername

Beschreibung:

Es wurde ein ungültiger Benutzername (NULL) übergeben.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzername ist NULL

Konstante:

`mpUSER_ERR_NO_USERNAME`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI
- MpUserCreateUser
- MpUserSignatureUI

8.5.3.51 -1064148968: Passwort fehlt

Beschreibung:

Es wurde kein Passwort eingegeben.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Passwort eingeben

Konstante:

`mpUSER_ERR_NO_PASSWORD`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserCreateUser
- MpUserSignatureUI

8.5.3.52 -1064148967: MpUserLogin fehlt

Beschreibung:

Der Funktionsblock MpUserLogin fehlt oder ist falsch konfiguriert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- MpUserLogin überprüfen

Konstante:

`mpUSER_ERR_MISSING_LOGIN`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI

8.5.3.53 -1064148966: Passwort zum Bestätigen falsch

Beschreibung:

Das zur Bestätigung eingegebene Passwort stimmt nicht mit dem neuen Passwort überein.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Passwörter überprüfen

Konstante:

`mpUSER_ERR_PASSWORD_CONFIRM`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserLoginUI
- MpUserManagerUI

8.5.3.54 -1064148965: Angegebenes Systemrecht existiert nicht

Beschreibung:

Das angegebene Systemrecht wurde nicht gefunden (reserviert für spätere Verwendung).

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Gewähltes Systemrecht überprüfen

Konstante:

`mpUSER_ERR_NO_SUCH_SYSRIGHT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.55 -1064148964: Import fehlgeschlagen

Beschreibung:

Der Import von Benutzergruppen und/oder Benutzerdaten ist fehlgeschlagen. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Importdatei überprüfen
- Den Inhalt der angegebenen Datei auf Richtigkeit und Vollständigkeit prüfen

Konstante:

`mpUSER_ERR_IMPORT_DATA`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.56 -1064148963: Fehler beim Laden der Importdatei

Beschreibung:

Beim Laden der Importdatei ist ein Fehler aufgetreten. Weitere Details siehe Logger.

Ursache / Abhilfe:

- Importdatei überprüfen
- Datei existiert nicht
- Speichermedium existiert nicht

Konstante:

`mpUSER_ERR_LOAD_IMPORT_FILE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.57 -1064148962: Fehler beim Speichern der Exportdatei

Beschreibung:

Beim Speichern der Exportdatei ist ein Fehler aufgetreten. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Exportdatei überprüfen
- Datei existiert nicht
- Speichermedium existiert nicht
- Überprüfen ob im Logger die Fehlernummer -1064303598 angezeigt wird. Ist dies der Fall, so ist ein Passwort eines Benutzers ungültig. Für mehr Information siehe Fehler -1064303598.

Konstante:

`mpUSER_ERR_SAVE_EXPORT_FILE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.58 -1064148961: Name der Benutzergruppe ist zu lange

Beschreibung:

Der Name der eingegebenen Benutzergruppe ist zu lange (> 20 Zeichen).

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Name der Benutzergruppe kürzen. Maximum sind 20 Zeichen

Konstante:

`mpUSER_ERR_GROUPNAME_TOO_LONG`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserCreateGroup
- MpUserCreateUser

8.5.3.59 -1064148960: Name des Benutzers ist zu lange

Beschreibung:

Der Name des eingegebenen Benutzers ist zu lange (> 20 Zeichen).

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Name des Benutzers kürzen. Maximum sind 20 Zeichen

Konstante:

`mpUSER_ERR_USERNAME_TOO_LONG`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserCreateUser

8.5.3.60 -1064148959: Name des Passwortes ist zu lange

Beschreibung:

Der Name des eingegebenen Passwortes ist zu lange (> 20 Zeichen).

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Name des Passwortes kürzen. Maximum sind 20 Zeichen

Konstante:

`mpUSER_ERR_PASSWORD_TOO_LONG`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserCreateUser

8.5.3.61 -1064148957: Name des Benutzers ist zu kurz

Beschreibung:

Der eingegebene Benutzername ist zu kurz.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Längeren Benutzernamen auswählen

Konstante:

`mpUSER_ERR_USERNAME_TOO_SHORT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.62 -1064148956: Benutzergruppe exisitiert bereits

Beschreibung:

Die eingegebene Benutzergruppe existiert bereits

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Benutzergruppe überprüfen und / oder ändern

Konstante:

`mpUSER_ERR_GROUP_EXISTS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.63 -1064148955: Benutzergruppe kann nicht gelöscht werden

Beschreibung:

Default Benutzergruppen können nicht gelöscht werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Default Benutzergruppen können nicht gelöscht werden

Konstante:

`mpUSER_ERR_GROUP_DEFAULT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.64 -1064148954: Signieren fehlgeschlagen

Beschreibung:

Maximale Anzahl an Signatur-Versuchen überschritten. Der Signatur Vorgang wurde unterbrochen

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Angabe der richtigen Aktions-ID

Konstante:

`mpUSER_ERR_MAX_FAILED_SIGNATURES`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserSignatureUI

8.5.3.65 -1064148953: Letzte Signatur wurde noch nicht geprüft

Beschreibung:

Es wurde eine neue Signatur gestartet, bevor die letzte signierte Aktion ausgeführt wurde.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Letze Signatur muss mithilfe von MpUserSignature erfolgreich ausgeführt werden (Released = `TRUE`)

Konstante:

`mpUSER_ERR_SIGNATURE_BUSY`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserSignatureUI

8.5.3.66 -1064148952: Passwort wurde bereits verwendet

Beschreibung:

Das eingegebene Passwort wurde bereits verwendet und kann nicht nocheinmal angegeben werden

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Angabe eines anderen Passworts

Konstante:

`mpUSER_ERR_PASSWORD_REPEAT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI
- MpUserLoginUI

8.5.3.67 -1064148951: Ungültige Import Datei

Beschreibung:

Die Datei welche Importiert werden soll, besitzt ein ungültiges Format.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen des Dateiformats

Konstante:

`mpUSER_ERR_IMPORT_FILE_FORMAT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserManagerUI

8.5.3.68 -1064148949: Information nicht gefunden

Beschreibung:

Die angefragte Information wurde nicht gefunden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen des angegebenen Keys
- Überprüfen der angefragen Information

Konstante:

`mpUSER_ERR_INFO_NOT_FOUND`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserGetData

8.5.3.69 -1064148948: Key ist leer

Beschreibung:

Der angegebene Key ist leer.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Angeben eines gültigen Keys

Konstante:

`mpUSER_ERR_EMPTY_KEY`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserGetData

8.5.3.70 -1064148947: Zu kleiner Buffer

Beschreibung:

Es wurde ein zu kleiner Buffer festgelegt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Festlegen eines größeren Buffers.

Konstante:

`mpUSER_ERR_BUFFER_TOO_SMALL`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpUserGetData

8.5.4 Alarme

8.5.4.1 mpUSER_ALM_USER_LOCKED: Benutzer wurde aufgrund zu vieler fehlerhafter Loginversuche gesperrt

Beschreibung:

Der Benutzer wurde nach zu vielen fehlerhaften Loginversuchen gesperrt.

Reaktion:

Wenn eine MpAlarmX Komponente aktiv ist, wird ein mapp Alarm ausgelöst.

Ursache / Abhilfe:

- Zum Wiederfreischalten des Benutzers muss der Administrator kontaktiert werden

Verhalten

Edge Alarm

9 mapp Alarm: VC4 Alarmverwaltung

mapp Alarm ist als Support für eine Alarmverwaltung mit Visual Components 4 konzipiert. Alarme werden gesammelt und ausgewertet.

9.1 Konzept

Allgemein

- Als Applikationist möchte ich meine Applikation mit dem VC4 Alarmsystem verbinden
- Als Applikationist möchte ich dasselbe Alarmsystem verwenden, welches mapp Technology auch intern nützt

Alarme visualisieren

- Als Maschinenbediener möchte ich informative und einfache Alarmmeldungen in meiner eigenen Sprache
- Als Maschinenbediener möchte ich bei der Anzeige von mehreren Alarmen immer eindeutig sehen, welche die wichtigsten bzw. kritischsten aktiven Alarme sind
- Als Maschinenbediener oder als Servicetechniker möchte ich meine Alarme logisch gruppiert wissen

9.1.1 Alarm Zustand

Ein Alarm kann die folgenden Zustände annehmen:

- **Inaktiv**: Der Alarm ist nicht aktiv
- **Aktiv**: Der Alarm ist aktiv
- **Zurückgesetzt**: Der Alarm ist nicht mehr aktiv, wurde jedoch nicht quittiert
- **Quittiert**: Der Alarm wurde quittiert, aber er ist noch immer aktiv

Als Beispiel dient der Wasserstand eines Tanks. Solange der Stand den Maximalwert nicht überschritten hat, ist der Alarm inaktiv. Sobald er überschritten wurde, wird der Alarm aktiv. Der Maschinenbediener sieht den aktiven Alarm und quittiert ihn, der Wasserstand ist jedoch immer noch zu hoch. Würde stattdessen der Alarm vom Maschinenbediener nicht quittiert und der Wasserstand sinkt wieder unter den Schwellwert, dann würde der Alarm zurückgesetzt werden.

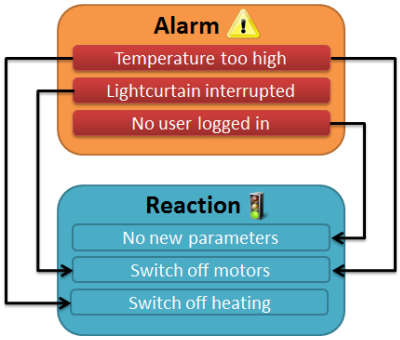
9.1.2 Reaktion

Als Reaktion bezeichnet man das Verhalten des Systems auf einen neu eintretenden Zustand (z.B. Wasserstand niedrig, Reaktion könnte das Blinken der Warnlampe sein). Reaktionen können für jeden Alarm definiert werden. Zu definieren sind:

- **Mögliche Reaktionen:** Sofort stoppen, Keine neuen Parameter akzeptieren, Motor aus, Heizung aus, ...
- **Reaktionen jedes Alarms:** Welcher Alarm löst welche der obigen Reaktion(en) aus

Der Funktionsbaustein MpAlarmBasic sammelt die Alarme aller verbundenen Komponenten und liefert für diese eine entsprechende Reaktion (bitkodiert). Auf diese Reaktion kann in der Applikationssoftware entsprechend reagiert werden. Die Konfiguration der Reaktionen und Zuweisung zu Alarmen erfolgt über den WebXs.

Die Reaktion ist applikationsspezifisch umzusetzen.



1. Alarm **Lightcurtain interrupted** wird aktiv
2. Alarm löst definierte Reaktion aus - **Switch off motors**
3. Der Benutzer sieht die Reaktion und stoppt den Motor mit seiner Software

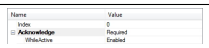
1. Alarm **Temperature too high** wird aktiv
2. Alarm löst definierte Reaktion aus - **Switch off motors** and **Switch off heating**
3. Der Benutzer sieht die Reaktion und stoppt den Motor und schaltet die Heizung aus mit seiner Software

1. Alarm **No user logged in** wird aktiv
2. Alarm löst definierte Reaktion aus - **No new parameters**
3. Der Benutzer sieht die Reaktion und sperrt die Eingabefelder auf der Visualisierung

9.1.3 Unstetige und anhaltende Alarme

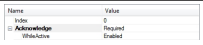
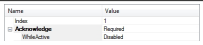
Ein **unstetiger** Alarm (*Edge* Alarm) ist ein einzelnes Ereignis, welches nur kurzzeitig ansteht. Ein Beispiel wäre die Nachricht, dass eine Datei nicht geöffnet werden konnte.

Unstetige Alarme werden in der Visual Components 4 Visualisierung folgendermaßen konfiguriert:

Für alle unstetigen Alarme gilt: • Acknowledge: Required • WhileActive: Enabled Beispiel: *Die Datei konnte nicht erfolgreich geöffnet werden*	

Ein **anhaltender** Alarm (*Persistent* Alarm) stellt einen System Zustand dar. Ein Beispiel wäre ein zu niedriger Tank Level. Dieser Alarm steht solange an, bis der Tank wieder voll ist.

Anhaltende Alarme werden in der Visual Components 4 Visualisierung folgendermaßen konfiguriert:

Folgende Einstellungen gelten für anhaltende Alarme, bei denen eine Quittierung erforderlich ist: • Acknowledge: Required • WhileActive: Enabled Beispiel: *Die Temperatur ist im kritischen Bereich, Heizvorgang wurde unterbrochen*	
Folgende Einstellungen gelten für anhaltende Alarme, bei denen eine Quittierung erforderlich ist (Quittierung aber erst möglich nachdem der Alarm inaktiv ist): • Acknowledge: Required • WhileActive: Disabled Beispiel: *Sicherheitstür wurde geöffnet, Motoren wurden gestoppt*	
Folgende Einstellungen gelten für anhaltende Alarme, bei denen eine Quittierung nicht erforderlich ist: • Acknowledge: Possible • WhileActive: Enabled Beispiel: *Der Wassertanklevel ist niedrig*	

9.1.4 Fehler, benutzerdefinierte Alarme und mapp Alarme

Grundsätzlich muss man im Umgang mit mapp Komponenten zwischen Fehlern und Alarmen unterscheiden. Fehler werden über die S"tatusID" am Ausgang einer Funktion / eines Funktionsbaustein angezeigt, bzw. im Logger eingetragen. Die "StatusID" dient zum diagnostizieren, was beim Entwicklen der Applikation nicht funktioniert hat, bzw. später um eine detailierte Diagnose vornzunehmen.

Alarme geben dem Maschinenbediener Auskunft über den Zustand der Maschine. Alarme sollten auf eine Art und Weise verfasst sein, dass der Maschinenbediener damit etwas anfangen kann. Das Alarmhandling ist nicht dazu gedacht, eine detailierte Diagnose durchzuführen (z.B. "StatusID" auswerten, ACOPOS Fehlertexte anzuzeigen, ...).

Alarme und Fehler sind grundsätzlich getrennt voneinander zu betrachten. Alarme müssen nicht die Konsequenz eines Fehler sein sondern können auch eigenständig auftreten (z.B. *Benutzer wurde gesperrt wegen falscher Passworteingabe*, ein Alarm wird ausgelöst). Fehler geben Auskunft über die falschen Anwendung einer Hardware bzw. Software und können parallel zu einem Alarm auftreten, muss aber nicht so sein (z.B. *Laden eines Rezepts*, wird über die "StatusID" angezeigt, außerdem wird eine Alarm erzeugt). Folgende Tabelle zeigt einige Unterschiede auf

mapp Komponenten Fehler	
mapp Komponenten Alarm	Standard Alarm
	Benutzer Alarm

9.1.5 Alarmzuordnung

Um die Alarme über die Visualisierung anzuzeigen ist es nötig eine Verbindung zur VC4 Visualisierung zu erstellen. Diese Verbindung bilden das Alarm- und Quittierabbild. Wichtig ist für den Benutzer zu wissen wie die Alarme im Alarmabbild bzw. Quittierabbild angeordnet sind. Beim ersten Hochstarten wird die Anordnung von MpAlarm automatisch erstellt. Die Anordnung der Alarme im Alarm- bzw. Quittierabbild am Funktionsbaustein MpAlarmBasicUI hängt von zwei verschiedenen Einstellungen ab:

Der Alarmquelle

Die Quelle hängt vom MpLink ab, welcher mit MpAlarmBasicUI verbunden wird. Abhängig von der Quelle verändert sich die Anzahl an Alarmen die über eine (oder mehrere) Alarmgruppen dargestellt werden sollen.

- Wird MpAlarmBasicUI über den MpLink an eine einzelne mapp Komponente gehängt wird so werden nur Alarme dieser einen Komponente in das Alarmabbild übernommen. Die Alarmzuordnung spielt in diesem Fall keine Rolle. Bei dieser Variante (dezentrales Alarmhandling) gibt es keine spezielle Anordnung, es gibt eine Alarmgruppe pro mapp Komponente (siehe als Beispiel Dezentrale Anwenderalarmverwaltung)
- Wird MpAlarmBasicUI über den MpLink an MpAlarmBasic gehängt, so werden alle Alarme in einer Gesamtliste angezeigt. In diesem Fall muss der Benutzer die Reihenfolge wissen in der die Alarme in der Gesamtliste eingetragen werden, damit eine entsprechende Verbindung zur VC4 Visualisierung möglich ist. In diesem Fall ist nur eine Alarmgruppe nötig (siehe als Beispiel Zentrale Alarmverwaltung)

Der Modus

In welchem Modus MpAlarmBasicUI betrieben wird ist ein weiterer Faktor der die Größe und Zuordnung des Alarmabbilds beeinflusst. Entweder es werden nur mapp Alarme oder nur Benutzeralarme in die Liste übernommen, oder aber alle Alarme.

- Im Modus mpALARM_UI_MODE_USER bzw. mpALARM_UI_MODE_MAPP werden die jeweiligen Alarme in die Gesamtliste übernommen. Für den Modus mpALARM_UI_MODE_USER müssen die mapp Alarme nicht berücksichtigt werden bei der Berechnung der Größe für das Alarmabbild / Quittierabbild. Für den Modus mpALARM_UI_MODE_MAPP müssen die Benutzeralarme nicht berücksichtigt werden bei der Berechnung der Größe für das Alarmabbild / Quittierabbild.
- Im Mode mpALARM_UI_MODE_ALL werden die Benutzeralarme und die mapp Alarmen in die Liste mit aufgenommen. Dabei werden die Benutzeralarme immer nach den mapp Alarmen angeordnet

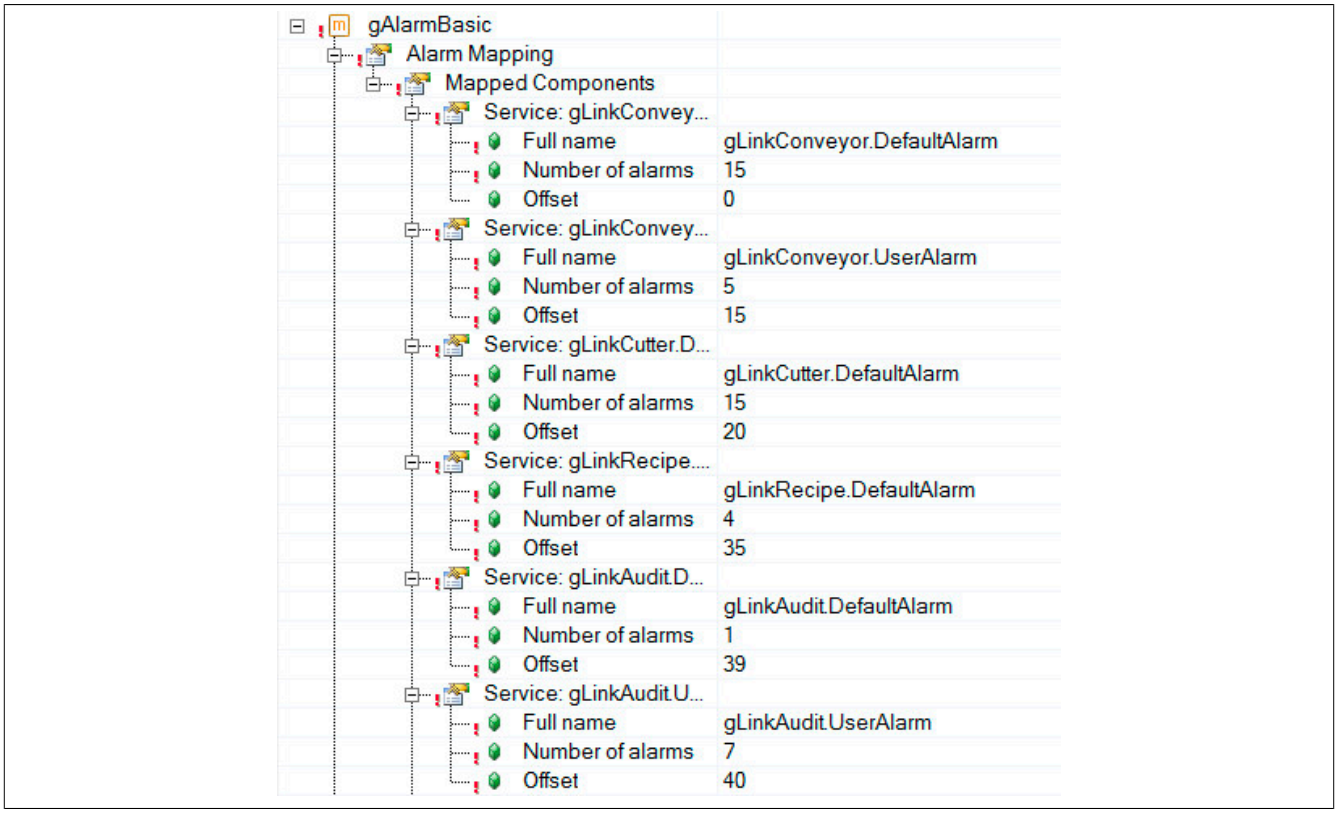
Mithilfe dieser Punkte ist man in der Lage zu erklären, wie sich das Alarmabbild bzw. Quittierabbild zusammensetzt. Eine Gesamtliste ist auch über den WebXs verfügbar.

Alarmzuordnung über die Konfiguration

Man hat jedoch auch die Möglichkeit die Alarmzuordnung im Alarm- bzw. Quittierabbild zu beeinflussen über die MpAlarmBasic Konfiguration. Das macht nur Sinn wenn man mit einer zentralen Alarmverwaltung arbeitet. Dafür gibt es in der Konfiguration den Punkt *Mapped Components*. Dieser Punkt ist nur sichtbar, wenn die erweiterten Paramter eingeblendet werden. Hier kann man selbst bestimmen wie groß das Alarm- bzw. Quittierabbild wird, indem man die Alarme der einzelnen Komponenten mit einem frei definierbaren Offset im Alarm- / Quittierabbild ablegt. Dabei werden folgende Daten konfiguriert:

- **Full name:** Name der mapp Komponente, für welche die Alarme abgelegt werden sollen. Zusätzlich muss angegeben werden ob die mapp Alarme (*.DefaultAlarm*) oder die Benutzeralarme (*.UserAlarms*) gemappt werden sollen. Alarme, egal ob mapp Alarme (*.DefaultAlarm*) oder Benutzeralarme (*.UserAlarm*), die nicht explizit gemappt wurde, werden automatisch von MpAlarmBasic am Ende des Alarm- bzw. Quittierabbilds angehängt
- **Number of Alarms:** Abhängig vom Modus, der über MpAlarmBasicUI ausgewählt ist. Jede mapp Komponente hat ihre eigenen Alarme, welche im jeweiligen Hilfe Abschnitt zu finden sind. Zusätzlich kann der Benutzer noch seine eigenen Alarme anfügen. MpAxis beispielsweise hat 15 mapp Alarme, bedeutet soll diese mapp Komponente gemappt werden, so muss zumindest Platz für 15 Alarme in Betracht gezogen werden. Kommen Benutzeralarme dazu, erhöht sich diese Zahl nocheinmal.
- **Offset:** Bestimmt an welcher Position im Alarmabbild die Alarme abgelegt werden. Stellt man hier einen Wert von 100 ein, so fangen die Alarme dieser mapp Komponente an im Alarmabbild / Quittierabbild an der Stellt 100 im Array. Die Plätze 0..99 sind dann anderweitig verfügbar. Lücken sind kein Problem. Würde man also die 100 übersprungenen Plätze nicht vergeben, führt das zu keinem Fehler

Verwendet man den Punkt *Mapped Components*, wo wird beim Hochstarten die Alarmliste nicht automatisch generiert, sondern abhängig von den Einstellungen in der MpAlarmBasic Konfiguration. Damit hat man den Vorteil nichts berechnen zu müssen. Folgendes Beispiel zeigt eine mögliche Konfiguration:



Es werden die Alarme für vier mapp Komponenten gemappt. Dabei werden bei allen vier Komponenten die mapp Alarme gemappt, aber nur für zwei davon, gLinkConveyor und gLinkAudit, auch Benutzeralarme. Das Offset ist so gewählt, dass an der Position an der die Alarme einer Komponenten enden, die Alarme der nächsten beginnen.

> **Verletzt ein gemappter Alarm den Index eines anderen Alarms, überschreibt der neu gemappte Alarm den vorhergehenden.**

9.1.6 Rahmenbedingungen

In diesem Abschnitt werden Rahmenbedingungen von mapp Alarm aufgelistet:

Verwendung des Alarmsystems

In einem Projekt darf man entweder mit mapp Alarm oder mit mapp AlarmX arbeiten. Nicht mit Beidem parallel. mapp Alarm ist als Erweiterung für das Alarmsystem in VC4 gedacht.

mapp AlarmX ist ein eigenständiges Alarmsystem, unabhängig von einer Visualisierung. Weiters bietet mapp AlarmX einen großen Funktionsumfang, welcher die Konfiguration des Alarmsystems, sowie erweiterte Funktionen zur Laufzeit anbietet.

9.2 Konfiguration

9.2.1 MpAlarmBasic Konfiguration

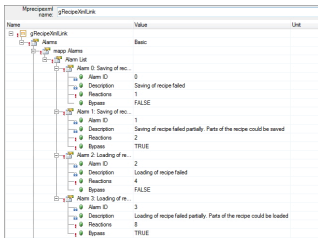
Beim einfügen von mapp Komponenten wird eine Konfiguration erstellt. In dieser Konfiguration können die mapp Alarme und die benutzerdefinierten Alarme einer mapp Komponente konfiguriert werden. Die Alarmlisten sind immer am Ende der Konfiguration zu finden und unterteilen sich in:

- **mapp Alarms:** vordefinierte Alarme, welche von der mapp Komponente ausgelöst werden können
- **User Alarms:** Möglichkeit eigene Alarme zu konfigurieren, welche dieser mapp Komponente zugeordnet sind. Die Benutzeralarme müssen außerdem über MpAlarmRegUser zur Konfiguration hinzugefügt werden

> **Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.**
>
> **Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein MpAlarmBasic de-, und wieder aktiviert werden.**

mapp Alarme

Die meisten mapp Komponenten haben vordefinierte mapp Alarme, welche von der Komponente ausgegeben werden, falls ein Problem aufritt. Die Nummern und die Texte von mapp Alarmen sind festgelegt und können nicht verändert werden. Die Reaktion und die Bypass Option können aber frei gewählt werden.

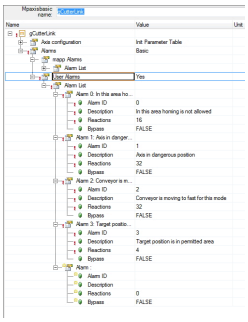


Es gibt Komponente, welche keine eigenen mapp Alarme besitzen. Bei diesen bleibt der Unterpunkt *mapp Alarms* in der Konfiguration leer.

User Alarms

Wenn gewünscht können auch Benutzeralarme bei den einzelnen mapp Komponenten konfiguriert werden. Diese müssen über MpAlarmRegUser in der Applikation zur Konfiguration hinzugefügt werden:

- **Alarm ID:** Die Nummer des Alarms. Wenn selbst Alarme hinzugefügt werden sollte die ID **immer** bei 0 starten und fortlaufend nach oben gezählt werden. Dadurch entsteht kein Konflikt mit den mapp Alarmen, da die getrennt betrachtet werden. Werden mapp Alarme und Benutzeralarme über eine Liste ausgegeben, so starten die Benutzeralarme nach den mapp Alarmen.
- **Description:** Beschreibungstext für den Alarm.
- **Reactions:** Reaktion des Alarms.
- **Bypass:** Option den Alarm zu ignorieren.



Alarmzuordnung über die Konfiguration

Es ist möglich über den ausgeblendeten Punkt *Mapped Components* die Reihenfolge der Alarme im Alarm- bzw. Quittierabbild zu beeinflussen. Weitere Details zu dem Punkt sind im Abschnitt Alarmzuordnung zu finden.

Reaktionen

Im Abschnitt "Reaction handling" kann eine Reaktion auf einen Alarm konfiguriert werden. Für mehr Information siehe Abschnitt Reaktion. Ob eine Reaktion aktiv ist kann mithilfe des Funktionsbausteins MpAlarmReaction überprüft werden.

9.3 Anwendungsfälle

In diesem Abschnitt werden mögliche Anwendungsfälle der MpAlarm Komponenten beschrieben.

9.3.1 Zentrale Alarmverwaltung

Anforderung

Zwei Förderbänder transportieren PET-Rohlinge durch einen Heizofen. Die Förderbänder werden mit je einer MpAxisBasic Komponente gesteuert. Die Temperaturregelung des Heizofens erfolgt über die Applikation. Wenn die Temperatur einen bestimmten Wert überschreitet, soll ein Alarm ausgelöst werden. Die Alarme der Achsen und der Alarm der Übertemperatur sollen gesammelt und auf der VC4 Visualisierung angezeigt werden.

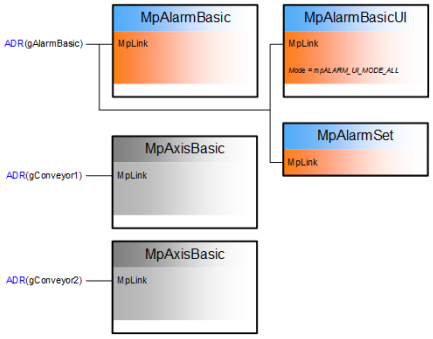
Der Einfachkeit halber sollen alle Alarme dieser Maschine in einer VC4 Alarmgruppe dargestellt werden, also zentral gesammelt und dargestellt werden.

Lösung

Komponentenliste

- ***MpAxisBasic*** (Eigener MpLink): Kontrolle der Förderbänder
- ***MpAlarmBasic*** (Eigener MpLink): Zentrale Alarmverwaltung
- ***MpAlarmBasicUI*** (MpLink von MpAlarmBasic): Verbindung der Alarme zentral zu Visual Components mittels eines Alarmabbilds. Der MpLink von MpAlarmBasic wird verwendet, damit entsteht eine zentrale Alarmkollektion (mit bestimmten Modus), welche mit einer Alarmgruppe in VC4 verbunden wird
- ***MpAlarmSet*** (MpLink von MpAlarmBasic): Setzen eines Anwenderalarms

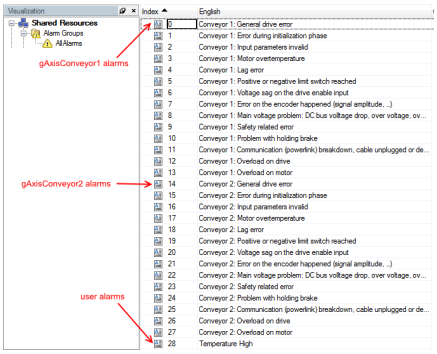
Verbindungsschema



Verwendung der mapp Komponenten

Die **MpAlarmBasicUI** Komponente kann über den MpLink mit der MpAlarmBasic Komponente verbunden werden. Im Modus mpALARM_UI_MODE_ALL sind dann mapp Alarme und Anwender-Alarme im gleichen Alarmabbild enthalten. Dabei werden Anwenderalarme immer an die Standard Alarme hinten angehängt.

Das bedeutet, dass in Visual Components eine Alarmgruppe mit alle Alarmen erzeugt werden muss:



Die Größe der Alarmgruppe ergibt sich aus der Anzahl an Alarmen im System. In diesem Fall:

ImageSize = MpAlarmBasic Alarme + MpAxisBasic Alarme + MpAxisBasic Alarme

130 Alarme = 100 Alarme + 15 Alarme + 15 Alarme

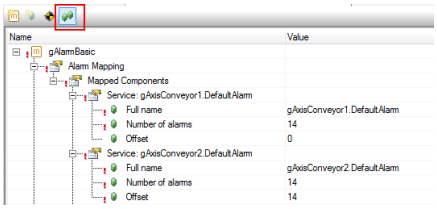
Die Alarmgruppe müsste also 130 Elemente besitzen, genau wie die Größe von Alarm- und Quittierabbild 130 sein muss.

Implizite Alarm Reihenfolge:

Nach dem Erstellen einer neuen Compact Flash und dem ersten Hochlaufen der Steuerung werden die mapp Alarme entsprechend der Freigabe Sequenz (Aktivierung) aller mapp Komponenten geordnet. Wenn neue mapp Komponenten hinzugefügt werden, werden diese am Ende der existierenden Liste angehängt.

Explizite Alarm Reihenfolge:

Alternativ kann der Benutzer explizit die Reihenfolge aller Alarme für das Alarm Array mittels der erweiterten Parameter der MpAlarmBasic Konfiguration festlegen:



Im Vergleich zu einer dezentralen Anbindung von MpAlarmBasicUI an einzelne mapp Komponenten (die mehrere Alarmabbilder ergeben würde) gibt es folgende Vor- und Nachteile.

Vorteile:

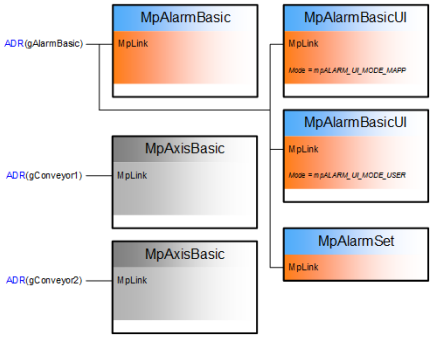
- Nur eine FB-Instanz von MpAlarmBasicUI benötigt
- Nur eine Alarmgruppe in Visual Components. Dies bietet einen besseren Überblick über alle Maschinenalarme

Nachteile:

- Wenn neue mapp Komponenten hinzugefügt werden, verschieben sich die Offsets der Anwenderalarme
- Zusätzlicher Aufwand durch Verwaltung der Offsets in der Konfiguration

Alternative: Trennung von mapp und Anwenderalarmen

Die Indexverschiebung der Anwenderalarme kann durch Trennen der Alarme und durch die Verwendung zweier Instanzen von MpAlarmBasicUI erreicht werden. Dafür wird eine Instanz mit Modus mpALARM_UI_MODE_MAPP und die andere mit mpALARM_UI_MODE_USER verwendet. In Visual Components wird dann für jede MpAlarmBasicUI eine Alarmgruppe erstellt.



9.3.2 Alarm Reaktionen

Anforderung

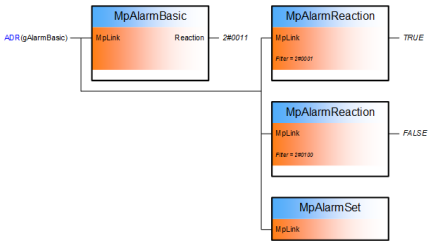
Die Temperatur eines Heizofens soll in der Applikation überwacht werden. Wenn die Temperatur zu hoch wird, soll eine rote Lampe aufleuchten und der Motor soll abgeschaltet werden.

Lösung

Komponentenliste

- ***MpAlarmBasic*** (Eigener MpLink): Zentrale Alarmverwaltung
- ***MpAlarmReaction***(MpLink von MpAlarmBasic): Kann wahlweise verwendet werden, um Reaktionen vor Ort zu bewerten (z.B. in anderen Programtasks als die der MpAlarmBasic Komponente)
- ***MpAlarmSet*** (MpLink von MpAlarmBasic): Setzen eines Anwenderalarms (z.B. Temperatur zu hoch)

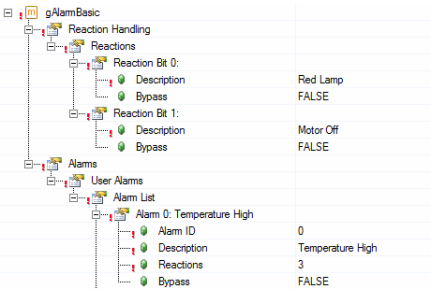
Verbindungsschema



Verwendung der mapp Komponenten

Jeder Alarm (Anwenderalarme und mapp Alarme) können mehrere Reaktionen auslösen und eine Reaktion kann durch mehrere Alarme ausgelöst werden. Die Konfiguration, welche Reaktion von einem Alarm auslöst wird, muss im Automation Studio durchgeführt werden.

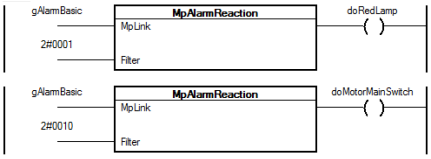
Konfiguration im Automation Studio:



MpAlarm unterstützt insgesamt 32 Reaktionen.

Diese werden am MpAlarmBasic als bitcodierter 4-Byte-Wert angezeigt und können zusätzlich mit der Funktion MpAlarmReaction ausgewertet werden.

Der Benutzer kann über die Bedeutung der einzelnen Reaktion entscheiden. Das heißt, im Applikationsprogram wird festgelegt, wie die Maschine schlussendlich reagieren soll:



9.3.3 Dezentrale Anwenderalarmverwaltung

Anforderung

Eine Maschine besteht aus 100 Förderbändern. Jedes Förderband wird von einer MpAxisBasic Komponente kontrolliert. Die Synchronisation zwischen den Transportbändern wird über das Anwenderprogram der Maschine durchgeführt.

Zusätzlich zu den Standard Alarmen von MpAxisBasic soll jedes Förderband 2 weitere Alarme haben:

- "Motortemperatur über Warngrenze"
- "Förderband nicht synchronisiert"

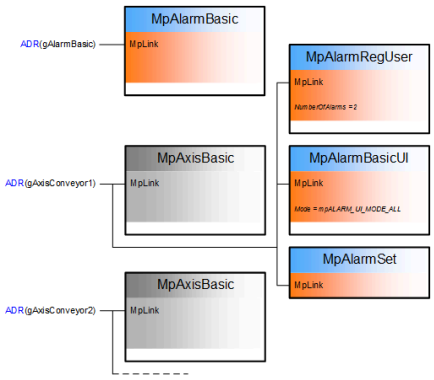
Die Alarme aller Förderbänder sollen gesammelt und auf der Visualisierung mittels der VC4 Alarmverwaltung für aktuelle und vergangene Alarme dargestellt werden. Dabei sollen die Alarme für die einzelnen Förderbänder jeweils in einer eigenen VC Alarmgruppe dargestellt sein. Durch diese deznetrale Aufteilung der Alarme für jedes Förderband soll eine übersichtliche Aufteilung der Alarme erreicht werden.

Lösung

Komponentenliste

- ***MpAxisBasic*** (Eigener MpLink): Kontrolle der Förderbänder
- ***MpAlarmBasic*** (Eignener MpLink): Zentrale Alarmverwaltung
- ***MpAlarmBasicUI*** (MpLink von MpAxisBasic): Verbindung der Alarme dezentral zu Visual Components mittels mehrerer Alarmabbilder. Für die "dezentrale" Alarmverwaltung verbunden mit dem MpLink von MpAxisBasic, damit hat jede mapp Komponente ihre eigene Alarmgruppe in VC4
- ***MpAlarmSet*** (MpLink von MpAxisBasic): Setzen eines Anwenderalarms
- ***MpAlarmRegUser*** (MpLink von MpAxisBasic): Registrierung von zusätzlichen Alarmen für eine mapp Komponente

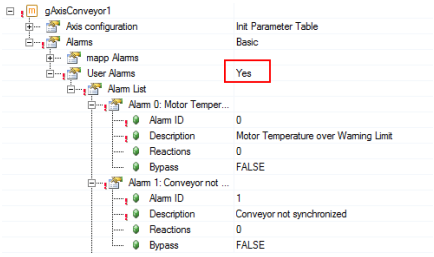
Verbindungsschema



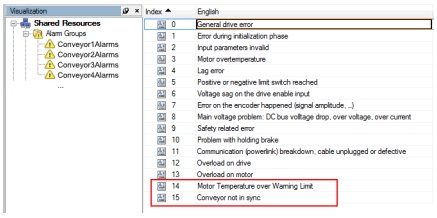
Verwendung der mapp Komponenten

Jede Instanz von **MpAlarmBasicUI** wird über den MpLink an die jeweilige **MpAxisBasic** Komponente angeschlossen. Im Modus mpALARM_UI_MODE_ALL sind dann mapp Alarme und Anwender Alarme im gleichen Alarmabbild enthalten. Dabei werden Anwender Alarme immer an die Standard Alarme angehängt. Standardmäßig hat jede mapp Komponente genug Platz für die eigenen Alarme. Um Anwenderalarme anzuhängen, müssen diese über **MpAlarmRegUser** registriert werden. Das bedeutet die Alarme können bereits in der Konfiguration konfiguriert werden, aber es müssen genug Plätze für die neuen Alarme registiert werden über MpAlarmRegUser.

In der Konfiguration von MpAxisBasic können die Eigenschaften der zusätzlichen Anwender Alarme angegeben werden:



In Visual Components muss eine Alarmgruppe mit den MpAxisBasic Standard Alarmen und den zusätzlichen Anwender Alarmen pro Förderband angelegt werden:



Die Größe der Alarmgruppe ergibt sich aus der Anzahl an Alarmen von der mapp Komponente, mit der MpAlarmBasicUI verbunden ist. In diesem Fall:

ImageSize = MpAxisBasic Alarme + Benutzeralarme

17 Alarme = 15 Alarme + 2 Alarme

Die Alarmgruppe müsste also 17 Elemente besitzen, genau wie die Größe von Alarm- und Quittierabbild 17 sein muss. Bei der dezentralen Anwendung von MpAlarm werden nur die Alarme der mapp Komponente, mit der MpAlarmBasicUI verbunden ist, beachtet.

9.3.4 Dezentrale Alarmverwaltung von 2 Achsen

Anforderung

Eine einfache synchrone Schneideanlage wird mit 2 MpAxisBasic Komponenten realisiert. Eine MpAxisBasic für das Förderband und eine zweite für die synchronisierte Bewegung der Schneideinrichtung.

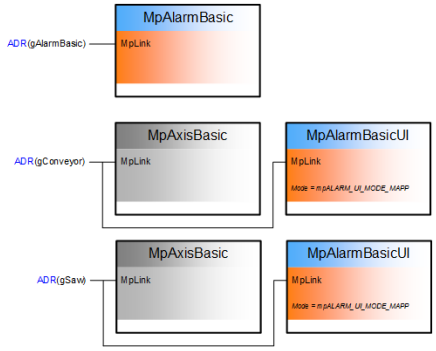
Die Alarme beider Achsen sollen gesammelt und auf der Visualisierung mittels der VC4 Alarmverwaltung für aktuelle und vergangene Alarme dargestellt werden.

Lösung

Komponentenliste

- ***MpAxisBasic*** (Eigener MpLink): Kontrolle des Förderbandes und der Schneidachse
- ***MpAlarmBasic*** (Eigener MpLink): Alarmverwaltung
- ***MpAlarmBasicUI*** (MpLink von MpAxisBasic): Verbindung der Alarme dezentral zu Visual Components mittels mehrerer Alarmabbilder. Für die "dezentrale" Alarmverwaltung verbunden mit dem MpLink von MpAxisBasic, damit hat jede mapp Komponente ihre eigene Alarmgruppe in VC4

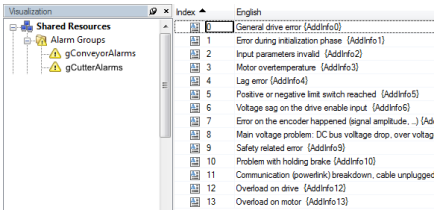
Verbindungsschema



Verwendung der mapp Komponenten

Jede Instanz von **MpAlarmBasicUI** wird über den MpLink an die jeweilige **MpAxisBasic** Komponente angeschlossen. Im Modus mpALARM_UI_MODE_MAPP sind dann nur die mapp Alarme im Alarmabbild enthalten.

Das bedeutet, dass in Visual Components eine Alarmgruppe mit den MpAxisBasic Standard Alarmen pro Achse angelegt werden muss:



Die Größe der Alarmgruppe ergibt sich aus der Anzahl an Alarmen von der mapp Komponente, mit der MpAlarmBasicUI verbunden ist. In diesem Fall:

ImageSize = MpAxisBasic Alarme
Alarmgruppe Förderband: 15 Alarme = 15 Alarme
Alarmgruppe Schneideachse: 15 Alarme = 15 Alarme

Die Alarmgruppen (2, eine pro Achse) müssten also jeweils 15 Elemente besitzen, genau wie die Größe der beiden Alarm- und Quittierabbilder 15 sein muss. Bei der dezentralen Anwendung von MpAlarm werden nur die Alarme der mapp Komponente, mit der MpAlarmBasicUI verbunden ist, beachtet.

