ControlLogix-Ethernet-Treiber von Allen-Bradley

© 2019 PTC Inc. Alle Rechte vorbehalten.

Inhaltsverzeichnis

ControlLogix-Ethernet-Treiber von Allen-Bradley 1
Inhaltsverzeichnis 2
ControlLogix-Ethernet-Treiber von Allen-Bradley 10
Übersicht 11
Setup 12
Allen-Bradley ControlLogix Konfigurations-API-Beispiel 14
Kanaleigenschaften - Allgemein 15
Kanaleigenschaften - Ethernet-Kommunikation 16
Kanaleigenschaften - Schreiboptimierungen 16
Kanaleigenschaften - Erweitert 17
Geräteeigenschaften - Allgemein 18
Betriebsmodus 18
Geräteeigenschaften - Scan-Modus 19
Geräteeigenschaften - Zeitvorgabe 19
Geräteeigenschaften - Automatische Herabstufung 20
Geräteeigenschaften - Tag-Generierung 21
Geräteeigenschaften - Logix-Kommunikationsparameter 23
Geräteeigenschaften - Logix-Optionen 24
Geräteeigenschaften - Logix-Datenbankeinstellungen 25
Geräteeigenschaften - Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway 27
Blockschreibvorgänge 27
Geräteeigenschaften - SLC 500-Steckplatzkonfiguration 28
Konfigurations-API - Steckplatzkonfiguration 29
Geräteeigenschaften - Redundanz 30
Richtlinien für modulare E/A-Auswahl für SLC 500 30
Leistungsoptimierungen 32
Kommunikation optimieren 32
Anwendung optimieren 34
Leistungsstatistik und -optimierung 35
Beispiel für Leistungsoptimierung 36
Datentypbeschreibung 47
Bedingungen für Standard-Datentyp 48
Adressbeschreibungen 48
Logix-Adressierung 49
MicroLogix-Adressierung 50
Feste E/A-Adressierung für SLC 500 52
Modulare E/A-Adressierung für SLC 500 53
Adressierung der PLC-5-Reihe 54
Auf Logix-Tags basierte Adressierung 56
Adressformate 57
Tag-Umfang 58

Interne Tags 59
Vordefinierte Ausdrucks-Tags 59
Adressieren unteilbarer Datentypen 59
Strukturdatentypen adressieren 61
STRING-Datentyp adressieren 61
Reihenfolge von Logix-Array-Daten 62
Erweiterte Logix-Adressierung 63
Erweiterte Adressierung: BOOL 64
Erweiterte Adressierung: SINT 65
Erweiterte Adressierung: INT 67
Erweiterte Adressierung: DINT 69
Erweiterte Adressierung: LINT 72
Erweiterte Adressierung: REAL 73
Erweiterte Adressierung: USINT 75
Erweiterte Adressierung: UINT 77
Erweiterte Adressierung: UDINT 79
Erweiterte Adressierung: ULINT 81
Erweiterte Adressierung: LREAL 82
Dateiauflistung 83
Ausgabedateien 84
Eingabedateien 87
Statusdateien 90
Binäre Dateien 91
Zeitgeberdateien 92
Zählerdateien 92
Steuerdateien 93
Ganzzahldateien 94
Float-Dateien 94
ASCII-Dateien 95
String-Dateien 96
BCD-Dateien 96
Long-Dateien 97
MicroLogix-PID-Dateien 97
PID-Dateien 98
MicroLogix-Meldungsdateien 100
Meldungsdateien 101
Blocktransferdateien 102
Funktionsdateien 102
HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler) 103
RTC-Datei (Real Time Clock, Echtzeituhr) 104
Kommunikationsstatusdatei für Kanal 0 (CS0, Channel 0 Communication Status File) 105
Kommunikationsstatusdatei für Kanal 1 (CS1, Channel 1 Communication Status File) 105
IOS-Datei (I/O Module Status, E/A-Modulstatus) 106
Automatische Tag-Datenbankgenerierung 106

Tag-Hierarchie ... 106
Namenskonvertierungen von Controller in Server ... 109
Automatische Tag-Datenbankgenerierung wird vorbereitet ... 110
Fehlercodes ... 112
Kapselungsfehlercodes ... 112
CIP-Fehlercodes ... 112
0x0001 Erweiterte Fehlercodes ... 113
0x001F Erweiterte Fehlercodes ... 114
0x00FF Erweiterte Fehlercodes ... 114
Ereignisprotokollmeldungen ... 115
Die folgenden Fehler sind beim Hochladen des Controller-Projekts vom Gerät aufgetreten. Das symbolische Protokoll wird stattdessen verwendet. ... 115
Ungültiges oder fehlerhaftes Controller-Projekt wurde beim Synchronisieren festgestellt. Die Synchronisierung wird in Kürze erneut versucht. ... 115
Während der Synchronisierung wurde ein Herunterladen des Projekts festgestellt. Die Synchronisierung wird in Kürze erneut versucht. ... 115
Datenbankfehler. Datentyp für Referenz-Tag ist unbekannt. Datentyp für Alias-Tag wird auf den Standardwert gesetzt. | Referenz-Tag = '<Tag>', Alias-Tag = '<Tag>', Standard-Datentyp = '<Typ>'. ... 115
Datenbankfehler. Datentyp für Mitglied in Tag-Importdatei wurde nicht gefunden. Datentyp wird auf Standardwert gesetzt. | Datentyp für Mitglied = '<Typ>', UDT = '<Typ>', Standard-Datentyp '<Typ>'. ... 116
Datenbankfehler. Datentyp wurde in Tag-Importdatei nicht gefunden. Tag nicht hinzugefügt. | Datentyp = '<Typ>', Tag-Name = '<Tag>'. ... 116
Datenbankfehler. Fehler beim Verarbeiten des Alias-Tags. Tag wurde nicht hinzugefügt. | Alias-Tag = '<Tag>'. ... 116
Datenbankfehler. Kapselungsfehler beim Registrieren der Sitzungsanforderung. | Kapselungsfehler = <Code>. ... 117
Datenbankfehler. Framing-Fehler beim Registrieren der Sitzungsanforderung. ... 117
Datenbankfehler. Kapselungsfehler beim Weiterleiten der Anforderung zum Öffnen. | Kapselungsfehler = <Code>. ... 117
Datenbankfehler. Framing-Fehler beim Weiterleiten der offenen Anforderung. ... 117
Datenbankfehler. Fehler beim Weiterleiten der offenen Anforderung. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. ... 117
Datenbankfehler. Kapselungsfehler beim Hochladen der Projektinformationen. | Kapselungsfehler = <Code>. ... 117
Datenbankfehler. Fehler beim Hochladen der Projektinformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. ... 118
Datenbankfehler. Framing-Fehler beim Hochladen der Projektinformationen. ... 118
Datenbankfehler. Interner Fehler aufgetreten. ... 118
Datenbankfehler. Kapselungsfehler beim Hochladen der Programminformationen. | Programmname = '<Name>', Kapselungsfehler = <Code>. ... 118
Datenbankfehler. Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. ... 119
Datenbankfehler. Framing-Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>'. ... 119
Datenbankfehler. CIP-Datentyp für Tag konnte nicht aufgelöst werden. Standardtyp wird verwendet. | CIP-Datentyp = <Typ>, Tag-Name = '<Tag>', Standard-Datentyp = '<Typ>'. ... 119
Kapselungsfehler beim Hochladen der Projektinformationen. | Kapselungsfehler = <Code>. ... 120
Fehler beim Hochladen der Projektinformationen. | CIP-Fehler = <Code>, erweiterter Fehler = 120

<Code>. .....
Framing-Fehler beim Hochladen der Projektinformationen. 120
Kapselungsfehler beim Hochladen der Programminformationen. | Programmname = '<Name>', Kapselungsfehler = <Code>. 121
Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 121
Framing-Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>'. 121
Kapselungsfehler beim Hochladen der Controller-Programminformationen. | Kapselungsfehler = <Code>. 121
Fehler beim Hochladen der Controller-Programminformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 121
Framing-Fehler beim Hochladen der Controller-Programminformationen. 121
Timeout von CIP-Verbindung, während Projektinformationen hochgeladen werden. 121
Datenbankfehler. Timeout von CIP-Verbindung, während Projektinformationen hochgeladen werden. 122
Datenbankfehler. Keine Verbindungen mehr verfügbar zum Weiterleiten offener Anforderungen. 122
Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Code>'. 122
Controller wird nicht unterstützt. | Händler-ID = <ID>, Produkttyp = <Typ>, Produktcode = <Code>, Produktname = '<Name>'. 122
Der vom Gerät empfangene Frame enthält Fehler. 122
Schreibanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Tag-Adresse = '<Adresse>'. 122
Leseanforderung für Tag ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Tag-Adresse = '<Adresse>'. 123
Block-Leseanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 123
Block-Leseanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Blockgröße = <Anzahl> (Byte), Blockname = '<Name>'. 124
In Tag kann nicht geschrieben werden. | Tag-Adresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 124
Tag kann nicht gelesen werden. | Tag-Adresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 124
Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 124
Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Byte), Tag-Name = '<Tag>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 125
In Tag kann nicht geschrieben werden. Controller-Tag-Datentyp ist unbekannt. | Tag-Adresse = '<Adresse>', Datentyp = <Typ>. 125
Tag kann nicht gelesen werden. Controller-Tag-Datentyp ist unbekannt. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = <Typ>. 125
Block kann nicht gelesen werden. Controller-Tag-Datentyp ist unbekannt. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = <Typ>. 125
In Tag kann nicht geschrieben werden. Datentyp wird nicht unterstützt. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 126
Tag kann nicht gelesen werden. Datentyp wird nicht unterstützt. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 126
Block kann nicht gelesen werden. Datentyp wird nicht unterstützt. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = '<Typ>'. 126
In Tag kann nicht geschrieben werden. Datentyp für dieses Tag ist unzulässig. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 126

Tag kann nicht gelesen werden. Datentyp für dieses Tag ist unzulässig. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 127
Block kann nicht gelesen werden. Datentyp für diesen Block ist unzulässig. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = '<Typ>'. 127
In Tag kann nicht geschrieben werden. Tag unterstützt keine Arrays mit mehreren Elementen. | Tag-Adresse = '<Adresse>'. 127
Tag kann nicht gelesen werden. Tag unterstützt keine Arrays mit mehreren Elementen. Tag deaktiviert. | Tag-Adresse = '<Adresse>'. 128
Block kann nicht gelesen werden. Block unterstützt keine Arrays mit mehreren Elementen. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 128
In Tag kann nicht geschrieben werden. Keine Größenübereinstimmung bei nativem Tag. | Tag-Adresse = '<Adresse>'. 128
Tag kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Tag-Adresse = '<Adresse>'. 128
Block kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 129
Block kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Blockgröße = <Anzahl> (Byte), Blockname = '<Name>'. 129
In Tag kann nicht geschrieben werden. | Tag-Adresse = '<Adresse>'. 129
Tag kann nicht gelesen werden. Tag deaktiviert. | Tag-Adresse = '<Adresse>'. 130
Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 130
Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Byte), Tag-Name = '<Tag>'. 131
Fehler während Anfrage an Gerät. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 131
Kapselungsfehler während Anfrage an Gerät. | Kapselungsfehler = <Code>. 131
Speicherplatz für Tag konnte nicht zugeordnet werden. | Tag-Adresse = '<Adresse>'. 132
Block kann nicht gelesen werden. Empfangener Frame enthält Fehler. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>'. 132
Funktionsdatei kann nicht von Gerät gelesen werden. Empfangener Frame enthält Fehler. | Funktionsdatei = '<Name>'. 132
Block kann nicht gelesen werden. Tags deaktiviert. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>. 132
Funktionsdatei kann nicht von Gerät gelesen werden. Tags deaktiviert. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>. 133
In Adresse kann nicht geschrieben werden. Empfangener Frame enthält Fehler. | Adresse = '<Adresse>'. 133
In Funktionsdatei kann nicht geschrieben werden. Empfangener Frame enthält Fehler. | Funktionsdatei = '<Name>'. 133
Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>. 133
Funktionsdatei kann nicht gelesen werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>. 134
Block kann nicht gelesen werden. Tags deaktiviert. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>. 134
Funktionsdatei kann nicht gelesen werden. Tags deaktiviert. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 135
In Adresse kann nicht geschrieben werden. | Adresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>. 135
In Funktionsdatei kann nicht geschrieben werden. | Funktionsdatei = '<Name>', DF1-Status = 135

<Code>, erweiterter Status = <Code>. .....
Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>. 136
Funktionsdatei kann nicht gelesen werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 136
In Adresse kann nicht geschrieben werden. | Adresse = '<Adresse>', DF1-Status = <Code>. 137
In Funktionsdatei kann nicht geschrieben werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 137
Tag kann nicht gelesen werden. Interner Speicher ist ungültig. | Tag-Adresse = '<Adresse>'. 137
Tag kann nicht gelesen werden. Datentyp für dieses Tag ist unzulässig. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 138
Block kann nicht gelesen werden. Interner Speicher ist ungültig. Tag deaktiviert. | Tag-Adresse = '<Adresse>'. 138
Block kann nicht gelesen werden. Interner Speicher ist ungültig. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 138
In Adresse kann nicht geschrieben werden. Interner Speicher ist ungültig. | Tag-Adresse = '<Adresse>'. 138
Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 138
Gerät antwortet nicht. Lokaler Knoten hat mit Fehler geantwortet. | DF1-Status = <Code>. 138
In Funktionsdatei kann nicht geschrieben werden. Lokaler Knoten hat mit Fehler geantwortet. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 139
In Adresse kann nicht geschrieben werden. Lokaler Knoten hat mit Fehler geantwortet. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 139
Unerwarteter Offset für Tag festgestellt. Tag verwendet stattdessen symbolisches Protokoll. | Tag-Adresse = '<Adresse>'. 139
Unerwarteter Offset für Tag festgestellt. | Tag-Adresse = '<Adresse>'. 139
Unerwarteter Offset bzw. Spanne für Tag festgestellt. | Tag-Adresse = '<Adresse>'. 140
Herunterladen von Projekt in Bearbeitung oder kein Projekt vorhanden. 140
Herunterladen von Projekt abgeschlossen. 140
Online-Bearbeitung des Projekts festgestellt. Zurzeit wird symbolische Adressierung verwendet. 140
Offline-Bearbeitung des Projekts festgestellt. Zurzeit wird symbolische Adressierung verwendet. 140
Die folgenden Fehler sind beim Hochladen des Controller-Projekts vom Gerät aufgetreten. Das symbolische Protokoll wird stattdessen verwendet. 140
ID für Gerät kann nicht abgerufen werden. Alle Tags verwenden stattdessen das symbolische Protokoll. | Kapselungsfehler = <Code>. 140
ID für Gerät kann nicht abgerufen werden. Alle Tags verwenden stattdessen das symbolische Protokoll. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 140
ID für Gerät kann nicht abgerufen werden. Empfangener Frame enthält Fehler. Alle Tags verwenden stattdessen das symbolische Protokoll. 141
Angeforderte CIP-Verbindungsgröße wird von diesem Gerät nicht unterstützt. Es wird automatisch auf die maximale Größe ausgewichen. | Angeforderte Größe = <Anzahl> (Byte), max. Größe = <Anzahl> (Byte). 141
Der Dateiname für den Tag-Import ist ungültig. Dateipfade sind nicht erlaubt. 141
Lese-/Schreibanfragen an Gerät beendet. Aktualisierung logischer Adressen vom Geräteprojekt läuft. 142
Lese-/Schreibanfragen an Gerät wurden fortgesetzt. Aktualisierung logischer Adressen vom Geräteprojekt abgeschlossen. Zurzeit wird logische Adressierung verwendet. 142
Datenbankstatus. Tags ohne Alias werden importiert. 142
Datenbankstatus. Tags mit Alias werden importiert. 142

Datenbankstatus. Tag-Projekte werden erstellt. Bitte warten. | Anzahl der Tag-Projekte = <Anzahl>. 142
Datenbankfehler. Tag umbenannt, da maximale Zeichenlänge überschritten wurde. | Tag-Name = '<Tag>', max. Länge = <Anzahl>, neuer Tag-Name = '<Tag>'. 142
Datenbankfehler. Array-Tags umbenannt, da sie maximale Zeichenlänge überschreiten. | Array-Tags = '<Tags>', max. Länge = <Anzahl>, neue Array-Tags = '<Tags>'. 142
Datenbankfehler. Programmgruppenname überschreitet maximale Zeichenlänge. Programmgruppe umbenannt. | Gruppenname = '<Name>', max. Länge = <Anzahl>, neuer Gruppenname = '<Name>'. 142
Datenbankstatus. Controller-Projekt wird abgerufen. 143
Datenbankstatus. | Programmanzahl = <Anzahl>, Datentypanzahl = <Anzahl>, Anzahl importierter Tags = <Anzahl>. 143
Datenbankstatus. OPC-Tags werden generiert. 143
Zu wenig Speicherressourcen. 143
Unbekannter Fehler aufgetreten. 143
Datenbankstatus. Tags werden aus L5X-Datei importiert. | Schemarevision = '<Wert>', Softwarerevision = '<Wert>'. 143
Details. | IP = '<Adresse>', Händler-ID = <Händler>, Produkttyp = <Typ>, Produktcode = <Code>, Revision= <Wert>, Produktname = '<Name>', Produkt-Seriennummer = <Anzahl>. 143
Verstrichene Zeit = <Anzahl> (Sekunden). 143
Symbolisch - Lesevorgänge von Gerät = <Anzahl>. 143
Symbolisch, Array-Block - Lesevorgänge von Gerät = <Anzahl>. 143
Symbolisch, Array-Block - Lesevorgänge von Cache = <Anzahl>. 143
Symbolinstanz nicht blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Symbolinstanz nicht blockierend, Array-Block - Lesevorgänge von Gerät = <Anzahl>. 144
Symbolinstanz nicht blockierend, Array-Block - Lesevorgänge von Cache = <Anzahl>. 144
Symbolinstanz blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Symbolinstanz blockierend - Lesevorgänge von Cache = <Anzahl>. 144
Physisch nicht blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Physisch nicht blockierend, Array-Block - Lesevorgänge von Gerät = <Anzahl>. 144
Physisch nicht blockierend, Array-Block - Lesevorgänge von Cache = <Anzahl>. 144
Physisch blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Physisch blockierend - Lesevorgänge von Cache = <Anzahl>. 144
Lesevorgänge für Tags = <Anzahl>. 144
Pakete gesendet = <Anzahl>. 144
Pakete empfangen = <Anzahl>. 145
Initialisierungstransaktionen = <Anzahl>. 145
Lese-/Schreibtransaktionen = <Anzahl>. 145
Pakete durchschnittl. gesendet/Sekunde = <Anzahl>. 145
Pakete durchschnittl. empfangen/Sekunde = <Anzahl>. 145
Durchschnittl. Lesevorgänge/Sekunde für Tag = <Anzahl>. 145
Durchschnittl. Tags/Transaktion = <Anzahl>. 145
---------- 145
%s | GERÄTESTATISTIK 145
Durchschnittliche Verarbeitungszeit für Gerät = <Anzahl> (Millisekunden) 145
%s | KANALSTATISTIK 145
TREIBERSTATISTIK 145

Importieren von Geräte-Tag abgebrochen. 146
Importdatei '%s' nicht gefunden unter Pfad '%s'. 146
Beim Abrufen des Controller-Projekts sind Fehler aufgetreten. 146
Interner Treiberfehler aufgetreten. 146
Ungültiges oder fehlerhaftes Controller-Projekt wurde beim Synchronisieren festgestellt. Versuchen Sie es später erneut. 146
Während der Synchronisierung wurde ein Herunterladen des Projekts festgestellt. Versuchen Sie es später erneut. 146
Zu wenig Speicherressourcen. 146
L5K-Datei ist ungültig oder fehlerhaft. 146
Unbekannter Fehler aufgetreten. 146
Datenbankfehler. PLC5/SLC/MicroLogix-Geräte unterstützen diese Funktion nicht. 146
L5X-Datei ist ungültig oder fehlerhaft. 146
Import-Datei '<empty>' nicht gefunden unter Pfad '<empty>'. 146
Import-Datei '%s' nicht gefunden unter Pfad '<empty>'. 147
Import-Datei '<empty>' nicht gefunden unter Pfad '%s'. 147
Fehler bei XML-Element in Post-Schema-Validierung. Modell unterstützt kein Importieren der Tags von Gerät. Verwenden Sie ein alternatives Element. | XML-Element = '{<Namespace>}<Element>', nicht unterstütztes Modell = '<Modell>', alternatives XML-Element = '{<Namespace>}<Element>'. 147
Wert wird für ein XML-Element in diesem Modell nicht unterstützt. Ein neuer Wert wird automatisch festgelegt. | Wert = '<Wert>', XML-Element = '{<Namespace>}<Element>', Modell = '<Modell>', neuer Wert = '<Wert>'. 147

Anhänge 148

Anhang A - Kanaleigenschaften 148
Anhang B - Geräteeigenschaften 148
Anhang C - Tag-Eigenschaften 149
Logix-Geräte-IDs 150
Ethernet-Geräte-ID für CompactLogix 5300 150
1761-NET-ENI-Setup 151
Data Highway Plus-Gateway-Setup 152
ControlNet™-Gateway-Setup 153
EtherNet/IP-Gateway-Setup 154
Serial Gateway-Setup 155
MicroLogix 1100-Setup 156
Kommunikations-Routing 156
Verbindungspfadspezifikation 157
Routing-Beispiele 158
Protokollmodus auswählen 161
Änderung im Controller-Projekt erkennen 163
Hinweise zur SoftLogix 5800-Verbindung 164

Index 166

ControlLogix-Ethernet-Treiber von Allen-Bradley

Hilfe-Version 1.167

INHALT

Übersicht
Was ist ControlLogix-Ethernet-Treiber von Allen-Bradley?

Setup
Wie konfiguriere ich ein Gerät für die Verwendung mit diesem Treiber?

Quickstart
Wie konfiguriere ich einen Kanal und ein Gerät mithilfe der Konfigurations-API?

Kommunikations-Routing
Wie kommuniziere ich mit einem ControlLogix 5000-Remote-Prozessor oder einem 1756-DHRIO/1756-CNB-Schnittstellenmodul?

Leistungsoptimierungen
Wie erziele ich die beste Leistung mit ControlLogix-Ethernet-Treiber von Allen-Bradley?

Datentypbeschreibung
Welche Datentypen unterstützt dieser Treiber?

Adressbeschreibungen
Wie adressiere ich ein Tag auf einem ControlLogix-Ethernet-Gerät von Allen-Bradley?

Automatische Tag-Datenbankgenerierung
Wie kann ich Tags einfach für ControlLogix-Ethernet-Treiber von Allen-Bradley konfigurieren?

Ereignisprotokollmeldungen
Welche Meldungen erzeugt der Treiber?

Fehlercodes
Welche Fehlercodes gibt es für das ControlLogix-Ethernet-Gerät von Allen-Bradley?

Anhänge
Wo kann ich zusätzliche Informationen zu ControlLogix-Ethernet-Treiber von Allen-Bradley finden?

Übersicht

ControlLogix-Ethernet-Treiber von Allen-Bradley bietet eine einfache und zuverlässige Möglichkeit, ControlLogix-Ethernet-Controller von Allen-Bradley mit OPC-Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und zahlreichen benutzerdefinierten Anwendungen, zu verbinden.

Unterstützte Controller von Allen-Bradley

ControlLogix® 5500-Reihe

Die Kommunikation mit ControlLogix kann über ein EtherNet/IP-Kommunikationsmodul für Ethernet-Kommunikation oder über ein 1761-NET-ENI-Modul für Ethernet-to-Serial-Kommunikation über den seriellen Port des Controllers erfolgen.

CompactLogix™ 5300-Reihe

Für die Ethernet-Kommunikation mit CompactLogix wird ein Prozessor mit integriertem EtherNet/IP-Port (z.B. 1769-L35E) benötigt. Für die Kommunikation mit CompactLogix wird andernfalls ein 1761-NET-ENI-Modul für Ethernet-to-Serial-Kommunikation über den seriellen Port des Controllers benötigt.

FlexLogix 5400-Reihe

Die Kommunikation mit FlexLogix kann über eine untergeordnete 1788-ENBT-Karte für Ethernet-Kommunikation oder über ein 1761-NET-ENI-Modul für Ethernet-to-Serial-Kommunikation über den seriellen Port des Controllers erfolgen.

SoftLogix 5800

Der Treiber unterstützt den Controller der SoftLogix 5800-Reihe von Allen-Bradley und benötigt eine Ethernet-Karte im SoftLogix-PC.

Data Highway Plus-Gateway

Der Treiber unterstützt die PLC-5- und SLC 500-Reihe mit einer Data Highway Plus-Schnittstelle. Dies erfolgt über ein DH+-Gateway und erfordert eine der zuvor erwähnten SPS, ein EtherNet/IP-Kommunikationsmodul und ein 1756-DHRIO-Schnittstellenmodul (beide im ControlLogix-Rack eingebaut).

ControlNet-Gateway

Der Treiber unterstützt die PLC-5C-Reihe. Dies erfolgt über ein ControlNet-Gateway und erfordert die zuvor erwähnte SPS, ein EtherNet/IP-Kommunikationsmodul und ein 1756-CNB/CNBR-Schnittstellenmodul (beide im ControlLogix-Rack eingebaut).

1761-NET-ENI

Der Treiber unterstützt die Kommunikation mit dem 1761-NET-ENI-Gerät. Das ENI-Gerät sorgt für zusätzliche Flexibilität des Gerätenetzwerks und der Gerätekommunikation, indem eine Ethernet-to-Serial-Schnittstelle für Vollduplex-DF1-Controller und -Logix-Controller bereitgestellt wird. Zusammen mit dem ENI-Gerät unterstützt dieser Treiber Folgendes:

- ControlLogix 5500-Reihe*
- CompactLogix 5300-Reihe*
- FlexLogix 5400-Reihe*
- MicroLogix-Reihe
- SLC 500 - Fester E/A-Prozessor
- SLC 500 - Modulare E/A-Reihe
- PLC-5-Reihe

*Diese Modelle erfordern 1761-NET-ENI-Reihe B oder höher.

MicroLogix 1100

Der Treiber unterstützt die Kommunikation mit MicroLogix 1100 (CH1 Ethernet) über EtherNet/IP.

ControlLogix ist eine eingetragene Marke von Allen-Bradley Company, LLC.
CompactLogix ist eine Marke von Rockwell Automation, Inc.

Alle Marken sind Eigentum ihrer jeweiligen Inhaber.

Setup

Unterstützte Geräte

Gerätfamilie	Kommunikation
Prozessoren für ControlLogix 5550/5553/5555/5561/5562/5563/5564/5565/5571/5572/5573/5574/5575/5580	Über 1756-ENBT/ENET/EN2F/EN2T/EN2TR/EN3TR/E-WEB/EN2TXT-Ethernet-Modul Über Serial Gateway Über 1761-NET-ENI-Reihe B oder höher mit Kanal 0 (seriell)
CompactLogix 5320 / 5323 / 5330 / 5331 / 5332 / 5335 / 5343 / 5345 / 5370 / 5380 / 5480	Integrierter Ethernet/IP-Port für Prozessoren mit E-Suffix* Über Serial Gateway Über 1761-NET-ENI-Reihe B oder höher mit Kanal 0 (seriell)
Prozessoren für FlexLogix 5433/5434	Über untergeordnete 1788-ENBT-Ethernet-Karte Über Serial Gateway Über 1761-NET-ENI-Reihe B oder höher mit Kanal 0 (seriell)
Prozessoren für SoftLogix 5810/5830/5860	Über SoftLogix-Ethernet/IP-Messaging-Modul Über Serial Gateway
MicroLogix 1000/1200/1500	Über 1761-NET-ENI Über EtherNet/IP-Gateway
MicroLogix 1100/1400	Über MicroLogix 1100/1400 mit Kanal 1 (Ethernet) Über 1761-NET-ENI Über EtherNet/IP-Gateway
SLC 500 - Fester E/A-Prozessor	Über 1761-NET-ENI Über EtherNet/IP-Gateway
Modulare E/A-Prozessoren (SLC 5/01, SLC 5/02, SLC 5/03, SLC 5/04, SLC 5/05) für SLC 500	Über DH+-Gateway** Über 1761-NET-ENI Über EtherNet/IP-Gateway
PLC-5-Reihe (ausschließlich PLC5/250-Reihe)	Über DH+-Gateway Über 1761-NET-ENI Über EtherNet/IP-Gateway
PLC-5/20C, PLC-5/40C, PLC-5/80C	Über ControlNet-Gateway Über 1761-NET-ENI

Gerätfamilie	Kommunikation
	Über EtherNet/IP-Gateway

*Beispielsweise 1769-L35E.
**Dieser Treiber unterstützt jede SPS der SLC 500-Reihe, die DH+ unterstützt oder eine Schnittstelle mit einem DH+-Netzwerk (z.B. das KF2-Schnittstellenmodul) bilden kann.

Firmware-Versionen

Gerätfamilie	Version
ControlLogix 5550 (1756-L1)	11.35-13.34
ControlLogix 5553 (1756-L53)	11.28
ControlLogix 5555 (1756-L55)	11.32-16.04
ControlLogix 5561 (1756-L61)	12.31-20.11
ControlLogix 5562 (1756-L62)	12.31-20.11
ControlLogix 5563 (1756-L63)	11.26-20.11
ControlLogix 5564 (1756-L64)	16.03-20.11
ControlLogix 5565 (1756-L65)	16.03-20.11
ControlLogix 5571 (1756-L71)	20.11 - 32.12
ControlLogix 5572 (1756-L72)	19.11 - 32.12
ControlLogix 5573 (1756-L73)	18.12 - 32.12
ControlLogix 5574 (1756-L74)	19.11 - 32.12
ControlLogix 5575 (1756-L75)	18.12 - 32.12
ControlLogix 5580 (1756-L8)	28.11 - 32.12
CompactLogix 5370 (1769-L1)	20.11 - 32.12
CompactLogix 5370 (1769-L2)	20.11 - 32.12
CompactLogix 5370 (1769-L3)	20.11 - 32.12
CompactLogix 5320 (1769-L20)	11.27-13.18
CompactLogix 5323 (1769-L23)	17.05-20.11
CompactLogix 5330 (1769-L30)	11.27-13.18
CompactLogix 5331 (1769-L31)	16.22-20.11
CompactLogix 5332 (1769-L32)	16.22-20.11
CompactLogix 5335 (1769-L35)	16.22-20.11
CompactLogix 5343 (1768-L43)	15.07-20.11
CompactLogix 5345 (1768-L45)	16.24-20.11
CompactLogix 5380 (5069-L3)	28.11 - 32.12
CompactLogix 5480 (5069-L4)	32.12
FlexLogix 5433 (1794-L33)	11.25-13.33
FlexLogix 5434 (1794-L34)	11.25-16.02
SoftLogix 5800 (1789-L60)	16.00-20.01
Serielle Kommunikation für ControlLogix, CompactLogix und FlexLogix	1761-NET-ENI-Reihe B oder höher oder Serial Gateway
MicroLogix 1100 (1763-L16AWA/BWA/BBB)	1.1

Kommunikationsprotokoll

Das Kommunikationsprotokoll ist EtherNet/IP (CIP über Ethernet) mit TCP/IP.

Die maximale Anzahl unterstützter Kanäle ist 1024.

Logix- und Gateway-Modelle

Logix- und Gateway-Modelle unterstützen Folgendes:

- Verbundenes Messaging
- Symbolische Lesevorgänge
- Symbolische Schreibvorgänge
- Symbol Instance Reads (Symbolinstanz-Lesevorgänge) (V21 oder höher)
- Physical (DMA) Reads (Physische DMA-Lesevorgänge) (V20 oder niedriger)
- Symbol Instance Writes (Symbolinstanz-Schreibvorgänge)

ENI-Modelle
ENI-Modelle unterstützen nicht verbundenes Messaging.

Allen-Bradley ControlLogix Konfigurations-API-Beispiel

Für eine Liste von Kanal- und Gerätedefinitionen sowie Aufzählungen greifen Sie mit dem REST-Client auf die folgenden Endpunkte zu.

Kanaldefinitionen

Endpunkt (GET):

```
https://<Host-Name_oder_IP>:<Port>/config/v1/doc/drivers/Allen-Brad-
ley%20ControlLogix%20Ethernet/channels
```

Gerätedefinitionen

Endpunkt (GET):

```
https://<Host-Name_oder_IP>:<Port>/config/v1/doc/drivers/Allen-Brad-
ley%20ControlLogix%20Ethernet/devices
```

Die folgenden API-Befehle sind mindestens erforderlich zum Erzeugen eines AllenBradley ControlLogix Ethernet-Kanals, -Geräts und -Tags.
Weitere Informationen zur allgemeinen Projektkonfiguration finden Sie in der Server-Hilfedatei.

Allen-Bradley ControlLogix Kanal erstellen

Endpunkt (POST):

```
https://<Host-Name_oder_IP>:<Port>/config/v1/project/channels
```

Textkörper:

```
{ "common.ALLTYPES_NAME": "MyChannel", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER":
"Allen-Bradley ControlLogix Ethernet" }
```

***Siehe auch**: Anhang A für eine Liste von Kanaleigenschaften.*

Allen-Bradley ControlLogix Gerät erstellen

Endpunkt (POST):

```
https://<Host-Name_oder_IP>:<Port>/config/v1/project/channels/MyChannel/devices
```

Textkörper:

```
{ "common.ALLTYPES_NAME": "MyDevice", "servermain.DEVICE_ID_STRING": "<IP>,0,1",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet",
"servermain.DEVICE_MODEL": <Modellaufzählung> }
```

Hinweis: Das Format des Werts für servermain.DEVICE_ID_STRING kann abhängig von der für servermain.DEVICE_MODEL angegebenen Modellaufzählung variieren. Das oben angezeigte Format der Geräte-ID-Zeichenfolge gilt für das Modell ControlLogix 5500.

***Siehe auch**: Anhang B für eine Liste von Geräteeigenschaften.*

Allen-Bradley ControlLogix Tags erstellen

Endpunkt (POST):

```
https://<Host-Name_oder_IP>:<Por-
t>/config/v1/project/channels/MyChannel/devices/MyDevice/tags
```

Textkörper:

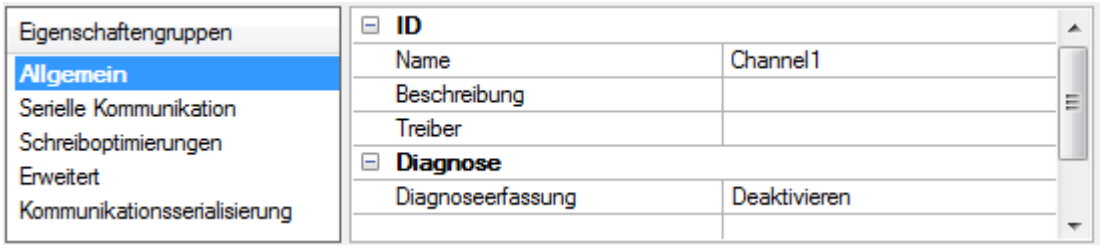
```
[ { "common.ALLTYPES_NAME": "MyTag1", "servermain.TAG_ADDRESS": "40001" } { "com-
mon.ALLTYPES_NAME": "MyTag2", "servermain.TAG_ADDRESS": "40002" } ]
```

***Siehe auch**: Anhang C für eine Liste von Tag-Eigenschaften.*

Weitere Informationen zum Konfigurieren von Tags und Tag-Gruppen über die Konfigurations-API finden Sie in der Server-Hilfe.

Kanaleigenschaften - Allgemein

Dieser Server unterstützt die Verwendung von gleichzeitigen Mehrfachkommunikationstreibern. Jedes Protokoll oder jeder Treiber, das/der in einem Serverprojekt verwendet wird, wird als Kanal bezeichnet. Ein Serverprojekt besteht unter Umständen aus vielen Kanälen mit demselben Kommunikationstreiber oder mit eindeutigen Kommunikationstreibern. Ein Kanal fungiert als grundlegender Baustein eines OPC-Links. Diese Gruppe wird verwendet, um allgemeine Kanaleigenschaften (wie z.B. die ID-Attribute und den Betriebsmodus) anzugeben.



Identifikation

Name: Benutzerdefinierte ID dieses Kanals. Bei jedem Serverprojekt muss jeder Kanalname eindeutig sein. Zwar können Namen bis zu 256 Zeichen lang sein, doch haben einige Client-Anwendungen beim Durchsuchen des Tag-Raums des OPC-Servers ein eingeschränktes Anzeigefenster. Der Kanalname ist ein Teil der OPC-Browserinformationen. Die Eigenschaft ist erforderlich, um einen Kanal zu erstellen.
Informationen über reservierte Zeichen finden Sie in der Serverhilfe unter „So benennen Sie Kanäle, Geräte, Tags und Tag-Gruppen richtig“.

Beschreibung: Benutzerdefinierte Informationen über diesen Kanal.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

Treiber: Ausgewähltes Protokoll/ausgewählter Treiber für diesen Kanal. Diese Eigenschaft gibt den Gerätetreiber an, der während der Kanalerstellung ausgewählt wurde. Es ist eine deaktivierte Einstellung in den Kanaleigenschaften. Die Eigenschaft ist erforderlich, um einen Kanal zu erstellen.
Hinweis: Beim Online-Vollzeitbetrieb des Servers können diese Eigenschaften jederzeit geändert werden. Dies schließt das Ändern des Kanalnamens ein, um zu verhindern, dass Clients Daten am Server registrieren. Wenn ein Client bereits ein Element vom Server abgerufen hat, bevor der Kanalname geändert wurde, sind die Elemente davon nicht beeinflusst. Wenn die Client-Anwendung das Element nach der Änderung des Kanalnamens freigibt und versucht, es mit dem alten Kanalnamen erneut abzurufen, wird das Element nicht akzeptiert. Unter Berücksichtigung dessen sollten keine Änderungen an den Eigenschaften erfolgen, sobald eine

große Client-Anwendung entwickelt wurde. Verwenden Sie den Benutzermanager, um zu verhindern, dass Operatoren Eigenschaften ändern, und um Zugriffsrechte auf Serverfunktionen zu beschränken.

Diagnose

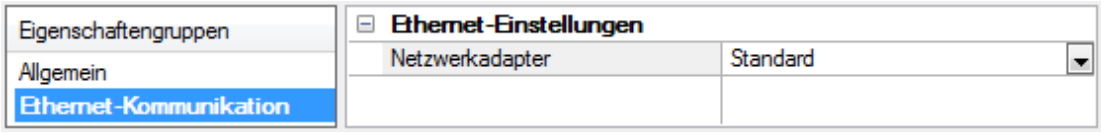
Diagnoseerfassung: Bei Aktivierung dieser Option stehen die Diagnoseinformationen des Kanals für OPC-Anwendungen zur Verfügung. Da für die Diagnosefunktionen des Servers eine minimale Mehraufwandsverarbeitung erforderlich ist, wird empfohlen, dass sie bei Bedarf verwendet werden und ansonsten deaktiviert sind. Die Standardeinstellung ist deaktiviert.

Hinweise: Diese Eigenschaft ist nicht verfügbar, wenn der Treiber Diagnosen nicht unterstützt.

Weitere Informationen finden Sie unter "Kommunikationsdiagnosen" und "Statistik-Tags" in der Serverhilfe.

Kanaleigenschaften - Ethernet-Kommunikation

Ethernet-Kommunikation kann für die Kommunikation mit Geräten verwendet werden.

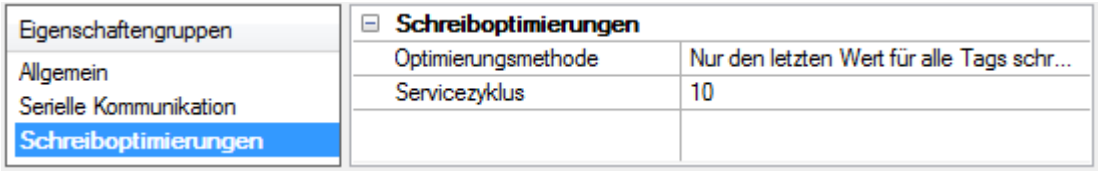


Ethernet-Einstellungen

Netzwerkadapter: Geben Sie den zu bindenden Netzwerkadapter an. Wird dieses Feld leer gelassen oder es wird "Standard" ausgewählt, so wählt das Betriebssystem den Standardadapter aus.

Kanaleigenschaften - Schreiboptimierungen

Wie bei jedem Server ist das Schreiben von Daten auf das Gerät unter Umständen der wichtigste Aspekt der Anwendung. Der Server soll sicherstellen, dass die von der Client-Anwendung geschriebenen Daten rechtzeitig auf das Gerät gelangen. In Anbetracht dieses Ziels stellt der Server Optimierungseigenschaften bereit, anhand derer die jeweiligen Anforderungen erfüllt oder die Reaktionsfähigkeit der Anwendungen verbessert werden können.



Schreiboptimierungen

Optimierungsmethode: Mit dieser Option wird gesteuert, wie Schreibdaten an den zugrunde liegenden Kommunikationstreiber weitergeleitet werden. Die Optionen sind:

- **Alle Werte für alle Tags schreiben**: Mit dieser Option wird der Server gezwungen, für jeden Wert einen Schreibvorgang auf dem Controller zu versuchen. In diesem Modus sammelt der Server weiterhin Schreibanforderungen und fügt sie der internen Schreibwarteschlange des Servers hinzu. Der Server verarbeitet die Schreibwarteschlange und versucht, sie zu leeren, indem er so schnell wie möglich Daten auf das Gerät schreibt. In diesem Modus wird sichergestellt, dass alles, was von den Client-Anwendungen geschrieben wird, an das Zielgerät gesendet wird. Dieser Modus sollte ausgewählt werden, wenn die Reihenfolge des Schreibvorgangs oder der Inhalt des Schreibelements eindeutig auf dem Zielgerät zu finden sein muss.
- **Nur den letzten Wert für nicht boolesche Tags schreiben**: Viele aufeinander folgende Schreibvorgänge für denselben Wert können sich aufgrund der Zeit, die tatsächlich zum Senden der Daten auf das Gerät erforderlich ist, in der Schreibwarteschlange ansammeln. Wenn der Server einen Schreibwert aktualisiert, der bereits in die Schreibwarteschlange eingefügt wurde, sind weitaus weniger Schreibvorgänge erforderlich, um denselben Endausgabewert zu erhalten. Auf diese Weise sammeln sich keine zusätzlichen Schreibvorgänge in der Warteschlange des Servers an. Wenn der Benutzer den Schiebeschalter

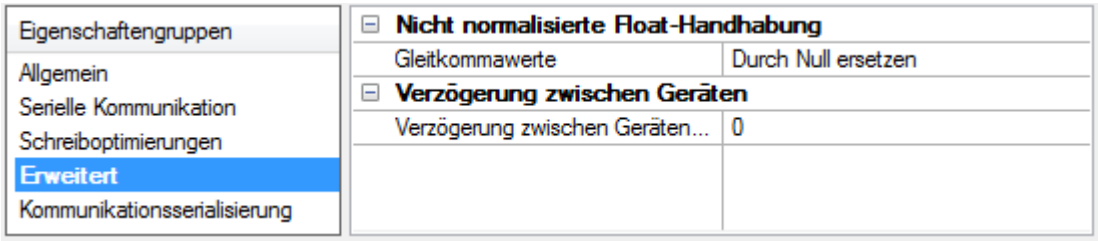
nicht mehr verschiebt, erreicht der Wert im Gerät praktisch in derselben Zeit den richtigen Wert. Dem Modus entsprechend wird jeder Wert, der kein boolescher Wert ist, in der internen Warteschlange des Servers aktualisiert und bei der nächstmöglichen Gelegenheit an das Gerät gesendet. Dies kann die Anwendungsleistung erheblich verbessern.
Hinweis: Mit dieser Option wird nicht versucht, Schreibvorgänge in Boolesche Werte zu optimieren. Dadurch können Benutzer den HMI-Datenvorgang optimieren, ohne Probleme mit Booleschen Operationen (z.B. eine vorübergehende Schaltfläche) zu verursachen.

- **Nur den letzten Wert für alle Tags schreiben**: Mit dieser Option wird die hinter der zweiten Optimierungsmethode stehende Theorie auf alle Tags angewendet. Sie ist besonders nützlich, wenn die Anwendung nur den letzten Wert an das Gerät senden muss. In diesem Modus werden alle Schreibvorgänge optimiert, indem die derzeit in der Schreibwarteschlange befindlichen Tags vor dem Senden aktualisiert werden. Dies ist der Standardmodus.

Servicezyklus: Wird verwendet, um das Verhältnis von Schreib- und Lesevorgängen zu steuern. Das Verhältnis basiert immer auf einem Lesevorgang für jeden zehnten Schreibvorgang. Für den Servicezyklus wird standardmäßig 10 festgelegt. Dies bedeutet, dass 10 Schreibvorgänge für jeden Lesevorgang erfolgen. Zwar führt die Anwendung eine große Anzahl fortlaufender Schreibvorgänge durch, doch muss sichergestellt werden, dass es für Lesedaten weiterhin Verarbeitungszeit gibt. Die Einstellung 1 hat zur Folge, dass ein Lesevorgang für jeden Schreibvorgang erfolgt. Wenn es keine durchzuführenden Schreibvorgänge gibt, werden Lesevorgänge fortlaufend verarbeitet. Dies ermöglicht eine Optimierung für Anwendungen mit fortlaufenden Schreibvorgängen gegenüber einem ausbalancierteren Datenzufluss und -abfluss.
Hinweis: Es wird empfohlen, dass für die Anwendung die Kompatibilität mit den Verbesserungen zur Schreiboptimierung charakteristisch ist, bevor sie in einer Produktionsumgebung verwendet wird.

Kanaleigenschaften - Erweitert

Diese Gruppe wird verwendet, um erweiterte Kanaleigenschaften anzugeben. Nicht alle Treiber unterstützen alle Eigenschaften; so wird die Gruppe "Erweitert" für jene Geräte nicht angezeigt.



Behandlung nicht normalisierter Gleitkommazahlen: Ein nicht normalisierter Wert wird als "Unendlich", "Nichtzahlenwert (NaN)" oder als "Denormalisierte Zahl" definiert. Die Standardeinstellung ist Durch Null ersetzen. Für Treiber, die eine native Float-Handhabung aufweisen, wird standardmäßig unter Umständen "Nicht geändert" verwendet. Durch Behandlung nicht normalisierter Gleitkommazahlen können Benutzer festlegen, wie ein Treiber mit nicht normalisierten IEEE-754-Gleitkommadaten umgeht. Es folgen Beschreibungen der Optionen:

- **Durch Null ersetzen**: Diese Option ermöglicht es einem Treiber, nicht normalisierte IEEE-754-Gleitkommawerte durch Null zu ersetzen, bevor sie an Clients übertragen werden.
- **Nicht geändert**: Diese Option ermöglicht es einem Treiber, denormalisierte, normalisierte IEEE-754-Nichtzahlenwerte und unendliche IEEE-754-Werte ohne jegliche Konvertierung oder Änderungen an Clients zu senden.

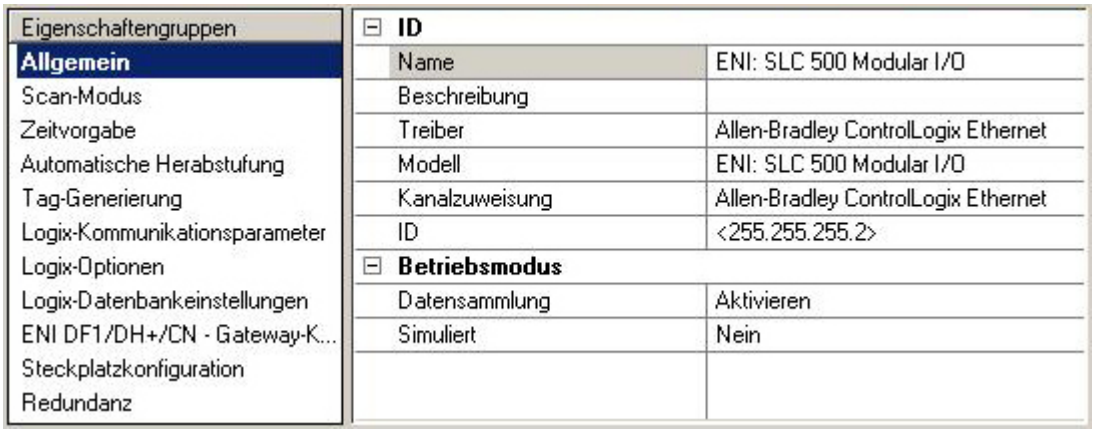
Hinweis: Diese Eigenschaft ist nicht verfügbar, wenn der Treiber keine Gleitkommawerte unterstützt, oder wenn er nur die angezeigte Option unterstützt. Gemäß der Float-Normalisierungseinstellung des Kanals unterliegen nur Echtzeit-Treiber-Tags (wie z.B. Werte und Arrays) der Float-Normalisierung. Beispielsweise werden EFM-Daten nicht durch diese Einstellung beeinflusst.

Weitere Informationen über die Gleitkommawerte finden Sie unter "So arbeiten Sie mit nicht normalisierten Gleitkommawerten" in der Serverhilfe.

Verzögerung zwischen Geräten: Geben Sie die Zeitdauer an, in der der Kommunikationskanal das Senden einer Anforderung an das nächste Gerät verzögert, nachdem Daten vom aktuellen Gerät in demselben Kanal empfangen wurden. Null (0) deaktiviert die Verzögerung.

Hinweis: Diese Eigenschaft ist nicht für alle Treiber, Modelle und abhängige Einstellungen verfügbar.

Geräteeigenschaften - Allgemein



Identifikation

Name: Benutzerdefinierte ID dieses Geräts.

Beschreibung: Benutzerdefinierte Informationen über dieses Gerät.

Kanalzuweisung: Benutzerdefinierter Name des Kanals, zu dem dieses Gerät derzeit gehört.