Im Vergleich zu einem zentralen Aufbau, das heißt 1 MpAlarmBasicUI direkt verbunden mit MpAlarmBasic, hat diese Variante folgende Vor- und Nachteile:

Vorteil:

- Keine Verschiebungen im Alarmindex des Alarmabbilds, wenn später zusätzliche mapp Komponenten hinzugefügt werden

Nachteil:

- Mehr MpAlarmBasicUI Instanzen und UIConnect Strukturen erforderlich

9.3.5 Anwender Alarme

Anforderung

Die Temperatur eines Heizofens wird von der Applikation aus überwacht. Es soll ein Alarm ausgelöst werden sobald die Temperatur einen bestimmten Wert überschreitet.

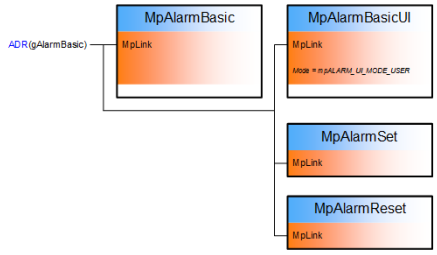
Dieser Alarm soll auf der Visualisierung mittels der VC4 Alarmverwaltung für aktuelle und vergangene Alarme dargestellt werden.

Lösung

Komponentenliste

- ***MpAlarmBasic*** (Eigener MpLink): Alarmverwaltung von mapp und Anwenderalarmen
- ***MpAlarmBasicUI*** (MpLink von MpAlarmBasic): Verbindung zu Visual Components mittels eines Alarmabbildes. Als MpLink wird jener von MpAlarmBasic verwendet, somit haben die Anwender Alarme ihre eigene Alarmgruppe in VC4
- ***MpAlarmSet*** (MpLink von MpAlarmBasic): Setzen eines Anwenderalarms
- ***MpAlarmReset*** (MpLink von MpAlarmBasic): Rücksetzen eines Anwenderalarms

Verbindungsschema



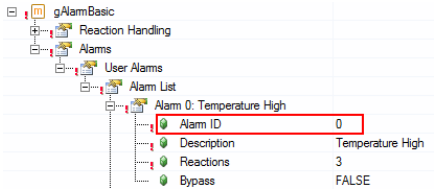
Verwendung der mapp Komponenten

Anwender Alarme können über die Funktionen **MpAlarmSet** und **MpAlarmReset** gesetzt und zurückgesetzt werden.

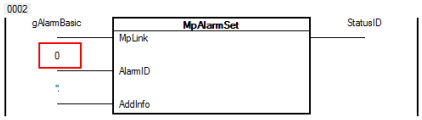
Die **MpAlarmBasic** Komponente sammelt alle Anwenderalarme und stellt sie mit Hilfe der Schnittstelle von **MpAlarmBasicUI** als Alarmabbild an Visual Components zur Verfügung. Im Modus mpALARM_UI_MODE_USER werden nur Anwender Alarme in das Alarmabbild übergeben.

In Visual Components wird eine Alarmgruppe für alle Anwender Alarme erstellt und mit der UIConnect Struktur von **MpAlarmBasicUI** verbunden:

Die ID eines Anwender Alarms kann im Automation Studio konfiguriert werden und muss dem Index in der VC Alarmgruppe entsprechen.



Der Alarm-ID wird verwendet, um den Alarm beim Aufruf der Funktion **MpAlarmSet** zu identifizieren:



9.4 Bibliotheken

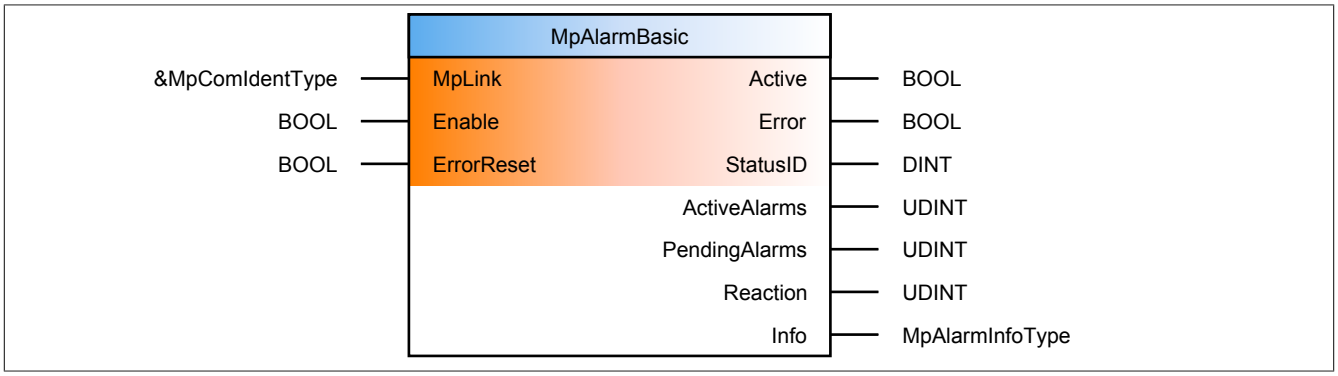
9.4.1 Funktionsbausteine

Unterstützt ab	Name	Beschreibung
5.0.0	MpAlarmBasic	Dieser Funktionsblock sammelt und verwaltet Alarme.
5.0.0	MpAlarmSet	Diese Funktion aktiviert benutzerdefinierte Alarme.
5.0.0	MpAlarmReset	Diese Funktion setzt benutzerdefinierte Alarme wieder zurück.
5.0.0	MpAlarmBasicUI	Dieser Funktionsblock wird benützt, um eine Verbindung zwischen dem mapp Alarmsystem und der VC4 Visualisierung zu schaffen.
5.0.0	MpAlarmReaction	Mit dieser Funktion wird anhand eines Filters festgestellt, ob eine bestimmte Reaktion aktiv ist.
5.0.0	MpAlarmRegUser	Mit diesem Funktionsblock wird Platz für benutzerdefinierte Alarme erzeugt, abhängig vom MpLink.

9.4.1.1 MpAlarmBasic

Dieser Funktionsbaustein sammelt und verwaltet Alarme.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmBasic Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.

I/O	Parameter	Datentyp	Beschreibung
OUT	ActiveAlarms	`UDINT`	Anzahl der momentan aktiven Alarme.
OUT	PendingAlarms	`UDINT`	Zeigt die Anzahl jener Alarme an, die entweder aktiv sind, oder noch nicht quittiert wurden. Spiegelt also genau jene Alarme wieder, die noch auf der Visualisierung zu sehen sind.
OUT	Reaction	`UDINT`	Bit kodierte Reaktion auf die aktuell aktiven Alarme.
OUT	Info	`MpAlarmInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

9.4.1.1.1 Beschreibung

Für diesen Funktionsbaustein wird der MpLink einer MpAlarmBasic Konfiguration verwendet. Die Hauptfunktionen dieses Funktionsbausteins sind:

Alarme zählen

Sobald der Funktionsbaustein aktiv ist, wird am Ausgang "ActiveAlarms" die Anzahl der aktuell aktiven Alarme angezeigt. Angezeigt werden alle mapp Komponenten Alarme und alle benutzerdefinierten Alarme. Inaktive, nicht quittierte Alarme werden nicht miteinberechnet.

Am Ausgang "PendingAlarms" wird die Anzahl aller aktiven und inaktiven, aber nicht quittierten Alarme angezeigt. Somit genau die Alarme, die auch auf der Visualisierung angezeigt werden, falls eine Quittierung erforderlich ist.

Maschinen Reaktion anzeigen

Pro Alarm kann eine konfigurierbare Reaktion ausgelöst werden. Diese Reaktion wird am Ausgang "Reaction" ausgegeben. Der Ausgang agiert bitkodiert, das bedeutet, für jeden Alarm wird ein einzelnes Bit des Ausgangs "Reaction" gesetzt. Das Ergebnis ist ein bitweise ODER verbundener Wert, der von MpAlarmBasic bereitgestellt wird.

Mögliche benutzerdefinierte Reaktionen könnten zum Beispiel ein sofortiger Maschinenstopp, pausieren der Maschine nach dem aktuellen Zyklus, Heizung ausschalten oder ein Wechsel in den Wartungszustand sein.

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	1	0	0		0	1
Stop immediate	Stop after this cycle	Switch to maintenance	Decrease machine speed		Switch off heating	No parameter change allowed

Machine reaction = 80000002 (Hex, Bit2 and Bit 32 are set)

In diesem Beispiel wird gezeigt, wie eine mögliche Maschinenreaktion aussehen könnte. In diesem Fall sind zwei Reaktionen aktiv (Bit 2 und Bit 32).

Die Konfiguration der Reaktionen erfolgt über Automation Studio. Für jede Komponente könne die mapp Alarme und die Benutzeralarme konfiguriert werden (siehe Abbschnitt MpAlarmBasic Konfiguration). Eine weitere Möglichkeit is die Konfiguration der Alarme über den WebXs (siehe Abschnitt Alarm WebXs). In beiden Fällen müssen die Benutzeralarme auch über MpAlarmRegUser hinzugefügt werden.

Alarme zusammenfassen

MpAlarmBasic fasst alle Alarme in einer gemeinsamen Liste zusammen. Wird MpAlarmBasic innerhalb einer mechatronischen Einheit verwendet, werden ausschließlich die Alarme dieser Einheit angezeigt. Besitzt diese Einheit auch hierarchisch untergeordnete Einheiten, werden auch deren Alarme miteinbezogen.

> **Alarme von mapp Komponenten, die vor MpAlarmBasic aktiviert werden ("Active" = `TRUE`), gehen verloren. Alarme welche nach der Aktivierung von MpAlarmBasic auftreten werden angezeigt.**

Benutzerdefinierte Alarme

Mithilfe der Funktion MpAlarmSet kann der Benutzer bis zu 100 eigene Alarme in die zentrale Alarmverwaltung einbinden. Diese Alarme können über den WebXs der MpAlarmBasic Komponente konfiguriert werden.

> **Es ist nicht notwendig die 100 Benutzeralarme von MpAlarmBasic über MpAlarmRegUser zu erstellen. Die Anzahl an möglichen Benutzeralarmen kann nicht weiter über MpAlarmRegUser erhöht werden.**

9.4.1.1.2 Zeitdiagramme

Hier werden die unterschiedlichen Reaktionen gezeigt bei unstetigen und anhaltenden Alarmen, sowie das Verhalten der MpAlarm.

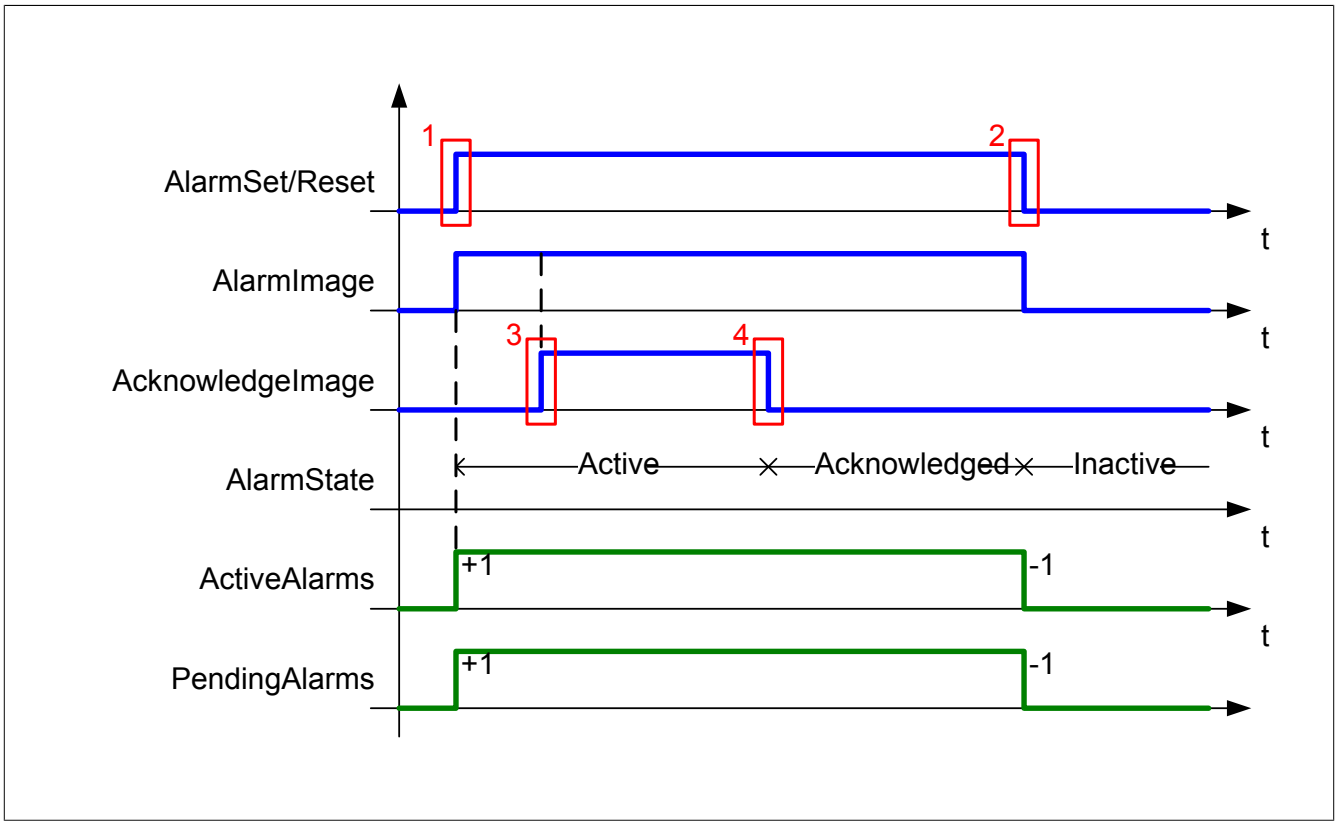
Anhaltender Alarm, Quittierung vor Zurücksetzen

Hier wird die Zusammenarbeit der verschiedenen Funktionen der MpAlarm gezeigt mithilfe eines anhaltenden Alarms. Der Alarm wird gesetzt durch das Benützen der MpAlarmSet Funktion[1]. Die Visualisierung setzt das "AcknowledgeImage"[3]. Dann wird vom Maschinenbediener der Alarm quittiert, aber der Alarm ist noch immer aktiv (anhaltender Alarm)[4]. Mit der Funktion MpAlarmReset wird der Alarm deaktiviert[2]. Am Ausgang des MpAlarmBasic Funktionsbaustein kann man mithilfe von "ActiveAlarms" beobachten wie lange der Alarm aktiv ist, bzw. wieviele Alarme aktiv sind. Der Alarm wird von "ActiveAlarms" entfernt, wenn er quittiert wurde oder wenn der Alarm zurückgesetzt wurde. Für "PendingAlarms" gilt: Erst wenn der Alarm inaktiv ist und quittiert wurde, wird der Alarm von "PendingAlarms" weggerechnet.

Visual Components Einstellungen:

- Acknowledge = Required

- WhileActive = Enabled



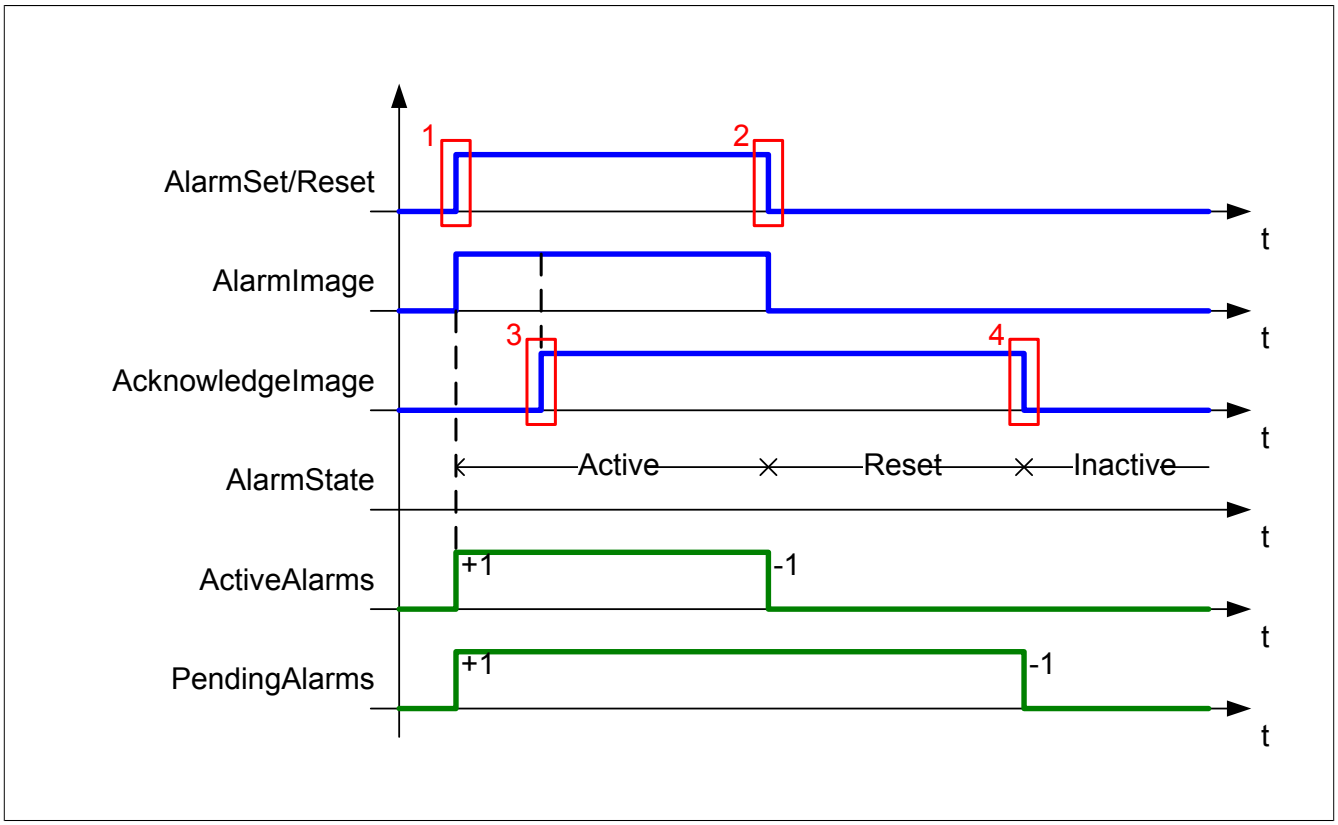
Anhaltender Alarm, Quittierung nach Zurücksetzen

Hier wird die Zusammenarbeit der verschiedenen Funktionen der MpAlarm gezeigt mithilfe eines anhaltenden Alarms. Der Alarm wird gesetzt durch das Benützen der MpAlarmSet Funktion[1]. Ab diesem Zeitpunkt gilt der Alarm als aktiv. Die Visualisierung setzt das "AcknowledgeImage"[3]. Dann wird vom Maschinenbediener der Alarm quittiert, aber er ist noch immer aktiv (anhaltender Alarm)[4]. Mit der Funktion MpAlarmReset wird der Alarm deaktiviert[2]. Am Ausgang des MpAlarmBasic Funktionsbaustein kann man mithilfe von "ActiveAlarms" beobachten wie lange der Alarm aktiv ist, bzw. wieviele Alarme aktiv sind. Der Alarm wird von "ActiveAlarms" entfernt, wenn er quittiert wurde oder wenn der Alarm zurückgesetzt wurde. Für "PendingAlarms" gilt: Erst wenn der Alarm inaktiv ist und quittiert wurde, wird der Alarm von "PendingAlarms" weggerechnet.

Visual Components Einstellungen:

- Acknowledge = Required

- WhileActive = Disabled



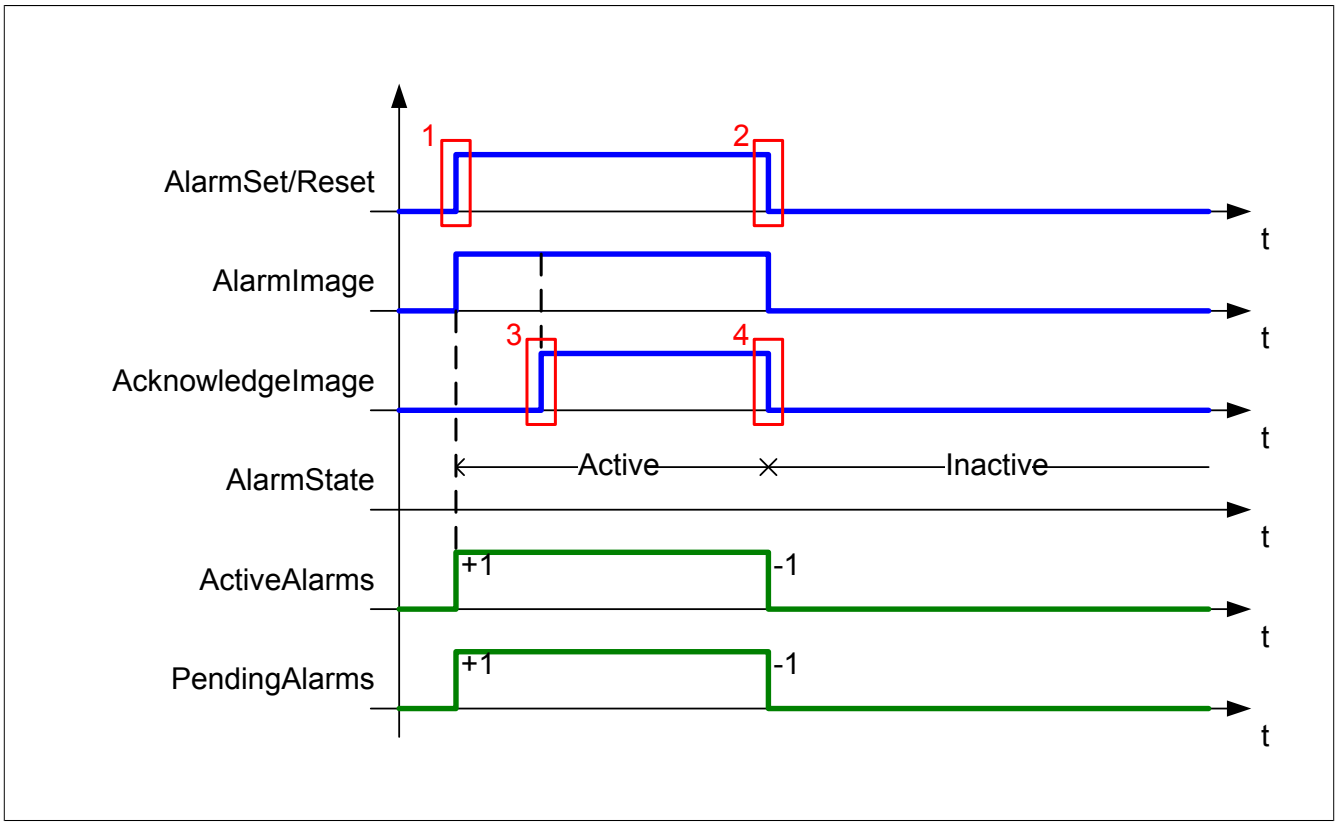
Anhaltender Alarm, Quittierung nicht erforderlich

Hier wird die Zusammenarbeit der verschiedenen Funktionen der MpAlarm gezeigt mithilfe eines anhaltenden Alarms. Der Alarm wird gesetzt durch das Benützen der MpAlarmSet Funktion[1]. Ab diesem Zeitpunkt gilt der Alarm als aktiv. Die Visualisierung setzt das "AcknowledgeImage"[3]. Quittierung ist nicht notwendig, aber möglich. Nach einiger Zeit wird mithilfe der Funktion MpAlarmReset der Alarm deaktiviert[2](auch das "AcknowledgeImage" wird zurückgesetzt[4]). Am Ausgang des MpAlarmBasic Funktionsbaustein kann man mithilfe von "ActiveAlarms" beobachten wie lange der Alarm aktiv ist, bzw. wieviele Alarme aktiv sind. Der Alarm wird von "ActiveAlarms" entfernt, wenn er quittiert wurde oder wenn der Alarm zurückgesetzt wurde. Für "PendingAlarms" gilt: Erst wenn der Alarm inaktiv ist und quittiert wurde, wird der Alarm von "PendingAlarms" weggerechnet.

Visual Components Einstellungen:

- Acknowledge = Possible

- WhileActive = Enabled



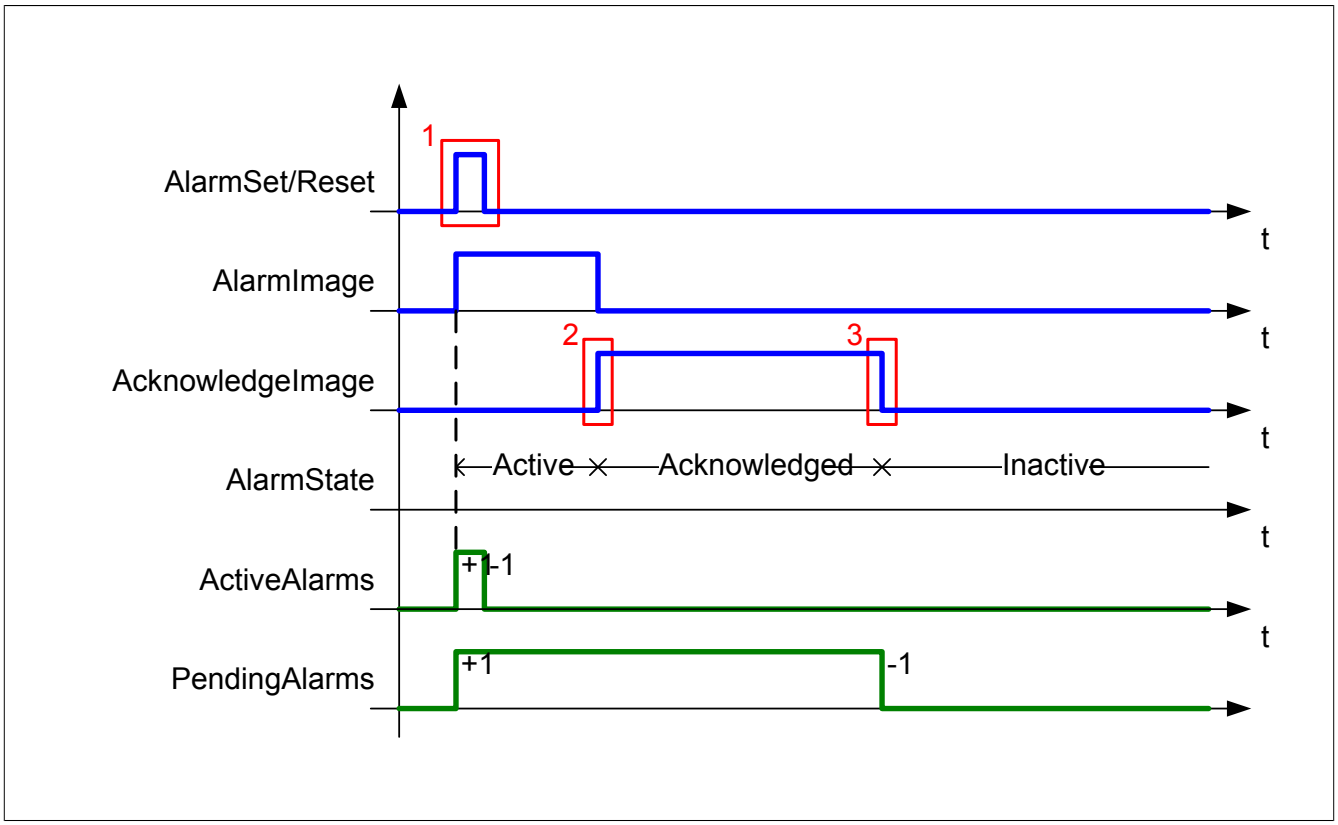
Unstetiger Alarm

In diesem Fall wird mit einem unstetigen Alarm gearbeitet (z.B. kein Zugriff, zu wenig Speicher, ...). MpAlarmSet und MpAlarmReset können dafür in dem selben Zyklus aufgerufen werden[1]. Die Visualisierung setzt das "AcknowledgeImage"[2]. Erst wenn der Maschinenbediener quittiert, wird auch im "AcknowledgeImage" das Signal zurückgesetzt[3]. Der Alarm wird von "ActiveAlarms" weggerechnet, wenn er quittiert wurde oder wenn der Alarm zurückgesetzt wurde. Für "PendingAlarms" gilt: Erst wenn der Alarm inaktiv ist und quittiert wurde, wird der Alarm von "PendingAlarms" weggerechnet.

Visual Components Einstellungen:

- Acknowledge = Required

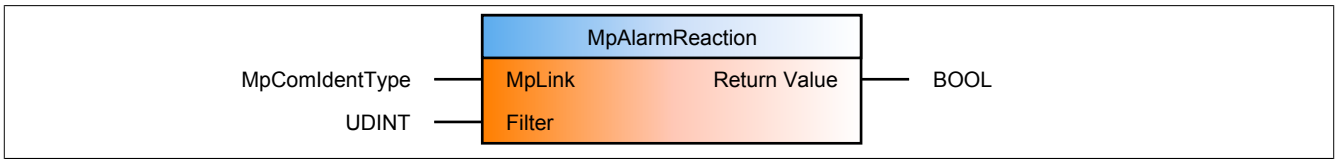
- WhileActive = Enabled



9.4.1.2 MpAlarmReaction

Mit dieser Funktion wird anhand eines Filters festgestellt, ob eine bestimmte Reaktion aktiv ist.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmBasic Konfiguration).
IN	Filter	`UDINT`	Filter (bitkodiert), mit dem spezielle Reaktionen überprüft werden können.
OUT	ReturnValue	`BOOL`	Zeigt an, ob eine der gefilterten Reaktionen aktiv ist.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

9.4.1.2.1 Beschreibung

MpAlarmReaction ist eine Funktion mit der bestimmte aktive Alarm Reaktionen gefiltert werden können. Für diese Funktion wird der MpLink einer MpAlarmBasic Konfiguration verwendet. Es kann nach allen Reaktionen die in der Konfiguration vorhanden sind gefiltert werden.

Dieser Filter wird definiert über den „Filter“ Parameter Eingang. Der Filter wird als UDINT übergeben (32 Reaktionen), gleich wie bei der Maschinenreaktion. Mit jedem im Filter gesetzten Bit wird genau dieses der 32 möglichen Reaktionen (Bits) überprüft. Es können somit eine oder mehrere Reaktionen gleichzeitig gefiltert werden.

Der Rückgabewert dieser Funktion ist boolesch. Ist eine der gefilterten Reaktionen aktiv/inaktiv, so wird der Rückgabewert `TRUE/FALSE`.

Beispiele:

Die gefilterte Reaktion ist aktuell nicht aktiv, daher wird `FALSE` zurückgegeben.

Machine reaction

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	1	0	0		0	1
Stop immediate	Stop after this cycle	Switch to maintenance	Decrease machine speed		Switch off heating	No parameter change allowed

(Bit 5 to Bit 30 are 0)

Filter

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	0	1	0		0	0

(Bit 5 to Bit 30 are 0)

Result: **FALSE**

Die gefilterte Reaktion ist aktuell aktiv, daher wird `TRUE` zurückgegeben. In diesem Fall ist auch genau bekannt, welche Reaktion aktiv ist, da nur eine Reaktion gefiltert wurde.

Machine reaction

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	1	0	0		0	1
Stop immediate	Stop after this cycle	Switch to maintenance	Decrease machine speed		Switch off heating	No parameter change allowed

(Bit 5 to Bit 30 are 0)

Filter

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	1	0	0		0	0

(Bit 5 to Bit 30 are 0)

Result: **TRUE**

Mindestens eine der hinterfragten Reaktionen ist aktuell aktiv, daher wird `TRUE` zurückgegeben. Da aber mehrere Reaktionen gefiltert werden, kann man nicht feststellen, welche aktiv ist.

Machine reaction

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	1	0	0		0	1
Stop immediate	Stop after this cycle	Switch to maintenance	Decrease machine speed		Switch off heating	No parameter change allowed

(Bit 5 to Bit 30 are 0)

Filter

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	1	0	0		1	1

(Bit 5 to Bit 30 are 0)

Result: **TRUE**

Mindestens eine der hinterfragten Reaktionen ist aktuell aktiv, daher wird `TRUE` zurückgegeben. In diesem Beispiel werden alle Reaktionen gefiltert(siehe Bild).

Machine reaction

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
0	1	0	0		0	1
Stop immediate	Stop after this cycle	Switch to maintenance	Decrease machine speed		Switch off heating	No parameter change allowed

(Bit 5 to Bit 30 are 0)

Filter

Bit 1	Bit 2	Bit 3	Bit 4		Bit 31	Bit 32
1	1	1	1		1	1

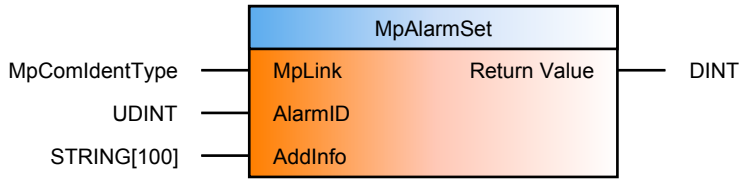
(Bit 5 to Bit 30 are 0)

Result: **TRUE**

9.4.1.3 MpAlarmSet

Diese Funktion setzt benutzerdefinierte Alarme.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmBasic Konfiguration)
IN	AlarmID	`UDINT`	Benutzeralarm (startet mit 0)
IN	AddInfo	`STRING[100]`	Zusätzliche Information in Textform (z.B. interne Fehlernummer, Alarmtext, Temperatur, ...)
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

9.4.1.3.1 Beschreibung

Beim Ausführen dieser Funktion wird ein benutzerdefinierter Alarm aktiviert. Bevor ein benutzerdefinierter Alarm gesetzt werden kann, muss er hinzugefügt werden. Benutzerdefinierte Alarme können auf folgende Weise hinzugefügt werden:

- **MpAlarmRegUser:** Nur über diesen Funktionsbaustein können zusätzlich zu den mapp Alarmen benutzerdefinierte Alarme erzeugt und einer beliebigen mapp Komponente zugeordnet werden

Somit ist MpAlarmRegUser die einzige Möglichkeit benutzerdefinierte Alarme zu erzeugen. Nur wenn die Alarme über MpAlarmRegUser hinzugefügt wurden können sie konfiguriert und verwendet werden. Zum Konfigurieren sind folgende Optionen verfügbar:

- **WebXs:** Über den WebXs können benutzerdefinerte Alarme konfiguriert werden (siehe Abschnitt Alarm Konfiguration über WebXs)
- **Automation Studio:** Bei jeder mapp Konfiguration in der Configuration View können zusätzlich benutzerdefinierte Alarme konfiguriert werden (siehe Abschnitt Alarm Konfiguration über Automation Studio)

Weiters besitzt MpAlarmBasic 100 Benutzeralarme, welche frei verwendet werden können.

Sobald die Alarme existieren (MpAlarmRegUser), können sie auch über MpAlarmSet aktiviert werden. Zum Zurücksetzen wird die Funktion MpAlarmReset verwendet.

Alarmzugehörigkeit:

Beim Setzen ist darauf zu achten, welche "AlarmID" verwendet wird. Abhängig vom MpLink, welcher verwendet wird, kann die "AlarmID" für denselben Benutzeralarm variieren:

- **MpLink von MpAlarmBasic:** Wird der MpLink von MpAlarmBasic angegeben, so findet man die korrekte "AlarmID" in der Gesamtübersicht am WebXs. Es wird ein MpAlarmBasicUI benötigt um die Gesamtliste der Benutzeralarme einzusehen
- **MpLink einer einzelnen Komponente:** Wird der MpLink von MpAlarmBasic angegeben, so findet man die korrekte "AlarmID" in der Einzelansicht dieser Komponente am WebXs (MpAlarmBasicUI notwendig) oder in der Automation Studio Konfiguration dieser Komponente

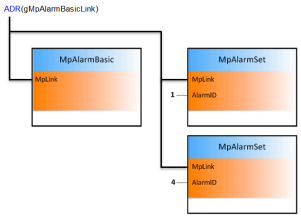
Beschreibung der Eingänge:

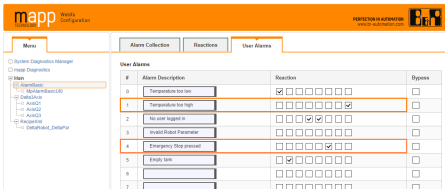
AlarmID: Gibt an, welcher Benutzeralarm aktiviert werden soll. Welcher Alarm welche ID besitzt, ist beim Erstellen im WebXs ersichtlich.

AddInfo (optional): Mit diesem optionalen Eingang kann für jeden Alarm eine zusätzlicher Info Text mitgeliefert werden. Dieser Eingang findet nur Verwendung in Verbindung mit einer MpAlarmBasicUI Komponente.

Beispiel:

In diesem Beispiel befindet sich eine MpAlarmBasic Komponente und 2 MpAlarmSet Funktionen, die alle mit einem globalen MpLink (gMpAlarmBasicLink) verbunden sind. Mit den beiden Funktionen werden die Benutzeralarme mit der ID 1 und 4, die in dem WebXs erstellt wurden, aktiviert.

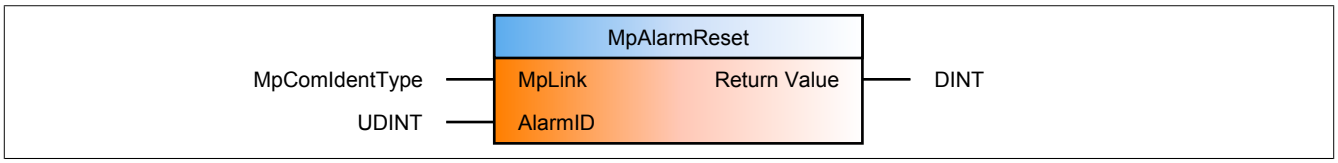




9.4.1.4 MpAlarmReset

Diese Funktion setzt benutzerdefinierte Alarme wieder zurück.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmBasic Konfiguration)
IN	AlarmID	`UDINT`	Benutzeralarm ID (startet mit 0)
OUT	Return Value	`DINT`	Status Information der Funktion

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

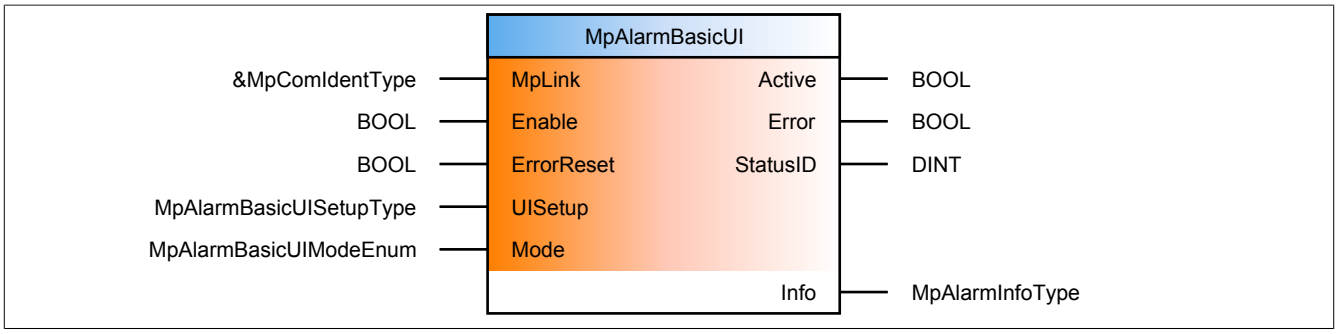
9.4.1.4.1 Beschreibung

Nachdem ein Alarm mithilfe von MpAlarmSet gesetzt wurde kann er mit dieser Funktion wieder zurückgesetzt werden. Für diese Komponente wird derselbe MpLink verwendet, der auch zum Setzen verwendet wurde.

9.4.1.5 MpAlarmBasicUI

Dieser Funktionsbaustein wird benützt, um eine Verbindung zwischen dem mapp Alarmsystem und der Visual Components 4 Visualisierung zu schaffen. MpAlarmUI erstellt die Alarmabbilder und Quittierabbilder für die Visualisierung.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmBasic Konfiguration oder einer anderer mapp Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Mode	`MpAlarmBasicUIModeEnum`	Modus: Zeige mapp Alarme / Benutzeralarme / alle Alarme - muss entschieden sein bevor der Funktionsbaustein aktiviert wird.
IN	UISetup	`MpAlarmBasicUISetupType`	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpAlarmInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

9.4.1.5.1 Beschreibung

Die Aufgabe dieser mapp Komponente ist es, eine Schnittstelle zwischen den Alarmen und der Visualisierung herzustellen. Es muss ein MpAlarmBasic aktiv sein. Der MpLink, der für MpAlarmBasicUI verwendet wird, entscheidet, woher die Informationen (Alarme) kommen. Diese mapp Komponente sollte erst aktiviert werden ("Ac-

tive" = `TRUE`), wenn die Komponente für welche die Alarme dargestellt werden sollen aktiv ist. Außerdem muss MpAlarmBasic aktiv sein um MpAlarmBasicUI nützen zu können. Als Beispiel: Die Alarme einer Achse sollten über MpAlarmBasicUI angezeigt werden. Die Reihenfolge, in welcher die verwendeten mapp Komponenten aktiviert werden sollten ist folgende:

1) MpAlarmBasic (eigener MpLink)
2) MpAxisBasic (eigener MpLink)
3) MpAlarmBasicUI (MpLink der Achse)

Sollen nur Benutzeralarme von MpAlarmBasic gesammelt und durch MpAlarmBasicUI in der Visualisierung angezeigt werden, wird MpAlarmBasicUI derselbe mapp Link angegeben wie an MpAlarmBasic. Wichtig hierbei ist, dass nur eine Instanz von MpAlarmBasicUI verwendet wird um Benutzeralarme anzeigen lassen zu können.

Modus

Am "Mode" Eingang kann man entscheiden, welche Alarme für die Visualisierung gefiltert und bereitgestellt werden.

- mpALARM_BASIC_UI_MODE_MAPP: Die Standard Alarme einer mapp Komponente
- mpALARM_BASIC_UI_MODE_USER: Nur benutzerdefinierte Alarme (Wird nur verwendet, wenn die MpAlarmBasicUI Komponente direkt mit einer MpAlarmBasic verbunden ist)
- mpALARM_BASIC_UI_MODE_ALL: Alle Alarme (Standard Komponenten und Benutzeralarme)

UISetup

In dieser Struktur werden die Größen für das Alarm und Quittier Array festgelegt. Die Größe der beiden Arrays muss in der Applikation bestimmt werden (abhängig von der Anzahl an Alarmen der Komponente, mit der MpAlarmBasicUI verbunden ist).

Ist diese mit MpAlarmBasic verbunden, so muss die Größe des Arrays in Abhängigkeit des Eingangs „Mode“ definiert werden.

- mpALARM_BASIC_UI_MODE_MAPP: Die Größe des Arrays ergibt sich aus der Summe der Standard Alarmen aller Komponenten, die aktiv sind (siehe "Complete Alarm List" im WebXs - ohne Benutzeralarme)
- mpALARM_BASIC_UI_MODE_USER: Die Größe des Arrays ergibt sich aus der maximalen Anzahl der Benutzeralarmen [0..99]
- mpALARM_BASIC_UI_MODE_ALL: Die Größe des Arrays ergibt sich aus der Summe aller Standard Komponentenalarmen und der Anzahl der Benutzeralarmen (siehe „Complete Alarm List“ im WebXs)

Ist die MpAlarmBasicUI mit einer anderen mapp Komponente verbunden, so ergibt sich die Größe des Arrays aus der Anzahl an Standard Alarmen dieser Komponente.

Weiters befinden sich in der UISetup Struktur die Parameter „AddInfo“ und „AddInfoStringSize“. Diese Parameter werden rein für die Aktivierung von Benutzeralarmen verwendet. Wird ein Benutzeralarm mit der Funktion MpAlarmSet aktiviert, so kann man diesem Alarm einen zusätzlichen Infotext anhängen.

- **AddInfo:** Wird als STRING Array übergeben und die Größe dieses Arrays ergibt sich aus der Anzahl der maximalen Benutzeralarmen [0..99]
- **AddInfoStringSize:** Länge des Strings

Nach Aktivierung eines Benutzeralarms wird die zusätzliche Information in dieses STRING Array eingetragen und kann zum Beispiel als Textschnipsel für die Visualisierung verwendet werden.