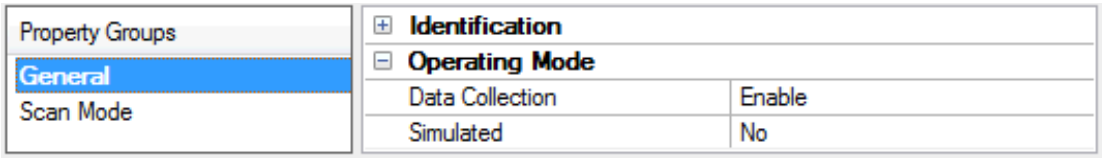
Treiber: Ausgewählter Protokolltreiber für dieses Gerät.

Modell: Die jeweilige Version des Geräts.

ID: Geben Sie die eindeutige Netzwerkadresse des Geräts ein, üblicherweise im Format <IP oder Host-Name>,1, <Routing-Pfad>,<Steckplatz>.

Die Konventionen für die Adressierung sind je nach Modell und Routing unterschiedlich. Weitere Informationen dazu finden Sie in den Themen zur modellspezifischen Adressierung unter Referenzmaterial.

Betriebsmodus



Datensammlung: Diese Eigenschaft steuert den aktiven Status des Geräts. Zwar sind Gerätekommunikationen standardmäßig aktiviert, doch kann diese Eigenschaft verwendet werden, um ein physisches Gerät zu deaktivieren. Kommunikationen werden nicht versucht, wenn ein Gerät deaktiviert ist. Vom Standpunkt eines Clients werden die Daten als ungültig markiert und Schreibvorgänge werden nicht akzeptiert. Diese Eigenschaft kann jederzeit durch diese Eigenschaft oder die System-Tags des Geräts geändert werden.

Simuliert: Diese Option versetzt das Gerät in den Simulationsmodus. In diesem Modus versucht der Treiber nicht, mit dem physischen Gerät zu kommunizieren, aber der Server gibt weiterhin gültige OPC-Daten zurück. Durch Auswählen von "Simuliert" wird die physische Kommunikation mit dem Gerät angehalten, OPC-Daten können jedoch als gültige Daten dem OPC-Client zurückgegeben werden. Im Simulationsmodus behandelt der Server alle Gerätedaten als reflektierend: was auch immer in das simulierte Gerät geschrieben wird, wird zurückgelesen, und jedes OPC-Element wird einzeln behandelt. Die Speicherzuordnung des Elements basiert auf der Gruppenaktualisierungsrate. Die Daten werden nicht gespeichert, wenn der Server das Element entfernt (z.B., wenn der Server neu initialisiert wird). Die Standardeinstellung ist "Nein".

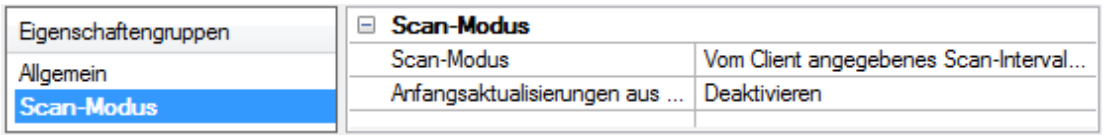
Hinweise:

1. Dieses System-Tag (_Simulated) ist schreibgeschützt und kann für den Laufzeitschutz nicht geschrieben werden. Das System-Tag ermöglicht es, dass diese Eigenschaft vom Client überwacht wird.
2. Im Simulationsmodus basiert die Speicherzuordnung des Elements auf Client-Aktualisierungsraten (Gruppenaktualisierungsrate für OPC-Clients oder Scan-Intervall für native und DDE-Schnittstellen). Das bedeutet, dass zwei Clients, die dasselbe Element mit unterschiedlichen Aktualisierungsraten referenzieren, verschiedene Daten zurückgeben.

Der Simulationsmodus ist nur für Test- und Simulationszwecke. Es sollte niemals in einer Produktionsumgebung nie verwendet werden.

Geräteeigenschaften – Scan-Modus

Der Scan-Modus gibt das vom abonnierten Client angeforderte Scan-Intervall für Tags an, die Gerätekommunikation erfordern. Synchrone und asynchrone Lese- und Schreibvorgänge des Geräts werden so bald wie möglich verarbeitet; unbeeinflusst von den Eigenschaften für den Scan-Modus.



Scan-Modus: Gibt an, wie Tags im Gerät für an abonnierende Clients gesendete Aktualisierungen gescannt werden. Es folgen Beschreibungen der Optionen:

- **Vom Client angegebenes Scan-Intervall berücksichtigen**: Dieser Modus verwendet das vom Client angeforderte Scan-Intervall.
- **Datenanfrage nicht schneller als Scan-Intervall**: Dieser Modus gibt den Wert an, der als maximales Scan-Intervall festgelegt wurde. Der gültige Bereich liegt zwischen 10 und 99999990 Millisekunden. Die Standardeinstellung ist 1000 Millisekunden.
  Hinweis: Wenn der Server über einen aktiven Client und Elemente für das Gerät verfügt und der Wert für das Scan-Intervall erhöht wird, werden die Änderungen sofort wirksam. Wenn der Wert für das Scan-Intervall verringert wird, werden die Änderungen erst wirksam, wenn alle Client-Anwendungen getrennt wurden.
- **Alle Datenanfragen im Scan-Intervall**: Dieser Modus erzwingt, dass Tags im angegebenen Intervall nach abonnierten Clients gescannt werden. Der gültige Bereich liegt zwischen 10 und 99999990 Millisekunden. Die Standardeinstellung ist 1000 Millisekunden.
- **Nicht scannen, nur Abruf anfordern**: In diesem Modus werden Tags, die zum Gerät gehören, nicht periodisch abgerufen, und es wird auch kein Lesevorgang durchgeführt, um den Anfangswert eines Elements abzurufen, sobald es aktiv wird. Es liegt in der Verantwortung des Clients, nach Aktualisierungen abzurufen, entweder durch Schreiben in das _DemandPoll-Tag oder durch Ausgeben expliziter Lesevorgänge des Geräts für einzelne Elemente. *Weitere Informationen finden Sie unter "Geräte-Bedarfsabruf" in der Serverhilfe*.
- **Durch Tag angegebenes Scan-Intervall berücksichtigen**: Dieser Modus erzwingt das Scannen statischer Tags im Intervall, das in ihrer statischen Konfiguration Tag-Eigenschaften angegeben wurde. Dynamische Tags werden in dem vom Client angegebenen Scan-Intervall gescannt.

Anfangsaktualisierungen aus Cache: Wenn diese Option aktiviert ist, kann der Server die ersten Aktualisierungen für neu aktivierte Tag-Referenzen aus gespeicherten (Cache-)Daten zur Verfügung stellen. Cache-Aktualisierungen können nur bereitgestellt werden, wenn die neue Elementreferenz dieselben Eigenschaften für Adresse, Scan-Intervall, Datentyp, Client-Zugriff und Skalierung gemeinsam nutzt. Ein Lesevorgang des Geräts wird nur für die Anfangsaktualisierung für die erste Client-Referenz verwendet. Der Standardeinstellung ist "Deaktiviert"; immer wenn ein Client eine Tag-Referenz aktiviert, versucht der Server, den Anfangswert vom Gerät zu lesen.

Geräteeigenschaften – Zeitvorgabe

Mithilfe der Zeitvorgabe-Eigenschaften des Geräts kann die Antwort des Treibers auf Fehlerbedingungen so angepasst werden, dass sie den Anforderungen der Anwendung entspricht. In vielen Fällen erfordert die Umgebung für eine optimale Leistung Änderungen an diesen Eigenschaften. Faktoren wie elektrisch generiertes Rauschen, Modemverzögerungen und fehlerhafte physische Verbindungen können beeinflussen, wie viele

Fehler oder Timeouts ein Kommunikationstreiber feststellt. Zeitvorgabe-Eigenschaften sind für jedes konfigurierte Gerät spezifisch.

Eigenschaftengruppen
Allgemein
Scan-Modus
Zeitvorgabe
Automatische Herabstufung

Kommunikations-Timeouts	
Anforderungs-Timeout (ms)	5000
Emeute Versuche	3
Zeitvorgabe	
Verzögerung zwischen Anfragen (ms)	0

Kommunikations-Timeouts

Verbindungs-Timeout: Mit dieser Eigenschaft (die in erster Linie von Ethernet-basierten Treibern verwendet wird) wird die Zeitdauer gesteuert, die zum Herstellen einer Socket-Verbindung mit einem Remote-Gerät erforderlich ist. Die Verbindungszeit des Gerät ist häufig länger als normale Kommunikationsanforderungen mit demselben Gerät. Der gültige Bereich liegt zwischen 1 und 30 Sekunden. Die Standardeinstellung ist normalerweise 3 Sekunden, kann jedoch abhängig vom jeweiligen Treiber unterschiedlich sein. Wenn diese Einstellung nicht vom Treiber unterstützt wird, ist sie deaktiviert.

Hinweis: Aufgrund der Art der UDP-Verbindungen ist die Einstellung für Verbindungs-Timeout nicht anwendbar, wenn die Kommunikation über UDP erfolgt.

Anforderungs-Timeout: Mit dieser Eigenschaft wird ein von allen Treibern verwendetes Intervall festgelegt, um zu bestimmen, wie lange der Treiber abschließend auf eine Antwort vom Zielgerät wartet. Der gültige Bereich liegt zwischen 50 und 9.999.999 Millisekunden (167,6667 Minuten). Die Standardeinstellung ist im Allgemeinen 1000 Millisekunden, kann jedoch abhängig vom Treiber unterschiedlich sein. Das Standard-Timeout für die meisten seriellen Treiber basiert auf einer Baudrate von 9600 Baud oder besser. Wenn ein Treiber bei niedrigeren Baudraten verwendet wird, erhöhen Sie das Timeout, um die erhöhte Zeit auszugleichen, die zum Abrufen von Daten erforderlich ist.

Versuche vor Timeout: Mit dieser Eigenschaft wird festgelegt, wie oft der Treiber eine Kommunikationsanforderung wiederholt, bevor er die Anforderung als fehlgeschlagen und das Gerät als fehlerhaft erachtet. Der gültige Bereich liegt zwischen 1 und 10. Die Standardeinstellung ist normalerweise 3, kann sich jedoch abhängig vom jeweiligen Treiber ändern. Die Anzahl der für eine Anwendung konfigurierten Wiederholungen hängt größtenteils von der Kommunikationsumgebung ab. Diese Eigenschaft trifft sowohl auf Verbindungsversuche als auch auf Anforderungsversuche zu.

Zeitvorgabe

Verzögerung zwischen Anfragen: Mit dieser Eigenschaft wird festgelegt, wie lange der Treiber wartet, bevor er die nächste Anforderung an das Zielgerät sendet. Sie setzt das dem Gerät zugewiesene normale Tag-Abfrageintervall sowie einmalige Lese- und Schreibvorgänge außer Kraft. Diese Verzögerung kann bei Geräten mit langsamen Durchlaufzeiten und in Situationen nützlich sein, in denen die Netzwerklast problematisch ist. Das Konfigurieren einer Verzögerung für ein Gerät wirkt sich auf die Kommunikation mit allen anderen Geräten im Kanal aus. Es wird empfohlen, dass Benutzer jedes Gerät trennen, das eine Verzögerung zwischen Anfragen für einen separaten Kanal erfordert (sofern möglich). Andere Kommunikationseigenschaften (z.B. Kommunikationsserialisierung) können diese Verzögerung verlängern. Der gültige Bereich liegt zwischen 0 und 300000 Millisekunden; jedoch können einige Treiber ggf. den maximalen Wert wegen einer Funktion ihrer spezifischen Konstruktion beschränken. Die Standardeinstellung ist 0. Dies weist darauf hin, dass es keine Verzögerung zwischen Anfragen mit dem Zielgerät gibt.

Hinweis: Nicht alle Treiber unterstützen Verzögerung zwischen Anfragen. Diese Einstellung wird nicht angezeigt, wenn sie nicht zur Verfügung steht.

Geräteeigenschaften - Automatische Herabstufung

Die Eigenschaften für automatische Herabstufung können ein Gerät vorübergehend in den Nicht-Scan-Modus versetzen, falls das Gerät nicht antwortet. Dadurch, dass ein nicht reagierendes Gerät für einen bestimmten Zeitraum offline gestellt wird, kann der Treiber weiterhin seine Kommunikation mit anderen Geräten in demselben Kanal optimieren. Nach Ablauf dieses Zeitraums versucht der Treiber die Kommunikation mit dem nicht reagierenden Gerät erneut. Wenn das Gerät reagiert, wird es wieder zum Scannen freigegeben. Andernfalls wird sein Nicht-Scan-Zeitraum erneut gestartet.

Eigenschaftengruppen
Allgemein
Scan-Modus
Zeitvorgabe
Automatische Herabstufung

Automatische Herabstufung	
Herabstufen bei Fehler	Aktivieren
Timeout bis zum Herabstufen	3
Herabstufungszeitraum (ms)	10000
Anfragen verwerfen, wenn herabgestuft	Deaktivieren

Herabstufen bei Fehler: Wird diese Option aktiviert, wird das Gerät automatisch in den Nicht-Scan-Modus versetzt, bis es wieder antwortet.
Tipp: Ermitteln Sie, wenn sich ein Gerät im Nicht-Scan-Modus befindet, indem Sie seinen herabgestuften Status mit dem _AutoDemoted-System-Tag überwachen.

Timeout bis zum Herabstufen: Legen Sie fest, wie viele aufeinander folgende Zyklen von Anforderungs-Timeouts und Wiederholungen vorkommen, bevor das Gerät in den Nicht-Scan-Modus versetzt wird. Der gültige Bereich ist 1 bis 30 aufeinander folgende Fehlschläge. Die Standardeinstellung ist 3.

Herabstufungszeitraum: Gibt an, wie lange das Gerät im Nicht-Scan-Modus sein sollte, wenn der Timeout-Wert erreicht wird. Während dieses Zeitraums werden keine Leseanforderungen an das Gerät gesendet, und für alle den Leseanforderungen zugeordneten Daten wird schlechte Qualität festgelegt. Wenn dieser Zeitraum abgelaufen ist, versetzt der Treiber das Gerät in den Scan-Modus und ermöglicht einen weiteren Kommunikationsversuch. Der gültige Bereich liegt zwischen 100 und 3600000 Millisekunden. Die Standardeinstellung ist 10000 Millisekunden.

Anfragen verwerfen, wenn herabgestuft: Durch Aktivieren dieser Option wird ausgewählt, ob Schreibanforderungen während des Nicht-Scan-Zeitraums versucht werden sollten. Deaktivieren Sie diese Option, damit Schreibanforderungen unabhängig vom Herabstufungszeitraum immer gesendet werden. Aktivieren Sie diese Option, um Schreibvorgänge zu verwerfen; auf dem Server schlägt jede von einem Client empfangene Schreibanforderung automatisch fehl, und es wird keine Meldung im Ereignisprotokoll angezeigt.

Geräteeigenschaften - Tag-Generierung

Mithilfe der Funktionen zur automatischen Tag-Datenbankgenerierung wird die Einrichtung einer Anwendung zu einem Plug-and-Play-Vorgang. Ausgewählte Kommunikationstreiber können so konfiguriert werden, dass automatisch eine Liste von Tags erstellt wird, die gerätespezifischen Daten entsprechen. Diese automatisch generierten Tags (die von der Art des unterstützenden Treibers abhängen) können von den Clients durchsucht werden.

Nicht alle Geräte und Treiber unterstützen die automatische Tag-Datenbankgenerierung und nicht alle unterstützen die gleichen Datentypen. Lesen Sie für weitere Informationen die Beschreibungen der Datentypen oder die Liste der unterstützten Datentypen für jeden Treiber.

Wenn das Zielgerät seine eigene lokale Tag-Datenbank unterstützt, liest der Treiber die Tag-Informationen des Geräts und verwendet die Daten zum Generieren von Tags innerhalb des Servers. Wenn das Gerät benannte Tags nicht nativ unterstützt, erstellt der Treiber eine Liste von auf treiberspezifischen Informationen basierenden Tags. Ein Beispiel dieser beiden Bedingungen sieht wie folgt aus:

1. Wenn ein Datenerfassungssystem seine eigene lokale Tag-Datenbank unterstützt, verwendet der Kommunikationstreiber die im Gerät gefundenen Tag-Namen, um die Tags des Servers zu erstellen.
2. Wenn ein Ethernet-E/A-System die Erkennung seiner eigenen verfügbaren E/A-Modultypen unterstützt, generiert der Kommunikationstreiber automatisch Tags auf dem Server, die auf den E/A-Modultypen im Ethernet-E/A-Rack basieren.

Hinweis: Der Betriebsmodus zur automatischen Tag-Datenbankgenerierung ist komplett konfigurierbar. *Weitere Informationen finden Sie in den nachfolgenden Eigenschaftsbeschreibungen.*

Eigenschaftengruppen
Allgemein
Scan-Modus
Zeitvorgabe
Automatische Herabstufung
Tag-Generierung

Tag-Generierung	
Bei Gerätestart	Nicht beim Start erstellen
Bei doppeltem Tag	Bei Erstellen löschen
Elternteilgruppe	
Automatisch generierte Untergruppen zulassen	Aktivieren

Bei Eigenschaftsänderung: Unterstützt das Gerät automatische Tag-Generierung, wenn bestimmte Eigenschaften geändert werden, so wird die Option **Bei Eigenschaftsänderung** angezeigt. Diese Option ist standardmäßig auf **Ja** festgelegt, kann jedoch auf **Nein** gesetzt werden, um zu steuern, wann eine Tag-Generierung stattfindet. In diesem Fall muss die Aktion Tags erstellen manuell aufgerufen werden, damit eine Tag-Generierung stattfindet.

Bei Gerätestart: Diese Eigenschaft gibt an, wann OPC-Tags automatisch generiert werden. Es folgen Beschreibungen der Optionen:

- **Nicht beim Start erstellen**: Mit dieser Option wird verhindert, dass der Treiber irgendwelche OPC-Tags dem Tag-Raum des Servers hinzufügt. Dies ist die Standardeinstellung.
- **Immer beim Start erstellen**: Das Auswählen dieser Option hat zur Folge, dass der Treiber das Gerät für Tag-Informationen bewertet. Es werden auch jedes Mal, wenn der Server gestartet wird, Tags dem Tag-Raum des Servers hinzugefügt.
- **Beim ersten Start erstellen**: Das Auswählen dieser Option hat zur Folge, dass der Treiber das Zielgerät für Tag-Informationen bewertet, wenn das Projekt zum ersten Mal ausgeführt wird. Es werden bei Bedarf auch sämtliche OPC-Tags dem Tag-Raum des Servers hinzugefügt.

Hinweis: Wenn die Option zum automatischen Generieren von OPC-Tags ausgewählt wird, müssen sämtliche Tags, die dem Tag-Raum des Servers hinzugefügt werden, mit dem Projekt gespeichert werden. Benutzer können das Projekt konfigurieren, um automatisch über das Menü **Tools** | **Optionen** zu speichern.

Bei doppeltem Tag: Wenn die automatische Tag-Datenbankgenerierung aktiviert wird, muss der Server wissen, wie mit Tags, die er möglicherweise zuvor hinzugefügt hat, oder mit Tags, die nach dem Kommunikationstreiber seit ihrer ursprünglichen Erstellung hinzugefügt oder geändert wurden, zu verfahren ist. Mit dieser Einstellung wird gesteuert, wie der Server OPC-Tags behandelt, die automatisch generiert wurden und derzeit im Projekt vorhanden sind. Es wird auch verhindert, dass sich automatisch generierte Tags auf dem Server ansammeln.

Beispiel: Wenn ein Benutzer die E/A-Module im Rack mit dem für **Immer beim Start erstellen** konfigurierten Server ändert, würden neue Tags jedes Mal dem Server hinzugefügt werden, wenn der Kommunikationstreiber ein neues E/A-Modul erkannt hat. Wenn die alten Tags nicht entfernt wurden, könnten sich viele unbenutzte Tags im Tag-Raum des Servers ansammeln. Die Optionen sind:

- **Bei Erstellen löschen**: Mit dieser Option werden sämtliche Tags gelöscht, die zuvor dem Tag-Raum hinzugefügt wurden, bevor sämtliche neuen Tags hinzugefügt werden. Dies ist die Standardeinstellung.
- **Nach Bedarf überschreiben**: Mit dieser Option wird der Server angewiesen, nur die Tags zu entfernen, die der Kommunikationstreiber durch neue Tags ersetzt. Sämtliche Tags, die nicht überschrieben werden, bleiben im Tag-Raum des Servers.
- **Nicht überschreiben**: Mit dieser Option wird verhindert, dass der Server sämtliche Tags entfernt, die zuvor generiert wurden oder bereits auf dem Server vorhanden waren. Der Kommunikationstreiber kann nur Tags hinzufügen, die völlig neu sind.
- **Nicht überschreiben, Fehler protokollieren**: Diese Option hat denselben Effekt wie die vorherige Option und sendet auch eine Fehlermeldung an das Ereignisprotokoll des Servers, wenn eine Tag-Überschreibung stattgefunden hätte.

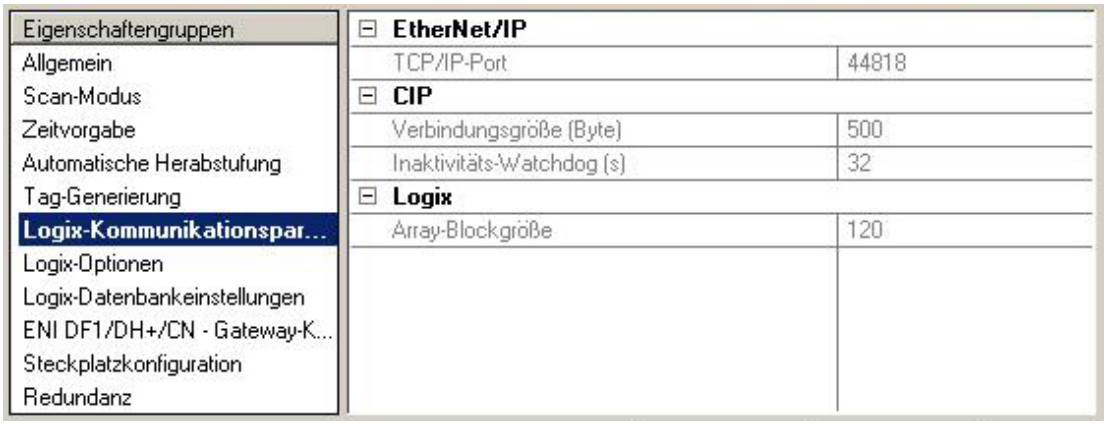
Hinweis: Das Entfernen von OPC-Tags wirkt sich auf Tags, die automatisch vom Kommunikationstreiber generiert wurden, sowie auf sämtliche Tags aus, die unter Verwendung von Namen, die generierten Tags entsprechen, hinzugefügt wurden. Benutzer sollten es vermeiden, Tags dem Server unter Verwendung von Namen hinzuzufügen, die möglicherweise den Tags entsprechen, die automatisch vom Treiber generiert werden.

Elternteilgruppe: Mit dieser Eigenschaft wird verhindert, dass sich automatisch generierte Tags mit Tags vermischen, die manuell eingegeben wurden, indem eine Gruppe festgelegt wurde, die für automatisch generierte Tags verwendet werden soll. Der Name der Gruppe kann bis zu 256 Zeichen lang sein. Diese Elternteilgruppe stellt einen Stammzweig bereit, dem alle automatisch generierten Tags hinzugefügt werden.

Automatisch generierte Untergruppen zulassen: Mit dieser Eigenschaft wird gesteuert, ob der Server automatisch Untergruppen für die automatisch generierten Tags erstellt. Dies ist die Standardeinstellung. Wenn diese Option deaktiviert ist, generiert der Server die Tags des Geräts in einer unstrukturierten Liste ohne jede Gruppierung. Im Serverprojekt werden die resultierenden Tags mit dem Adresswert benannt. Beispielsweise werden die Tag-Namen während des Generierungsprozesses nicht beibehalten.
Hinweis: Wenn beim Generieren von Tags durch den Server einem Tag derselbe Name wie einem bestehenden Tag zugewiesen wird, erhöht das System automatisch auf die nächste höchste Nummer, sodass der Tag-Name nicht dupliziert wird. Beispiel: Wenn der Generierungsprozess das Tag "AI22" erstellt, das bereits existiert, wird stattdessen das Tag als "AI23" erstellt.

Erstellen: Initiiert die Erstellung automatisch generierter OPC-Tags. Wenn die Konfiguration des Geräts geändert wurde, wird der Treiber durch die Option **Tags erstellen** gezwungen, das Gerät erneut auf mögliche Tag-Änderungen zu bewerten. Ihre Fähigkeit, über die System-Tags aufgerufen zu werden, ermöglicht einer Client-Anwendung das Initiieren der Tag-Datenbankerstellung.
Hinweis: **Tags erstellen** ist deaktiviert, wenn die Konfiguration ein Projekt offline bearbeitet.

Geräteeigenschaften – Logix-Kommunikationsparameter



EtherNet/IP

TCP/IP-Port: Gibt die TCP/IP-Port-Nummer an, die das Gerät der Konfiguration entsprechend verwenden soll. Die Standardeinstellung ist 44818.

CIP

Verbindungsgröße: Gibt die Anzahl der bei der CIP-Verbindung für Datenanforderungen und -antworten zur Verfügung stehenden Byte an. Der gültige Bereich liegt zwischen 500 und 4000 Byte. Die Standardeinstellung ist 500 Byte.
Hinweis: Nur die ControlLogix 5500- und CompactLogix 5300-Gerätemodelle unterstützen diese Funktion. Zur Unterstützung von Verbindungsgrößen über 500 Byte muss das Gerät Controller und Ethernet-Bridge EN3x, EN2x oder EN5.x ab Firmware-Version 20 unterstützen. Ältere Ethernet-Module wie ENBT und ENET unterstützen diese Funktion nicht. Geräte, die die notwendigen Anforderungen nicht erfüllen, kehren automatisch zur Standardeinstellung von 500 Byte zurück, obwohl die angeforderte Größe nach Kommunikationsfehler erneut versucht wird.
Der Wert Verbindungsgröße kann ggf. auch über das System-Tag "_CIPConnectionSizeRequested" angefordert werden. Weitere Informationen dazu finden Sie unter Interne Tags.

Inaktivitätsüberwachung: Gibt die Dauer in Sekunden an, die eine Verbindung inaktiv (ohne Lese-/Schreibtransaktionen) bleiben kann, bevor sie vom Controller geschlossen wird. Je größer der Wert, desto mehr Zeit wird benötigt, bis Verbindungsressourcen durch den Controller freigegeben werden und umgekehrt. Der Standardwert beträgt 32 Sekunden.

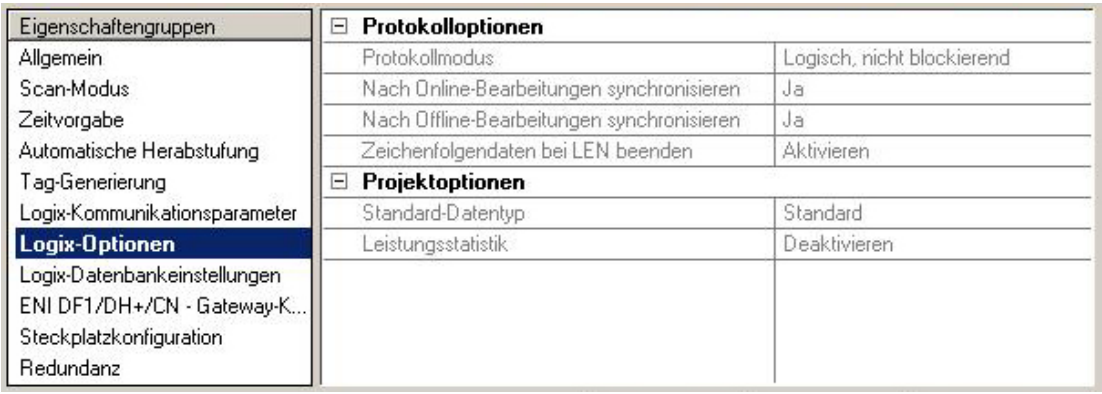
Hinweis: Wenn beim Hochladen eines Projekts häufig Fehler bezüglich des Timeouts der CIP-Verbindung auftreten, erhöhen Sie den Wert Inaktivitätsüberwachung. Andernfalls wird der Standardwert empfohlen.

Logix

Array-Blockgröße: Diese Eigenschaft gibt die maximale Anzahl der Array-Elemente an, die in einer einzelnen Transaktion gelesen werden. Der Wert ist anpassbar und liegt zwischen 30 und 3840 Elementen. Der Standardwert ist 120 Elemente.

Tipp: Für Boolesche Arrays wird ein einzelnes Element als Bit-Array mit 32 Elementen angesehen. Durch Festlegen der Blockgröße auf 30 Elemente werden 960 Bit-Elemente konvertiert, wohingegen bei 3840 Elementen 122880 Bit-Elemente konvertiert werden.

Geräteeigenschaften – Logix-Optionen



Protokolloptionen

Protokollmodus: Wählen Sie aus, wie Logix-Tag-Daten vom Controller gelesen werden: Logisch, nicht blockierend, Logisch, blockierend und Symbolisch. Die Standardeinstellung ist Logisch, nicht blockierend. Diese Option sollte nur von fortgeschrittenen Benutzern geändert werden, die die Client-Server-Tag-Aktualisierungsleistung zu erhöhen versuchen.
Weitere Informationen dazu finden Sie unter Protokollmodus auswählen.
Hinweis: Die Modi "Logisch, nicht blockierend" und "Logisch, blockierend" stehen für Serial Gateway-Modelle nicht zur Verfügung.

Nach Online-Bearbeitungen synchronisieren: Bei Aktivierung dieser Option synchronisiert der Treiber sein Projekt-Image mit jenem des Controller-Projekts, wenn eine Online-Bearbeitung des Projekts (oder ein Projekt-Download von RSLogix/Studio5000 aus) erkannt wird. Mit dieser Option werden unnötige Fehler während einer Projektänderung verhindert. Sie steht nur zur Verfügung, wenn der ausgewählte Protokollmodus "Logisch, nicht blockierend" oder "Logisch, blockierend" ist. Die Standardeinstellung ist Ja.

Nach Offline-Bearbeitungen synchronisieren: Bei Aktivierung dieser Option synchronisiert der Treiber sein eigenes Projekt-Image mit jenem des Controller-Projekts, wenn eine Offline-Bearbeitung des Projekts (oder ein Projekt-Download von RSLogix/Studio5000 aus) erkannt wird. Mit dieser Option werden unnötige Fehler während einer Projektänderung verhindert. Sie steht nur zur Verfügung, wenn der ausgewählte Protokollmodus "Logisch, nicht blockierend" oder "Logisch, blockierend" ist. Die Standardeinstellung ist Ja.
Schlägt das Synchronisieren mit Projektänderungen fehl, kann es dazu führen, dass von der falschen nativen Tag-Adresse gelesen bzw. in die falsche native Tag-Adresse geschrieben wird.

Zeichenfolgendaten bei LEN beenden: Bei Aktivierung dieser Option liest der Treiber automatisch das LEN-Mitglied der STRING-Struktur, immer wenn das DATA-Mitglied gelesen wird. Die DATA-Zeichenfolge wird beim ersten festgestellten Nullzeichen, beim Zeichen, dessen Position gleich dem LEN-Wert ist oder bei der maximalen DATA-Zeichenfolgenlänge (was auch immer zuerst vorkommt) beendet. Sofern deaktiviert, umgeht der Treiber das Lesen des LEN-Mitglieds und beendet die DATA-Zeichenfolge entweder beim ersten festgestellten Nullzeichen oder bei der maximalen DATA-Zeichenfolgenlänge (was auch immer zuerst vorkommt). Wenn LEN ohne Änderung an DATA durch eine externe Quelle reduziert wird, beendet der Treiber aus diesem Grund DATA gemäß dieser reduzierten Länge nicht. Die Standardeinstellung ist Aktivieren.

Projektoptionen

Standard-Datentyp: Wählen Sie den einem Client-/Server-Tag zugewiesenen Datentyp aus, wenn der Standardtyp während des Hinzufügens, Änderns oder Importierens des Tags ausgewählt wird. Die Standardeinstellung ist Standard.
Weitere Informationen dazu finden Sie unter Bedingungen für Standard-Datentyp.

Leistungsstatistik: ControlLogix-Ethernet-Treiber von Allen-Bradley hat die Möglichkeit zur Erfassung von Kommunikationsstatistiken, um die Leistung des Treibers zu bestimmen. Durch Auswahl dieser Option wird sie aktiviert. Der Treiber verfolgt die Anzahl und Arten der Client-/Server-Tag-Aktualisierungen. Beim Neustart der Serveranwendung werden die Ergebnisse im Ereignisprotokoll des Servers angezeigt. Die Standardeinstellung ist Nein.
Hinweis: Sobald eine Projektkonfiguration zur Leistungsoptimierung vorgesehen ist, wird empfohlen, dass Benutzer die Leistungsstatistik deaktivieren. Da die Statistik beim Herunterfahren in das Ereignisprotokoll geschrieben wird, muss der Server zum Anzeigen der Ergebnisse erneut gestartet werden.
Siehe auch: *Änderung im Controller-Projekt erkennen*

Geräteeigenschaften – Logix-Datenbankeinstellungen

Datenbank-Importmethode

Datenbank-Importmethode: Wählen Sie aus, wie die Tag-Datenbank gefüllt werden sollte:

- **Von Gerät erstellen**: Ruft Tags direkt vom Controller über dieselbe Ethernet-Verbindung ab, die für Datenzugriff verwendet wird. Diese ist schnell und es werden die meisten Tags importiert, jedoch wird Zugriff auf den Controller benötigt, und es werden keine Beschreibungen importiert. Zu den nicht importierten Tags gehören InOut-Eigenschaften von AOI (Add On Instruction).
  Hinweis: Diese Funktion steht für Serial Gateway-Modelle nicht zur Verfügung.
- **Von Importdatei erstellen**: Ruft Tags von einer ausgewählten RSLogix-L5K/L5X-Datei ab. Controller-Zugriff ist nicht erforderlich, Beschreibungen werden importiert und Benutzer können offline arbeiten. Diese Option ist jedoch langsamer und es werden nicht alle Tags in den Controller importiert. Zu den nicht importierten Tags gehören:
  - E/A-Tags
  - InOut-Eigenschaften von AOI (Add On Instruction)
  - AOI-Eigenschaften, die als Alias anderer Eigenschaften fungieren
  - Equipment Phase-Eigenschaften, die als Alias von Eigenschaften von anderer Equipment Phase oder von anderem Program fungieren
  - Program-Eigenschaften, die als Alias von Eigenschaften von anderem Program oder von anderer Equipment Phase fungieren
  - Timer/Counter-CTL-Bits

Tag-Importdatei: Klicken Sie auf die Schaltfläche zum Durchsuchen (...), um die L5K/L5X-Datei, aus der Tags importiert werden sollen, zu suchen und auszuwählen. Diese Datei wird verwendet, wenn die automatische

Tag-Datenbankgenerierung angewiesen wird, die Tag-Datenbank zu erstellen. Alle Tags, einschließlich "Global" und "Program", werden ihren jeweiligen Datentypen entsprechend importiert und erweitert.

Tag-Beschreibungen: Wählen Sie **Aktivieren**, um Tag-Beschreibungen für Nicht-Struktur- bzw. Nicht-Array-Tags zu importieren. Bei Bedarf werden Tags mit langen Namen, die den ursprünglichen Tag-Namen angeben, mit einer Beschreibung versehen.

Tag-Generierung mit der API

Die Konfigurations-API-Eigenschaften für CLX-Offline-ATG sind:

```
"controllogix_ethernet.DEVICE_DATABASE_IMPORT_METHOD": 1, "controllogix_ether-
net.DEVICE_TAG_IMPORT_FILE": "myFile.l5x", "controllogix_ethernet.DEVICE_DISPLAY_
DESCRIPTIONS": true,
```

wobei die Aufzählung für die Importmethode wie folgt ist:
0 für "Von Gerät erstellen"
und
1 für "Von Importdatei erstellen"

Logix-Datenbankoptionen

Namenlänge begrenzen: Wählen Sie Aktivieren aus, um die Tag- und Gruppennamen auf 31 Zeichen zu beschränken. Die Standardeinstellung ist **Deaktivieren**.

1. Vor OPC-Server Version 4.70 wurde die Länge von Tag- und Gruppennamen auf 31 Zeichen beschränkt. Die aktuelle Längenbeschränkung von 256 Zeichen kann Logix-Tag-Namen mit Logix 40-Zeichen anpassen.
2. Wenn eine ältere Serverversion zum Importieren von Tags über den L5K/L5X-Importvorgang verwendet wurde, überprüfen Sie das Ereignisprotokoll oder untersuchen Sie das Serverprojekt, um festzustellen, ob irgendwelche Tags aufgrund der Zeichenbeschränkung abgeschnitten wurden. Falls ja, aktivieren Sie diese Eigenschaft, um die Server-Tag-Namen beizubehalten. Tag-Referenzen für den OPC-Client sind davon nicht betroffen. Sofern deaktiviert, werden längere Tag-Namen erstellt und Clients, die das abgeschnittene Tag referenzieren, müssen entsprechend geändert werden, um den neuen Tag-Namen zu referenzieren.
3. Wenn eine ältere OPC-Serverversion zum Importieren von Tags über den L5K/L5X-Importvorgang verwendet wurde und keine Tags aufgrund der Beschränkung auf 31 Zeichen abgeschnitten wurden, lassen Sie diese Optionen deaktiviert.
4. Wenn Tags über L5K/L5X mit Serverversion 4.70 oder höher importiert wurden, lassen Sie diese Optionen deaktiviert.
   Siehe auch: Namenskonvertierungen von Controller in Server

Tag-Hierarchie: Diese Eigenschaft legt die Strukturorganisation der Tag-Hierarchie fest. Die Option **Komprimiert** besagt, dass die durch automatische Tag-Generierung erstellten Server-Tags einer mit der Adresse des Tags konsistenten Gruppen-/Tag-Hierarchie folgen. Gruppen werden für jedes dem Zeitraum vorangehende Segment erstellt. Die Option **Erweitert** besagt, dass die durch automatische Tag-Generierung erstellten Server-Tags einer mit der Tag-Hierarchie in RSLogix 5000 konsistenten Gruppen-/Tag-Hierarchie folgen. Gruppen werden für jedes dem Zeitraum vorangehende Segment erstellt und stellen logische Gruppierungen dar. Um diese Funktionalität zu verwenden, aktivieren Sie **Untergruppen zulassen** in den Eigenschaften Tag-Generierung.
Weitere Informationen zu den erstellten Gruppen finden Sie unter Tag-Hierarchie und Namenskonvertierungen von Controller in Server.

Logix-Datenbankfilterung

Array-Grenze festlegen: Wählen Sie Aktivieren aus, um die Anzahl von Array-Elementen zu beschränken. Tags im Controller können mit sehr großen Array-Dimensionen deklariert werden. Standardmäßig werden Arrays während des Tag-Generierungsprozesses vollständig erweitert, was für große Arrays zeitaufwendig wird. Durch Festlegen einer Höchstgrenze wird nur eine angegebene Anzahl von Elementen aus jeder Dimen-

sion generiert. Die Begrenzung wird nur wirksam, wenn die Array-Dimensionsgröße die Höchstgrenze überschreiten würde. Die Standardeinstellung ist **Deaktivieren**.

Oberer Grenzwert der Array-Anzahl: Legt die Array-Anzahl-Grenze fest. Der Standardwert ist 2000.

Geräteeigenschaften – Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway

TCP/IP-Port: Gibt die Port-Nummer an, deren Verwendung durch das Remote-Gerät konfiguriert wurde (z.B. 1756-ENBT). Die Standardeinstellung ist 44818.

Anforderungsgröße: Wählen Sie die Anzahl Byte aus, die von einem Gerät gleichzeitig angefordert werden kann, um die Leistung zu verfeinern. Mögliche Optionen sind 32, 64, 128 oder 232. Die Standardeinstellung ist 232 Byte.

Blockschreibvorgänge zulassen für Funktionsdateien: Funktionsdateien sind strukturbasierte Dateien (wie PD- und MG-Datendateien) und für MicroLogix 1100, 1200 und 1500 eindeutig. Bei anwendbaren Funktionsdateien können Daten in einem Schreibvorgang in das Gerät geschrieben werden. Wenn Daten in ein Unterelement der Funktionsdatei (Feld innerhalb der Funktionsdateistruktur) geschrieben werden, findet sofort eine Schreiboperation für dieses Tag statt. Für solche Dateien wie die RTC-Datei, deren Unterelemente Stunde (HR), Minute (MIN) und Sekunde (SEC) enthalten, sind einzelne Schreibvorgänge nicht immer akzeptabel. Bei solchen Unterelementen, die sich einzig auf die Zeit verlassen, müssen Werte in einer Operation geschrieben werden, um zu vermeiden, dass Zeit zwischen den Schreibvorgängen der Unterelemente verstreicht. Deshalb gibt es die Option, diese Unterelemente im Block zu schreiben. Die Standardeinstellung ist deaktiviert.

Weitere Informationen dazu finden Sie unter Blockschreibvorgänge und Funktionsdateien.

Blockschreibvorgänge

Beim Blockschreiben werden die Werte jedes Lesen/Schreiben-Unterelements in der Funktionsdatei in einem einzigen Schreibvorgang in das Gerät geschrieben. Es ist nicht erforderlich, vor der Durchführung eines Blockschreibvorgangs in jedes Unterelement zu schreiben. Für Unterelemente, die nicht davon betroffen sind (geschrieben werden), wird dessen aktueller Wert wieder in sie zurückgeschrieben. Beispiel: Wenn die aktuellen (zuletzt gelesenen) Werte für Datum und Uhrzeit der 1.1.2001, 12:00.00, DOW = 3 sind und die Stunde auf 1 Uhr geändert wird, werden folgende Werte in das Gerät geschrieben: 1.1.2001, 1:00.00, DOW = 3. Weitere Informationen dazu finden Sie in den folgenden Anweisungen.

1. Starten Sie mit der Suche nach den **Kommunikationsparametern für ENI DF1-/DH+-/CN-Gateway** unter **Geräteeigenschaften**.
2. Aktivieren Sie die Option **Blockschreibvorgänge für Funktionsdateien zulassen**, um den Treiber darüber zu benachrichtigen, Blockschreibvorgänge bei Funktionsdateien zu verwenden, die Blockschreibvorgänge unterstützen.
3. Klicken Sie auf **OK** oder **Anwenden**.

4. Schreiben Sie den gewünschten Wert in das betreffende Unterelement-Tag. Das Unterelement-Tag übernimmt unverzüglich den Wert, der in es geschrieben wurde.

   Hinweis: Nachdem ein Unterelement mindestens einmal im Blockschreibvorgangsmodus geschrieben wurde, stammt der Wert des Tags nicht vom Controller, sondern aus dem Schreibcache des Treibers. Nachdem der Blockschreibvorgang abgeschlossen ist, stammen alle Tag-Werte des Unterelements vom Controller.

5. Sobald die gesamten gewünschten Unterelemente geschrieben wurden, führen Sie den Blockschreibvorgang durch, der diese Werte an den Controller sendet. Referenzieren Sie zum Instanziieren eines Blockschreibvorgangs die Tag-Adresse *RTC:<Element>._SET*. Das Festlegen dieses Tag-Werts auf 'wahr' hat zur Folge, dass ein Blockschreibvorgang basierend auf den aktuellen (zuletzt gelesenen) Unterelementen und den betroffenen Unterelementen (in die geschriebenen wurde) stattfindet. Unmittelbar nach dem Festlegen des Tags auf "wahr" wird es automatisch auf "falsch" zurückgesetzt. Dies ist der Standardstatus und es werden keine Aktionen durchgeführt.

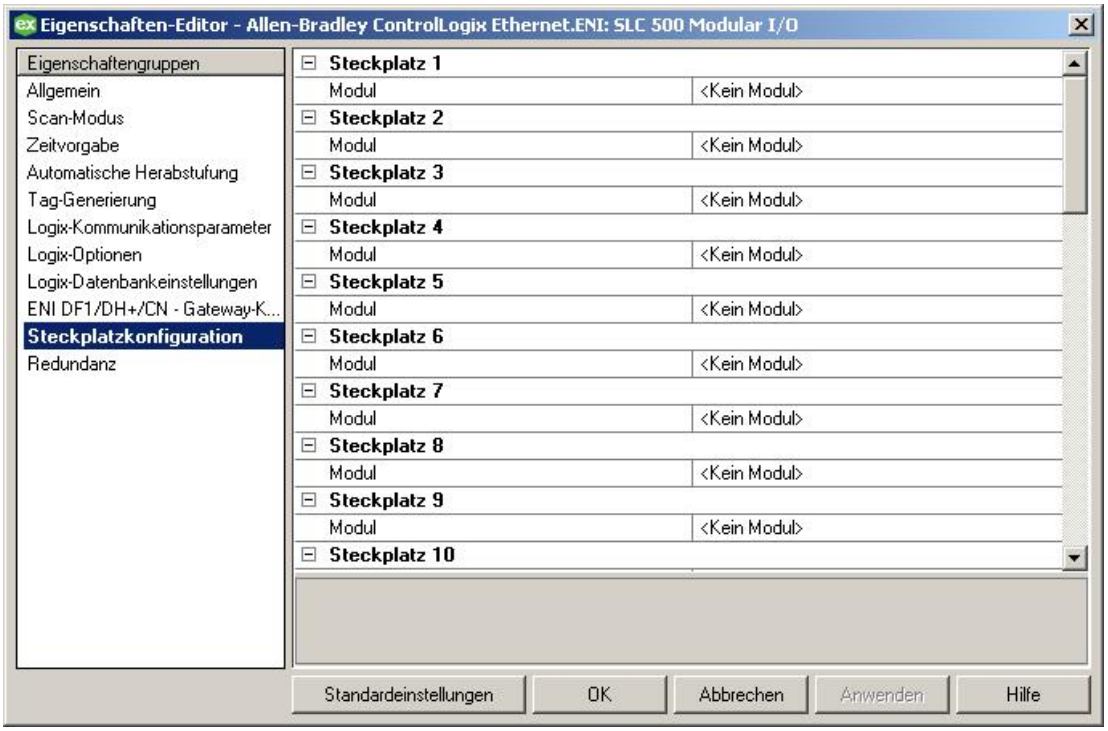
Anwendbare Funktionsdateien/Unterelemente

RTC	
Jahr	YR
Monat	MON
Tag	DAY
Wochentag	DOW
Stunde	HR
Minute	MIN
Sekunde	SEC

Siehe auch: Auflistung der Funktionsdatei

Geräteeigenschaften - SLC 500-Steckplatzkonfiguration

Damit auf E/A zugegriffen werden kann, müssen SLC5/01/02/03/04/05-Modelle (modulare E/A-Racks) für die Verwendung mit ControlLogix-Ethernet-Treiber von Allen-Bradley konfiguriert werden. Bis zu 30 Steckplätze können pro Gerät konfiguriert werden.



Steckplatz *n*: Der physische Steckplatz, der konfiguriert wird. Mit dem Plus-Symbol können Sie die Eigenschaften erweitern.

Modul: Legt den Modultyp im Steckplatz anhand der Optionen in der Dropdown-Liste fest.
Informationen zum Konfigurieren von Steckplätzen über den Konfigurations-API-Dienst finden Sie in diesem Beispiel.

Eingabewörter: Falls dies vom ausgewählten Modul benötigt wird, geben Sie die maximale Anzahl von Eingabewörtern für dieses Modul ein.

Ausgabewörter: Falls dies vom ausgewählten Modul benötigt wird, geben Sie die maximale Anzahl von Ausgabewörtern für dieses Modul ein.

Wenn Sie die Steckplatzkonfiguration verwenden möchten, beachten Sie die nachstehenden Anweisungen:

1. Wählen Sie den zu konfigurierenden Steckplatz aus, indem Sie im Modul-Listenfeld auf die Zeile klicken.
2. Klicken Sie zum Auswählen eines Moduls in der Dropdown-Liste mit den verfügbaren Modulen darauf.
3. Konfigurieren Sie bei Bedarf die Eingabewörter und Ausgabewörter.
4. Wenn Sie einen Steckplatz bzw. ein Modul entfernen möchten, wählen Sie in der Dropdown-Liste mit den verfügbaren Modulen **Kein Modul** aus.
5. Klicken Sie nach Abschluss auf **OK**.

Tipps:

- Mit 0000-Generic Module können Sie nicht in der Liste mit den verfügbaren Modulen enthaltene E/A konfigurieren.
- Die zur Verfügung stehende Modulauswahl ist mit jener in der APS-Software von Allen-Bradley identisch.

Hinweis: Es ist üblich, offene Steckplätze im Rack zu haben, in denen sich kein physisches Modul befindet. Damit ein korrekter Zugriff auf Daten für die verschiedenen Steckplätze, die ein Modul aufweisen, erfolgen kann, muss den vorhergehenden Modulen die richtige Anzahl von Wörtern zugeordnet sein. Beispiel: Wenn nur Interesse an E/A in Steckplatz 3 besteht, die Steckplätze 1 und 2 jedoch E/A-Module enthalten, müssen von dieser Gruppe zur Steckplatzkonfiguration aus die richtigen Module für die Steckplätze 1, 2 und 3 ausgewählt werden.

0000-Generic Module
Mit dem Generic Module können Sie Eingabe- und Ausgabewörter für Module zuordnen, die in der Liste der verfügbaren Module nicht enthalten sind. Um das Generic Module ordnungsgemäß zu verwenden, müssen Benutzer die Anzahl der für jedes Modul erforderlichen Eingabe- und Ausgabewörter kennen.

Schlagen Sie zur Bestätigung von Eingabe- und Ausgabeanforderungen in der E/A-Benutzerhandbuchdokumentation von Allen-Bradley nach und denken Sie daran, dass die Anforderungen je nach Operationen der Klasse 1 oder 3 unterschiedlich sein können.
Weitere Informationen zur Anzahl der für jedes E/A-Modul zur Verfügung stehenden Eingabe- und Ausgabewörter finden Sie unter Richtlinien für modulare E/A-Auswahl.

Konfigurations-API - Steckplatzkonfiguration

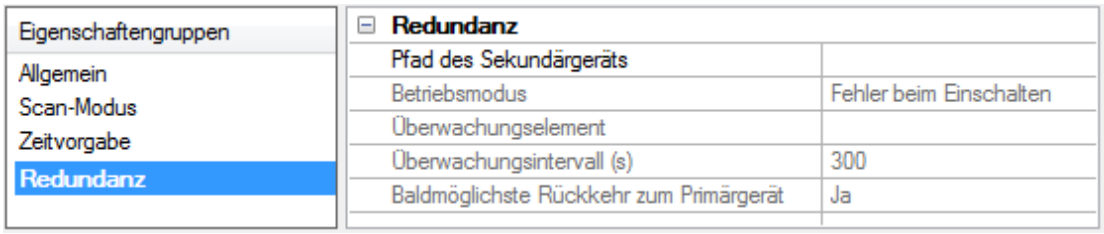
Basierend auf dieser Beispielsteckplatzkonfiguration:

Steckplatz 1: 1746-OV32 - 32 Output [Trans] (Senke) 10/50 V_{DC}
Steckplatz 2: 1746-I*16 - Jedes einzelne 16-pt-Eingabemodul
Steckplatz 3: <Kein Modul>
Steckplatz 4: 0000-Generisches Modul [1 Eingabewort, 2 Ausgabewörter]
Steckplatz 5: 1746-NT8 Analog 8 Ch - Thermoelementeingang
Steckplatz 6-30: <Kein Modul>

Konfigurieren Sie die Steckplätze mit der API:

```
"controllogix_ethernet.DEVICE_SLOT_CONFIGURATION": { "controllogix_ether-
net.DEVICE_SLOT_CONFIGURATION_MODULE": [ 30, 1, 255, 62, 76, 255, 255, 255, 255,
255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255, 255, 255, 255, 255 ], "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_INPUT_
WORDS": [ 0, 1, 0, 1, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0 ], "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_OUTPUT_
WORDS": [ 2, 0, 0, 2, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0 ]
```

Geräteeigenschaften - Redundanz



Redundanz steht mit dem Plugin für Redundanz auf Medienebene zur Verfügung.
Weitere Informationen dazu erhalten Sie auf der Website, von einem Vertriebsrepräsentanten oder im Benutzerhandbuch.

Richtlinien für modulare E/A-Auswahl für SLC 500

In der folgenden Tabelle wird die Anzahl der Eingabe- und Ausgabewörter aufgelistet, die für jedes E/A-Modul in der Liste der Steckplatzkonfiguration zur Verfügung stehen.

Modul-ID	Modultyp	Eingabewörter	Ausgabewörter
0	1746-I*8 - Jedes einzelne 8-pt-Eingabemodul	1	0
1	1746-I*16 - Jedes einzelne 16-pt-Eingabemodul	1	0
2	1746-I*32 - Jedes einzelne 32-pt-Eingabemodul	2	0
3	1746-O*8 - Jedes einzelne 8-pt-Ausgabemodul	0	1
4	1746-O*16 - Jedes einzelne 16-pt-Ausgabemodul	0	1
5	1746-O*32 - Jedes einzelne 32-pt-Ausgabemodul	0	2
6	1746-IA4 - 4 Input 100/120 VAC	1	0
7	1746-IA8 - 8 Input 100/120 VAC	1	0
8	1746-IA16 - 16 Input 100/120 VAC	1	0
9	1746-IB8 - 8 Input (Senke) 24 VDC	1	0
10	1746-IB16 - 16 Input (Senke) 24 VDC	1	0
11	1746-IB32 - 32 Input (Senke) 24 VDC	2	0
12	1746-IG16 - 16 Input [TTL] (Quelle) 5 VDC	1	0
13	1746-IM4 - 4 Input 200/240 VAC	1	0
14	1746-IM8 - 8 Input 200/240 VAC	1	0
15	1746-IM16 - 16 Input 200/240 VAC	1	0
16	1746-IN16 - 16 Input 24 VAC/VDC	1	0
17	1746-ITB16 - 16 Input [Schnell] (Senke) 24 VDC	1	0
18	1746-ITV16 - 16 Input [Schnell] (Quelle) 24 VDC	1	0
19	1746-IV8 - 8 Input (Quelle) 24 VDC	1	0
20	1746-IV16 - 16 Input (Quelle) 24 VDC	1	0

Modul-ID	Modultyp	Eingabewörter	Ausgabewörter
21	1746-IV32 - 32 Input (Quelle) 24 VDC	2	0
22	1746-OA8 - 8 Output (TRIAC) 100/240 VAC	0	1
23	1746-OA16 - 16 Output (TRIAC) 100/240 VAC	0	1
24	1746-OB8 - 8 Output [Trans] (Quelle) 10/50 VDC	0	1
25	1746-OB16 - 16 Output [Trans] (Quelle) 10/50 VDC	0	1
26	1746-OB32 - 32 Output [Trans] (Quelle) 10/50 VDC	0	2
27	1746-OBP16 - 16 Output [Trans 1 amp] (SRC) 24 VDC	0	1
28	1746-OV8 - 8 Output [Trans] (Senke) 10/50 VDC	0	1
29	1746-OV16 - 16 Output [Trans] (Senke) 10/50 VDC	0	1
30	1746-OV32 - 32 Output [Trans] (Senke) 10/50 VDC	0	2
31	1746-OW4 - 4 Output [Relais] VAC/VDC	0	1
32	1746-OW8 - 8 Output [Relais] VAC/VDC	0	1
33	1746-OW16 - 16 Output [Relais] VAC/VDC	0	1
34	1746-OX8 - 8 Output [Isoliertes Relais] VAC/VDC	0	1
35	1746-OVP16 - 16 Output [Trans 1 amp] (Senke) 24VDC3	0	1
36	1746-IO4 - 2 In 100/120 VAC 2 Out [Rly] VAC/VDC3	1	1
37	1746-IO8 - 4 In 100/120 VAC 4 Out [Rly] VAC/VDC4	1	1
38	1746-IO12 - 6 In 100/120 VAC 6 Out [Rly] VAC/VDC	1	1
39	1746-NI4 - 4 Ch Analoge Eingabe	4	0
40	1746-NIO4I - Analog Comb 2 in & 2 Stromausgang	2	2
41	1746-NIO4V - Analog Comb 2 in & 2 Spannungsausgang	2	2
42	1746-NO4I - 4 Ch Analoger Stromausgang	0	4
43	1746-NO4V - 4 Ch Analoger Spannungsausgang	0	4
44	1746-NT4 - 4 Ch Thermoelement-Eingangsmodul	8	8
45	1746-NR4 - 4 Ch Rtd/Widerstand-Eingangsmodul	8	8
46	1746-HSCE - Hochgeschwindigkeitszähler/Encoder	8	1
47	1746-HS - Einzelachsen-Bewegungs-Controller	4	4
48	1746-OG16 - 16 Output [TLL] (SENKE) 5 VDC	0	1
49	1746-BAS - Basismodul 500 5/01 (Konfiguration)	8	8
50	1746-BAS - Basismodul 5/02 (Konfiguration)	8	8
51	1747-DCM - Direktkommunikationsmodul (1/4 Rack)	2	2
52	1747-DCM - Direktkommunikationsmodul (1/2 Rack)	4	4
53	1747-DCM - Direktkommunikationsmodul (3/4 Rack)	6	6
54	1747-DCM - Direktkommunikationsmodul (Volles Rack)	8	8
55	1747-SN - Remote-E/A-Scanner	32	32
56	1747-DSN - Verteilter E/A-Scanner - 7 Blöcke	8	8
57	1747-DSN - Verteilter E/A-Scanner - 30 Blöcke	32	32
58	1747-KE - Schnittstellenmodul, Reihe A	1	0
59	1747-KE - Schnittstellenmodul - Reihe B	8	8
60	1746-NI8 - 8 Ch Analoge Eingabe, Klasse 1	8	8
61	1746-NI8 - 8 Ch Analoge Eingabe, Klasse 3	16	12
62	0000-Generic Module	-	-
63	1746-IC16 - 16 Input (Senke) 48 VDC	1	0
64	1746-IH16 - 16 Input [Trans] (Senke) 125 VDC	1	0
65	1746-OAP12 - 12 Output [Triac] 120/240 VDC	0	1

Modul-ID	Modultyp	Eingabewörter	Ausgabewörter
66	1746-OB6EI - 6 Output [Trans] (Quelle) 24 VDC	0	1
67	1746-OB16E - 16 Output [Trans] (Quelle) Geschützt	0	1
68	1746-OB32E - 32 Output [Trans] (Quelle) 10/50 VDC	0	2
69	1746-OBP8 - 8 Output [Trans 2 amp] (Quelle) 24 VDC	0	1
70	1746-IO12DC - 6 Input 12 VDC, 6 Output [Rly	1	1
71	1746-INI4I - Analog 4 Ch. Isol. Stromeingang	8	8
72	1746-INI4VI - Analog 4 Ch. Isol. Spannungs-/Stromeingang	8	8
73	1746-INO4I Analog 4 Ch. Isol. Stromausgang	8	8
74	1746-INO4VI Analog 4 Ch. Isol. Volt/Stromausgang	8	8
75	1746-INT4 4 Ch. Isolierter Thermoelementeingang	8	8
76	1746-NT8 Analog 8 Ch - Thermoelementeingang	8	8
77	1746-HSRV - Bewegungssteuerungsmodul	12	8
78	1746-HSTP1 - Stepper-Controller-Modul	8	8
79	1747-MNET - MNET-Netzwerk-Kommunikationsmodul	0	0
80	1746-QS - Synchronisiertes Achsenmodul	32	32
81	1747-QV - Open Loop-Geschwindigkeitsregelung	8	8
82	1747-RCIF - Robot-Steuerung-Schnittstellenmodul	32	32
83	1747-SCNR - ControlNet-SLC-Scanner	32	32
84	1747-SDN - DeviceNet-Scanner-Modul	32	32
85	1394-SJT - GMC-Turbosystem	32	32
86	1203-SM1 - SCANport-Kommunikationsmodul - Einfach	8	8
87	1203-SM1 - SCANport-Kommunikationsmodul - Erweitert	32	32
88	AMCI-1561 AMCI Series 1561 Resolver Module	8	8

Leistungsoptimierungen

Wenn Sie weitere Informationen zur Optimierung auf Kommunikations- und Anwendungsebene erhalten möchten, wählen Sie eine Verknüpfung in der Liste unten aus.

Kommunikation optimieren
Anwendung optimieren
Leistungsstatistik und -optimierung
Beispiel für Leistungsoptimierung

Kommunikation optimieren

Wie bei jedem programmierbaren Controller gibt es viele Möglichkeiten, die Leistung und Systemkommunikation zu verbessern.

Protokollmodus

Der Protokollmodus bestimmt, wie vom Controller auf Logix-Tag-Daten zugegriffen wird. Es gibt drei Arten von Protokollmodi: "Symbolisch", "Logisch, nicht blockierend" und "Logisch, blockierend". Beschreibungen dazu sehen wie folgt aus:

- **Modus "Symbolisch":** Jede Client/Server-Tag-Adresse wird im Paket durch ihren ASCII-Zeichennamen dargestellt.
- **Modus "Logisch, nicht blockierend":** Jedes Client/Server-Tag wird durch seine logische Speicheradresse in der SPS dargestellt.
- **Modus "Logisch, blockierend":** Das Logix-Tag wird als einzelnes Datenelement aufgerufen. Jedes Client/Server-Tag (z.B. MYTIMER.ACC) weist ein entsprechendes Logix-Tag (MYTIMER) auf. Viele

Client/Server-Tags können wie im Fall von Strukturen zu demselben Logix-Tag gehören. Bei jedem Lesezyklus wird das Logix-Tag gelesen, sein Block im Treiber-Cache aktualisiert und alle Client/Server-Tags werden von diesem Cache aus aktualisiert.

Der logische, nicht blockierende Modus wird im Allgemeinen empfohlen, da es der effizienteste Modus zum Sammeln und Verarbeiten von Logix-Tag-Daten ist. Der symbolische Modus wird für Abwärtskompatibilität empfohlen, wohingegen der logische, nicht blockierende Modus für Projekte empfohlen wird, die eine kleine Anzahl von UDT-Referenzen und/oder Logix-Tags mit vordefinierter Struktur enthalten. Zwar kann der logische, blockierende Modus effizient sein, doch kann er auch der Leistung schaden, falls nicht korrekt verwendet. Weitere Informationen zu den Vor- und Nachteilen des jeweiligen Modus finden Sie unter Protokollmodus auswählen.

Tipps zur Tag-Einteilung
Benutzer sollten mindestens ein Gerät für logische, blockierende Zwecke und mindestens ein Gerät für logische, nicht blockierende Zwecke ausweisen. Dadurch wird die Leistung verbessert, da unterschiedliche Tags in einem Projekt für verschiedene Modi häufig besser geeignet sind. Bei Verwendung einer Tag-Einteilung sollten Benutzer Folgendes tun:

1. Weisen Sie Server-Tags, die unteilbare Logix-Tags (Array oder Nicht-Array) referenzieren, dem logischen, nicht blockierenden Gerät zu.
2. Weisen Sie Server-Tags, die ein Struktur-Logix-Tag referenzieren, das zu einem Drittel* oder weniger des Struktur-Tags besteht, den logischen, nicht blockierenden Geräten zu. Beispiel: Wenn es 55** oder weniger Mitglied-Tags gibt, die ein PID_ENHANCED-Logix-Tag referenzieren, müssen alle diese Tags dem logischen, nicht blockierenden Gerät zugewiesen werden.
3. Weisen Sie Server-Tags, die ein Struktur-Logix-Tag referenzieren, das zu einem Drittel* oder mehr des Struktur-Tags besteht, den logischen, blockierenden Geräten zu. Beispiel: Wenn es mehr als 55** Mitglied-Tags gibt, die ein PID_ENHANCED-Logix-Tag referenzieren, müssen alle diese Tags dem logischen, blockierenden Gerät zugewiesen werden.

*Ein Drittel ist kein exakter Grenzwert, sondern eher eine Zahl, die sich in mehreren Studien als wahr erwiesen hat.
**Eine PID_ENHANCED-Struktur weist 165 Tags auf, somit entspricht ein Drittel 55 Tags.

Verbindungsgröße

Das Erhöhen der Verbindungsgröße ermöglicht mehr Lese-/Schreibanforderungen pro Datenpaket, was für einen größeren Durchsatz sorgt. Zwar wird auch die CPU-Auslastung und die Durchlaufzeit der Antwort erhöht, doch verbessert sich die Leistung beträchtlich. Die Eigenschaft "Verbindungsgröße" kann ggf. nur in den ControlLogix 5500- und CompactLogix 5300-Gerätemodellen geändert werden. Weitere Informationen dazu finden Sie unter Logix-Kommunikationsparameter.

UDT-Substruktur-Aliasing

Wenn ein UDT große Substrukturen enthält und ein Drittel oder mehr der Substruktur-Mitglieder im Client referenziert sind, beziehen Sie sich auf die folgenden Anweisungen zur Optimierung von Lesevorgängen für die Substruktur.

1. Erstellen Sie einen Alias der Substruktur in RSLogix 5000. Weisen Sie anschließend Server-Tags, die den Rest der UDT-Substruktur referenzieren, einem logischen, blockierenden Gerät zu.
2. Als Nächstes weisen Sie die Server-Tags, die den Rest des UDT (jedoch nicht der Substruktur) referenzieren, einem logischen, nicht blockierenden Gerät zu.

System Overhead Time Slice (SOTS, Systemaufwand-Zeitabschnitt)

SOTS ist der Prozentsatz der Zeit, die für die Durchführung von Kommunikationsaufgaben (beispielsweise OPC-Treiber-Kommunikation) in RSLogix 5000 festgelegt ist. 100 % SOTS ist der Prozentsatz der Zeit für Controller-Aufgaben (z.B. Kontaktplan). Der Standard-SOTS beträgt 10 %. Bei jedem erfolgten Programm-Scan von 10 ms verbringt der Controller 1 ms mit der Verarbeitung von Treiberanforderungen (falls der Controller eine fortlaufende Aufgabe hat). Der SOTS-Wert definiert die Priorität der Aufgabe. Wenn Controller-Aufgaben hohe Priorität aufweisen, sollte SOTS unter 30 % eingestellt werden. Wenn die Kommunikationsaufgaben hohe Priorität aufweisen, sollte SOTS auf oder über 30 % eingestellt werden. Für eine optimale Balance aus Kommunikationsleistung und CPU-Auslastung sollte SOTS zwischen 10 und 40 % eingestellt werden.

Mehrfachanforderungspakete

ControlLogix-Ethernet-Treiber von Allen-Bradley wurde zur Optimierung von Lese- und Schreibvorgängen entwickelt. Bei Nicht-Array-, Nicht-String-Tags (die nur ein Element anfordern) werden Anforderungen in einer einzigen Transaktion blockiert. Dies sorgt für eine drastische Verbesserung der Leistung gegenüber Einzel-Tag-Transaktionen. Die einzige Beschränkung ist die Anzahl Datenbyte, die in eine einzige Transaktion passen können.

Wichtig: Im symbolischen Modus wird der ASCII-Zeichenfolgenwert jedes Tags in das Anforderungspaket eingefügt, bis keine weiteren Tag-Anforderungen mehr passen. Benutzer sollten für eine optimale Leistung die Größe der Tag-Namen auf ein Minimum beschränken. Je kleiner der Tag-Name, desto mehr Tags passen in eine einzelne Transaktion und desto weniger Transaktionen sind zur Verarbeitung aller Tags erforderlich.

Array-Elemente blockiert (nur Modi "Symbolisch" und "Logisch, nicht blockierend")

Um das Lesen unteilbarer Array-Elemente zu optimieren, lesen Sie einen Array-Block in einer einzelnen Anfrage statt individuell. Je mehr Elemente in einem Block gelesen werden, desto größer die Leistung. Da der Transaktionsaufwand und die -verarbeitung die meiste Zeit beansprucht, führen Sie so wenige Transaktionen wie möglich aus, während Sie so viele gewünschte Tags wie möglich scannen. Dies ist wesentlich für das Blockieren von Array-Elementen.

Blockgrößen werden als eine Elementanzahl angegeben. Eine Blockgröße von 120 Elementen bedeutet, dass maximal 120 Array-Elemente bei einer Anforderung gelesen werden. Die maximale Blockgröße beträgt 3840 Elemente. Boolean-Arrays werden unterschiedlich behandelt: Im Protokoll ist ein Boolean-Array ein 32-Bit-Array. Somit fordert das anfordernde Element 0 die Bits 0 bis 31 an. Um bei der Diskussion Konsistenz beizubehalten, wird ein Boolean-Array-Element als einzelnes Bit angesehen. Zusammenfassend betrachtet, sieht die maximale Anzahl von Array-Elementen (basierend auf Blockgröße 3840), die angefordert werden können, wie folgt aus: 122880 BOOL, 3840 SINT, 3840 INT, 3840 DINT, 3840 REAL, 3840 LINT, 3840 UINT, 3840 USINT, 3840 ULINT und 3840 LREAL.

Wie unter Logix-Kommunikationsparameter erörtert, ist die Blockgröße anpassbar und sollte je nach aktuellem Projekt ausgewählt werden. Beispiel: Wenn die Array-Elemente 0-26 und das Element 3839 zu lesende Tags sind, ist das Verwenden einer Blockgröße von 3840 nicht nur zu viel Aufwand, sondern auch schädlich für die Leistung des Treibers. Dies liegt daran, dass alle Elemente zwischen 0 und 3839 bei jeder Anforderung gelesen werden, selbst wenn nur 28 dieser Elemente von Bedeutung sind. In diesem Fall ist eine Blockgröße von 30 geeigneter. Die Elemente 0-26 würden in einer Anforderung bedient und das Element 3839 in der nächsten Anforderung.

Zeichenfolgen optimieren

In den logischen Adressierungsmodi würde beim Schreiben in STRING.DATA auch in STRING.LEN geschrieben, und zwar mit dem richtigen Längenwert.

Zeichenfolgendaten bei LEN beenden

In diesem Treiber sind Zeichenfolgen-Tags Strukturen mit separaten Zeichendaten- und Längenkomponenten. Dementsprechend liest der Treiber ein Zeichenfolgen-Tag automatisch in zwei Transaktionen: eine im logischen Protokollmodus für die Zeichenfolgen-Zeichendaten (DATA) und eine im symbolischen Modus für die Zeichenfolgenlänge (LEN). Wenn die Option Zeichenfolgendaten bei LEN beenden deaktiviert ist, erfolgt eine einzige Transaktion zum Lesen der Zeichenfolgen-Zeichendaten. In diesem Fall wird der Lesevorgang der Zeichenfolgenlänge im symbolischen Modus umgangen. Bei einem Projekt mit vielen Zeichenfolgen-Tags kann dies beträchtlich die Zeit reduzieren, die zum Lesen aller Tags benötigt wird.

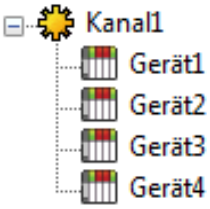
Weitere Informationen zur Option Zeichenfolgendaten bei LEN beenden finden Sie unter Logix-Optionen.

Anwendung optimieren

ControlLogix-Ethernet-Treiber von Allen-Bradley wurde dafür konzipiert, eine optimale Leistung mit der geringsten Auswirkung auf die Gesamtleistung des Systems zu bieten. Zwar ist ControlLogix-Ethernet-Treiber von Allen-Bradley schnell, doch gibt es eine Reihe von Richtlinien zur Optimierung der Anwendung und zum Erreichen der maximalen Leistung.

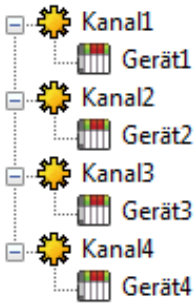
Der Server bezeichnet Kommunikationsprotokolle wie ControlLogix-Ethernet von Allen-Bradley als Kanal. Jeder in der Anwendung definierte Kanal stellt einen separaten Ausführungspfad im Server dar. Sobald ein Kanal festgelegt wurde, muss eine Reihe von Geräten unter diesem Kanal definiert werden. Jedes dieser Geräte stellt eine einzelne Logix-CPU von Allen-Bradley dar, von der Daten gesammelt werden. Zwar

ermöglicht diese Methode zum Definieren der Anwendung ein hohes Leistungsniveau, doch kann damit nicht vollständig Nutzen aus ControlLogix-Ethernet-Treiber von Allen-Bradley oder dem Netzwerk gezogen werden. Ein Beispiel dafür, wie die Anwendung bei Konfiguration mit einem Kanal möglicherweise aussehen kann, wird im Folgenden gezeigt.



Jedes Gerät wird unter einem einzelnen Kanal namens"Ethernet_Channel1" angezeigt. In dieser Konfiguration bewegt sich der Treiber schnellstmöglich von einem Gerät zum nächsten, um Informationen in einem effektiven Intervall zu sammeln. Je mehr Geräte hinzugefügt oder je mehr Informationen von einem einzelnen Gerät angefordert werden, desto mehr leidet die Aktualisierungsrate insgesamt.

Wenn der Treiber nur einen Kanal definieren könnte, würde das Beispiel oben die einzige verfügbare Option darstellen. Der Treiber kann jedoch bis zu 1024 Kanäle definieren. Durch Verwenden mehrerer Kanäle wird die Arbeitsbelastung bei der Datensammlung verteilt, indem mehrere Anfragen gleichzeitig an das Netzwerk gestellt werden. Ein Beispiel dafür, wie dieselbe Anwendung aussehen kann, wenn sie mit mehreren Kanälen konfiguriert wird, um die Leistung zu verbessern, wird im Folgenden gezeigt.



Jedes Gerät wurde jetzt unter seinem eigenen Kanal festgelegt. In dieser neuen Konfiguration wird ein einziger Ausführungspfad dediziert für das Sammeln von Daten von jedem Gerät eingesetzt. Wenn die Anwendung über weniger Geräte verfügt, kann sie wie hier gezeigt optimiert werden.

Die Leistung wird verbessert, selbst wenn die Anwendung über mehr Geräte verfügt. Zwar sind weniger Geräte möglicherweise ideal, doch zieht die Anwendung weiterhin Nutzen aus zusätzlichen Kanälen. Obwohl das Verteilen der Gerätelast auf alle Kanäle zur Folge hat, dass sich der Server erneut von Gerät zu Gerät bewegt, kann dies nun mit weit weniger zu verarbeitenden Geräten auf einem einzigen Kanal erfolgen.

Leistungsstatistik und -optimierung

Die Funktion "Leistungsstatistik" liefert Benchmarks und Statistiken über die Leistung der Anwendung. Da die Leistungsstatistik eine zusätzliche Verarbeitungsebene darstellt, kann es die Leistung des Servers beeinflussen. Sie ist standardmäßig deaktiviert. Wenn Sie die Funktion **Leistungsstatistik** aktivieren möchten, öffnen Sie die Geräteeigenschaften und wählen Sie **Logix-Optionen** aus.

Leistungsstatistiktypen

Die Leistungsstatistik stellt aussagekräftige numerische Ergebnisse für drei Bereiche bereit: Gerät, Kanal und Treiber. Es folgen Beschreibungen der Typen:

- **Gerät:** Diese Statistik stellt die Datenzugriffsleistung auf einem bestimmten Gerät bereit.
- **Kanal:** Diese Statistik stellt die durchschnittliche Datenzugriffsleistung für alle Geräte in einem bestimmten Kanal mit aktivierter Leistungsstatistik bereit.
- **Treiber:** Diese Statistik stellt die durchschnittliche Datenzugriffsleistung für alle Geräte bereit, die ControlLogix-Ethernet-Treiber von Allen-Bradley mit aktivierter Leistungsstatistik verwenden.

Statistiktyp auswählen

Der erforderliche Statistiktyp hängt von der Anwendung ab. Im Allgemeinen stellt die Treiberstatistik eine echte Messgröße der Anwendungsleistung bereit, wohingegen Kanal- und Gerätestatistik beim Optimieren der Anwendung am wichtigsten sind. Beispiel: Wird das Verschieben 10 bestimmter Tags von Gerät A auf Gerät B die Leistung von Gerät A erhöhen? Wird das Verschieben des Geräts A von Kanal 1 auf Kanal 2 die Leistung von Kanal 1 erhöhen? Diese Fragen stellen gute Beispiele für Situationen dar, in denen Geräte- und Kanalstatistik verwendet werden sollten.

Statistik finden

Die Serverstatistik wird beim Herunterfahren an das Ereignisprotokoll des Servers ausgegeben. Um die Ergebnisse anzuzeigen, fahren Sie den Server herunter und starten Sie ihn neu.

Unterschiede zwischen Server- und Leistungsstatistik

Die Leistungsstatistik stellt die Zusammensetzung der durchgeführten Lesevorgangstypen (beispielsweise symbolisch im Vgl. zu Symbolinstanz im Vgl. zu physisch oder Geräte-Lesevorgänge im Vgl. zu Cache-Lesevorgängen) bereit, wohingegen die Serverstatistik einen allgemeinen Lese-Anzahl-Wert liefert.

Anwendung für erhöhte Leistung optimieren

Informationen zum Erhöhen der Geräte- und Kanalstatistikergebnisse finden Sie in den Anweisungen unten. Weitere Informationen dazu finden Sie unter Kommunikation optimieren.

- Server-Tags, die unteilbare Logix-Tags (Array oder Nicht-Array) referenzieren, sollten logischen, nicht blockierenden Geräten zugewiesen werden.
- Server-Tags, die ein Struktur-Logix-Tag referenzieren, das aus einem Drittel oder weniger des Struktur-Tags besteht, sollten logischen, nicht blockierenden Geräten zugewiesen werden.
- Server-Tags, die ein Struktur-Logix-Tag referenzieren, das aus einem Drittel oder mehr des Struktur-Tags besteht, sollten logischen, blockierenden Geräten zugewiesen werden.
- Wenn der symbolische Modus verwendet wird, sollte für Logix-Namen eine minimale Länge beibehalten werden.
- Logix-Arrays sollten so oft wie möglich verwendet werden.
- Nur die notwendige Menge des SOTS (System Overhead Time Slice, Systemaufwand-Zeitabschnitt) für Kontaktplan/FBD sollte zugeordnet werden, um den Rest für die Treiberkommunikation zu überlassen.
- Bei Projekten, die eine große Anzahl von Zeichenfolgen-Tags im logischen Modus lesen, deaktivieren Sie unter **Logix-Optionen** in den **Geräteeigenschaften** die Option **Zeichenfolgendaten bei LEN beenden**.

Informationen zum Erhöhen der Treiberstatistikergebnisse finden Sie in den Anweisungen unten. Weitere Informationen dazu finden Sie unter Anwendung optimieren.

- Geräte sollten über Kanäle hinweg verteilt werden. Mehr als ein Gerät sollte nicht in einem Kanal platziert werden (sofern nicht erforderlich).
- Die Arbeitslast sollte gleichmäßig auf die Geräte verteilt werden. Ein einzelnes Gerät sollte nicht überladen werden (sofern nicht erforderlich).
- Dasselbe Logix-Tag sollte nicht für verschiedene Geräte referenziert werden.

Hinweis: Zwar können diese allgemeinen Regeln die Leistung optimieren, doch hängt sie letztendlich von der Anwendung ab. Das Scan-Intervall kann Ergebnisse verdecken: Wenn Tag-Anforderungen gelegentlich erfolgen, können Lese- und Schreibtransaktionen abgeschlossen werden, bevor die nächste Anforderung eingeht. In diesem Fall weisen "Logisch, blockierend" und "Logisch, nicht blockierend" dieselben Leistungsstatistikergebnisse auf. Wenn Tag-Anforderungen hoch sind (viele Tags oder hohe Scan-Intervalle), benötigt es möglicherweise mehr Zeit bis zum Abschluss der Transaktion. Dies ist der Fall, wenn die Stärken und Schwächen von "Logisch, blockierend" und "Logisch, nicht blockierend" offensichtlich werden. Mit der Leistungsstatistik kann die Anwendung auf optimale Leistung eingestellt werden. Ein Beispiel dazu finden Sie unter Beispiel für Leistungsoptimierung.

Beispiel für Leistungsoptimierung

Statistiken können auf jede Anwendung angewendet werden. Im Beispiel unten wird der Quick Client im Leistungsoptimierungsprozess eingesetzt. Es geht darum, dass alle im Projekt verwendeten Tags gleichzeitig und mit schnellem Scan-Intervall gelesen werden. Zwar ist dies nicht realistisch, doch liefert es einen exakten Benchmark zum Projektlayout im Server (Tags, die zu bestimmten Geräten gehören, Geräte, die zu bestimmten Kanälen gehören usw.).

Die erfassten Statistiken sind relativ. Benutzer sollten mit einem Serverprojektlayout beginnen, die Statistiken erfassen und dann optimieren. Es wird empfohlen, mehr als einen Probelauf zu machen, um die Ergebnisse für ein gegebenes Layout richtig bewerten zu können. Sobald das effizienteste Layout bestimmt wurde, kann die Client-Anwendung mit der Gewissheit konzipiert werden, dass der Server optimal ist.

Mit dem Quick Client abgerufene Leistungsergebnisse gleichen nicht den mit einer Client-Anwendung abgerufenen Leistungsergebnissen: Mehrere Faktoren sorgen für Abweichungen. Zwar ist die Leistungsoptimierung mit der Client-Anwendung präziser als mit dem Quick Client, doch wirkt sich die erforderliche Optimierung nicht nur auf das Serverprojekt aus, sondern auch auf die Client-Anwendung. Es wird empfohlen, den Quick Client zur Optimierung der Anwendung zu nutzen, bevor die Client-Anwendung entwickelt wird.

Hinweis: Beim unten beschriebenen Optimierungsprozess wird davon ausgegangen, dass alle Tags mit einem schnellen Scan-Intervall gelesen werden. Schreibvorgänge beeinträchtigen die Leistung.

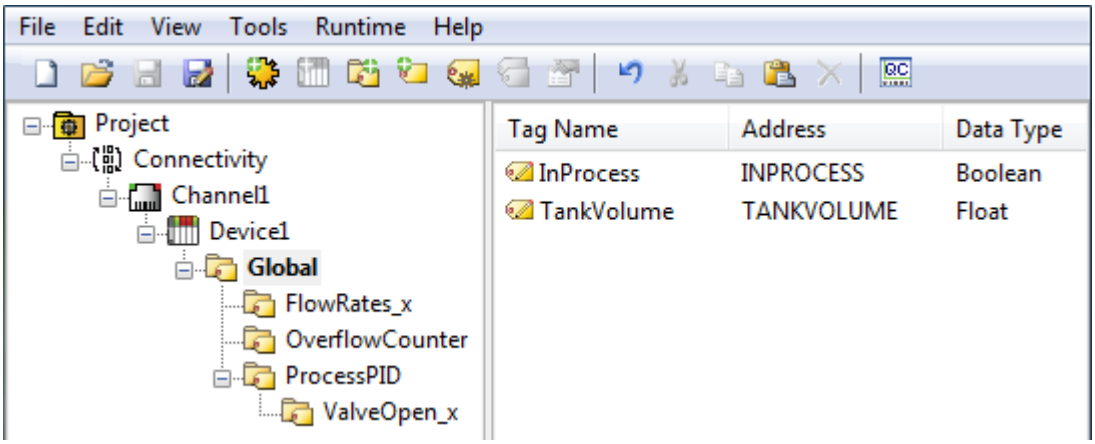
1. Beim unten angezeigten Controller-Projekt steht Folgendes zur Verfügung:

   2 unteilbare Elemente
   1 unteilbares Array
   1 UDT
   1 UDT-Array
   1 vordefinierter Typ

   Hinweis: Overhead Time Slice (OTS) = 10 %.

	P	Tag Name	Alias For	Base Tag	Type
		InProcess			BOOL
	☐	⊞-OverflowCounter			COUNTER
	☐	⊞-ValveOpen			DINT[10]
	☐	⊞-ProcessPID			PIDLoop
	☐	⊞-FlowRates			ProcessVariable[2]
▶	☐	TankVolume			REAL
✳	☐				

2. Nach der Durchführung der automatischen Tag-Datenbankgenerierung von diesem Controller aus erzeugt der Server das folgende Projekt.



   Hinweis: Die Tag-Gruppe "Global" enthält 130 Tags.

3. Um die Vorteile der Tag-Einteilung zu verdeutlichen, werden in diesem Beispiel nicht alle Tags referenziert. Mehr als ein Drittel der ProcessPID-Tags, weniger als ein Drittel der FlowRates-Tags und alle anderen Tags werden referenziert. Dementsprechend liegt die neue Tag-Anzahl bei 105.

4. Bereiten Sie den Client auf den Test vor. Dazu starten Sie den Quick Client von der Serveranwendung aus, indem Sie auf das QuickClient-Symbol klicken.

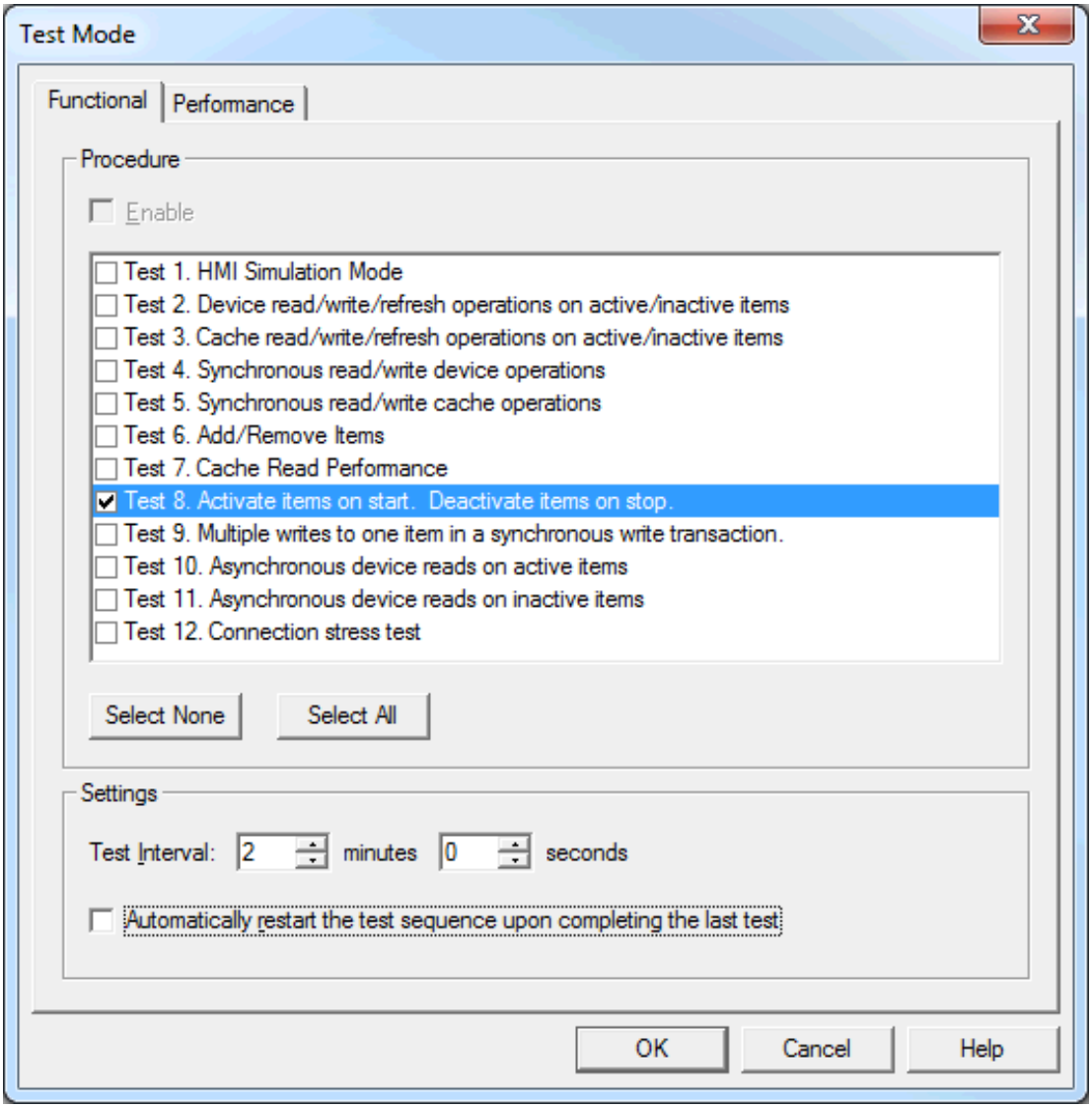
5. Sobald das Projekt geladen wurde, entfernen Sie alle Gruppen mit Ausnahme derjenigen, die Tags von Interesse enthalten. Statistik- und System-Tags werden zum Beispiel nicht benötigt.

   Hinweis: Legen Sie für kleine Projekte die Gruppe **Aktualisierungsintervall** auf 0-10 ms fest. Legen Sie das Intervall für große Projekte auf 10-50 ms fest.

6. Wählen Sie **Tools | Test Mode** aus.

7. Aktivieren Sie **Test 8. Activate items on start. Deactivate items on stop.** Legen Sie anschließend ein Testintervall fest.

   Hinweis: Da dieses Projekt ziemlich klein ist, wurde das Intervall auf 2 Minuten festgelegt. Bei größeren Projekten sollte das Intervall erhöht werden, um ein präziseres Lesen zu ermöglichen.



8. Kehren Sie zu **Tools | Test Mode** zurück, und deaktivieren Sie den Testmodus. Alle Tags müssen deaktiviert sein.
9. Trennen Sie den Quick Client, damit die Zeiten-Probeläufe beginnen können.
10. Fahren Sie den Server herunter.
11. Starten Sie den Server, und legen Sie unter den Geräteeigenschaften den **Protokollmodus** auf **Logisch, blockierend** fest.
12. Setzen Sie **Leistungsstatistik** auf "Ja".

13. Stellen Sie mit dem Quick Client eine Verbindung mit dem Server her. Wählen Sie **Tools | Test Mode** aus. Aktivieren Sie den Testmodus.

    Hinweis: Das Lesen von Daten beginnt. Sobald das Testintervall abgelaufen ist, werden alle Tags deaktiviert und der Treiber beendet die Statistikerfassung. Die Ergebnisse können dann angezeigt werden.

14. Trennen Sie den Quick Client vom Server und fahren Sie den Server anschließend herunter.

15. Starten Sie den Server neu, und durchsuchen Sie das Ereignisprotokoll nach Statistiken. In der Abbildung unten wird der erste Probelauf angezeigt, bei dem "Logisch, blockierend" für das Gerät verwendet wird.

```
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5972.26
```

Hinweis: In der Abbildung unten wird der erste Probelauf angezeigt, bei dem "Logisch, blockierend" für den Kanal und Treiber verwendet wird.

```
DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 5972.26
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 5972.26
```

Hinweis: Dies ist der Steuerungssatz für Vergleiche.

16. Legen Sie auf dem Server den **Protokollmodus** auf **Logisch, nicht blockierend fest**.
17. Verbinden Sie den Server mit dem Quick Client. Wählen Sie **Tools | Test Mode** aus, und aktivieren Sie den Testmodus.

    Hinweis: Das Lesen von Daten beginnt. Sobald das Testintervall abgelaufen ist, werden alle Tags deaktiviert und der Treiber beendet die Statistikerfassung. Die Ergebnisse können dann angezeigt werden.

18. Trennen Sie den Quick Client vom Server und fahren Sie den Server anschließend herunter.
19. Starten Sie den Server neu, und durchsuchen Sie das zugehörige Ereignisprotokoll nach Statistiken. In der Abbildung unten wird der zweite Probelauf angezeigt, bei dem "Logisch, nicht blockierend" für das Gerät verwendet wird.

```
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 3705.23
```

Hinweis: In der Abbildung unten wird der zweite Probelauf angezeigt, bei dem "Logisch, nicht blockierend" für den Kanal und Treiber verwendet wird.

```
DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 3705.23
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 3705.23
```

20. Legen Sie auf dem Server den **Protokollmodus** auf **Symbolisch** fest, um festzustellen, wie die Leistung vor ControlLogix-Ethernet-Treiber von Allen-Bradley Version 4.6.0.xx war.
21. Stellen Sie mit dem Quick Client eine Verbindung mit dem Server her. Wählen Sie anschließend die Optionsfolge **Tools | Test Mode**, und aktivieren Sie den Testmodus.

    Hinweis: Das Lesen von Daten beginnt. Sobald das Testintervall abgelaufen ist, werden alle Tags deaktiviert und der Treiber beendet die Statistikerfassung. Die Ergebnisse können dann angezeigt werden.
22. Trennen Sie den Quick Client vom Server und fahren Sie den Server anschließend herunter.
23. Starten Sie den Server neu, und durchsuchen Sie das Ereignisprotokoll nach Statistiken. In der Abbildung unten wird der dritte Probelauf angezeigt, bei dem "Symbolisch" für das Gerät verwendet wird.

```
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 777.442
```

In der Abbildung unten wird der dritte Probelauf angezeigt, bei dem "Symbolisch" für den Kanal und Treiber verwendet wird.

```
DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 777.442
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 777.442
```

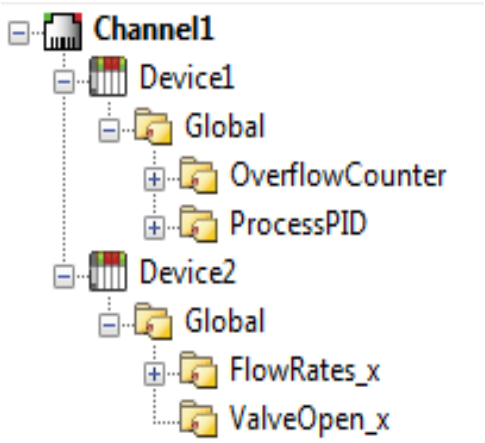
Hinweis: Der Modus "Logisch, blockierend" scheint für die angegebene Anwendung am optimalsten zu sein.

Kanalkommunikation optimieren

Die Kanalkommunikation kann optimiert werden, indem Tags für "Logisch, blockierend" in ein Gerät und Tags für "Logisch, nicht blockierend" in ein anderes Gerät verschoben werden. Dies wird als Tag-Einteilung bezeichnet.

Logisch, blockierend (Gerät 1)
ProcessPID
OverflowCounter

Logisch, nicht blockierend (Gerät 2)
FlowRate
ValveOpen
InProcess
Tank Volume (Kernvolumen)



1. Wiederholen Sie die Schritte 4 bis 15. Achten Sie bei Schritt 11 darauf, dass für Gerät 1 der Modus "Logisch, blockierend" und für Gerät 2 "Logisch, nicht blockierend" festgelegt ist.
2. Starten Sie den Server, und durchsuchen Sie das Ereignisprotokoll des Servers nach Statistiken. In der Abbildung unten wird der vierte Probelauf angezeigt, bei dem die Tag-Einteilung für das Gerät verwendet wird.

```
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 6933
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 76
Total tags read = 630979, Elapsed Time = 119782 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5268.43
DEVICE Channel1:Device2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 27732
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 104044, Elapsed Time = 119969 ms
DEVICE Channel1:Device2 PERFORMANCE: AvgTagReadPerSec = 867.373
```

Hinweis: In der Abbildung unten wird der vierte Probelauf angezeigt, bei dem die Tag-Einteilung für den Kanal und Treiber verwendet wird.

```
DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 34665
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 735023, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 6126.77
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 34665
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 735023, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 6126.77
```

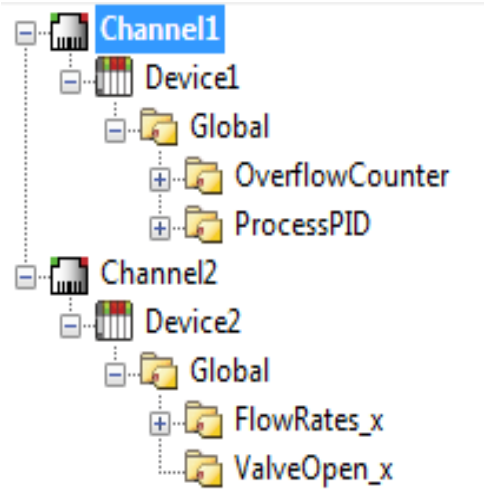
Hinweis: Die einzelnen Gerätestatistiken sehen nicht eindrucksvoll aus, weil die beiden Geräte auf separaten Statistikzählern ausgeführt werden. Das Entscheidende dieses Tests besteht darin, dass die Kanal- und Treiberstatistiken besser (6126) als bei Verwendung eines Kanals/eines Geräts mit "Logisch, blockierend" (5972) oder "Logisch, nicht blockierend" (3705) sind.

Anwendung optimieren

Die Anwendung kann optimiert werden, indem das Gerät 1 in einen Kanal verschoben wird und das Gerät 2 in einen anderen.

Logisch, blockierend (Channel1.Device 1)
ProcessPID
OverflowCounter

Logisch, nicht blockierend (Channel2.Device 2)
FlowRate
ValveOpen
InProcess
Tank Volume (Kernvolumen)



1. Wiederholen Sie die Schritte 4 bis 15. Achten Sie bei Schritt 11 darauf, dass für Channel1.Device1 der Modus "Logisch, blockierend" und für Channel2.Device2 "Logisch, nicht blockierend" festgelegt ist.
2. Starten Sie den Server, und durchsuchen Sie das Ereignisprotokoll des Servers nach Statistiken. In der Abbildung unten wird der fünfte Probelauf angezeigt, bei dem das Logix-Tag mit mehreren Kanälen für Channel 1.Device1 verbunden ist.

```
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 7271
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 661741, Elapsed Time = 119968 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 5517.4
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 7271
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 661741, Elapsed Time = 119968 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5517.4
```

Hinweis: In der Abbildung unten wird der vierte Probelauf angezeigt, bei dem das Logix-Tag für Channel2.Device2 verwendet wird.

```
CHANNEL Channel2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 29120
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 109205, Elapsed Time = 119968 ms
CHANNEL Channel2 PERFORMANCE: AvgTagReadPerSec = 910.52
DEVICE Channel2:Device2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 29120
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 109205, Elapsed Time = 119968 ms
DEVICE Channel2:Device2 PERFORMANCE: AvgTagReadPerSec = 910.52
```

Hinweis: In der Abbildung unten wird der vierte Probelauf angezeigt, bei dem die Tag-Einteilung für den Treiber verwendet wird.

```
DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 36391
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 84
Total tags read = 770946, Elapsed Time = 119968 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 6426.26
```

Ergebnis

Serverprojektlayout	Treiberleistung (Lesevorgänge/Sekunde)	Verbesserung gegenüber "Symbolisch"
Einzelkanal Einzelgerät mit "Logisch, blockierend"	5972	768 %
Einzelkanal Einzelgerät mit "Logisch, nicht blockierend"	3705	476 %
Einzelkanal Einzelgerät mit "Symbolisch"	777	k.A.
Einzelkanal Mehrere Geräte mit Tag-Einteilung	6126	788 %
Mehrere Kanäle	6426	827 %

Serverprojektlayout	Treiberleistung (Lesevorgänge/Sekunde)	Verbesserung gegenüber "Symbolisch"
Mehrere Geräte mit Tag-Einteilung		

Schlussfolgerungen

Das Projekt hat mit einem Einzelkanal und einem Einzelgerät begonnen. Dies ist das Standardverhalten für einen einzelnen Controller. Alle Tags wurden von diesem Controller in dieses channel.device importiert. Alle drei Protokollmodi wurden anschließend getestet, um festzustellen, welcher die beste Leistung ermöglichen würde. In diesem Fall war das Protokoll "Logisch, blockierend" das Beste. Das beste Protokoll ist von der jeweiligen Anwendung abhängig. Wenn die Leistung entscheidend ist, lohnt es sich, Probeläufe für "Logisch, blockierend" und "Logisch, nicht blockierend" durchzuführen, um den besten Protokollmodus für die Anwendung zu bestimmen. Das Protokoll "Symbolisch" ist nicht erforderlich, weil es niemals die Leistungsbereiche der beiden anderen Protokollmodi erreicht. Es wird hier als Beispiel gezeigt.