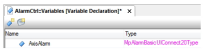
Vordefinierte UIConnect Strukturen

Zwei vordefinierte Strukturen sind über MpAlarm berfügbar welche bereits die Parameter "AlarmImage", "AcknowledgeImage" und "AddInfo" enthalten:

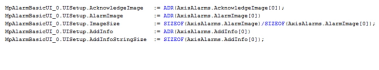
- MpAlarmBasicUIConnect20Type: Struktur für 20 Alarme
- MpAlarmBasicUIConnect100Type: Struktur für 100 Alarme (Passt genau für die Benutzeralarme von MpAlarmBasic)

Das bedeutet z.B. für eine MpAxisBasic kann man MpAlarmBasicUIConnect20Type verwenden. Man muss nur noch die Adressen der einzelnen Parameter ("AlarmImage", "AcknowledgeImage" und "AddInfo") and die UISetup Struktur von MpAlarmBasicUI übergeben.

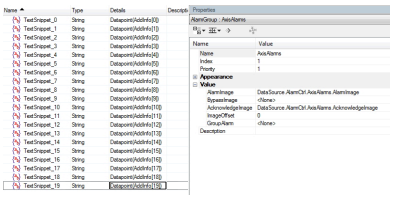
Es wird eine Variable erstellt für die Alarme einer Achse. Der vordefinierte Typ wird verwendet.



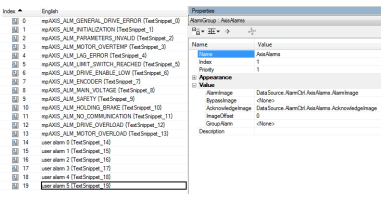
Die Variable wird mit der UISetup Struktur von MpAlarmBasicUI verbunden.



Es werden Textschnipsel für alle Alarme erstellt.



Alarmabbild, Quittierabbild und Textschnipsel werden mit der VC4 Alarmgruppe verbunden.



Zentral oder dezentrale Alarmverwaltung

Zentral

Alle Alarme werden von MpAlarmBasic gesammelt. Für MpAlarmBasicUI wird derselbe MpLink wie für MpAlarm verwendet. Damit wird eine Gesamtliste aller Alarme erstellt, welche dann über eine Alarmgruppe in VC4 abgebildet werden kann. Damit werden die Alarme zentral an *einer* Stellt verwaltet und über dargestellt. Die Anordnung der Alarme in dieser Gesamtliste wird im Abschnitt Alarmzuordnung genauer beschrieben.

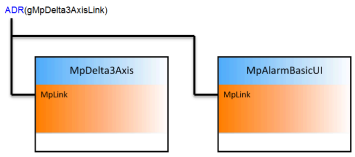
Dezentral

Eine weitere Möglichkeit ist die denzentrale Verwaltung der Alarme. Das bedeutet, dass die Alarme zwar zentral von MpAlarmBasic verwaltet werden, die Darstellung aber dezentral erfolgt. Für jede mapp Komponente, deren Alarme angezeigt werden sollen, wird eine Alarmgruppe in VC4 erstellt. Damit sind die Alarme der jeweiligen Komponente zugeordnet und es kann pro Komponente / pro Alarmgruppe eine unterschiedliche Priorität definiert werden.

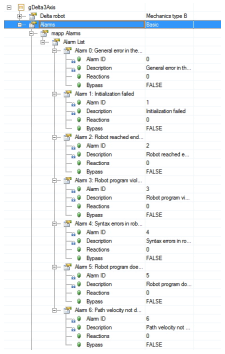
Es ist nicht möglich zentrale und dezentrale Alarmverwaltung zu mischen, in anderen Worten jeder Alarm darf nur in einem Alarmabbild / Quittierabbild vorkommen.

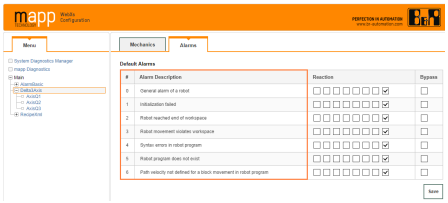
Beispiel (Alarm Handling für MpDelta3Axis):

Wenn man die Standard Alarme einer Komponente in das Alarmsystem der Visualisierung einbinden möchte, so muss dieser Komponente ein MpAlarmBasicUI angehängt werden (gleicher MpLink). Für jede Komponente die ins Alarmsystem eingebunden wird, muss eine Alarmgruppe mit den Standard Komponenten Alarmen in der Visualisierung erstellt werden.

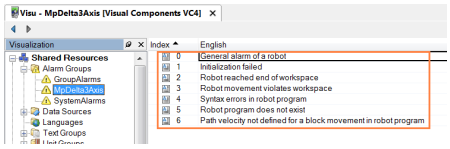


Standard Alarme der MpDelta3Axis Komponente (Ersichtlich in der Automation Studio Konfiguration des Delta Roboters oder im WebXs):





Alarmgruppe in der Visualisierung erstellen:



Parametrierung von MpAlarmBasicUI:

MpLink: MpLink der MpDelta3Axis Komponente

Mode: mpALARM_BASIC_UI_MODE_MAPP

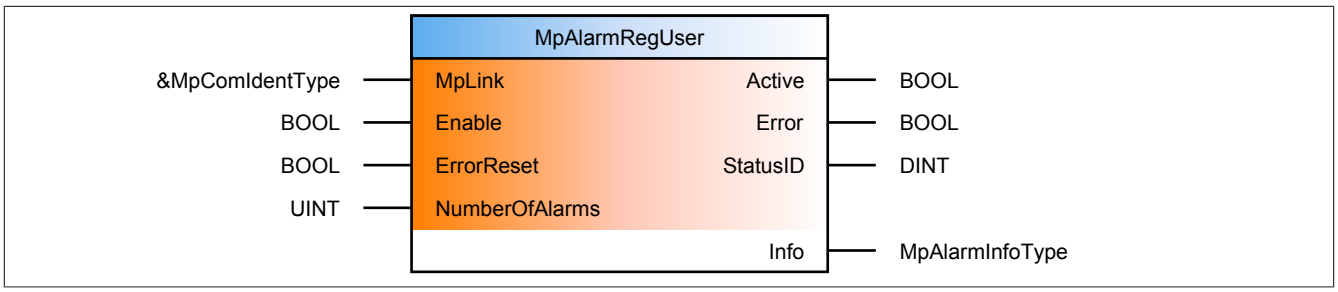
UISetup:

- AlarmImage[0..6]
- AcknowledgeImage[0..6]
- ImageSize = 7

9.4.1.6 MpAlarmRegUser

Mit diesem Funktionsbaustein wird Platz für benutzerdefinierte Alarme erzeugt, abhängig vom MpLink.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp Konfiguration (MpLink einer MpAlarmBasic Konfiguration).
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	NumberOfAlarms	`UINT`	Anzahl an Alarmen für welche Platz geschaffen wird.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpAlarmInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

9.4.1.6.1 Beschreibung

Falls einer beliebigen mapp Komponente benutzerdefinierte Alarme zugeordnet werden sollen, können diese mithilfe von MpAlarmRegUser erzeugt werden. Diese werden an die Konfiguration der mapp Komponente angehängt, von welcher der MpLink für MpAlarmRegUser kommt.

Man möchte beispielsweise einen Alarm der Achse zuweisen, an welcher er auftreten kann. Für diesen Alarm (z.B. "Achse kann an dieser Position keinen Referenziervorgang starten."), muss zuerst Platz geschaffen werden. Dafür wird MpAlarmRegUser mit dem MpLink der MpAxisBasic Konfiguration aufgerufen. Der Eingang "NumberOfAlarms" wird auf 1 gesetzt, da in diesem Beispiel nur Platz für einen zusätzlichen Benutzeralarm benötigt wird. Der Aufruf von MpAlarmRegUser erfolgt einmal im Init Programm. Sobald der Platz für den zusätzlichen Alarm reserviert wurde, kann MpAlarmRegUser deaktiviert werden. Sobald MpAlarmBasic eingeschaltet wird, darf MpAlarmRegUser nicht mehr aktiv sein.

Ab diesem Zeitpunkt kann mithilfe von MpAlarmSet der Benutzeralarm gesetzt werden. Die Alarmnummer beginnt dabei immer bei 0 für die Benutzeralarme.

Die mithilfe von MpAlarmRegUser erstellten Alarme, können über MpAlarmBasicUI auf zwei unterschiedliche Arten dargestellt werden:

- **Mode = mpALARM_BASIC_UI_MODE_USER:** Hierbei wird ein Array nur mit den Benutzeralarmen angezeigt, welche über MpAlarmRegUser erstellt wurden. Dabei ist das Element 0 des Arrays der erste benutzerdefinierte Alarm
- **Mode = mpALARM_BASIC_UI_MODE_ALL:** In diesem Fall wird ein Array mit allen Alarmen die von der Komponente sind, von welcher der MpLink verwendet wurde, erzeugt. Dabei beginnen die Benutzeralarme nach den mapp Alarmen. Für MpAxis bedeutet das z.B., dass der erste benutzerdefinierte Alarm das 13te Element in dem Array für das Alarmabbild verwendet, da MpAxis 13 mapp Alarme besitzt

> **Benutzerdefinierte Alarme können auch über die Konfiguration einer mapp Komponente konfiguriert werden. Trotzdem ist es notwendig über MpAlarmRegUser die benutzerdefinierten Alarme zur Konfiguration hinzufügen. (siehe Abschnitt MpAlarmBasic Konfiguration). Ausnahme sind die 100 Standard Benutzeralarme von MpAlarmBasic, dafür muss kein Platz über MpAlarmRegUser reserviert werden.**

9.4.2 Datentypen und Enumeratoren

9.4.2.1 Datentypen

9.4.2.1.1 MpAlarmBasicUIConnect20Type

Dieser Datentyp ist ein vorgefertigter UIConnect Datentyp für eine VC4 Visualisierung mit MpAlarmBasicUI. Über diesen Datentyp können bis zu 20 Alarme verwaltet werden. Alarmabbild und Quittierabbild aus dieser Struktur können direkt in der VC4 Visualisierung bhei der jeweiligen Alarmgruppe angeschlossen werden. Der "AddInfo" Parameter kann über Textschnipsel in die einzelnen Alarme eingebunden werden.

Parameter	Datentyp	Beschreibung
AlarmImage	`ARRAY[0..19] von BOOL`	Alarmabbild einer VC4 Alarmgruppe
AcknowledgeImage	`ARRAY[0...19] von BOOL`	Quittierabbild einer VC4 Alarmgruppe
AddInfo	`ARRAY[0..19] von STRING[100]`	Zusätzliche Informationen einer VC4 Alarmgruppe

9.4.2.1.2 MpAlarmBasicUIConnect100Type

Dieser Datentyp ist ein vorgefertigter UIConnect Datentyp für eine VC4 Visualisierung mit MpAlarmBasicUI. Über diesen Datentyp können bis zu 100 Alarme verwaltet werden. Alarmabbild und Quittierabbild aus dieser Struktur können direkt in der VC4 Visualisierung bei der jeweiligen Alarmgruppe angeschlossen werden. Der "AddInfo" Parameter kann über Textschnipsel in die einzelnen Alarme eingebunden werden.

Die Größe dieses Datentyps entspricht genau der Größe die notwendig ist, um die 100 Benutzeralarme von MpAlarmBasic zu verwalten.

Parameter	Datentyp	Beschreibung
AlarmImage	`ARRAY[0..99] von BOOL`	Alarmabbild einer VC4 Alarmgruppe
AcknowledgeImage	`ARRAY[0...99] von BOOL`	Quittierabbild einer VC4 Alarmgruppe
AddInfo	`ARRAY[0..99] von STRING[100]`	Zusätzliche Informationen einer VC4 Alarmgruppe

9.4.2.1.3 MpAlarmBasicUISetupType

Struktur, in der alle Variablen vereint sind die mit der Visual Components 4 Visualisierung verbunden sind, speziell geht es um die Abbilder für die Aktivierung und Rücksetzung von Alarmen, sowie zusätzliche Infos in Textform.

Parameter	Datentyp	Wert	Beschreibung
ImageSize	`UDINT`		Anzahl der Alarme - Alarm - / Quittierabbild muss von dieser Größe sein.
AlarmImage	`UDINT`		Zeiger auf Alarmabbild (BOOL Array welches auf **VC4** AlarmImage verbunden werden kann)
AcknowledgeImage	`UDINT`		Zeiger auf Quittierabbild (BOOL Array welches auf **VC4** Quittierabbild verbunden werden kann)
AddInfoStringSize	`UINT`		Größe eines Strings des "AddInfo" Eingangs
AddInfo	`UDINT`		Zeiger auf zusätzliche Alarm Informationen (Array von STRING[AddInfoStringSize-1]), die als Textschnipsel eingefügt werden können
ImageUpdateTime	`UDINT`	50	Aktualisierungsintervall des Alarmabbilds [ms]

In den "Gemeinsamen Ressourcen" im VC4 können Alarm Gruppen erstellt werden. Für jede Alarm Gruppe können verschiedene Abbilder gemappt werden. (z.B. AlarmImage & AcknowledgeImage)

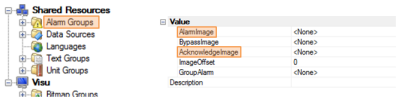
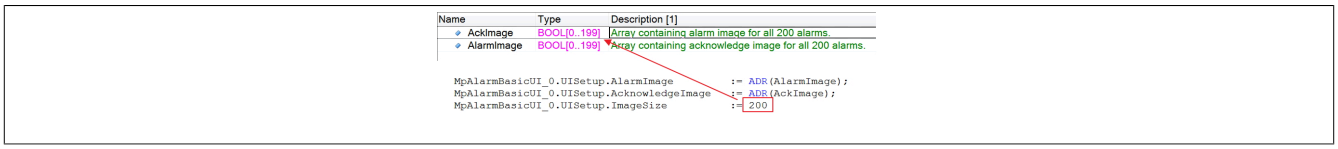


Image Größe

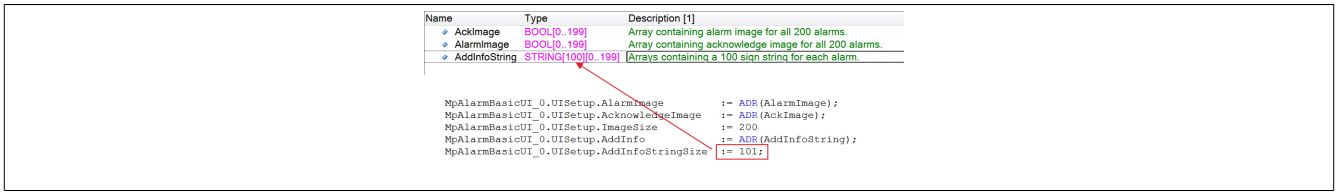
Der Parameter "ImageSize" wird entsprechend der Größe des Alarmabbilds / Quittierabbilds konfiguriert. Das bedeutet, wenn es 200 Alarme im System gibt, dann müssen Alarm- und Quittierabbild eine Größe von 200 Elementen besitzen [0..199]. Weiters muss die "ImageSize" auf 200 gesetzt werden.



AddInfo String Größe

Weiters können auch zusätzliche Daten für jeden Alarm weitergegeben werden. Das passiert über ein Stringarray. Jedes Element des Arrays ist ein String, welcher für einen bestimmten Alarm Zusatzdaten enhält. Der Index innerhalb des Arrays korrespondiert mit dem Index eines Alarms innerhalb des Alarm- / Quittierabbilds. Die Länge des Arrays muss deshalb dieselbe sein wie beim Alarm- bzw. Quittierabbild und wird somit über "ImageSize" angegeben. Was noch fehlt ist die Information wie lange ein einzelnes Element des Stringarrays, sprich ein String, ist. Das wird über den Parameter "AddInfoStringSize" angegeben. Dabei ist zu beachten, dass beim Erstellen eines Strings im Variableneditor die Nullterminierung nicht inkludiert ist, beim Erstellen eines Strings innerhalb eines Tasks jedoch schon. Ein konkreter Beispiel:

Für die obigen 200 Alarme sollen auch Zusatzdaten übergeben werden. Die Länge der Zusatzdaten pro Alarm sind 100 Zeichen. Erstelle ich dafür eine Variable im Variableneditor sieht das folgendermaßen aus:



Beim Erstellen der Variable im Editor wird die Nullterminierung außer Acht gelassen, da diese automatisch mit reserviert wird. Wichtig ist, dass in der UISetup Struktur eine Größe von 101 Zeichen pro String angegeben werden muss. Die Nullterminierung darf hier nicht vergessen werden!

Wird stattdessen die Variable nicht im Variableneditor, sondern im Task erstellt (wie es in ANSI-C möglich ist), dann sieht dasselbe Beispiel folgendermaßen aus:

```
_LOCAL BOOL AlarmImage[200];
_LOCAL BOOL AckImage[200];
_LOCAL STRING AddInfoString[200][101];

MpAlarmBasicUI_0.UISetup.AlarmImage              = ADR(AlarmImage);
MpAlarmBasicUI_0.UISetup.AcknowledgeImage        = ADR(AckImage);
MpAlarmBasicUI_0.UISetup.ImageSize               = 200
MpAlarmBasicUI_0.UISetup.AddInfo                 = ADR(AddInfoString);
MpAlarmBasicUI_0.UISetup.AddInfoStringSize       = 101;
```

Auch hier muss wieder auf die Nullterminierung geachtet werden, nur diesmal muss die Nullterminierung bereits beim Erstellen der Variable in Betracht gezogen werden.

9.4.2.1.4 MpAlarmDiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	`MpAlarmStatusIDType`	StatusID Diagnose Struktur

9.4.2.1.5 MpAlarmInfoType

Mit diesem Datentyp wird für die MpAlarm Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	`MpAlarmDiagType`	Eine Diagnosestruktur für die StatusID

9.4.2.1.6 MpAlarmStatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	`MpAlarmErrorEnum`	Fehlerkode der mapp Komponente
Severity	`MpComSeveritiesEnum`	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler)
Code	`UINT`	Kode der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden

9.4.2.2 Enumeratoren

9.4.2.2.1 MpAlarmBasicUIModeEnum

Dieser Aufzählungsdatentyp entscheidet beim Erstellen der Abbilder (Alarm und Quittier) ob alle Alarme der verbundenen Komponenten, nur mapp Alarme oder nur benutzerdefinierte Alarme angezeigt werden.

Aufzählung	Beschreibung
mpALARM_BASIC_UI_MODE_ALL	Alle Alarme der verbundenen Komponente anzeigen
mpALARM_BASIC_UI_MODE_MAPP	Nur mapp Alarme der verbundenen Komponente anzeigen
mpALARM_BASIC_UI_MODE_USER	Nur benutzerdefinierte Alarme der verbundenen Komponente anzeigen

9.4.3 Statusnummern

9.4.3.1 1083326469: MpAlarmBasic ist noch nicht aktiv

Beschreibung:

Diese Komponente kann erst aktiv werden, wenn MpAlarmBasic aktiv ist.

Reaktion:

Der Funktionsbaustein wird diese Information an dem Ausgang "StatusID" anzeigen.

Ursache / Abhilfe:

- MpAlarmBasic aktivieren

Konstante:

`mpALARM_INF_WAIT_BASIC_FB`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasicUI

9.4.3.2 1083182117: Konfiguration erfolgreich geladen

Beschreibung:

Konfiguration des Dienstes {2:AlarmService}: {3:NumberOfService} untergeordnete Dienste wurden erfolgreich geladen.

Zusätzliche Informationen

- {2:AlarmService}: Der Alarmservice, der die Information getriggert hat.
- {3:NumberOfServices}: Anzahl der geladenen untergeordneten Dienste.

9.4.3.3 1083182116: Kein Komponentendienst

Beschreibung:

In der gegebenen Registrierung wurde kein Komponentendienst gefunden.

9.4.3.4 -2138043362: Offset Überschneidung im Eintrag

Beschreibung:

Problem in Konfiguration des Dienstes {2:AlarmService}: Offset Überschneidung im Eintrag {3:EntryNumber}

Zusätzliche Informationen

- {2:AlarmService}: Der Alarmservice, der die Warnung ausgelöst hat.
- {3:EntryNumber}: Der Mapping-Eintrag in welchem eine Offset Überschneidung erkannt wurde

Ursache / Abhilfe:

- Wurde auf eine neuere mapp Version upgedated kann es zu diesem Fehler kommen, wenn in dieser mapp Version eine neue Komponente mit neuen Alarmen vorhanden ist
- Überprüfung der MpAlarmBasic Konfiguration. Im Abschnitt Mapping könnten sich Einträge überschneiden, deshalb sollte "NumberOfAlarms" und "Offset" geprüft werden

9.4.3.5 -2138043361: Anzahl der Alarme im Eintrag ist 0

Beschreibung:

Konfiguration des Dienstes {2:AlarmService}: Anzahl der Alarme ist 0 im Eintrag {3:EntryNumber}

Zusätzliche Informationen

- {2:AlarmService}: Der Alarmservice, der die Warnung ausgelöst hat.
- {3:EntryNumber}: Der Eintrag in der Alarm Mapping Konfiguration, in welchem der Fehler erkannt wurde

Ursache / Abhilfe:

- Überprüfen ob Parameter "NumberOfAlarms" 0 in einem Eintrag in der Mapping Konfiguration von MpAlarmBasic ist

9.4.3.6 -2138043360: Konfigurationsdaten des Dienstes konnten nicht geladen werden.

Beschreibung:

Die Konfigurationsdaten des Dienstes {2:AlarmService} konnten nicht geladen werden. Fehlerursache: {1:ErrorNumber}, Konfigurationsgruppe: {3:ConfigGroup}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund der Warnung.
- {2:AlarmService}: Der Alarmservice, der die Warnung ausgelöst hat.
- {3:ConfigGroup}: Die Konfigurationsgruppe, zu der der Alarmservice gehört.

Ursache / Abhilfe:

- Überprüfung der Konfiguration

9.4.3.7 -2138043359: Konfiguration mit Warnungen geladen

Beschreibung:

Die Konfiguration des Dienstes {2:AlarmService} wurde mit Warnungen geladen.

Zusätzliche Informationen

- {2:AlarmService}: Der Alarmservice, der die Warnung ausgelöst hat.

Ursache / Abhilfe:

- Siehe Logger für mehr Informationen

9.4.3.8 -2138043358: Konfiguration konnte nicht gespeichert werden

Beschreibung:

Die Konfigurationsdaten des Dienstes {2:AlarmService} konnten nicht gespeichert werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund der Warnung
- {2:AlarmService}: Der Alarmservice, der die Warnung ausgelöst hat

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

9.4.3.9 -2138043357: Anzahl der Alarme korrigiert

Beschreibung:

Die Mapping Konfiguration wurde geladen, aber die Anzahl der Alarme von einzelnen Komponenten wurde limitiert.

Ursache / Abhilfe:

- Überprüfung der Alarm Mapping Konfiguration von MpAlarmBasic. Dazu müssen die erweiterten Eigenschaften in der Konfiguration aktiviert werden

9.4.3.10 -2138043354: Kein Zugriff auf Konfigurationsdaten

Beschreibung:

Es konnte nicht auf die Konfigurationsdaten des Dienstes {2:AlarmService} zugegriffen werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund der Warnung.
- {2:AlarmService}: Der Alarmservice, der die Warnung ausgelöst hat.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

9.4.3.11 -2138043353: Alarmindex zu groß

Beschreibung:

Der Alarmindex {2:AlarmIndex} in den Konfigurationsdaten {3:ConfigEntry} ist zu groß.

Zusätzliche Informationen

- {2:AlarmIndex}: Der Index des Alarms.
- {3:ConfigEntry}: Der Konfigurationseintrag des Alarms.

Ursache / Abhilfe:

- Verkleinern Sie den Index des Alarms in der Konfiguration

9.4.3.12 -2138043352: Zuordnung konnte nicht geladen werden

Beschreibung:

Die Konfiguration des Dienstes {2:AlarmService} konnte nicht geladen werden. Fehlerursache: {1:ErrorNumber}.

Handelt es sich um den ersten Systemstart sollte ein Warmstart durchgeführt werden.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund der Warnung.
- {2:AlarmService}: Der Alarmservice, der die Warnung ausgelöst hat.

Ursache / Abhilfe:

- Erster Systemstart

9.4.3.13 -2138043351: Benutzer Alarmkonfiguration konnte nicht hinzugefügt werden

Beschreibung:

Die Benutzeralarmkonfiguration konnte nicht geladen werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

9.4.3.14 -2137899006: Abbild kleiner als maximale Alarmanzahl

Beschreibung:

Das erstellte Abbild (Alarm oder Quittier) ist kleiner als die maximale Anzahl an möglichen Alarmen.

Reaktion:

Der Funktionsbaustein wird am Ausgang "StatusID" anzeigen, dass eine Warnung aktiv ist.

Ursache / Abhilfe:

- Es kann zu Problemen kommen wenn sehr viele Alarme ins System aufgenommen werden, da die Abbilder zu klein sind

Konstante:

`mpALARM_WRN_IMAGE_SIZE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasicUI

9.4.3.15 -1064301526: Dienst konnte nicht registriert werden

Beschreibung:

Der Dienst {2:AlarmService} konnte nicht registriert werden.

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:AlarmService}: Der Alarmservice, der den Fehler ausgelöst hat.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

9.4.3.16 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasic
- MpAlarmBasicUI

Konstante:

`mpALARM_ERR_ACTIVATION`

9.4.3.17 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasic
- MpAlarmBasicUI
- MpAlarmSet
- MpAlarmReset
- MpAlarmReaction

Konstante:

`mpALARM_ERR_MPLINK_NULL`

9.4.3.18 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasic
- MpAlarmBasicUI
- MpAlarmSet
- MpAlarmReset
- MpAlarmReaction

Konstante:

`mpALARM_ERR_MPLINK_INVALID`

9.4.3.19 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasic
- MpAlarmBasicUI

Konstante:

`mpALARM_ERR_MPLINK_CHANGED`

9.4.3.20 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasic
- MpAlarmBasicUI
- MpAlarmSet
- MpAlarmReset
- MpAlarmReaction

Konstante:

`mpALARM_ERR_MPLINK_CORRUPT`

9.4.3.21 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasic

Konstante:

`mpALARM_ERR_MPLINK_IN_USE`

9.4.3.22 -1064239091: Ungültige Konfiguration

Beschreibung:

Konfiguration konnte während dem Erstellen der Komponente nicht gelesen werden. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Konfiguration ist beschädigt
- Die Funktion ist in der Konfiguration nicht aktiviert

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasic

Konstante:

`mpALARM_ERR_CONFIG_INVALID`

9.4.3.23 -1064157184: Alarmnummer zu hoch

Beschreibung:

Die übergebene Alarmnummer ist höher als die maximale Anzahl an Alarmen

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "AlarmID" Eingang überprüfen
- Reduzieren der möglichen Alarme

Konstante:

`mpALARM_ERR_ALARM_ID_OVER_MAX`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmSet
- MpAlarmReset

9.4.3.24 -1064157183: Abbild ist NULL

Beschreibung:

Entweder das Alarmabbild oder das Quittierabbild ist ein NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen das ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Angeschlossenes Alarmabbild überprüfen ("AlarmImage" in MpAlarmBasicUISetupType)
- Angeschlossenes Quittierabbild überprüfen ("AcknowledgeImage" in MpAlarmBasicUISetupType)

Konstante:

`mpALARM_ERR_IMAGE_NULL`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasicUI

9.4.3.25 -1064157181: Komponente hat keine Alarme

Beschreibung:

Die Komponente, mit der der Funktionsblock verbunden ist, hat keine eigenen Alarme.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Komponente, mit der der Funktionsblock verbunden ist, hat keine eigenen Alarme
- Überprüfen, ob der korrekte MpLink verwendet wurde
- Im eingestellten Modus (mpALARM_BASIC_UI_MODE_USER / mpALARM_BASIC_UI_MODE_MAPP) hat der Funktionsblock keine eigenen Alarme. Moduse anpassen

Konstante:

`mpALARM_ERR_NO_ALARMS_FOUND`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmBasicUI
- MpAlarmSet
- MpAlarmReset
- MpAlarmReaction

9.4.3.26 -1064157180: Ungültige Alarmanzahl

Beschreibung:

Es müssen mehr als null benutzerdefinierte Alarme erzeugt werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Der Parameter "NumberOfAlarms" muss größer als 0 sein

Konstante:

`mpALARM_ERR_NUMBER_OF_ALARMS`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpAlarmRegUser

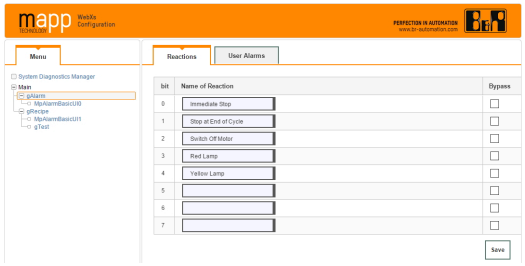
9.5 Alarm WebXs

Für MpAlarm gibt es einen speziellen Eintrag im WebXs. Damit das Alarmsystem im WebXs sichtbar und verwendbar wird, muss mindestens eine MpAlarmBasic Komponente auf der Steuerung aktiv sein. Diese zusätzliche Funktion im WebXs ermöglicht es, seine Alarme der Komponenten zu verwalten. Zum einen wird eine komplette Liste aller Alarme erstellt und dient als Übersicht. Weiters kann man seine eigenen Maschinenreaktionen erstellen und konfigurieren. Auch Benutzeralarme werden über den WebXs erstellt und konfiguriert (Maschinenreaktionen zuweisen).

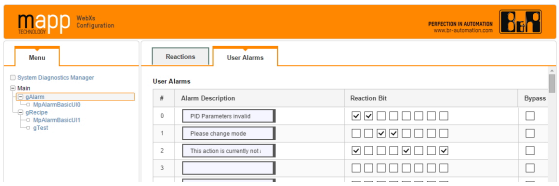
9.5.1 Konfiguration

Damit die Alarm Konfiguration auf dem WebXs verwendet werden kann, muss eine MpAlarmBasic Komponente im System aktiv sein.

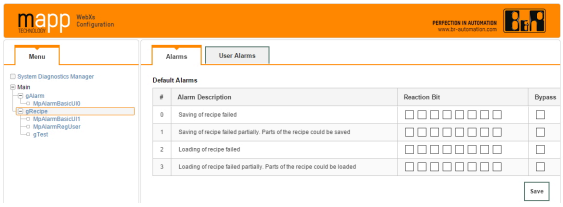
Wird auf der WebXs Konfigurationsseite (IP Adresse/mapp/config, z.B. 127.0.0.1/mapp/config) die MpAlarmBasic Komponente ausgewählt, so wird der Reiter *Reactions* angezeigt. Über diesen können verschiedene Maschinenreaktionen definiert werden. Mit der *Bypass* Check Box hat man die Option unerwünschte Reaktionen zu deaktivieren. Mithilfe der Schaltfläche *Save* kann die Konfiguration gespeichert werden.



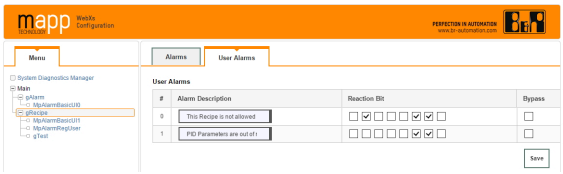
Über den Reiter *User Alarms* können benutzerdefinierte Alarme erstellt werden. Es muss eine Beschreibung angegeben werden. Für jeden Benutzeralarm kann eine Maschinenreaktion angegeben werden. Es können auch mehrere Reaktionen für einen Alarm ausgewählt werden. Die Reaktionen starten bitweise von rechts nach links (Rechts: Bit 0 / Links: Bit 7). Um genauere Informationen zu bekommen, welche Check Box für welche Reaktion steht, wird ein Tooltip angezeigt, wenn man mit der Maus über einer Checkbox fährt. Mit der *Bypass* Check Box hat man die Option, unerwünschte Alarme zu deaktivieren. Wird ein Haken in der Check Box zu einem Alarm gesetzt, so wird dieser Alarm und dessen Reaktion(en) in der Software ignoriert. Um die Konfiguration zu übernehmen, muss die Schaltfläche *Save* gedrückt werden. Das erfolgreiche Speichern wird mit einem Popup Fenster bestätigt. Weiters muss der Funktionsbaustein MpAlarmBasic neu aktiviert werden, damit die Änderungen übernommen werden.



Wird auf der WebXs Konfigurationsseite irgendeine andere Komponente ausgewählt, so werden deren Standard Alarme angezeigt, sofern sie welche besitzt. Der Text für Standard Alarme kann nicht verändert werden. Deren Reaktion kann jedoch konfiguriert werden. Auch die *Bypass* Option ist verfügbar.

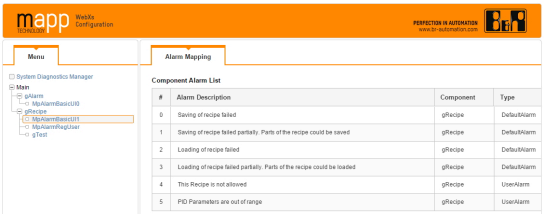


Über den Funktionsbaustein MpAlarmRegUser können für jede Komponente zusätzliche Benutzeralarme erstellt werden. Um diese zu konfigurieren, muss die der Reiter User Alarms der jeweiligen Komponente geöffnet werden. Dort kann Beschreibung, sowie Reaktion eingestellt werden.

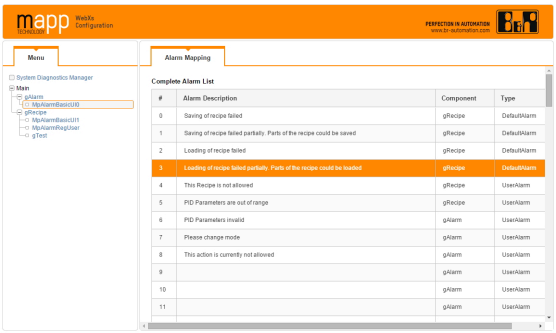


9.5.2 Übersicht

Möchte man die Alarme einer Komponente über eine VC4 Visualisierung anzeigen, so wird ein MpAlarmBasicUI Funktionsbaustein benötigt. Dieser muss denselben MpLink verwenden wie die Komponente, von welcher die Alarme angezeigt werden sollen. Ist MpAlarmBasicUI aktiv, so kann man eine Liste der Alarme auf der WebXs Konfigurationsseite der jeweiligen MpAlarmBasicUI finden. Dabei sind Standard Alarme und Benutzeralarme zusammengefasst. In der Spalte *Component* wird über den Namen des MpLinks angezeigt von welcher Komponente der Alarm stammt. Die Spalte *Type* zeigt, ob der jeweilige Alarm ein Standard Alarm, oder ein Benutzeralarm ist.



Wird ein MpAlarmBasicUI Funktionsbaustein an eine MpAlarmBasic Komponente angeschlossen (selber MpLink), so wird auf der WebXs Konfigurationsseite eine Liste mit den Alarmen aller Komponenten angezeigt. Eine Unterteilung kann über Applikationsmodulen erreicht werden. In der Spalte *Component* wird über den Namen des MpLinks angezeigt von welcher Komponente der Alarm stammt. Die Spalte *Type* zeigt, ob der jeweilige Alarm ein Standard Alarm, oder ein Benutzeralarm ist.



10 mapp Com: mapp Verwaltung

mapp Com strukturiert die Komponeten und verwaltet die Einheitenverwaltung.

10.1 Konzept

Allgemeines

- Als Servicetechniker möchte ich die automatisch über mapp verfügbare Online Diagnose und Konfiguration nützen
- Als Applikationist möchte ich mich nicht um die Querkommunikation zwischen mapp Komponenten kümmern müssen
- Als Applikationist möchte ich meine Software modular in Gruppen gliedern können
- Als Applikationist möchte ich schnell und einfach Einheiten für eine Variable definieren. Diese sollen zur Laufzeit, abhängig vom verwendeten Einheitensystem, automatisch angepasst werden
- Als Applikationist möchte ich meine File Devices einfach konfigurieren können, sowie mich ohne Aufwand mit USB Sticks verbinden können

Export-, Import-Funktion

- Als Maschinenbediener möchte ich die Konfiguration meiner aktuellen Maschine exportieren und bei einer anderen Maschine wieder importieren können
- Als Maschinenbediener möchte ich meine Daten auf ein beliebiges File Device (z.B. USB Stick, FTP Server, ...) exportieren können

Daten visuell darstellen

- Als Applikationist möchte ich alle Einträge die mapp im Logger hinterlässt auch über meine Visualisierung darstellen können

10.1.1 Einheitenverwaltung

Zu diesem Thema gibt es auch den Anwendungsfall Daten mit Einheit aufzeichnen.

Auf unserer B&R Tutorial Portal, wird im Tutorial Einheitenverwaltung in mapp erklärt wie die Einheitenverwaltung in mapp Komponenten funktioniert.

Das Einheitensystem im Alarmsystem funktioniert erst mit mapp Services Version 5.1.x

Standardmäßig besitzen Prozessvariablen keine Einheit. Um die Prozessdaten jedoch inklusive Einheit darstellen zu können, muss für jede sichtbare Prozessvariable die Einheit definiert werden.

Name	Type	Value	Description [1]
Temperature	REAL		Current room temperature in [°C]

Die Definition der Einheit einer Prozessvariable erfolgt über die OPC UA Konfiguration.

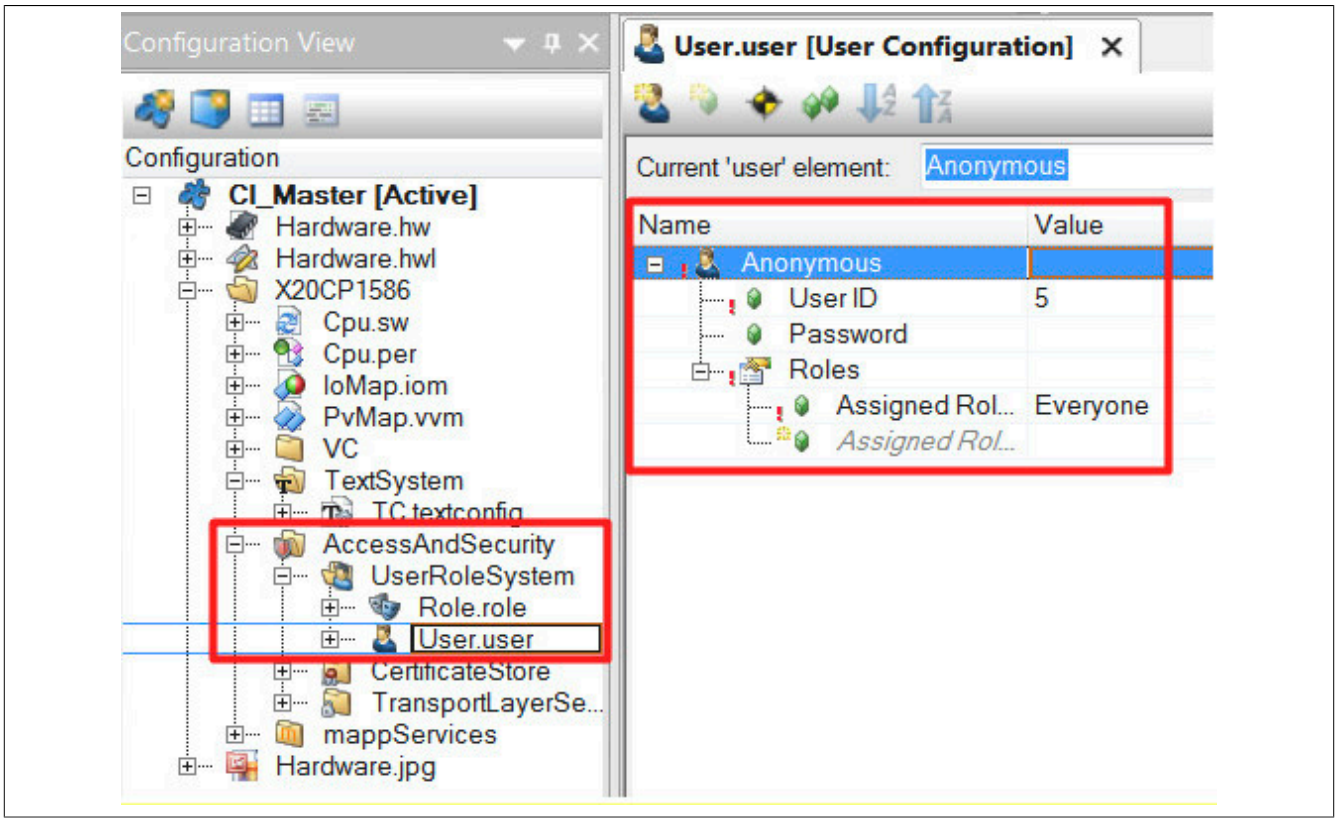
Properties - Temperature

Name	Value	Value
Temperature		
Enable	True	
Audit Events	Off	
Minimum Sampling Interval (ms)		
Authorization		
EU Range Violation		
EU Range	<Default>	Add Role
Engineering Unit		
Name	degree Celsius	
Symbol	°C	
Unit ID	4408652	
Namespace	http://www.opcfoundation.org/UA/units/un/c...	
Authorization		
Minimum Sampling Interval (ms)		

Wichtig hierbei ist, dass der OPC-UA Server in der Konfiguration des Zielsystems aktiviert ist:

OPC-UA System	
Activate OPC-UA System	on
Network Settings	
Port number	4840

Es gilt zu beachten, dass für die Funktion der Benutzer "Anonymous" mit der Rolle "Everyone" benötigt wird. Diese sind standardmäßig im Benutzer-Rollen-System von Automation Studio enthalten und sollten nicht verändert werden.



Ist die Einheit einer Prozessvariable definiert, kann über das Einheitensystem zwischen verschiedenen Einheiten umgerechnet werden, zum Beispiel:

- [mm] auf [m]
- [°C] auf [°F]
- [bar] auf [Pa]

Aus den Beispielen ist ersichtlich, dass zum Einen auf verschiedene Größen innerhalb eines Einheitensystems umgewandelt wird ([mm] auf [m]), zum Anderen zwischen verschiedenen Einheitensystemen gewechselt wird ([°C] auf [°F]).

Für folgende Anwendungsfälle ist es wichtig die Einheit definieren zu können:

- Darstellung einer Prozessvariable auf der Visualisierung
- Export eines Maschinenberichts (MpReport) inklusive verschiedener Prozessdaten
- Darstellung der Alarmhistorie über die Visualisierung, wobei viele der Alarmtexte Prozesswerte enthalten (z.B. die überhöhte Temperatur, die aktuelle Geschwindigkeit, etc ...)
- Ablegen von Audit - Ereignissen in einer Datenbank in einem bestimmten Einheitensystem (z.B. Metrisch), unabhängig davon, welches Einheitensystem derzeit auf der Visualisierung angezeigt wird (z.B. Imperial-US)

Bei den Einheiten müssen folgende Begriffe unterschieden werden:

Bezeichnung	**Enginnering-Einheit**	**Anzeige-Einheit**
Abkürzung	E-Unit	D-Unit
Beschreibung	Wird über die OPC UA Konfiguration festgelegt. Beschreibt die Einheit, in welcher die Prozessvariable in der Applikation verarbeitet wird.	Die Anzeige-Einheit wird auf verschieden Arten festgelegt, Detail siehe weiter unten. Beschreibt mit welcher Einheit eine Prozessvariable auf der Visualisierung oder in einer Export-Datei angezeigt werden soll. Anzeige-Einheiten müssen immer für alle möglichen Einheitensystem definiert werden.
Beispiel	Wird die Variable "Temperatur" in der Applikation verwendet, ist diese grundsätzlich einheitenlos, meist wird aber von einer bestimmten Einheit ausgegangen, z.B. [°C]. Für die Applikation selbst ist dies jedoch belanglos.	Soll die aktuelle Temperatur auf der Visualisierung dargestellt werden, muss definiert werden in welcher Einheit diese dargestellt werden soll. Reicht die Engineering-Einheit oder muss explizit eine Anzeige-Einheit festgelegt werden, z.B. in einem anderen Einheitensystem [imperial-US, °F], da die Maschine nach Amerika geliefert wurde.

Tabelle 5: E-Unit & D-Unit

Soll sich die Anzeige-Einheit von der Engineering-Einheit unterscheiden, so muss diese definiert werden. Das erfolgt auf zwei Arten:

- Die Definition der Anzeige-Einheit für jedes Einheitensystem direkt auf der Visualisierung am Widget.
  Wird ein einzelner Prozesswert an ein Widget angebunden (z.B. NumericInput-Widget), dann kann die Anzeige-Einheit direkt über die Eigenschaften des Widgets definiert werden.

unit	{"metric":"1","imperial":"2","imperial-us":"2"};

- Definiton der Anzeige-Einheit für jedes Einheitensystem über die MpComUnit Konfiguration
  Müssen für mehrere Prozesswerte Anzeige-Einheiten definiert werden, dann kann dies über die MpComUnit Konfiguration erreicht werden. Für Widgets wie das AlarmList-Widget, oder das Sequencer-Widget ist es nicht möglich direkt am Widget die Anzeige-Einheit festzulegen, da verschiedenste Einheiten und Mengen an Werten dargestellt werden (z.B. sollen Alarme zu überhöhter Temperatur in [°C] genauso dargestellt werden, wie auch Alarme über zu niedrigen Druck in [Pa]). Weiters sollen die Anzeige-Einheiten auch nützbar sein für die Generierung von Berichten (z.B. Audit Trail, Alarm Historie, Maschinenberichte, ...).

Alarm Text Snippets	
Snippet: actPres	
Key	actPres
Value	Process Variable
Process Vari...	::Program:Pressure
Snippet: actTemp	
Key	actTemp
Value	Process Variable
Process Vari...	

Alarm List	
Alarm: PressureLow	
Name	PressureLow
Message	The current Pressure {&actPres}{&actPres[UNIT= %s]} is too low!
Code	0
Severity	2
Behavior	Persistent Alarm
Alarm: TemperatureHi...	
Name	TemperatureHigh
Message	The current temperature {&actTemp}{&actTemp[UNIT=%s]} is too high!
Code	0
Severity	2
Behavior	Persistent Alarm

Die Konfiguration der Einheiten sieht so aus:

Datapoints		
DataPoint 1		
PV Name	::Program:Pressure	
DisplayUnits		
Metric		
Unit-Code	PAL	Unit: Pascal
Imperial		
Unit-Code	PS	Unit: Pound per Square-Inch
ImperialUS		
Unit-Code	PS	Unit: Pound per Square-Inch
Custom Units		
Unit 1		
Unit-Code		
DataPoint 2		
PV Name	::Program:Temperature	
DisplayUnits		
Metric		
Unit-Code	CEL	Unit: Degree Celsius
Imperial		
Unit-Code	FAH	Unit: Degree Fahrenheit
ImperialUS		
Unit-Code	FAH	Unit: Degree Fahrenheit
Custom Units		
Unit 1		
Unit-Code		

Die hier konfigurierten Alarme könnten zur Laufzeit so aussehen:

InstanceID	Name	Message	Code	Severity	StateActive	StateAck	Timestamp
2	TemperatureHigh	The current temperature 120°C is too high!	0	2	false	false	2017-06-09 11:20:13:960
3	PressureLow	The current Pressure 80 kPa is too low!	0	2	true	false	2017-06-09 11:20:52:760

Wurde die Anzeige-Einheit definiert, so erfolgt die Darstellung, sowie Umrechnung des Wertes automatisch für folgende Komponenten:

- **Audit Trail:** Einheitenumrechnung für alle Archive und Export-Dateien, sowie die aktuelle Ereignisliste von MpAuditTrailUI
- **mapp AlarmX: Alarmverwaltung:** Alarmlist-Widget stellt Werte mit der gewünschten Einheit um geforderten Einheitensystem dar. Beim Exportieren der Alarm Historie werden Anzeigen-Einheiten berücksichtigt
- **Maschinenberichte:** Beim Erstellen von Maschinenberichten, werden für alle Prozesswerte die Einheiten berücksichtigt

Ist die Anzeige-Einheit nicht definiert für einen Prozesswert, wird zuerst auf die Engineering-Einheit zurückgegriffen. Falls auch diese fehlt dann wird die Variable ganz ohne Einheit angezeigt.

Darstellung der Einheit innerhalb eines Textes

Neben dem Darstellen und Umrechnen der Prozesswerte, welche Teil eines Textes sind (z.B. Alarme, Audit-Ereignisse), kann definiert werden in welcher Form die Einheit dargestellt werden soll:

- **Symbol %s:** Hiermit wird definiert, dass das Symbol der aktuellen Anzeige-Einheit angezeigt wird (z.B. "mm").
- **Kurztext %n:** Hiermit wird definiert, dass der vollständige Name der aktuellen Anzeige-Einheit angezeigt wird (z.B. "Millimeter").
- **Beschreibung %d:** Hiermit wird definiert, dass die Beschreibung der aktuellen Anzeige-Einheit angezeigt wird (z.B. "Abstand gemessen in Millimeter"). Nicht für jede Einheit existiert auch eine Beschreibung.

Die Umrechnung erfolgt automatisch, die Darstellung der Einheit wird über folgende Syntax definiert:

{&key}{&key[UNIT= xxx]}

- {&key}: Definiert die Prozessvariable, von welcher der Wert angezeigt werden soll. Der Wert wird automatisch auf die aktuelle Anzeige-Einheit umgerechnet. Wurde keine Anzeige-Einheit definiert, wird der Wert in seiner Engineering-Unit angegeben.
- {&key[UNIT=xx]}: Druckt den Einheiten-Text für die Einheit, die dem Wert zugeordnet ist.
  Unter "xx" können die vorher erwähnten Parameter (%s, %n, %d) angegeben werden.

Einige Beispiele:

- "Die aktuelle Temperatur {&actualTemperature}{&actualTemperature[UNIT=%s]} hat den kritischen Wert erreicht!"
  Die aktuelle Temperatur 120°C hat den kritischen Wert erreicht!
- "Die Achse steht an ihrer Endposition: {&actualPosition}{&actualPosition[UNIT= %n]}!"
  Die Achse steht an ihrer Endposition: 500.0 Millimeter!
- "Der aktuelle Druck {&currentPressure}{&currentPressure[UNIT= %s]} überschreitet den empfohlenen Maximalwert!"
  Der aktuelle Druck 50 bar überschreitet den empfohlenen Maximalwert!

Definition von Sprache und Einheit für Anzeige und Export

Die Darstellung der Einheit wird im Punkt "Darstellung der Einheit innerhalb eines Textes" definiert. Beim Export bzw. bei der Anzeige ist es allerdings notwendig die Sprache, sowie das Einheitensystem anzugeben. Das erfolgt bei den oben genannten Komponenten über den Eingang, bzw. Parameter "Language". Um beim Export bzw. der Anzeige Sprache und Einheitensystem festzulegen ist der Eingang "Language" auf folgende Weise zu verwenden:

'Sprache|Einheitensystem'

Einige Beispiele:

- **'de'** : Sprache ist Deutsch, Werte in Engineering-Einheit
- **'en|imperial-us'** : Sprache Englisch, Einheitensystem ist Imperial-US

- **'de|metric'** : Sprache Deutsch, Einheitensystem ist Metrisch
- **'|imperial'** : Default-Sprache, Einheitensystem ist Imperial

Details zur Definition der Anzeige-Einheiten sind im Abschnitt MpComUnit Konfiguration zu finden.

10.1.2 Rahmenbedingungen

In diesem Abschnitt werden die Rahmenbedingungen von mapp Com aufgelistet.

Allgemeines

- mapp Com kann nur mit mapp Services Komponenten verwendet werden. Komponenten von anderen mapp Paketen, wie beispielsweise mapp Motion, können nicht mit mapp Com verwendet werden.

10.2 Konfiguration

10.2.1 MpComGroup Konfiguration

Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein de-, und wieder aktiviert werden.

Die geänderten Konfigurationsparameter werden sofort von der Komponente übernommen.

10.2.2 MpComUnit Konfiguration

Wie die Konfigurationsparameter zur Laufzeit geändert werden können, wird hier im Abschnitt "Konfiguration zur Laufzeit ändern" erklärt.

Damit die geänderten Konfigurationsparameter übernommen werden, muss der Funktionsbaustein de-, und wieder aktiviert werden.

Die geänderten Konfigurationsparameter werden sofort von der Komponente übernommen.

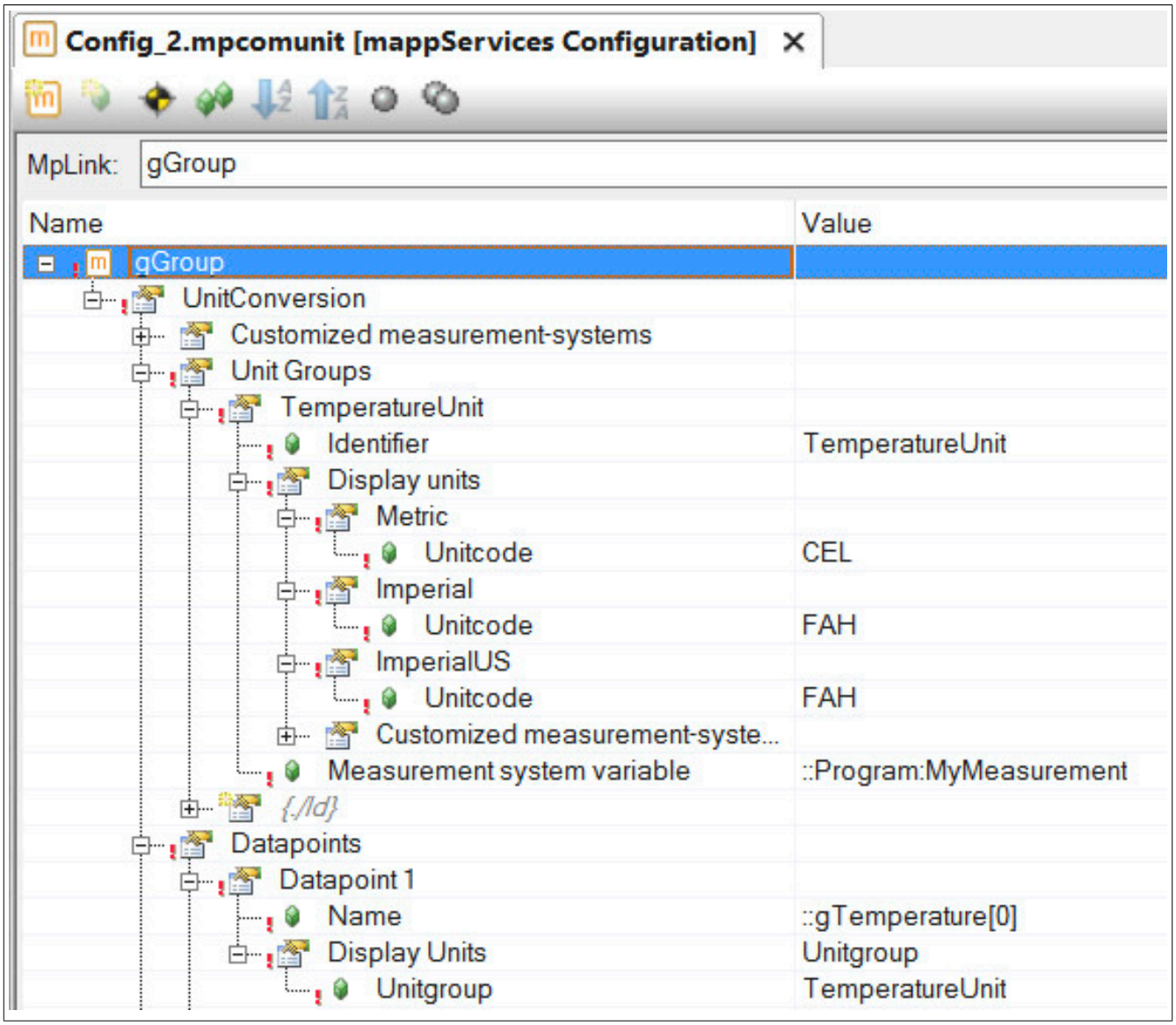
Benutzerdefinierte Einheit

Über "Customized measurement-systems" kann eine Benutzerdefinierte Einheit angegeben werden. Über "Identifier" wird der Name der Benutzerdefinierten Einheit bestimmt. Zusätzlich muss eine Fallback Einheit angegeben werden. Dabei muss eine der 3 Standard-Einheiten Imperial, Imperial-US oder Metric ausgewählt werden.

Einheitengruppen

Unter "Unit groups" können Einheitengruppen definiert werden. Hat man zum Beispiel in einem Projekt eine Temperatur-Variable vom Datentyp INT[100], so müsste für jede Array-Variable eine Einheit pro Einheitensystem (Imperial, Metrisch,...) angegeben werden. Mithilfe einer Einheitengruppe kann man zum Beispiel die Gruppe "TemperatureUnit" erstellen, wo einmal festgelegt wird welche Einheit pro Einheitensystem verwendet werden soll. Anschließend verwendet die Temperatur-Variable die Einheitengruppe "TemperatureUnit". Der Konfigurationsaufwand wird dadurch verringert.

Die Einheitengruppe wird über "Identifier" eindeutig im Projekt identifiziert.



Unter "Measurement system variable" kann eine Variable vom Datentypen STRING angehängt werden. Diese Variable bestimmt das Einheiten-System das für diese Gruppe verwendet wird. Hat die PV den Wert "metric", so wird, wie im obigen Beispiel gezeigt, CEL verwendet. Bei der Angabe von "imperial" wird FAH verwendet. Wurde zum Beispiel die Benutzerdefinierte Einheit "MySystem" angegeben, so kann als ID "MySystem" verwendet werden.

3		
4	Timestamp	gTemperature[0] [degree Celsius]
5	2019 01 30 07:36:04:823	30
6	2019 01 30 07:36:06:823	5
7		

Abbildung 27: MyMeasurement = 'metric'

3		
4	Timestamp	gTemperature[0] [degree Fahrenheit]
5	2019 01 30 07:35:17:823	41
6	2019 01 30 07:35:19:823	86
7		

Abbildung 28: MyMeasurement = 'imperial'

Datapoints

Unter "Datapoints" können Prozessvariablen angegeben werden, auf welche ein Einheitensystem angewendet werden soll. Dabei wird die "Engineering Unit" in der OPC UA Konfiguration angegeben. Diese Konfiguration stellt die Möglichkeit zur Verfügung, die Prozessvariablen für die Anzeige auf einer mappView Visualisierung auf andere Einheiten umzurechnen.

Unit-code: Der Einheitencode definiert die auf der mappView Visualisierung anzuzeigende Einheit.

Namespace: Über den Namespace wird definiert, wo die Einheitendefinition gefunden werden kann. Wird nur für Einheiten benötigt, welche nicht in den verfügbaren Standardeinheiten enthalten sind.

Für jede Prozessvariable können zusätzlich zu den Einheiten in Metrisch, Imperial und Imperial-US auch Einheiten für Benutzerdefinierte Einheitensysteme hinzugefügt werden. Es besteht auch die Möglichkeit eine Einheitengruppe auszuwählen. Dabei wird unter "Display Units" "Unitgroup" ausgewählt.

Units

Unter "Units" können Standardeinheitenumrechnungen angegeben werden. Damit kann eine Einheitenumrechnung für eine bestimmte Einheit angegeben werden. Diese Umrechnung wird dann auf alle Prozessvariablen angewendet, welche in OPC UA mit dieser Einheit definiert wurden.

Engineering Unit: Die "Engineering Unit", welche für die Anzeige auf der mappView Visualisierung umgerechnet werden soll.

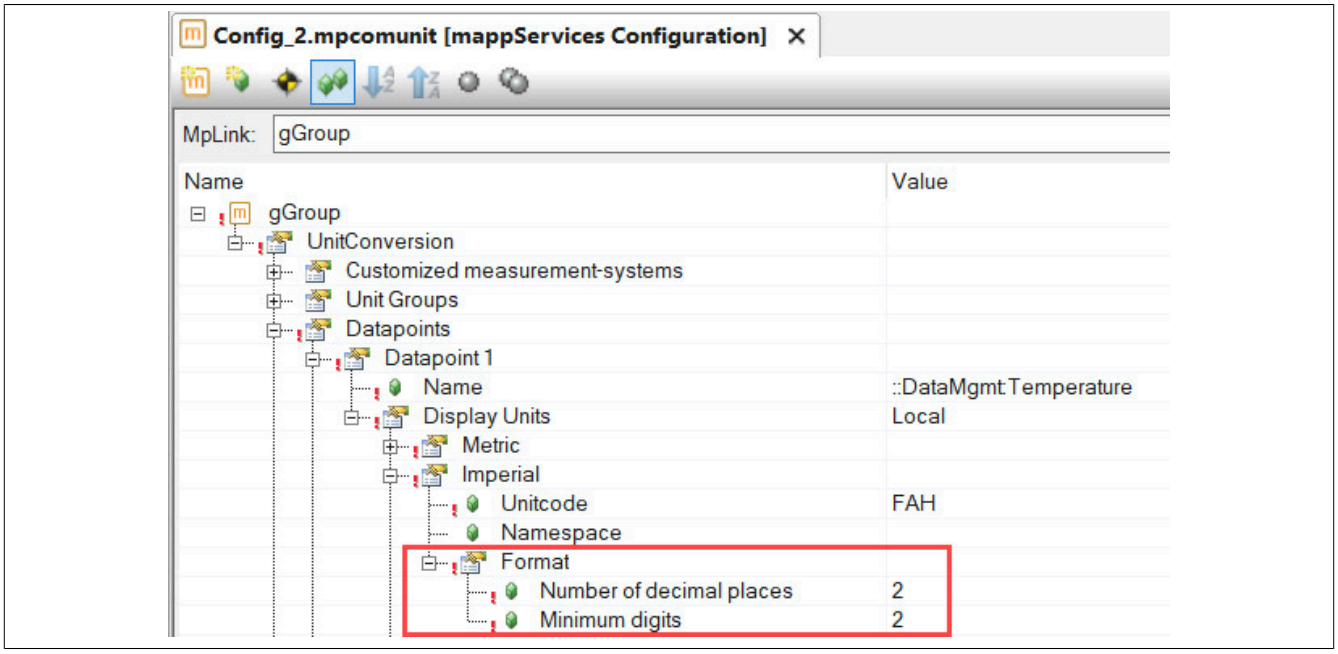
Unit-code: Der Einheitencode definiert die auf der mappView Visualisierung anzuzeigende Einheit.

Namespace: Über den Namespace wird definiert, wo die Einheitendefinition gefunden werden kann. Wird nur für Einheiten benötigt, welche nicht in den verfügbaren Standardeinheiten enthalten sind.

Für jede "Engineering Unit" können zusätzlich zu den Einheitenumrechnungen zu Metrisch, Imperial und Imperial-US auch Umrechnungen zu Benutzerdefinierten Einheiten konfiguriert werden.

Festlegen eines Formats

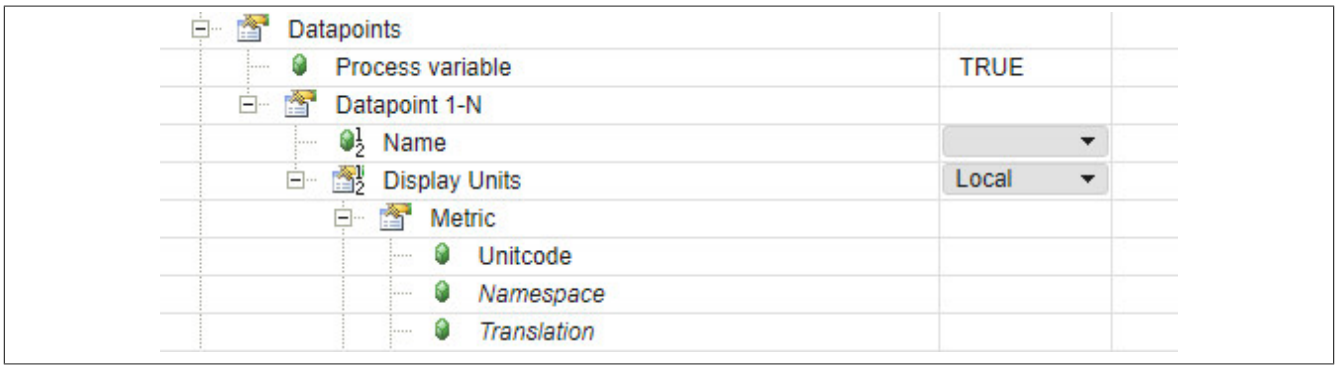
Bei Aktivierung der erweiterten Parameter kann das Format einer Prozessvariable, Einheit oder Einheitengruppe definiert werden. Dabei wird festgelegt wieviele Nachkommastellen und welche Mindestanzahl an Gesamtstellen verwendet werden soll. Führende Ziffern, die nicht verwendet werden, werden durch Leerzeichen ersetzt. Wird beispielsweise der Wert "5" aufgezeichnet und die Mindestanzahl an Gesamtstellen beträgt 2, so wird " 5" angezeigt.



Die hier definierten Einstellungen wirken sich direkt auf die mapp Komponenten aus, welche Prozessvariablen mit einer Einheit aufzeichnen bzw. anzeigen können. Dies ist beispielsweise in mapp Data oder mapp AlarmX der Fall.

Festlegen einer Einheitenumrechnung

Werden die erweiterten Parameter aktiviert, so erscheint der Parameter "Translation" in den einzelnen Abschnitten.

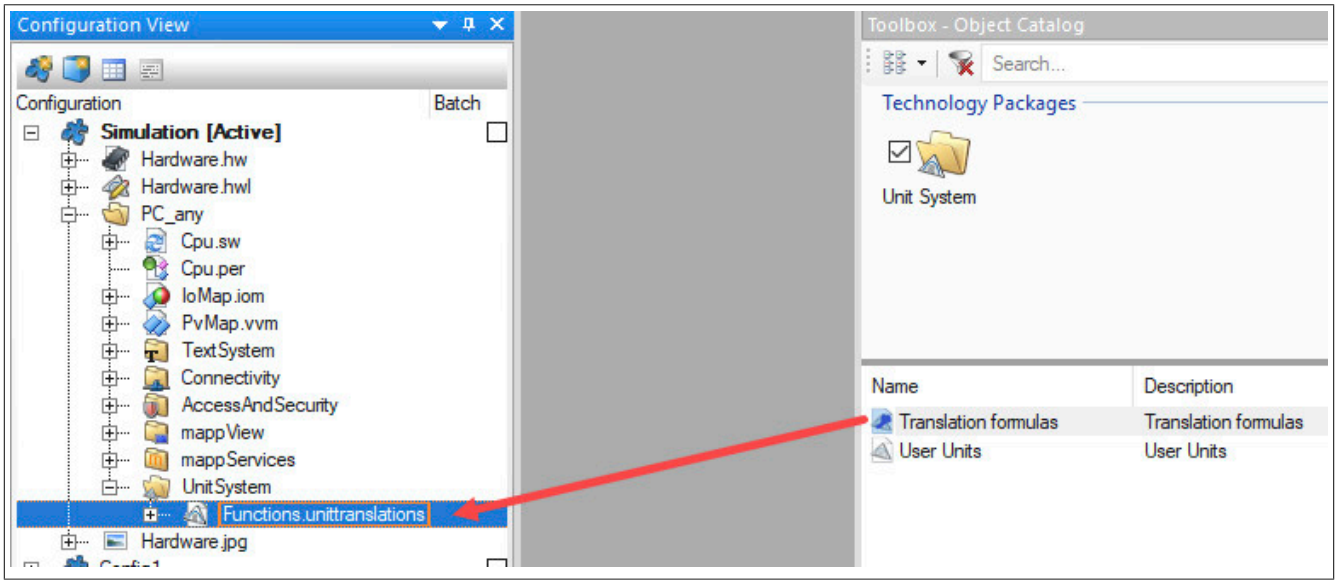


Die Einheitenumrechnung kann verwendet werden, wenn die Engineering-Einheit nicht kompatibel zur Anzeige-Einheit ("Display-Unit") ist. Die Umrechnung der Engineering-Einheit zur Anzeige-Einheit wird dabei selbst definiert und erfolgt im Einheiteneditor. Für mehr Information wie eine Umrechnung erstellt wird, siehe Abschnitt Unit System.

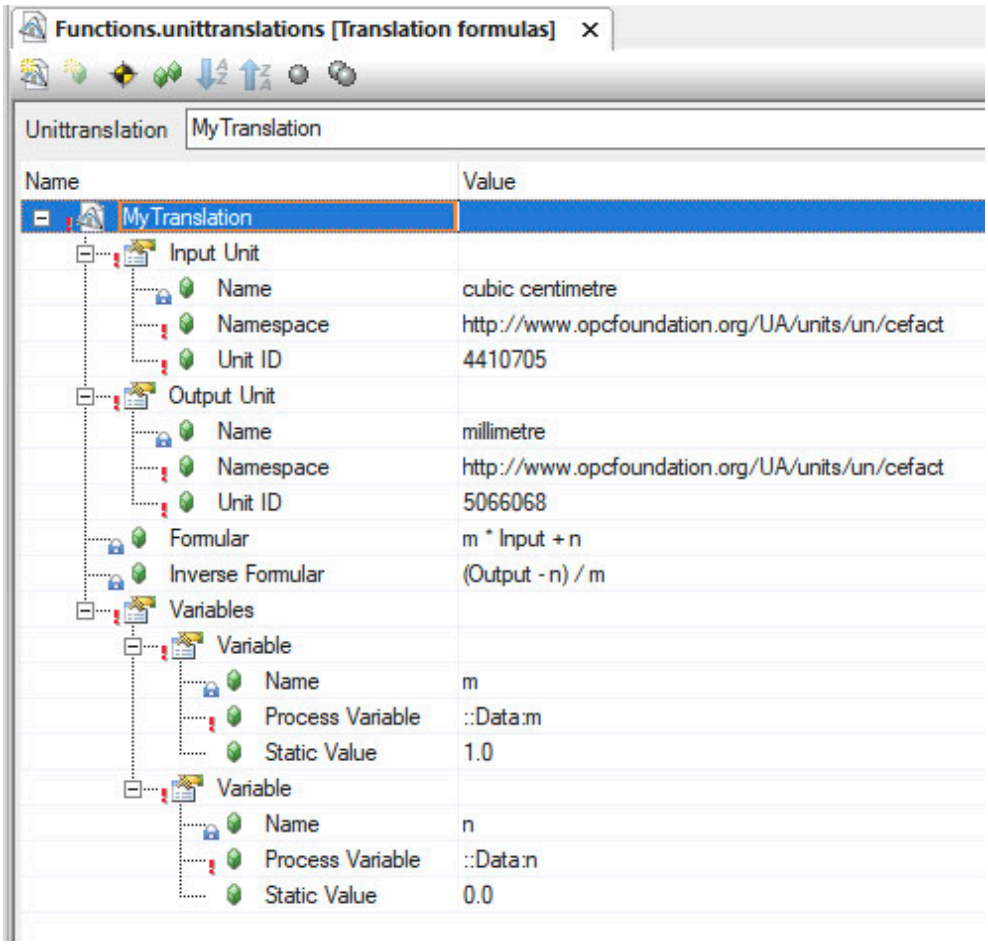
> **Der Parameter "Translation" sowie die Umrechnung der Einheit im Einheiteneditor ist nur verfügbar bei einer Automation Studio Version >= 4.10.**

Beispiel:

Als Engineering-Einheit wird für eine Prozessvariable Kubikzentimeter("Input Unit") definiert. Als Anzeige-Einheit ("Output Unit") soll aber die Einheit Millimeter verwendet werden. Die Umrechnung der Engineering-Einheit auf die Anzeige-Einheit ist nicht automatisch möglich, da die beiden definierten Einheiten nicht miteinander kompatibel sind. Die Umrechnung wird im Einheiteneditor des Unit System durchgeführt. Dazu wird die Datei "Translation formulas" eingefügt. In dieser Datei wird die Umrechnung festgelegt.

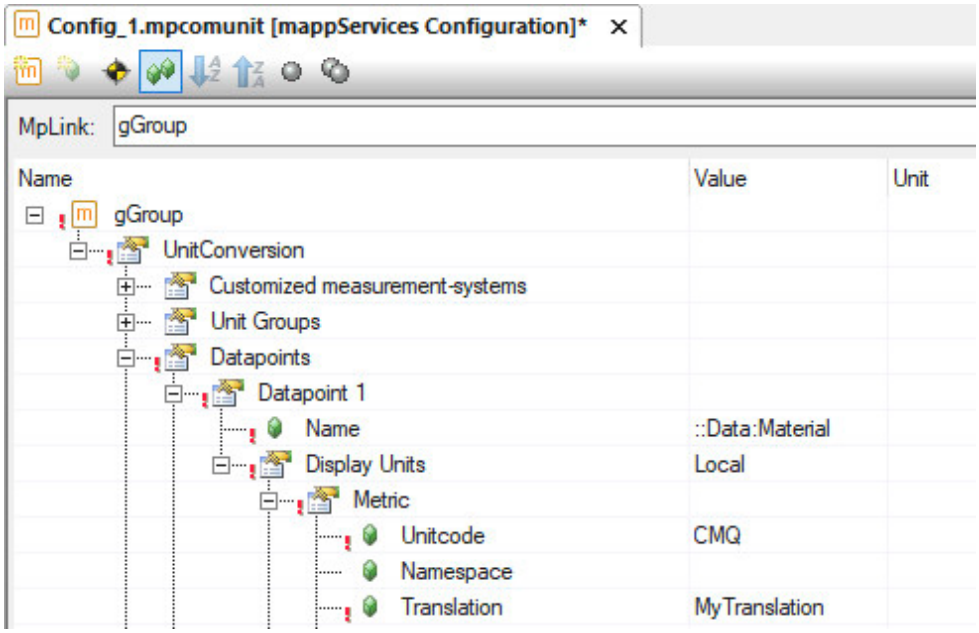


Es wurde die Umrechung "DistanceToVolume" definiert. Diese kann folgendermaßen aussehen:



Unter "Variables" wird definiert mit welchem Faktor die Engineering-Einheit multipliziert bzw. dividiert wird.

Der Name der zuvor definierten Umrechnung wird in der MpComUnit Konfiguration am gewünschten Datenpunkt angegeben.



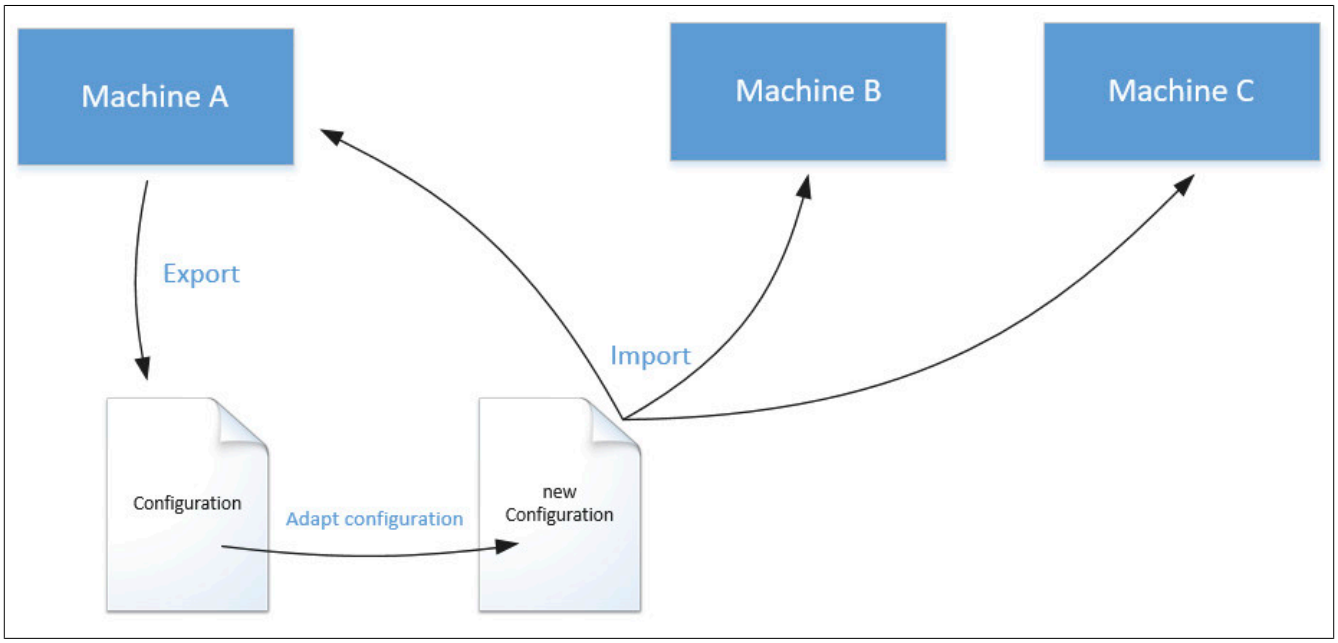
Zu diesem Thema gibt es auch den Anwendungsfall Verwenden der Einheitenumrechnung.

10.3 Anwendungsfälle

10.3.1 Importieren von Benutzeralarmen

Anforderung

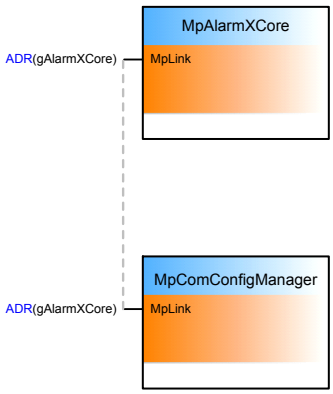
Die Gegebenheiten in einer Fabrik haben sich geändert. Dadurch mussten die Grenzwerte von Alarmen angepasst sowie neue Benutzeralarme in einer Maschine erstellt werden. Der Maschinentyp existiert 3 mal in der Fabrik. Damit die Alarmkonfiguration nicht pro Maschine neu adaptiert werden muss, soll die einmal bearbeitete Alarmkonfiguration für alle Maschinen importiert werden können.



Lösung

Komponentenliste

- ***MpComConfigManager***: Importiert die Alarmkonfiguration.
- ***MpAlarmXCore:*** Sammelt und verwaltet alle Alarme



Alarmkonfiguration

Die Alarme der Maschine werden über die MpAlarmXCore Konfiguration verwaltet.

Wie eine Alarmverwaltung erstellt werden kann, ist im Abschnitt Benutzeralarme mit VC4 mapp Vorlage zu finden.

Verwendung der mapp Komponenten

Zur Laufzeit soll mit Hilfe von **MpComConfigManager** die bestehende MpAlarmXCore Konfiguration exportiert und auf anderen Maschinen importiert werden.

Der Funktionsbaustein **MpComConfigManager** wird eingefügt und parametriert:

```
MpComConfigManager_0.Enable := TRUE;
MpComConfigManager_0.MpLink := ADR(gAlarmXCore);
MpComConfigManager_0.DeviceName := ADR('HD');
MpComConfigManager_0.FileName := ADR('MyAlarmConfig');

MpComConfigManager_0();
```

Als mapp Link wird der mapp Link des MpAlarmXCore Funktionsbausteins verwendet, da die Alarmkonfiguration exportiert bzw. importiert werden soll.

Danach wird mithilfe von "Export = `TRUE`" die Alarmkonfiguration auf das angegebene Speichermedium exportiert. Die Export-Datei kann dabei folgendermaßen aussehen: MyAlarmConfig

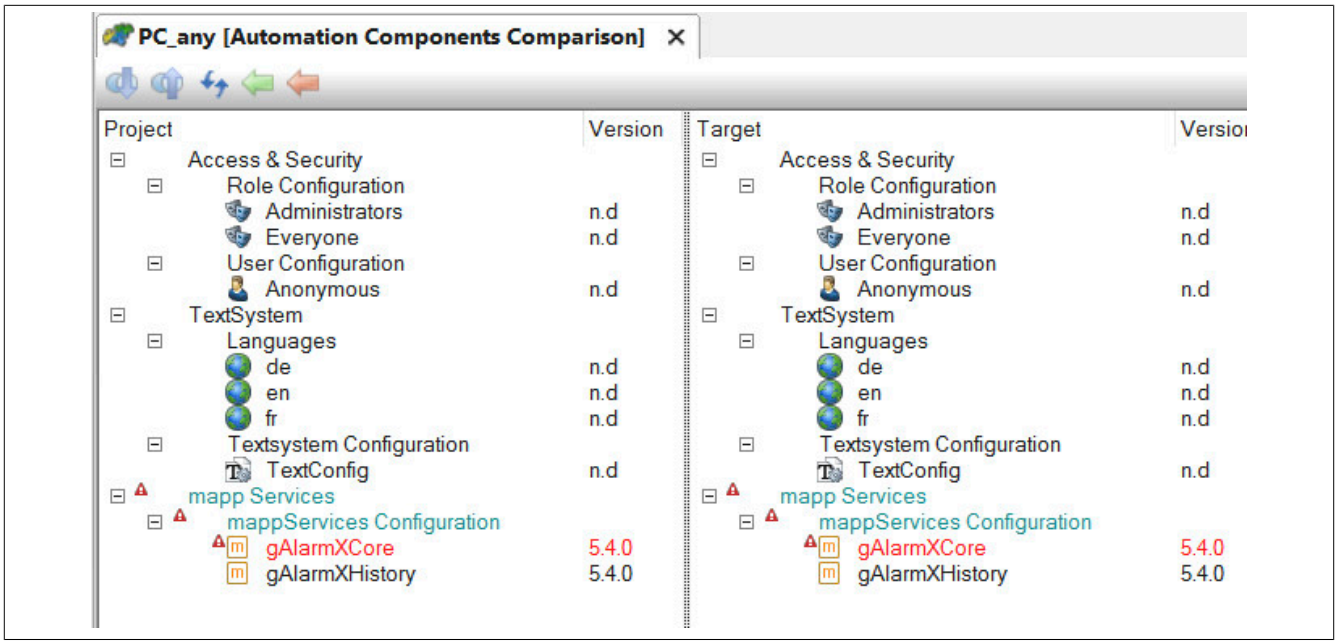
Die exportierte Datei kann nun bearbeitet werden. Die Benutzeralarme können nun angepasst oder neue erstellt werden. Jeder neuer Alarm muss dabei eine neue Gruppen ID ("Group ID") und einen neuen Alarmnamen ("Name") erhalten. Ein neuer Alarm kann zum Beispiel folgendermaßen aussehen:

```
...

<Group ID="[856]">
    <Property ID="AdditionalInformation1" DataType="STRING" Value="" />
    <Property ID="AdditionalInformation2" DataType="STRING" Value="" />
    <Property ID="Code" DataType="UDINT" Value="0" />
    <Property ID="Disable" DataType="BOOL" Value="false" />
    <Property ID="Message" DataType="STRING" Value="The temperature is too low" />
    <Property ID="Name" DataType="STRING" Value="TemperatureLow" />
    <Property ID="Severity" DataType="UDINT" Value="1" />
    <Group ID="Behavior">
        <Property ID="Acknowledge" DataType="DINT" Value="1" />
        <Property ID="AutoReset" DataType="BOOL" Value="false" />
        <Property ID="MultipleInstances" DataType="BOOL" Value="false" />
        <Property ID="ReactionUntilAcknowledged" DataType="BOOL" Value="true" />
        <Property ID="Type" DataType="STRING" Value="PersistentAlarm" />
        <Group ID="Recording">
            <Property ID="ActiveToInactive" DataType="BOOL" Value="false" />
            <Property ID="InactiveToActive" DataType="BOOL" Value="true" />
            <Property ID="UnacknowledgedToAcknowledged" DataType="BOOL" Value="true" />
        </Group>
</Group>

...
```

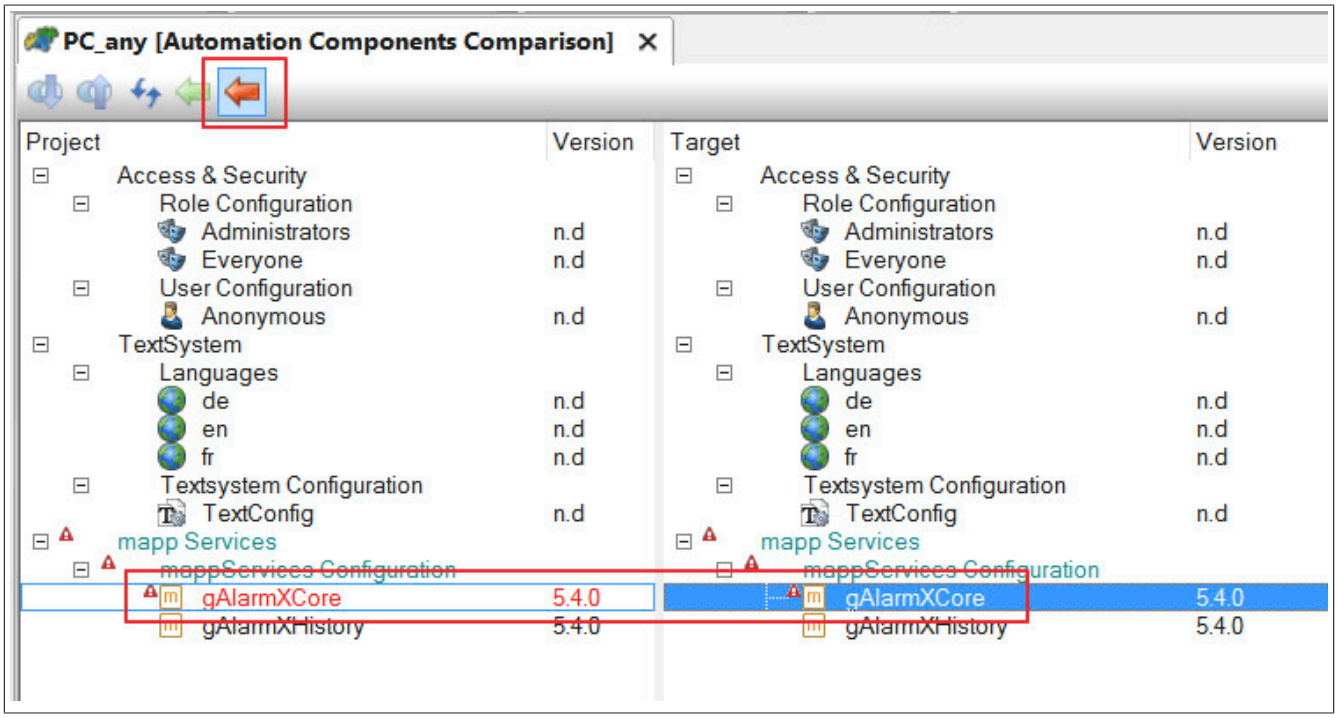
Die bearbeitete Datei wird anschließend an allen Maschinen über "Import = `TRUE`" an **MpComConfigManager** wieder importiert. Die adaptierte Alarmkonfiguration wird nun an an allen Maschinen verwendet.

Zusätliche Information:

Der Funktionsbaustein **MpComConfigManager** hat die angepassten und neuen Benutzeralarme in die MpAlarmXCore Konfiguration importiert. Die Änderung ist über Online > Vergleich > Automation Components ersichtlich:



Die neuen Alarme existieren aktuell nur in der Konfiguration welche sich auf dem Zielsystem befindet. Damit die neuen Benutzeralarmen auch in der Konfiguration im AS Projekt sind, wird der Befehl "nach links Spiegeln" ausgeführt. Die Alarme sind nun auch im AS Projekt verfügbar.



10.3.2 Ändern und importieren der Konfiguration durch MpComConfigManager

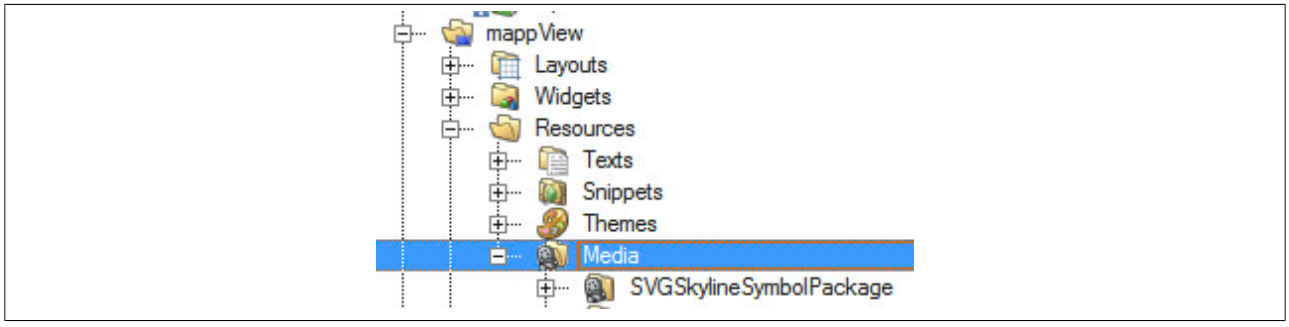
Anforderung

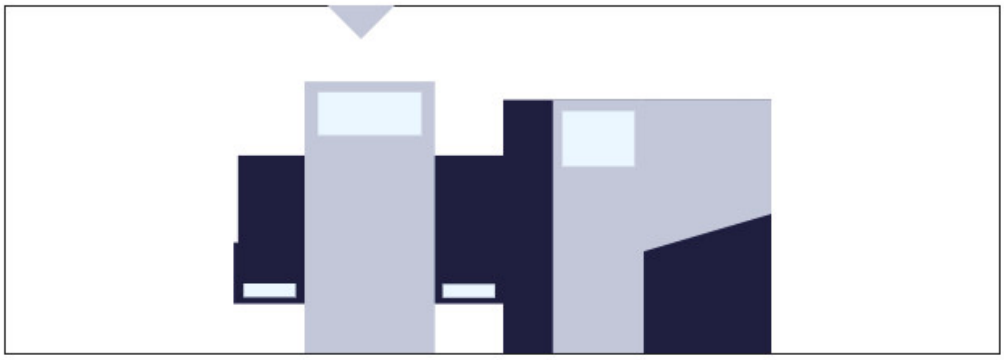
Eine Drucker-Maschinenlinie wird mithilfe von mapp Skyline und dem Skyline-Widget angezeigt

Diese Drucker-Maschinenlinie soll erweitert werden. Aktuell besteht die Maschinenline aus einem Printing-Modul und einem Stacker-Modul. Nun soll ein weiteres Printing-Modul ("Printer cyan") zur Laufzeit hinzugefügt werden.