Es wurden anhand der unter Kommunikation optimieren beschriebenen Tipps Maßnahmen zur Optimierung der Kommunikation ergriffen. Dabei soll besonders hervorgehoben werden, dass die Tag-Einteilung verwendet wurde, um Tags des Typs "Logisch, blockierend" in ein Gerät (dem "Logisch, blockierend" zugewiesen ist) und Tags des Typs "Logisch, nicht blockierend" in Gerät (dem "Logisch, nicht blockierend" zugewiesen ist) einzufügen. Außerdem haben sich beide Geräte im selben Kanal befunden. In den Ergebnissen zeigte sich eine Verbesserung gegenüber "Logisch, blockierend" auf einem Einzelgerät. Dies liegt daran, dass einige Tags für einen bestimmten Protokollmodus optimaler eingesetzt werden können als für einen anderen. Beispiel: Beim Lesen von COUNTER insgesamt hat "Logisch, blockierend" Vorteile gegenüber "Logisch, nicht blockierend" da es viel schneller ist, den COUNTER blockweise statt einzelne Mitglieder zu lesen.

Es wurden auch Maßnahmen zur Optimierung der Anwendung ergriffen, indem Geräte in ihren eigenen Kanal platziert wurden. Unter Verwendung der im vorherigen Probelauf erstellten Geräte wurde ein Gerät "Logisch, blockierend" in einen und ein Gerät "Logisch, nicht blockierend" in einen anderen Kanal platziert. In den Ergebnissen zeigten sich Verbesserungen gegenüber dem Szenario mit einem Einzelkanal und mehreren Geräten aus dem vorherigen Probelauf. Dies untermauert die Vermutung, dass die Leistung verbessert werden kann, wenn wenige Geräte pro Kanal und so viele Kanäle wie erforderlich verfügbar sind.

Nach dem Einsatz dieser drei Optimierungsmethoden weist das Projekt einen Leistungszuwachs von 827 % gegenüber der ControlLogix-Ethernet-Treiber von Allen-Bradley Version vor 4.6.0.xx auf. Die Tag-Einteilung und mehrere Kanäle haben die Leistung um 107 % verbessert. Der Leistungszuwachs ist bei größeren Projekten auffälliger.

Datentypbeschreibung

Datentypen	Description
Boolean	Einzelnes Bit
Byte	8-Bit-Wert ohne Vorzeichen
Char	8-Bit-Wert mit Vorzeichen
Word	16-Bit-Wert ohne Vorzeichen
Short	16-Bit-Wert mit Vorzeichen
DWord	32-Bit-Wert ohne Vorzeichen
Long	32-Bit-Wert mit Vorzeichen
BCD	BCD mit zwei Byte gepackt, vier Dezimalstellen
LBCD	BCD mit vier Byte gepackt, acht Dezimalstellen
Float	32-Bit-IEEE-Gleitkommazahl
Double	64-Bit-IEEE-Gleitkommazahl
Datum	64-Bit-Datum/Uhrzeit
Zeichenfolge	Mit Null beendetes Zeichen-Array

Eine Beschreibung der plattformspezifischen Logix-Datentypen finden Sie unter Erweiterte Logix-Adressierung.

Spezifische Informationen zu Änderungen an den Boolean-Arrays in der Firmware V30 finden Sie in diesem Wissensdatenbank-Artikel.

Bedingungen für Standard-Datentyp

Client/Server-Tags wird der Standard-Datentyp zugewiesen, wenn eine oder mehrere der folgenden Bedingungen auftreten:

1. Ein dynamisches Tag wird auf dem Client mit "Nativ" als seinem zugewiesenen Datentyp erstellt.
2. Ein statisches Tag wird auf dem Server mit "Standard" als seinem zugewiesenen Datentyp erstellt.
3. Bei der automatischen Tag-Generierung offline, wenn ein unbekannter Datentyp in der L5K/L5X-Datei für UDT-Mitglieder und Alias-Tags festgestellt wurde.
4. Bei der automatischen Tag-Generierung offline, wenn ein Alias des folgenden Typs in der L5K/L5X-Datei festgestellt wurde:
   a. Alias eines Alias-Namens.
   b. Alias eines E/A-Modul-Tags mit keinem Bit innerhalb von Word/DWord. Beispiel: Wenn das Tag "AliasTag" das E/A-Modul-Tag "Local:5:C.ProgToFaultEn" @ BOOL referenziert, kann der Datentyp für "AliasTag" nicht aufgelöst werden, sodass ihm dieser Standardtyp zugewiesen wird. Wenn andererseits "AliasTag" das E/A-Modul-Tag "Local:5:C.Ch0Config.RangeType.0" @ BOOL referenziert, kann der Datentyp aufgrund des .(dot)-BIT, das ihn als Bit innerhalb von Word/DWord definiert, aufgelöst werden. Alias-Namen von E/A-Modul-Tags mit Bit innerhalb von Word/DWord werden automatisch dem Datentyp "Boolean" zugewiesen.

Hinweise:

1. Wenn Standard ausgewählt ist, ruft der Treiber den Datentyp des Logix-Tags vom Controller ab, wenn ein Client dynamisch auf ein Tag zugreift und dem Element nicht explizit einen Datentyp zuweist. Beispiel: Im Controller ist das Tag "MyTag" mit dem Datentyp REAL vorhanden. Das entsprechende Client-Element wird als "Channel1.Device1.MyTag" ohne zugewiesenen Datentyp angegeben. Wenn Standard als Standard-Datentyp auf dem Server ausgewählt ist, liest der Treiber "MyTag" vom Controller und ermittelt REAL in der Antwort, in der dem Client der Datentyp "Float" bereitgestellt wird.
2. Da es sich bei der Mehrheit der E/A-Modul-Tags um keine Tags mit Bit innerhalb von Word/Dword handelt, wird empfohlen, dass für den Standardtyp der im ACD-Projekt beobachtete Mehrheits-Datentyp festgelegt wird. Beispiel: Wenn es sich bei 75 % der Alias-E/A-Modul-Tags um INT-Tags handelt, legen Sie für den Standardtyp INT fest.

Adressbeschreibungen

Adressspezifikationen sind je nach verwendetem Modell unterschiedlich. Adressinformationen für das entsprechende Modell finden Sie in der Tabelle unten.

Modell	Output	Input	Status	Binary	Timer	Counter	Control	Integer	Float	ASCII	String	BCD	Long	PID	Message	Block Transfer	Function
MicroLogix	X	X	X	X	X	X	X	X	X	X	X	X		X	X		
PLC5	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	
SLC5/05	X	X	X	X	X	X	X	X	X	X	X						

Siehe auch:
Logix-Adressierung
MicroLogix-Adressierung

Adressierung der PLC-5-Reihe
Modulare E/A-Adressierung für SLC 500

Protokollklasse	Modelle	Hilfe-Link
Logix-Ethernet	ControlLogix 5500 Ethernet, CompactLogix 5300 Ethernet, FlexLogix 5400 Ethernet, SoftLogix 5800	Logix-Adressierung
DH+-Gateway	DH+-Gateway: PLC-5 DH+-Gateway: SLC 5/04	Adressierung der PLC-5-Reihe Modulare E/A-Adressierung für SLC 500
ControlNet-Gateway	ControlNet-Gateway: PLC-5C	Adressierung der PLC-5-Reihe
1761-NET-ENI	ENI: ControlLogix 5500 ENI: CompactLogix 5300 ENI: FlexLogix 5400 ENI: MicroLogix ENI: Feste E/A für SLC 500 ENI: Modulare E/A für SLC 500 ENI: PLC-5	Logix-Adressierung MicroLogix-Adressierung Feste E/A-Adressierung für SLC 500 Modulare E/A-Adressierung für SLC 500 Adressierung der PLC-5-Reihe
MicroLogix 1100 Ethernet	MicroLogix 1100	MicroLogix-Adressierung
MicroLogix 1400 Ethernet	MicroLogix 1400	MicroLogix-Adressierung

Weitere Informationen zu den vordefinierte Datentypen des Controllers finden Sie in der Dokumentation zum Gerät.

Logix-Adressierung

Weitere Informationen zur tagbasierten Adressierung und Beziehung dieser Modelle zu ControlLogix-Ethernet-Treiber von Allen-Bradley finden Sie unter Auf Logix-Tags basierte Adressierung.

ControlLogix 5500-Adressierung für Ethernet

ControlLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

ControlLogix 5500-Adressierung für ENI

ControlLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

ControlLogix 5500-Adressierung für Serial Gateway

ControlLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

CompactLogix 5300-Adressierung für Ethernet

CompactLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

CompactLogix 5300-Adressierung für ENI

CompactLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

CompactLogix 5300-Adressierung für Serial Gateway

CompactLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

FlexLogix 5400-Adressierung für Ethernet

FlexLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

FlexLogix 5400-Adressierung für ENI

FlexLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

FlexLogix 5400-Adressierung für Serial Gateway

FlexLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

SoftLogix 5800-Adressierung

SoftLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

SoftLogix 5800-Adressierung für Serial Gateway

SoftLogix ist Mitglied der Logix-Familie und Teil der Integrated Architecture von Rockwell Automation. Dies bedeutet, dass es eine tag- oder symbolbasierte Adressierungsstruktur verwendet. Logix-Tags unterscheiden sich von herkömmlichen PLC-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

MicroLogix-Adressierung

MicroLogix-Adressierung für EtherNet/IP-Gateway

Die tatsächliche Anzahl der zur Verfügung stehenden Adressen ist vom SPS-Modell abhängig. Die Bereiche wurden ausgeweitet, um maximale Flexibilität mit zukünftigen Modellen zu ermöglichen. Wenn der Treiber bei Laufzeit feststellt, dass eine Adresse im Gerät nicht vorhanden ist, sendet er eine Fehlermeldung und entfernt anschließend das Tag aus seiner Scanliste. Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien

ASCII-Dateien
String-Dateien
Long-Dateien
MicroLogix-PID-Dateien
MicroLogix-Meldungsdateien

Wenn Sie Informationen zu Funktionsdateien benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)
RTC-Datei (Real Time Clock, Echtzeituhr)
CS0-Datei (Channel 0 Communication Status, Kommunikationsstatus für Kanal 0)
CS1 (Channel 1 Communication Status, Kommunikationsstatus für Kanal 1)
IOS-Datei (I/O Module Status, E/A-Modulstatus)

MicroLogix-Adressierung für ENI

Die tatsächliche Anzahl der zur Verfügung stehenden Adressen ist vom SPS-Modell abhängig. Die Bereiche wurden ausgeweitet, um maximale Flexibilität mit zukünftigen Modellen zu ermöglichen. Wenn der Treiber bei Laufzeit feststellt, dass eine Adresse im Gerät nicht vorhanden ist, sendet er eine Fehlermeldung und entfernt anschließend das Tag aus seiner Scanliste. Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien
Long-Dateien
MicroLogix-PID-Dateien
MicroLogix-Meldungsdateien

Wenn Sie Informationen zu Funktionsdateien benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)
RTC-Datei (Real Time Clock, Echtzeituhr)
CS0-Datei (Channel 0 Communication Status, Kommunikationsstatus für Kanal 0)
CS1 (Channel 1 Communication Status, Kommunikationsstatus für Kanal 1)
IOS-Datei (I/O Module Status, E/A-Modulstatus)

MicroLogix 1100-Adressierung

Die tatsächliche Anzahl der zur Verfügung stehenden Adressen ist vom SPS-Modell abhängig. Die Bereiche wurden ausgeweitet, um maximale Flexibilität mit zukünftigen Modellen zu ermöglichen. Wenn der Treiber bei Laufzeit feststellt, dass eine Adresse im Gerät nicht vorhanden ist, sendet er eine Fehlermeldung und entfernt anschließend das Tag aus seiner Scanliste. Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien

Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
String-Dateien
Long-Dateien
MicroLogix-PID-Dateien
MicroLogix-Meldungsdateien

Wenn Sie Informationen zu Funktionsdateien benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)
RTC-Datei (Real Time Clock, Echtzeituhr)
CS0-Datei (Channel 0 Communication Status, Kommunikationsstatus für Kanal 0)
CS1 (Channel 1 Communication Status, Kommunikationsstatus für Kanal 1)
IOS-Datei (I/O Module Status, E/A-Modulstatus)

MicroLogix 1400-Adressierung

Die tatsächliche Anzahl der zur Verfügung stehenden Adressen ist vom SPS-Modell abhängig. Die Bereiche wurden ausgeweitet, um maximale Flexibilität mit zukünftigen Modellen zu ermöglichen. Wenn der Treiber bei Laufzeit feststellt, dass eine Adresse im Gerät nicht vorhanden ist, sendet er eine Fehlermeldung und entfernt anschließend das Tag aus seiner Scanliste. Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien
Long-Dateien
MicroLogix-PID-Dateien
MicroLogix-Meldungsdateien

Wenn Sie Informationen zu Funktionsdateien benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)
RTC-Datei (Real Time Clock, Echtzeituhr)
CS0-Datei (Channel 0 Communication Status, Kommunikationsstatus für Kanal 0)
CS1 (Channel 1 Communication Status, Kommunikationsstatus für Kanal 1)
IOS-Datei (I/O Module Status, E/A-Modulstatus)

Feste E/A-Adressierung für SLC 500

Feste E/A-Adressierung für SLC 500 für EtherNet/IP-Gateway

Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien

Feste E/A-Adressierung SLC 500 für ENI

Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien

Modulare E/A-Adressierung für SLC 500

Modulare E/A-Adressierung für SLC 500 für DH+

Die tatsächliche Anzahl der zur Verfügung stehenden Adressen ist vom SPS-Modell abhängig. Die Bereiche wurden ausgeweitet, um maximale Flexibilität mit zukünftigen Modellen zu ermöglichen. Wenn der Treiber bei Laufzeit feststellt, dass eine Adresse im Gerät nicht vorhanden ist, sendet er eine Fehlermeldung und entfernt anschließend das Tag aus seiner Scanliste. Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien

Modulare E/A-Adressierung für SLC 500 für EtherNet/IP-Gateway

Die tatsächliche Anzahl der zur Verfügung stehenden Adressen ist vom SPS-Modell abhängig. Die Bereiche wurden ausgeweitet, um maximale Flexibilität mit zukünftigen Modellen zu ermöglichen. Wenn der Treiber bei Laufzeit feststellt, dass eine Adresse im Gerät nicht vorhanden ist, sendet er eine Fehlermeldung und entfernt anschließend das Tag aus seiner Scanliste. Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien

Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien

Modulare E/A-Adressierung für SLC 500 für ENI

Die tatsächliche Anzahl der zur Verfügung stehenden Adressen ist vom SPS-Modell abhängig. Die Bereiche wurden ausgeweitet, um maximale Flexibilität mit zukünftigen Modellen zu ermöglichen. Wenn der Treiber bei Laufzeit feststellt, dass eine Adresse im Gerät nicht vorhanden ist, sendet er eine Fehlermeldung und entfernt anschließend das Tag aus seiner Scanliste. Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien

Adressierung der PLC-5-Reihe

Adressierung der PLC-5-Reihe für ControlNet

Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien
BCD-Dateien
PID-Dateien
Meldungsdateien
Blocktransferdateien

Adressierung der PLC-5-Reihe für DH+

Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien

Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien
BCD-Dateien
PID-Dateien
Meldungsdateien
Blocktransferdateien

Adressierung der PLC-5-Reihe für EtherNet/IP-Gateway

Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien
BCD-Dateien
PID-Dateien
Meldungsdateien
Blocktransferdateien

Adressierung der PLC-5-Reihe für ENI

Wenn Sie weitere Informationen zur dateispezifischen Adressierung benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien
BCD-Dateien
PID-Dateien
Meldungsdateien
Blocktransferdateien

Auf Logix-Tags basierte Adressierung

Integrated Architecture von Rockwell Automation verwendet eine tag- oder symbolbasierte Adressierungsstruktur, die im Allgemeinen als Logix-Tags (oder native Tags) bezeichnet wird. Diese Tags unterscheiden sich von herkömmlichen SPS-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

Hinweis: Sofern nicht anderweitig angegeben, wird bei Logix-Tags in dieser Hilfedatei davon ausgegangen, dass sie global sind.

Mit ControlLogix-Ethernet-Treiber von Allen-Bradley können Benutzer auf die unteilbaren Datentypen des Controllers zugreifen: BOOL, SINT, INT, DINT, LINT, REAL, LREAL, USINT, UINT, UDINT und ULINT. Zwar handelt es sich bei einigen der vordefinierten Typen um Strukturen, doch basieren sie letztendlich auf diesen unteilbaren Datentypen. Somit sind alle nicht strukturierten (unteilbaren) Mitglieder einer Struktur zugänglich. Beispiel: Ein TIMER (ZEITGEBER) kann keinem Server-Tag zugewiesen werden, wohingegen ein unteilbares TIMER-Mitglied dem Tag zugewiesen werden kann (beispielsweise TIMER.EN, TIMER.ACC usw.). Wenn es sich bei einem Strukturmitglied um eine Struktur selbst handelt, müssten beide Strukturen entsprechend erweitert werden, damit auf ein unteilbares Mitglied der Substruktur zugegriffen werden kann. Dies ist üblicherweise bei benutzer- und moduldefinierten Typen und bei keinem der vordefinierten Typen zu finden.

Unteilbarer Datentyp	Beschreibung		Bereich
BOOL	Einzel-Bit-Wert	VT_ BOOL	0, 1
SINT	8-Bit-Wert mit Vorzeichen	VT_ UI1	-128 bis 127
INT	16-Bit-Wert mit Vorzeichen	VT_I2	-32.768 bis 32.767
DINT	32-Bit-Wert mit Vorzeichen	VT_I4	-2.147.483.648 bis 2.147.483.647
LINT	64-Bit-Wert mit Vorzeichen	VT_I8	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807
USINT	8-Bit-Wert ohne Vorzeichen	VT_ UI1	0 bis 255
UINT	16-Bit-Wert ohne Vorzeichen	VT_ UI2	0 bis 65535
UDINT	32-Bit-Wert ohne Vorzeichen	VT_ UI4	0 bis 4294967295
ULINT	64-Bit-Wert ohne Vorzeichen	VT_I8	0 bis 18,446,744,073,709,551,615
REAL	32-Bit-IEEE-Gleitkommazahl	VT_R4	±1.1754943508222875E-38 bis ±3.4028234663852886E+38 (normalisiert) 0 ±1.4012984643248170E-45 bis ±1.1754942106924411E-38 (denormalisiert)
LREAL	64-Bit-Wert mit Vorzeichen	VT_R8	±2.2250738585072014E-308 bis ±1.7976931348623157E+308 (normalisiert) 0, ±4.9406564584124654E-324 bis ±2.2250738585072010E-308 (denormalisiert)

Siehe auch: Erweiterte Logix-Adressierung

Tag-Adressregeln für Client/Server

Logix-Tag-Namen entsprechen Tag-Adressen für Client/Server. Für Logix-Tag-Namen (Eingabe erfolgt über RSLogix5000) gelten die IEC 1131-3-ID-Regeln. Für Tag-Adressen für Client/Server gelten dieselben Regeln. Die folgenden Symbole werden verwendet:

- Müssen mit einem Buchstaben (A-Z, a-z) oder Unterstrich (_) beginnen.
- Dürfen nur Buchstaben und Unterstriche enthalten.
- Dürfen bis zu 40 Zeichen lang sein.
- Dürfen keine aufeinander folgenden Unterstriche enthalten.
- Dabei wird die Groß-/Kleinschreibung nicht beachtet.

Tag-Namensregeln für Client/Server
Die Zuweisung des Tag-Namens im Server unterscheidet sich von der Adresszuweisung dahingehend, dass Namen nicht mit einem Unterstrich beginnen dürfen.

Hinweis: Zur Leistungsoptimierung sollte für Logix-Tag-Namen eine minimale Länge beibehalten werden. Je kürzer der Name, desto mehr mögliche Anfragen können in eine einzelne Transaktion passen.

Benutzer des symbolischen Modus sollten für Tag-Adressen für Client/Server unter 400 Zeichen beibehalten. Beispiel: tagarray[1,2,4].somestruct.substruct_array[3].basetag.[4] ist 57 Zeichen in der Länge. Da in einem Paket nur 500 Daten-Byte enthalten sein können, können sämtliche Overhead-Byte, die dem Paket hinzugefügt werden müssen, den für die Zeichen selbst zur Verfügung stehenden Platz beträchtlich verringern. Wenn die Adresse unter 400 Zeichen gehalten wird, bleibt die Tag-Anforderung vollständig und gültig.

Siehe auch: *Leistungsoptimierungen*

Adressformate

Es gibt mehrere Möglichkeiten, Logix-Tags statisch im Server oder dynamisch von einem Client aus zu adressieren. Das verwendete Format hängt vom Typ und der Verwendung des Tags ab. Beispiel: Das Bit-Format würde beim Zugriff auf ein Bit innerhalb eines Tags des SINT-Typs verwendet werden. Informationen zum Adressformat und zur Syntax finden Sie in der Tabelle unten.

Hinweis: Alle Formate mit Ausnahme von Array und String sind für RSLogix5000 nativ. Aus diesem Grund kann beim Referenzieren eines unteilbaren Datentyps ein Tag-Name für RSLogix 5000 kopiert und in das Tag-Adressfeld des Servers eingefügt werden und gültig sein.

Format	Syntax	Beispiel	Hinweise
Standard	<Logix-Tag-Name>	tag_1	Tag kann kein Array sein.
Array-Element	<Tag-Name für Logix-Array> [dim1, dim2, dim3]	tag_1 [2, 58, 547] tag_1 [0, 3]	Dimensionsbereich = 1 bis 3 | Elementbereich = 0 bis 65535
Array ohne Offset*	<Tag-Name für Logix-Array> {Spaltenanzahl} <Tag-Name für Logix-Array> {Zeilenanzahl}{Spaltenanzahl}	tag_1 {8} tag_1 {2} {4}	Dimensionsbereich = 1 bis 2 | Elementbereich = 1 bis 65535 Die Anzahl der Elemente zum Lesen/Schreiben ist gleich der Zeilenanzahl multipliziert mit der Spaltenanzahl. Wenn keine Zeilen angegeben werden, liegt die Zeilenanzahl standardmäßig bei 1. Das Array beginnt bei einem Null-Offset (Array-Index gleich 0 für alle Dimensionen).
Array mit Offset*	<Element-Tag für Logix-Array> [Offset1, Offset2] {Spaltenanzahl} <Element-Tag für Logix-Array> [Offset1, Offset2] {Zeilenanzahl}{Spaltenanzahl}	tag_1 [2, 3] {10} tag_1 [2, 3] 2} {5}	Das Array beginnt bei einem durch die Dimensionen im Array-Element-Tag festgelegten Offset. Das Array deckt immer die höchste Dimension ab. Tag_1[2,3]{10} würde ein Array von Elementen tag_1[2,3] -> tag_1[2,13] erzeugen
Bit	<Logix-Tag-Name>.-bit <Logix-Tag-Name>.	tag_1.0 tag_1.[0]	Bit-Bereich = 0 bis 31 Wenn das Tag ein Array ist, muss es ein BOOL-Array

Format	Syntax	Beispiel	Hinweise
	[Bit]		sein, andernfalls kann das Tag kein Array sein.
String	<Logix-Tag-Name>/<Maximale Zeichenfolgenlänge>	tag_1.Data/4 Stringtag_1.Data/82 SINTarraytag_1/16	Längenbereich = 1 bis 65535 Die maximale Anzahl Zeichen, die ein Lesen/Schreiben in der Zeichenfolge ermöglichen.

*Da dieses Format möglicherweise mehr als ein Format anfordert, hängt die Reihenfolge, in der Array-Daten weitergeleitet werden, von der Dimension des Logix-Array-Tags ab. Beispiel: Wenn "Zeilen" multipliziert mit "Spalten" = 4 und das Controller-Tag ein 3X3-Element-Array ist, handelt es sich bei den Elementen, die referenziert werden, um array_tag [0,0], array_tag [0,1], array_tag [0,2] und array_tag [1,0] in dieser genauen Reihenfolge. Die Ergebnisse wären unterschiedlich, wenn es sich beim Controller-Tag um ein 2X10-Element-Array handeln würde.

Informationen zum Referenzieren von Elementen für Arrays mit 1, 2 und 3 Dimensionen finden Sie unter Reihenfolge von Array-Daten.

Tag-Umfang

Globale Tags

Bei globalen Tags handelt es sich um Logix-Tags, die über globalen Umfang im Controller verfügen. Jedes Programm bzw. jede Aufgabe kann auf globale Tags zugreifen. Die Anzahl der Möglichkeiten, wie ein globales Tag referenziert werden kann, ist jedoch von seinem Logix-Datentyp und dem verwendeten Adressformat abhängig.

Programm-Tags

Programm-Tags sind mit globalen Tags identisch, jedoch mit der Ausnahme, dass der Umfang eines Programm-Tags lokal auf das Programm beschränkt ist, in dem es definiert wurde. Programm-Tags richten sich nach denselben Adressierungsregeln und -beschränkungen wie globale Tags, ihnen wird jedoch die folgende Notation vorangestellt:

Programm: <Programmname>.

Beispiel: Das Logix-Tag "tag_1" in Programm "prog_1" würde in einer Client-/Server-Tag-Adresse als "Program:prog_1.tag_1" adressiert werden.

Adressierung von Struktur-Tags

Bei Logix-Struktur-Tags (globale oder Programm-Tags) handelt es sich um Tags mit mindestens einem Mitglied-Tag. Mitglied-Tags können unteilbar oder strukturiert sein.

<Strukturname>. <unteilbares Tag>

Dies impliziert, dass eine Substruktur wie folgt adressiert werden würde:

<Strukturname>. <Substrukturname> .<unteilbares Tag>

Arrays von Strukturen würden wie folgt adressiert werden:

<Strukturname für Array> [dim1, dim2, dim3] . <unteilbares Tag>

Dies ist impliziert, dass ein Array von Substrukturen wie folgt adressiert werden würde:

<Strukturname>. <Substrukturname für Array> [dim1, dim2, dim3] . <unteilbares Tag>

Hinweis: Die oben genannten Beispiele stellen nur einige Adressierungsmöglichkeiten dar, die Strukturen beinhalten, und wurden als Einführung in die Strukturadressierung exemplarisch genannt. Weitere Informationen dazu finden Sie in der Allen-Bradley- oder Rockwell-Dokumentation.

Interne Tags

Interne Tags sind in der Serverkonfiguration nicht sichtbar, können jedoch vom OPC-Client durchsucht und unter <*Kanalname*> gefunden werden.Gruppe <*Gerätename*>. Das Tag "_CIPConnectionSizeRequested" stellt die angeforderte CIP-Verbindungsgröße dar. Das Tag "_CIPConnectionSizeActual" stellt die tatsächlich verwendete CIP-Verbindungsgröße dar. Der zugehörige Wert unterscheidet sich vom Tag "_CIPConnectionSizeRequested", wenn der angeforderte Wert nicht durch das Gerät unterstützt wird. Weitere Informationen zur Verbindungsgröße finden Sie unter Logix-Kommunikationsparameter.

Hinweis: Die in der Tabelle unten beschriebenen Tags sind nur für die ControlLogix 5500- und CompactLogix 5300-Gerätemodelle gültig.

Typ	Tag-Name	Unterstützung	Zugriff
System-Tag	_CIPConnectionSizeRequested	Logix-Modelle mit Ausnahme von Serial Gateway-Modellen.	Lesen/Schreiben*
Status-Tag	_CIPConnectionSizeActual	Logix-Modelle mit Ausnahme von Serial Gateway-Modellen.	Schreibgeschützt

*Dieses Tag ist für ENI-Logix-Modelle schreibgeschützt.

CIP-Verbindungsgröße ändern

Das Tag "_CIPConnectionSizeRequested" ermöglicht es Benutzern, die Eigenschaft der CIP-Verbindungsgröße in Echtzeit zu ändern. Die Verbindungsgrößeneigenschaft (unter **Logix-Kommunikationsparameter** in **Geräteeigenschaften**) und das System-Tag können konfiguriert werden, während die Clients verbunden sind. Änderungen werden angewendet, bevor die nächste Lese-/Schreibanforderung durchgeführt wird.

Vordefinierte Ausdrucks-Tags

Mithilfe der in der Tabelle unten angezeigten Tags können allgemeine Prozessorinformationen von einer SPS, auf der die Firmware-Version 13 oder höher ausgeführt wird, abgerufen werden.

Tag-Name	Description
#MODE	Eine Beschreibung des aktuellen Schlüsselwechselmodus der SPS. Mögliche Zeichenfolgenwerte sind: Programm, Ausführen, Remote-Programm, Remote-Ausführung und Remote-Debugging. Unterstützte Datentypen schließen String ein.
#PLCTYPE	Ein Ganzzahlwert, der dem in der EDS-Datei der SPS angegebenen Attribut "ProdType" entspricht. Zu den unterstützten Datentypen gehören alle außer String.
#REVISION	Firmware-Revision als "<Hauptrevision>.<Nebenrevision>" angezeigt. Unterstützte Datentypen schließen String ein.
#PROCESSORNAME	Der Prozessorname, der dem in der EDS-Datei der SPS angegebenen Attribut "ProdName" entspricht. Unterstützte Datentypen schließen String ein.
#STATUS	Gibt den Status der SPS an. Mögliche Werte sind OK (1) und Fehlerhaft (0). Zu den unterstützten Datentypen gehören alle außer Date.
#PRODUCTCODE	Ein Ganzzahlwert, der dem in der EDS-Datei der SPS angegebenen Attribut "ProdCode" entspricht. Zu den unterstützten Datentypen gehören alle außer String.
#VENDORID	Ein Ganzzahlwert, der dem in der EDS-Datei der SPS angegebenen Attribut "VendCode" entspricht. Zu den unterstützten Datentypen gehören alle außer String.

Adressieren unteilbarer Datentypen

Im Folgenden finden Sie empfohlene Verwendungs- und Adressierungsmöglichkeiten für einen Logix-Datentyp unter Angabe der verfügbaren Adressformate. Beispiele werden auch zur Hervorhebung angegeben. Klicken Sie auf **Erweitert**, wenn Sie erweiterte Adressierungsmöglichkeiten für den angegebenen unteilbaren Datentyp benötigen.

Hinweis: Leere Zellen weisen nicht zwangsläufig auf mangelnde Unterstützung hin.

Unteilbarer Datentyp	Standard	Array-Element	Array mit oder ohne Offset	Bit	String
BOOL					
Datentyp für Client/Server Erweitert	Boolean	Boolean (BOOL - 1-dimensionales Array)	Boolean-Array (BOOL - 1-dimensionales Array)		
Tag-Beispiel für Client/Server	BOOLTAG	BOOLARR[0]	BOOLARR[0]{32}		
SINT					
Datentyp für Client/Server Erweitert	Byte, Char	Byte, Char	Byte-Array, Char-Array (SINT - 1-, 2-, 3-dimensionales Array)	Boolean (Bit in SINT)	String (SINT - 1-, 2-, 3-dimensionales Array)
Tag-Beispiel für Client/Server	SINTTAG	SINTARR[0]	SINTARR[0]{4}	SINTTAG.0	SINTARR/4
INT					
Datentyp für Client/Server Erweitert	Word, Short	Word, Short	Wort-Array, Short-Array (INT - 1-, 2-, 3-dimensionales Array)	Boolean (Bit in INT)	*Siehe Erweiterte Adressierung: INT*
Tag-Beispiel für Client/Server	INTTAG	INTARR[0]	INTARR[0]{4}	INTTAG.0	
DINT					
Datentyp für Client/Server Erweitert	DWord, Long	DWord, Long	DWord-Array, Long-Array	Boolean (Bit in DINT)	*Siehe Erweiterte Adressierung: DINT*
Tag-Beispiel für Client/Server	DINTTAG	DINTARR[0]	DINTARR[0]{4}	DINTTAG.0	
LINT					
Datentyp für Client/Server Erweitert	Double, Date	Double, Date	Double-Array		
Tag-Beispiel für Client/Server	LINTTAG	LINTARR[0]	LINTARR[0]{4}		
REAL					
Datentyp für Client/Server Erweitert	Float	Float	Float-Array		*Siehe Erweiterte Adressierung: REAL*
Tag-Beispiel für Client/Server	REALTAG	REALARR[0]	REALARR[0]{4}		
USINT					
Datentyp für Client/Server Erweitert	Byte	Byte	Byte-Array	Boolean (Bit in USINT)	*Siehe Erweiterte Adressierung: USINT*
Datentyp für Client/Server	USINTTAG	USINTTARR[0]	USINTTARR[0]{4}	USINTTAG.0	
UINT					
Datentyp für Client/Server	Word, BCD	Word, BCD	Word Array, BCD Array	Boolean (Bit in UINT)	*Siehe Erweiterte Adres-*

Unteilbarer Datentyp	Standard	Array-Element	Array mit oder ohne Offset	Bit	String
Erweitert					*sierung: UINT*
Tag-Beispiel für Client/Server	UINTTAG	UINTARR[0]	UINTARR[0]{4}	UINTTAG.0	
UDINT					
Datentyp für Client/Server Erweitert	DWord, LBCD	DWord, LBCD	DWord-Array, LBCD-Array	Boolean	*Siehe Erweiterte Adressierung: UDINT*
Tag-Beispiel für Client/Server	UDINTTAG	UDINTARR[0]	UDINTARR[0]{4}	UDINTAG.0	
ULINT					
Datentyp für Client/Server Erweitert	Double	Double	Double-Array		*Siehe Erweiterte Adressierung: ULINT*
Tag-Beispiel für Client/Server	ULINTTAG	ULINTARR[0]	ULINTARR[0]{4}		
LREAL					
Datentyp für Client/Server Erweitert	Double	Double	Double-Array		*Siehe Erweiterte Adressierung: LREAL*
Tag-Beispiel für Client/Server	LREALTAG	LREALARR[0]	LREALARR[0]{4}		

Strukturdatentypen adressieren

Nur die unteilbaren Strukturmitglieder können auf Strukturebene adressiert werden. Weitere Informationen dazu finden Sie in den Beispielen unten.

Logix-Tag
MyTimer @ TIMER

Client/Server-Tag
1. Ungültig

TimerTag-Adresse = MyTimer
TimerTag-Datentyp = ??

2. Gültig
TimerTag-Adresse = MyTimer.ACC
TimerTag-Datentyp = DWord

STRING-Datentyp adressieren

STRING ist ein vordefinierter Logix-Datentyp, dessen Struktur zwei Mitglieder enthält: DATA und LEN. DATA ist ein Array von SINTs und speichert die Zeichen der Zeichenfolge. LEN ist ein DINT und steht für die Anzahl von Zeichen in DATA, die einem Client angezeigt werden.

Da LEN und DATA unteilbare Mitglieder sind, müssen sie unabhängig vom Client/Server referenziert werden. Die Syntax ist wie unten angezeigt.

Beschreibung	Syntax	Beispiel
STRING-Wert	DATA/<Maximale STRING-Länge>	MYSTRING.DATA/82
Tatsächliche STRING-Länge	LEN	MYSTRING.LEN

Lesevorgänge

Der STRING-Lesevorgang von DATA wird durch Folgendes beendet:

a. Das erste Nullabschlusszeichen wurde festgestellt.

b. Der Wert in LEN, wenn a) nicht zuerst auftritt.

c. Die <Maximale STRING-Länge>, wenn entweder a) oder b) nicht zuerst auftritt.

Beispiel
MYSTRING.DATA enthält "Hello World" in der SPS, aber LEN wird manuell auf 5 eingestellt. Beim Lesen von MYSTRING.DATA/82 wird "Hello" angezeigt. Wenn LEN auf 20 eingestellt ist, zeigt MYSTRING.DATA/82 "Hello World" an.

Schreibvorgänge

Wenn ein STRING-Wert in DATA geschrieben wird, schreibt der Treiber auch mit der geschriebenen DATA-Länge in LEN. Wenn das Schreiben in LEN aus irgendeinem Grund fehlschlägt, wird der Schreibvorgang in DATA ebenso als fehlgeschlagen erachtet (ungeachtet der Tatsache, dass der DATA-Schreibvorgang in den Controller erfolgreich war).

Hinweis: Dieses Verhalten wurde speziell für Logix-Tags des Typs STRING oder eine benutzerdefinierte Ableitung davon konzipiert. Die folgenden Vorsichtsmaßnahmen gelten für Benutzer, die ihren eigenen STRING in UDTs implementieren möchten.

- Wenn ein UDT vorhanden ist, das über ein als STRING referenziertes DATA-Mitglied und ein als DINT referenziertes LEN-Mitglied verfügt, ist das Schreiben in LEN erfolgreich, und zwar unabhängig von den Absichten für LEN für das angegebene UDT. Beim Entwickeln von UDTs muss sorgfältig vorgegangen werden, um diese Möglichkeit zu vermeiden, wenn LEN nicht die DATA-Länge sein soll.
- Wenn ein UDT vorhanden ist, das über ein als STRING referenziertes DATA-Mitglied, jedoch über kein LEN-Mitglied verfügt, schlägt der Schreibvorgang in LEN im unbeaufsichtigten Modus ohne Folgen für DATA fehl.

Beispiel
MYSTRING.DATA/82 behält den Wert "Hello World" bei. MYSTRING.LEN behält 11 bei. Wenn der Wert "Alarm ausgelöst" in MYSTRING.DATA/82 geschrieben wird, wird 15 in MYSTRING.LEN geschrieben. Wenn der Schreibvorgang in MYSTRING.LEN fehlschlägt, behält MYSTRING.LEN seinen vorherigen Wert 11, während MYSTRING.DATA/82 die ersten 11 Zeichen ("Alarm ausgelöst") anzeigt. Wenn der Schreibvorgang in MYSTRING.DATA/82 fehlschlägt, ist kein Tag davon betroffen.

Zeichenfolgendaten bei LEN beenden

In den logischen Adressierungsmodi hat das Lesen von STRING.DATA das automatische Lesen von STRING.LEN im symbolischen Modus zur Folge. Dies kann durch Deaktivieren der Option Zeichenfolgendaten bei LEN beenden umgangen werden. *Weitere Informationen dazu finden Sie unter Logix-Optionen.*

Reihenfolge von Logix-Array-Daten

Eindimensionale Arrays - Array [dim1]
Eindimensionale Array-Daten werden dem bzw. vom Controller in aufsteigender Reihenfolge weitergeleitet.
for (dim1 = 0; dim1 < dim1_max; dim1++)

Beispiel: Array mit 3 Elementen
array [0]
array [1]
array [2]

Zweidimensionale Arrays - Array [dim1, dim2]
Zweidimensionale Array-Daten werden dem bzw. vom Controller in aufsteigender Reihenfolge weitergeleitet.
for (dim1 = 0; dim1 < dim1_max; dim1++)
for (dim2 = 0; dim2 < dim2_max; dim2++)

Beispiel: Array mit 3x3 Elementen
array [0, 0]
array [0, 1]
array [0, 2]

array [1, 0]
array [1, 1]
array [1, 2]
array [2, 0]
array [2, 1]
array [2, 2]

dreidimensionale Arrays - Array [dim1, dim2, dim3]
Dreidimensionale Array-Daten werden dem bzw. vom Controller in aufsteigender Reihenfolge weitergeleitet.
for (dim1 = 0; dim1 < dim1_max; dim1++)
for (dim2 = 0; dim2 < dim2_max; dim2++)
for (dim3 = 0; dim3 < dim3_max; dim3++)

Beispiel: Array mit 3x3x3 Elementen
array [0, 0, 0]
array [0, 0, 1]
array [0, 0, 2]
array [0, 1, 0]
array [0, 1, 1]
array [0, 1, 2]
array [0, 2, 0]
array [0, 2, 1]
array [0, 2, 2]
array [1, 0, 0]
array [1, 0, 1]
array [1, 0, 2]
array [1, 1, 0]
array [1, 1, 1]
array [1, 1, 2]
array [1, 2, 0]
array [1, 2, 1]
array [1, 2, 2]
array [2, 0, 0]
array [2, 0, 1]
array [2, 0, 2]
array [2, 1, 0]
array [2, 1, 1]
array [2, 1, 2]
array [2, 2, 0]
array [2, 2, 1]
array [2, 2, 2]

Erweiterte Logix-Adressierung

Die erweiterte Adressierung steht für die folgenden unteilbaren Datentypen zur Verfügung. Wählen Sie eine Verknüpfung aus der nachfolgenden Liste aus, um weitere Informationen zu einem bestimmten Datentyp zu erhalten.

BOOL
SINT
INT
DINT
LINT
REAL
USINT
UINT
UDINT
ULINT
LREAL

Erweiterte Adressierung: BOOL

Format	Unterstützte Datentypen	Hinweise
Standard	Boolean Byte, Char Word, Short, BCD DWord, Long, LBCD Float*	Keine
	Boolean	Das Controller-Tag muss ein eindimensionales Array sein.
Array ohne Offset	Boolean-Array	1. Das Controller-Tag muss ein eindimensionales Array sein. 2. Die Anzahl der Elemente muss einem Faktor von 8 entsprechen.
Array ohne Offset	Byte-Array, Char-Array Word-Array, Short-Array, BCD-Array DWord-Array, Long-Array, LBCD-Array Float-Array*	Wird nicht unterstützt.
Array mit Offset	Boolean-Array	1. Das Controller-Tag muss ein eindimensionales Array sein. 2. Das Offset muss auf einer 32-Bit-Grenze liegen. 3. Die Anzahl der Elemente muss einem Faktor von 8 entsprechen.
Bit	Boolean	1. Das Controller-Tag muss ein eindimensionales Array sein. 2. Der Bereich ist von 0 bis 31 begrenzt.
String	Zeichenfolge	Wird nicht unterstützt.

*Der Float-Wert entspricht dem Nennwert des Controller-Tags in Gleitkommazahlform (Nicht-IEEE-Gleitkommazahl).

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

BOOL-Controller-Tag - booltag = wahr

Server-Tag-Adresse	Format	Datentyp	Hinweise
booltag	Standard	Boolean	Wert = wahr
booltag	Standard	Byte	Wert = 1
booltag	Standard	Word	Wert = 1
booltag	Standard	DWord	Wert = 1
booltag	Standard	Float	Wert = 1.0
booltag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array.
booltag [3]	Array-Element	Word	Ungültig: Tag ist kein Array.
booltag {1}	Array ohne Offset	Word	Ungültig: Wird nicht unterstützt.
booltag {1}	Array ohne Offset	Boolean	Ungültig: Wird nicht unterstützt.
booltag [3] {32}	Array mit Offset	Boolean	Ungültig: Tag ist kein Array.
booltag . 3	Bit	Boolean	Ungültig: Tag ist kein Array.
booltag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Wird nicht unterstützt.
booltag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Wird nicht unterstützt.

BOOL-Array-Controller-Tag - bitarraytag = [0,1,0,1]

Server-Tag-Adresse	Format	Datentyp	Hinweise
bitarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
bitarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
bitarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
bitarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.
bitarraytag	Standard	Float	Ungültig: Tag kann kein Array sein.
bitarraytag [3]	Array-Element	Boolean	Wert = wahr
bitarraytag [3]	Array-Element	Word	Ungültig: Fehlerhafter Datentyp.
bitarraytag {3}	Array ohne Offset	Word	Ungültig: Tag kann kein Array sein.
bitarraytag {1}	Array ohne Offset	Word	Ungültig: Tag kann kein Array sein.
bitarraytag {1}	Array ohne Offset	Boolean	Ungültig: Array-Größe muss einem Faktor von 8 entsprechen.
bitarraytag {32}	Array ohne Offset	Boolean	Wert = [0,1,0,1,...]
bitarraytag [3] {32}	Array mit Offset	Boolean	Offset muss an 32-Bit-Grenze beginnen.
bitarraytag[0]{32}	Array mit Offset	Boolean	Wert = [0,1,0,1,...]
bitarraytag[32]{64}	Array mit Offset	Boolean	Wert = [...] *Werte oben nicht angegeben*
bitarraytag . 3	Bit	Boolean	Wert = wahr
bitarraytag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Wird nicht unterstützt.
bitarraytag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Wird nicht unterstützt.

Erweiterte Adressierung: SINT

Format	Unterstützte Datentypen	Hinweise
Standard	Boolean*, Byte, Char, Word, Short, BCD, DWord, Long, LBCD, Float***	Keine
Array-Element	Byte, Char, Word, Short, BCD, DWord, Long, LBCD, Float***	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Boolean-Array	1. Verwenden Sie diesen Fall, damit die Bit innerhalb von SINT in Array-Form vorliegen. **Hinweis:** Dies ist kein Array von SINTs in boolescher Schreibweise. 2. Gilt nur für Bit innerhalb SINT. Beispiel: tag_1.0{8}. 3. Die Summe aus .bit und Array-Größe darf 8 Bit nicht überschreiten. Beispiel: tag_1.1{8} überschreitet SINT, tag_1.0{8} nicht. 4. Array-Größe muss ein Vielfaches von acht (8) sein.
Array ohne Offset	Byte-Array, Char-Array, Word-Array, Short-Array, BCD-Array**, DWord-Array, Long-Array, LBCD-Array**, Float-Array**,***	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.

Format	Unterstützte Datentypen	Hinweise
Array mit Offset	Byte-Array, Char-Array, Word-Array, Short-Array, BCD-Array**, DWord-Array, Long-Array, LBCD-Array**, Float-Array**,***	Das Controller-Tag muss ein Array sein.
Bit	Boolean	1. Der Bereich ist von 0 bis 7 begrenzt. 2. Wenn das Controller-Tag ein Array ist, muss der Bit-Klassenreferenz eine Array-Element-Klassenreferenz vorangestellt werden. Beispiel: tag_1 [2,2,3].0.
String	Zeichenfolge	1. Wenn auf ein einzelnes Element zugegriffen wird, muss das Controller-Tag kein Array sein. **Hinweis:** Der Wert der Zeichenfolge ist das ASCII-Äquivalent des SINT-Werts. Beispiel: SINT = 65 (dezimal) = "A". 2. Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein. Der Wert der Zeichenfolge ist das mit Null beendete ASCII-Äquivalent aller SINTs in der Zeichenfolge. 1 Zeichen in Zeichenfolge = 1 SINT.

*Werte ungleich Null sind auf "wahr" fixiert.
**Jedes Element des Arrays entspricht einem Element im SINT-Array. Arrays sind nicht gepackt.
*** Der Float-Wert entspricht dem Nennwert des Controller-Tags in Gleitkommazahlform (Nicht-IEEE-Gleitkommazahl).

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

SINT-Controller-Tag - sinttag = 122 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
sinttag	Standard	Boolean	Wert = wahr
sinttag	Standard	Byte	Wert = 122
sinttag	Standard	Word	Wert = 122
sinttag	Standard	DWord	Wert = 122
sinttag	Standard	Float	Wert = 122.0
sinttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist auch ungültig.
sinttag [3]	Array-Element	Byte	Ungültig: Tag ist kein Array.
sinttag {3}	Array ohne Offset	Byte	Ungültig: Tag ist kein Array.
sinttag {1}	Array ohne Offset	Byte	Wert = [122]
sinttag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
sinttag [3] {1}	Array mit Offset	Byte	Ungültig: Tag ist kein Array.
sinttag . 3	Bit	Boolean	Wert = wahr
sinttag . 0 {8}	Array ohne Offset	Boolean	Wert = [0,1,0,1,1,1,1,0] Bit-Wert von 122
sinttag / 1	Zeichenfolge	Zeichenfolge	Wert = "z"
sinttag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Tag ist kein Array.

SINT-Array-Controller-Tag - sintarraytag [4,4] = [[83,73,78,84],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

Server-Tag-Adresse	Format	Datentyp	Hinweise
sintarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
sintarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
sintarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
sintarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.
sintarraytag	Standard	Float	Ungültig: Tag kann kein Array sein.
sintarraytag [3]	Array-Element	Byte	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
sintarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
sintarraytag [1,3]	Array-Element	Byte	Wert = 8
sintarraytag {10}	Array ohne Offset	Byte	Wert = [83,73,78,84,5,6,7,8,9,10]
sintarraytag {2} {5}	Array ohne Offset	Word	Wert = [83,73,78,84,5] [6,7,8,9,10]
sintarraytag {1}	Array ohne Offset	Byte	Wert = 83
sintarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
sintarraytag [1,3] {4}	Array mit Offset	Byte	Wert = [8,9,10,11]
sintarraytag . 3	Bit	Boolean	Ungültig: Tag muss unteilbare Position referenzieren.
sintarraytag [1,3] . 3	Bit	Boolean	Wert = 1
sintarraytag [1,3] . 0 {8}	Array ohne Offset	Boolean	Wert = [0,0,0,1,0,0,0,0]
sintarraytag / 1	Zeichenfolge	Zeichenfolge	Wert = "S"
sintarraytag / 4	Zeichenfolge	Zeichenfolge	Wert = "SINT"

Erweiterte Adressierung: INT

Format	Unterstützte Datentypen	Hinweise
Standard	Boolean*, Byte, Char**, Word, Short, BCD, DWord, Long, LBCD, Float****	Keine
Array-Element	Byte, Char**, Word, Short, BCD, DWord, Long, LBCD, Float****	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Boolean-Array	1. Verwenden Sie diesen Fall, damit die Bit innerhalb von INT in Array-Form vorliegen. **Hinweis:** Dies ist kein Array von INTs in boolescher Schreibweise. 2. Gilt nur für Bit innerhalb INT. Beispiel: tag_1.0{16}. 3. Die Summe aus .bit und Array-Größe darf 16 Bit nicht überschreiten. Beispiel: tag_1.1{16} überschreitet INT, tag_1.0{16} nicht. 4. Array-Größe muss ein Vielfaches von acht (8) sein.
Array	Byte-Array, Char-Array**, Word-	Wenn auf mehr als ein einzelnes Element zugegriffen wird,

Format	Unterstützte Datentypen	Hinweise
ohne Offset	Array, Short-Array, BCD-Array, DWord-Array, Long-Array, LBCD-Array***, Float-Array***,****	muss das Controller-Tag ein Array sein.
Array mit Offset	Byte-Array, Char-Array**, Word-Array, Short-Array, BCD-Array, DWord-Array, Long-Array, LBCD-Array***, Float-Array***,****	Das Controller-Tag muss ein Array sein.
Bit	Boolean	1. Der Bereich ist von 0 bis 15 begrenzt. 2. Wenn das Controller-Tag ein Array ist, muss der Bit-Klassenreferenz eine Array-Element-Klassenreferenz vorangestellt werden. Beispiel: tag_1 [2,2,3].0.
String	Zeichenfolge	1. Wenn auf ein einzelnes Element zugegriffen wird, muss das Controller-Tag kein Array sein. Hinweis: Der Wert der Zeichenfolge ist das ASCII-Äquivalent des INT-Werts (auf 255 fixiert). Beispiel: INT = 65 (dezimal) = "A". 2. Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein. Der Wert der Zeichenfolge ist das mit Null beendete ASCII-Äquivalent aller INTs (auf 255 fixiert) in der Zeichenfolge. 1 Zeichen in Zeichenfolge = 1 INT, auf 255 fixiert INT-Zeichenfolgen sind nicht gepackt. Verwenden Sie für eine größere Effizienz stattdessen SINT-Zeichenfolgen oder die STRING-Struktur.

*Werte ungleich Null sind auf "wahr" fixiert.
**Werte, die 255 überschreiten, sind auf 255 fixiert.
***Jedes Element des Arrays entspricht einem Element im INT-Array. Arrays sind nicht gepackt.
****Der Float-Wert entspricht dem Nennwert des Controller-Tags in Gleitkommazahlform (Nicht-IEEE-Gleitkommazahl).

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

INT-Controller-Tag - inttag = 65534 (dezimal)

Server-Tag-Adresse	Klasse	Datentyp	Hinweise
inttag	Standard	Boolean	Wert = wahr
inttag	Standard	Byte	Wert = 255
inttag	Standard	Word	Wert = 65534
inttag	Standard	DWord	Wert = 65534
inttag	Standard	Float	Wert = 65534.0
inttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist ungültig.
inttag [3]	Array-Element	Word	Ungültig: Tag ist kein Array.
inttag {3}	Array ohne Offset	Word	Ungültig: Tag ist kein Array.
inttag {1}	Array ohne Offset	Word	Wert = [65534]
inttag {1}	Array ohne Off-	Boolean	Ungültig: Fehlerhafter Datentyp.

Server-Tag-Adresse	Klasse	Datentyp	Hinweise
	set		
inttag [3] {1}	Array mit Offset	Word	Ungültig: Tag ist kein Array.
inttag . 3	Bit	Boolean	Wert = wahr
inttag . 0 {16}	Array ohne Offset	Boolean	Wert = [0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] Bit-Wert von 65534
inttag / 1	Zeichenfolge	Zeichenfolge	Wert = nicht druckbares Zeichen = 255 dezimal.
inttag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Tag ist kein Array.

INT-Array-Controller-Tag - intarraytag [4,4] = [[73,78,84,255],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

Server-Tag-Adresse	Klasse	Datentyp	Hinweise
intarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
intarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
intarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
intarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.
intarraytag	Standard	Float	Ungültig: Tag kann kein Array sein.
intarraytag [3]	Array-Element	Word	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
intarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
intarraytag [1,3]	Array-Element	Word	Wert = 259
intarraytag {10}	Array ohne Offset	Byte	Wert = [73,78,84,255,255,255,255,255,9,10]
intarraytag {2} {5}	Array ohne Offset	Word	Wert = [73,78,84,255,256] [257,258,259,9,10]
intarraytag {1}	Array ohne Offset	Word	Wert = 73
intarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
intarraytag [1,3] {4}	Array mit Offset	Word	Wert = [259,9,10,11]
intarraytag . 3	Bit	Boolean	Ungültig: Tag muss unteilbare Position referenzieren.
intarraytag [1,3] . 3	Bit	Boolean	Wert = 0
intarraytag [1,3] . 0 {16}	Array ohne Offset	Boolean	Wert = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] Bit-Wert für 259
intarraytag / 1	Zeichenfolge	Zeichenfolge	Wert = "I"
intarraytag / 3	Zeichenfolge	Zeichenfolge	Wert = "INT"

Erweiterte Adressierung: DINT

Format	Unterstützte Datentypen	Hinweise
Standard	Boolean*, Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float ****	Keine
Array-Element	Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float ****	Das Controller-Tag muss ein Array sein.

Format	Unterstützte Datentypen	Hinweise
Array ohne Offset	Boolean-Array	1. Verwenden Sie diesen Fall, damit die Bit innerhalb von DINT in Array-Form vorliegen. **Hinweis:** Dies ist kein Array von DINTs in boolescher Schreibweise. 2. Gilt nur für Bit innerhalb DINT. Beispiel: tag_1.0{32}. 3. Die Summe aus .bit und Array-Größe darf 32 Bit nicht überschreiten. Beispiel: tag_1.1{32} überschreitet DINT, tag_1.0{32} nicht. 4. Array-Größe muss ein Vielfaches von acht (8) sein.
Array ohne Offset	Byte-Array, Char-Array**, Word-Array, Short-Array, BCD-Array***, DWord-Array, Long-Array, LBCD-Array, Float-Array ****	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.
Array mit Offset	Byte-Array, Char-Array**, Word-Array, Short-Array, BCD-Array***, DWord-Array, Long-Array, LBCD-Array, Float-Array ****	Das Controller-Tag muss ein Array sein.
Bit	Boolean	1. Der Bereich ist von 0 bis 31 begrenzt. 2. Wenn das Controller-Tag ein Array ist, muss der Bit-Klassenreferenz eine Array-Element-Klassenreferenz vorangestellt werden. Beispiel: tag_1 [2,2,3].0.
String	Zeichenfolge	1. Wenn auf ein einzelnes Element zugegriffen wird, muss das Controller-Tag kein Array sein. **Hinweis:** Der Wert der Zeichenfolge ist das ASCII-Äquivalent des DINT-Werts (auf 255 fixiert). Beispiel: SINT = 65 (dezimal) = "A". 2. Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein. Der Wert der Zeichenfolge ist das mit Null beendete ASCII-Äquivalent aller DINTs (auf 255 fixiert) in der Zeichenfolge. 1 Zeichen in Zeichenfolge = 1 DINT, auf 255 fixiert **Hinweis:** DINT-Zeichenfolgen sind nicht gepackt. Verwenden Sie für eine größere Effizienz stattdessen SINT-Zeichenfolgen oder die STRING-Struktur.

*Werte ungleich Null sind auf "wahr" fixiert.
**Werte, die 255 überschreiten, sind auf 255 fixiert.
***Werte, die 65535 überschreiten, sind auf 65535 fixiert.
****Der Float-Wert entspricht dem Nennwert des Controller-Tags in Gleitkommazahlform (Nicht-IEEE-Gleitkommazahl).

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

DINT-Controller-Tag - dinttag = 70000 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
dinttag	Standard	Boolean	Wert = wahr
dinttag	Standard	Byte	Wert = 255
dinttag	Standard	Word	Wert = 65535
dinttag	Standard	DWord	Wert = 70000

Server-Tag-Adresse	Format	Datentyp	Hinweise
dinttag	Standard	Float	Wert = 70000.0
dinttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist ungültig.
dinttag [3]	Array-Element	DWord	Ungültig: Tag ist kein Array.
dinttag {3}	Array ohne Offset	DWord	Ungültig: Tag ist kein Array.
dinttag {1}	Array ohne Offset	DWord	Wert = [70000]
dinttag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
dintag [3] {1}	Array mit Offset	DWord	Ungültig: Tag ist kein Array.
dinttag . 3	Bit	Boolean	Wert = falsch
dinttag . 0 {32}	Array ohne Offset	Boolean	Wert = [0,0,0,0,1,1,1,0,1,0,0,0,1,0,0,0,1,0,...0] Bit-Wert für 70000
dinttag / 1	Zeichenfolge	Zeichenfolge	Wert = nicht druckbares Zeichen = 255 dezimal
dinttag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Tag ist kein Array.

DINT-Array-Controller-Tag - dintarraytag [4,4] = [[68,73,78,84],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

Server-Tag-Adresse	Format	Datentyp	Hinweise
dintarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
dintarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
dintarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
dintarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.
dintarraytag	Standard	Float	Ungültig: Tag kann kein Array sein.
dintarraytag [3]	Array-Element	DWord	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
dintarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
dintarraytag [1,3]	Array-Element	DWord	Wert = 259
dintarraytag {10}	Array ohne Offset	Byte	Wert = [68,73,78,84,255,255,255,255,9,10]
dintarraytag {2}{5}	Array ohne Offset	DWord	Wert = [68,73,78,84,256] [257,258,259,9,10]
dintarraytag {1}	Array ohne Offset	DWord	Wert = 68
dintarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
dintarraytag [1,3] {4}	Array mit Offset	DWord	Wert = [259,9,10,11]
dintarraytag . 3	Bit	Boolean	Ungültig: Tag muss unteilbare Position referenzieren.
dintarraytag [1,3] . 3	Bit	Boolean	Wert = 0
dintarraytag [1,3] .0 {32}	Array ohne Offset	Boolean	Wert = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] Bit-Wert für 259
dintarraytag / 1	Zeichenfolge	Zeichenfolge	Wert = "D"
dintarraytag / 3	Zeichenfolge	Zeichenfolge	Wert = "DINT"

Erweiterte Adressierung: LINT

Format	Unterstützte Datentypen	Hinweise
Standard	Double*, Date**	Keine
Array-Element	Double*, Date**	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Double, Array*	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.
Array mit Offset	Double, Array*	Das Controller-Tag muss ein Array sein.
Bit	k.A.	Wird nicht unterstützt.
String	k.A.	Wird nicht unterstützt.

*Der Double-Wert entspricht dem Nennwert des Controller-Tags in Gleitkommazahlform (Nicht-IEEE-Gleitkommazahl).
**Datumswerte liegen in UTC-Zeit und nicht in lokalisierter Zeit vor.

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

LINT-Controller-Tag - linttag = 2007-01-01T16:46:40.000 (Datum) == 1.16767E+15 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
linttag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
linttag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.
linttag	Standard	Word	Ungültig: Word wird nicht unterstützt.
linttag	Standard	Double	Wert = 1.16767E+15
linttag	Standard	Date	Wert = 2007-01-01T16:46:40.000*
linttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist ungültig.
linttag [3]	Array-Element	Double	Ungültig: Tag ist kein Array.
linttag {3}	Array ohne Offset	Double	Ungültig: Tag ist kein Array.
linttag {1}	Array ohne Offset	Double	Wert = [1.16767E+15]
linttag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
lintag [3] {1}	Array mit Offset	Double	Ungültig: Tag ist kein Array.
linttag . 3	Bit	Boolean	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
linttag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

*Datumswerte liegen in UTC-Zeit und nicht in lokalisierter Zeit vor.

LINT-Array-Controller-Tag -
lintarraytag [2,2] = [0, 1.16767E+15],[9.4666E+14, 9.46746E+14] Dabei gilt:
1.16767E+15 == 2007-01-01T16:46:40.000 (Datum)
9.4666E+14 == 1999-12-31T17:06:40.000
9.46746E+14 == 2000-01-1T17:00:00.000
0 == 1970-01-01T00:00:00.000

Server-Tag-Adresse	Format	Datentyp	Hinweise
lintarraytag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
lintarraytag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.

Server-Tag-Adresse	Format	Datentyp	Hinweise
lintarraytag	Standard	Word	Ungültig: Word wird nicht unterstützt.
lintarraytag	Standard	Double	Ungültig: Tag kann kein Array sein.
lintarraytag	Standard	Date	Ungültig: Tag kann kein Array sein.
lintarraytag [1]	Array-Element	Double	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
lintarraytag [1,1]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
lintarraytag [1,1]	Array-Element	Double	Wert = 9.46746E+14
lintarraytag [1,1]	Array-Element	Date	Wert = 2000-01-01T17:00:00.000*
lintarraytag {4}	Array ohne Offset	Double	Wert = [0, 1.16767E+15, 9.4666E+14, 9.46746E+14]
lintarraytag {2} {2}	Array ohne Offset	Double	Wert = [0, 1.16767E+15][9.4666E+14, 9.46746E+14]
lintarraytag {4}	Array ohne Offset	Date	Ungültig: Date-Array wird nicht unterstützt.
lintarraytag {1}	Array ohne Offset	Double	Wert = 0
lintarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
lintarraytag [0,1] {2}	Array mit Offset	Double	Wert = [1.16767E+15, 9.4666E+14]
lintarraytag . 3	Bit	Boolean	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
lintarraytag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

*Datumswerte liegen in UTC-Zeit und nicht in lokalisierter Zeit vor.

Erweiterte Adressierung: REAL

Format	Unterstützte Datentypen	Hinweise
Standard	Boolean*, Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float****	Keine
Array-Element	Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float****	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Boolean-Array	1. Verwenden Sie diesen Fall, damit die Bit innerhalb von REAL in Array-Form vorliegen. **Hinweis:** Dies ist kein Array von REALs in boolescher Schreibweise. 2. Gilt nur für Bit innerhalb REAL. Beispiel: tag_1.0{32}. 3. Die Summe aus .bit und Array-Größe darf 32 Bit nicht überschreiten. Beispiel: tag_1.1{32} überschreitet REAL, tag_1.0{32} nicht. 4. Array-Größe muss ein Vielfaches von acht (8) sein.
Array ohne Offset	Byte-Array, Char-Array**, Word-Array, Short-Array, BCD-Array***, DWord-Array, Long-Array, LBCD-Array,	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.

Format	Unterstützte Datentypen	Hinweise
	Float-Array****	
Array mit Offset	Byte-Array, Char-Array**, Word-Array, Short-Array, BCD-Array***, DWord-Array, Long-Array, LBCD-Array, Float-Array****	Das Controller-Tag muss ein Array sein.
Bit	Boolean	1. Der Bereich ist von 0 bis 31 begrenzt. 2. Wenn das Controller-Tag ein Array ist, muss der Bit-Klassenreferenz eine Array-Element-Klassenreferenz vorangestellt werden. Beispiel: tag_1 [2,2,3].0. **Hinweis:** Float ist auf DWord festgelegt, um das Referenzieren von Bit zu ermöglichen.
String	Zeichenfolge	1. Wenn auf ein einzelnes Element zugegriffen wird, muss das Controller-Tag kein Array sein. **Hinweis:** Der Wert der Zeichenfolge ist das ASCII-Äquivalent des REAL-Werts (auf 255 fixiert). Beispiel: SINT = 65 (dezimal) = "A". 2. Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein. Der Wert der Zeichenfolge ist das mit Null beendete ASCII-Äquivalent aller REALs (auf 255 fixiert) in der Zeichenfolge. 1 Zeichen in Zeichenfolge = 1 REAL, auf 255 fixiert. **Hinweis:** REAL-Zeichenfolgen sind nicht gepackt. Verwenden Sie für eine größere Effizienz stattdessen SINT-Zeichenfolgen oder die STRING-Struktur.

*Werte ungleich Null sind auf "wahr" fixiert.
**Werte, die 255 überschreiten, sind auf 255 fixiert.
***Werte, die 65535 überschreiten, sind auf 65535 fixiert.
****Der Float-Wert ist eine gültige IEEE-Gleitkommazahl mit einfacher Präzision.

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

REAL-Controller-Tag - realtag = 512.5 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
realtag	Standard	Boolean	Wert = wahr
realtag	Standard	Byte	Wert = 255
realtag	Standard	Word	Wert = 512
realtag	Standard	DWord	Wert = 512
realtag	Standard	Float	Wert = 512.5
realtag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist auch ungültig.
realtag [3]	Array-Element	DWord	Ungültig: Tag ist kein Array.
realtag {3}	Array ohne Offset	DWord	Ungültig: Tag ist kein Array.
realtag {1}	Array ohne Offset	Float	Wert = [512.5]
realtag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
realtag [3] {1}	Array mit Offset	Float	Ungültig: Tag ist kein Array.

Server-Tag-Adresse	Format	Datentyp	Hinweise
realtag . 3	Bit	Boolean	Wert = wahr
realtag . 0 {32}	Array ohne Offset	Boolean	Wert = [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,...0] Bit-Wert für 512
realtag / 1	Zeichenfolge	Zeichenfolge	Wert = nicht druckbares Zeichen = 255 dezimal
realtag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Tag ist kein Array.

REAL-Array-Controller-Tag - realarraytag [4,4] = [[82.1,69.2,65.3,76.4],[256.5,257.6,258.7,259.8],[9.0,10.0,11.0,12.0],[13.0,14.0,15.0,16.0]]

Server-Tag-Adresse	Format	Datentyp	Hinweise
realarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
realarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
realarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
realarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.
realarraytag	Standard	Float	Ungültig: Tag kann kein Array sein.
realarraytag [3]	Array-Element	Float	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
realarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
realarraytag [1,3]	Array-Element	Float	Wert = 259,8
realarraytag {10}	Array ohne Offset	Byte	Wert = [82,69,65,76,255,255,255,255,9,10]
realarraytag {2} {5}	Array ohne Offset	Float	Wert = [82.1,69.2,65.3,76.4,256.5] [257.6,258.7,259.8,9,10]
realarraytag {1}	Array ohne Offset	Float	Wert = 82.1
realarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
realarraytag [1,3] {4}	Array mit Offset	Float	Wert = [259.8,9.0,10.0,11.0]
realarraytag . 3	Bit	Boolean	Ungültig: Tag muss unteilbare Position referenzieren.
realarraytag [1,3] . 3	Bit	Boolean	Wert = 0
realarraytag [1,3] . 0 {32}	Array ohne Offset	Boolean	Wert = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] Bit-Wert für 259
realarraytag / 1	Zeichenfolge	Zeichenfolge	Wert = "R"
realarraytag / 3	Zeichenfolge	Zeichenfolge	Wert = "REAL"

Erweiterte Adressierung: USINT

Format	Unterstützte Datentypen	Hinweise
Standard	Byte	Keine
Array-Element	Byte	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Boolean-Array	1. Verwenden Sie diesen Fall, damit die Bit innerhalb von USINT in Array-Form vorliegen. **Hinweis:** Dies ist kein Array von USINTs in boolescher Schreibweise.

Format	Unterstützte Datentypen	Hinweise
		2. Gilt nur für Bit innerhalb USINT. Beispiel: tag_1.0{8}. 3. Die Summe aus .bit und Array-Größe darf 8 Bit nicht überschreiten. Beispiel: tag_1.1{8} überschreitet USINT, tag_1.0{8} nicht.
Array ohne Offset	Byte Array	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.
Array mit Offset	Byte Array	Das Controller-Tag muss ein Array sein.
Bit	Boolean	1. Der Bereich ist von 0 bis 7 begrenzt. 2. Wenn das Controller-Tag ein Array ist, muss der Bit-Klassenreferenz eine Array-Element-Klassenreferenz vorangestellt werden. Beispiel: tag_1 [2,2,3].0.
String	k.A.	Nicht unterstützt

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

USINT-Controller-Tag - usinttag = 122 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
usinttag	Standard	Boolean	Wert = wahr
usinttag	Standard	Byte	Wert = 122
usinttag	Standard	Word	Ungültig: Word wird nicht unterstützt.
usinttag	Standard	DWord	Ungültig: DWord wird nicht unterstützt.
usinttag	Standard	Float	Ungültig: Float wird nicht unterstützt.
usinttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist auch ungültig.
usinttag [3]	Array-Element	Byte	Ungültig: Tag ist kein Array.
usinttag {3}	Array ohne Offset	Byte	Ungültig: Tag ist kein Array.
usinttag {1}	Array ohne Offset	Byte	Wert = [122]
usinttag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
usinttag [3] {1}	Array mit Offset	Byte	Ungültig: Tag ist kein Array.
usinttag . 3	Bit	Boolean	Wert = wahr
usinttag . 0 {8}	Array ohne Offset	Boolean	Wert = [0,1,0,1,1,1,1,0] Bit-Wert von 122
usinttag / 1	Zeichenfolge	Zeichenfolge	Wert = "z"
usinttag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Tag ist kein Array.

USINT-Array-Controller-Tag - usintarraytag [4,4] = [[83,73,78,84],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

Server-Tag-Adresse	Format	Datentyp	Hinweise
usintarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
usintarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
usintarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
usintarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.

Server-Tag-Adresse	Format	Datentyp	Hinweise
usintarraytag	Standard	Float	Ungültig: Tag kann kein Array sein.
usintarraytag [3]	Array-Element	Byte	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
usintarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
usintarraytag [1,3]	Array-Element	Byte	Wert = 8
usintarraytag {10}	Array ohne Offset	Byte	Wert = [83,73,78,84,5,6,7,8,9,10]
usintarraytag {2} {5}	Array ohne Offset	Word	Wert = [83,73,78,84,5] [6,7,8,9,10]
usintarraytag {1}	Array ohne Offset	Byte	Wert = 83
usintarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
usintarraytag [1,3] {4}	Array mit Offset	Byte	Wert = [8,9,10,11]
usintarraytag . 3	Bit	Boolean	Ungültig: Tag muss unteilbare Position referenzieren.
usintarraytag [1,3] . 3	Bit	Boolean	Wert = 1
usintarraytag [1,3] . 0 {8}	Array ohne Offset	Boolean	Wert = [0,0,0,1,0,0,0,0]
usintarraytag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
usintarraytag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

Erweiterte Adressierung: UINT

Format	Unterstützte Datentypen	Hinweise
Standard	Word, BCD	Keine
Array-Element	Word, BCD	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Boolean-Array	1. Verwenden Sie diesen Fall, damit die Bit innerhalb von UINT in Array-Form vorliegen. **Hinweis:** Dies ist kein Array von UINTs in boolescher Schreibweise. 2. Gilt nur für Bit innerhalb UINT. Beispiel: tag_1.0{16}. 3. Die Summe aus .bit und Array-Größe darf 8 Bit nicht überschreiten. Beispiel: tag_1.1{16} überschreitet UINT, tag_1.0{16} nicht. 4. Array-Größe muss ein Vielfaches von acht (8) sein.
Array ohne Offset	Word Array, BCD Array	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.
Array mit Offset	Word Array, BCD Array	Das Controller-Tag muss ein Array sein.
Bit	Boolean	1. Der Bereich ist von 0 bis 15 begrenzt. 2. Wenn das Controller-Tag ein Array ist, muss der Bit-Klassenreferenz eine Array-Element-Klassenreferenz vorangestellt werden. Beispiel: tag_1 [2,2,3].0.

Format	Unterstützte Datentypen	Hinweise
String	k.A.	Nicht unterstützt

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

UINT-Controller-Tag - uinttag = 65534 (dezimal)

Server-Tag-Adresse	Klasse	Datentyp	Hinweise
uinttag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
uinttag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.
uinttag	Standard	Word	Wert = 65534
uinttag	Standard	DWord	Ungültig: DWord wird nicht unterstützt.
uinttag	Standard	Float	Ungültig: Float wird nicht unterstützt.
uinttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist ungültig.
uinttag [3]	Array-Element	Byte	Ungültig: Tag ist kein Array.
uinttag {3}	Array ohne Offset	Byte	Ungültig: Tag ist kein Array.
uinttag {1}	Array ohne Offset	Byte	Wert = [65534]
uinttag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
uinttag [3] {1}	Array mit Offset	Word	Ungültig: Tag ist kein Array.
uinttag . 3	Bit	Boolean	Wert = wahr
uinttag . 0 {16}	Array ohne Offset	Boolean	Wert = [0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] Bit-Wert von 65534
uinttag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
uinttag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

UINT-Array-Controller-Tag - uintarraytag [4,4] = [[73,78,84,255],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

Server-Tag-Adresse	Format	Datentyp	Hinweise
uintarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
uintarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
uintarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
uintarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.
uintarraytag	Standard	Float	Ungültig: Tag kann kein Array sein.
uintarraytag [3]	Array-Element	Word	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
uintarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
uintarraytag [1,3]	Array-Element	Word	Wert = 259
uintarraytag {10}	Array ohne Offset	BCD	Wert =[49,54,54,165,100,101,102,103,9,10]
uintarraytag {2} {5}	Array ohne Offset	Word	Wert = [73,78,84,255,256] [257,258,259,9,10]
uintarraytag {1}	Array ohne Offset	Word	Wert = 73

Server-Tag-Adresse	Format	Datentyp	Hinweise
uintarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
uintarraytag [1,3] {4}	Array mit Offset	Byte	Wert = [259,9,10,11]
uintarraytag . 3	Bit	Boolean	Ungültig: Tag muss unteilbare Position referenzieren.
uintarraytag [1,3] . 3	Bit	Boolean	Wert = 0
uintarraytag [1,3] . 0 {16}	Array ohne Offset	Boolean	Wert = [0,0,0,1,0,0,0,0] Wert = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] Bit-Wert für 259
uintarraytag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
uintarraytag / 3	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

Erweiterte Adressierung: UDINT

Format	Unterstützte Datentypen	Hinweise
Standard	DWord, LBCD	Keine
Array-Element	DWord, LBCD	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Boolean-Array	1. Verwenden Sie diesen Fall, damit die Bit innerhalb von UDINT in Array-Form vorliegen. **Hinweis:** Dies ist kein Array von UDINTs in boolescher Schreibweise. 2. Gilt nur für Bit innerhalb UDINT. Beispiel: tag_1.0{32}. 3. Die Summe aus .bit und Array-Größe darf 32 Bit nicht überschreiten. Beispiel: tag_1.1{32} überschreitet UDINT, tag_1.0{32} nicht. 4. Array-Größe muss ein Vielfaches von acht (8) sein.
Array ohne Offset	DWord-Array, LBCD-Array	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.
Array mit Offset	DWord-Array, LBCD-Array	Das Controller-Tag muss ein Array sein.
Bit	Boolean	1. Der Bereich ist von 0 bis 31 begrenzt. 2. Wenn das Controller-Tag ein Array ist, muss der Bit-Klassenreferenz eine Array-Element-Klassenreferenz vorangestellt werden. Beispiel: tag_1 [2,2,3].0
String	k.A.	Nicht unterstützt

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

UDINT-Controller-Tag - udinttag = 70000 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
udinttag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
udinttag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.
udinttag	Standard	Word	Ungültig: Word wird nicht unterstützt.

Server-Tag-Adresse	Format	Datentyp	Hinweise
udinttag	Standard	DWord	Wert = 70000
udinttag	Standard	LBCD	Wert = 11170
udinttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist ungültig.
udinttag [3]	Array-Element	DWord	Ungültig: Tag ist kein Array.
udinttag {3}	Array ohne Offset	DWord	Ungültig: Tag ist kein Array.
udinttag {1}	Array ohne Offset	DWord	Wert = [70000]
udinttag {1}	Array ohne Offset	Boolean	Ungültig: Boolesches Array wird nicht unterstützt.
udintag [3] {1}	Array mit Offset	DWord	Ungültig: Tag ist kein Array.
udinttag . 3	Bit	Boolean	Wert = falsch
udinttag . 0 {32}	Array ohne Offset	Boolean	Wert = [0,0,0,0,1,1,1,0,1,0,0,0,1,0,0,0,1,0,...0] Bit-Wert für 70000
udinttag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
udinttag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

UDINT-Array-Controller-Tag - udintarraytag [4,4] = [[68,73,78,84],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

Server-Tag-Adresse	Format	Datentyp	Hinweise
udintarraytag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
udintarraytag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.
udintarraytag	Standard	Word	Ungültig: Word wird nicht unterstützt.
udintarraytag	Standard	DWord	Ungültig: Tag kann kein Array sein.
udintarraytag	Standard	Float	Ungültig: Float wird nicht unterstützt.
udintarraytag [3]	Array-Element	DWord	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
udintarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
udintarraytag [1,3]	Array-Element	DWord	Wert = 259
udintarraytag {10}	Array ohne Offset	LCBD	Wert = [44,49,54,54,100,101,102,103,9,10]]
udintarraytag {2}{5}	Array ohne Offset	DWord	Wert = [68,73,78,84,256] [257,258,259,9,10]
udintarraytag {1}	Array ohne Offset	DWord	Wert = 68
udintarraytag {1}	Array ohne Offset	Boolean	Ungültig: Fehlerhafter Datentyp.
udintarraytag [1,3] {4}	Array mit Offset	DWord	Wert = [259,9,10,11]
udintarraytag . 3	Bit	Boolean	Ungültig: Tag muss unteilbare Position referenzieren.
udintarraytag [1,3] . 3	Bit	Boolean	Wert = falsch
udintarraytag [1,3] .0 {32}	Array ohne Offset	Boolean	Wert = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] Bit-Wert für 259
udintarraytag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
udintarraytag / 3	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

Erweiterte Adressierung: ULINT

Format	Unterstützte Datentypen	Hinweise
Standard	Double*	Keine
Array-Element	Double*	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Double, Array*	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.
Array mit Offset	Double, Array*	Das Controller-Tag muss ein Array sein.
Bit	k.A.	Nicht unterstützt
String	k.A.	Nicht unterstützt

*Der Double-Wert entspricht dem Nennwert des Controller-Tags in Gleitkommazahlform (Nicht-IEEE-Gleitkommazahl).

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

ULINT-Controller-Tag - ulinttag = 1.8446744073709560e+19 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
ulinttag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
ulinttag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.
ulinttag	Standard	Word	Ungültig: Word wird nicht unterstützt.
ulinttag	Standard	Double	Wert = 1.8446744073709560e+19
ulinttag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist ungültig.
ulinttag [3]	Array-Element	Double	Ungültig: Tag ist kein Array.
ulinttag {3}	Array ohne Offset	Double	Ungültig: Tag ist kein Array.
ulinttag {1}	Array ohne Offset	Double	Wert = [1.8446744073709560e+19]
ulinttag {1}	Array ohne Offset	Boolean	Ungültig: Boolesches Array wird nicht unterstützt.
ulintag [3] {1}	Array mit Offset	Double	Ungültig: Tag ist kein Array.
ulinttag . 3	Bit	Boolean	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
ulinttag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

ULINT-Array-Controller-Tag - ulintarraytag [2,2] = [0, 1.16767E+15],[9.4666E+14, 1.8446744073709560e+19]

Server-Tag-Adresse	Format	Datentyp	Hinweise
ulintarraytag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
ulintarraytag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.
ulintarraytag	Standard	Word	Ungültig: Word wird nicht unterstützt.
ulintarraytag	Standard	Double	Ungültig: Tag kann kein Array sein.
ulintarraytag	Standard	Date	Ungültig: Date wird nicht unterstützt.
ulintarraytag [1]	Array-Element	Double	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
ulintarraytag [1,1]	Array-Ele-	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.

Server-Tag-Adresse	Format	Datentyp	Hinweise
	ment		
ulintarraytag [1,1]	Array-Element	Double	Wert = 1.8446744073709560e+19
ulintarraytag {4}	Array ohne Offset	Double	Wert = [0, 1.16767E+15, 9.4666E+14, 1.8446744073709560e+19]
ulintarraytag {2} {2}	Array ohne Offset	Double	Wert =[0, 1.16767E+15][9.4666E+14, 1.8446744073709560e+19]
ulintarraytag {4}	Array ohne Offset	Date	Ungültig: Date-Array wird nicht unterstützt.
ulintarraytag {1}	Array ohne Offset	Double	Wert = 0
ulintarraytag {1}	Array ohne Offset	Boolean	Ungültig: Boolesches Array wird nicht unterstützt.
ulintarraytag [0,1] {2}	Array mit Offset	Double	Wert = [1.16767E+15, 9.4666E+14]
ulintarraytag . 3	Bit	Boolean	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
ulintarraytag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

Erweiterte Adressierung: LREAL

Format	Unterstützte Datentypen	Hinweise
Standard	Double	Keine
Array-Element	Double	Das Controller-Tag muss ein Array sein.
Array ohne Offset	Double-Array	Wenn auf mehr als ein einzelnes Element zugegriffen wird, muss das Controller-Tag ein Array sein.
Array mit Offset	Double-Array	Das Controller-Tag muss ein Array sein.
Bit	k.A.	Nicht unterstützt
String	k.A.	Nicht unterstützt

Beispiele

Die hervorgehobenen Beispiele stellen allgemeine Anwendungsfälle dar.

LREAL-Controller-Tag - lrealtag = 1.7976931348623157E+308 (dezimal)

Server-Tag-Adresse	Format	Datentyp	Hinweise
lrealtag	Standard	Boolean	Ungültig: Boolean wird nicht unterstützt.
lrealtag	Standard	Byte	Ungültig: Byte wird nicht unterstützt.
lrealtag	Standard	Word	Ungültig: Word wird nicht unterstützt.
lrealtag	Standard	Double	Wert = 1.7976931348623157E+308
lrealtag [3]	Array-Element	Boolean	Ungültig: Tag ist kein Array. Boolean ist auch ungültig.
lrealtag {1}	Array ohne Offset	Double	Wert = [1.7976931348623157E+308]
lrealtag {1}	Array ohne Offset	Boolean	Ungültig: Boolesches Array wird nicht unterstützt.

Server-Tag-Adresse	Format	Datentyp	Hinweise
lrealtag [3] {1}	Array mit Offset	Double	Ungültig: Tag ist kein Array.
lrealtag . 3	Bit	Boolean	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
lrealtag . 0 {32}	Array ohne Offset	Boolean	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
lrealtag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
lrealtag / 4	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

LREAL-Array-Controller-Tag - lrealarraytag [4,4] = [[82.1,69.2,65.3,76.4],[256.5,257.6,258.7,259.8],[9.0,10.0,11.0,12.0],[13.0,14.0,15.0,16.0]]

Server-Tag-Adresse	Format	Datentyp	Hinweise
lrealarraytag	Standard	Boolean	Ungültig: Tag kann kein Array sein.
lrealarraytag	Standard	Byte	Ungültig: Tag kann kein Array sein.
lrealarraytag	Standard	Word	Ungültig: Tag kann kein Array sein.
lrealarraytag	Standard	Double	Ungültig: Tag kann kein Array sein.
lrealarraytag [3]	Array-Element	Double	Ungültig: Server-Tag weist fehlende Adresse für Dimension 2 auf.
lrealarraytag [1,3]	Array-Element	Boolean	Ungültig: Boolean nicht für Array-Elemente zulässig.
lrealarraytag [1,1]	Array-Element	Double	Wert = 257.6
lrealarraytag {2} {5}	Array ohne Offset	Double	Wert = [82.1,69.2,65.3,76.4,256.5] [257.6,258.7,259.8,9,10]
lrealarraytag {1}	Array ohne Offset	Double	Wert = 82.1
lrealarraytag {1}	Array ohne Offset	Boolean	Ungültig: Boolesches Array wird nicht unterstützt.
lrealarraytag [1,3] {4}	Array mit Offset	Double	Wert = [259.8,9.0,10.0,11.0]
lrealarraytag . 3	Bit	Boolean	Ungültig: Syntax-/Datentyp wird nicht unterstützt.
lrealarraytag / 1	Zeichenfolge	Zeichenfolge	Ungültig: Syntax-/Datentyp wird nicht unterstützt.

Dateiauflistung

Wählen Sie eine Verknüpfung in der Liste unten aus, um Informationen zu einer bestimmten Datei zu erhalten, die von verschiedenen Gerätemodellen unterstützt wird.

Ausgabedateien
Eingabedateien
Statusdateien
Binäre Dateien
Zeitgeberdateien
Zählerdateien
Steuerdateien
Ganzzahldateien
Float-Dateien
ASCII-Dateien
String-Dateien
BCD-Dateien
Long-Dateien
MicroLogix-PID-Dateien

PID-Dateien
MicroLogix-Meldungsdateien
Meldungsdateien
Blocktransferdateien

Funktionsdateiauflistung

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)
RTC-Datei (Real Time Clock, Echtzeituhr)
CS0-Datei (Channel 0 Communication Status, Kommunikationsstatus für Kanal 0)
CS1 (Channel 1 Communication Status, Kommunikationsstatus für Kanal 1)
IOS-Datei (I/O Module Status, E/A-Modulstatus)

Weitere Informationen zu Gerätemodellen und ihren unterstützten Dateien finden Sie unter Adressbeschreibungen.

Ausgabedateien

Die Syntax für den Zugriff auf Daten in der Ausgabedatei ist je nach SPS-Modell unterschiedlich. Arrays werden für Ausgabedateien nicht unterstützt. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
O:<Wort>	Short, **Word**, BCD	Lesen/Schreiben
O:<Wort>/<Bit>	**Boolean**	Lesen/Schreiben
O/Bit	**Boolean**	Lesen/Schreiben

Hinweis: Wort- und Bit-Adressinformationen liegen für PLC-5-Modelle im Oktalformat vor. Dies entspricht der Konvention der Programmiersoftware.

MicroLogix-Syntax

Syntax	Datentyp	Zugriff
O:<Wort>	Short, **Word**, BCD	Lesen/Schreiben
O:<Wort>/<Bit>	**Boolean**	Lesen/Schreiben
O/Bit	**Boolean**	Lesen/Schreiben

MicroLogix-Modelle verfügen über zwei E/A-Typen: eingebettete E/A und Erweiterungs-E/A (nicht zutreffend für MicroLogix 1000). Eingebettete E/A befindet sich bei der CPU-Basiseinheit, während Erweiterungs-E/A an die CPU-Basiseinheit angeschlossen ist. In der Tabelle unten werden die E/A-Funktionen jedes MicroLogix-Modells aufgelistet.

MicroLogix-Modell	Eingebettete E/A	Erweiterungs-E/A
1000	Steckplatz 0	k.A.
1100	Steckplatz 0	Steckplätze 1-4
1200	Steckplatz 0	Steckplätze 1-6
1400	Steckplatz 0	Steckplätze 1-7
1500	Steckplatz 0	Steckplätze 1-16

Die Adresssyntax für MicroLogix-E/A referenziert ein nullbasiertes Wort-Offset und keinen Steckplatz. Benutzer müssen das Wort-Offset gegenüber einem bestimmten Steckplatz bestimmen. Dies erfordert Kenntnisse über die Module und deren jeweilige Größe in Wörtern. In der Tabelle unten wird die Größe einiger verfügbarer Module angegeben. Es wird jedoch empfohlen, in der MicroLogix-Dokumentation und im Controller-Projekt nachzusehen, um die tatsächliche Wortgröße des Moduls zu bestimmen.

MicroLogix - Eingebettete E/A-Wortgrößen

MicroLogix-Modell	# Eingabewörter	# Ausgabewörter
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

MicroLogix - Erweiterungs-E/A-Wortgrößen

Module	# Eingabewörter	# Ausgabewörter
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1

Module	# Eingabewörter	# Ausgabewörter
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

Berechnung

Ausgabewort-Offset für Steckplatz x = # Ausgabewörter in Steckplatz 0 durch Steckplatz (x-1).

Hinweise:

1. Die Eingebettete E/A muss beim Offset auf Erweiterungs-E/A berücksichtigt werden.
2. Die Anzahl der Eingabewörter ist kein Faktor bei der Berechnung des Ausgabewort-Offsets.

E/A-Beispiel

Annahme
Steckplatz 0 = MicroLogix 1500 LRP-Reihe C = 4 Ausgabewörter
Steckplatz 1 = 1769-OF2 = 2 Ausgabewörter
Steckplatz 2 = 1769-OW8 = 1 Ausgabewort
Steckplatz 3 = 1769-IA16 = 0 Ausgabewort
Steckplatz 4 = 1769-OF8V = 9 Ausgabewort
Bit 5 von Steckplatz 4 = 4 + 2 + 1 = 7 Wörter = O:7/5

SLC 500-Syntax

Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
O:<Steckplatz>	Short, **Word**, BCD	Schreibgeschützt
O:<Steckplatz>.<Wort>	Short, **Word**, BCD	Schreibgeschützt
O:<Steckplatz>/<Bit>	**Boolean**	Schreibgeschützt
O:<Steckplatz>.<Wort>/<Bit>	**Boolean**	Schreibgeschützt

Bereiche

SPS-Modell	Min. Steckplatz	Max. Steckplatz	Max. Wort
MicroLogix	k.A.	k.A.	2047
Feste E/A für SLC 500	k.A.	k.A.	1
Modulare E/A für SLC 500	1	30	*
PLC-5-Reihe	k.A.	k.A.	277 (oktal)

*Die Anzahl der für jedes E/A-Modul verfügbaren Eingabe- oder Ausgabewörter ist unter Richtlinien für modulare E/A-Auswahl für SLC 500 zu finden.

Beispiele

MicroLogix	Description
O:0	Wort 0
O/2	Bit 2
O:0/5	Bit 5

Feste E/A für SLC 500	Description
O:0	Wort 0
O:1	Wort 1
O/16	Bit 16
O:1/0	Bit 0 Wort 1 (identisch mit O/16)

PLC5*	Description
O:0	Wort 0
O:37	Wort 31 (37 oktal = 31 dezimal)
O/42	Bit 34 (42 oktal = 34 dezimal)
O:2/2	Bit 2 Wort 2 (identisch mit O/42)

*Adressen liegen im Oktalformat vor.

Modulare E/A für SLC 500	Description
O:1	Wort 0 Steckplatz 1
O:1.0	Wort 0 Steckplatz 1 (identisch mit O:1)
O:12	Wort 0 Steckplatz 12
O:12.2	Wort 2 Steckplatz 12
O:4.0/0	Bit 0 Wort 0 Steckplatz 4
O:4/0	Bit 0 Steckplatz 4 (identisch mit O:4.0/0)
O:4.2/0	Bit 0 Wort 2 Steckplatz 4
O:4/32	Bit 32 Steckplatz 4 (identisch mit O:4.2/0)

Eingabedateien

Die Syntax für den Zugriff auf Daten in der Eingabedatei ist je nach SPS-Modell unterschiedlich. Arrays werden für Eingabedateien nicht unterstützt. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
I:<Wort>	Short, **Word**, BCD	Lesen/Schreiben
I:<Wort>/<Bit>	**Boolean**	Lesen/Schreiben
I/Bit	**Boolean**	Lesen/Schreiben

Hinweis: Wort- und Bit-Adressinformationen liegen für PLC-5-Modelle im Oktalformat vor. Dies entspricht der Konvention der Programmiersoftware.

MicroLogix-Syntax

Syntax	Datentyp	Zugriff
I:<Wort>	Short, **Word**, BCD	Lesen/Schreiben
I:<Wort>/<Bit>	**Boolean**	Lesen/Schreiben
I/Bit	**Boolean**	Lesen/Schreiben

MicroLogix-Modelle verfügen über zwei E/A-Typen: eingebettete E/A und Erweiterungs-E/A (nicht zutreffend für MicroLogix 1000). Eingebettete E/A befindet sich bei der CPU-Basiseinheit, während Erweiterungs-E/A an die CPU-Basiseinheit angeschlossen ist. In der Tabelle unten werden die E/A-Funktionen jedes MicroLogix-Modells aufgelistet.

MicroLogix-Modell	Eingebettete E/A	Erweiterungs-E/A
1000	Steckplatz 0	k.A.
1100	Steckplatz 0	Steckplätze 1-4
1200	Steckplatz 0	Steckplätze 1-6
1400	Steckplatz 0	Steckplätze 1-7
1500	Steckplatz 0	Steckplätze 1-16

Die Adresssyntax für MicroLogix-E/A referenziert ein nullbasiertes Wort-Offset und keinen Steckplatz. Benutzer müssen das Wort-Offset gegenüber einem bestimmten Steckplatz bestimmen. Dies erfordert Kenntnisse über die Module und deren jeweilige Größe in Wörtern. In der Tabelle unten wird die Größe einiger verfügbarer Module angegeben. Es wird jedoch empfohlen, in der MicroLogix-Dokumentation und im Controller-Projekt nachzusehen, um die tatsächliche Wortgröße eines Moduls zu bestimmen.

MicroLogix - Eingebettete E/A-Wortgrößen

MicroLogix-Modell	# Eingabewörter	# Ausgabewörter
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

MicroLogix - Erweiterungs-E/A-Wortgrößen

Module	# Eingabewörter	# Ausgabewörter
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1

Module	# Eingabewörter	# Ausgabewörter
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

Berechnung

Eingabewort-Offset für Steckplatz x = # Eingabewörter in Steckplatz 0 durch Steckplatz (x-1).

Hinweise:

1. Die Eingebettete E/A muss beim Offset auf Erweiterungs-E/A berücksichtigt werden.
2. Die Anzahl der Ausgabewörter ist kein Faktor bei der Berechnung des Eingabewort-Offsets.

E/A-Beispiel

Annahme
Steckplatz 0 = MicroLogix 1500 LRP-Reihe C = 4 Eingabewörter
Steckplatz 1 = 1769-OF2 = 2 Eingabewörter
Steckplatz 2 = 1769-OW8 = 0 Eingabewort
Steckplatz 3 = 1769-IA16 = 1 Eingabewort
Steckplatz 4 = 1769-OF8V = 11 Eingabewörter
Bit 5 von Steckplatz 3 = 4 + 2 = 6 Wörter = I:6/5

SLC 500-Syntax

Syntax	Datentyp	Zugriff
I:<Steckplatz>	Short, **Word**, BCD	Schreibgeschützt
I:<Steckplatz>.<Wort>	Short, **Word**, BCD	Schreibgeschützt
I:<Steckplatz>/<Bit>	**Boolean**	Schreibgeschützt
I:<Steckplatz>.<Wort>/<Bit>	**Boolean**	Schreibgeschützt

Bereiche

SPS-Modell	Min. Steckplatz	Max. Steckplatz	Max. Wort
MicroLogix	k.A.	k.A.	2047
Feste E/A für SLC 500	k.A.	k.A.	1

SPS-Modell	Min. Steckplatz	Max. Steckplatz	Max. Wort
Modulare E/A für SLC 500	1	30	*
PLC-5-Reihe	k.A.	k.A.	277 (oktal)

*Die Anzahl der für jedes E/A-Modul verfügbaren Eingabe- oder Ausgabewörter ist unter Richtlinien für modulare E/A-Auswahl für SLC 500 zu finden.

Beispiele

MicroLogix	Description
I:0	Wort 0
I/2	Bit 2
I:1/5	Bit 5 Wort 1

Feste E/A für SLC 500	Description
I:0	Wort 0
I:1	Word 1
I/16	Bit 16
I:1/0	Bit 0 Wort 1 (identisch mit I/16)

PLC5*	Description
I:0	Wort 0
I:10	Word 8 (10 oktal = 8 dezimal)
I/20	Bit 16 (20 oktal = 16 dezimal)
I:1/0	Bit 0 Wort 1 (identisch mit I/20)

*Adressen liegen im Oktalformat vor.

Modulare E/A für SLC 500	Description
I:1	Wort 0 Steckplatz 1
I:1.0	Wort 0 Steckplatz 1 (identisch mit I:1)
I:12	Wort 0 Steckplatz 12
I:12.2	Wort 2 Steckplatz 12
I:4.0/0	Bit 0 Wort 0 Steckplatz 4
I:4/0	Bit 0 Steckplatz 4 (identisch mit I:4.0/0)
I:4.2/0	Bit 0 Wort 2 Steckplatz 4
I:4/32	Bit 32 Steckplatz 4 (identisch mit I:4.2/0)

Statusdateien

Geben Sie für den Zugriff auf Statusdateien ein Wort und ein optionales Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
S:<Wort>	Short, **Word**, BCD, DWord, Long, LBCD	Lesen/Schreiben
S:<Wort> [Zeilen][Spalten]	Short, **Word**, BCD, DWord, Long, LBCD (Array-Typ)	Lesen/Schreiben
S:<Wort> [Spalten]	Short, **Word**, BCD, DWord, Long, LBCD (Array-Typ)	Lesen/Schreiben
S:<Wort>/<Bit>	**Boolean**	Lesen/Schreiben
S/Bit	**Boolean**	Lesen/Schreiben

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 Word-Werte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Max. Wort
MicroLogix	999
Feste E/A für SLC 500	96
Modulare E/A für SLC 500	999
PLC-5-Reihe	999

Die maximale Wortposition ist eine weniger, wenn als 32-Bit-Datentyp (z.B. Long, DWord oder Long-BCD) zugegriffen wird.

Beispiele

Beispiel	Description
S:0	Wort 0
S/26	Bit 26
S:4/15	Bit 15, Wort 4
S:10 [16]	Array mit 16 Elementen, beginnend bei Wort 10.
S:0 [4] [8]	Array mit 4 mal 8 Elementen, beginnend bei Wort 0

Binäre Dateien

Geben Sie für den Zugriff auf binäre Dateien eine Dateinummer, ein Wort und ein optionales Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
B<Datei>:<Wort>	Short, **Word**, BCD, DWord, Long, LBCD	Lesen/Schreiben
B<Datei>:<Wort> [Zeilen][Spalten]	Short, **Word**, BCD, DWord, Long, LBCD (Array-Typ)	Lesen/Schreiben
B<Datei>:<Wort> [Spalten]	Short, **Word**, BCD, DWord, Long, LBCD (Array-Typ)	Lesen/Schreiben
B<Datei>:<Wort>/<Bit>	**Boolean**	Lesen/Schreiben
B<Datei>/Bit	**Boolean**	Lesen/Schreiben

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 Wortwerte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
MicroLogix	3, 9-999	999
Feste E/A für SLC 500	3, 9-255	255
Modulare E/A für SLC 500	3, 9-999	999
PLC-5-Reihe	3-999	1999

Die maximale Wortposition ist eine weniger, wenn als 32-Bit-Datentyp (z.B. Long, DWord oder Long-BCD) zugegriffen wird.