Die in der Visualisierung angezeigte Skyline soll ebenfalls aktualisiert werden, damit alle bestehenden Module vorhanden sind.

> **In der von mapp View bereitsgestellten SVG Skyline Bibliothek sind Bilder enthalten, welche für die Skyline verwendet werden können. Die Bibliothek ist über die Toolbox verfügbar und kann im mapp View Paket in der Logical View unter Resources -> Media abgelegt werden.**





mapp Skyline

Wie mapp Skyline und das Widget eingefügt und parametriert wird, ist im Getting Started Maschinenlinie visualisieren erklärt.

Vorbereiten der Konfiguration

Als erstes muss die bereits vorhandene Konfiguration durch den Funktionsbaustein MpComConfigManager exportiert werden. Die Datei wird dabei auf einem Speichermedium, wie zum Beispiel einen USB-Stick, abgespeichert. Die exportierte Datei kann anschließend mithilfe eines Texteditors bearbeitet werden.

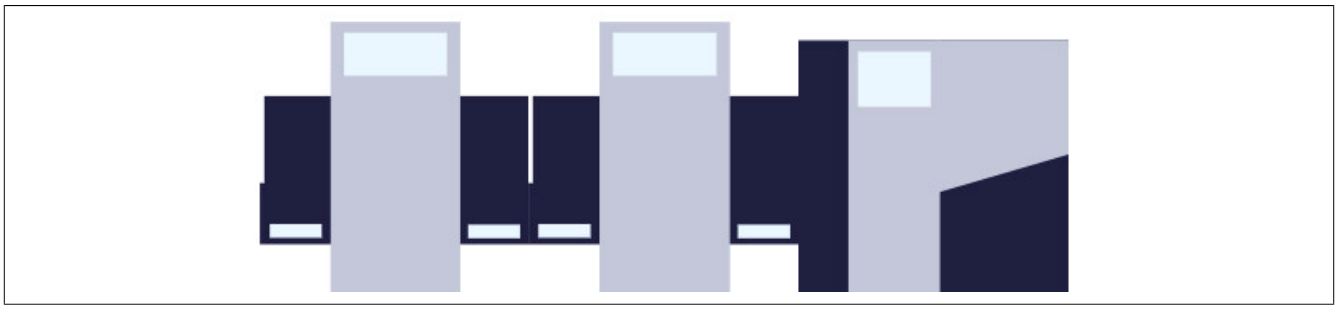
Es wird ein zustätzliches Modul "Printer_Cyan" unter "Modules" eingefügt.