Beispiele

Beispiel	Description
B3:0	Wort 0
B3/26	Bit 26
B12:4/15	Bit 15, Wort 4
B3:10 [20]	Array mit 20 Elementen, beginnend bei Wort 10
B15:0 [6][6]	Array mit 6 mal 6 Elementen, beginnend bei Wort 0

Zeitgeberdateien

Bei Zeitgeberdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
T<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
ACC	**Short**, Word	Lesen/Schreiben
PRE	**Short**, Word	Lesen/Schreiben
DN	**Boolean**	Schreibgeschützt
TT	**Boolean**	Schreibgeschützt
EN	**Boolean**	Schreibgeschützt

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	4, 9-999	999
Feste E/A für SLC 500	4, 9-255	255
Modulare E/A für SLC 500	4, 9-999	999
PLC-5-Reihe	3-999	1999

Beispiele

Beispiel	Description
T4:0.ACC	Akkumulator von Zeitgeber 0, Datei 4
T4:10.DN	Abschluss-Bit von Zeitgeber 10, Datei 4
T15:0.PRE	Voreinstellung von Zeitgeber 0, Datei 15

Zählerdateien

Bei Zählerdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
C<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
ACC	**Word**, Short	Lesen/Schreiben
PRE	**Word**, Short	Lesen/Schreiben
UA	**Boolean**	Schreibgeschützt
UN	**Boolean**	Schreibgeschützt
OV	**Boolean**	Schreibgeschützt
DN	**Boolean**	Schreibgeschützt
CD	**Boolean**	Schreibgeschützt
CU	**Boolean**	Schreibgeschützt

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	5, 9-999	999
Feste E/A für SLC 500	5, 9-255	255
Modulare E/A für SLC 500	5, 9-999	999
PLC-5-Reihe	3-999	1999

Beispiele

Beispiel	Description
C5:0.ACC	Akkumulator von Zähler 0, Datei 5
C5:10.DN	Abschluss-Bit von Zähler 10, Datei 5
C15:0.PRE	Voreinstellung von Zähler 0, Datei 15

Steuerdateien

Bei Steuerdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
R<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
LEN	**Word**, Short	Lesen/Schreiben
POS	**Word**, Short	Lesen/Schreiben
FD	**Boolean**	Schreibgeschützt
IN	**Boolean**	Schreibgeschützt
UL	**Boolean**	Schreibgeschützt
ER	**Boolean**	Schreibgeschützt
EM	**Boolean**	Schreibgeschützt
DN	**Boolean**	Schreibgeschützt
EU	**Boolean**	Schreibgeschützt
EN	**Boolean**	Schreibgeschützt

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	6, 9-999	999

SPS-Modell	Dateinummer	Max. Element
Feste E/A für SLC 500	6, 9-255	255
Modulare E/A für SLC 500	6, 9-999	999
PLC-5-Reihe	3-999	1999

Beispiele

Beispiel	Description
R6:0.LEN	Längenfeld von Steuerung 0, Datei 6
R6:10.DN	Abschluss-Bit von Steuerung 10, Datei 6
R15:18.POS	Positionsfeld von Steuerung 18, Datei 15

Ganzzahldateien

Geben Sie für den Zugriff auf Ganzzahldateien eine Dateinummer, ein Wort und ein optionales Bit im Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
N<Datei>:<Wort>	Short, **Word**, BCD, DWord, Long, LBCD	Lesen/Schreiben
N<Datei>:<Wort> [rows][cols]	Short, **Word**, BCD, DWord, Long, LBCD (Array-Typ)	Lesen/Schreiben
N<Datei>:<Wort> [cols]	Short, **Word**, BCD, DWord, Long, LBCD (Array-Typ)	Lesen/Schreiben
N<Datei>:<Wort>/<Bit>	**Boolean**	Lesen/Schreiben
N<Datei>/Bit	**Boolean**	Lesen/Schreiben

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 Wortwerte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
MicroLogix	7, 9-999	999
Feste E/A für SLC 500	7, 9-255	255
Modulare E/A für SLC 500	7, 9-999	999
PLC-5-Reihe	3-999	1999

Die maximale Wortposition ist eine weniger, wenn als 32-Bit-Datentyp (z.B. Long, DWord oder Long-BCD) zugegriffen wird.

Beispiele

Beispiel	Beschreibung
N7:0	Wort 0
N7/26	Bit 26
N12:4/15	Bit 15, Wort 4
N7:10 [8]	Array mit 8 Elementen, beginnend bei Wort 10
N15:0 [4] [5]	Array mit 4 von 5 Elementen, beginnend bei Wort 0

Float-Dateien

Geben Sie für den Zugriff auf Float-Dateien eine Dateinummer und ein Element an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
F<Datei>:<Element>	**Float**	Lesen/Schreiben
F<Datei>:<Element> [rows][cols]	**Float** (Array-Typ)	Lesen/Schreiben
F<Datei>:<Element> [cols]	**Float** (Array-Typ)	Lesen/Schreiben

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 8 Float-Werte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
MicroLogix	8-999	999
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	8-999	999
PLC-5-Reihe	3-999	1999

Beispiele

Beispiel	Beschreibung
F8:0	Float (Gleitkommazahl) 0
F8:10 [16]	Array mit 16 Elementen, beginnend bei Wort 10.
F15:0 [4] [4]	Array mit 16 Elementen, beginnend bei Wort 0

ASCII-Dateien

Geben Sie für den Zugriff auf Daten in einer ASCII-Datei eine Dateinummer und Zeichenposition an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
A<Datei>:<Char>	**Char**, Byte*	Lesen/Schreiben
A<Datei>:<Char> [Zeilen][Spalten]	**Char**, Byte*	Lesen/Schreiben
A<Datei>:<Char> [Spalten]	**Char**, Byte*	Lesen/Schreiben
A<Datei>:<Wort-Offset>/Länge	String**	Lesen/Schreiben

*Die Anzahl der Array-Elemente darf die angegebene Blockanforderungsgröße nicht überschreiten. Intern packt die SPS zwei Zeichen pro Wort in die Datei, wobei das High-Byte das erste Zeichen und das Low-Byte das zweite Zeichen enthält. Die SPS-Programmiersoftware ermöglicht Zugriff auf der Wortebene oder der Zwei-Zeichen-Ebene. ControlLogix-Ethernet-Treiber von Allen-Bradley ermöglicht Zugriff auf die Zeichenebene.

Wenn Sie die Programmiersoftware verwenden, würde bei "A10:0 = AB" der Wert 'A' im High-Byte von A10:0 und 'B' im Low-Byte gespeichert. Unter Verwendung von ControlLogix-Ethernet-Treiber von Allen-Bradley würden zwei Zuweisungen vorgenommen: "A10:0 = A" und "A10:1 = B". Dies würde zur Folge haben, dass dieselben Daten im SPS-Speicher gespeichert werden.

**Das Referenzieren dieser Datei als Zeichenfolgendaten ermöglicht Zugriff auf Daten an Wortgrenzen wie die Programmiersoftware. Die Länge kann aus bis zu 232 Zeichen bestehen. Wenn eine an das Gerät gesendete Zeichenfolge kürzer ist als die durch die Adresse angegebene Länge, beendet der Treiber die Zeichenfolge mit Null, bevor sie an den Controller gesendet wird.

Bereiche

SPS-Modell	Dateinummer	Max. Zeichen
MicroLogix	3-255	511
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	9-999	1999

SPS-Modell	Dateinummer	Max. Zeichen
PLC-5-Reihe	3-999	1999

Hinweis: Nicht alle MicroLogix- und SLC 500-SPS-Geräte unterstützen ASCII-Dateitypen. Weitere Informationen dazu finden Sie in der SPS-Dokumentation.

Beispiele

Beispiel	Description
A9:0	Zeichen 0 (High-Byte von Wort 0)
A27:10 [80]	Char-Array aus 80 Zeichen, das bei Zeichen 10 beginnt.
A15:0 [4] [16]	Char-Array aus 4 mal 16 Zeichen, das bei Zeichen 0 beginnt
A62:0/32	Char-Array aus 32 Zeichenfolgen, das bei Wort-Offset 0 beginnt.

String-Dateien

Geben Sie für den Zugriff auf String-Dateien eine Dateinummer und ein Element an. Zeichenfolgen sind mit Null beendete Arrays mit 82 Zeichen. Der Treiber platziert das Nullabschlusszeichen basierend auf der von der SPS zurückgegebenen Zeichenfolgenlänge. Die Standard-Datentypen werden **fett** dargestellt.

Hinweis: Arrays werden für String-Dateien nicht unterstützt.

Syntax	Datentyp	Zugriff
ST<Datei>:<Element><Feld>	**String**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Wort
MicroLogix	9-999	999
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	9-999	999
PLC-5-Reihe	3-999	999

Beispiele

Beispiel	Description
ST9:0	Zeichenfolge 0
ST18:10	Zeichenfolge 10

BCD-Dateien

Geben Sie für den Zugriff auf BCD-Dateien eine Dateinummer und ein Wort an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
D<Datei>:<Wort>	**BCD**, LBCD	Lesen/Schreiben
D<Datei>:<Wort> [Zeilen][Spalten]	**BCD**, LBCD (Array-Typ)	Lesen/Schreiben
D<Datei>:<Wort> [Spalten]	**BCD**, LBCD (Array-Typ)	Lesen/Schreiben

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 16 BCD-Werte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
MicroLogix	k.A.	k.A.
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	k.A.	k.A.
PLC-5-Reihe	3-999	999

Beispiele

Beispiel	Beschreibung
D9:0	Wort 0
D27:10 [16]	Array mit 16 Elementen, beginnend bei Wort 10.
D15:0 [4][8]	Array mit 32 Elementen, beginnend bei Wort 0.

Long-Dateien

Um auf lange Ganzzahldateien zuzugreifen, geben Sie eine Dateinummer und ein Element an. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
L<Datei>:<DWord>	Long, **DWord**, LBCD	Lesen/Schreiben
L<Datei>:<DWord> [Zeilen][Spalten]	Long, **DWord**, LBCD (Array-Typ)	Lesen/Schreiben
L<Datei>:<DWord> [Spalten]	Long, **DWord**, LBCD (Array-Typ)	Lesen/Schreiben

Die Anzahl der Array-Elemente (in Byte) darf die angegebene Blockanforderungsgröße nicht überschreiten. Dies bedeutet, dass die Array-Größe 8 Long-Werte bei einer Blockanforderungsgröße von 32 Byte nicht überschreiten darf.

Bereiche

SPS-Modell	Dateinummer	Max. Wort
MicroLogix	9-999	999
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	k.A.	k.A.
PLC-5-Reihe	k.A.	k.A.

Beispiele

Beispiel	Description
L9:0	Wort 0
L9:10 [8]	Array mit 8 Elementen, beginnend bei Wort 10
L15:0 [4] [5]	Array mit 4 von 5 Elementen, beginnend bei Wort 0

MicroLogix-PID-Dateien

Bei PID-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
PD<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
SPS	**Word**, Short	Lesen/Schreiben

Elementfeld	Datentyp	Zugriff
KC	**Word**, Short	Lesen/Schreiben
TI	**Word**, Short	Lesen/Schreiben
TD	**Word**, Short	Lesen/Schreiben
MAXS	**Word**, Short	Lesen/Schreiben
MINS	**Word**, Short	Lesen/Schreiben
ZCD	**Word**, Short	Lesen/Schreiben
CVH	**Word**, Short	Lesen/Schreiben
CVL	**Word**, Short	Lesen/Schreiben
LUT	**Word**, Short	Lesen/Schreiben
SPV	**Word**, Short	Lesen/Schreiben
CVP	**Word**, Short	Lesen/Schreiben
TM	**Boolean**	Lesen/Schreiben
AM	**Boolean**	Lesen/Schreiben
CM	**Boolean**	Lesen/Schreiben
OL	**Boolean**	Lesen/Schreiben
RG	**Boolean**	Lesen/Schreiben
SC	**Boolean**	Lesen/Schreiben
TF	**Boolean**	Lesen/Schreiben
DA	**Boolean**	Lesen/Schreiben
DB	**Boolean**	Lesen/Schreiben
UL	**Boolean**	Lesen/Schreiben
LL	**Boolean**	Lesen/Schreiben
SP	**Boolean**	Lesen/Schreiben
PV	**Boolean**	Lesen/Schreiben
DN	**Boolean**	Lesen/Schreiben
EN	**Boolean**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	3-255	255
Alle SLC	k.A.	k.A.
PLC-5	PID-Dateien	PID-Dateien

Beispiele

Beispiel	Description
PD14:0.KC	Proportionaler Zuwachs von PD 0 Datei 14
PD18:6.EN	PID-Aktivierungs-Bit von PD 6 Datei 18

PID-Dateien

Bei PID-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
PD<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
SP	**Real**	Lesen/Schreiben
KP	**Real**	Lesen/Schreiben
KI	**Real**	Lesen/Schreiben
KD	**Real**	Lesen/Schreiben
BIAS	**Real**	Lesen/Schreiben
MAXS	**Real**	Lesen/Schreiben
MINS	**Real**	Lesen/Schreiben
DB	**Real**	Lesen/Schreiben
SO	**Real**	Lesen/Schreiben
MAXO	**Real**	Lesen/Schreiben
MINO	**Real**	Lesen/Schreiben
UPD	**Real**	Lesen/Schreiben
PV	**Real**	Lesen/Schreiben
ERR	**Real**	Lesen/Schreiben
OUT	**Real**	Lesen/Schreiben
PVH	**Real**	Lesen/Schreiben
PVL	**Real**	Lesen/Schreiben
DVP	**Real**	Lesen/Schreiben
DVN	**Real**	Lesen/Schreiben
PVDB	**Real**	Lesen/Schreiben
DVDB	**Real**	Lesen/Schreiben
MAXI	**Real**	Lesen/Schreiben
MINI	**Real**	Lesen/Schreiben
TIE	**Real**	Lesen/Schreiben
FILE	**Word**, Short	Lesen/Schreiben
ELEM	**Word**, Short	Lesen/Schreiben
EN	**Boolean**	Lesen/Schreiben
CT	**Boolean**	Lesen/Schreiben
CL	**Boolean**	Lesen/Schreiben
PVT	**Boolean**	Lesen/Schreiben
DO	**Boolean**	Lesen/Schreiben
SWM	**Boolean**	Lesen/Schreiben
CA	**Boolean**	Lesen/Schreiben
MO	**Boolean**	Lesen/Schreiben
PE,	**Boolean**	Lesen/Schreiben
INI	**Boolean**	Lesen/Schreiben
SPOR	**Boolean**	Lesen/Schreiben
OLL	**Boolean**	Lesen/Schreiben
OLH	**Boolean**	Lesen/Schreiben
EWD	**Boolean**	Lesen/Schreiben
DVNA	**Boolean**	Lesen/Schreiben
DVHA	**Boolean**	Lesen/Schreiben
PVLA	**Boolean**	Lesen/Schreiben
PVHA	**Boolean**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	k.A.
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	k.A.	k.A.
PLC-5-Reihe	3-999	999

Beispiele

Beispiel	Description
PD14:0.SP	Setpoint-Feld von PD 0 Datei 14
PD18:6.EN	Statusaktivierungs-Bit von PD 6 Datei 18

MicroLogix-Meldungsdateien

Bei Meldungsdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax	Datentyp	Zugriff
MG<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
IA	**Word**, Short	Lesen/Schreiben
RBL	**Word**, Short	Lesen/Schreiben
LBN	**Word**, Short	Lesen/Schreiben
RBN	**Word**, Short	Lesen/Schreiben
CHN	**Word**, Short	Lesen/Schreiben
NOD	**Word**, Short	Lesen/Schreiben
MTO	**Word**, Short	Lesen/Schreiben
NB	**Word**, Short	Lesen/Schreiben
TFT	**Word**, Short	Lesen/Schreiben
TFN	**Word**, Short	Lesen/Schreiben
ELE	**Word**, Short	Lesen/Schreiben
SEL	**Word**, Short	Lesen/Schreiben
TO	**Boolean**	Lesen/Schreiben
CO	**Boolean**	Lesen/Schreiben
EN	**Boolean**	Lesen/Schreiben
RN	**Boolean**	Lesen/Schreiben
EW	**Boolean**	Lesen/Schreiben
ER	**Boolean**	Lesen/Schreiben
DN	**Boolean**	Lesen/Schreiben
ST	**Boolean**	Lesen/Schreiben
BK	**Boolean**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	3-255	255
Alle SLC	k.A.	k.A.
PLC5	Meldungsdateien	Meldungsdateien

Beispiele

Beispiel	Description
MG14:0.TO	Timeout-Bit für MSG-Element 0 in Datei 14
MG18:6.CO	Fortsetzungs-Bit für MSG-Element 6 in Datei 18

Meldungsdateien

Bei Meldungsdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
MG<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
ERR	**Short**, Word	Lesen/Schreiben
RLEN	**Short**, Word	Lesen/Schreiben
DLEN	**Short**, Word	Lesen/Schreiben
EN	**Boolean**	Lesen/Schreiben
ST	**Boolean**	Lesen/Schreiben
DN	**Boolean**	Lesen/Schreiben
ER	**Boolean**	Lesen/Schreiben
CO	**Boolean**	Lesen/Schreiben
EW	**Boolean**	Lesen/Schreiben
NR	**Boolean**	Lesen/Schreiben
TO	**Boolean**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	k.A.
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	k.A.	k.A.
PLC-5-Reihe	3-999	999

Beispiele

Beispiel	Description
MG14:0.RLEN	Angefordertes Längenfeld von MG 0 Datei 14
MG18:6.CO	Fortsetzungs-Bit von MG 6 Datei 18

Blocktransferdateien

Bei Blocktransferdateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen einer Dateinummer, eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Syntax für PLC-5

Syntax	Datentyp	Zugriff
BT<Datei>:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
RLEN	**Word**, Short	Lesen/Schreiben
DLEN	**Word**, Short	Lesen/Schreiben
FILE	**Word**, Short	Lesen/Schreiben
ELEM	**Word**, Short	Lesen/Schreiben
RW	**Boolean**	Lesen/Schreiben
ST	**Boolean**	Lesen/Schreiben
DN	**Boolean**	Lesen/Schreiben
ER	**Boolean**	Lesen/Schreiben
CO	**Boolean**	Lesen/Schreiben
EW	**Boolean**	Lesen/Schreiben
NR	**Boolean**	Lesen/Schreiben
TO	**Boolean**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	k.A.
Feste E/A für SLC 500	k.A.	k.A.
Modulare E/A für SLC 500	k.A.	k.A.
PLC-5-Reihe	3-999	1999

Beispiele

Beispiel	Description
BT14:0.RLEN	Angefordertes Längenfeld von BT 0, Datei 14
BT18:6.CO	Fortsetzungs-Bit von BT 6, Datei 18

Funktionsdateien

Wenn Sie Informationen zu den Dateien, die von ENI MicroLogix- und MicroLogix 1100-Gerätemodellen unterstützt werden, erhalten möchten, wählen Sie eine Verknüpfung in der Liste unten aus.

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)
RTC-Datei (Real Time Clock, Echtzeituhr)
CS 0-Datei (Channel 0 Communication Status, Kommunikationsstatus für Kanal 0)
CS1 (Channel 1 Communication Status, Kommunikationsstatus für Kanal 1)
IOS-Datei (I/O Module Status, E/A-Modulstatus)

Weitere Informationen zu Gerätemodellen und ihren unterstützten Dateien finden Sie unter Adressbeschreibungen.

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler)

Bei HSC-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Siehe auch: Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway

Syntax	Datentyp	Zugriff
HSC:<Element>.<Feld>	Hängt von Feld ab.	Hängt von Feld ab.

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Standardtyp	Zugriff
ACC	**DWord**, Long	Schreibgeschützt
HIP	**DWord**, Long	Lesen/Schreiben
LOP	**DWord**, Long	Lesen/Schreiben
OVF	**DWord**, Long	Lesen/Schreiben
UNF	**DWord**, Long	Lesen/Schreiben
PFN	**Word**, Short	Schreibgeschützt
ER	**Word**, Short	Schreibgeschützt
MOD	**Word**, Short	Schreibgeschützt
OMB	**Word**, Short	Schreibgeschützt
HPO	**Word**, Short	Lesen/Schreiben
LPO	**Word**, Short	Lesen/Schreiben
UIX	**Boolean**	Schreibgeschützt
UIP	**Boolean**	Schreibgeschützt
AS	**Boolean**	Schreibgeschützt
ED	**Boolean**	Schreibgeschützt
SP	**Boolean**	Schreibgeschützt
LPR	**Boolean**	Schreibgeschützt
HPR	**Boolean**	Schreibgeschützt
DIR	**Boolean**	Schreibgeschützt
CD	**Boolean**	Schreibgeschützt
CU	**Boolean**	Schreibgeschützt
UIE	**Boolean**	Lesen/Schreiben
UIL	**Boolean**	Lesen/Schreiben
FE	**Boolean**	Lesen/Schreiben
CE	**Boolean**	Lesen/Schreiben
LPM	**Boolean**	Lesen/Schreiben
HPM	**Boolean**	Lesen/Schreiben
UFM	**Boolean**	Lesen/Schreiben
OFM	**Boolean**	Lesen/Schreiben
LPI	**Boolean**	Lesen/Schreiben
HPI	**Boolean**	Lesen/Schreiben
UFI	**Boolean**	Lesen/Schreiben
OFI	**Boolean**	Lesen/Schreiben
UF	**Boolean**	Lesen/Schreiben
OF	**Boolean**	Lesen/Schreiben
MD	**Boolean**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
HSC:0.OMB	Ausgabemaskeneinstellung für Hochgeschwindigkeitszähler 0.
HSC:1.ED	Hinweis auf festgestellten Fehler für Hochgeschwindigkeitszähler 1.

RTC-Datei (Real Time Clock, Echtzeituhr)

Bei RTC-Dateien handelt es sich um einen strukturierten Typ, auf dessen Daten durch Festlegen eines Elements und eines Felds zugegriffen wird. Die Standard-Datentypen werden **fett** dargestellt.

Siehe auch: Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway

Syntax	Datentyp	Zugriff
RTC:<Element>.<Feld>	Hängt von Feld ab	Hängt von Feld ab

Die folgenden Felder sind für jedes Element zulässig. Weitere Informationen zur Bedeutung der einzelnen Felder finden Sie in der SPS-Dokumentation.

Elementfeld	Datentyp	Zugriff
YR	**Word**, Short	Lesen/Schreiben
MON	**Word**, Short	Lesen/Schreiben
DAY	**Word**, Short	Lesen/Schreiben
HR	**Word**, Short	Lesen/Schreiben
MIN	**Word**, Short	Lesen/Schreiben
SEC	**Word**, Short	Lesen/Schreiben
DOW	**Word**, Short	Lesen/Schreiben
DS	**Boolean**	Schreibgeschützt
BL	**Boolean**	Schreibgeschützt
_SET (for Blockschreibvorgänge)	**Boolean**	Lesen/Schreiben

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
RTC:0.YR	Jahreseinstellung für Echtzeituhr 0.
RTC:0.BL	Hinweis auf niedrigen Batteriestand für Echtzeituhr 0.

Kommunikationsstatusdatei für Kanal 0 (CS0, Channel 0 Communication Status File)

Geben Sie für den Zugriff auf die Kommunikationsstatusdatei für Kanal 0 ein Wort (und optional ein Bit im Wort) an. Die Standard-Datentypen werden **fett** dargestellt.

Siehe auch: Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway

Syntax	Datentyp	Zugriff
CS0:<Wort>	Short, **Word**, BCD, DWord, Long, LBCD	Hängt von <Wort> und <Bit> ab
CS0:<Wort>/<Bit>	**Boolean**	Hängt von <Wort> und <Bit> ab
CS0/Bit	**Boolean**	Hängt von <Wort> und <Bit> ab

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Beschreibung
CS0:0	Wort 0.
CS0:4/2	Bit 2, Wort 4 = MCP.

Weitere Informationen zu Bedeutungen von CS0-Wörtern/Bits finden Sie in der Rockwell-Dokumentation.

Kommunikationsstatusdatei für Kanal 1 (CS1, Channel 1 Communication Status File)

Geben Sie für den Zugriff auf die Kommunikationsstatusdatei für Kanal 1 ein Wort (und optional ein Bit im Wort) an. Die Standard-Datentypen werden **fett** dargestellt.

Siehe auch: Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway

Syntax	Datentyp	Zugriff
CS1:<Wort>	Short, **Word**, BCD, DWord, Long, LBCD	Hängt von <Wort> und <Bit> ab
CS1:<Wort>/<Bit>	**Boolean**	Hängt von <Wort> und <Bit> ab
CS1/Bit	**Boolean**	Hängt von <Wort> und <Bit> ab

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
CS1:0	Wort 0.
CS1:4/2	Bit 2, Wort 4 = MCP.

Weitere Informationen zu Bedeutungen von CS1-Wörtern/Bit finden Sie in der Rockwell-Dokumentation.

IOS-Datei (I/O Module Status, E/A-Modulstatus)

Geben Sie für den Zugriff auf eine E/A-Modulstatus-Datei ein Wort und optional ein Bit an. Die Standard-Datentypen werden **fett** dargestellt.

Siehe auch: Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway

Syntax	Datentyp	Zugriff
IOS:<Wort>	Short, **Word**, BCD, DWord, Long, LBCD	Hängt von <Wort> und <Bit> ab
IOS:<Wort>/<Bit>	**Boolean**	Hängt von <Wort> und <Bit> ab
IOS/Bit	**Boolean**	Hängt von <Wort> und <Bit> ab

Bereiche

SPS-Modell	Dateinummer	Max. Element
MicroLogix	k.A.	254
Alle SLC	k.A.	k.A.
PLC5	k.A.	k.A.

Beispiele

Beispiel	Description
IOS:0	Wort 0.
IOS:4/2	Bit 2, Wort 4.

Hinweis: Eine Auflistung von 1769 Erweiterungs-E/A-Statuscodes finden Sie in der Bedienungsanleitung.

Automatische Tag-Datenbankgenerierung

ControlLogix-Ethernet-Treiber von Allen-Bradley kann so konfiguriert werden, dass automatisch eine Liste von Server-Tags auf dem Server generiert wird, die gerätespezifischen Daten entsprechen. Die automatisch generierten Tags basieren auf den im Logix-Gerät definierten Logix-Tags und können vom OPC-Client durchsucht werden. Logix-Tags können unteilbar oder strukturiert sein. Struktur- und Array-Tags können recht schnell die Anzahl importierter Tags (und somit die Anzahl der auf dem Server verfügbaren Tags) erhöhen.

Hinweis: ENI/DH+-, ControlNet-Gateway- und MicroLogix-Modelle unterstützen die automatische Tag-Datenbankgenerierung nicht, sondern nur die ENI ControlLogix-, CompactLogix- und FlexLogix-Modelle.

Unteilbares Tag -> **1:1** -> Server-Tag
Struktur-Tag -> **1:n** -> Server-Tags
Array-Tag -> **1:n**> -> Server-Tags

Weitere Informationen zu den Einstellungen für die Datenbankerstellung finden Sie in der Server-Hilfedatei.

Hinweis: Die Eigenschaft "Externer Zugriff" für Controller-Tags in der RSLogix5000-Programmierungsumgebung muss auf "Schreibgeschützt" oder Lesen/Schreiben" festgelegt werden, damit die Tags gelesen werden können. Für automatisch generierte Tags kann die Eigenschaft "Externer Zugriff" standardmäßig auf "Keine" festgelegt werden. Konfigurieren Sie "Externer Zugriff" bei Bedarf in den Add-On Instruction-Parametern der RSLogix neu, um Controller-Tags zu lesen. *Lesen Sie die Dokumentation des Herstellers.*

Tag-Hierarchie

Die durch die automatische Tag-Generierung erstellten Server-Tags können einer der beiden Hierarchien folgen: Erweitert oder Komprimiert. Um diese Funktionalität zu verwenden, aktivieren Sie in den Geräteeigenschaften Untergruppen zulassen.

Erweiterter Modus

Die Option "Erweitert" besagt, dass die durch automatische Tag-Generierung erstellten Server-Tags einer mit der Tag-Hierarchie in RSLogix 5000 konsistenten Gruppen-/Tag-Hierarchie folgen. Gruppen werden wie bei

der Komprimierung für jedes dem Zeitraum vorangehende Segment erstellt, werden jedoch auch in logischen Gruppierungen erstellt. Zu den erstellten Gruppen gehören:

- Globaler Umfang (im Controller)
- Programmumfang
- Strukturen und Substrukturen
- Arrays

Hinweis: Gruppen werden nicht für .bit-Adressen erstellt.

Die Gruppen auf Stammebene (oder Untergruppenebene der in Elternteilgruppe angegebenen Gruppe), lauten "Prgm _<program name>" und "Global". Jedes Programm im Controller verfügt über seine eigene Gruppe "Prgm_<Programmname>". Der Treiber erkennt dies als die erste Gruppenebene an.

Grundlegende globale Tags (oder Nicht-Struktur-, Nicht-Array-Tags) werden unter der globalen Gruppe eingefügt, grundlegende Programm-Tags unter ihrer jeweiligen Programmgruppe. Jedes Struktur- und Array-Tag wird in seiner eigenen Untergruppe der übergeordneten Gruppe bereitgestellt. Durch Organisieren der Daten auf diese Weise ahmt die Tag-Ansicht des Servers RSLogix5000 nach.

Im Namen der Struktur-/Array-Untergruppe ist auch eine Beschreibung der Struktur bzw. des Arrays enthalten. Beispielsweise würde ein im Controller definiertes Array mit tag1[1,6] einen Untergruppennamen "tag1_x_y" aufweisen; x bedeutet, dass Dimension 1 vorhanden ist, und y bedeutet, dass Dimension 2 vorhanden ist. Bei den Tags innerhalb einer Array-Untergruppe handelt es sich um alle Elemente dieses Arrays (sofern nicht explizit begrenzt). Bei den Tags innerhalb einer Strukturuntergruppe handelt es sich um die Strukturmitglieder selbst. Wenn eine Struktur ein Array enthält, wird ebenso eine Array-Untergruppe der Strukturgruppe erstellt.

Bei einem komplexen Projekt kann die Tag-Hierarchie mehrere Gruppenebenen erfordern. Die maximale Anzahl der durch die automatische Tag-Generierung erstellten Gruppenebenen liegt bei 7. Dabei ist die unter "Generierte Tags zu folgender Gruppe hinzufügen" angegebene Gruppe nicht enthalten. Wenn mehr als sieben Ebenen erforderlich sind, werden die Tags in der siebten Gruppe platziert (was eine Plateau-Hierarchie zur Folge hat).

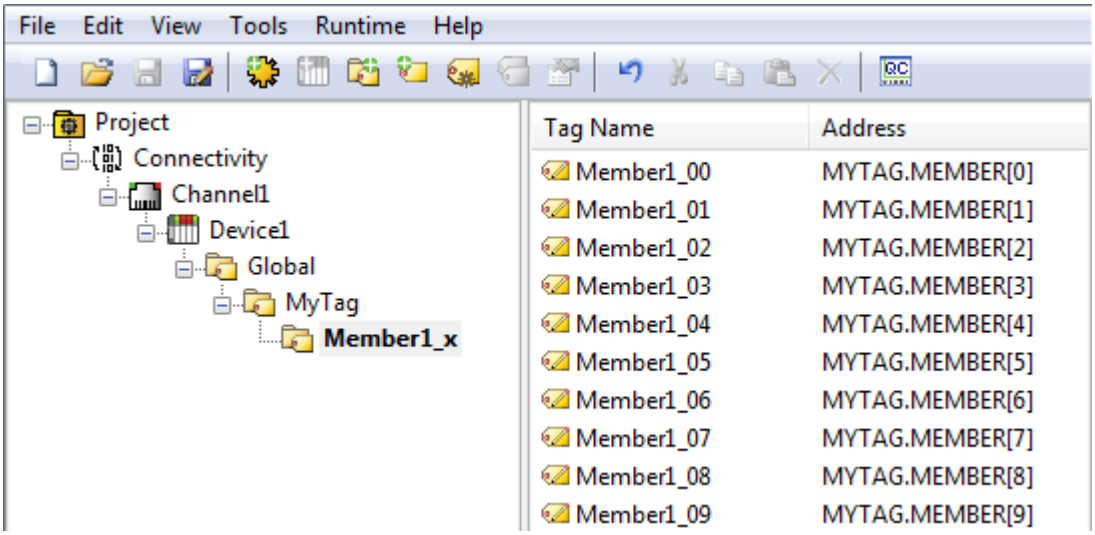
Array-Tags
Eine Gruppe wird für jedes Array erstellt, das die Elemente des Arrays enthält. Gruppennamen haben die folgende Notation: *<Array-Name>_x_y_z* Dabei gilt:

x_y_z = 3-dimensionales Array
x_y = 2-dimensionales Array
x = 1-dimensionales Array

Array-Tags haben die folgende Notation: *<Tag-Element>_XXXXX_YYYYY_ZZZZZ*. Beispiel: Das Element tag1[12,2,987] würde den Tag-Namen "tag1_12_2_987" aufweisen.

Einfaches Beispiel

Name	Value	Force Mask	Style	Data Type
MyTag	{...}	{...}		MyDataType
MyTag.Member1	{...}	{...}	Decimal	DINT[10]
MyTag.Member1[0]	0		Decimal	DINT
MyTag.Member1[1]	0		Decimal	DINT
MyTag.Member1[2]	0		Decimal	DINT
MyTag.Member1[3]	0		Decimal	DINT



Komplexes Beispiel
Ein Logix-Tag wird mit der Adresse "Local:1:O.Slot[9].Data" definiert. Dies würde in den Gruppen "Global" - "Local_1_O" - "Slot_x" - "Slot_09" dargestellt. Innerhalb der letzten Gruppe würde das Tag "Data" sein.

Die statische Referenz zu "Data" wäre "Channel1.Device1.Global.Local_1_O.Slot_x.Slot_09.Data". Die dynamische Referenz zu "Data" wäre "Channel1.Device1.Local:1:O.Slot[9].Data".

Komprimierter Modus

Im komprimierten Modus folgen die durch automatische Tag-Generierung erstellten Server-Tags einer mit der Adresse des Tags konsistenten Gruppen-/Tag-Hierarchie. Gruppen werden für jedes dem Zeitraum vorangehende Segment erstellt. Zu den erstellten Gruppen gehören:

- Programmumfang
- Strukturen und Substrukturen

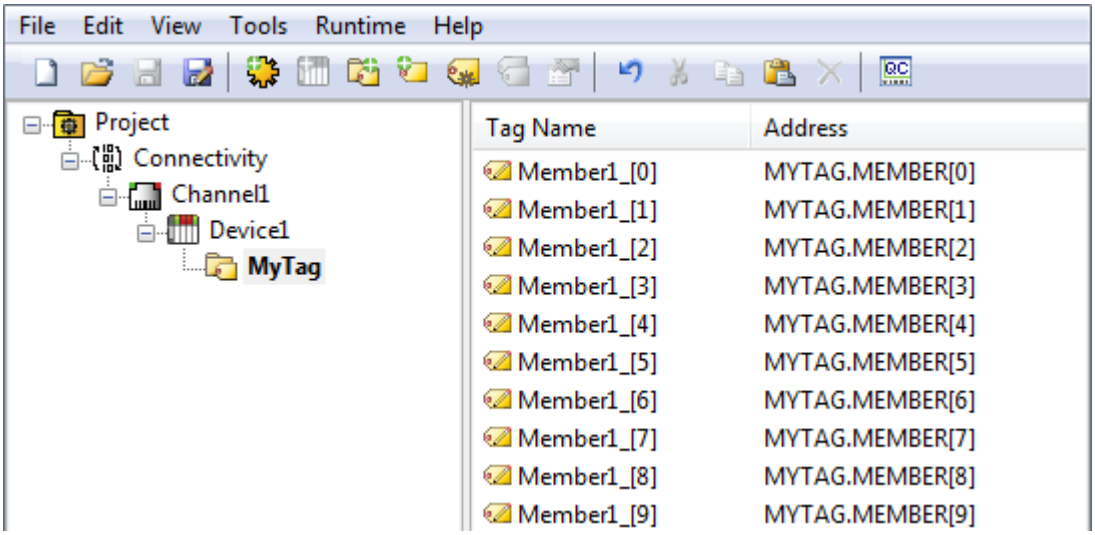
Hinweis: Gruppen werden nicht für Arrays oder .bit-Adressen erstellt.

Bei einem komplexen Projekt kann einfach festgestellt werden, wie die Tag-Hierarchie mehrere Gruppenebenen benötigt. Die maximale Anzahl der durch die automatische Tag-Generierung erstellten Gruppenebenen liegt bei 7. Dabei ist die unter "Generierte Tags zu folgender Gruppe hinzufügen" angegebene Gruppe nicht enthalten. Wenn mehr als sieben Ebenen erforderlich sind, werden die Tags in der siebten Gruppe platziert (was eine Plateau-Hierarchie zur Folge hat).

Hinweis: Vorangestellte Namen von Tag- oder Strukturmitgliedern, die mit einem Unterstrich enden, werden in "U_" konvertiert. Dies ist erforderlich, weil der Server keine vorangestellten Unterstriche unterstützt. Weitere Informationen dazu finden Sie unter Namenskonvertierungen von Controller in Server.

Einfaches Beispiel

Name	Value	Force Mask	Style	Data Type
MyTag	{...}	{...}		MyDataType
MyTag.Member1	{...}	{...}	Decimal	DINT[10]
MyTag.Member1[0]	0		Decimal	DINT
MyTag.Member1[1]	0		Decimal	DINT
MyTag.Member1[2]	0		Decimal	DINT
MyTag.Member1[3]	0		Decimal	DINT



Komplexes Beispiel
Das Logix-Tag wird mit der Adresse "Local:1:O.Slot[9].Data" definiert. Dies würde in den Gruppen "Local:1:O" -> "Slot[9]" dargestellt werden. Innerhalb der letzten Gruppe würde das Tag "Data" sein.

Die statische Referenz zu "Data" wäre "Channel1.Device1.Local:1:O.Slot[9].Data". Die dynamische Referenz wäre "Channel1.Device1.Local:1:O.Slot[9].Data".

Hinweis: E/A-Modul-Tags können im Offline-Modus nicht direkt importiert werden. Da Alias-Namen importiert werden können, wird empfohlen, dass sie für die entsprechenden E/A-Modul-Tags in RSLogix5000 erstellt werden.

Namenskonvertierungen von Controller in Server

Vorangestellte Unterstriche

Vorangestellte Unterstriche "_" in Tag- oder Programmnamen werden durch "U_" ersetzt. Dies ist erforderlich, da der Server keine Tag- oder Gruppennamen akzeptiert, die mit einem Unterstrich beginnen.

Lange Namen (OPC-Server Version 4.64 und niedriger)

Bei älteren OPC-Serverversionen ControlLogix-Ethernet-Treiber von Allen-Bradley gab es eine Begrenzung auf 31 Zeichen in Gruppen- und Tag-Namen. Wenn der Name eines Controller-Programms oder Tags 31 Zeichen überschritten hatte, musste er abgeschnitten werden. OPC-Server Version 4.70 und höher weist eine Begrenzung von 256 Zeichen auf, sodass die Regeln nicht gelten. Namen werden wie folgt abgeschnitten:

Nicht-Array

1. Bestimmen Sie eine 5-stellige eindeutige ID für dieses Tag.
2. Angegebener Tag-Name: ThisIsALongTagNameAndProbablyExceeds31
3. Schneiden Sie das Tag bei 31 ab: ThisIsALongTagNameAndProbablyEx
4. Für die eindeutige ID wird Platz gemacht: ThisIsALongTagNameAndProba#####
5. Fügen Sie diese ID ein: ThisIsALongTagNameAndProba00000

Array

1. Bestimmen Sie eine 5-stellige eindeutige ID für dieses Array.
2. Angegebener Array-Tag-Name: ThisIsALongTagNameAndProbablyExceeds31_23_45_8
3. Schneiden Sie das Tag bei 31 unter Beibehaltung der Elementwerte ab: ThisIsALongTagNameAndPr_23_45_8

4. Für die eindeutige ID wird Platz gemacht: ThisIsALongTagName#####_23_45_8
5. Fügen Sie diese ID ein: ThisIsALongTagName00001_23_45_8

Lange Programmnamen werden auf dieselbe Weise wie lange Nicht-Array-Tag-Namen abgeschnitten. Für jeden abgeschnittenen Tag- oder Programmnamen wird die eindeutige ID inkrementiert. Array-Tag-Namen (Elemente) eines abgeschnittenen Array-Namens weisen dieselbe eindeutige ID auf. Dies ermöglicht 100000 eindeutige Tag-/Programmnamen.

Hinweis: Wenn Namenlänge begrenzen aktiviert ist, gelten die Regeln selbst dann, wenn Namen mit 256 Zeichen unterstützt werden. Weitere Informationen dazu finden Sie unter Logix-Datenbankeinstellungen.

Automatische Tag-Datenbankgenerierung wird vorbereitet

Informationen zur Verwendung der automatischen Tag-Datenbankgenerierung finden Sie in den Anweisungen unten.

Online

Es wird empfohlen, dass sämtliche Kommunikation mit der betreffenden Logix-CPU während des Datenbankerstellungsprozesses eingestellt wird.

In RSLogix5000

Setzen Sie das Projekt OFFLINE.

Im Server:

1. Prüfen Sie die Geräteeigenschaften des Geräts, für das Tags generiert werden.
2. Wählen Sie **Logix-Datenbankeinstellungen** aus, und legen Sie **Von Gerät erstellen** als **Datenbank-Importmethode** fest.
3. Nehmen Sie in **Logix-Datenbankoptionen** ggf. Änderungen vor, und klicken Sie auf **OK**.
4. Nehmen Sie in **Logix-Datenbankfilterung** ggf. Änderungen vor, und klicken Sie auf **OK**.
5. Wählen Sie **Tag-Generierung** aus, und klicken Sie unter Erstellen auf den blauen Link für Tags erstellen.

Hinweis: Legen Sie unter **Logix-Optionen** für **Protokollmodus** die Option **Symbolisch** und für **Standard-Datentyp** die Option **Standard** fest, damit die Tags mit den im Controller verwendeten Datentypen importiert werden.

Offline

ControlLogix-Ethernet-Treiber von Allen-Bradley verwendet eine von RSLogix5000 aus generierte Datei, eine sog. L5K/L5X-Import-/Exportdatei zum Generieren der Tag-Datenbank.

In RSLogix5000

1. Öffnen Sie das Projekt, das die Tags enthält, die an den OPC-Server übertragen werden sollen.
2. Klicken Sie auf **Datei | Speichern als**.
3. Wählen Sie die Option **L5K/L5X Import/Export File** aus, und geben Sie anschließend einen Namen an. Daraufhin exportiert RSLogix den Inhalt des Projekts in diese L5K/L5X-Datei.

Im OPC-Server

1. Öffnen Sie die Geräteeigenschaften des Geräts, für das Tags generiert werden.
2. Wählen Sie **Logix-Datenbankeinstellungen** aus, und wählen Sie **Von Importdatei erstellen** als **Datenbank-Importmethode** aus.
3. Geben Sie die zuvor erstellte Datei ein, bzw. suchen Sie nach ihrem Speicherort.
4. Nehmen Sie in **Logix-Datenbankoptionen** ggf. Änderungen vor, und klicken Sie auf **OK**.

5. Nehmen Sie in **Logix-Datenbankfilterung** ggf. Änderungen vor, und klicken Sie auf **OK**.
6. Wählen Sie **Tag-Generierung** aus, und klicken Sie unter Erstellen auf den blauen Link für **Tags erstellen**.

Hinweis: Importierte vordefinierte Tag-Typen basieren auf der aktuellen, vom Treiber unterstützten Version. Weitere Informationen dazu finden Sie in den Firmware-Versionen.

Fehlercodes

In den folgenden Abschnitten sind Fehlercodes definiert, die im Ereignisprotokoll des Servers möglicherweise erfasst wurden. Weitere Informationen zu einem bestimmten Fehlercode erhalten Sie, wenn Sie eine Verknüpfung in der Liste unten auswählen.

Kapselungsfehlercodes
CIP-Fehlercodes

Kapselungsfehlercodes

Die folgenden Fehlercodes liegen in hexadezimaler Schreibweise vor.

Fehlercode	Description
0001	Befehl nicht verarbeitet.
0002	Speicher nicht für Befehl verfügbar.
0003	Schlecht gebildete oder unvollständige Daten.
0064	Ungültige Sitzungs-ID.
0065	Ungültige Länge des Headers.
0069	Angeforderte Protokollversion nicht unterstützt.
0070	Ungültige Ziel-ID.

CIP-Fehlercodes

Die folgenden Fehlercodes liegen in hexadezimaler Schreibweise vor.

Fehlercode	Protokollcode	Description
0001	0x01	Verbindungsfehler.*
0002	0x02	Nicht genügend Ressourcen.
0003	0x03	Wert ungültig.
0004	0x04	IOI konnte nicht entziffert werden oder Tag ist nicht vorhanden.
0005	0x05	Unbekanntes Ziel.
0006	0x06	Angeforderte Daten würden nicht in Antwortpaket passen.
0007	0x07	Unterbrochene Verbindung.
0008	0x08	Nicht unterstützter Dienst.
0009	0x09	Fehler in Datensegment oder ungültiger Attributwert.
000A	0x0A	Attributlistenfehler.
000B	0x0B	Status bereits vorhanden.
000C	0x0C	Objektmodellkonflikt.
000D	0x0D	Objekt bereits vorhanden.
000E	0x0E	Attribut nicht festlegbar.
000F	0x0F	Berechtigung verweigert.
0010	0x10	Gerätestatuskonflikt.
0011	0x11	Antwort passt nicht.
0012	0x12	Fragment einfach.
0013	0x13	Nicht ausreichende Befehlsdaten/-parameter zur Ausführung des Diensts angegeben.
0014	0x14	Attribut nicht unterstützt.
0015	0x15	Zu viele Daten angegeben.
001A	0x1A	Bridge-Anforderung zu groß.
001B	0x1B	Bridge-Antwort zu groß.
001C	0x1C	Attributliste nicht ausreichend.

Fehlercode	Protokollcode	Description
001D	0x1D	Ungültige Attributliste.
001E	0x1E	Fehler bei eingebettetem Dienst.
001F	0x1F	Fehler während Verbindung.**
0022	0x22	Ungültige Antwort erhalten.
0025	0x25	Schlüsselsegmentfehler.
0026	0x26	Die Anzahl der angegebenen IOI-Wörter entspricht nicht der IOI-Wort-anzahl.
0027	0x27	Unerwartetes Attribut in Liste.

- *Siehe auch: 0x0001 Erweiterte Fehlercodes
- **Siehe auch: 0x001F Erweiterte Fehlercodes

Logix5000-spezifische (1756-L1) Fehlercodes

Die folgenden Fehlercodes liegen in hexadezimaler Schreibweise vor.

Fehlercode	Beschreibung
00FF	Allgemeiner Fehler.*

- *Siehe auch: 0x00FF Erweiterte Fehlercodes
- **Siehe auch:** Nicht aufgelistete Fehlercodes finden Sie in der Dokumentation von Rockwell.

0x0001 Erweiterte Fehlercodes

Die folgenden Fehlercodes liegen in hexadezimaler Schreibweise vor.

Fehlercode	Beschreibung
0100	Verbindung wird verwendet.
0103	Transport wird nicht unterstützt.
0106	Besitzerkonflikt.
0107	Verbindung nicht gefunden.
0108	Ungültiger Verbindungstyp.
0109	Ungültige Verbindungsgröße.
0110	Modul nicht konfiguriert.
0111	EPR nicht unterstützt.
0114	Falsches Modul.
0115	Falscher Gerätetyp.
0116	Falsche Revision.
0118	Ungültiges Konfigurationsformat.
011A	Keine Verbindungen für Anwendung.
0203	Verbindungs-Timeout
0204	Nicht verbundenes Meldungs-Timeout.
0205	Parameterfehler für nicht verbundenes Senden.
0206	Nachricht zu groß.
0301	Kein Pufferspeicher.
0302	Bandbreite nicht verfügbar.
0303	Keine Screener verfügbar.
0305	Signaturübereinstimmung.
0311	Port nicht verfügbar.
0312	Verknüpfungsadresse nicht verfügbar.

Fehlercode	Beschreibung
0315	Ungültiger Segmenttyp.
0317	Verbindung nicht geplant.
0318	Eigene Verknüpfungsadresse ist ungültig.

Hinweis: Nicht aufgelistete Fehlercodes finden Sie in der Dokumentation von Rockwell.

0x001F Erweiterte Fehlercodes

Die folgenden Fehlercodes liegen in hexadezimaler Schreibweise vor.

Fehlercode	Beschreibung
0203	Timeout der Verbindung.

Hinweis: Nicht aufgelistete Fehlercodes finden Sie in der Dokumentation von Rockwell.

0x00FF Erweiterte Fehlercodes

Die folgenden Fehlercodes liegen in hexadezimaler Schreibweise vor.

Fehlercode	Beschreibung
2104	Adresse außerhalb des zulässigen Bereichs.
2105	Versuch, über das Ende des Datenobjekts hinaus zuzugreifen.
2106	Daten werden verwendet.
2107	Datentyp ist ungültig oder wird nicht unterstützt.

Hinweis: Nicht aufgelistete Fehlercodes finden Sie in der Dokumentation von Rockwell.

Ereignisprotokollmeldungen

Die folgenden Informationen betreffen Meldungen, die im Fensterbereich Ereignisprotokoll in der Hauptbenutzeroberfläche angezeigt werden. Informationen zum Filtern und Sortieren der Detailansicht Ereignisprotokoll finden Sie in der Serverhilfe. In der Serverhilfe sind viele allgemeine Meldungen enthalten, die also auch gesucht werden sollten. Im Allgemeinen werden die Art der Meldung (Information, Warnung) sowie Fehlerbehebungsinformationen bereitgestellt (sofern möglich).

Die folgenden Fehler sind beim Hochladen des Controller-Projekts vom Gerät aufgetreten. Das symbolische Protokoll wird stattdessen verwendet.

Fehlertyp:
Fehler

Ungültiges oder fehlerhaftes Controller-Projekt wurde beim Synchronisieren festgestellt. Die Synchronisierung wird in Kürze erneut versucht.

Fehlertyp:
Fehler

Mögliche Ursache:
Ein ungültiges oder fehlerhaftes Controller-Projekt wurde beim Synchronisieren festgestellt.

Mögliche Lösung:
Keine Aktion erforderlich. Der Treiber versucht, nach 30 Sekunden eine erneute Synchronisierung durchzuführen.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt synchronisiert werden.

Während der Synchronisierung wurde ein Herunterladen des Projekts festgestellt. Die Synchronisierung wird in Kürze erneut versucht.

Fehlertyp:
Fehler

Mögliche Ursache:
Ein Herunterladen des Projekts wurde versucht, während das Gerät mit dem Controller-Projekt synchronisiert wurde.

Mögliche Lösung:
Keine Aktion erforderlich. Der Treiber versucht, nach 30 Sekunden eine erneute Synchronisierung durchzuführen.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt synchronisiert werden.

Datenbankfehler. Datentyp für Referenz-Tag ist unbekannt. Datentyp für Alias-Tag wird auf den Standardwert gesetzt. | Referenz-Tag = '<Tag>', Alias-Tag = '<Tag>', Standard-Datentyp = '<Typ>'.

Fehlertyp:
Fehler

Mögliche Ursache:
Der Datentyp des Tags 'Alias For', der in der Alias-Tag-Deklaration referenziert ist, wurde in der Tag-Import-Datei nicht gefunden. Dieser Datentyp ist erforderlich, um das Alias-Tag ordnungsgemäß zu generieren.

Mögliche Lösung:
Das Alias-Tag übernimmt den als Standardtyp angegebenen Datentyp.

Hinweis:
In RSLogix5000 gibt es eine Spalte 'Alias For' in der Tag-Ansicht unter der Registerkarte 'Edit Tags', in der die Referenz zu Tag, Struktur-Tag-Mitglied oder Bit, die das Alias-Tag darstellt, eingegeben wird.

Siehe auch:
Logix-Optionen

Datenbankfehler. Datentyp für Mitglied in Tag-Importdatei wurde nicht gefunden. Datentyp wird auf Standardwert gesetzt. | Datentyp für Mitglied = '<Typ>', UDT = '<Typ>', Standard-Datentyp '<Typ>'.

Fehlertyp:
Fehler

Mögliche Ursache:
Die Definition des Datentyps für ein Mitglied im benutzerdefinierten Typ konnte in der Tag-Importdatei nicht gefunden werden. Das Mitglied erhält den Standardtyp, der in den Geräteeigenschaften angegeben ist.

Mögliche Lösung:
Überprüfen bzw. korrigieren Sie die Definition der benutzerdefinierten Datentypen für die angegebenen Tags, und führen Sie den Import erneut durch.

Siehe auch:
Logix-Optionen

Datenbankfehler. Datentyp wurde in Tag-Importdatei nicht gefunden. Tag nicht hinzugefügt. | Datentyp = '<Typ>', Tag-Name = '<Tag>'.

Fehlertyp:
Fehler

Mögliche Ursache:
Die Definition des Datentyps für das angegebene Tag konnte in der Tag-Importdatei nicht gefunden werden. Tag wurde nicht zur Datenbank hinzugefügt.

Mögliche Lösung:
Überprüfen bzw. korrigieren Sie die Definition der Datentypen für die angegebenen Tags, und führen Sie den Import erneut durch.

Datenbankfehler. Fehler beim Verarbeiten des Alias-Tags. Tag wurde nicht hinzugefügt. | Alias-Tag = '<Tag>'.

Fehlertyp:
Fehler

Mögliche Ursache:
Ein interner Fehler ist beim Verarbeiten des Alias-Tags aufgetreten. Alias-Tag konnte nicht generiert werden.

Mögliche Lösung:
Überprüfen bzw. korrigieren Sie die Definition der Datentypen für die angegebenen Tags, und führen Sie den Import erneut durch.

Datenbankfehler. Kapselungsfehler beim Registrieren der Sitzungsanforderung. | Kapselungsfehler = <Code>.

Fehlertyp:
Fehler

Mögliche Ursache:
Während einer Anforderung wurde vom Gerät ein Fehler im Kapselungsteil des Ethernet/IP-Pakets zurückgegeben. Alle Lese- und Schreibvorgänge innerhalb der Anforderung sind fehlgeschlagen.

Mögliche Lösung:
Wiederherstellung des Treibers nach diesem Fehler wird versucht. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support. Dies schließt den Fehler 0x02 aus, der sich auf das Gerät und nicht auf den Treiber bezieht.

Siehe auch:
Kapselungsfehlercodes

Datenbankfehler. Framing-Fehler beim Registrieren der Sitzungsanforderung.

Fehlertyp:
Fehler

Datenbankfehler. Kapselungsfehler beim Weiterleiten der Anforderung zum Öffnen. | Kapselungsfehler = <Code>.

Fehlertyp:
Fehler

Datenbankfehler. Framing-Fehler beim Weiterleiten der offenen Anforderung.

Fehlertyp:
Fehler

Datenbankfehler. Fehler beim Weiterleiten der offenen Anforderung. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Fehler

Datenbankfehler. Kapselungsfehler beim Hochladen der Projektinformationen. | Kapselungsfehler = <Code>.

Fehlertyp:
Fehler

Mögliche Ursache:
Während des Hochladens des Controller-Projekts wurde vom Gerät ein Fehler im Kapselungsteil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:
Die Lösung hängt vom zurückgegebenen Fehlercode ab. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Siehe auch:
Kapselungsfehlercodes

Datenbankfehler. Fehler beim Hochladen der Projektinformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Fehler

Mögliche Ursache:
Während des Hochladens des Controller-Projekts wurde vom Gerät ein Fehler im CIP-Teil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:
Die Lösung hängt vom zurückgegebenen Fehlercode ab. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Siehe auch:
CIP-Fehlercodes

Datenbankfehler. Framing-Fehler beim Hochladen der Projektinformationen.

Fehlertyp:
Fehler

Mögliche Ursache:

1. Die Pakete sind falsch ausgerichtet (aufgrund der Verbindung/Trennung zwischen dem PC und dem Gerät).
2. Fehlerhafte Verbindungskabel zwischen den Geräten verursachen Störungen.

Mögliche Lösung:

1. Verwenden Sie für das Gerät ein Netzwerk mit weniger Datenverkehr.
2. Erhöhen Sie den Wert für Anforderungs-Timeout und/oder -versuche.
3. Starten Sie den Server neu, und führen Sie den Vorgang erneut durch.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Datenbankfehler. Interner Fehler aufgetreten.

Fehlertyp:
Fehler

Datenbankfehler. Kapselungsfehler beim Hochladen der Programminformationen. | Programmname = '<Name>', Kapselungsfehler = <Code>.

Fehlertyp:
Fehler

Mögliche Ursache:
Während des Hochladens des Controller-Projekts wurde vom Gerät ein Fehler im Kapselungsteil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:

Die Lösung hängt vom zurückgegebenen Fehlercode ab. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Siehe auch:
Kapselungsfehlercodes

Datenbankfehler. Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Fehler

Mögliche Ursache:
Während des Hochladens des Controller-Projekts wurde vom Gerät ein Fehler im CIP-Teil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:
Die Lösung hängt vom zurückgegebenen Fehlercode ab. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Siehe auch:
CIP-Fehlercodes

Datenbankfehler. Framing-Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>'.

Fehlertyp:
Fehler

Mögliche Ursache:

1. Die Pakete sind falsch ausgerichtet (aufgrund der Verbindung/Trennung zwischen dem PC und dem Gerät).
2. Fehlerhafte Verbindungskabel zwischen den Geräten verursachen Störungen.

Mögliche Lösung:

1. Verwenden Sie für das Gerät ein Netzwerk mit weniger Datenverkehr.
2. Erhöhen Sie den Wert für Anforderungs-Timeout und/oder -versuche.
3. Starten Sie den Server neu, und führen Sie den Vorgang erneut durch.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Datenbankfehler. CIP-Datentyp für Tag konnte nicht aufgelöst werden. Standardtyp wird verwendet. | CIP-Datentyp = <Typ>, Tag-Name = '<Tag>', Standard-Datentyp = '<Typ>'.

Fehlertyp:
Fehler

Mögliche Ursache:

1. Der CIP-Datentyp in der Importdatei ist unbekannt.
2. Die Importdatei enthält möglicherweise einen Fehler.

Mögliche Lösung:

Beheben Sie mögliche Fehler in RSLogix, und führen Sie anschließend den Tag-Export erneut durch, um eine neue Tag-Importdatei zu erstellen.

Siehe auch:

Automatische Tag-Datenbankgenerierung wird vorbereitet

Kapselungsfehler beim Hochladen der Projektinformationen. | Kapselungsfehler = <Code>.

Fehlertyp:

Fehler

Mögliche Ursache:

Während des Hochladens des Controller-Projekts wurde vom Gerät ein Fehler im Kapselungsteil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:

Die Lösung hängt vom zurückgegebenen Fehlercode ab. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support.

Hinweis:

Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Siehe auch:

Kapselungsfehlercodes

Fehler beim Hochladen der Projektinformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:

Fehler

Mögliche Ursache:

Während des Hochladens des Controller-Projekts wurde vom Gerät ein Fehler im CIP-Teil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:

Die Lösung hängt vom zurückgegebenen Fehlercode ab. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support.

Hinweis:

Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Siehe auch:

CIP-Fehlercodes

Framing-Fehler beim Hochladen der Projektinformationen.

Fehlertyp:

Fehler

Mögliche Ursache:

1. Die Pakete sind falsch ausgerichtet (aufgrund der Verbindung/Trennung zwischen dem PC und dem Gerät).
2. Fehlerhafte Verbindungskabel zwischen den Geräten verursachen Störungen.

Mögliche Lösung:

1. Verwenden Sie für das Gerät ein Netzwerk mit weniger Datenverkehr.
2. Erhöhen Sie den Wert für Anforderungs-Timeout und/oder -versuche.
3. Starten Sie den Server neu, und führen Sie den Vorgang erneut durch.

Hinweis:
Für die Modi mit logischer Adressierung muss das Projekt hochgeladen werden.

Kapselungsfehler beim Hochladen der Programminformationen. | Programmname = '<Name>', Kapselungsfehler = <Code>.

Fehlertyp:
Fehler

Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Fehler

Framing-Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>'.

Fehlertyp:
Fehler

Kapselungsfehler beim Hochladen der Controller-Programminformationen. | Kapselungsfehler = <Code>.

Fehlertyp:
Fehler

Fehler beim Hochladen der Controller-Programminformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Fehler

Framing-Fehler beim Hochladen der Controller-Programminformationen.

Fehlertyp:
Fehler

Timeout von CIP-Verbindung, während Projektinformationen hochgeladen werden.

Fehlertyp:
Fehler

Mögliche Ursache:

Der Wert für den Inaktivitäts-Watchdog ist zu niedrig, sodass das Projekt nicht geladen werden kann.

Mögliche Lösung:

Erhöhen Sie den Wert für den Inaktivitäts-Watchdog, und führen Sie den Vorgang erneut durch.

Datenbankfehler. Timeout von CIP-Verbindung, während Projektinformationen hochgeladen werden.

Fehlertyp:

Fehler

Mögliche Ursache:

Der Wert für den Inaktivitäts-Watchdog ist zu niedrig, sodass das Projekt nicht geladen werden kann.

Mögliche Lösung:

Erhöhen Sie den Wert für den Inaktivitäts-Watchdog, und führen Sie den Vorgang erneut durch.

Datenbankfehler. Keine Verbindungen mehr verfügbar zum Weiterleiten offener Anforderungen.

Fehlertyp:

Fehler

Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Code>'.

Fehlertyp:

Fehler

Controller wird nicht unterstützt. | Händler-ID = <ID>, Produkttyp = <Typ>, Produktcode = <Code>, Produktname = '<Name>'.

Fehlertyp:

Warnung

Der vom Gerät empfangene Frame enthält Fehler.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Aufgrund der Verbindung/Trennung zwischen dem PC und dem Gerät sind die Pakete falsch ausgerichtet.
2. Fehlerhafte Verbindungskabel zwischen den Geräten verursachen Störungen.

Mögliche Lösung:

1. Verwenden Sie für das Gerät ein Netzwerk mit weniger Datenverkehr.
2. Erhöhen Sie den Wert für Anforderungs-Timeout und/oder -versuche.

Schreibanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Es gibt einen falschen Dienstcode für Anforderungen.
2. Die vom Treiber empfangene Byte-Anzahl war geringer oder höher als erwartet.
3. Wenn dieser Fehler häufig auftritt, besteht möglicherweise ein Problem bei der Verkabelung oder dem Gerät selbst.

Mögliche Lösung:

1. Erhöhen Sie den Wert für 'Erneute Versuche', um eine Wiederherstellung des Treibers nach diesem Fehler zu ermöglichen.
2. Überprüfen Sie die Verkabelung und das Gerät auf eine ordnungsgemäße Funktionsweise.

Leseanforderung für Tag ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Es gibt einen falschen Dienstcode für Anforderungen.
2. Die vom Treiber empfangene Byte-Anzahl war geringer oder höher als erwartet.
3. Wenn dieser Fehler häufig auftritt, besteht möglicherweise ein Problem bei der Verkabelung oder dem Gerät selbst.

Mögliche Lösung:

1. Erhöhen Sie den Wert für 'Erneute Versuche', um eine Wiederherstellung des Treibers nach diesem Fehler zu ermöglichen.
2. Überprüfen Sie die Verkabelung und das Gerät auf eine ordnungsgemäße Funktionsweise.

Block-Leseanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die Ethernet-Verbindung zwischen Gerät und Host-PC ist unterbrochen.
2. Die Kommunikationsparameter für die Ethernet-Verbindung sind falsch.
3. Möglicherweise wurde dem benannten Gerät eine falsche IP-Adresse zugewiesen.

Mögliche Lösung:

1. Überprüfen Sie die Verkabelung zwischen dem PC und dem Gerät.
2. Vergewissern Sie sich, dass für das benannte Gerät der richtige Port angegeben wurde.
3. Vergewissern Sie sich, dass die IP-Adresse des benannten Geräts mit der des eigentlichen Geräts übereinstimmt.

Block-Leseanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Blockgröße = <Anzahl> (Byte), Blockname = '<Name>'.

Fehlertyp:
Warnung

Mögliche Ursache:

1. Es gibt einen falschen Dienstcode für Anforderungen.
2. Die vom Treiber empfangene Byte-Anzahl war geringer oder höher als erwartet.
3. Wenn dieser Fehler häufig auftritt, besteht möglicherweise ein Problem bei der Verkabelung oder dem Gerät selbst.

Mögliche Lösung:

1. Erhöhen Sie den Wert für 'Erneute Versuche', um eine Wiederherstellung des Treibers nach diesem Fehler zu ermöglichen.
2. Überprüfen Sie die Verkabelung und das Gerät auf eine ordnungsgemäße Funktionsweise.

In Tag kann nicht geschrieben werden. | Tag-Adresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Während einer Schreibanforderung für das angegebene Tag wurde vom Gerät ein Fehler im Ethernet/IP-Teil des Pakets zurückgegeben.

Mögliche Lösung:
Die Lösung hängt davon ab, welche Fehlercodes zurückgegeben wurden.

Siehe auch:
CIP-Fehlercodes

Tag kann nicht gelesen werden. | Tag-Adresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Während einer Leseanforderung für das angegebene Tag wurde vom Gerät ein Fehler im CIP-Teil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:
Die Lösung hängt davon ab, welche Fehlercodes zurückgegeben wurden.

Siehe auch:
CIP-Fehlercodes

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Warnung

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Byte), Tag-Name = '<Tag>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Warnung

In Tag kann nicht geschrieben werden. Controller-Tag-Datentyp ist unbekannt. | Tag-Adresse = '<Adresse>', Datentyp = <Typ>.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Schreibanforderung für das angegebene Tag ist fehlgeschlagen, da der Controller-Tag-Datentyp nicht unterstützt wird.

Mögliche Lösung:
Wenden Sie sich an den technischen Support, damit eine Unterstützung für diesen Typ hinzugefügt wird.

Siehe auch:
Adressieren unteilbarer Datentypen

Tag kann nicht gelesen werden. Controller-Tag-Datentyp ist unbekannt. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = <Typ>.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für das angegebene Tag ist fehlgeschlagen, da der Controller-Tag-Datentyp nicht unterstützt wird.

Mögliche Lösung:
Wenden Sie sich an den technischen Support, damit eine Unterstützung für diesen Typ hinzugefügt wird.

Siehe auch:
Adressieren unteilbarer Datentypen

Block kann nicht gelesen werden. Controller-Tag-Datentyp ist unbekannt. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = <Typ>.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für den angegebenen Block ist fehlgeschlagen, da ein Controller-Tag-Datentyp im Block nicht unterstützt wird.

Mögliche Lösung:
Wenden Sie sich an den technischen Support, damit eine Unterstützung für diesen Typ hinzugefügt wird.

Siehe auch:
Adressieren unteilbarer Datentypen

In Tag kann nicht geschrieben werden. Datentyp wird nicht unterstützt. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Schreibanforderung für das angegebene Tag ist fehlgeschlagen, da der Client-Tag-Datentyp nicht unterstützt wird.

Mögliche Lösung:
Ändern Sie Tag-Datentyp in einen unterstützten Typ.

Siehe auch:
Adressieren unteilbarer Datentypen

Tag kann nicht gelesen werden. Datentyp wird nicht unterstützt. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für das angegebene Tag ist fehlgeschlagen, da der Controller-Tag-Datentyp nicht unterstützt wird.

Mögliche Lösung:
Wenden Sie sich an den technischen Support, damit eine Unterstützung für diesen Typ hinzugefügt wird.

Siehe auch:
Adressieren unteilbarer Datentypen

Block kann nicht gelesen werden. Datentyp wird nicht unterstützt. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = '<Typ>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für den angegebenen Block ist fehlgeschlagen, da ein Controller-Tag-Datentyp im Block nicht unterstützt wird.

Mögliche Lösung:
Wenden Sie sich an den technischen Support, damit eine Unterstützung für diesen Typ hinzugefügt wird.

Siehe auch:
Adressieren unteilbarer Datentypen

In Tag kann nicht geschrieben werden. Datentyp für dieses Tag ist unzulässig. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Schreibanforderung für das angegebene Tag ist fehlgeschlagen, da der Client-Tag-Datentyp für das angegebene Controller-Tag unzulässig ist.

Mögliche Lösung:
Ändern Sie den Tag-Datentyp in einen unterstützten Typ. Beispiel: Der Datentyp 'Short' ist für ein Controller-Tag in einem BOOL-Array nicht zulässig. Ändern Sie den Datentyp in 'Boolean', um dieses Problem zu beheben.

Siehe auch:
Adressieren unteilbarer Datentypen

Tag kann nicht gelesen werden. Datentyp für dieses Tag ist unzulässig. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für das angegebene Tag ist fehlgeschlagen, da der Client-Tag-Datentyp für das angegebene Controller-Tag unzulässig ist.

Mögliche Lösung:
Ändern Sie den Tag-Datentyp in einen unterstützten Typ. Beispiel: Der Datentyp 'Short' ist für ein Controller-Tag in einem BOOL-Array nicht zulässig. Ändern Sie den Datentyp in 'Boolean', um das Problem zu beheben.

Siehe auch:
Adressieren unteilbarer Datentypen

Block kann nicht gelesen werden. Datentyp für diesen Block ist unzulässig. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = '<Typ>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für den Block ist fehlgeschlagen, da der Client-Tag-Datentyp für das angegebene Controller-Tag unzulässig ist.

Mögliche Lösung:
Ändern Sie den Tag-Datentyp in diesem Block auf einen unterstützten Typ. Beispiel: Der Datentyp 'Short' ist für ein Controller-Tag in einem BOOL-Array nicht zulässig. Ändern Sie den Datentyp in 'Boolean', um dieses Problem zu beheben.

Siehe auch:
Adressieren unteilbarer Datentypen

In Tag kann nicht geschrieben werden. Tag unterstützt keine Arrays mit mehreren Elementen. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für das angegebene Tag ist fehlgeschlagen, da der Treiber für den angegebenen Controller-Tag keinen Zugriff durch Arrays mit mehreren Elementen unterstützt.

Mögliche Lösung:
Ändern Sie den Tag-Datentyp in einen unterstützten Typ.

Siehe auch:

Adressieren unteilbarer Datentypen

Tag kann nicht gelesen werden. Tag unterstützt keine Arrays mit mehreren Elementen. Tag deaktiviert. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für das angegebene Tag ist fehlgeschlagen, da der Treiber für den angegebenen Controller-Tag keinen Zugriff durch Arrays mit mehreren Elementen unterstützt.

Mögliche Lösung:
Ändern Sie Datentyp oder Adresse des Tags in unterstützte Einstellungen. Als Reaktion auf diesen Fehler wird das Tag deaktiviert und nicht erneut verarbeitet.

Siehe auch:
Adressieren unteilbarer Datentypen

Block kann nicht gelesen werden. Block unterstützt keine Arrays mit mehreren Elementen. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für Tags in diesem Block ist fehlgeschlagen, da der Treiber für den angegebenen Controller-Tag keinen Zugriff durch Arrays mit mehreren Elementen unterstützt.

Mögliche Lösung:
Ändern Sie den Datentyp oder die Adresse für Tags in diesem Block in unterstützte Einstellungen. Als Reaktion auf diesen Fehler werden <Anzahl> Blockelemente deaktiviert und nicht erneut verarbeitet.

Siehe auch:
Adressieren unteilbarer Datentypen

In Tag kann nicht geschrieben werden. Keine Größenübereinstimmung bei nativem Tag. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Größe des nativen Tags (Speicherbedarf) stimmt nicht mit der erwarteten Größe überein, die beim Hochladen des Projekts festgestellt wurde.

Mögliche Lösung:

1. Ändern Sie den Protokollmodus in den symbolischen Modus, und führen Sie den Vorgang erneut durch.
2. Wenden Sie sich an den technischen Support, um den Fehler zu melden.

Tag kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Mögliche Ursache:

Die Größe des nativen Tags (Speicherbedarf) stimmt nicht mit der erwarteten Größe überein, die beim Hochladen des Projekts festgestellt wurde.

Mögliche Lösung:

1. Ändern Sie den Protokollmodus in den symbolischen Modus, und führen Sie den Vorgang erneut durch.
2. Wenden Sie sich an den technischen Support, um den Fehler zu melden.

Block kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Block der Größe des nativen Tags (Speicherbedarf) stimmt nicht mit der erwarteten Größe überein, die beim Hochladen des Projekts festgestellt wurde.

Mögliche Lösung:

1. Ändern Sie den Protokollmodus in den symbolischen Modus, und führen Sie den Vorgang erneut durch.
2. Wenden Sie sich an den technischen Support, um den Fehler zu melden.

Block kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Blockgröße = <Anzahl> (Byte), Blockname = '<Name>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Block der Größe des nativen Tags (Speicherbedarf) stimmt nicht mit der erwarteten Größe überein, die beim Hochladen des Projekts festgestellt wurde.

Mögliche Lösung:

1. Ändern Sie den Protokollmodus in den symbolischen Modus, und führen Sie den Vorgang erneut durch.
2. Wenden Sie sich an den technischen Support, um den Fehler zu melden.

In Tag kann nicht geschrieben werden. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die Ethernet-Verbindung zwischen Gerät und Host-PC ist unterbrochen.
2. Die Kommunikationsparameter für die Ethernet-Verbindung sind falsch.
3. Möglicherweise wurde dem benannten Gerät eine falsche IP-Adresse zugewiesen.

Mögliche Lösung:

1. Überprüfen Sie die Verkabelung zwischen dem PC und dem Gerät.
2. Vergewissern Sie sich, dass für das benannte Gerät der richtige Port angegeben wurde.
3. Vergewissern Sie sich, dass die IP-Adresse des benannten Geräts mit der des eigentlichen Geräts übereinstimmt.

Tag kann nicht gelesen werden. Tag deaktiviert. | Tag-Adresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die Ethernet-Verbindung zwischen Gerät und Host-PC ist unterbrochen.
2. Die Kommunikationsparameter für die Ethernet-Verbindung sind falsch.
3. Möglicherweise wurde dem benannten Gerät eine falsche IP-Adresse zugewiesen.

Mögliche Lösung:

1. Überprüfen Sie die Verkabelung zwischen dem PC und dem Gerät.
2. Vergewissern Sie sich, dass für das benannte Gerät der richtige Port angegeben wurde.
3. Vergewissern Sie sich, dass die IP-Adresse des benannten Geräts mit der des eigentlichen Geräts übereinstimmt.

Hinweis:

Als Reaktion auf diesen Fehler wird das Tag deaktiviert und nicht erneut verarbeitet.

Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die Ethernet-Verbindung zwischen Gerät und Host-PC ist unterbrochen.
2. Die Kommunikationsparameter für die Ethernet-Verbindung sind falsch.
3. Möglicherweise wurde dem benannten Gerät eine falsche IP-Adresse zugewiesen.

Mögliche Lösung:

1. Überprüfen Sie die Verkabelung zwischen dem PC und dem Gerät.
2. Vergewissern Sie sich, dass für das benannte Gerät der richtige Port angegeben wurde.
3. Vergewissern Sie sich, dass die IP-Adresse des benannten Geräts mit der des eigentlichen Geräts übereinstimmt.

Hinweis:

Als Reaktion auf diesen Fehler werden die Blockelemente deaktiviert und nicht erneut verarbeitet.

Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Byte), Tag-Name = '<Tag>'.

Fehlertyp:
Warnung

Mögliche Ursache:

1. Die Ethernet-Verbindung zwischen Gerät und Host-PC ist unterbrochen.
2. Die Kommunikationsparameter für die Ethernet-Verbindung sind falsch.
3. Möglicherweise wurde dem benannten Gerät eine falsche IP-Adresse zugewiesen.

Mögliche Lösung:

1. Überprüfen Sie die Verkabelung zwischen dem PC und dem Gerät.
2. Vergewissern Sie sich, dass für das benannte Gerät der richtige Port angegeben wurde.
3. Vergewissern Sie sich, dass die IP-Adresse des benannten Geräts mit der des eigentlichen Geräts übereinstimmt.

Hinweis:
Als Reaktion auf diesen Fehler werden die Blockelemente deaktiviert und nicht erneut verarbeitet.

Fehler während Anfrage an Gerät. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Während einer Anforderung wurde vom Gerät ein Fehler im CIP-Teil des Ethernet/IP-Pakets zurückgegeben. Alle Lese- und Schreibvorgänge innerhalb der Anforderung sind fehlgeschlagen.

Mögliche Lösung:
Die Lösung hängt davon ab, welche Fehlercodes zurückgegeben wurden.

Siehe auch:
CIP-Fehlercodes

Kapselungsfehler während Anfrage an Gerät. | Kapselungsfehler = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Während einer Anforderung wurde vom Gerät ein Fehler im Kapselungsteil des Ethernet/IP-Pakets zurückgegeben. Alle Lese- und Schreibvorgänge innerhalb der Anforderung sind fehlgeschlagen.

Mögliche Lösung:
Wiederherstellung des Treibers nach diesem Fehler wird versucht. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support. Dies schließt den Fehler 0x02 aus, der sich auf das Gerät und nicht auf den Treiber bezieht.

Siehe auch:
Kapselungsfehlercodes

Speicherplatz für Tag konnte nicht zugeordnet werden. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Die zum Erstellen eines Tags erforderlichen Ressourcen konnten nicht zugeordnet werden. Das Tag wird nicht zum Projekt hinzugefügt.

Mögliche Lösung:
Schließen Sie nicht verwendete Anwendungen, und/oder erhöhen Sie den virtuellen Arbeitsspeicher. Versuchen Sie es anschließend erneut.

Block kann nicht gelesen werden. Empfangener Frame enthält Fehler. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>'.

Fehlertyp:
Warnung

Mögliche Ursache:

1. Es gibt einen falschen Dienstcode für Anforderungen.
2. Die vom Treiber empfangene Byte-Anzahl war geringer oder höher als erwartet.

Mögliche Lösung:

1. Erhöhen Sie den Wert für 'Erneute Versuche', um eine Wiederherstellung des Treibers nach diesem Fehler zu ermöglichen.
2. Wenn dieser Fehler häufig auftritt, besteht möglicherweise ein Problem bei der Verkabelung oder dem Gerät selbst. Wenn der Fehler häufig für ein bestimmtes Tag auftritt, wenden Sie sich an den technischen Support.

Funktionsdatei kann nicht von Gerät gelesen werden. Empfangener Frame enthält Fehler. | Funktionsdatei = '<Name>'.

Fehlertyp:
Warnung

Block kann nicht gelesen werden. Tags deaktiviert. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:
Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:
Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem

bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Funktionsdatei kann nicht von Gerät gelesen werden. Tags deaktiviert. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:
Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:
Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

In Adresse kann nicht geschrieben werden. Empfangener Frame enthält Fehler. | Adresse = '<Adresse>'.

Fehlertyp:
Warnung

In Funktionsdatei kann nicht geschrieben werden. Empfangener Frame enthält Fehler. | Funktionsdatei = '<Name>'.

Fehlertyp:
Warnung

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:
Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:

Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:

Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Funktionsdatei kann nicht gelesen werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:

Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:

Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Block kann nicht gelesen werden. Tags deaktiviert. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:

Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:

Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Funktionsdatei kann nicht gelesen werden. Tags deaktiviert. | Funktionsdatei = '<Name>', DF1-Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:
Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:
Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

In Adresse kann nicht geschrieben werden. | Adresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:
Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:
Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

In Funktionsdatei kann nicht geschrieben werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:

Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:

Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:

Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:

Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Funktionsdatei kann nicht gelesen werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:

Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:

Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

In Adresse kann nicht geschrieben werden. | Adresse = '<Adresse>', DF1-Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:
Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:
Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

In Funktionsdatei kann nicht geschrieben werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die Adresse ist in der SPS nicht vorhanden.

Mögliche Lösung:
Überprüfen Sie den Statuscode und den erweiterten Statuscode, die von der SPS zurückgegeben werden. Erweiterte Statuscodes werden möglicherweise nicht immer zurückgegeben, und die Fehlerinformationen sind im Statuscode enthalten. Die Codes werden im Hexadezimalformat angezeigt.

Hinweis:
Statuscodefehler im Low-Nibble des Statuscodes weisen auf vom lokalen Knoten festgestellte Fehler hin. Vom lokalen Knoten festgestellte Fehler treten auf, wenn das KF-Modul die Ziel-SPS im Netzwerk aus einem bestimmten Grund nicht erkennen kann. Statuscodefehler im High-Nibble des Statuscodes weisen auf von der SPS festgestellte Fehler hin. Diese Fehler werden generiert, wenn die Datenposition in der SPS nicht verfügbar oder nicht beschreibbar ist.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Tag kann nicht gelesen werden. Interner Speicher ist ungültig. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Tag kann nicht gelesen werden. Datentyp für dieses Tag ist unzulässig. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'.

Fehlertyp:
Warnung

Mögliche Ursache:
Eine Leseanforderung für das angegebene Tag ist fehlgeschlagen, da der Client-Tag-Datentyp für das angegebene Controller-Tag unzulässig ist.

Mögliche Lösung:
Ändern Sie den Tag-Datentyp in einen unterstützten Typ. Beispiel: Der Datentyp 'Short' ist für ein Controller-Tag in einem BOOL-Array nicht zulässig. Ändern Sie den Datentyp in 'Boolean', um das Problem zu beheben.

Siehe auch:
Adressieren unteilbarer Datentypen

Block kann nicht gelesen werden. Interner Speicher ist ungültig. Tag deaktiviert. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Block kann nicht gelesen werden. Interner Speicher ist ungültig. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'.

Fehlertyp:
Warnung

In Adresse kann nicht geschrieben werden. Interner Speicher ist ungültig. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Während einer Leseanforderung für den angegebenen Block wurde vom Gerät ein Fehler im CIP-Teil des Ethernet/IP-Pakets zurückgegeben.

Mögliche Lösung:
Die Lösung hängt davon ab, welche Fehlercodes zurückgegeben wurden.

Siehe auch:
CIP-Fehlercodes

Gerät antwortet nicht. Lokaler Knoten hat mit Fehler geantwortet. | DF1-Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Die SPS hat auf die Anforderung vom lokalen Knoten nicht geantwortet. Ein lokaler Knoten kann ein Zwischenknoten (z.B. 1756-DHRIO, 1756-CNB, 1761-NET-ENI) sein.

Mögliche Lösung:
Weitere Informationen zu Fehlercodedefinitionen erhalten Sie in der Allen-Bradley-Dokumentation. Wenn beispielsweise der STS-Code '0x02' (hex) zurückgegeben wird, überprüfen Sie die Verkabelung zwischen dem Remote-Knoten (SPS) und dem lokalen Knoten.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

In Funktionsdatei kann nicht geschrieben werden. Lokaler Knoten hat mit Fehler geantwortet. | Funktionsdatei = '<Name>', DF1-Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Dieser Fehler bedeutet, dass die SPS nicht auf die Schreibanforderung vom lokalen Knoten geantwortet hat. Ein lokaler Knoten kann ein Zwischenknoten (z.B. 1756-DHRIO, 1756-CNB, 1761-NET-ENI) sein.

Mögliche Lösung:
Weitere Informationen zu STS-Fehlercodedefinitionen erhalten Sie in der Allen-Bradley-Dokumentation. Wenn beispielsweise der STS-Code '0x02' (hex) zurückgegeben wird, überprüfen Sie die Verkabelung zwischen dem Remote-Knoten (SPS) und dem lokalen Knoten.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

In Adresse kann nicht geschrieben werden. Lokaler Knoten hat mit Fehler geantwortet. | Funktionsdatei = '<Name>', DF1-Status = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Dieser Fehler bedeutet, dass die SPS nicht auf die Schreibanforderung vom lokalen Knoten geantwortet hat. Ein lokaler Knoten kann ein Zwischenknoten (z.B. 1756-DHRIO, 1756-CNB, 1761-NET-ENI) sein.

Mögliche Lösung:
Weitere Informationen zu STS-Fehlercodedefinitionen erhalten Sie in der Allen-Bradley-Dokumentation. Wenn beispielsweise der STS-Code '0x02' (hex) zurückgegeben wird, überprüfen Sie die Verkabelung zwischen dem Remote-Knoten (SPS) und dem lokalen Knoten.

Siehe auch:
Allen-Bradley-Dokumentation für Fehlercodedefinitionen

Unerwarteter Offset für Tag festgestellt. Tag verwendet stattdessen symbolisches Protokoll. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Unerwarteter Offset für Tag festgestellt. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Unerwarteter Offset bzw. Spanne für Tag festgestellt. | Tag-Adresse = '<Adresse>'.

Fehlertyp:
Warnung

Herunterladen von Projekt in Bearbeitung oder kein Projekt vorhanden.

Fehlertyp:
Warnung

Herunterladen von Projekt abgeschlossen.

Fehlertyp:
Warnung

Online-Bearbeitung des Projekts festgestellt. Zurzeit wird symbolische Adressierung verwendet.

Fehlertyp:
Warnung

Offline-Bearbeitung des Projekts festgestellt. Zurzeit wird symbolische Adressierung verwendet.

Fehlertyp:
Warnung

Die folgenden Fehler sind beim Hochladen des Controller-Projekts vom Gerät aufgetreten. Das symbolische Protokoll wird stattdessen verwendet.

Fehlertyp:
Warnung

ID für Gerät kann nicht abgerufen werden. Alle Tags verwenden stattdessen das symbolische Protokoll. | Kapselungsfehler = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Während einer Anforderung wurde vom Gerät ein Fehler im Kapselungsteil des Ethernet/IP-Pakets zurückgegeben. Geräte, für die ein logischer Modus festgelegt ist, werden auf symbolischen Modus zurückgesetzt, bis der Fehler behoben ist.

Mögliche Lösung:
Wiederherstellung des Treibers nach diesem Fehler wird versucht. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support. Dies schließt den Fehler 0x02 aus, der sich auf das Gerät und nicht auf den Treiber bezieht.

Siehe auch:
Kapselungsfehlercodes

ID für Gerät kann nicht abgerufen werden. Alle Tags verwenden stattdessen das symbolische Protokoll. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>.

Fehlertyp:
Warnung

Mögliche Ursache:
Während einer Anforderung wurde vom Gerät ein Fehler im CIP-Teil des Ethernet/IP-Pakets zurückgegeben. Geräte, für die ein logischer Modus festgelegt ist, werden auf symbolischen Modus zurückgesetzt, bis der Fehler behoben ist.

Mögliche Lösung:
Die Lösung hängt vom zurückgegebenen Fehlercode ab. Wenn das Problem weiterhin auftritt, wenden Sie sich an den technischen Support.

Siehe auch:
CIP-Fehlercodes

ID für Gerät kann nicht abgerufen werden. Empfangener Frame enthält Fehler. Alle Tags verwenden stattdessen das symbolische Protokoll.

Fehlertyp:
Warnung

Mögliche Ursache:

1. Aufgrund der Verbindung/Trennung zwischen dem PC und dem Gerät sind die Pakete falsch ausgerichtet.
2. Fehlerhafte Verbindungskabel zwischen den Geräten verursachen Störungen.
3. Die falsche Frame-Größe wurde empfangen.
4. Keine TNS-Übereinstimmung.
5. Vom Gerät wurde ein ungültiger Antwortbefehl zurückgegeben.
6. Das Gerät ist nicht Ethernet/IP-fähig.

Mögliche Lösung:

1. Der Treiber kann ohne weitere Maßnahmen nach diesem Fehler wiederhergestellt werden. Wenn dieser Fehler häufig auftritt, besteht möglicherweise ein Fehler bei der Verkabelung, dem Netzwerk oder dem Gerät selbst.
2. Vergewissern Sie sich, dass das Gerät, mit dem die Kommunikation erfolgt, ein Ethernet-fähiges Gerät ist.

Angeforderte CIP-Verbindungsgröße wird von diesem Gerät nicht unterstützt. Es wird automatisch auf die maximale Größe ausgewichen. | Angeforderte Größe = <Anzahl> (Byte), max. Größe = <Anzahl> (Byte).

Fehlertyp:
Warnung

Mögliche Ursache:
Die angeforderte CIP-Verbindungsgröße wird vom Gerät nicht unterstützt.

Mögliche Lösung:
Ändern Sie die CIP-Verbindungsgröße auf eine vom Gerät unterstützte Größe.

Siehe auch:
Logix-Kommunikationsparameter

Der Dateiname für den Tag-Import ist ungültig. Dateipfade sind nicht erlaubt.

Fehlertyp:

Warnung

Mögliche Ursache:
Der Dateiname für den Tag-Import enthält einen Pfad.

Mögliche Lösung:
Pfad aus dem Dateinamen entfernen.

Lese-/Schreibanfragen an Gerät beendet. Aktualisierung logischer Adressen vom Geräteprojekt läuft.

Fehlertyp:
Warnung

Lese-/Schreibanfragen an Gerät wurden fortgesetzt. Aktualisierung logischer Adressen vom Geräteprojekt abgeschlossen. Zurzeit wird logische Adressierung verwendet.

Fehlertyp:
Warnung

Datenbankstatus. Tags ohne Alias werden importiert.

Fehlertyp:
Informationen

Datenbankstatus. Tags mit Alias werden importiert.

Fehlertyp:
Informationen

Datenbankstatus. Tag-Projekte werden erstellt. Bitte warten. | Anzahl der Tag-Projekte = <Anzahl>.

Fehlertyp:
Informationen

Datenbankfehler. Tag umbenannt, da maximale Zeichenlänge überschritten wurde. | Tag-Name = '<Tag>', max. Länge = <Anzahl>, neuer Tag-Name = '<Tag>'.

Fehlertyp:
Informationen

Datenbankfehler. Array-Tags umbenannt, da sie maximale Zeichenlänge überschreiten. | Array-Tags = '<Tags>', max. Länge = <Anzahl>, neue Array-Tags = '<Tags>'.

Fehlertyp:
Informationen

Datenbankfehler. Programmgruppenname überschreitet maximale Zeichenlänge. Programmgruppe umbenannt. | Gruppenname = '<Name>', max. Länge = <Anzahl>, neuer Gruppenname = '<Name>'.

Fehlertyp:
Informationen

Datenbankstatus. Controller-Projekt wird abgerufen.

Fehlertyp:
Informationen

Datenbankstatus. | Programmanzahl = <Anzahl>, Datentypanzahl = <Anzahl>, Anzahl importierter Tags = <Anzahl>.

Fehlertyp:
Informationen

Datenbankstatus. OPC-Tags werden generiert.

Fehlertyp:
Informationen

Zu wenig Speicherressourcen.

Fehlertyp:
Informationen

Unbekannter Fehler aufgetreten.

Fehlertyp:
Informationen

Datenbankstatus. Tags werden aus L5X-Datei importiert. | Schemarevision = '<Wert>', Softwarerevision = '<Wert>'.

Fehlertyp:
Informationen

Details. | IP = '<Adresse>', Händler-ID = <Händler>, Produkttyp = <Typ>, Produktcode = <Code>, Revision= <Wert>, Produktname = '<Name>', Produkt-Seriennummer = <Anzahl>.

Fehlertyp:
Informationen

Verstrichene Zeit = <Anzahl> (Sekunden).

Fehlertyp:
Informationen

Symbolisch - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Symbolisch, Array-Block - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Symbolisch, Array-Block - Lesevorgänge von Cache = <Anzahl>.

Fehlertyp:
Informationen

Symbolinstanz nicht blockierend - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Symbolinstanz nicht blockierend, Array-Block - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Symbolinstanz nicht blockierend, Array-Block - Lesevorgänge von Cache = <Anzahl>.

Fehlertyp:
Informationen

Symbolinstanz blockierend - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Symbolinstanz blockierend - Lesevorgänge von Cache = <Anzahl>.

Fehlertyp:
Informationen

Physisch nicht blockierend - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Physisch nicht blockierend, Array-Block - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Physisch nicht blockierend, Array-Block - Lesevorgänge von Cache = <Anzahl>.

Fehlertyp:
Informationen

Physisch blockierend - Lesevorgänge von Gerät = <Anzahl>.

Fehlertyp:
Informationen

Physisch blockierend - Lesevorgänge von Cache = <Anzahl>.

Fehlertyp:
Informationen

Lesevorgänge für Tags = <Anzahl>.

Fehlertyp:
Informationen

Pakete gesendet = <Anzahl>.

Fehlertyp:
Informationen

Pakete empfangen = <Anzahl>.

Fehlertyp:
Informationen

Initialisierungstransaktionen = <Anzahl>.

Fehlertyp:
Informationen

Lese-/Schreibtransaktionen = <Anzahl>.

Fehlertyp:
Informationen

Pakete durchschnittl. gesendet/Sekunde = <Anzahl>.

Fehlertyp:
Informationen

Pakete durchschnittl. empfangen/Sekunde = <Anzahl>.

Fehlertyp:
Informationen

Durchschnittl. Lesevorgänge/Sekunde für Tag = <Anzahl>.

Fehlertyp:
Informationen

Durchschnittl. Tags/Transaktion = <Anzahl>.

Fehlertyp:
Informationen

Fehlertyp:
Informationen

%s | GERÄTESTATISTIK

Fehlertyp:
Informationen

Durchschnittliche Verarbeitungszeit für Gerät = <Anzahl> (Millisekunden)

Fehlertyp:
Informationen

%s | KANALSTATISTIK

Fehlertyp:
Informationen

TREIBERSTATISTIK

Fehlertyp:
Informationen

Importieren von Geräte-Tag abgebrochen.

Fehlertyp:
Informationen

Importdatei '%s' nicht gefunden unter Pfad '%s'.

Fehlertyp:
Informationen

Beim Abrufen des Controller-Projekts sind Fehler aufgetreten.

Fehlertyp:
Informationen

Interner Treiberfehler aufgetreten.

Fehlertyp:
Informationen

Ungültiges oder fehlerhaftes Controller-Projekt wurde beim Synchronisieren festgestellt. Versuchen Sie es später erneut.

Fehlertyp:
Informationen

Während der Synchronisierung wurde ein Herunterladen des Projekts festgestellt. Versuchen Sie es später erneut.

Fehlertyp:
Informationen

Zu wenig Speicherressourcen.

Fehlertyp:
Informationen

L5K-Datei ist ungültig oder fehlerhaft.

Fehlertyp:
Informationen

Unbekannter Fehler aufgetreten.

Fehlertyp:
Informationen

Datenbankfehler. PLC5/SLC/MicroLogix-Geräte unterstützen diese Funktion nicht.

Fehlertyp:
Informationen

L5X-Datei ist ungültig oder fehlerhaft.

Fehlertyp:
Informationen

Import-Datei '<empty>' nicht gefunden unter Pfad '<empty>'.

Fehlertyp:

Informationen

Import-Datei '%s' nicht gefunden unter Pfad '<empty>'.

Fehlertyp:
Informationen

Import-Datei '<empty>' nicht gefunden unter Pfad '%s'.

Fehlertyp:
Informationen

Fehler bei XML-Element in Post-Schema-Validierung. Modell unterstützt kein Importieren der Tags von Gerät. Verwenden Sie ein alternatives Element. | XML-Element = '{<Namespace>}<Element>', nicht unterstütztes Modell = '<Modell>', alternatives XML-Element = '{<Namespace>}<Element>'.

Fehlertyp:
Sicherheit

Wert wird für ein XML-Element in diesem Modell nicht unterstützt. Ein neuer Wert wird automatisch festgelegt. | Wert = '<Wert>', XML-Element = '{<Namespace>}<Element>', Modell = '<Modell>', neuer Wert = '<Wert>'.

Fehlertyp:
Sicherheit

Anhänge

Wählen Sie einen Link aus der nachfolgenden Liste aus, um weitere Informationen zu einem bestimmten Thema zu erhalten.

Kanaleigenschaften
Geräteeigenschaften
Tag-Eigenschaften

Logix-Setup
1761-NET-ENI-Setup
Data Highway Plus-Gateway-Setup
Kommunikations-Routing
Serial Gateway-Setup
Data Highway Plus-Gateway
ControlNet-Gateway
EtherNet/IP-Gateway-Setup
MicroLogix 1100 Setup
Protokollmodus auswählen
Änderung im Controller-Projekt erkennen
Hinweise zu SoftLogix 5800-Verbindung
Glossar

Anhang A - Kanaleigenschaften

Nachfolgend finden Sie eine vollständige Liste aller Eigenschaften für die Allen-Bradley ControlLogix Ethernet-Kanalebene.


```
{ "common.ALLTYPES_NAME": "MyChannel", "common.ALLTYPES_DESCRIPTION": "", "ser-
vermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet", "ser-
vermain.CHANNEL_DIAGNOSTICS_CAPTURE": false, "servermain.CHANNEL_UNIQUE_ID":
4126021724, "servermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING":
"", "servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2, "servermain.CHANNEL_WRITE_
OPTIMIZATIONS_DUTY_CYCLE": 10, "servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_
HANDLING": 0, }
```


Anhang B - Geräteeigenschaften

Nachfolgend finden Sie eine vollständige Liste aller Eigenschaften für die Allen-Bradley ControlLogix Ethernet Geräteebene.


```
{ "common.ALLTYPES_NAME": "MyDevice", "common.ALLTYPES_DESCRIPTION": "", "ser-
vermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet", "ser-
vermain.DEVICE_MODEL": 0, "servermain.DEVICE_UNIQUE_ID": 1286734974,
"servermain.DEVICE_CHANNEL_ASSIGNMENT": "Channel1", "servermain.DEVICE_ID_FORMAT":
0, "servermain.DEVICE_ID_STRING": "<10.10.110.15>,1,0", "servermain.DEVICE_ID_
HEXADECIMAL": 0, "servermain.DEVICE_ID_DECIMAL": 0, "servermain.DEVICE_ID_OCTAL":
0, "servermain.DEVICE_DATA_COLLECTION": true, "servermain.DEVICE_SIMULATED":
false, "servermain.DEVICE_SCAN_MODE": 0, "servermain.DEVICE_SCAN_MODE_RATE_MS":
1000, "servermain.DEVICE_SCAN_MODE_PROVIDE_INITIAL_UPDATES_FROM_CACHE": false,
"servermain.DEVICE_CONNECTION_TIMEOUT_SECONDS": 3, "servermain.DEVICE_REQUEST_
TIMEOUT_MILLISECONDS": 1000, "servermain.DEVICE_RETRY_ATTEMPTS": 3, "ser-
vermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS": 0, "servermain.DEVICE_AUTO_
DEMOTION_ENABLE_ON_COMMUNICATIONS_FAILURES": false, "servermain.DEVICE_AUTO_
DEMOTION_DEMOTE_AFTER_SUCCESSIVE_TIMEOUTS": 3, "servermain.DEVICE_AUTO_DEMOTION_
PERIOD_MS": 10000, "servermain.DEVICE_AUTO_DEMOTION_DISCARD_WRITES": false,
```

```
"servermain.DEVICE_TAG_GENERATION_ON_STARTUP": 0, "servermain.DEVICE_TAG_
GENERATION_DUPLICATE_HANDLING": 0, "servermain.DEVICE_TAG_GENERATION_GROUP": "",
"servermain.DEVICE_TAG_GENERATION_ALLOW_SUB_GROUPS": true, "controllogix_ether-
net.DEVICE_PORT_NUMBER": 44818, "controllogix_ethernet.DEVICE_CONNECTION_SIZE_
BYTES": 500, "controllogix_ethernet.DEVICE_INACTIVITY_WATCHDOG_SECONDS": 32, "con-
trollogix_ethernet.DEVICE_ARRAY_BLOCK_SIZE_ELEMENTS": 120, "controllogix_ether-
net.DEVICE_PROTOCOL_MODE": 1, "controllogix_ethernet.DEVICE_ONLINE_EDITS": true,
"controllogix_ethernet.DEVICE_OFFLINE_EDITS": true, "controllogix_ethernet.DEVICE_
AUTOMATICALLY_READ_STRING_LENGTH": true, "controllogix_ethernet.DEVICE_DEFAULT_
DATA_TYPE": -1, "controllogix_ethernet.DEVICE_ENABLE_PERFORMANCE_STATISTICS":
false, "controllogix_ethernet.DEVICE_DATABASE_IMPORT_METHOD": 1, "controllogix_
ethernet.DEVICE_TAG_IMPORT_FILE": "Veracode2.L5X", "controllogix_ethernet.DEVICE_
DISPLAY_DESCRIPTIONS": true, "controllogix_ethernet.DEVICE_LIMIT_TAG_NAMES":
false, "controllogix_ethernet.DEVICE_TAG_HIERARCHY": 1, "controllogix_ether-
net.DEVICE_IMPOSE_ARRAY_ELEMENT_COUNT_LIMIT": false, "controllogix_ether-
net.DEVICE_ARRAY_ELEMENT_COUNT_LIMIT": 2000, "controllogix_ethernet.DEVICE_CL_
ENET_PORT_NUMBER": 44818, "controllogix_ethernet.DEVICE_REQUEST_SIZE": 232, "con-
trollogix_ethernet.DEVICE_PERFORM_BLOCK_WRITES": false, "controllogix_ether-
net.DEVICE_SLOT_CONFIGURATION": { "controllogix_ethernet.DEVICE_SLOT_
CONFIGURATION_MODULE": [ 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255, 255, 255 ], "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_INPUT_WORDS": [
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0 ], "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_OUTPUT_WORDS": [ 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
] } }
```


Anhang C – Tag-Eigenschaften

Nachfolgend finden Sie eine vollständige Liste aller Allen-Bradley ControlLogix Ethernet Tag-Eigenschaften.


```
{ "common.ALLTYPES_NAME": "MyTag1", "common.ALLTYPES_DESCRIPTION": "", "ser-
vermain.TAG_ADDRESS": "MyTag1", "servermain.TAG_DATA_TYPE": 6, "servermain.TAG_
READ_WRITE_ACCESS": 1, "servermain.TAG_SCAN_RATE_MILLISECONDS": 100, "ser-
vermain.TAG_AUTOGENERATED": true, "servermain.TAG_SCALING_TYPE": 0, "ser-
vermain.TAG_SCALING_RAW_LOW": 0, "servermain.TAG_SCALING_RAW_HIGH": 1000,
"servermain.TAG_SCALING_SCALED_DATA_TYPE": 9, "servermain.TAG_SCALING_SCALED_LOW":
0, "servermain.TAG_SCALING_SCALED_HIGH": 1000, "servermain.TAG_SCALING_CLAMP_LOW":
false, "servermain.TAG_SCALING_CLAMP_HIGH": false, "servermain.TAG_SCALING_NEGATE_
VALUE": false, "servermain.TAG_SCALING_UNITS": "" }
```

Logix-Geräte-IDs

Informationen zum ENI-Geräte-ID-Setup finden Sie unter 1761-NET-ENI-Setup.