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
    <Version ID="1" />
    <Element ID="gSkyline">
        <Group ID="MpSkyline">
            <Group ID="ModuleStateCollections">
            <Group ID="ModuleTypes">
            <Group ID="Modules">
                <Group ID="[0]">
                    <Property ID="Grouping" DataType="DINT" Value="1" />
                    <Property ID="Name" DataType="STRING" Value="Printer_Black" />
                    <Property ID="Text" DataType="STRING" Value="" />
                    <Property ID="Type" DataType="STRING" Value="PrintingUnit" />
                </Group>
                <Group ID="[1]">
                    <Property ID="Grouping" DataType="DINT" Value="1" />
                    <Property ID="Name" DataType="STRING" Value="Printer_Cyan" />
                    <Property ID="Text" DataType="STRING" Value="" />
                    <Property ID="Type" DataType="STRING" Value="PrintingUnit" />
                </Group>
                <Group ID="[2]">
                    <Property ID="Grouping" DataType="DINT" Value="1" />
                    <Property ID="Name" DataType="STRING" Value="Stacker" />
                    <Property ID="Text" DataType="STRING" Value="" />
                    <Property ID="Type" DataType="STRING" Value="StackerUnit" />
                </Group>
            </Group>
            <Group ID="NotificationCollections">
            <Group ID="OptionStateCollections">
```

Die Group-ID muss entsprechend angepasst werden. Jedes Modul braucht eine eigene ID, welche nicht mehrfach vorkommen darf. Die Konfiguration wird gespeichert und kann verwendet werden.

Konfiguration zur Laufzeit aktualisieren

Über den Funktionsbaustein MpComConfigManager wird die zuvor bearbeitete Konfiguration importiert. Nachdem die Visualisierung aktualisiert wurde, wird die neue Skyline angezeigt.



10.3.3 Daten mit Einheit aufzeichnen

Anforderung

In einer Maschine soll ein Temperaturwert alle 2 Sekunden aufgezeichnet werden. Die aufgezeichneten Daten sollen in einer Visualisierung sichtbar sein. Der Temperaturwert soll mit einer Einheit dargestellt werden.

Abhängig davon welches Einheitensystem verwendet wird, soll der Temperaturwert in der Visualisierung in unterschiedlichen Einheiten angezeigt werden können.

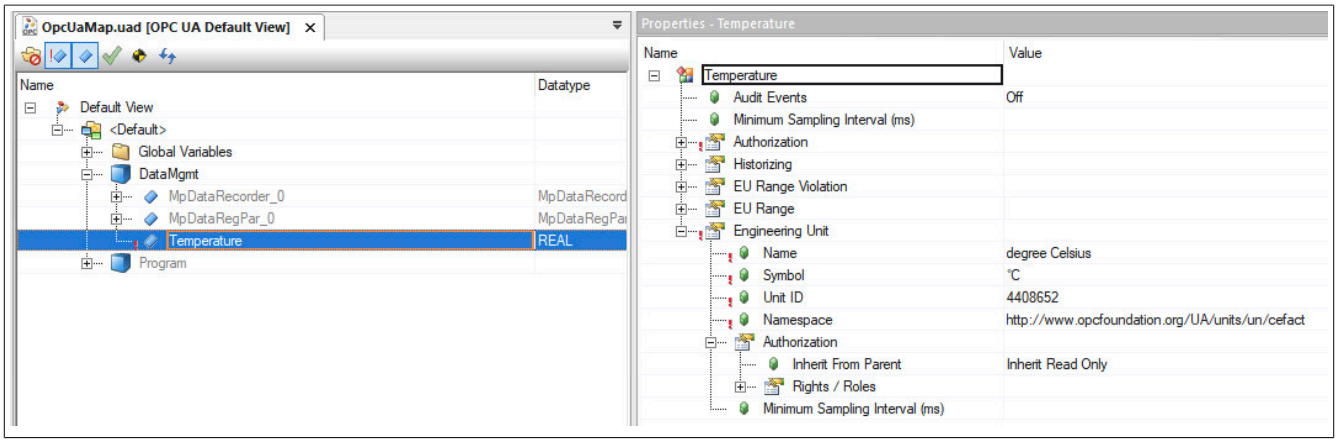
Es kann zwischen 3 verschiedenen Einheitensystemen gewählt werden. Dem metrischen, imperialen und dem imperialen US System.

Bei der Auswahl des metrischen Systems soll der Temperaturwert in Grad Celsius dargestellt werden. Wird hingegen ein imperiales System gewählt, soll der Temperaturwert in Grad Fahrenheit angezeigt werden.

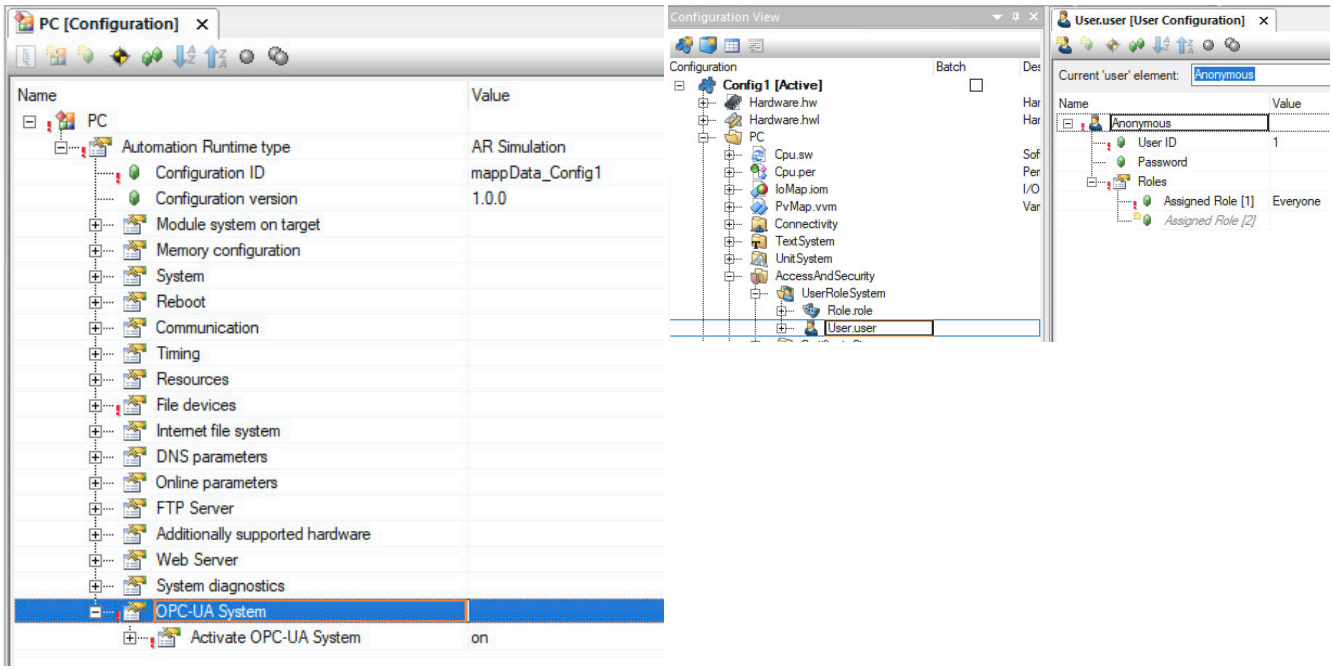
Festlegen der Einheit

Um die Einheit einer Prozessvariable zu definieren muss die Engineering-Einheit, sowie die Anzeige-Einheit festgelegt werden. Im ersten Schritt muss die Engineering-Einheit für eine Prozessvariable definiert werden, für welche im späteren Verlauf die Anzeigeeinheit umgeschaltet werden soll. Die Einheit wird dabei in der OPC UA Default View Konfiguration angegeben.

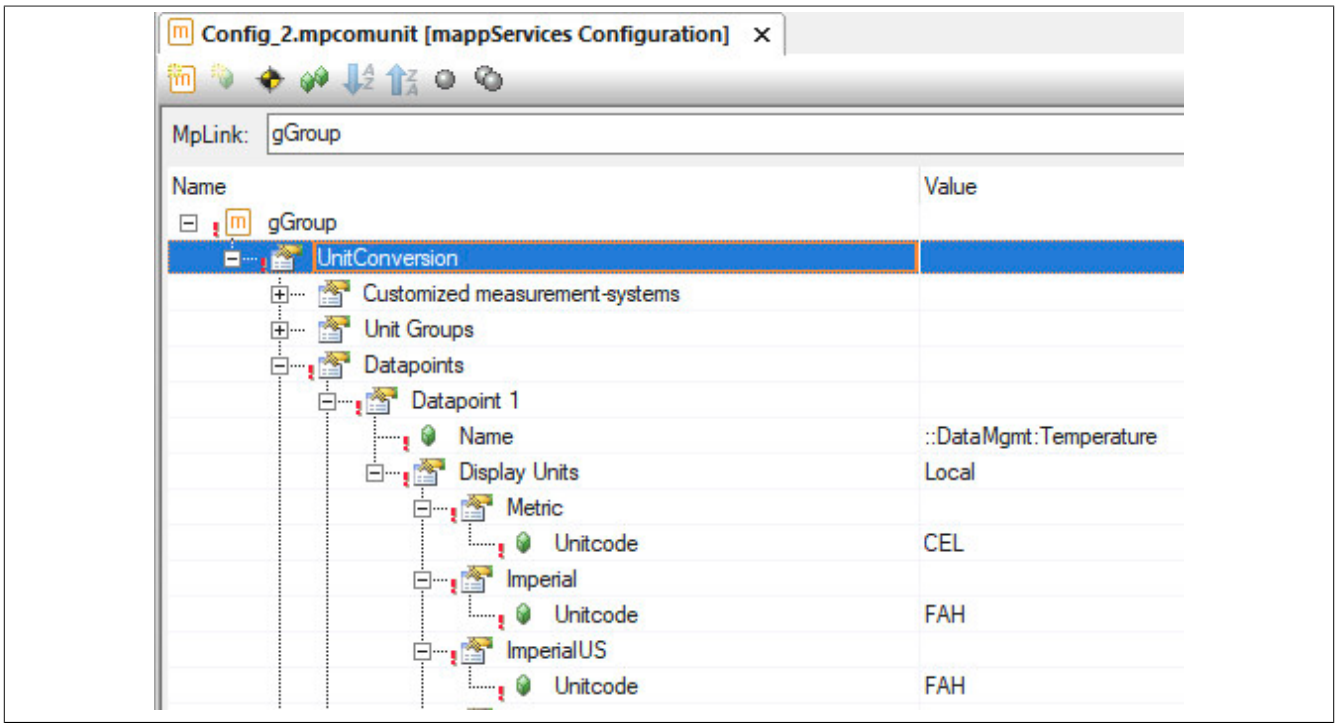
Die hier definierte Einheit ist die sogenannte Engineering-Einheit. Das bedeutet diese Einheit wird in der Applikation für die Prozessvariable verwendet. Für die Temperatur-Prozessvariable soll als Engineering-Einheit Grad Celsius definiert werden. Für mehr Information siehe Abschnitt Unit System.



Damit das Einheitensystem verwendet werden kann, muss der OPC-UA Server des verwendeten Zielsystems aktiviert werden. Außerdem muss das Benutzer-Rollen-System eingefügt werden, indem der Benutzer "Anonymous" die Rolle "Everyone" hat.

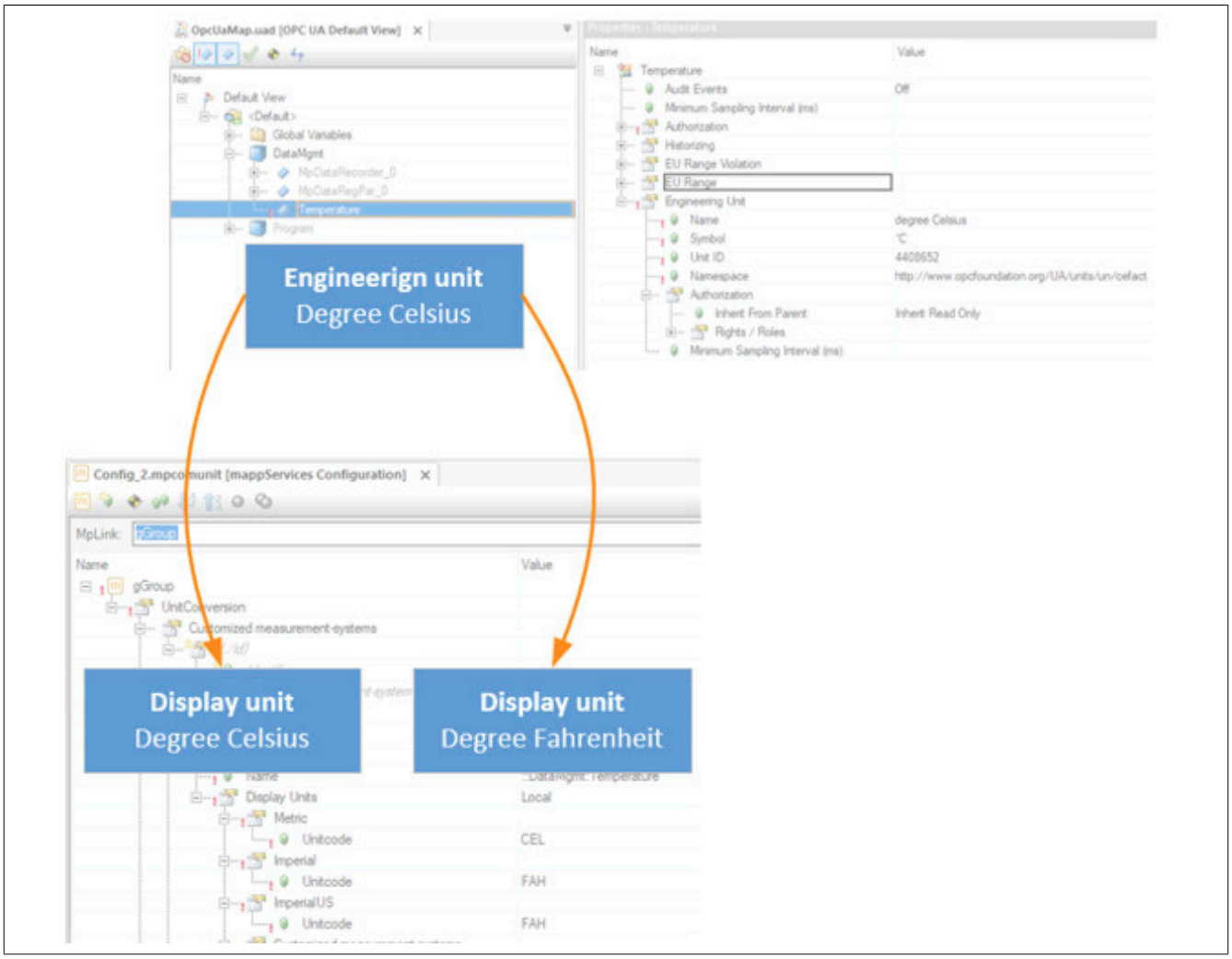


Abhängig davon welches Einheitensystem verwendet wird, muss die Prozessvariable die richtige Einheit anzeigen. Das bedeutet, im nächsten Schritt wird die Anzeige-Einheit definiert. Bei der Verwendung des metrischen Systems soll die Prozessvariable in Grad Celsius dargestellt werden. Wird hingegen das Imperial-, oder Imperial-US-System gewählt, soll die Anzeige der Prozessvariable in Grad Fahrenheit erfolgen. Die Zuordnung zwischen Einheitensystem und Anzeigeeinheit für die Prozessvariable erfolgt in der MpComUnit Konfiguration. Dabei wird die gewünschte Prozessvariable unter Datapoints angegeben.



Abhängig vom Einheitensystem muss die Einheit in Form eines Einheitencodes angegeben werden. Welche Einheit welchen Einheitencode verwendet, ist hier zu finden. Grad Celsius besitzt den Einheitencode CEL und Grad Fahrenheit FAH.

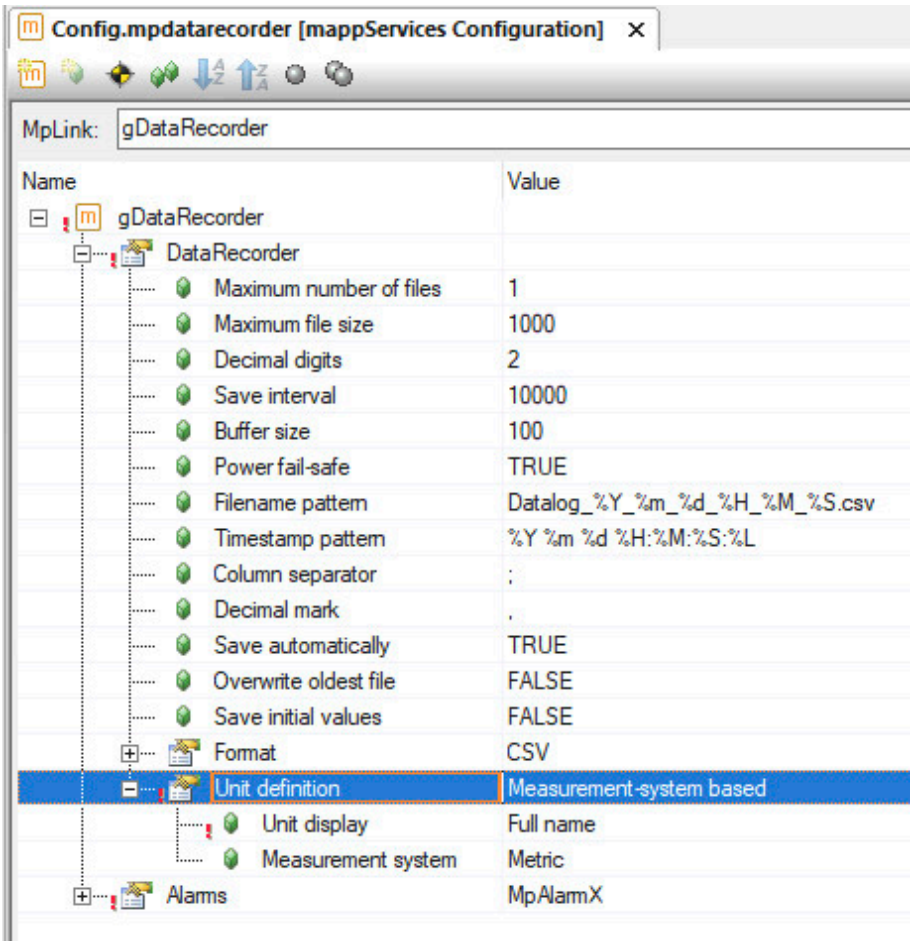
Die Umrechnung von Grad Celsius auf Grad Fahrenheit erfolgt automatisch, da die festgelegte Engineering-Einheit Grad Celsius ist. Von dieser Einheit aus kann auf die gewünschten Anzeige-Einheiten umgerechnet werden. Für mehr Information zu Einheitenverwaltung in mapp Komponenten siehe Einheitenverwaltung.



Konfiguration

Um Prozessvariablen aufzuzeichnen wird die MpDataRecorder Konfiguration eingefügt.

Welches Einheitensystem in mapp Data verwendet werden soll wird in der Konfiguration unter „Unit definition“ festgelegt. Es wird „Measurement-system based“ ausgewählt. Das Einheitensystem wird unter „Measurement system“ definiert. Es soll das Metrische System verwendet werden. Unit display gibt an wie die Einheit der PV in mapp Data dargestellt werden soll. Die Einheit kann als Kürzel, mit vollem Namen oder gar nicht angezeigt werden. Es wird der volle Einheiten-Name ausgewählt.



Verwendung der mapp Komponenten

MpDataRecorder sammelt und verwaltet alle Daten-Aufzeichnungen. Als Aufnahmemodus wird "mpDATA_RECORD_MODE_TIME" verwendet. Als "SamplingTime" werden 2 Sekunden angegeben.

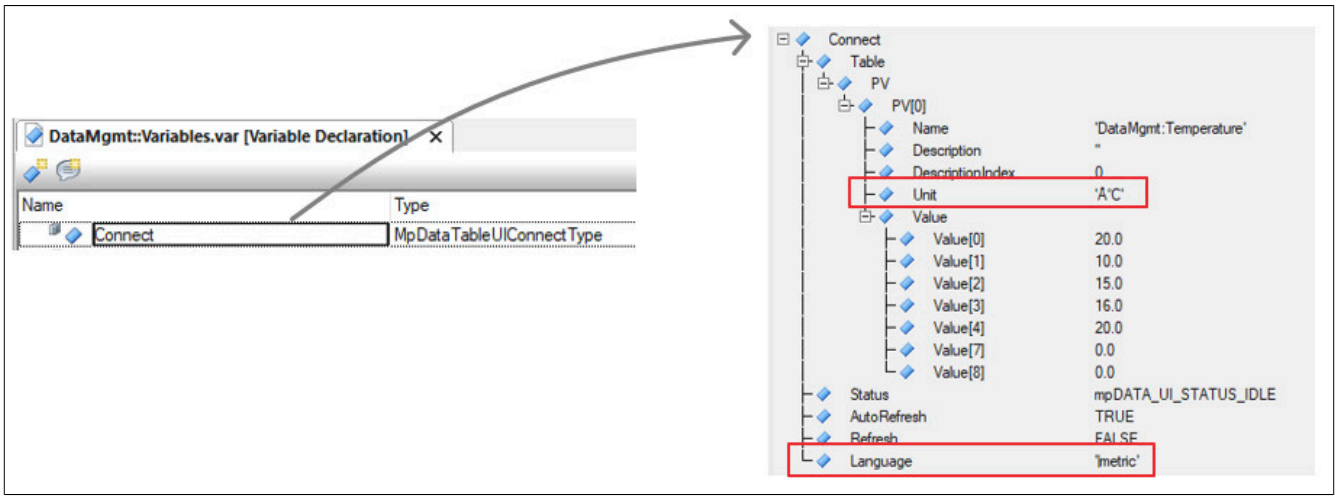
Mithilfe von **MpDataRegPar** wird bestimmt welche Prozessvariable aufgezeichnet werden soll. Es wird die Temperatur-Prozessvariable angegeben.

Wie MpDataRecorder und MpDataRegPar verwendet und parametriert werden, ist im mapp Data Tutorial zu finden.

Mithilfe von "Record =*TRUE*" an **MpDataRecorder** wird die Aufzeichnung gestartet. Es wird, wie vorher konfiguriert, alle 2 Sekunden die Prozessvariable aufgezeichnet.

MpDataTableUI kann dazu verwendet werden die aufgezeichneten Daten an einer Visualisierung darzustellen. Am Eingangsparameter "UIConnect" werden die aufgezeichneten Daten angezeigt. Dazu muss eine Prozessvariable vom Datentypen MpDataTableUIConnectType erstellt werden und an "UIConnect" angehängt werden. Diese Prozessvariable kann mit einer Visualisierung verbunden werden.

Um nun zu bestimmen in welcher Einheit der aufgezeichnete Temperaturwert in der UIConnect-Prozessvariable dargestellt werden soll, wird der Parameter "Language" verwendet. Bei der Angabe von '|metric' wird das metrische System verwendet. Bei '|imperial' das imperiale System. Für weitere Information siehe Abschnitt Einheiten von Prozessdaten.



10.3.4 Verwenden der Einheitenumrechnung

Anforderung

Eine Maschine füllt Material in Formen ein. Abhängig von der verwendeten Form sind unterschiedliche Materialmengen notwendig. Die benötigte Materialmenge wird über die Visualisierung definiert.

Intern im Programm wird die Materialmenge in Kubikzentimeter verwaltet (= Engineering-Einheit). Über die Visualisierung wird zur besseren Darstellung und je nach verwendeten Messsystem, die Materialmenge in Millimeter oder Inch angegeben. Wird das metrische System verwendet soll die Eingabe in Millimeter erfolgen. Bei der Auswahl von Imperial bzw. Imperial-US wird der Wert in Inch eingegeben (= Anzeige-Einheit).

Da die Engineering-Einheit (Kubikzentimeter) mit der Anzeige-Einheit (Millimeter bzw. Inch) nicht kompatibel ist, muss eine Einheitenumrechnung durchgeführt werden.

Bei einer Änderung der Materialmenge soll ein Eintrag in eine Ereignisliste erfolgen, dabei soll auch die Einheit der Materialmenge angezeigt werden. Um die geänderten Materialmengen besser analysieren zu können, soll die Ereignisliste auf ein Speichermedium exportiert werden können.

Umsetzung

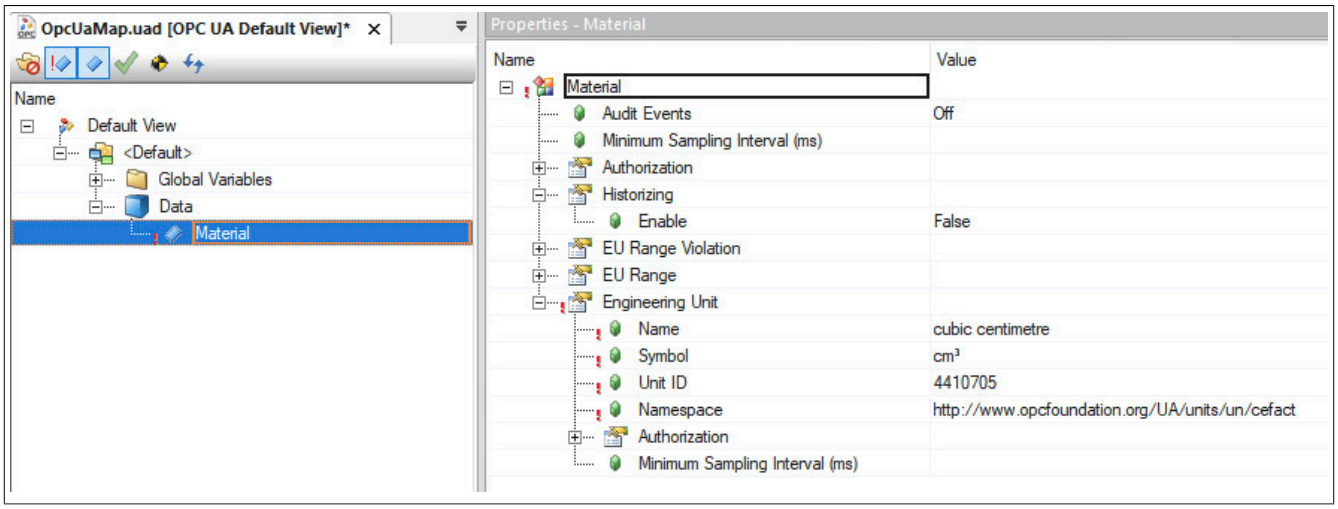
Um diesen Anwendungsfall zu lösen wird der Funktionsbaustein MpAuditTrail benötigt. Durch MpAuditTrail werden alle gewünschten Ereignisse gesammelt. Die Einheitenumrechnung erfolgt mithilfe der Datei "Translation formulas" im Unit System von Automation Studio. Die Visualisierung wird über mapp View realisiert.

Wie die benötigten Komponenten verwendet werden, is nun Schritt für Schritt erklärt.

> **Um die Funktion der Einheitenumrechnung verwenden zu können wird eine Automation Studio Version >= 4.10 benötigt.**

OPC UA Konfiguration

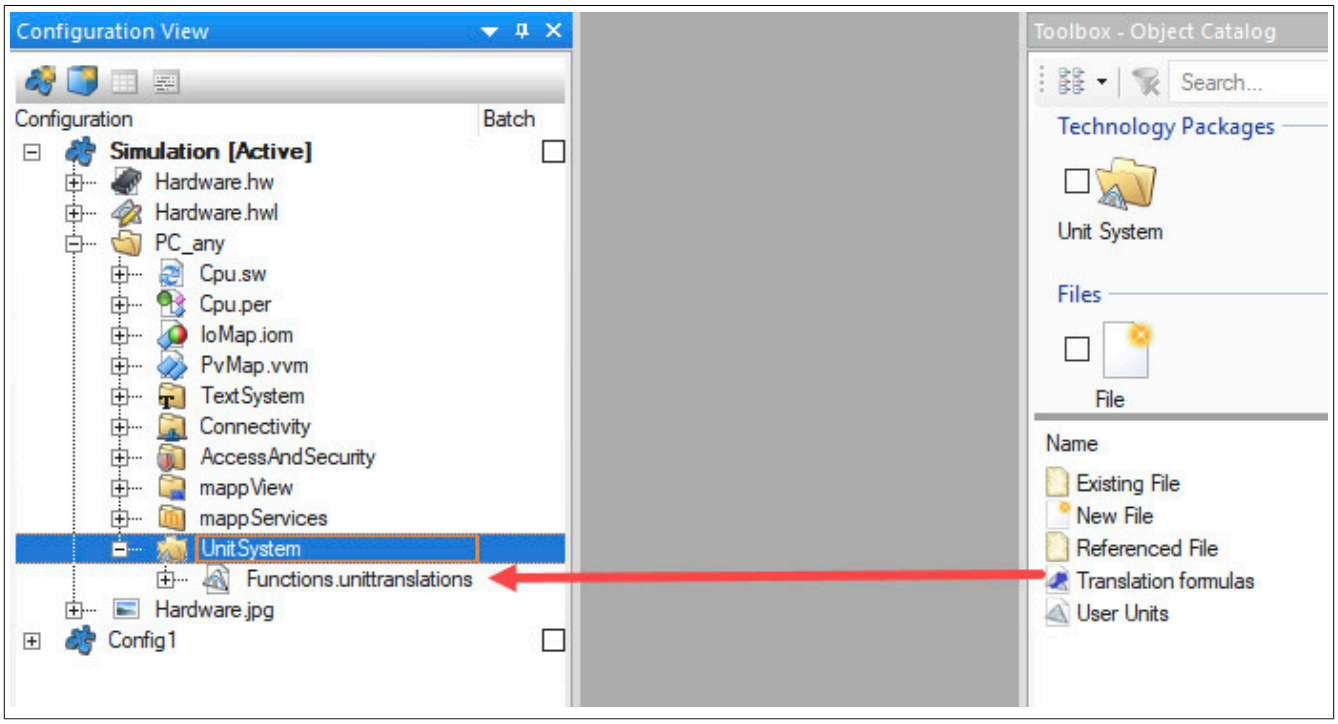
Die Einheit einer Prozessvariable wird in der OPC UA Default View Konfiguration definiert. Die Materialmenge wird über die Prozessvariable "Material" definiert. Die Prozessvariable wird für die OPC UA Kommunikation freigeschalten um die Engineering-Einheit Kubikzentimeter anzugeben.



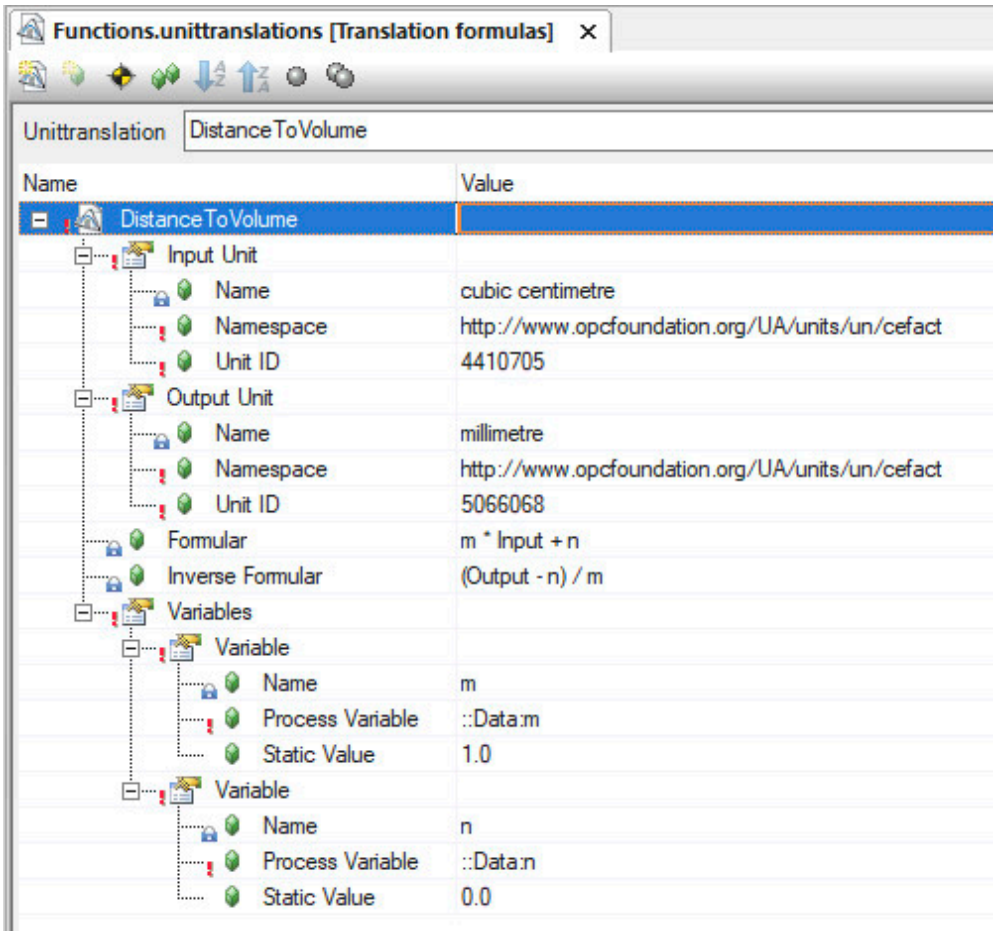
Einheitenumrechnung

Da als Engineering Einheit für die Prozessvariable Kubikzentimeter verwendet wird, die Angabe der Materialmenge über die Visualisierung aber Millimeter bzw. Inch erfolgt, muss eine Einheitenumrechnung definiert werden. In dieser Einheitenumrechnung wird festgelegt, wie die Angabe in Millimeter auf Kubikzentimeter umgerechnet wird.

Für die Einheitenumrechnung wird die Datei "Translation formulars" eingefügt. Für mehr Information zur Datei siehe Abschnitt Unit System.



In der Datei wird die Einheitenumrechnung "DistanceToVolume" definiert. An "Input Unit" wird die Engineering-Einheit und an "Output Unit" die Anzeige-Einheit angegeben. Über die Variablen kann definiert werden mit welchem Wert die "Input Unit" bzw. die "Output Unit" multipliziert bzw. dividiert wird.

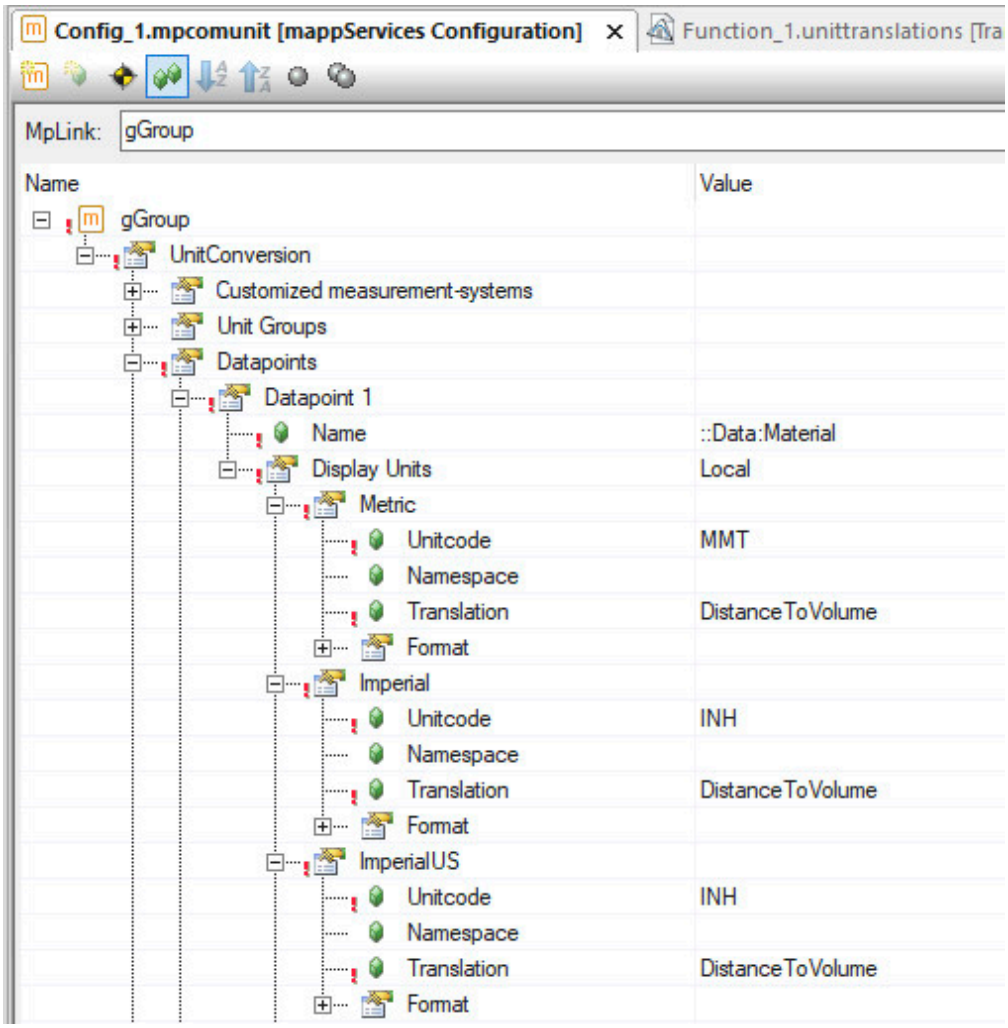


MpComUnit Konfiguration

Die Definition, welche Einheit in welchem Einheitensystem angezeigt werden soll, findet in der MpComUnit Konfiguration statt.

Unter "Datapoints" wird die aufzuzeichnende Prozessvariable "Material" angegeben. Im Abschnitt "DisplayUnits" wird festgelegt welche Anzeigeeinheit pro Einheitensystem verwendet werden soll. Im Metrischen System soll Millimeter verwendet werden. Im Imperial und Imperial-US soll Inch verwendet werden. Aus diesem Grund werden die Einheiten-Abkürzungen MMT bzw. INH verwendet. Damit die zuvor festgelegte Einheitenumrechnung für die Prozessvariable durchgeführt wird, wird unter dem Parameter "Translation" die Umrechnung angegeben. Der Parameter wird sichtbar wenn die erweiterten Parameter der Konfiguration aktiviert werden.

Welches Kürzel welche Einheit darstellt kann hier gefunden werden.



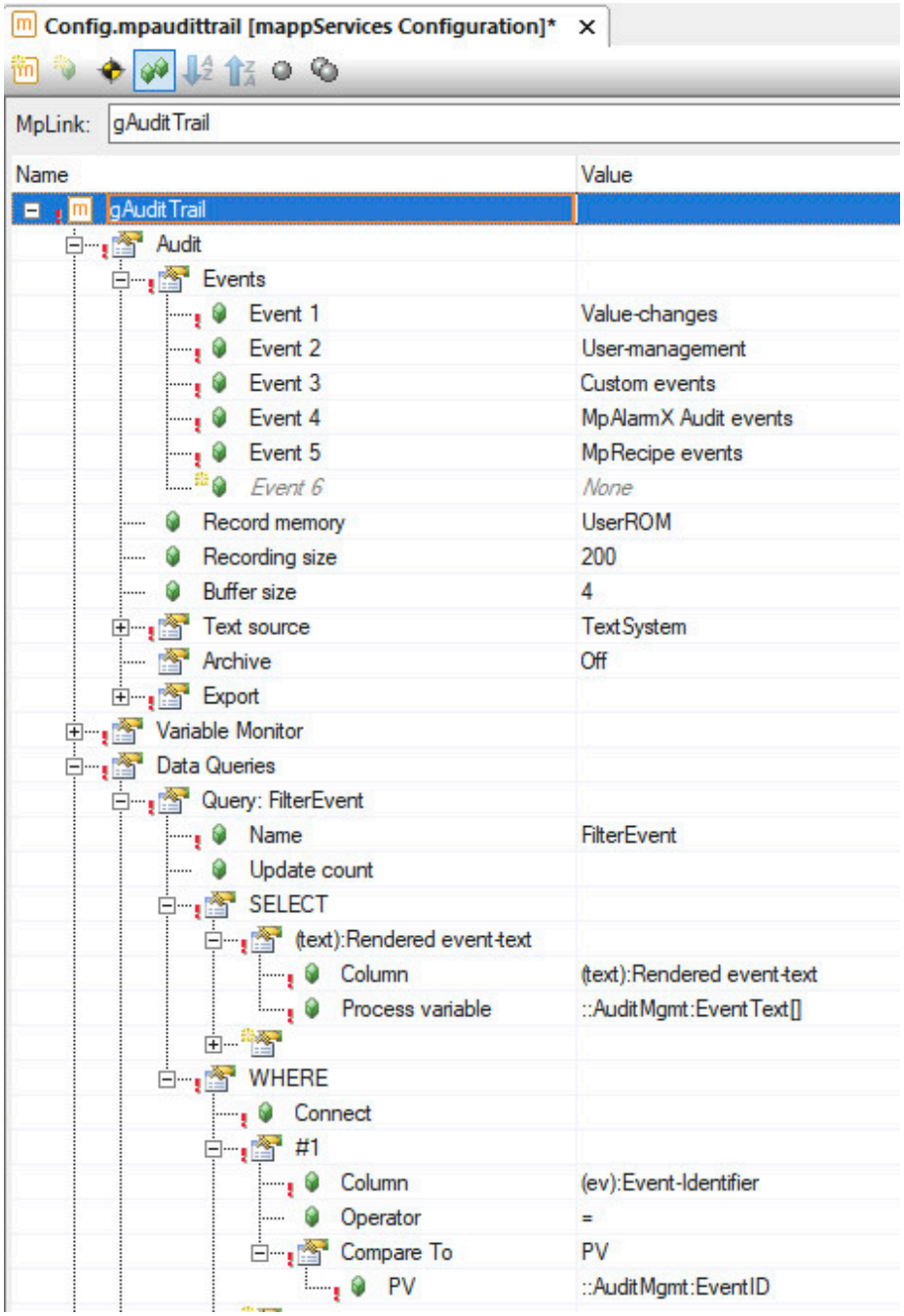
MpAuditTrail Konfiguration

Damit Ereignisse aufgezeichnet werden können, muss die MpAuditTrail Konfiguration eingefügt werden. Die aufzuzeichnenden Ereignisse werden unter "Events" ausgewählt. Für mehr Information dazu siehe Abschnitt Ereignisarten. Bei einer Änderung der Materialmenge soll ein Ereignis aufgezeichnet werden, aus diesem Grund wird das Ereignis "mapp Audit: Value change event" angegeben.

Die Prozessvariable "Material" wird unter "Monitored Values" angegeben. Dadurch wird die Prozessvariable von mapp Audit überwacht und bei Wertänderungen wird automatisch ein Ereignis erzeugt.

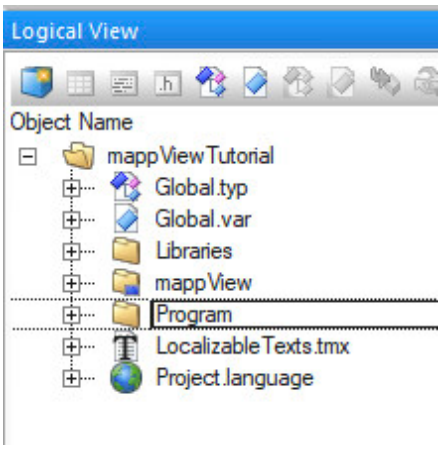
Zum Schluss wird der Ereignistext definiert. Als Textquelle wird das Textsystem definiert.

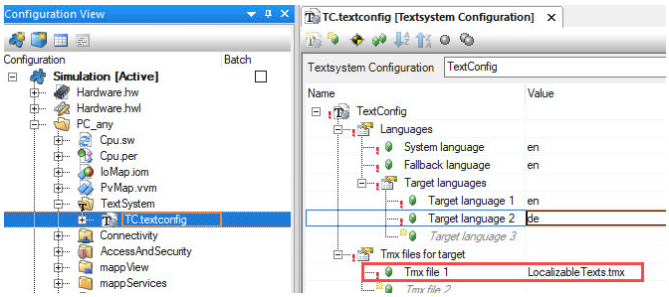
Der Ereignistext wird über "Text source" definiert. Es wird das Textsystem als Textquelle definiert. Wie die einzelnen Ereignistexte im Textsystem definiert werden, ist unter Definieren des Ereignistexts im Abschnitt "Textsystem als Textquelle" erklärt.



Textsystem Konfiguration

Der Text, welcher für das Ereignis ausgegeben wird, wird über das Textsystem definiert. Dazu wird eine .tmx Datei eingefügt, die zu verwendenden Projektsprachen, sowie die Konfiguration des Textsystems. In der Konfiguration muss die .tmx Datei unter "Tmx files for target" angegeben werden. Für mehr Information zum Textsystem siehe Textsystem.





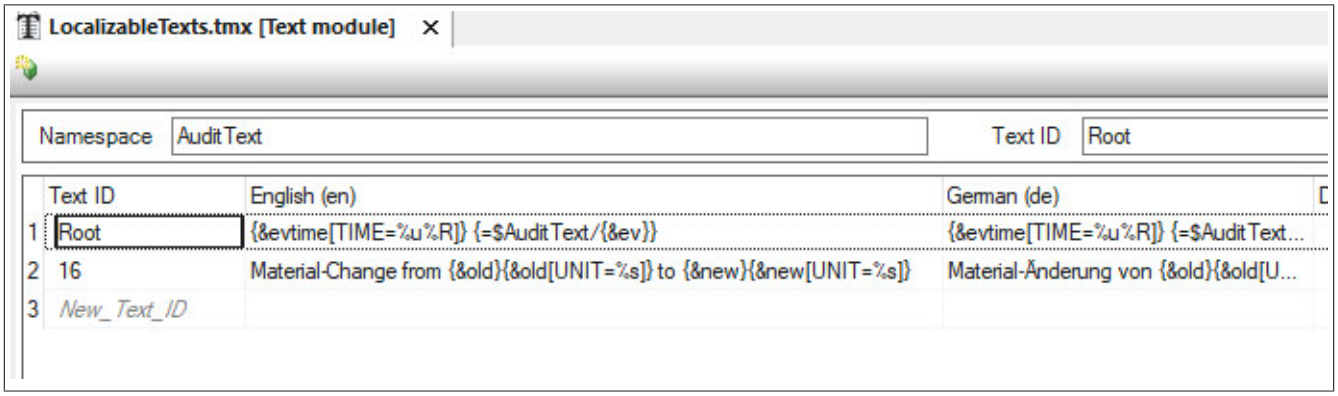
In der Textdatei muss der Namespace sowie die TextID definiert werden. Es wird, wie in der MpAudit Konfiguration definiert, der Namespace "AuditText" und die Text ID "Root" verwendet. Für mehr Information siehe Abschnitt Textsystem als Textquelle.

Sobald sich der Wert der "Material" Prozessvariable ändert, wird als Ereignistext der Text verwendet, welcher unter der TextID "Root" bzw. "16" angegeben ist.

Im Kürzel *old* wird der ursprüngliche Wert der Prozessvariable angezeigt. Im Kürzel *new* der neue Wert. Mithilfe von [UNIT=%s] wird die Einheit (z.B. mm) der Variable angezeigt.

Für mehr Information zu den Kürzeln eines Wertänderungs-Ereignis siehe hier.

Für mehr Information zur Angabe der Einheit siehe Einheiten von Variablen anzeigen.



mapp View Visualisierung

Um die Materialmenge über die Visualisierung zu ändern wird eine mapp View Visualisierung eingefügt. Für die Eingabe des Materials kann beispielsweise das NumericInput-Widget verwendet werden. Um das Messsystem über die Visualisierung zu ändern kann das MeasurementSystemSelector-Widget verwendet werden.

Wie eine mapp View Visualisierung erstellt wird, ist Schritt für Schritt im Getting Started von mapp View erklärt.

Verwendung der Funktionsbausteine

Nach der Konfiguration wird der Funktionsbaustein **MpAuditTrail** eingefügt. Dieser Funktionsbaustein implementiert den Haupt-Ereignis-Rekorder. Hier werden alle Ereignisse in einem Ringpuffer aufgezeichnet.

Sobald sich die Prozessvariable "Material" ändert wird die Wertänderung von **MpAuditTrail** aufgezeichnet. Durch das Kommando "Export = `TRUE`" wird die Ereignisliste am definierten Speichermedium exportiert. In der Ereignisliste wird die definierte Anzeige-Einheit angezeigt (Millimiter bzw. Inch) und im Programm wird die Engineering-Einheit (Kubikzentimeter) verwendet.

10.4 Diagnose

Als Diagnosemöglichkeit für diese Komponente stehen FAQs zur Verfügung.

10.4.1 FAQ

In diesem Abschnitt findet man mögliche Fragen und Hindernisse, welche beim Implementieren von mapp Com auftreten können.

Nach importieren einer neuen Konfiguration werden die neuen Parameter nicht verwendet

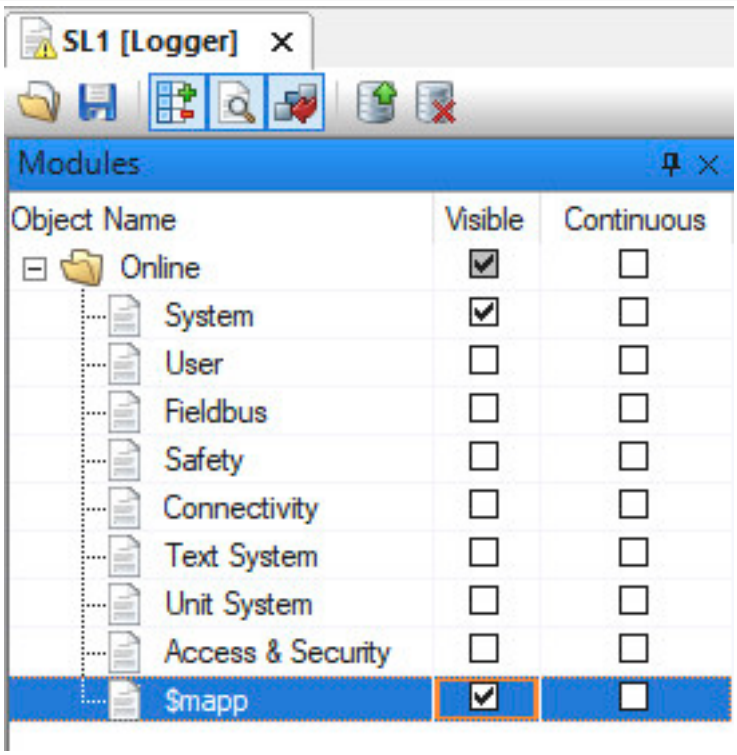
Standardmäßig werden nicht alle Parameter automatisch durch einen Import übernommen. Manche Komponenten müssen sich in einem bestimmten Zustand befinden, damit die neuen Parameter übernommen und verwendet werden können.

Beispiel mapp Data

In der Struktur MpDataRecorderConfigType wird über die Spalte "Konfigurationsupdate" erklärt, wann der Parameter in der Komponente übernommen wird. Was die Symbole in der Spalte genau bedeuten, wird im Abschnitt Konfiguration während der Laufzeit erklärt.

Meine Logger Events werden nicht über MpComLoggerUI angezeigt

Es gilt zu beachten, dass nur der Inhalt des "$mapp" Logbuchs angezeigt werden kann.



Die MpCom Bibliothek funktioniert nicht mit anderen mapp Paketen

Es gilt zu beachten, dass mapp Com nur mit mapp Services Komponenten verwendet werden kann. Komponenten von anderen mapp Paketen, wie beispielsweise mapp Motion, können nicht verwendet werden.

10.5 Bibliotheken

10.5.1 Funktionsbausteine

Unterstützt ab	Name	Beschreibung
5.0.0	MpComLoggerUI	Dieser Funktionsbaustein loggt Ereignisse mit, welche auch im mapp Logger angezeigt werden. Diese können dann über ein VC4 Visualisierung dargestellt werden
5.0.0	MpComConfigManager	Dieser Funktionsbaustein ermöglicht den Export und Import von mapp Konfigurationen
5.0.0	MpComDump	Dieser Funktionsbaustein erstellt eine Datei mit allen Informationen über verwendete Komponenten und Dienste
5.0.0	MpComGetLink	Dieser Funktionsbaustein liefert den MpLink einer Komponenten anhand deren Name zurück

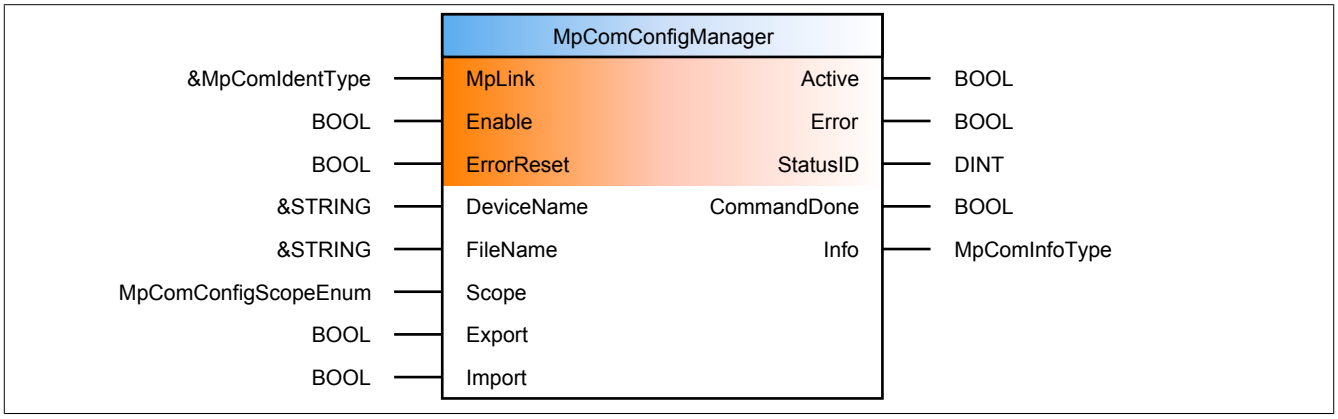
Unterstützt ab	Name	Beschreibung
5.0.0	MpComLinkToParent	Dieser Funktionsbaustein wird benützt, um eine Hierarchie unter den Komponenten zu Erstellen
5.0.0	MpComLink	Diese Funktion wird benützt, um eine Hierarchie unter den Komponenten zu Erstellen
5.0.0	MpComHandle	Diese Funktion wird intern verwendet
5.0.0	MpComVersion	Diese Funktion wird intern verwendet

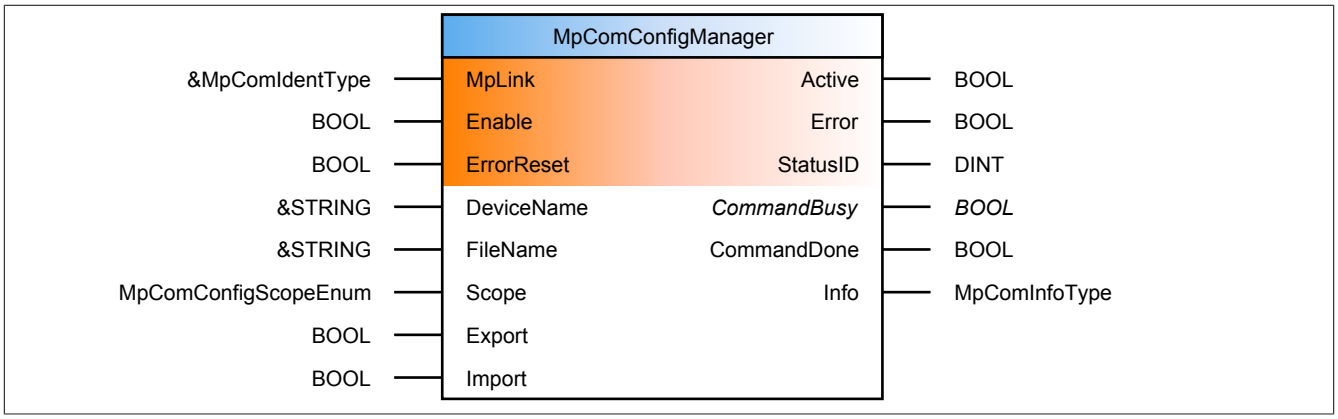
10.5.1.1 MpComConfigManager

Dieser Funktionsbaustein ermöglicht den Export und Import von mapp Konfigurationen.

Funktionsbaustein

Optionale Parameter





Schnittstelle

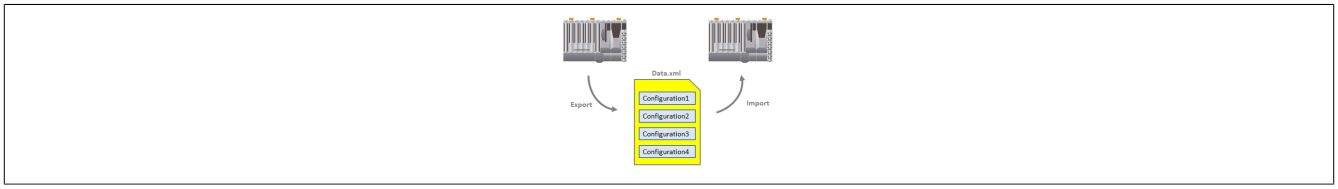
I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Definiert für welche mapp Komponente, bzw. welche Maschinengruppe die Konfigurationen im-/exportiert werden sollen Bleibt offen, falls alle Konfigurationen im-/exportiert werden sollen.
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	DeviceName	`Zeiger auf STRING[50]`	File Device (Speichermedium), auf welches die Konfigurationen exportiert werden, bzw. von welchem die Konfigurationen importiert werden.
IN	FileName	`Zeiger auf STRING[255]`	Name der Datei, welche die Konfiguration(en) enthält.
IN	Scope	`MpComConfigScopeEnum`	Definiert ob die Konfiguration einer einzelnen mapp Komponente, oder einer Maschinengruppe im-/exportiert werden sollen.
IN	Export	`BOOL`	Befehl zum Exportieren der Konfigurationen.
IN	Import	`BOOL`	Befehl zum Importieren der Konfigurationen.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.
OUT	Info	`MpComInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

10.5.1.1.1 Beschreibung

Über diesen Funktionsbaustein können mapp Konfigurationen von einer Maschine exportiert werden und in einer anderen Maschine wiederum importiert werden. Dabei können einzelne Konfigurationen, die Konfiguration von einer Maschinengruppe (Bilden von modularen Applikationen), oder alle Konfigurationen einer kompletten Maschine ausgewählt werden.



Auswahl der Konfiguration(en)

Die Auswahl der Konfiguration erfolgt über die Eingänge "Scope" und "MpLink".

Alle Konfigurationen

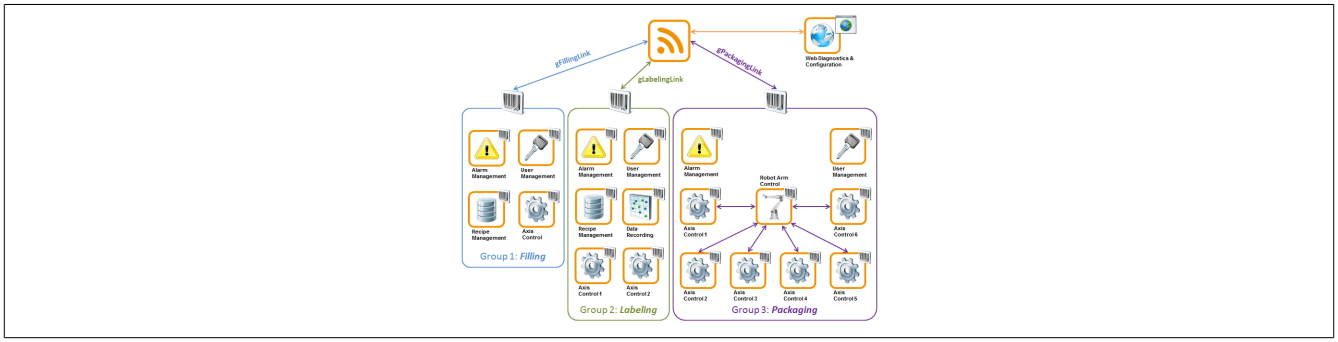
Wird der Eingang "MpLink" **nicht** verbunden, so werden alle Konfigurationen ausgewählt. Der Eingang "Scope" wird in diesem Fall nicht beachtet.

Eine einzelne Konfiguration

Die mapp Komponente von welcher die Konfiguration exportiert werden soll, wird über den "MpLink" am Eingang angegeben (z.B. die Konfiguration eines Förderbandes mit *gConveyorLink* als MpLink). Außerdem wird am Eingang "Scope" das Element *mpCOM_CONFIG_SCOPE_COMPONENT* angegeben.

Eine Maschinengruppe

Eine Maschinengruppe kann ein Teil einer Maschine sein, oder auch ein Roboter mit allen Achsen die er steuert. Wie man eine Maschinengruppe (Hierarchie) aufbaut ist im Abschnitt Bilden von modularen Applikationen beschrieben. Um alle Konfigurationen einer Maschinengruppe auszuwählen, muss der "MpLink" der Gruppe ausgewählt werden.



Möchte man in diesem Beispiel die Konfigurationen der *Filling* - Gruppe auswählen, so wird am Eingang "MpLink" *gFillingLink* verbunden. Soll der Roboter inklusive aller Einzelachsen im- oder exportiert werden, so muss der MpLink des Roboters ausgewählt werden. Außerdem muss über den Eingang "Scope" der richtige Bereich eingestellt werden. Das bedeutet, für einen Gruppe muss *mpCOM_CONFIG_SCOPE_BRANCH* ausgewählt werden.

Export

Über den Befehl "Export" können die zuvor ausgewählten Konfigurationen exportiert werden. Die Konfigurationen werden aus der Automation Runtime exportiert und auf dem angegebenen Speichermedium ("DeviceName"), unter dem angegebenen Dateinamen ("FileName") abgelegt. Dabei werden alle Konfigurationen in eine Datei gespeichert. In dieser Datei kann man die Konfiguration einer mapp Komponente über den Namen des MpLinks identifizieren. Die Daten werden im XML Format abgelegt.

Import

Über den Befehl "Import" kann eine Datei mit Konfigurationen importiert werden. Welche Datei wird über das angegebene Speichermedium ("DeviceName") und den angegebenen Dateinamen ("FileName") definiert. Es können auch mehrere Dateien zur selben Zeit importiert werden:

- ***.*** : Wird mit '*.*' gearbeitet, so werden alle Dateien von dem angegebenen Speichermedium importiert
- ***.xxx** : Wird mit '*.xxx' gearbeitet, so werden alle Dateien mit der definierten Endung importiert (z.B. *.xml, *.csv, *.mcg, ...)

Welche Konfiguration(en) aus der Datei genau importiert werden, hängt von der Auswahl ab. Sollen mehrere Konfigurationen importiert werden, so muss der MpLink einer Konfiguration aus der Importdatei mit dem MpLink aus der Maschine übereinstimmen. Nur dann wird die Konfiguration aus der Datei in die Automation Runtime übernommen.

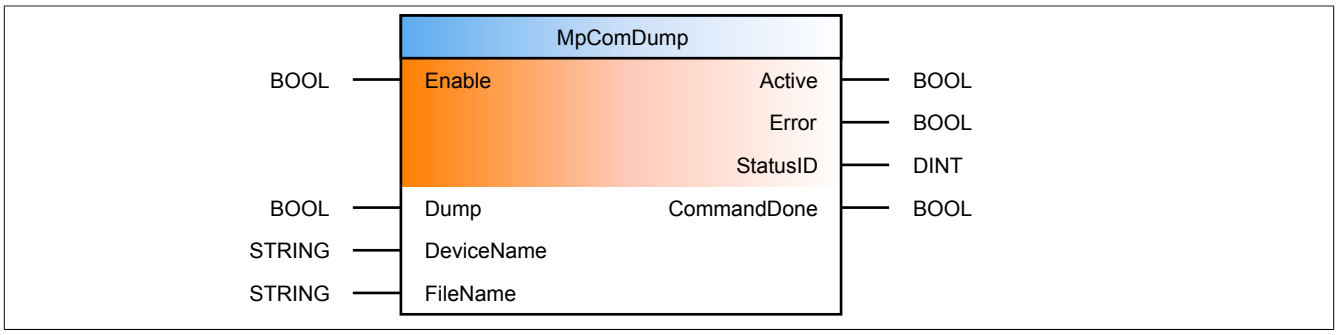
> **Es ist möglich Konfigurationsdateien der mapp Version 0.05.x über diesen Funktionsbaustein zu importieren. Die Konfigurationsdatei kann direkt über den Namen, oder über die Endung .mcg ausgewählt werden.**

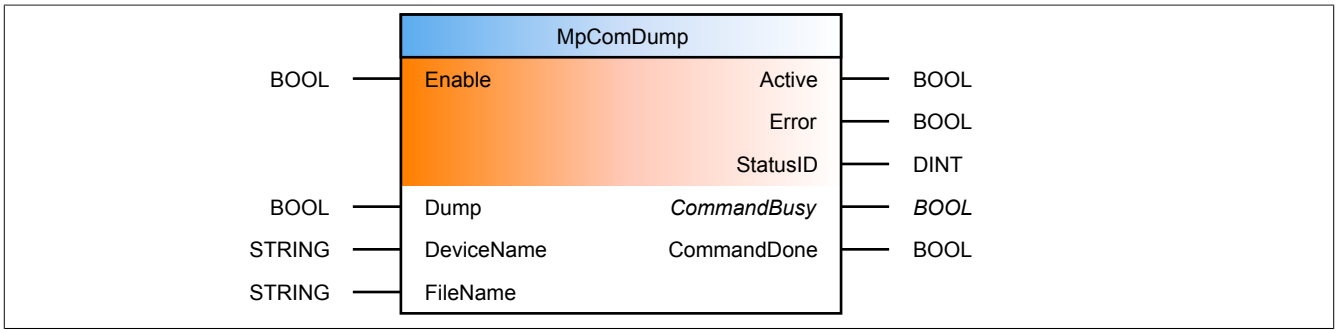
10.5.1.2 MpComDump

Dieser Funktionsbaustein erstellt eine Datei mit allen Informationen über verwendete Komponenten und Dienste.

Funktionsbaustein

Optionale Parameter





Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	DeviceName	`STRING[20]`	Das File Device auf dem die Datei mit den Systeminformationen abgelegt werden soll.
IN	FileName	`STRING[40]`	Der Name den die Datei mit den Systeminformationen haben soll.
IN	Dump	`BOOL`	Erstellt die Datei mit den mapp Informationen.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	*CommandBusy*	`BOOL`	Funktionsbaustein führt aktuell einen Befehl aus.
OUT	CommandDone	`BOOL`	Abarbeitung erfolgreich. FB ist fertig.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

10.5.1.2.1 Beschreibung

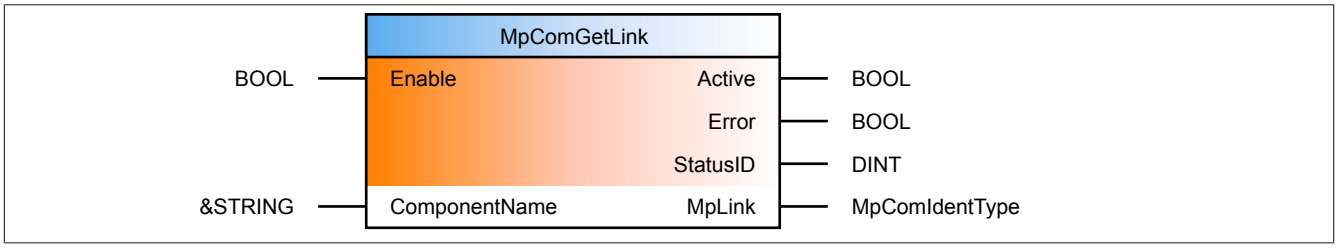
Dieser Funktionsbaustein wird verwendet, um an zusätzliche Diagnoseinformationen über mapp zu kommen. Mithilfe dieser Informationen kann eine weitergehende Analyse erfolgen.

Bei einer positiven Flanke am "Dump" Eingang wird eine XML - Datei mit allen Diensten und mapp Komponenten, sowie deren Version erzeugt. Der Name der erzeugten Datei hängt vom Eingang "FileName" ab. Das Speichermedium (File Device) wird am Eingang "DevName" angegeben". An diesem Ort wird die Datei nach erfolgreichem Erzeugen abgelegt.

10.5.1.3 MpComGetLink

Dieser Funktionsbaustein liefert den MpLink einer Komponenten anhand deren Name zurück.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ComponentName	`Zeiger auf STRING[100]`	Name der mapp Komponente (zu finden ganz oben in der Konfiguration).
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	StatusID	`DINT`	Statusinformation.
OUT	MpLink	`MpComIdentType`	MpLink der am Eingang definierten mapp Komponente.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

10.5.1.3.1 Beschreibung

Normal wird für jede mapp Komponente, welche in der Configuration View eingefügt ist automatisch eine globale Variable erstellt - der mapp Link. Wird dieser MpLink nirgends in der Applikation verwendet, so wird zur Laufzeit, wie bei jeder anderen Variable, kein Speicher dafür reserviert. Normalerweise jedoch wird der MpLink an den unterschiedlichen Funktionen und Funktionsbausteinen angegeben, um die jeweilige mapp Komponente zu identifizieren.

Möchte man keine globalen Variablen verwenden, so kann dies über den Funktionsbaustein MpComGetLink erreicht werden. Dieser Funktionsbaustein liefert anhand des Namens einer mapp Komponente deren MpLink zurück. Dabei sind folgende Schritte zu beachten:

Automatisch generierten MpLink nicht verwenden

Die automatisch generierten globalen Variablen sollten nicht am als Parameter für eine Funktion oder am Eingang eines Funktionsbausteins verwendet werden. Dadurch wird für den automatisch generierten globalen MpLink keine Speicher reserviert, er existiert nicht.

Verwendung von MpComGetLink

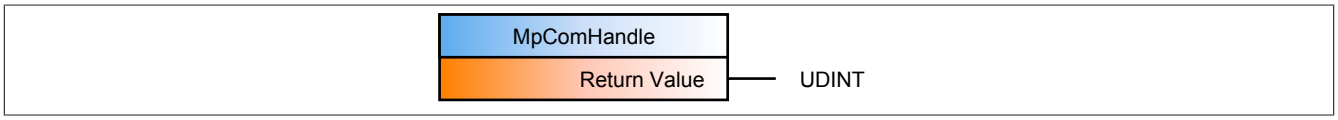
Stattdessen liefert dieser Funktionsbaustein den MpLink am Ausgang zurück. Anzugeben ist der Name der jeweiligen Komponente am Eingang "ComponentName". Den Namen der Komponenten kann man in der Konfiguration der Komponenten ganz oben finden.

Dieser kann dann direkt (Adresse des Funktionsbausteins-Ausgang "MpLink"), oder indirekt (lokale Variable mit Datentyp MpComIdentType mit Ausgang verbinden) für andere mapp Komponenten verwendet werden.

10.5.1.4 MpComHandle

Diese Funktion wird intern verwendet.

Funktion



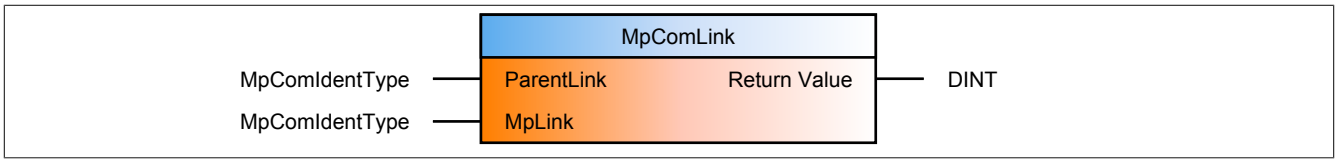
Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
OUT	ReturnValue	`UDINT`	Für interne Verwendung.

10.5.1.5 MpComLink

Diese Funktion wird benützt, um eine Hierarchie unter den Komponenten zu erstellen. Die Funktionsweise ist identisch mit der von MpComLinkToParent. Der Unterschied liegt darin, dass MpComLinkToParent ein Funktionsbaustein ist und MpComLink eine Funktion.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	ParentLink	`MpComIdentType`	Verbindung zur Parent Komponente.
IN	MpLink	`MpComIdentType`	Verbindung zur mapp.
OUT	ReturnValue	`DINT`	Status Information der Funktion.

mapp Konzept

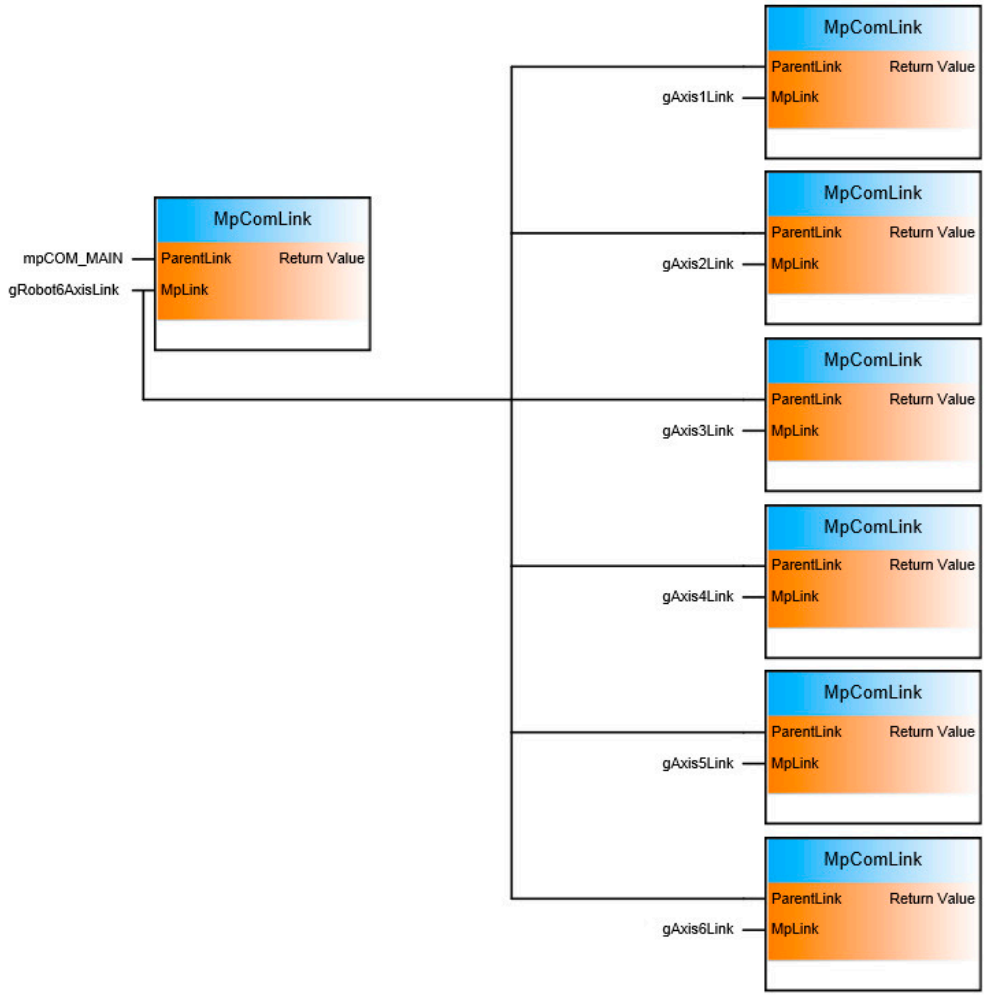
Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

10.5.1.5.1 Beschreibung

Mit dieser Funktion kann eine Komponentenhierarchie erstellt werden. Diese wird zum Beispiel verwendet, um eine Fertigungsanlage in verschieden Gruppen zu unterteilen. Siehe Bilden von modularen Applikationen

Am Eingang "MpLink" wird der MpLink der Komponente angegeben, die der hierarchisch höheren Gruppe angeordnet werden soll. Diese Gruppe wird über den Eingang "ParentLink" definiert. Dies kann zum Beispiel eine Maschinengruppe sein, oder auch eine normale Komponente (z.B. MpAxis wird mit der Komponente MpRoboArm6Axis verbunden). Der Rückgabewert wird dann als ParentLink für die untergeordnete Komponente verwendet.

Beispiel: 6 Achs Roboter



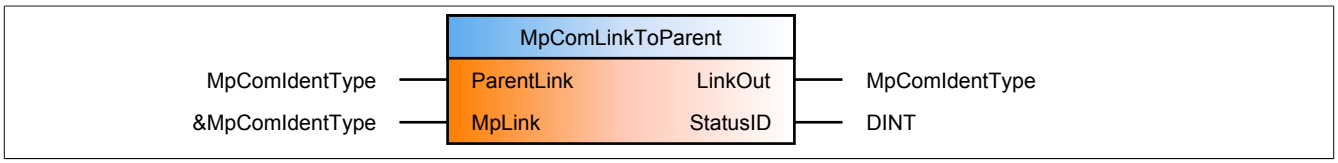
An der ersten MpComLink Funktion wird an den "ParentLink" die Konstante mpCOM_MAIN übergeben. Mit dieser Konstante wird eine Komponente in der MpCom registriert. In diesem Beispiel wird die Roboterkomponente in der MpCom registriert. Die einzelnen Achsen sind der Roboterkomponente hierarchisch untergeordnet und müssen dadurch mit dieser verbunden werden. Dies geschieht wieder mit der MpComLink Funktion.

> **Das Erstellen der Hierarchie muss im INIT Programmteil ausgeführt werden und dabei müssen alle Komponenten, die miteingebunden sind, deaktiviert sein ("Enable" = `FALSE`). Die Hierarchie kann während der Laufzeit nicht mehr geändert werden. Nach einem Neustart der Steuerung wird die Hierarchie wieder neu gebildet.**

10.5.1.6 MpComLinkToParent

Dieser Funktionsbaustein wird benützt, um eine Hierarchie unter den Komponenten zu erstellen. Die Funktionsweise ist identisch mit der von MpComLink. Der Unterschied liegt darin, dass MpComLinkToParent ein Funktionsbaustein ist und MpComLink eine Funktion.

Funktionsbaustein



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
IN	ParentLink	`MpComIdentType`	Verbindung zur Parent Komponente.
IN	MpLink	`Zeiger auf MpComIdentType`	Verbindung zur mapp.
OUT	LinkOut	`MpComIdentType`	Verbindung zur Child Komponente.
OUT	StatusID	`DINT`	Statusinformation.

mapp Konzept

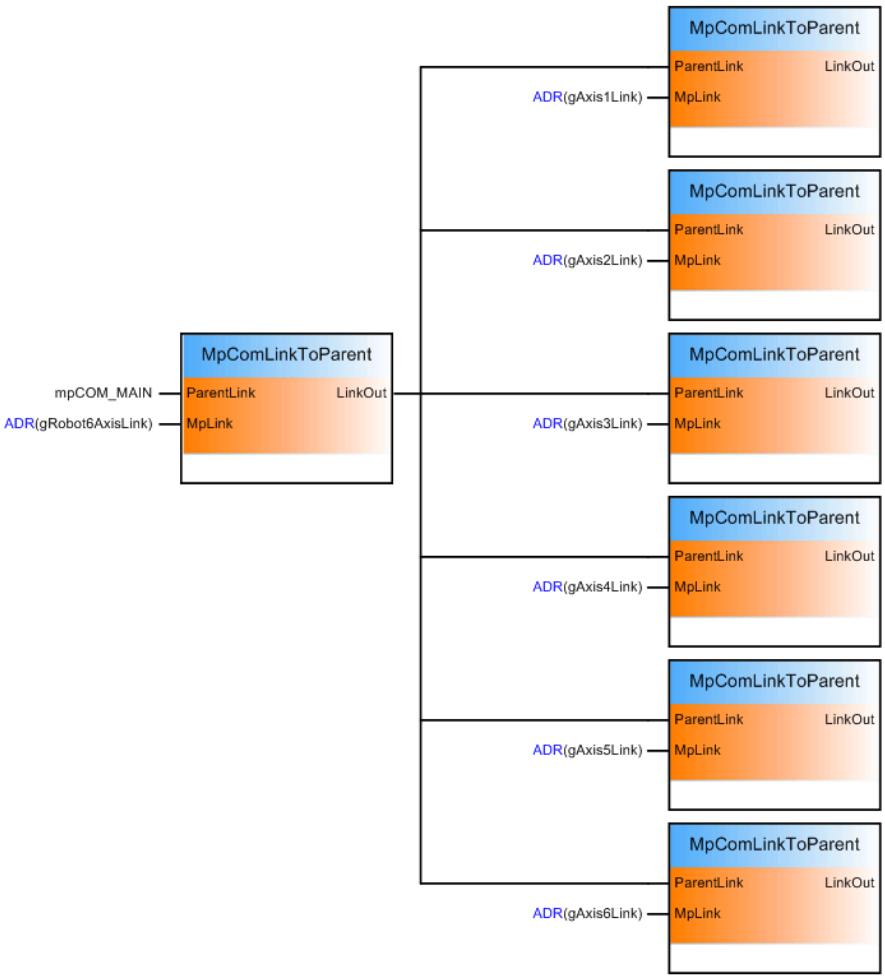
Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

10.5.1.6.1 Beschreibung

Mit diesem Funktionsbaustein kann eine Komponentenhierarchie erstellt werden. Diese wird zum Beispiel verwendet, um eine Fertigungsanlage in verschieden Gruppen zu unterteilen. Siehe Bilden von modularen Applikationen

Am Eingang "MpLink" wird der MpLink der Komponente angegeben, die der hierarchisch höheren Gruppe angeordnet werden soll. Diese Gruppe wird über den Eingang "ParentLink" definiert. Dies kann zum Beispiel eine Maschinengruppe sein, oder auch eine normale Komponente (z.B. MpAxis wird mit der Komponente MpRoboArm6Axis verbunden). Der Ausgang "LinkOut" wird dann als ParentLink für die untergeordnete Komponente verwendet.

Beispiel: 6 Achs Roboter



An den ersten MpComLinkToParent Funktionsbaustein wird an den "ParentLink" die Konstante mpCOM_MAIN übergeben. Mit dieser Konstante wird eine Komponente in der MpCom registriert. In diesem Beispiel wird die Roboterkomponente in der MpCom registriert. Die einzelnen Achsen sind der Roboterkomponente hierarchisch untergeordnet und müssen dadurch mit dieser verbunden werden. Dies geschieht wieder mit dem MpComLinkToParent Funktionsbausteins.

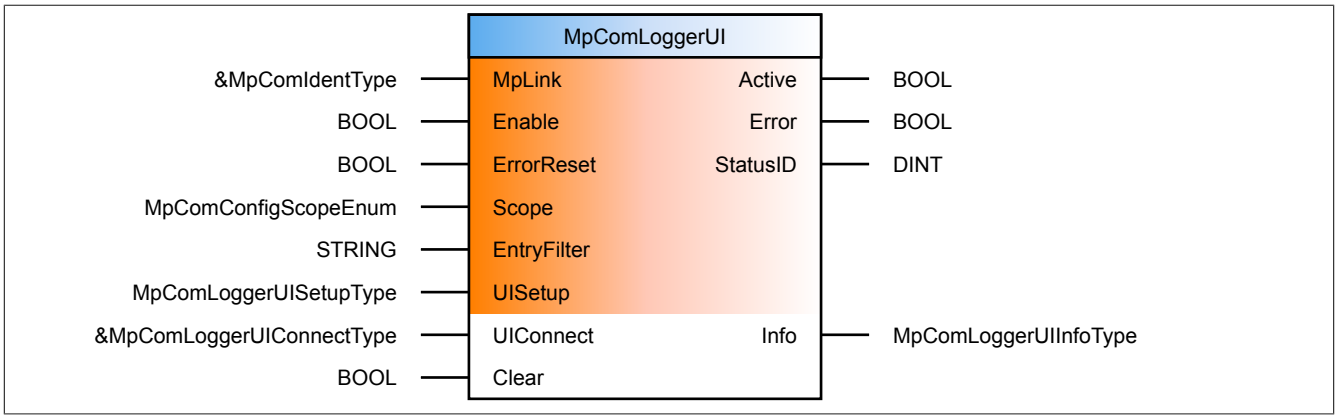
> **Das Erstellen der Hierarchie muss im INIT Programmteil ausgeführt werden und dabei müssen alle Komponenten, die miteingebunden sind, deaktiviert sein ("Enable" = `FALSE`). Die Hierarchie kann während der Laufzeit nicht mehr geändert werden. Nach einem Neustart der Steuerung wird die Hierarchie wieder neu gebildet.**

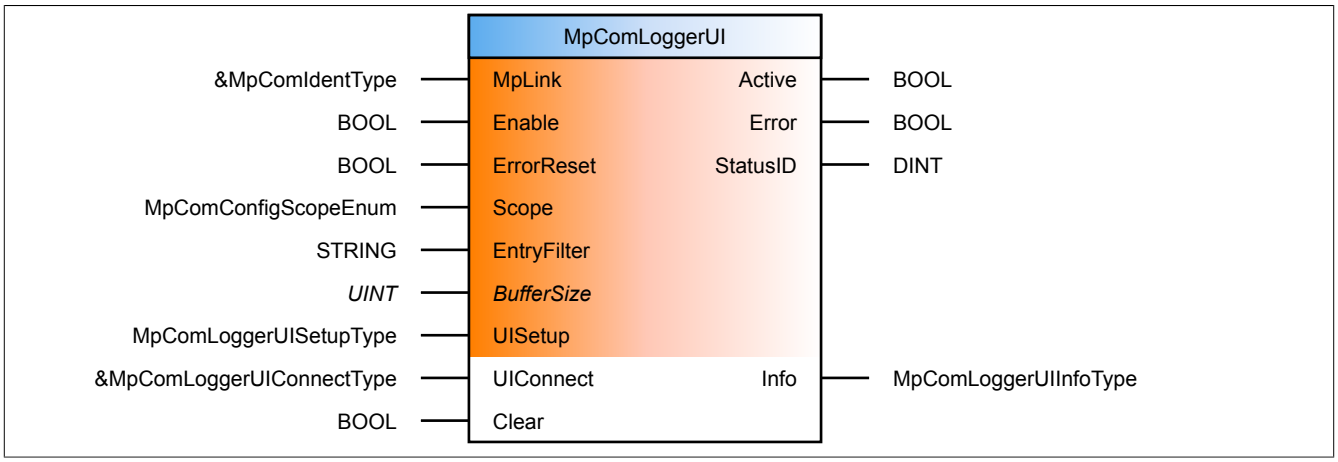
10.5.1.7 MpComLoggerUI

Dieser Funktionsbaustein loggt Ereignisse mit, welche auch im mapp Services Logger angezeigt werden. Diese können dann über ein VC4 Visualisierung dargestellt werden.

Funktionsbaustein

Optionale Parameter





Schnittstelle

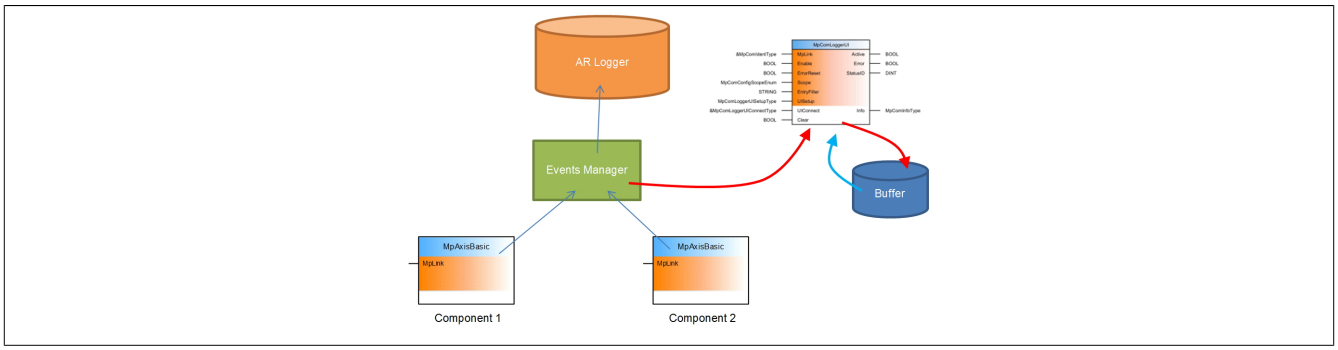
I/O	Parameter	Datentyp	Beschreibung
IN	MpLink	`Zeiger auf MpComIdentType`	Definiert für welche mapp Komponente, bzw. welche Maschinengruppe die Ereignisse geloggt werden sollen
IN	Enable	`BOOL`	FB ist aktiv solange der Eingang gesetzt ist.
IN	ErrorReset	`BOOL`	Dient zum Zurücksetzen von Funktionsbaustein Fehlern.
IN	Scope	`MpComConfigScopeEnum`	Definiert ob die Ereignisse einer einzelnen mapp Komponente, oder einer Maschinengruppe geloggt werden sollen.
IN	EntryFilter	`STRING[255]`	Filtert generell, ob ein Ereignis aufgenommen wird, oder ignoriert wird.
IN	*BufferSize*	`UINT`	Legt fest, wieviele Ereignisse im Buffer verwaltet werden. Ist der Buffer voll, so wird bei einem neuen Ereignis das älteste überschrieben.
IN	UISetup	`MpComLoggerUISetupType`	Dient zur Konfiguration der Elemente, welche zur Visualisierung verbunden werden.
IN	UIConnect	`Zeiger auf MpComLoggerUIConnectType`	Diese Struktur beinhaltet Parameter, die für die Verbindung zur Visualisierung gedacht sind.
IN	Clear	`BOOL`	Leert den Ereignisspeicher.
OUT	Active	`BOOL`	FB ist aktiv.
OUT	Error	`BOOL`	Fehler bei Abarbeitung aufgetreten.
OUT	StatusID	`DINT`	Statusinformation.
OUT	Info	`MpComLoggerUIInfoType`	Zusätzliche Funktionsbausteins-Information.

mapp Konzept

Im Abschnitt mapp Komponenten wird erklärt wie die mapp Komponenten aufgebaut sind. Außerdem wird auf die richtige Verwendung von mapp Komponenten hingewiesen (z.B. beim Download).

10.5.1.7.1 Beschreibung

Über diesen Funktionsbaustein werden alle Ereignisse, welche auch im mapp Services Logger eingetragen werden in einen Buffer gespeichert und für die VC4 Visualisierung aufbereitet.



Alle mapp Komponenten leiten unterschiedliche Ereignisse an den Logger weiter. Unter anderem werden Fehler an den Logger weitergeleitet. Diese Ereignisse werden von diesem Funktionsbaustein in einen internen Buffer gespeichert, um sie dann für die VC4 Visualisierung aufzubereiten.

Auswahl der Ereignisse

Die Auswhal der Ereignisse, welche geloggt werden sollen erfolgt über die Eingänge "Scope", "MpLink" und "EntryFilter".

Alle Ereignisse

Wird der Eingang "MpLink" nicht verbunden, so werden alle Ereignisse ausgewählt. Der Eingang "Scope" wird in diesem Fall nicht beachtet.

Ereignisse einer einzelnen mapp Komponente

Die mapp Komponente von welcher die Ereignisse geloggt werden soll, wird über den "MpLink" am Eingang angegeben (z.B. die Ereignisse eines Förderbandes mit gConveyorLink als MpLink). Außerdem wird am Eingang "Scope" das Element mpCOM_CONFIG_SCOPE_COMPONENT angegeben.

Ereignisse einer Maschinengruppe

Eine Maschinengruppe kann ein Teil einer Maschine sein, oder auch ein Roboter mit allen Achsen die er steuert. Wie man eine Maschinengruppe (Hierarchie) aufbaut ist im Abschnitt Bilden von modularen Applikationen beschrieben. Um alle Ereignisse einer Maschinengruppe auszuwählen, muss der "MpLink" der Gruppe ausgewählt werden.

Filtern der Ereignisse über "EntryFilter"

Weiters kann über den Eingang "EntryFilter" nocheinmal gefiltert werden welche Ereignisse in den Buffer kommen sollen und welche nicht. Ereignisse welche über diesen Filter ausgefiltert werden, werden verworfen. Dieser Filter ermöglicht das Filtern nach mapp Komponenten, Level, Bereich oder z.B. auch Fehlernummern. Das Eingabeformat des Filters sieht folgendermaßen aus:

- FilterTyp=FilterBereich
- FilterTyp1=FilterBereich1,FilterTyp2=FilterBereich2
- FilterTyp1=FilterBereich1,FilterTyp2=FilterBereich2-FilterBereich3,FilterType4=FilterBereich4

Wie gezeigt können auch mehrere Filter hintereinander definiert werden. Folgende Tabelle zeigt welche Filter-Typen und Filter-Bereiche es gibt:

Typ des Filters		Bereich des Filters	Beschreibung
StatusID	=	• -1064074747 • -1064239102--1064238844 • XXXXXXXXXX	Es kann nach einer speziellen StatusID oder nach eine
ErrorNumber	=	• 4007 • 29000-29999 • XXXXX	Es kann nach einer speziellen Fehlernummer oder nac
Component	=	• gAxisConveyorLink • gPackMLLink • gXxxxxLink	Hier wird ein spezieller MpLink angegeben. Alle Ereigni angezeigt.
Facility	=	• ARCORE • MAPP_INDUSTRY • SAFETY1 • XXXXXXX (siehe MpComFacilitiesEnum , letzter Teil)	Über diesen Typ kann der Bereich gefiltert werden.
Severity	=	• ERROR • WARNING • INFORMATIONAL • SUCCESS (siehe MpComSeveritiesEnum , letzter Teil)	Über diesen Typ kann das Level gefiltert werden.

Die Filter Typen können auch beliebig kombiniert werden:

- Component=gAxisConveyorLink,Severity=ERROR
- Component=gAxisConveyorLink,ErrorNumber=4007
- Component=gAxisConveyorLink,Component=gAxisCutterLink,Severity=ERROR

Buffergröße

Grundsätzlich werden bis zu 100 Einträge im Buffer des Funktionsbausteins abgelegt. Ist eine andere Anzahl notwendig kann über den optionalen Eingang "BufferSize" definiert werden, für wieviele Einträge im Buffer Platz sein soll. Über eine positive Flanke am Eingang "Clear" kann der Buffer geleert werden. Dabei werden alle geloggten Einträge gelöscht. Der Befehl "Clear" kann in der Appliaktion z.B. in Kombination mit dem Befehl "ErrorReset" einer Achse genützt werden um die letzten Fehler der Achse zu quittieren und gleichzeitig den Buffer zu leeren, über welchen die Fehler auf der Visualisierung angezeigt wurden.

MpComLoggerUIConnectType

Die UIConnect Struktur unterteilt sich in folgende Themen:

- **List:** Zeigt die Ereignisse mit allen dazugehörigen Details an
- **Clear:** Löscht alle Ereignisse aus dem Buffer.
- **SearchFilter:** Ein zusätzlicher Filtermechanismus per String, welcher über die Visualisierung verwendet werden kann. Filtert die Ereignisse nach der eingegebenen Komponente. Wird der Filter gelöscht, werden wieder alle Ereignisse angezeigt

Der "Clear" Befehl in der UIConnect Struktur gleicht dem "Clear" Befehl am Eingang des Funktionsbausteins. Nur ist dieser dafür gedacht über die Visualisierung verwendet zu werden, nicht in der Applikation. Der "SearchFilter" löscht keine Ereignisse aus dem Buffer. Er filtert sie nur temporär, bis der Filterstring wieder zurückgesetzt wird. Über den "SearchFilter" kann eine Komponente über den MpLink ausgewählt werden:

- gAxis01Link
- gConveyorLink
- gAxis*

Über den * können auch mehrere Komponenten gefiltert werden. Falls spezielle Kombinationen nötig sind, können dieselben Filtertypen und -bereiche, wie beim "EntryFilter" verwendet werden.

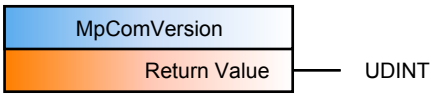
MpComLoggerUISetupType

In dieser Struktur wird über "LoggerListSize" definiert, wieviele Ereignisse auf einer Seite der Visualisierung angezeigt werden. Über "LoggerListScrollWindow" wird entschieden wieviele Ereignisse beim auf- oder abscrollen im Vorhinein angezeigt werden.

10.5.1.8 MpComVersion

Diese Funktion wird intern verwendet.

Funktion



Schnittstelle

I/O	Parameter	Datentyp	Beschreibung
OUT	ReturnValue	`UDINT`	Für interne Verwendung.

10.5.2 Datentypen und Enumeratoren

10.5.2.1 Datentypen

10.5.2.1.1 MpComDiagType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
StatusID	`MpComStatusIDType`	StatusID Diagnose Struktur

10.5.2.1.2 MpComIdentType

Eine Variable mit diesem Datentyp wird benötigt, um eine Komponente mit mapp zu verbinden.

Parameter	Datentyp	Beschreibung
Internal	`ARRAY[0.. 1] von UDINT`	Für interne Verwendung. Manipulation dieses Wertes kann zu Fehlern führen

10.5.2.1.3 MpComInfoType

Mit diesem Datentyp wird für die MpCom Komponente zusätzliche Information bereitgestellt.

Parameter	Datentyp	Beschreibung
Diag	`MpComDiagType`	Eine Diagnosestruktur für die StatusID

10.5.2.1.4 MpComInternalIDType

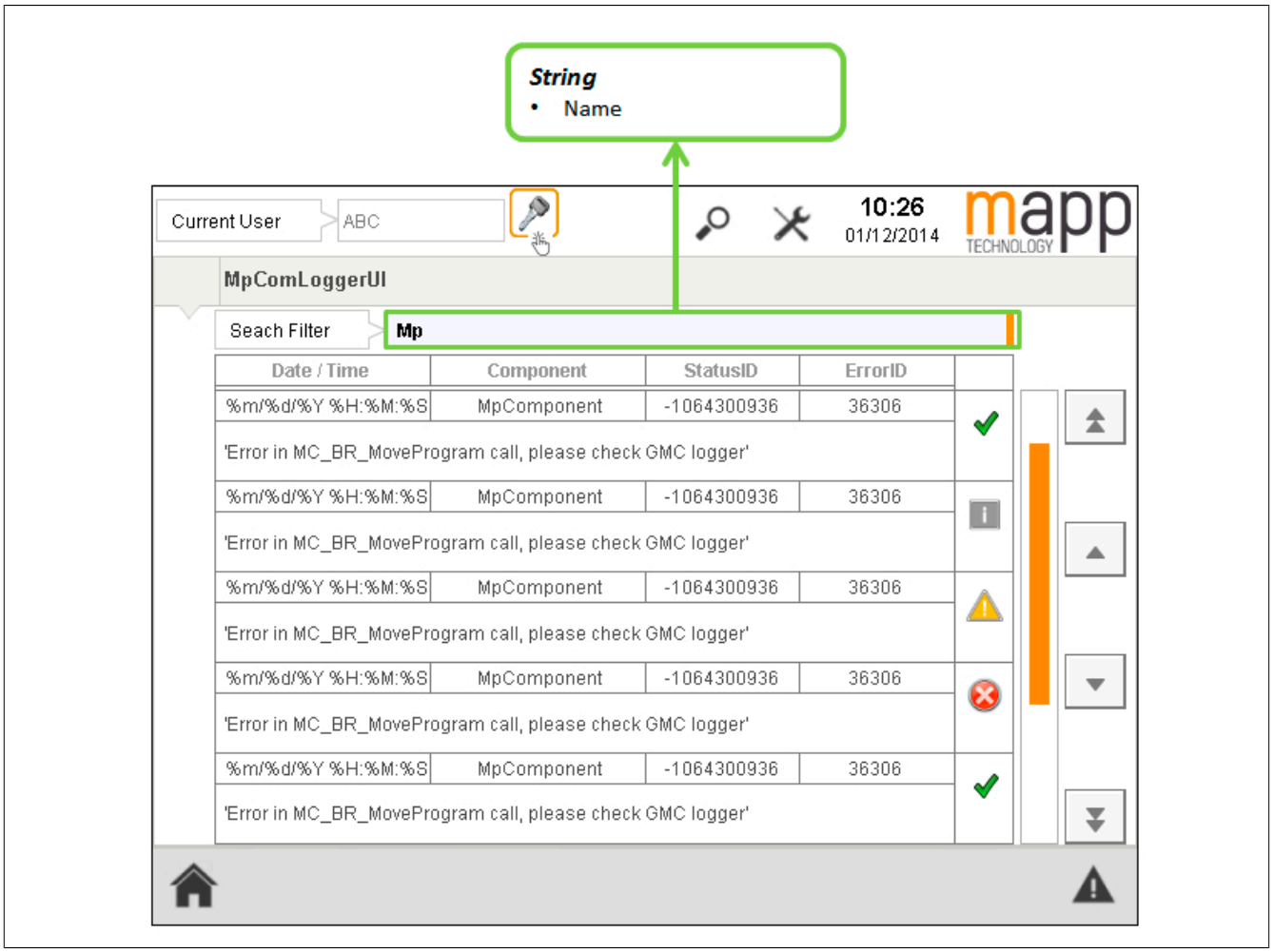
Dieser Datentyp schlüsselt den komponenteninternen Fehler in seine Einzelteile auf und gibt genauere Auskunft über Schweregrad, Ursprung und genaue Fehlernummer.

Parameter	Datentyp	Beschreibung
ID	`DINT`	Status ID
Severity	`MpComSeveritiesEnum`	Status Schweregrad - Wie schwerwiegend ist die erhaltene Information (Warnung, Fehler, ...)
Facility	`MpComFacilitiesEnum`	Status Ursprung (z.B. Infrastruktur, Mechatronik, Industrie, ...)
Code	`UINT`	Dekodierte Fehlernummer

10.5.2.1.5 MpComLoggerUIConnectType

Dieser Datentyp enthält alle notwendigen Informationen, um eine Verbindung zur Visualisierung herzustellen.

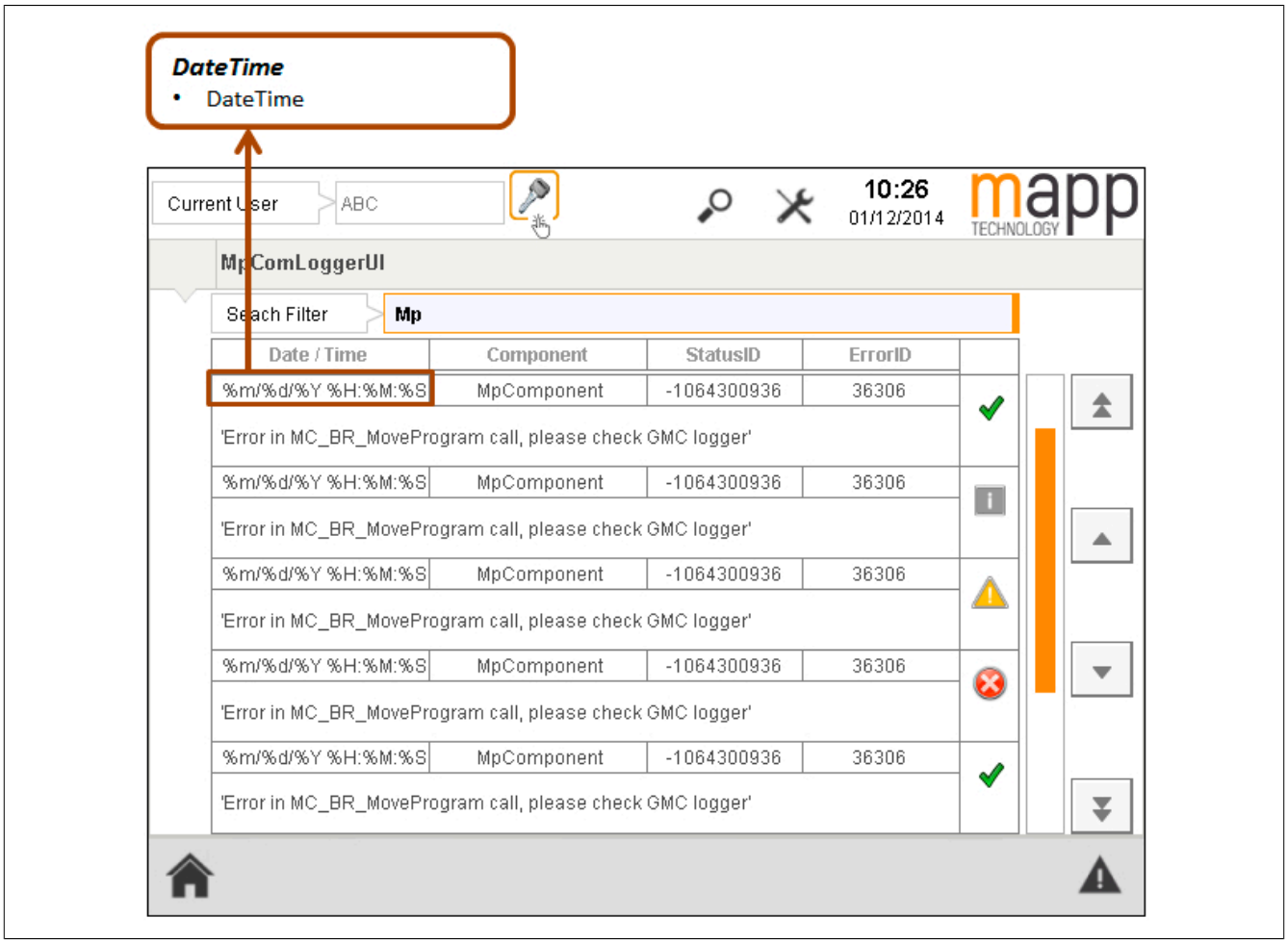
Parameter	Datentyp	Beschreibung
LoggerList	`MpComLoggerUILoggerListType`	Zeigt die Ereignisse an, welche auch im Logger eingetragen werden
Clear	`BOOL`	Leert den Buffer mit den Ereignissen aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
SearchFilter	`STRING[255]`	Filter, mit dem die Ereignisse aussortiert werden können **VC4 Anbindung:** Datapoint von *String*



10.5.2.1.6 MpComLoggerUIDateTimeType

Beschreibt die Zeit, zu welcher ein Ereignis stattgefunden hat.

Parameter	Datentyp	Beschreibung
DateTime	`DATE_AND_TIME`	Uhrzeit und Datum wann das Ereignis stattgefunden hat **VC4 Anbindung:** Datapoint von *DateTime*
Microseconds	`UDINT`	Genauer Zeitpunkt in Mikrosekunden wann das Ereignis stattgefunden hat **VC4 Anbindung:** Datapoint von *Numeric*



10.5.2.1.7 MpComLoggerUIInfoType

Mit diesem Datentyp wird für die MpComLoggerUI Komponente zusätzliche Information bereitgestellt.

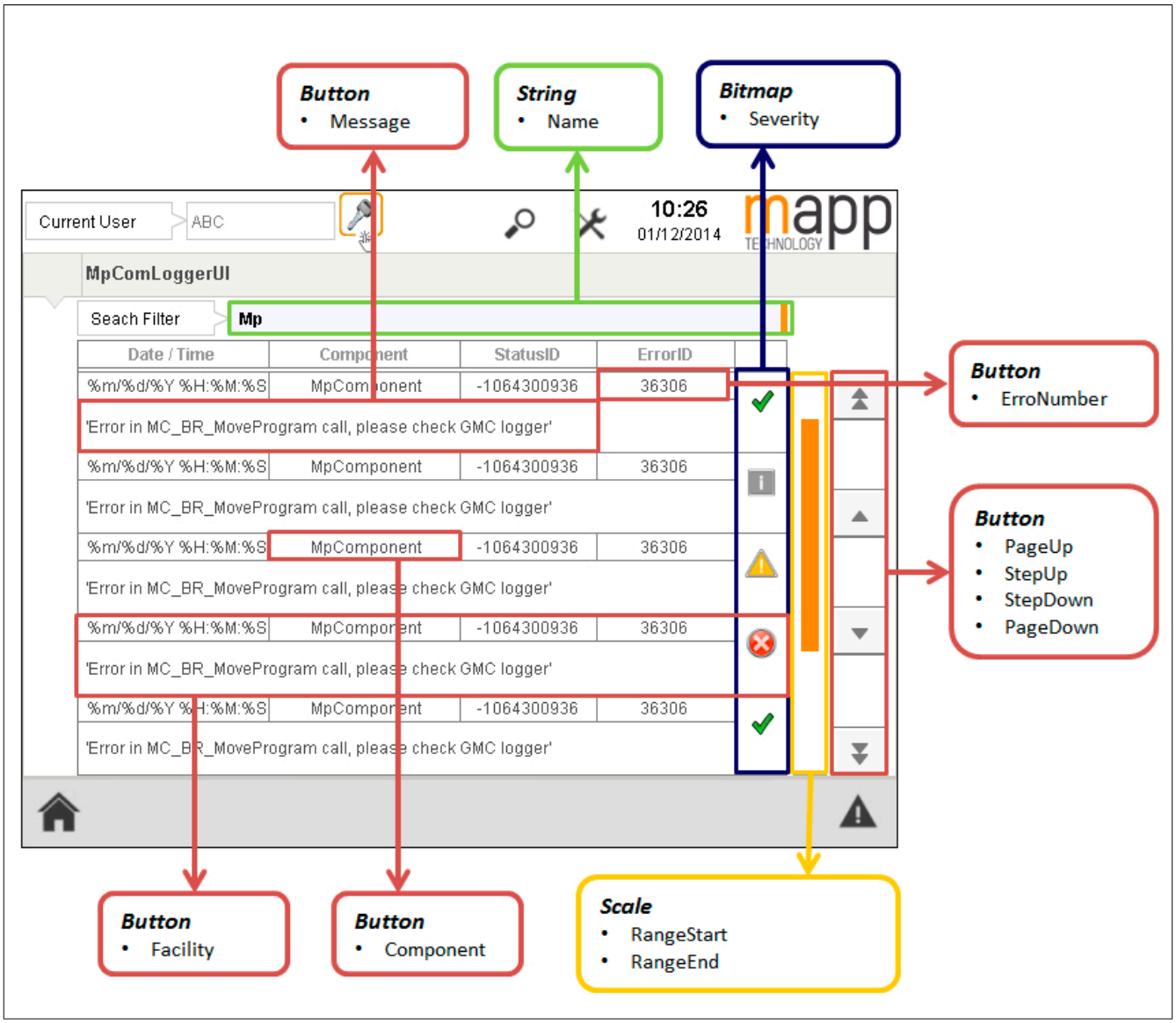
Parameter	Datentyp	Beschreibung
CurrentBufferEntries	`UINT`	Aktuelle Anzahl an Einträgen in Buffer. Es wird mit einem Ringbuffer gearbeitet. Ist dieser voll, werden die ältesten Einträge überschrieben.
Diag	`MpComDiagType`	Eine Diagnosestruktur für die StatusID

10.5.2.1.8 MpComLoggerUILoggerListType

Liste mit Ereignissen, welche auch im Logger eingetragen werden und die dazugehörige Navigation.

Parameter	Datentyp	Beschreibung
Message	`Array[0..19] von STRING[255]`	Liste aller Ereignisse als Text **VC4 Anbindung:** TextIndexOffset von *Button* (Textgruppe für Events) - siehe Info
ErrorNumber	`Array[0..19] von UINT`	Liste der Fehlernummern aller Ereignisse **VC4 Anbindung:** TextIndexOffset von *Button* (Textgruppe für Events) - siehe Info
Time	`Array[0..19] von MpComLoggerUIDateTimeType`	Liste der Zeitpunkte, an denen die Ereignisse stattgefunden haben
Component	`Array[0..19] von STRING[50]`	Liste mit mapp Komponenten, welche die Ursache eines Ereignisses sind **VC4 Anbindung:** TextIndexOffset von *Button* (Textgruppe für Events) - siehe Info
Facility	`Array[0..19] von MpComFacilitiesEnum`	Liste mit dem Bereich aus dem die Ereignisse kommen **VC4 Anbindung:** BitmapIndexDatapoint von *Button* (BitmapGroup für Bereiche) - siehe Dateiformat anzeigen
Severity	`Array[0..19] von MpComSeveritiesEnum`	Liste mit Levels, welchen die Ereignisse entsprechen **VC4 Anbindung:** BitmapIndexDatapoint von *Button* (BitmapGroup für Levels) - siehe Dateiformat anzeigen

Parameter	Datentyp	Beschreibung
PageUp	BOOL	Springt an den Anfang der aktuellen Seite und danach immer eine Seite weiter nach oben. Die Größe der Seite wird über den Parameter "LoggerListSize" der Struktur MpComLoggerUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
PageDown	BOOL	Springt ans Ende der aktuellen Seite und danach immer eine Seite weiter nach unten. Die Größe der Seite wird über den Parameter "LoggerListSize" der Struktur MpComLoggerUISetupType definiert **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepUp	BOOL	Wählt vorhergehenden Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
StepDown	BOOL	Wählt nächsten Eintrag in der Liste aus **VC4 Anbindung:** Datapoint von *Button* / Type: SetDatapoint / SetValue: 1 / ResetValue: 0
RangeStart	BOOL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Startwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** StartDatapoint von *Scale*
RangeEnd	BOOL	Zeigt in Balkenform an, welcher Teil der Liste gerade angezeigt wird. Wird für den Endwert der "Range" eines Scale Elements der Visualisierung verwendet. Diese Scale Element sollte mit der (möglichen) Größe der Liste korrespondieren. Die "Range" sollte mit der Anzahl an Einträgen, die auf einer Seite angezeigt werden, korrespondieren **VC4 Anbindung:** EndDatapoint von *Scale*



10.5.2.1.9 MpComLoggerUISetupType

Zusätzliche Konfigurationsmöglichkeiten, um die Visualisierung zu beschreiben.

Parameter	Datentyp	Wert	Beschreibung
LoggerListSize	`UINT`	20	Zahl an Ereignissen, die auf einer Seite der Visualisierung dargestellt werden können Es können maximal 20 Ereignisse angezeigt werden
LoggerListScrollWindow	`USINT`	1	Zeigt an, wieviele Ereignisse beim auf- oder abscrollen im Vorhinein angezeigt werden

10.5.2.1.10 MpComStatusIDType

Dieser Datentyp wird in der Struktur für zusätzliche Informationen als Unterstruktur für die Diagnose verwendet und liefert weitere Infos über die StatusID.

Parameter	Datentyp	Beschreibung
ID	`MpComErrorEnum`	Fehlerkode der mapp Komponente
Severity	`MpComSeveritiesEnum`	Beschreibt welche Art Info die StatusID liefert (Erfolg, Information, Warnung, Fehler)
Code	`UINT`	Code der Status ID. Diese Fehlernummer kann in der Hilfe für weitere Informationen gesucht werden

10.5.2.2 Enumeratoren

10.5.2.2.1 MpComConfigScopeEnum

Über diesen Aufzählungsdatentyp wird definiert welche Konfigurationen exportiert, bzw. importiert werden sollen.

Aufzählung	Beschreibung
mpCOM_CONFIG_SCOPE_COMPONENT	Importiert, bzw. exportiert die Konfiguration nur für die definierte Komponente
mpCOM_CONFIG_SCOPE_BRANCH	Importiert, bzw. exportiert die Konfigurationen für den gesamten Zweig (siehe Abschnitt Bilden von modularen Applikationen)

10.5.2.2.2 MpComFacilitiesEnum

Dieser Aufzählungsdatentyp unterteilt die "StatusID" Informationen in unterschiedliche Anwendungsbereiche (z.B. Infrastruktur, Industrie, Mechatronik, ...).

Aufzählung	Beschreibung
mpCOM_FAC_UNDEFINED	Kein aktiver Fehler
mpCOM_FAC_ARCORE	Dieser Code wird für Fehler verwendet, die aus der Altlast von AR Bibliotheken entstehen (alte Fehler die nicht mit einem Anwendungsbereich versehen wurden)
mpCOM_FAC_SAFETY1	Sicherheit als Anwendungsbereich
mpCOM_FAC_SAFETY2	Sicherheit als Anwendungsbereich
mpCOM_FAC_GMC1	GMC als Anwendungsbereich
mpCOM_FAC_GMC2	GMC als Anwendungsbereich
mpCOM_FAC_MAPP_INTERNAL	mapp Intern als Anwendungsbereich
mpCOM_FAC_MAPP_CORE	mapp Kern als Anwendungsbereich
mpCOM_FAC_MAPP_INFRASTRUCTURE	mapp Infrastruktur als Anwendungsbereich
mpCOM_FAC_MAPP_MECHATRONIC	mapp Mechatronik als Anwendungsbereich
mpCOM_FAC_MAPP_INDUSTRY	mapp Industrie als Anwendungsbereich

10.5.2.2.3 MpComSeveritiesEnum

Dieser Aufzählungsdatentyp unterteilt die "StatusID" Informationen in unterschiedliche Schweregrade von "erfolgreich" bis "Fehler".

Aufzählung	Beschreibung
mpCOM_SEV_SUCCESS	Aktion erfolgreich ausgeführt
mpCOM_SEV_INFORMATIONAL	Schweregrad: Infomation
mpCOM_SEV_WARNING	Schweregrad: Warnung
mpCOM_SEV_ERROR	Schweregrad: Fehler

10.5.3 Statusnummern

10.5.3.1 1083180608: Das Filterelement hat keinen Wertebereich

Beschreibung:

Das Filterelement hat keinen Wertebereich

Ursache / Abhilfe:

- Überprüfen ob im Filterelement das Symbol "-" verwendet wurde

10.5.3.2 1083180557: Daten-Traversal wurde unterbrochen

Beschreibung:

Daten-Traversal wurde unterbrochen

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.3 1083180538: Durchlauf abgebrochen

Beschreibung:

Der Durchlauf wurde abgebrochen.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.4 1083180031: Laufzeit unterstützt das Lizenzsystem nicht

Beschreibung:

Verwendete Laufzeit unterstützt das Lizenzsystem nicht. Technology Guarding wurde übersprungen.

10.5.3.5 1083180030: Lizenzsystem konnte nicht lokalisiert werden

Beschreibung:

Das Lizenzsystem konnte nicht lokalisiert werden. Technology Guard wurde übersprungen.

10.5.3.6 1083180007: Informations-Eintrag

Beschreibung:

Informations-Eintrag. Wird von mapp-Bibliotheken eingetragen. Für mehr Information siehe Beschreibungstext im Logger.

10.5.3.7 1083180039: mapp Services Version

Beschreibung:

Aktuell verwendete mapp Services Version: {1:Major}.{2:Minor}.{1:Bugfix}

10.5.3.8 1083180047: Aktuelle mapp Version

Beschreibung:

Aktuell verwendete mapp Legacy Version: {1:Major}.{2:Minor}.{3:Bugfix}

10.5.3.9 -2138044953: Komponente ist bereits verlinkt.

Beschreibung:

Die Komponente/Konfiguration {2:RegistryName} wurde bereits in {3:OldParent} verlinkt und wurde in {4:NewParent} nocheinmal verlinkt.

Ursache / Abhilfe:

- Kontrolle ob nur eine Komponente zu einem Parent verlinkt ist
- Überprüfung der Konfiguration, ob ein Parameter mehrmals verlinkt wurde

Zusätzliche Information

- {2:RegistryName}: Komponente/Konfiguration in welcher der Fehler aufgetreten ist
- {3:OldParent}: Erste Verlinkung
- {4:NewParent}: Neue Verlinkung

10.5.3.10 -2138044941: Konfigurationstyp ist nicht gleich Dateiinhalt

Beschreibung:

Der gegebene Konfigurationstyp stimmt nicht mit dem Dateiinhalt überein.

Ursache / Abhilfe:

- Überprüfen der Konfiguration

10.5.3.11 -2138045423: Taskname zu lang

Beschreibung:

Der verwendete Taskname "{2:TaskName}" ist zu lang und wird deshalb auf "{3:NewTaskName}" gekürzt.

Zusätzliche Information

- {2:TaskName}: Name des Tasks welcher zu lange ist
- {3:NewTaskName}: Neuer gekürzter Taskname

Ursache / Abhilfe:

- Werden Prozessvariablen aus diesem Task referenziert, mit beispielsweise MpRecipeRegPar oder MpDataRegPar, so gilt es zu beachten, dass bei der Angabe des Tasknamen der gekürze Taskname verwendet werden muss. ("ShortName:PVName")
- Anpassen des Tasknamens, damit dieser nicht gekürzt werden muss.

10.5.3.12 -106430380: Angegebener Abonennt existiert nicht

Beschreibung:

Der angegebene Abonennt existiert nicht.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.13 -1064303640 : Kein Zugriff auf MpCom

Beschreibung:

Es konnte nicht auf MpCom zugegriffen werden.

Ursache / Abhilfe:

- Überprüfen der mapp Version
- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.14 -1064303639: Service wurde nicht gefunden

Beschreibung:

Der Service {2:ServiceName} konnte nicht gefunden werden.

Zusätzliche Information

- {2:ServiceName}: Name des Service

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.15 -1064303638 : Konfiguration wurde nicht gefunden

Beschreibung:

Die Konfiguration {2:ServiceName} konnte nicht gefunden werden.

Zusätzliche Information

- {2:ServiceName}: Name der Konfiguration

Ursache / Abhilfe:

- Überprüfen ob Konfiguration hinzugefügt wurde.
- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.16 -1064303637 : Mehrfachzugriff nicht möglich

Beschreibung:

Gleichzeitiger Zugriff auf Konfigurations-Daten.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.17 -1064303636 : Datei ist nicht geöffnet

Beschreibung:

Geforderte Datei ist nicht geöffnet.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.18 -1064303635 : Es wurde ein NULL Zeiger angegeben

Beschreibung:

Es wurde ein NULL Zeiger angegeben.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.19 -1064303634 : Speicherzugriff nicht erlaubt

Beschreibung:

Der Speicherzugriff ist nicht erlaubt.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.20 -1064303633 : Ungültiges Argument

Beschreibung:

Es wurde ein ungültiges Argument angegeben.

Ursache / Abhilfe:

- Überprüfen des Arguments
- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.21 -1064303632 : Funktion/ Methode ist nicht implementiert

Beschreibung:

Die Funktion/ Methode ist nicht implementiert.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.22 -1064303631: Registrierung konnte nicht erstellt werden

Beschreibung:

Die Registrierung der Komponente {2:ComponentName} konnte nicht durchgeführt werden.

Zusätzliche Information

- {2:ComponentName}: Komponenten Name

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.23 -1064303630: Registrierung des Diagnose Web-Service fehlgeschlagen

Beschreibung:

Web-Server / MpServer konnte nicht instanziert werden. (Fehlerursache: {1:ErrorNumber})

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.24 -1064303629: Child konnte nicht hinzugefügt werden

Beschreibung:

Child {2:ChildName} konnte dem Parent {3:ParentName} nicht hinzugefügt werden.

Zusätzliche Information

- {2:ChildName}: Child Name
- {3:ParentName}: Parent Name

Ursache / Abhilfe:

- Es wurde evtl. eine Komponent mehrmals der gleichen Parent-Komponente zugewiesen
- Überprüfen der Konfiguration

10.5.3.25 -1064303628: Child konnte nicht entfernt werden

Beschreibung:

Child {2:ChildName} konnte vom Parent {3:ParentName} nicht entfernt werden.

Zusätzliche Information

- {2:ChildName}: Child Name
- {3:ParentName}: Parent Name

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.26 -1064303627: Komponenten Web-Service konnte nicht entfernt werden

Beschreibung:

Komponenten Web-Service konnte nicht entfernt werden. (Fehlerursache: {1:ErrorNumber})

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.27 -1064303626: Zugriff auf Konfiguration fehlgeschlagen

Beschreibung:

Es konnte nicht auf die Konfiguration {2:ConfigName} zugegriffen werden. (Fehlerursache: {1:ErrorNumber})

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache
- {2:ConfigName}: Name der Konfiguration

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.28 -1064303625: Addon existiert bereits

Beschreibung:

Das Addon {2:ChildName} existiert bereits für {3:ParentName}

Zusätzliche Information

- {1:ChildName}: Name des Addons
- {2:ParentName}: Name des Parents

Ursache / Abhilfe:

- Es wurden mehr als 1 UI-FB an der gleichen mapp-Komponente verwendet.
- Es wurde der gleiche mapp Link für mehrere Funktionsbausteine verwendet, welche eigentlich eine eigene mapp Konfiguration benötigen.
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

10.5.3.29 -1064303624: Konfigurations Funktionsblock konnte nicht lokalisiert werden

Beschreibung:

Der Konfigurations Funktionsblock konnte nicht lokalisiert werden. (Fehlerursache: {2:ErrorNumber})

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Konfig-FB wurde nicht als PV deklariert (sondern z.B. als C-Variable)

10.5.3.30 -1064303623: Konfigurations PV konnte nicht lokalisiert werden

Beschreibung:

Die Konfigurations PV konnte nicht lokalisiert werden. (Fehlerursache: {2:ErrorNumber})

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen der PV
- Konfigurations-Varaible evtl. nicht als PV deklariert

10.5.3.31 -1064303622: Bibliothek Faceplate konnte nicht registriert werden

Beschreibung:

Das Bibliotheks Faceplate konnte nicht registriert werden. (Fehlerursache: {2:ErrorNumber})

Zusätzliche Information

- {1:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.32 -1064303621: Revisionsinformation der Bibliothek unbekannt

Beschreibung:

Die Revision der Bibliothek ist unbekannt. Erwartet {2:RevisionSet} / Aktuell {3:RevisionAct}

Zusätzliche Information

- {1:RevisionSet}: Erwartete Revision
- {2:RevisionAct}: Aktuelle Revision

Ursache / Abhilfe:

- Version überprüfen
- Version updaten

10.5.3.33 -1064303620: Registrierung der Lizenz fehlgeschlagen

Beschreibung:

Die Lizenz {2:LicenceName} konnte nicht registriert werden. (Fehlerursache: {1:ErrorNumber})

Zusätzliche Information

- {1:LicenceName}: Name der Lizenz
- {2:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen der Lizenz
- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.34 -1064303619: Deregistrierung der Lizenz fehlgeschlagen

Beschreibung:

Die Lizenz {2:LicenceName} konnte nicht deregistriert werden. (Fehlerursache: {1:ErrorNumber})

Zusätzliche Information

- {1:LicenceName}: Name der Lizenz
- {2:ErrorNumber}: Fehlerursache

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.35 -1064303616: PV nicht gefunden

Beschreibung:

Die angegebene PV konnte nicht gefunden werden.

Ursache / Abhilfe:

- Überprüfen der angegebene PV
- Überprüfen ob PV im Programm verwendet wurde

10.5.3.36 -1064303615: Referenz ist NULL

Beschreibung:

Die angegebene Referenz ist NULL.

Ursache / Abhilfe:

- Überprüfen der angegebene Referenz

10.5.3.37 -1064303614: Fehler in der Abfrage

Beschreibung:

Die Konfiguration der Abfrage ist fehlerhaft.

Ursache / Abhilfe:

- Überprüfen der Abfrage in der Konfiguration

10.5.3.38 -1064303613: Ungültige Filter Bedingung

Beschreibung:

Die angegebene Filter Bedingung für die Abfrage ist in der Konfiguration ungültig.

Ursache / Abhilfe:

- Überprüfen der Abfrage in der Konfiguration
- Überprüfen der Filter Bedingung in der Konfiguration

10.5.3.39 -1064303612: Ungültige Selektion

Beschreibung:

Die angegebene Selektion für die Abfrage ist in der Konfiguration ungültig.

Ursache / Abhilfe:

- Überprüfen der Selektion für die Abfrage in der Konfiguration

10.5.3.40 -1064303611: Ungültige Filter-Logik

Beschreibung:

Die angegebene Filter-Logik für die Abfrage ist in der Konfiguration ungültig.

Ursache / Abhilfe:

- Überprüfen der Filter-Logik für die Abfrage in der Konfiguration

10.5.3.41 -1064303610: Ungültige Update-PV

Beschreibung:

Die angegebene PV für die Aktualisierungs-Information einer Abfrage ist ungültig.

Ursache / Abhilfe:

- Überprüfen der Update PV in der Konfiguration für die Abfrage

10.5.3.42 -1064303140: Der Service konnte nicht entfernt werden

Beschreibung:

Der Service konnte nicht entfernt werden, da dieser noch immer referenziert wird.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.43 -1064303139: IDs sind nicht eindeutig

Beschreibung:

Ein oder mehrere IDs sind nicht eindeutig.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.44 -1064303138: Dienst wurde nicht gefunden

Beschreibung:

Der Dienst wurde nicht gefunden.

Ursache / Abhilfe:

- Überprüfen ob alle benötigten Bibliotheken hinzugefügt wurden

10.5.3.45 -1064303137: Der Dienst existiert bereits

Beschreibung:

Der Dienst existiert bereits.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

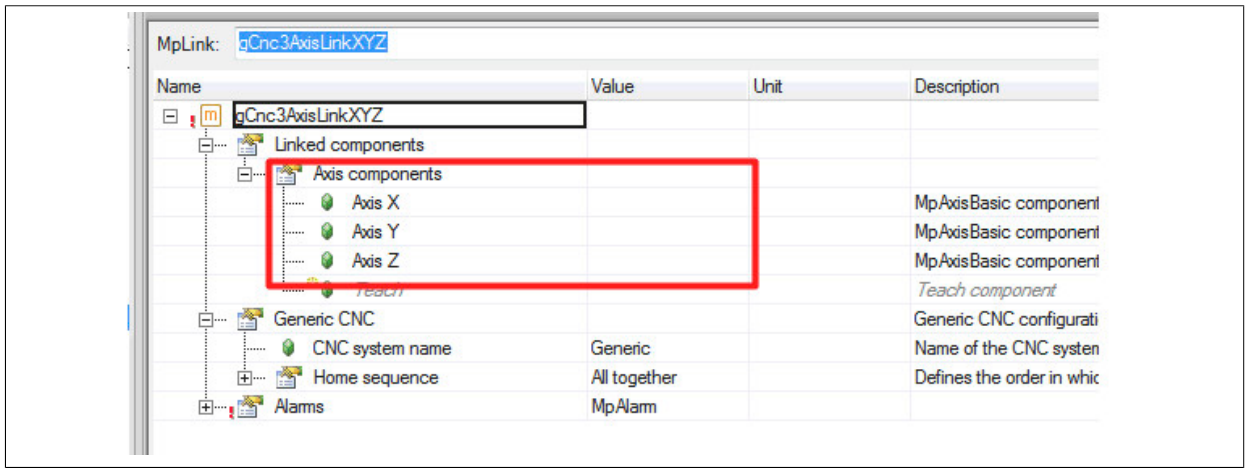
10.5.3.46 -1064303136: Ungültige Registrierung

Beschreibung:

Ungültige Registrierung für gewünschte Operation. Verlinkung von anderen mapp Komponenten fehlt.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.
- Muss in einer mapp Konfiguration auf andere mapp Konfigurationen verlinkt werden, so muss darauf geachtet werden, dass der mapp Link auch angegeben wird:



10.5.3.47 -1064303135: Dienst bereits registriert

Beschreibung:

Der angegebene Dienst ist bereits registriert.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.48 -1064303134: Registrierung enthält aktive Links

Beschreibung:

Die Registrierung enthält aktive Links.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.49 -1064303133: Fehler beim Initialisieren von Systemelementen

Beschreibung:

Fehler beim Initialisieren von Systemelementen. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.50 -1064303132: Verbindung eines System-Elements zu Parent fehlgeschlagen

Beschreibung:

Fehler beim Verbinden eines System-Elements zu einem Parent

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers

Ursache / Abhilfe:

- Konfiguration überprüfen

10.5.3.51 -1064303131: Systemelement-PV ist ungültig

Beschreibung:

Systemelement-Prozessvariable {2:SysElementName} ist ungültig. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers
- {2:SysElementName}: Name der Systemelement-PV

Ursache / Abhilfe:

- Überprüfung der PV auf gültigen Datentyp

10.5.3.52 -1064303130: Fehler beim Zugriff auf Pfad

Beschreibung:

Fehler beim Zugriff auf Pfad {2:DeviceName}. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:DeviceName}: Name des Pfades, wie zum Beispiel *C:\Temp*

Ursache / Abhilfe:

- Überprüfen des angegbenen Speichermedium

10.5.3.53 -1064303120: Konfiguration nicht gefunden

Beschreibung:

Die Konfiguration konnte nicht gefunden werden.

Ursache / Abhilfe:

- Überprüfen ob richtige Konfiguration hinzugefügt wurde
- Überprüfen ob Konfiguraiton sich auf dem Zielsystem befindet

10.5.3.54 -1064303119: Sektion nicht gefunden

Beschreibung:

Der Abschnitt {3:SectionName} innerhalb der Konfiguration {2:ConfigurationName} wurde nicht gefunden.

Zusätzliche Informationen

- {2:ConfigurationName}: Name der Konfiguration.
- {3:SectionName}: Name des nicht gefundenen Abschnitts

Ursache / Abhilfe:

- Überprüfen der Konfiguration auf Vollständigkeit

10.5.3.55 -1064303118: Mehrfachzugriff nicht möglich

Beschreibung:

Mehrfachzugriff auf Konfiguration nicht möglich.

Ursache / Abhilfe:

- Sicherstellen, dass auf die Konfiguration nur einmal zugegriffen wird

10.5.3.56 -1064303116: Pfad ungültig

Beschreibung:

Der Pfad innerhalb der Konfiguration ist nicht gültig.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.57 -1064303115: Fehlender Konfigurationseintrag

Beschreibung:

Der Eintrag {3:PropertyName} fehlt in der Konfiguration {2:ConfigurationName}.

Zusätzliche Informationen

- {2:ConfigurationName}: Name der Konfiguration.
- {3:PropertyName}: Name des Eintrages, der nicht gefunden werden konnte.

Ursache / Abhilfe:

- Überprüfung der Konfiguration

10.5.3.58 -1064303114: Konfiguration enthält Einträge mit ungültigem Format

Beschreibung:

Konfiguration {2:ConfigurationName}: {3:PropertyName} enthält Einträge mit ungültigem Format.

Zusätzliche Informationen

- {2:ConfigurationName}: Name der betroffenen Konfiguration.
- {3:PropertyName}: Name der Eigenschaft, welche ein ungültiges Format aufweist.

Ursache / Abhilfe:

- Eintrag mit gültigen Format angeben

10.5.3.59 -1064303113: Speichern der Konfiguration fehlgeschlagen

Beschreibung:

Fehler beim Speichern einer Konfiguration

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.60 -1064303112: Fehler beim Laden der Konfiguration

Beschreibung:

Fehler beim Laden der Konfiguration {2:ConfigName}. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:ConfigName}: Name der betroffenen Konfiguration.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.61 -1064303111: Fehler beim Löschen

Beschreibung:

Fehler beim Löschen des Datenobjekts.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.62 -1064303109: Keine Daten vorhanden

Beschreibung:

Keine Daten vorhanden.

Ursache / Abhilfe:

- Überprüfen ob Daten angegeben wurden

10.5.3.63 -1064303108: Eintrag existiert bereits

Beschreibung:

Der Konfigurationseintrag existiert bereits.

Ursache / Abhilfe:

- Überprüfen ob Eintrag nur einmal existiert

10.5.3.64 -1064303107: Fehler beim Setzen des Selektors

Beschreibung:

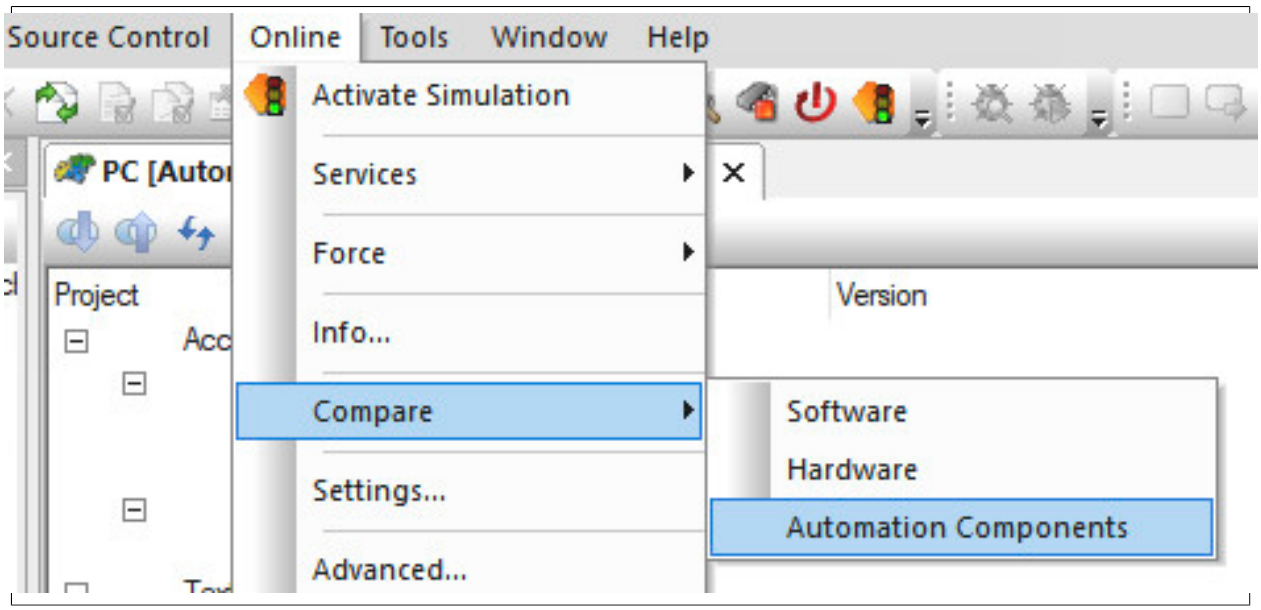
Fehler beim Setzen des internen Selektors {2} auf {3}

Zusätzliche Information

- {2}: Selektor-Pfad
- {3}: Wert

Ursache / Abhilfe:

- Konfiguration wird über die Vergleichs-Funktion "Automation Components" nicht richtig dargestellt. Konfiguration funktioniert einwandfrei. Bei einer neuen Übertragung der Konfiguration wird die Konfiguration wieder richtig dargestellt.



10.5.3.65 -1064294640: Struktur nicht gefunden

Beschreibung:

Die angegebene Konfigurations-Struktur konnte nicht gefunden werden

Ursache / Abhilfe:

- Überprüfen der angegebenen Prozessvariablen in der Konfiguration
- Prüfen ob Prozessvariable in Task verwendet wird und Task in der Softwarekonfiguration angegeben wurde

10.5.3.66 -1064303100: Pfad nicht gültig

Beschreibung:

Der Pfad innerhalb des Datenmodells ist nicht gültig.

Ursache / Abhilfe:

- Überprüfen des Datenmodells

10.5.3.67 -1064303099: Eintrag nicht vorhanden

Beschreibung:

Der Eintrag ist nicht vorhanden.

Ursache / Abhilfe:

- Überprüfen des Eintrags

10.5.3.68 -1064303098: Ungültige Datei

Beschreibung:

Datei ist ungültig {2:FileName}.

Zusätzliche Informationen

- {2:FileName}: Name der betroffenen Datei.

Ursache / Abhilfe:

- Überprüfen der Datei auf gültigen Inhalt und gültiges Format
- Bei der Verwendung von verschlüsselten Passwörtern in der Konfiguration muss darauf geachtet werden, dass das Passwort im Nachhinein nicht manipuliert wurde, wie beispielsweise hinzufügen oder löschen von Zeichen. Für mehr Information siehe hier im Abschnitt "Passwortverschlüsselung in der Konfiguration"

10.5.3.69 -1064303097: Daten konnten nicht gespeichert werden

Beschreibung:

Die Daten der Datei {2:FileName} konnten nicht auf dem Speichermedium {3:DeviceName} abgespeichert werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:FileName}: Die Datei, welche gespeichert werden sollte.
- {3:DeviceName}: Speichermedium, auf dem gespeichert werden sollte.

Ursache / Abhilfe:

- Überprüfen des angegebenen Speichermediums
- Überprüfen ob abzuspeichernde Datei keinen ungültigen Inhalt enthält

10.5.3.70 -1064303096: Daten konnten nicht geladen werden

Beschreibung:

Die Daten der Datei {2:FileName} konnten vom Speichermedium {3:DeviceName} nicht geladen werden. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.
- {2:FileName}: Name der betroffenen Datei.
- {3:DeviceName}: Name des betroffenen Gerätes.

Ursache / Abhilfe:

- Überprüfen des Speichermediums
- Überprüfen der Datei auf gültigen Inhalt

10.5.3.71 -1064303095: Unerlaubter Zugriff auf Auflistung

Beschreibung:

Unerlaubter Zugriff auf Auflistung, wie zum Beispiel versuchter Zugriff mittels "Key".

Ursache / Abhilfe:

- Kontrolle von durcheführten Zugriff

10.5.3.72 -1064303094: Keine Daten vorhanden

Beschreibung:

Für {2:GroupName} sind keine Daten vorhanden.

Zusätzliche Informationen

- {2:GroupName}: Name der Gruppe (Knoten) in dem Daten fehlen

Ursache / Abhilfe:

- Hinzufügen der fehlenden Daten

10.5.3.73 -1064303093: Datenüberlauf

Beschreibung:

Datenüberlauf: Daten können nicht gespeichert werden

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.74 -1064303092: Datenmodell existiert

Beschreibung:

Das Datenmodell existiert bereits.

Ursache / Abhilfe:

- Name des Datenmodells ändern

10.5.3.75 -1064303090: Einheit kann nicht konvertiert werden

Beschreibung:

Die Einheit kann nicht konvertiert werden.

Ursache / Abhilfe:

- Überprüfen ob für eine Variable 2 Einheiten (Engineering-Einheit, Display-Einheit) angegeben wurde, welche nicht ineinander konvertierbar sind. Es müssen immer konvertierbare Einheiten angegeben werden, wie zum Beispiel funktioniert die Konvertierung von "mm" nicht nach "kg"!

10.5.3.76 -1064303089: Keine Einheit vorhanden

Beschreibung:

Keine Einheit bei der angegebenen Variable vorhanden

Ursache / Abhilfe:

- Einheit einer PV zuweisen

10.5.3.77 -1064303087: Ungültiger Datenpunkt

Beschreibung:

Ungültiger Datenpunkt {2:Name} in der Anzeige-Konfiguration ("Display-Unit")

Zusätzliche Information

- {2:Name}: Name des Datenpunkts

Ursache / Abhilfe:

- Überprüfen der Konfiguration
- Für weitere Details siehe Logger

10.5.3.78 -1064303086: Ungültige Fallback-Einheit

Beschreibung:

Ungültige Konfiguration für Fallback-Einheit {3:Namespace}/{2:Unit-ID}.

Zusätzliche Information

- {3:Namespace}: Namespace
- {2:Unit-ID}: Einheiten-ID

Ursache / Abhilfe:

- Überprüfen der angegebenen Fallback-Einheit in der Einheiten-Konfiguration
- Für weitere Details siehe Logger

10.5.3.79 -1064303085: Ungültige PV für Limit

Beschreibung:

Für die Limit-Angabe wurde eine ungültige PV angegeben.

Ursache / Abhilfe:

- Überprüfen der angegebenen PV in der mapp Konfiguration

10.5.3.80 -1064303084: Nicht übereinstimmende Einheit

Beschreibung:

Die Einheiten zwischen {2:EU} und {2:DU} stimmen nicht überein. Die Einheiten können nicht ineinander konvertiert werden.

Zusätzliche Information

- {2:EU}: Engineering-Einheit
- {2:DU}: Display-Einheit

Ursache / Abhilfe:

- Überprüfen ob für eine Variable 2 Einheiten (Engineering-Einheit, Display-Einheit) angegeben wurde, welche nicht ineinander konvertierbar sind. Es müssen immer konvertierbare Einheiten angegeben werden, wie zum Beispiel funktioniert die Konvertierung von "mm" nicht nach "kg"!

10.5.3.81 -1064303083: Ungültige Einheiten-Definition

Beschreibung:

Ungültige Einheiten-Definition für {3:Namespace}/{2:Unit-ID}

Zusätzliche Information

- {3:Namespace}: Namespace
- {2:Unit-ID}: Eineheiten ID

Ursache / Abhilfe:

- Überprüfen der angegebenen Einheiten-Code + Einheiten-Namespace Kombination

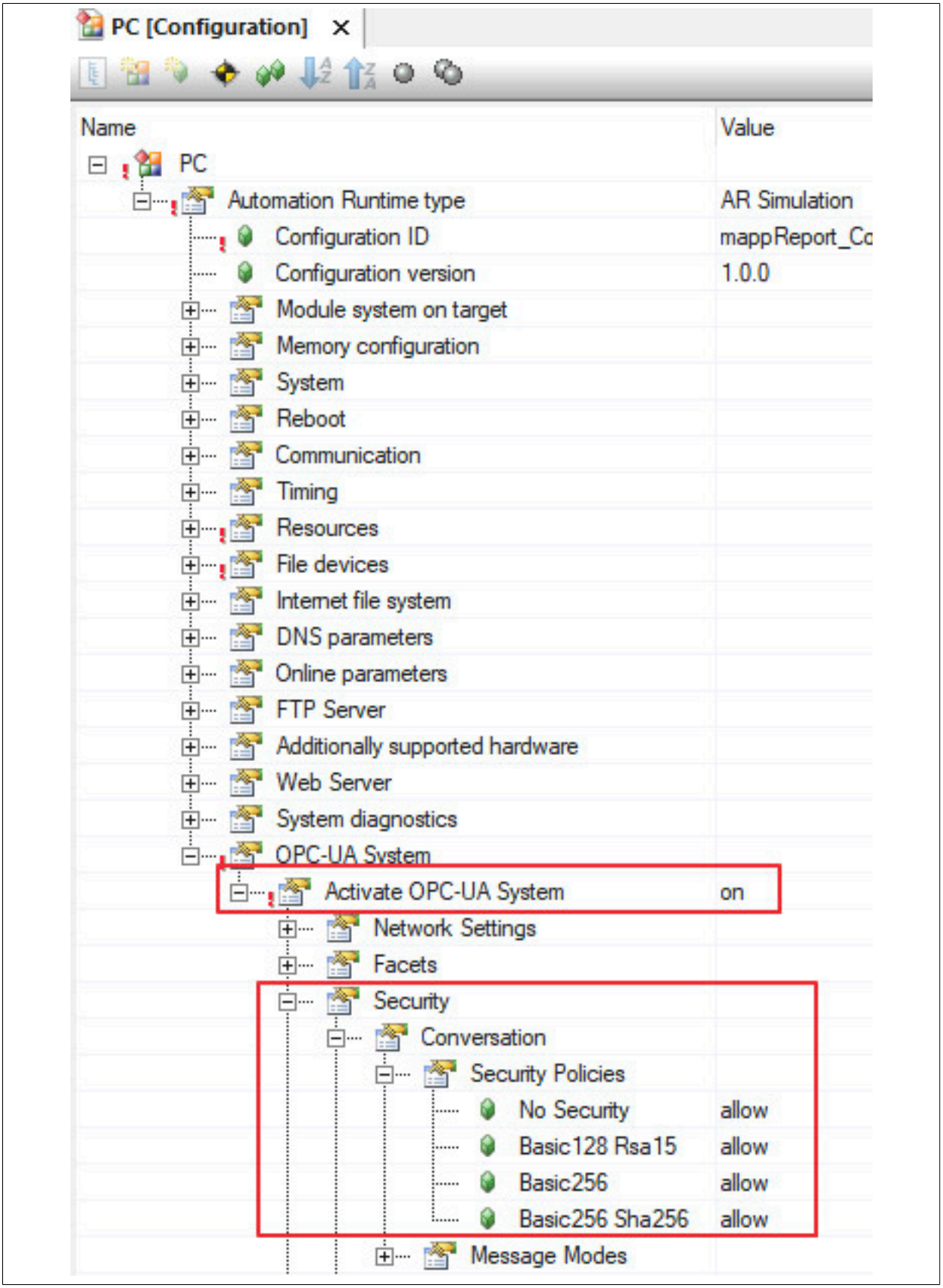
10.5.3.82 -1064303082: OPC UA Server nicht verfügbar

Beschreibung:

Der OPC UA Server ist nicht verfügbar.

Ursache / Abhilfe:

- Aktivieren des OPC UA Servers in der Konfiguration des Zielsystems
- Überprüfen der Sicherheitsrichtlinien in den OPC UA Einstellungen. Die Kommunikation durch OPC UA muss erlaubt sein.



10.5.3.83 -1064303081: Engineering Einheit nicht verfügbar

Beschreibung:

Die Engineering-Einheit der OPC-UA-Node ist nicht verfügbar.

Ursache / Abhilfe:

- Für die Variable muss eine Engineering-Einheit angegeben werden.

10.5.3.84 -1064303080: Angegebener Empfänger existiert nicht

Beschreibung:

Der angegebene Empfänger existiert nicht.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.85 -1064303070: Ungültige Messsystem ID

Beschreibung:

Die angegebene Messsystem-ID ist ungültig.

Ursache / Abhilfe:

- Überprüfen der Messsystem ID

10.5.3.86 -1064303069: Ungültige Konfiguration einer Einheitengruppe

Beschreibung:

Die angegebene Konfiguration einer Einheitengruppe ist ungültig.

Ursache / Abhilfe:

- Kontrollieren der angegebenen Einheitengruppe
- Für weitere Details siehe Logger

10.5.3.87 -1064303068: Ungültige Einheitengruppen-ID

Beschreibung:

Angabe einer ungültigen Einheitengruppen-ID.

Ursache / Abhilfe:

- Überprüfen der angegebenen ID
- Vorhandene Einheitengruppe angeben

10.5.3.88 -1064303067: Ungültige Einheitengruppen-PV

Beschreibung:

Angabe einer ungültigen PV in der Einheitengruppe

Ursache / Abhilfe:

- Angabe einer gültigen PV

10.5.3.89 -1064303060: PV nicht gefunden

Beschreibung:

An der angegebenen Adresse wurde keine Prozessvariable gefunden.

Ursache / Abhilfe:

- Überprüfen der angegebenen PV
- Sicherstellen ob PV im Programm verwendet wird
- Sicherstellen, ob Variable welche an mapp Komponente verbunden wurde, auch wirklich eine PV ist

10.5.3.90 -1064303050: Eintrag nicht vollständig

Beschreibung:

Eintrag für Konfigurations-Eigenschaft unvollständig.

Ursache / Abhilfe:

- Überprüfen der Konfiguration

10.5.3.91 -1064303049: Eintrag nicht korrekt

Beschreibung:

Typeneintrag für Eigenschaft {2:PropertyName} nicht korrekt.

Ursache / Abhilfe:

- Überprüfung der Konfiguration

10.5.3.92 -1064303048: Ungültiger Knoten

Beschreibung:

Ungültiger Knoten {2:NodeName} in Importdaten. Datei wurde übersprungen

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.93 -1064303039: Logger nicht gefunden

Beschreibung:

Logger FB-Instanz konnte nicht gefunden werden.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.94 -1064303030: Fehler beim Registrieren

Beschreibung:

Fehler beim Registrieren einer neuen Textformatierungs Datenquelle. Fehlerursache: {1:ErrorNumber}

Zusätzliche Informationen

- {1:ErrorNumber}: Grund des Fehlers.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.95 -1064303029: Fehler beim Lokalisieren

Beschreibung:

Fehler beim Lokalisieren der Laufzeitunterstützung für zusätzliche Textformatierungs Datenquellen.

Ursache / Abhilfe:

- Installieren der richtigen Laufzeit

10.5.3.96 -1064294634: Schreibprozess fehlgeschlagen

Beschreibung:

Schreiben von {3} auf {3} fehlgeschlagenn (Fehlerursache: {1})

Zusätzliche Information

- {3}: Konfiguration
- {1}: Fehlerursache

Ursache / Abhilfe:

- Überprüfen der Konfiguration

10.5.3.97 -1064239103: Komponente konnte nicht erstellt werden

Beschreibung:

Die mapp Komponente konnte nicht erstellt werden und wird nicht aktiviert. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Service zur Konfiguration des Funktionsbausteins nicht verfügbar - Problem mit MpCom
- Registry nicht lesbar - Problem mit MpCom
- Details zur Fehlerursache im Logger
- Es müssen die Transfereinstellungen für mapp Services Komponenten beachtet werden. Der Fehler tritt auch auf, wenn die Funktionsbausteine beispielsweise nicht im EXIT-Teil des Programms deaktiviert wurden, als Transfereinstellung aber "Execute Exit/Init routines" ausgewählt wurde. Details können hier gefunden werden.

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLoggerUI

Konstante:

`mpCOM_ERR_ACTIVATION`

10.5.3.98 -1064239102: MpLink ist ein Nullzeiger

Beschreibung:

Der "MpLink" Eingang ist nicht verbunden - NULL Zeiger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- "MpLink" Eingang am Funktionsbaustein überprüfen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLink
- MpComLinkToParent
- MpComDump

Konstante:

`mpCOM_ERR_MPLINK_NULL`

10.5.3.99 -1064239101: MpLink Verbindung nicht erlaubt

Beschreibung:

Der verwendete Wert am "MpLink" Eingang ist nicht erlaubt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Komponente ist direkt mit mpCOM_MAIN oder mpCOM_STANDALONE verbunden, das wird nicht unterstützt

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLink
- MpComLinkToParent

Konstante:

`mpCOM_ERR_MPLINK_INVALID`

10.5.3.100 -1064239100: MpLink wurde verändert

Beschreibung:

Der Wert am "MpLink" Eingang hat sich zur Laufzeit der Komponenten ("Enable" = `TRUE`) verändert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Den Wert am "MpLink" Eingang nur verändern während die Komponente inaktiv ist ("Enable" = `FALSE`)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLink
- MpComLinkToParent

Konstante:

`mpCOM_ERR_MPLINK_CHANGED`

10.5.3.101 -1064239099: MpLink Inhalt ist ungültig

Beschreibung:

Der Wert in der Variable "MpLink" am Eingang des Funktionsbaustein ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die Werte in der Struktur MpComIdentType dürfen nicht beschrieben werden
- Es muss eine entsprechende Konfiguration für diese mapp Komponenten vorhanden sein (siehe erster Absatz der Beschreibung von den unten angeführten Funktionen / Funktionsbausteinen)

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLink
- MpComLinkToParent

Konstante:

`mpCOM_ERR_MPLINK_CORRUPT`

10.5.3.102 -1064239098: MpLink wird bereits benützt

Beschreibung:

Dieser MpLink wird bereits benützt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Ist der ausgewählte Download Modus "Overload", sollte im Exit Programm die Mapp Komponente mit "Enable = FALSE" deaktiviert werden. Als Abhilfe kann auch der Modus "Copy" oder "One cycle" verwendet werden. Hierbei ist es nicht nötig die mapp Komponente zu deaktivieren.
- Überprüfen welche Komponente diesen MpLink bereits benützt
- Neuen MpLink erstellen

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLinkToParent
- MpComLink

Konstante:

`mpCOM_ERR_MPLINK_IN_USE`

10.5.3.103 -1064238848: Verbindung nicht möglich

Beschreibung:

Erstellen der geforderten Verbindung trotz gültiger Identifikation nicht möglich. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Die genaue Ursache wird im Logger eingetragen

Konstante:

`mpCOM_ERR_LINK_NOT_POSSIBLE`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLink
- MpComLinkToParent

10.5.3.104 -1064238847: Fehler beim Datei bearbeiten

Beschreibung:

System Fehler aufgetreten beim bearbeiten der Datei. Weitere Details siehe Logger.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Weitere Details siehe Logger

Konstante:

`mpCOM_ERR_FILE_SYSTEM`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComDump
- MpComConfigManager

10.5.3.105 -1064238846: Ungültiges File Device

Beschreibung:

Das gewählte Speichermedium (File Device) ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Verbindung zum Speichermedium (FileDevice) überprüfen ("DeviceName" Eingang)

Konstante:

`mpCOM_ERR_INVALID_FILE_DEV`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComDump
- MpComConfigManager

10.5.3.106 -1064238845: Ungültiger Dateiname

Beschreibung:

Der Dateiname am Eingang "FileName" ist ungültig.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Am Eingang "FileName" ist nichts angehängt
- Die angehängte Datei existiert nicht
- Keine Zugriffsberechtigung

Konstante:

`mpCOM_ERR_INVALID_FILE_NAME`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComDump
- MpComConfigManager

10.5.3.107 -1064238844: Es wurde keine Konfiguration importiert

Beschreibung:

Nach einem "Import" Befehl wurden keine Konfigurationen importiert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Kein gültiger MpLink in der Datei vorhanden
- Keine gültige Konfiguration in der Datei vorhanden

Konstante:

`mpCOM_ERR_NO_CONFIG_IMPORT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComConfigManager

10.5.3.108 -1064238843: Fehlender Wert an UIConnect

Beschreibung:

An "UIConnect" wurde NULL angehängt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Eingang "UIConnect" vergessen

Konstante:

`mpCOM_ERR_MISSING_UICONNECT`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLoggerUI

10.5.3.109 -1064238842: Fehler beim Konfigurationsimport

Beschreibung:

Ein Syntax Fehler wurde beim Importieren der Konfiguration(en) festgestellt.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Syntax der zu importierenden Konfiguration prüfen

Konstante:

`mpCOM_ERR_CONFIG_IMPORT_ERR`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComConfigManager

10.5.3.110 -1064238841: Filter Syntax falsch

Beschreibung:

Der String am "EntryFilter" Eingang, oder der Parameter "SearchFilter" der Struktur MpComLoggerUIConnectType ist falsch konfiguriert.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Syntax am Eingang "EntryFilter" prüfen
- Syntax am Eingang "SearchFilter" prüfen

Konstante:

`mpCOM_ERR_WRONG_FILTER_STRING`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComLoggerUI

10.5.3.111 -1064238840: Komponente nicht gefunden

Beschreibung:

Die angegebene mapp Komponente konnte nicht gefunden werden.

Reaktion:

Der Funktionsbaustein wird an den Ausgängen "StatusID" und "Error" anzeigen, dass ein Fehler aktiv ist, weiters sind alle Funktionen solange nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen ob richtiger mapp Link angegeben wurde.

Konstante:

`mpCOM_ERR_COMPONENT_NOT_FOUND`

Diese Funktionsbausteine / Funktionen können diesen Fehler melden:

- MpComGetLink
- MpComLoggerUI

10.5.3.112 -1064303608: mapp Version stimmt nicht

Beschreibung:

Aktuell verwendete mapp Version stimmt nicht mit dem Technologie-Paket-Service überein

Ursache / Abhilfe:

- Die Version aller mapp Bibliotheken sollte geprüft und gegebenfalls ersetzt werden.
- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.113 -2138045431: mapp Version passt nicht

Beschreibung:

Die verwendete mapp Version stimmt nicht mit dem Technologie-Paket-Service überein.

Ursache / Abhilfe:

- Die Version aller mapp Bibliotheken sollte geprüft und gegebenfalls ersetzt werden.

10.5.3.114 -1064303606: Konflikt in der Lizenzversion

Beschreibung:

Konflikt in der Lizenzversion.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.115 -1064303605: mapp TP nicht bekannt

Beschreibung:

Das angegebene mapp TP ist nicht bekannt.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.116 -1064303604: Bibliothek existiert bereits

Beschreibung:

Die Bibliothek existiert bereits.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.117 -1064303603: Bibliothek nicht registiert

Beschreibung:

Die Bibliothek ist nicht registiert.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.118 -1064303602: Logbuch existiert nicht

Beschreibung:

Das angegebene Logbuch existiert nicht.

Ursache / Abhilfe:

- Erstellen Sie mit Hilfe von MpComDump eine Datei, welche alle Informationen über verwendete Komponenten und Dienste beinhaltet und Speichern Sie zusätzlich die Loggerinformationen ab. Kontaktieren Sie mit diesen Informationen den B&R Support.

10.5.3.119 -1064303598: Password nicht lesbar

Beschreibung:

Das Passwort kann nicht von {2:Path} gelesen werden.

Zusätzliche Information

- {2:Path}: Angabe des betroffenen Passwort-Parameters

Ursache / Abhilfe:

- Wird ein Passwort in einer Konfiguration, wie beispielsweise in der MpDataRecorder Konfiguration, verschlüsselt angegeben, so kann das verschlüsselte Passwort im Nachhinein manipuliert werden. Es kann beispielsweise ein Zeichen hinzugefügt oder gelöscht werden. Wird nun die dementsprechende Komponente, wie beispielsweise mapp Data: Data Logging, aktiviert und die Konfiguration wird gelesen, so wird dieser Fehler ausgelöst. Das Passwort ist nicht mehr lesbar und muss wieder angepasst werden. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.120 -1064303597: Password ist ungültig

Beschreibung:

Das Passwort ist ungültig. Kommando kann nicht ausgeführt werden.

Ursache / Abhilfe:

- In einigen Komponenten können in der Konfiguration Passwörter verschlüsselt angegeben werden, wie beispielsweise in der MpReportCore Konfiguration. Nach der Verschlüsselung kann der Benutzer das Passwort manipulieren, wie beispielsweise ein Zeichen hinzufügen oder löschen. Das Passwort ist daraufhin ungültig. Wird nun ein Report erstellt, so erscheint dieser Fehler, da dass Passwort nicht gelesen werden kann und somit der Report nicht verschlüsselt werden kann. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.121 -1064303081: Engineering-Einheit nicht verfügbar

Beschreibung:

Für die verwendete Prozessvariable ist keine Engineering-Einheit verfügbar.

Ursache / Abhilfe:

- Definieren einer Engineering-Einheit für die Prozessvariable in der OPC-UA Default View Konfiguration

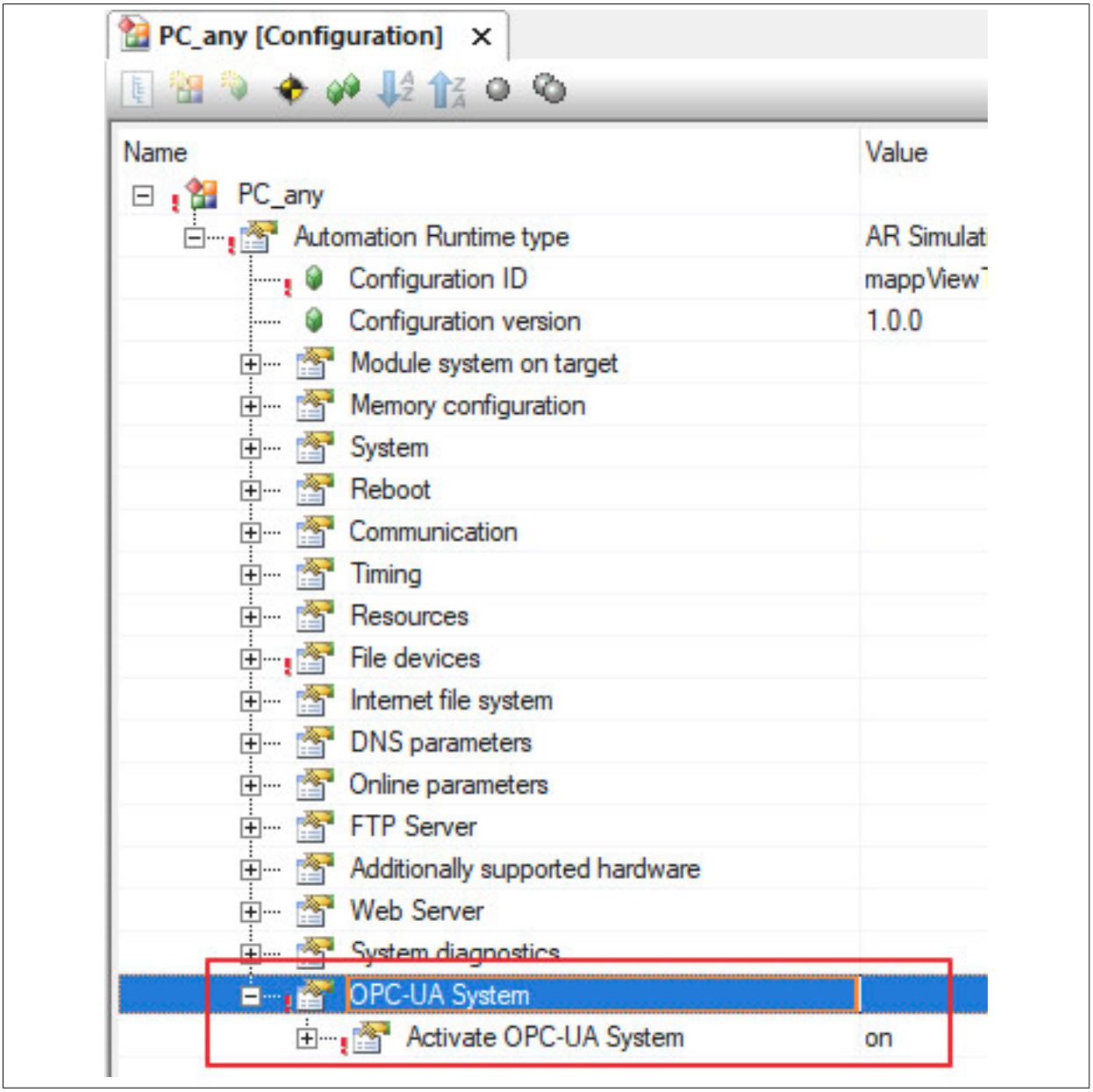
10.5.3.122 -1064303082: OPC UA Server nicht verfügbar

Beschreibung:

Der OPC UA Server ist nicht verfügbar.

Ursache / Abhilfe:

- Aktivieren des OPC UA Servers in der Konfiguration des Zielsystems:



10.5.3.123 -1064303084: Einheiten stimmen nicht überein

Beschreibung:

Die Einheiten zwischen den verwendeten Einheitensystemen stimmen nicht überein. Die Einheiten können nicht konvertiert werden.

Ursache / Abhilfe:

- Überprüfen der verwendeten Engineering-Einheit im OPC-UA Default View Konfiguration
- Überprüfen der verwendeten Einheiten in der MpComUnit Konfiguration

10.5.3.124 -1064303085: Ungültige Fallback-Einheit

Beschreibung:

Es wurde eine ungültige Prozessvariable {2:Name} für das Limit konfiguriert.

Zusätzliche Information

- {2:Name}: Betroffene Prozessvariable

Ursache / Abhilfe:

- Überprüfen der verwendeten Prozessvariable in der Konfiguration. Es dürfen nur numerische Prozessvariablen verwendet werden. STRING-Variablen können beispielsweise nicht verwendet werden.

10.5.3.125 -1064303086: Ungültige Fallback-Einheit

Beschreibung:

Es wurde eine ungültige Fallback-Einheit ({3:Namespace}|{2:Unit-ID}) im Abschnitt "Display unit" in der Konfiguration definiert.

Zusätzliche Information

- {3:Namespace}: Verwendeter Namespace
- {2:Unit-ID}: Einheiten-Code

Ursache / Abhilfe:

- Überprüfen der verwendeten Einheiten in der Konfiguration
- Überprüfen der verwendeten Engineering-Einheit im OPC-UA Default View Konfiguration

10.5.3.126 -1064303087: Ungültiger Datenpunkt

Beschreibung:

Es wurde ein ungültiger Datenpunkt im Abschnitt "Display unit".

Ursache / Abhilfe:

- Kontrollieren der verwendeten Prozessvariablen in der Konfiguration. Es dürfen nur numerische Prozessvariablen verwendet werden. STRING-Variablen können beispielsweise nicht verwendet werden.

10.5.3.127 1083180560: Konvertierung nicht durchführbar

Beschreibung:

Die Einheiten-Konvertierung kann nicht durchgeführt werden, da keine Einheit für den Datenpunkt definiert wurde.

Ursache / Abhilfe:

- Angeben einer Einheit für den verwendeten Datenpunkt in der MpComUnit Konfiguration
- Überprüfen der verwendeten Engineering-Einheit im OPC-UA Default View Konfiguration

10.5.3.128 -1064303089: Einheiten nicht angegeben

Beschreibung:

Für den verwendeten Datenpunkt wurde keine Einheit definiert.

Ursache / Abhilfe:

- Angeben einer Einheit für den verwendeten Datenpunkt in der MpComUnit Konfiguration
- Überprüfen der verwendeten Engineering-Einheit im OPC-UA Default View Konfiguration

10.5.3.129 -1064303090: Einheiten stimmen nicht überein

Beschreibung:

Die verwendeten Einheiten stimmen nicht überein und können somit nicht konvertiert werden.

Ursache / Abhilfe:

- Überprüfen der verwendeten Einheit in der MpComUnit Konfiguration
- Überprüfen der verwendeten Engineering-Einheit im OPC-UA Default View Konfiguration

10.5.3.130 -2138045423: Taskname zu lang

Beschreibung:

Der verwendete Taskname "{2:TaskName}" ist zu lang und wird deshalb auf "{3:NewTaskName}" gekürzt.

Zusätzliche Information

- {2:TaskName}: Name des Tasks welcher zu lange ist
- {3:NewTaskName}: Neuer gekürzter Taskname

Ursache / Abhilfe:

- Werden Prozessvariablen aus diesem Task referenziert, mit beispielsweise MpRecipeRegPar oder MpDataRegPar, so gilt es zu beachten, dass bei der Angabe des Tasknamen der gekürze Taskname verwendet werden muss. ("ShortName:PVName")
- Anpassen des Tasknamens, damit dieser nicht gekürzt werden muss.

10.5.3.131 -1064303598: Password nicht lesbar

Beschreibung:

Das Passwort kann nicht von {2:Path} gelesen werden.

Zusätzliche Information

- {2:Path}: Angabe des betroffenen Passwort-Parameters

Ursache / Abhilfe:

- Wird ein Passwort in einer Konfiguration, wie beispielsweise in der MpDataRecorder Konfiguration, verschlüsselt angegeben, so kann das verschlüsselte Passwort im Nachhinein manipuliert werden. Es kann beispielsweise ein Zeichen hinzugefügt oder gelöscht werden. Wird nun die dementsprechende Komponente, wie beispielsweise mapp Data: Data Logging, aktiviert und die Konfiguration wird gelesen, so wird dieser Fehler ausgelöst. Das Passwort ist nicht mehr lesbar und muss wieder angepasst werden. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.132 -1064303597: Password ist ungültig

Beschreibung:

Das Passwort ist ungültig. Kommando kann nicht ausgeführt werden.

Ursache / Abhilfe:

- In einigen Komponenten können in der Konfiguration Passwörter verschlüsselt angegeben werden, wie beispielsweise in der MpReportCore Konfiguration. Nach der Verschlüsselung kann der Benutzer das Passwort manipulieren, wie beispielsweise ein Zeichen hinzufügen oder löschen. Das Passwort ist daraufhin ungültig. Wird nun ein Report erstellt, so erscheint dieser Fehler, da dass Passwort nicht gelesen werden kann und somit der Report nicht verschlüsselt werden kann. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.133 -1064303596: Allokierung des Speichers fehlgeschlagen

Beschreibung:

Es konnte nicht genügend Speicher allokiert werden.

Ursache / Abhilfe:

- Das Zielsystem hat nicht mehr genügend Speicher zur Verfügung
- Überprüfen der Speichereinstellungen des Zielsystems

10.5.3.134 -1064303106: Verschlüsselung definiert

Beschreibung:

Es wurde bereits eine inkompatible Verschlüsselung für die Eigenschaft definiert.

Ursache / Abhilfe:

- Überprüfen der betroffenen Eigenschaft. Neu definieren der Verschlüsselung. Für mehr Information siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.135 -1064303105: Verschlüsselung nicht unterstützt

Beschreibung:

Der Verschlüsselungs-Typ oder die Schlüssel-Länge wird nicht untersützt.

Ursache / Abhilfe:

- Überprüfen der betroffenen Eigeschaft, gegebenfalls Angabe eines kürzeren Passworts. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.136 -1064303104: Verschlüsselung ungültig

Beschreibung:

Die Verschlüsselung ist ungültig.

Ursache / Abhilfe:

- Überprüfen der betroffenen Eigenschaft. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.137 -1064303103: Verschlüsselungs-Version ungültig

Beschreibung:

Die verwendete Verschlüsselungs-Version ist ungültig.

Ursache / Abhilfe:

- Aktualisieren der verwendeten mapp Technology Version. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.138 -1064303102: Eigenschaft nicht sicher

Beschreibung:

Die Eigenschaft wurde nicht verschlüsselt und ist somit nicht sicher.

Ursache / Abhilfe:

- Neu verschlüsseln der betroffenen Eigenschaft. Für mehr Information zur Verschlüsselung siehe Abschnitt "Passwortverschlüsselung in der Konfiguration" in Rahmenbedingungen.

10.5.3.139 -1064303079: Einheitenumrechnung ungültig

Beschreibung:

Die Einheitenumrechnung ist ungültig.

Ursache / Abhilfe:

- Überprüfen der Einheitenumrechnung. Für mehr Information siehe Abschnitt Unit System.

10.5.3.140 -1064303078: Division durch Null

Beschreibung:

Die Einheitenumrechnung kann nicht angewandt werden, da eine Division durch Null definiert wurde.

Ursache / Abhilfe:

- Überprüfen der Einheitenumrechnung. Für mehr Information siehe Abschnitt Unit System.

10.5.3.141 -1064303077: Variable ungültig

Beschreibung:

Die Einheitenumrechnung kann nicht angewandt werden, da eine Prozessvariable ungültig ist bzw. fehlt.

Ursache / Abhilfe:

- Überprüfen der Einheitenumrechnung. Für mehr Information siehe Abschnitt Unit System.
- Überprüfen der verwendeten Prozessvariablen.

10.5.3.142 -1064303077: Einheitenumrechnung fehlt

Beschreibung:

Die angegebene Einheitenumrechnung ist nicht verfügbar.

Ursache / Abhilfe:

- Überprüfen der Einheitenumrechnung. Für mehr Information siehe Abschnitt Unit System.
- Überprüfen der angegebenen Einheitenumrechnung in der MpComUnit Konfiguration.