ControlLogix 5500-Ethernet

Die Geräte-ID gibt die IP-Adresse des Geräts sowie die Steckplatznummer an, unter der sich die Controller-CPU befindet. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<optionaler Routing-Pfad>],<CPU-Steckplatz>

Designator	Designator-Typ*	Description	Formate	Bereich
IP/Host-Name	k.A.	IP-Adresse oder Host-Name.	Dezimalwert	0-255
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-Pfad	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem EtherNet/IP-Schnittstellenmodul fest und muss gleich 1 sein (Port zur Hintergrundebene).	Dezimalwert	*
CPU-Steckplatz	Verknüpfungsadresse	Steckplatznummer des ControlLogix-Prozessors.	Dezimalwert	0-255

**Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*

Beispiel
123.123.123.123,1,0

Dies entspricht einer Ethernet/IP von 123.123.123.123. Die Port-ID ist 1 und die CPU befindet sich im Steckplatz 0.

Ethernet-Geräte-ID für CompactLogix 5300

Die Geräte-ID gibt die IP-Adresse des Geräts sowie die Steckplatznummer an, unter der sich die Controller-CPU befindet. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<optionaler Routing-Pfad>],<CPU-Steckplatz>

Designator	Designator-Typ*	Description	Formate	Bereich
IP/Host-Name	k.A.	CompactLogix-Ethernet-IP-Adresse oder -Host-Name.	Dezimalwert	0-255
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-Pfad	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem Ethernet-Port fest und muss gleich 1 sein (Port zur Hintergrundebene).	Dezimalwert	*
CPU-Steckplatz	Verknüpfungsadresse	Steckplatznummer des CompactLogix-Prozessors.	Dezimalwert	0-255

**Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*

Beispiel
123.123.123.123,1,0

Dies entspricht einer CompactLogix-IP von 123.123.123.123. Die Port-ID ist 1 und die CPU befindet sich im Steckplatz 0.

Ethernet-Geräte-ID für FlexLogix 5400

Die Geräte-ID gibt die IP-Adresse des Geräts sowie die Steckplatznummer an, unter der sich die Controller-CPU befindet. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<optionaler Routing-Pfad>],<CPU-Steckplatz>

Designator	Designator-Typ*	Description	Formate	Bereich
IP/Host-Name	k.A.	IP-Adresse oder Host-Name für 1788-ENBT.	Dezimalwert	0-255
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-Pfad	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem 1788-ENBT-Schnittstellenmodul fest und muss gleich 1 sein (Port zur Hintergrundebene).	Dezimalwert	*
CPU-Steckplatz	Verknüpfungsadresse	Steckplatznummer des FlexLogix-Prozessors.	Dezimalwert	0-255

**Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*

Beispiel
123.123.123.123,1,0

Dies entspricht einer 1788-ENBT-IP von 123.123.123.123. Die Port-ID ist 1 und die CPU befindet sich im Steckplatz 0.

Geräte-ID für SoftLogix 5800

Die Geräte-ID gibt die IP-Adresse für den SoftLogix-PC sowie die virtuelle Steckplatznummer an, unter der sich die Controller-CPU befindet. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<optionaler Routing-Pfad>],<CPU-Steckplatz>

Designator	Designator-Typ*	Description	Formate	Bereich
IP/Host-Name	k.A.	NIC-IP-Adresse oder Host-Name für SoftLogix-PC.	Dezimalwert	0-255
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-Pfad	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem EtherNet/IP-Messaging-Modul fest und muss gleich 1 sein (Port zur virtuellen Hintergrundebene).	Dezimalwert	*
CPU-Steckplatz	Verknüpfungsadresse	Steckplatznummer des SoftLogix-Prozessors in der virtuellen Hintergrundebene.	Dezimalwert	0-255

**Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*

Beispiel
123.123.123.123,1,1

Dies entspricht einer SoftLogix-PC-IP-Adresse von 123.123.123.123. Die Port-ID ist 1 und die CPU befindet sich im Steckplatz 1.

Informationen zum Ergänzen einer Geräte-ID mit einem Routing-Pfad zu einer Remote-Hintergrundebene finden Sie unter Kommunikations-Routing.

Siehe auch: *Hinweise zur SoftLogix 5800-Verbindung*

1761-NET-ENI-Setup

1761-NET-ENI ermöglicht die Kommunikation mit SPS der Reihe ControlLogix, CompactLogix, FlexLogix, MicroLogix, SLC 500 und PLC-5 über Ethernet mit ControlLogix-Ethernet-Treiber von Allen-Bradley.

Anforderungen

SPS der Reihe MicroLogix, SLC 500 oder PLC-5, die DF1 (Vollduplex) unterstützen und dabei den Kanal CH0 RS232 verwenden.
1761-NET-ENI-Gerät der Reihe A, B, C oder D.

ControlLogix-, CompactLogix- oder FlexLogix-SPS, die den Kanal CH0 RS232 verwenden.
1761-NET-ENI-Gerät der Reihe B und neuer.

Hinweise:

1. Was Kommunikationsparameter, Datenbankeinstellungen und Projekt-/Protokolloptionen angeht, sollten Benutzer von ControlLogix, CompactLogix und FlexLogix für ENI Informationen im Handbuch "Logix-Setup" im Inhaltsverzeichnis finden.
2. Verwenden Sie zur Aktivierung der Option **CompactLogix Routing** (auf der Registerkarte **ENI IP Addr** des Dienstprogramms) das von Allen-Bradley bereitgestellte ENI/ENIW-Dienstprogramm. Dies wurde für ein ENI-Modul mit Firmware-Revision 2.31 getestet.

Das ENI-Modul verfügt über eine begrenzte Zahl von TCP-Verbindungen. Dementsprechend sollten Benutzer Anwendungen vermeiden, die mit dem Modul (beispielsweise RSLinx/RSWho) kommunizieren, sodass Verbindungen für den Treiber zur Verfügung stehen.

ENI-Geräte-ID

Die Geräte-ID gibt die IP-Adresse des 1761-NET-ENI an. Geräte-IDs werden wie folgt angegeben:

<IP-Adresse>

Designator	Designator-Typ	Description	Formate	Bereich
IP-Adresse	k.A.	IP-Adresse für 1761-NET-ENI	Dezimalwert	0-255

Beispiel
123.123.123.123

Dies entspricht einer ENI-IP von 123.123.123.123. Da das Gerät nur DF1 (Vollduplex) unterstützt, ist keine Knoten-ID erforderlich.

Weitere Informationen zu Kommunikationsparametern finden Sie unter Logix-Kommunikationsparameter.

Data Highway Plus-Gateway-Setup

Das DH+-Gateway ermöglicht die Kommunikation mit SPS der SLC 500- und PLC-5-Reihe über DH+ mit ControlLogix-Ethernet-Treiber von Allen-Bradley.

Anforderungen

EtherNet/IP-Schnittstellenmodul.
1756-DHRIO-Schnittstellenmodul mit geeignetem, für DH+ konfigurierten Kanal.
SPS der SLC500- oder PLC-5-Reihe im DH+-Netzwerk.

Hinweis: DH+-Gateway-Modelle unterstützen die automatische Tag-Datenbankgenerierung nicht.

Geräte-ID für DH+-Gateway

Die Geräte-ID gibt die IP-Adresse des Geräts sowie die DH+-Parameter an, die zum Herstellen einer Verbindung erforderlich sind. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<optionaler Routing-Pfad>],<DHRIO-Steckplatz>.<DHRIO-Kanal>.<DH+-Knoten-ID (Dez)>

Designator	Designator-Typ*	Beschreibung	Formate	Bereich
IP/Host-Name	k.A.	IP-Adresse oder Host-Name.	Dezimalwert	0-255
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-Pfad	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem EtherNet/IP-Schnittstellenmodul fest und muss gleich 1 sein (Port zur Hintergrundebene).	Dezimalwert	*
DHRIO-Steckplatz	Verknüpfungsadresse	Steckplatznummer des 1756-DHRIO-Schnittstellenmoduls.	Dezimalwert	0-255

Designator	Designator-Typ*	Beschreibung	Formate	Bereich
DHRIO-Kanal		Zu verwendender DH+-Kanal.	Alpha	A und B
DH+-Kno-ten-ID		DG+-Knoten-ID der Ziel-SPS im Dezi-malformat.**	Dezimalwert	0-99

**Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*
***Weitere Informationen dazu finden Sie im Folgenden unter "Oktaladressierung für Knoten-ID".*

Beispiel
123.123.123.123,1,2.A.3

Dies entspricht einer Ethernet/IP von 123.123.123.123. Die DH+-Karte befindet sich im Steckplatz 2: Verwenden Sie DH+-Kanal A und DH+-Knoten-ID 3 (Dez).

Oktaladressierung für Knoten-ID

Die DH+-Knoten-ID wird in der SPS im Oktalformat angegeben und erfordert für die Verwendung in der DH+-Gateway-Geräte-ID eine Konvertierung in das Dezimalformat. Die Knoten-ID kann in RSWho innerhalb von RSLinx gefunden werden. Sie wird im Oktalformat angezeigt.

Beispiel
DH+-Knoten 10 (oktal) in RSWho = DH+-Knoten 8 (dezimal) in DH+-Gateway-Geräte-ID.

Es ist wichtig, die Kommunikation mit dem richtigen Controller zu überprüfen. Für das Beispiel oben gilt Folgendes: Wenn 10 als DH+-Knoten-ID in der DH+-Gateway-Geräte-ID eingegeben wurde, würde die Kommunikation mit Knoten 12 (oktal entspricht 10 dezimal) und nicht mit Knoten 10 (oktal) stattfinden. Wenn Knoten 12 (oktal) nicht vorhanden ist, würde das DHRIO-Modul DF1 STS 0x02 zurückgeben. Dies bedeutet, dass die Sicherungsschicht die Paketlieferung nicht gewährleisten kann. Kurzum konnte der DH+-Knoten im DH+-Netzwerk nicht gefunden werden.

Informationen zum Ergänzen einer Geräte-ID mit einem Routing-Pfad zum Remote-DH+-Knoten finden Sie unter Kommunikations-Routing.
Weitere Informationen zu Kommunikationsparametern finden Sie unter Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway.

ControlNet™-Gateway-Setup

Das ControlNet-Gateway ermöglicht die Kommunikation mit SPS der PLC-5C-Reihe unter ControlNet mit ControlLogix-Ethernet-Treiber von Allen-Bradley.

Anforderungen

EtherNet/IP-Schnittstellenmodul.
1756-CNB- oder 1756-CNBR-Schnittstellenmodul.
SPS der PLC-5C-Reihe im ControlNet-Netzwerk.

Hinweis: ControlNet-Gateway-Modelle unterstützen die automatische Tag-Datenbankgenerierung nicht.

Geräte-ID für den ControlNet-Gateway

Die Geräte-ID gibt die IP-Adresse des Geräts zusätzlich zu den ControlNet-Parametern an, die zum Herstellen einer Verbindung erforderlich sind. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<optionaler Routing-Pfad>],<CNB-Steckplatz>.<CNB-Kanal>.<ControlNet-Knoten-ID (Dez)>

Designator	Designator-Typ*	Description	Formate	Bereich
IP/Host-Name	k.A.	IP-Adresse oder Host-Name.	Dezimalwert	0-255
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem EtherNet/IP-Kommunikationsmodul fest und muss gleich 1	Dezimalwert	*

Designator	Designator-Typ*	Description	Formate	Bereich
Pfad		sein (Port zur Hintergrundebene).		
CNB-Steckplatz	Verknüpfungsadresse	Steckplatznummer des 1756-CNB/CNBR-Schnittstellenmoduls.	Dezimalwert	0-255
CNB-Kanal	Port-ID	Der zu verwendende ControlNet-Kanal.	Alpha	A und B
ControlNet-Knoten-ID	Verknüpfungsadresse	ControlNet-Knoten-ID der Ziel-SPS im Dezimalformat.**	Dezimalwert	0-99

- **Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*
- ***Weitere Informationen dazu finden Sie im Folgenden unter "Oktaladressierung für Knoten-ID".*

Beispiel
123.123.123.123,1,2.A.3

Dies entspricht einer Ethernet/IP von 123.123.123.123. Die ControlNet-Karte befindet sich im Steckplatz 2: Verwenden Sie ControlNet-Kanal A und ControlNet-Knoten-ID 3 als Adressierungsziel.

Oktaladressierung für Knoten-ID

Die ControlNet-Knoten-ID wird in der SPS im Oktalformat angegeben und erfordert für die Verwendung in der ControlNet-Gateway-Geräte-ID eine Konvertierung ins Dezimalformat. Die Knoten-ID kann in RSWho innerhalb von RSLinx gefunden werden. Sie wird im Oktalformat angezeigt.

Beispiel
CN-Knoten 10 (oktal) in RSWho = CN-Knoten 8 (dezimal) in ControlNet-Gateway-Geräte-ID.

Es ist wichtig, die Kommunikation mit dem richtigen Controller zu überprüfen. Für das Beispiel oben gilt Folgendes: Wenn 10 als ControlNet-Knoten-ID in der ControlNet-Gateway-Geräte-ID eingegeben wurde, findet die Kommunikation mit Knoten 12 (oktal entspricht 10 dezimal) und nicht mit Knoten 10 (oktal) statt. Wenn Knoten 12 (oktal) nicht vorhanden ist, gibt das CNB-Modul DF1 STS 0x02 zurück. Dies bedeutet, dass die Sicherungsschicht die Paketlieferung nicht gewährleisten konnte. Kurzum konnte der ControlNet-Knoten im ControlNet-Netzwerk also nicht gefunden werden.

Hinweise:

1. Weitere Informationen zum Ergänzen einer Geräte-ID mit einem Routing-Pfad zum Remote-ControlNet-Knoten finden Sie unter Kommunikations-Routing.
2. Weitere Informationen zu Kommunikationsparametern finden Sie unter Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway.

EtherNet/IP-Gateway-Setup

Das EtherNet/IP-Gateway ermöglicht die Kommunikation mit SPS der MicroLogix-, SLC 500- und PLC-5-Reihe auf EtherNet/IP mit ControlLogix-Ethernet-Treiber von Allen-Bradley.

Anforderungen

Mindestens zwei EtherNet/IP-Schnittstellenmodule (z.B. 1756-ENBT).
SPS der MicroLogix-, SLC500- oder PLC-5-Reihe mit EtherNet/IP-Konnektivität.

Hinweis: EthernetIP-Gateway-Modelle unterstützen die automatische Tag-Datenbankgenerierung nicht.

EtherNet/IP-Gateway-Geräte-ID

Die Geräte-ID gibt die IP-Adresse des lokalen Geräts sowie die Remote-EtherNet/IP-Adresse an, die zum Herstellen einer Verbindung erforderlich sind. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<optionaler Routing-Pfad>],<ENBT-Steckplatz>.<ENBT-Kanal>.<Remote-IP>

Designator	Designator-Typ*	Description	Formate	Bereich
IP/Host-Name	k.A.	IP-Adresse oder Host-Name des lokalen EtherNet/IP-Schnittstellenmoduls.	Dezimalwert	0-255

Designator	Designator-Typ*	Description	Formate	Bereich
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-Pfad	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem EtherNet/IP-Schnittstellenmodul fest und muss gleich 1 sein (Port zur Hintergrundebene).	Dezimalwert	*
ENBT-Steckplatz	Verknüpfungsadresse	Die Steckplatznummer des zweiten EtherNet/IP-Schnittstellenmoduls.	Dezimalwert	0-255
ENBT-Kanal	Port-ID	Der zu verwendende Ethernet/IP-Port.	Alpha	A und B
Remote-IP-Adresse	Verknüpfungsadresse	Die Remote-IP-Adresse der Ziel-SPS.	Dezimalwert	0-255

**Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*

Beispiel
123.123.123.123,1,2.A.192.168.1.10

Dies entspricht einer lokalen IP von 123.123.123.123. Die Ethernet/IP-Karte befindet sich im Steckplatz 2: Verwenden Sie Port A und das Gerät mit IP 192.168.1.10 als Adressierungsziel.

Hinweise:

1. Informationen zum Ergänzen einer Geräte-ID mit einem Routing-Pfad zum Remote-Ethernet/IP-Gerät finden Sie unter Kommunikations-Routing.
2. Weitere Informationen zu Kommunikationsparametern finden Sie unter Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway.
3. Beim Konfigurieren der Geräte-ID sollten Benutzer überprüfen, ob das Gerät mit derselben Route durch RSLinx erkannt werden kann.

Serial Gateway-Setup

Das Serial Gateway ermöglicht die Kommunikation mit ControlLogix-, CompactLogix-, FlexLogix- und SoftLogix-SPS in einem seriellen Netzwerk mit ControlLogix-Ethernet-Treiber von Allen-Bradley.

Anforderungen

EtherNet/IP-Schnittstellenmodul.
Lokale CPU mit seriellem Port.
Remote-CPU von ControlLogix, CompactLogix, FlexLogix oder SoftLogix mit seriellem Port.

Hinweise:

1. Lokale und Remote-CPUs müssen sich in demselben seriellen Netzwerk befinden.
2. Serial Gateway-Modelle unterstützen die automatische Tag-Datenbankgenerierung nicht.

Geräte-ID für Serial Gateway

Die Geräte-ID gibt die IP-Adresse des lokalen Geräts sowie die Remote-Gerätestations-ID an, die zum Herstellen einer Verbindung erforderlich sind. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>,1,[<Optionaler Routing-Pfad>],<CPU-Steckplatz>.<Kanal des seriellen Ports>.<Stations-ID (Dez)>

Designator	Designator-Typ*	Description	Formate	Bereich
IP/Host-Name	k.A.	IP-Adresse oder Host-Name.	Dezimalwert	0-255
1	Port-ID	Port zu Hintergrundebene.	Dezimalwert	1
Optionaler Routing-	Mehrfachverknüpfung, Port-Paare	Legt einen Weg aus dem EtherNet/IP-Schnittstellenmodul fest und muss gleich 1	Dezimalwert	*

Designator	Designator-Typ*	Description	Formate	Bereich
Pfad		sein (Port zur Hintergrundebene).		
CPU-Steckplatz	Verknüpfungsadresse	Steckplatznummer des CPU-Moduls, das den zur Kommunikation verwendeten seriellen Port enthält.	Dezimalwert	0-255
Kanal des seriellen Ports		Zu verwendender Kanal des seriellen Ports.	Alpha	A und B
Stations-ID		DF1-Stations-ID der Ziel-SPS im Dezimalformat.**	Dezimalwert	0-255

**Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation.*

Beispiel
123.123.123.123,1,0.A.3

Dies entspricht einer Ethernet/IP von 123.123.123.123. Die CPU-Karte befindet sich im Steckplatz 0: Verwenden Sie Kanal A (serieller Port) und Stations-ID 3 (Dez) als Adressierungsziel.

Hinweise:

1. Informationen zum Ergänzen einer Geräte-ID mit einem Routing-Pfad zum seriellen Remote-Knoten finden Sie unter Kommunikations-Routing.
2. Weitere Informationen zu Kommunikationsparametern finden Sie unter Logix-Kommunikationsparameter.
3. Beim Konfigurieren der Geräte-ID sollten Benutzer überprüfen, ob das Gerät mit derselben Route durch RSLinx erkannt werden kann.

MicroLogix 1100-Setup

Geräte-ID für MicroLogix 1100

In der Geräte-ID wird die IP-Adresse für MicroLogix 1100 angegeben. Geräte-IDs werden wie folgt angegeben:

<IP oder Host-Name>

Designator	Designator-Typ	Description	Formate	Bereich
IP/Host-Name	k.A.	IP-Adresse oder Host-Name.	Dezimalwert	0-255

Beispiel
123.123.123.123

Dies entspricht einer IP von 123.123.123.123.

Weitere Informationen zu Kommunikationsparametern finden Sie unter Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway.

Kommunikations-Routing

Das Routing ermöglicht die Kommunikation mit einem Remote-Gerät über verschiedene Netzwerke. Es kann als Brücke zwischen dem lokalen Gerät und einem Remote-Gerät gedacht sein, selbst wenn sie sich auf zwei unterschiedlichen Feldbusnetzwerken befinden. Der Zugriff auf eine Remote-Hintergrundebene (Ziel) ermöglicht die direkte Kommunikation mit den unterstützten Modulen auf dieser Hintergrundebene. Zu den unterstützten Modulen gehören:

- ControlLogix 5500-Prozessor für ControlLogix-Anwendungen.
- SoftLogix 5800-Prozessor für SoftLogix-Anwendungen.

- 1756-DHRIO-Schnittstellenmodul für DH+-Gateway-Anwendungen.
- 1756-CNB- oder 1756-CNBR-Schnittstellenmodul für ControlNet-Gateway-Anwendungen.

Ein Routing-Pfad besteht aus einer Reihe von Hintergrundebenen-Hops, wobei die letzten Hops auf die Ziel-Hintergrundebene zeigen. Jeder Hop erfordert eine Logix-Hintergrundebene (keinen Logix-Prozessor). Ein einzelner Hop kann eines der folgenden Netzwerke als Medium verwenden:

- ControlNet
- DH+
- TCP/IP (Ethernet/IP)

Wichtig: Routing wird für ENI- und MicroLogix 1100-Modelle nicht unterstützt.

Verbindungspfadspezifikation

Der Routing-Pfad wird in der Geräte-ID angegeben. Wie bei Anwendungen ohne Routing stammt die Kommunikation von ControlLogix-Ethernet-Treiber von Allen-Bradley auf dem PC und wird an das lokale Ethernet-Modul geleitet. In diesem lokalen Ethernet-Modul legt die Geräte-ID einen Weg aus dem Modul und auf die Hintergrundebene fest, und zwar so wie bei Anwendungen ohne Routing. Über den Routing-Pfad wird die Meldung an die gewünschte Logix-Hintergrundebene geleitet. Die Geräte-ID bestimmt auch, mit welchem Gerät kommuniziert wird (z.B. dem ControlLogix-Prozessor, SoftLogix-Prozessor, DH+- oder ControlNet-Knoten).

Die Spezifikation des Routing-Pfads beginnt und endet jeweils mit der linken und rechten eckigen Klammer ([ ]). Der Pfad selbst ist eine Reihe von Port/Link-Adresspaaren und identisch mit der Kommunikationspfadsyntax im Dialogfenster für die Meldungskonfiguration von RSLogix 5000.

Designator-Typ	Description	Formate	Bereich
Port-ID	Legt einen Weg aus dem betreffenden Schnittstellenmodul fest.*	Dezimalwert	0-65535
Verknüpfungsadresse	Wenn der entsprechende Port die Hintergrundebene ist, handelt es sich bei der Verknüpfungsadresse um die Steckplatznummer des ausgehenden Schnittstellenmoduls. Wenn der entsprechende Port ein Port des Schnittstellenmoduls ist, gibt die Verknüpfungsadresse wie folgt einen Zielknoten an. - DH+/ControlNet: Knoten-ID - EtherNet/IP-Kommunikationsmodul: IP-Adresse - SoftLogix-EtherNet/IP-Modul: IP-Adresse	Dezimalwert	0-255

**Weitere Informationen finden Sie unter "Port-Referenz" unten.*

Single-Hop
IP Address, Port ID0, [Link Address0, Port ID1, Link Address1, Port ID2], Link Address2.

Multi-Hop (N-Hops)
IP Address, Port ID0, [Link Address0, Port ID1, Link Address1, Port ID2, Link Address2, ... Port ID(N+1), Link Address(N+1), Port ID(N+2)], Link Address(N+2).

Hinweise:

1. Die letzte Port-ID im Pfad (Port ID2 und Port ID(N+2) jeweils für Single-Hop und Multi-Hop) muss 1 (Port für Hintergrundebene) sein.
2. Port ID0 muss 1 (Port für Hintergrundebene) sein. Link Address2 und Link Address (N+2) sind die Steckplatznummern des Remote-Logix-Prozessors/1756-DHRIO-Moduls/1756-CNB-Moduls.

Port-Referenz

Schnittstellenmodul	Port 1	Port 2	Port 3
EtherNet/IP-Kommunikationsmodul	Hintergrundebene	Ethernet-Netzwerk	k.A.
SoftLogix-EtherNet/IP-Messaging-Modul	Virtuelle Hintergrundebene	Ethernet-Netzwerk	k.A.
1756-DHRIO	Hintergrundebene	DH+-Netzwerk auf Ka. A	DH+-Netzwerk auf Ka. B
1756-CNB	Hintergrundebene	ControlNet-Netzwerk	k.A.

Anwendungshinweise

1. Meldungen können mehr als einmal innerhalb des Pfads in oder aus dem Kanal des Schnittstellenmoduls geleitet werden. Dies führt zu CIP-Fehler 0x01 (ext. Fehler 0x100B).
2. Bei Mehrkanal-Schnittstellenmodulen können Meldungen nicht in dasselbe und anschließend unmittelbar aus demselben Modul (unter Verwendung verschiedener Kanäle) geleitet werden, unabhängig davon, ob die Meldung zuerst an die Hintergrundebene geleitet oder ob die Hintergrundebene vollständig vermieden wird. Wie zuvor erwähnt, wird Letzteres nicht unterstützt, da jeder Hop eine ControlLogix-Hintergrundebene benötigt. Eine Möglichkeit wäre es beispielsweise, eine DH+-Meldung über ein 1756-DHRIO-Schnittstellenmodul von einer DH+- (z.B. Kanal A von 1756-DHRIO) zur anderen DH+-Verknüpfung (z.B. Kanal B desselben 1756-DHRIO) zu leiten. Dies wird im Allgemeinen als Remote-DH+-Messaging bezeichnet und wird nicht unterstützt.

Routing-Beispiele

In den unten genannten Routing-Beispielen ist die gesamte Geräte-ID abzüglich der IP des lokalen 1756-ENBT enthalten. Die Geräte-ID bzw. der Routing-Pfad ist aus der Perspektive des lokalen 1756-ENBT-Moduls dargestellt. Hop-Beschreibungen liegen in folgender Form vor:

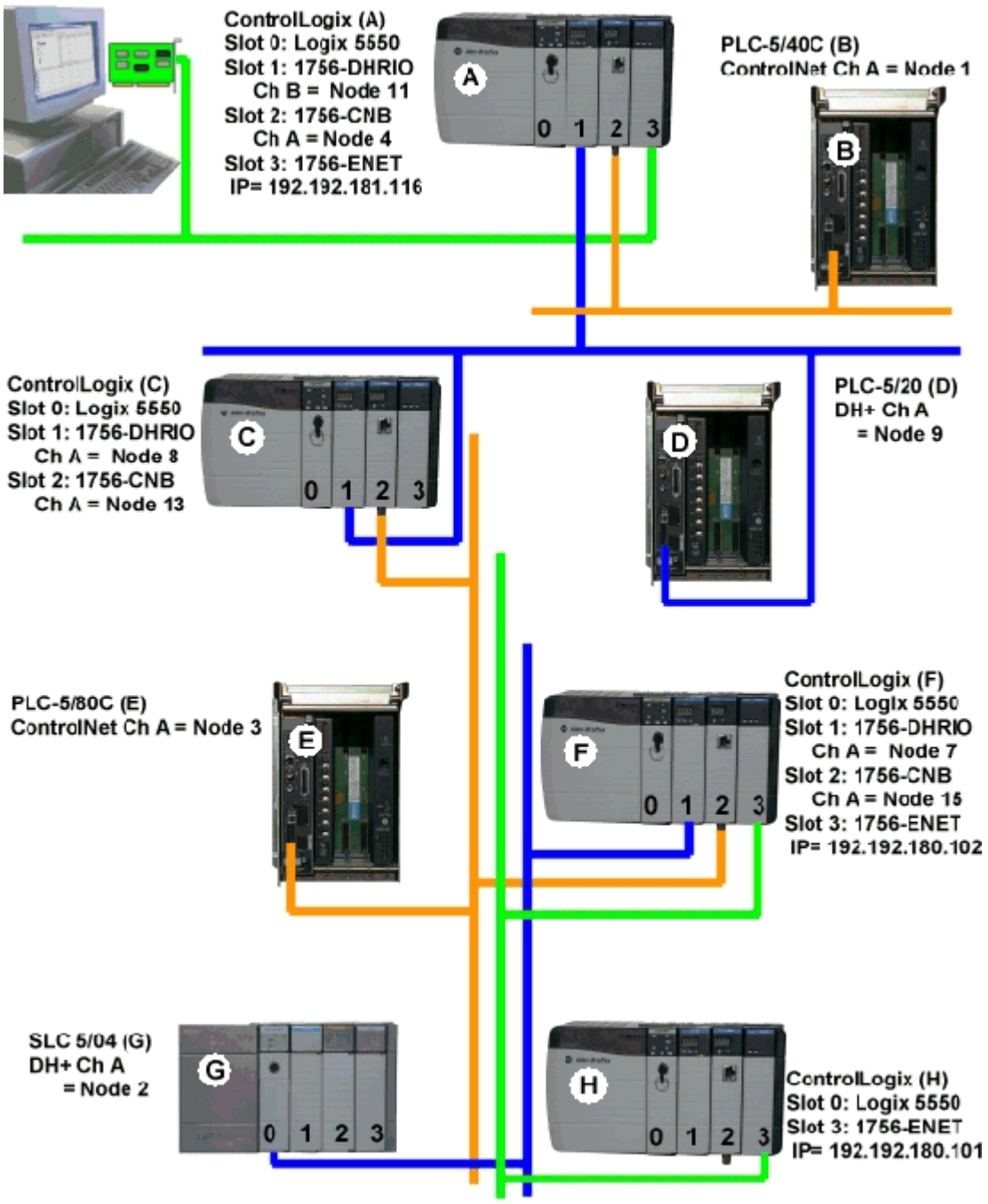
Verknüpfungsadresse (N), Port-ID(N+1), Verknüpfungsadresse(N+1), Port-ID(N+2)

Weitere Informationen dazu finden Sie unter Verbindungspfadspezifikation. Weitere Informationen zum Erstellen eines Verbindungs-/Routing-Pfads finden Sie in der Publikation 1756-6.5.14 von Allen-Bradley (S. 4-5 bis 4-8).

In der Abbildung unten sind alle DH+-/ControlNet-Knoten-IDs im Dezimalformat angegeben. Die in der SPS angegebene und in RSWho angezeigte Knoten-ID liegt im Oktalformat vor. Für die Farben gilt:

- Grün = Ethernet
- Blau = DH+
- Orange = ControlNet

Weitere Informationen dazu finden Sie unter Data Highway Plus-Gateway-Setup und ControlNet-Gateway-Setup.



Beispiel 1

Logix5550 zu PLC-5 über DH+-Gateway.

Zielknoten	Modell	Routing	Geräte-ID kleiner IP
PLC-5/20 (D)	DH+-Gateway	Nein	1,1.B.9

Beispiel 2

Logix5550 zu PLC-5C über CN-Gateway.

Zielknoten	Modell	Routing	Geräte-ID kleiner IP
PLC-5/40C (B)	CN-Gateway	Nein	1,2.A.1

Beispiel 3

Logix5550 zu Logix5550 bei Routing über DH+.

Zielknoten	Modell	Routing	Geräte-ID kleiner IP
Logix5550 (C)	ControlLogix 5550	Ja	1,[1,2,8,1],0

Aufgliederung des Routing-Pfads für Beispiel 3.

Hop	Segment	Beschreibung
1	1,2,8,1	Steckplatz 1 (DHRIO) -> Port 2 (DH+ Ch A) -> DH+-Knoten 8 -> Logix C-Hintergrundebene

Beispiel 4

Logix5550 zu PLC-5C über CN Gateway, Routing über DH+.

Zielknoten	Modell	Routing	Geräte-ID kleiner IP
PLC-5/80C (E)	CN-Gateway	Ja	1,[1,2,8,1],2.A.3

Aufgliederung des Routing-Pfads für Beispiel 4.

Hop	Segment	Beschreibung
1	1,2,8,1	Steckplatz 1 (DHRIO) -> Port 2 (DH+ Ch A) -> DH+-Knoten 8 -> Logix C-Hintergrundebene

Beispiel 5

Logix5550 zu Logix5550 bei Routing über DH+, ControlNet

Zielknoten	Modell	Routing	Geräte-ID kleiner IP
Logix5550 (F)	ControlLogix 5550	Ja	1,[1,2,8,1,2,2,15,1],0

Aufgliederung des Routing-Pfads für Beispiel 5.

Hop	Segment	Beschreibung
1	1,2,8,1	Steckplatz 1 (DHRIO) -> Port 2 (DH+ Ch A) -> DH+-Knoten 8 -> Logix C-Hintergrundebene
2	2,2,15,1	Steckplatz 2 (CNB) -> Port 2 (CN Ch A) -> CN-Knoten 15 -> Logix F-Hintergrundebene

Beispiel 6

Logix5550 zu SLC 5/04 bei Routing über DH+, ControlNet.

Zielknoten	Modell	Routing	Geräte-ID kleiner IP
SLC 5/04 (G)	DH+-Gateway	Ja	1,[1,2,8,1,2,2,15,1],1.A.2

Aufgliederung des Routing-Pfads für Beispiel 6.

Hop	Segment	Beschreibung
1	1,2,8,1	Steckplatz 1 (DHRIO) -> Port 2 (DH+ Ch A) -> DH+-Knoten 8 -> Logix C-Hintergrundebene
2	2,2,15,1	Steckplatz 2 (CNB) -> Port 2 (CN Ch A) -> CN-Knoten 15 -> Logix F-Hintergrundebene

Beispiel 7

Logix5550 zu Logix5550 bei Routing über DH+, ControlNet, Ethernet.

Zielknoten	Modell	Routing	Geräte-ID kleiner IP
Logix5550 (H)	ControlLogix 5550	Ja	1,[1,2,8,1,2,2,15,1,3,2,192.192.180.101,1],0

Aufgliederung des Routing-Pfads für Beispiel 7.

Hop	Segment	Beschreibung
1	1,2,8,1	Steckplatz 1 (DHRIO) -> Port 2 (DH+ Ch A) -> DH+-Knoten 8 -> Logix C-Hintergrundebene
2	2,2,15,1	Steckplatz 2 (CNB) -> Port 2 (CN Ch A) -> CN-Knoten 15 -> Logix F-Hintergrundebene
3	3,2,192.192.180.101,1	Steckplatz 3 (ENBT) -> Port 2 -> Remote-1756-ENBT-IP -> Logix H-Hintergrundebene

Protokollmodus auswählen

Symbolischer Modus

Im Modus "Symbolisch" wird jede Client-/Server-Tag-Adresse wird im Paket durch ihren ASCII-Zeichennamen dargestellt.

Vorteile	Nachteile
1. Alle für eine Datenanforderung benötigten Informationen befinden sich in der Client-/Server-Tag-Adresse. 2. Es werden lediglich die Daten, auf die in den Client-/Server-Tags zugegriffen wird, vom SPS angefordert. 3. Rückwärtskompatibel	1. Lange Geräte-Verarbeitungszeit bei der Verarbeitung von symbolischen Adressen 2. Weniger Anforderungen pro Mehrfachanforderungspaket, da die Größe der Anforderungen variiert

Hinweise:

1. Um Nutzen aus der Optimierung des Mehrfachanforderungspakets zu ziehen, sollten so viele Tags wie möglich in einem einzigen Paket repräsentiert sein. Da Tag-Adressen durch ihren ASCII-Zeichennamen im Paket dargestellt werden, sollten die Tag-Adressen so kurz wie möglich sein. Beispiel: "MeinTag" ist besser als "MeinSehrLangerTagNameMit36Zeichen".
2. Wenn die Eigenschaft "Standard-Datentyp" auf "Standard" festgelegt ist, werden bei der automatischen Tag-Generierung Tags mit einem Datentyp, der mit dem Typ im Controller übereinstimmt, erstellt.

Logische Modi

"Logisch, nicht blockierend" und "Logisch, blockierend" schließt zwei Leseprotokolle ein. Das zu verwendende Protokoll wird automatisch vom Treiber bestimmt und basiert auf der Controller-Revision. Die nachfolgende Tabelle enthält eine Übersicht über die Modi und die Protokolle, denen sie zugeordnet werden.

Protokollmodus	Verwendetes Leseprotokoll		Verwendetes Schreibprotokoll
	FRN V21 und höher	FRN V20 und niedriger	Alle FRN
Symbolisch	Symbolich (nicht blockierend)	Symbolich (nicht blockierend)	Symbolisch
Logisch, nicht blockierend	Symbolinstanz, nicht blockierend	Physisch, nicht blockierend*	Symbolinstanz
Logisch, blockierend	Symbolinstanz, blockierend	Physisch, blockierend*	Symbolinstanz

*Veraltet in V21.

Die für logische Leseoperationen notwendigen Informationen werden in einer Controller-Projekt-Upload-Sequenz abgerufen, die automatisch vom Treiber ausgeführt wird. Der Kürze wegen repräsentiert der Begriff "Logische Adresse" die Symbolinstanz-ID oder die physische Adresse, abhängig vom verwendeten Protokoll. Die logischen Modi vermeiden zeitaufwändige Analysen von Adressen und Suchvorgänge, die für jede symbolische Anforderung erforderlich sind.

Hinweis: Diese logischen Modi sind für Serial Gateway-Modelle nicht verfügbar.

Modus "Logisch, nicht blockierend"

Bei Verwendung des Modus "Logisch, nicht blockierend" werden alle Client-/Server-Tags individuell und mit einer festen Größe angefordert.

Vorteile	Nachteile
1. Enthält die maximalen Anforderungen pro Mehrfachanforderungspaket, da jede Anforderung eine feste Größe hat. 2. Geringe Geräte-Verarbeitungszeit, da die Client-/Server-Tags im Paket mit den logischen Adressen angegeben sind. 3. Es werden lediglich die Daten, auf die in den Client-/Server-Tags zugegriffen wird, vom SPS angefordert.	Initialisierungsmehraufwand beim Hochladen des Projekts, um die logischen Adressen zu bestimmen.

Hinweis: Dieser Modus wird bevorzugt, wenn die Minderheit der Struktur-Tags von einem Client/Server referenziert wird.

Modus "Logisch, blockierend"

Bei Verwendung des Modus "Logisch, blockierend" werden alle Daten für ein Logix-Tag in einer einzigen Anforderung, die von lediglich einem Client-/Server-Tag initiiert werden kann, abgerufen. Nach Erhalt des Datenblocks wird dieser im Cache im Treiber abgelegt und anschließend mit einem Zeitstempel versehen. Die nachfolgenden Client-/Server-Tags, die zu dem angegebenen Logix-Tag gehören, erhalten ihre Daten dann von diesem Cache. Sobald alle Tags aktualisiert wurden, wird eine neue Anforderung initiiert, vorausgesetzt der Cache ist nicht veraltet. Der Cache ist veraltet, wenn aktuelle Zeit > Cache-Zeitstempel + Tag-Scan-Intervall. Trifft dies zu, wird eine weitere Block-Anforderung an das Gerät gesendet, der Cache wird aktualisiert und der Zyklus wiederholt sich.

Vorteile	Nachteile
1. Inhalte werden bei jedem Lesevorgang abgerufen. 2. Geringe Geräte-Verarbeitungszeit, da die Client-/Server-Tags im Paket mit den logischen Adressen angegeben sind. 3. Enthält die maximalen Anforderungen pro Mehrfachanforderungspaket, da jede Anforderung eine feste Größe hat.	1. Initialisierungsmehraufwand beim Hochladen des Projekts, um die logischen Adressen zu bestimmen. 2. Wird die Minderheit der Logix-Tags referenziert, ist dieser Modus langsamer als der Modus "Logisch, nicht blockierend", da vom SPS auf mehr Daten zugegriffen wird als in den Client-/Server-Tags referenziert sind.

Hinweis: Diesem Modus ist der Vorzug zu geben, wenn die Mehrheit der Struktur-Tags von einem Client/Server referenziert wird.

Siehe auch: Leistungsstatistik und -optimierung

Symbolinstanz vs. physisches Protokoll

Symbolinstanz-Lesevorgänge sind CIP-Anforderungen, in denen die CIP-Instanz-ID verwendet wird, um ein natives Tag in einer Leseanforderung anzugeben. Wird der Modus "Nicht blockierend" verwendet, ist u.U. die CIP-Instanz-ID erforderlich, um den Pfad zu Strukturmitgliedern und Array-Elementen vollständig zu

qualifizieren. Beispiel: Die CIP-Instanz-ID steht für die Struktur und die CIP-Mitglieds-ID steht für das Mitglied innerhalb der Struktur. Da es das Hinzufügen der CIP-Mitglieds-ID erforderlich macht, ein Client-/Server-Tag vollständig zu qualifizieren, können die Anforderungen unterschiedlich groß sein. Eine tiefere Verschachtelung von Strukturen bedeutet, dass mehr CIP-Mitglieds-IDs erforderlich sind, um ein Tag anzugeben und ein einziges Paket enthält weniger Anforderungen. Symbolinstanz-Lesevorgänge wurden in FRN V21 eingeführt.

Physische Lesevorgänge sind CIP-Anforderungen, für die die DMA-Adresse verwendet wird, um ein natives Tag in einer Leseanforderung anzugeben. Wird der Modus "Nicht blockierend" verwendet, ist u.U. das Byte-Offset erforderlich, um den Pfad zu Strukturmitgliedern und Array-Elementen vollständig zu qualifizieren. Beispiel: Die DMA-Startadresse steht für die Struktur dar und das Byte-Offset steht für das Mitglied innerhalb der Struktur. Start + Offset ist die in der Anforderung angegebene DMA-Adresse: Alle Anforderungen haben eine feste Größe, im Gegensatz zu Symbolinstanz-Lesevorgängen. Unabhängig von der Verschachtelung der Strukturen, ist die Anforderung von gleicher Größe im Paket. Physische Lesevorgänge sind ab FRN V21 veraltet.

Änderung im Controller-Projekt erkennen

ControlLogix-Ethernet-Treiber von Allen-Bradley überwacht die Projektänderungen und kann erkennen, ob Downloads ausgeführt werden oder Bearbeitungen online oder offline vorgenommen werden. Ist das Protokoll auf logisch festgelegt, so haben Benutzer die Möglichkeit, das Projektbild des Treibers mit dem des Controller-Projekts zu synchronisieren. Durch die Synchronisierung wird sichergestellt, dass der Treiber die aktuelle, logische Adresse für jedes native Tag verwendet, wenn Lese- und Schreibvorgänge ausgeführt werden.

- **Downloads werden ausgeführt:** Der Treiber überwacht die Online- und Offline-Bearbeitungen in jeder Anforderung. Der Treiber erkennt, ob ein Download-Vorgang ausgeführt wird, während native Tags aktiv gelesen werden oder aktiv in sie geschrieben wird, und führt, abhängig von seinem Modus, ein Projektänderungs-Verfahren durch. Klicken Sie mit der rechten Maustaste auf das Gerät und wählen Sie **Eigenschaften…** aus, um diese Synchronisation zu aktivieren. Suchen Sie in der Gruppe **Logix-Optionen** entweder nach **Nach Online-Bearbeitungen synchronisieren** oder nach **Nach Offline-Bearbeitungen synchronisieren** und wählen Sie **Ja** aus.
- **Nach Online-Bearbeitungen synchronisieren**: Der Treiber überwacht die Online-Bearbeitungen in jeder Anforderung. Der Treiber erkennt, ob eine Online-Bearbeitung erfolgt für den Controller für den folgenden Lese- oder Schreibvorgang, und führt, abhängig von seinem Modus, ein Projektänderungs-Verfahren durch. Klicken Sie mit der rechten Maustaste auf das Gerät und wählen Sie **Eigenschaften…** aus, um diese Synchronisation zu aktivieren. Suchen Sie in der Gruppe **Logix-Optionen** nach **Nach Online-Bearbeitungen synchronisieren** und wählen Sie **Ja** aus.
- **Nach Offline-Bearbeitungen synchronisieren**: Der Treiber überwacht die Offline-Bearbeitungen in jeder Anforderung. Der Treiber erkennt, ob eine Offline-Bearbeitung erfolgt für den Controller für den folgenden Lese- oder Schreibvorgang, und führt, abhängig von seinem Modus, ein Projektänderungs-Verfahren durch. Klicken Sie mit der rechten Maustaste auf das Gerät und wählen Sie **Eigenschaften…** aus, um diese Synchronisation zu aktivieren. Suchen Sie in der Gruppe **Logix-Optionen** nach **Nach Offline-Bearbeitungen synchronisieren** und wählen Sie **Ja** aus.

Projektänderungs-Verfahren (symbolischer Modus)

1. Es wird eine Projektänderung erkannt.
2. Es wird eine Nachricht im Ereignisprotokoll veröffentlicht, die angibt, dass eine Änderung erkannt wurde.
3. Während der Projektänderung ist dies das Szenario für Downloads:
    - Alle Lese- und Schreibvorgänge, die derzeit ausgeführt werden, werden angehalten und schlagen fehl.
    - Der Controller wird alle 2 Sekunden abgerufen, um zu überwachen, ob die Projektänderung abgeschlossen wurde.
    - Die Projektänderung wird nicht länger erkannt.
    - Es wird eine Nachricht im Ereignisprotokoll veröffentlicht, die angibt, dass keine Änderung mehr erkannt wird.

4. Während der Projektänderung ist dies das Szenario für Online- und Offline-Bearbeitungen:
    - Die Antwortdaten werden ignoriert.
    - Alle Lese- und Schreibvorgänge, die derzeit ausgeführt werden, werden deaktiviert.
5. Die Lese- und Schreibvorgänge werden unter Verwendung des symbolischen Modus wiederaufgenommen.

Projektänderungs-Verfahren (logische Modi)

1. Es wird eine Projektänderung erkannt.
2. Es wird eine Nachricht im Ereignisprotokoll veröffentlicht, die angibt, dass eine Änderung erkannt wurde.
3. Während der Projektänderung ist dies das Szenario für Downloads:
    - Alle Lese- und Schreibvorgänge, die derzeit ausgeführt werden, werden angehalten und schlagen fehl.
    - Der Controller wird alle 2 Sekunden abgerufen, um zu überwachen, ob die Projektänderung abgeschlossen wurde.
    - Die Projektänderung wird nicht länger erkannt.
    - Es wird eine Nachricht im Ereignisprotokoll veröffentlicht, die angibt, dass die Änderung nicht länger erkannt wird.
4. Während der Projektänderung ist dies das Szenario für Online- und Offline-Bearbeitungen:
    - Die Antwortdaten werden ignoriert.
    - Alle Lese- und Schreibvorgänge, die derzeit ausgeführt werden, werden deaktiviert.
5. Die Lese- und Schreibvorgänge werden unter Verwendung des symbolischen Modus wiederaufgenommen.
6. Bei Aktivierung der Optionen für die Synchronisierung mit dem Controller:
    - Nach 30 Sekunden im symbolischen Modus, lädt der Treiber das Projekt vom Controller hoch, d.h. es findet eine Synchronisierung statt.
    - Die Lese- und Schreibvorgänge werden unter Verwendung des logischen Modus mit neuen logischen Adressen wiederaufgenommen.
7. Sind die Optionen für die Synchronisierung mit dem Controller deaktiviert, so werden die Lese- und Schreibvorgänge unter Verwendung des logischen Modus mit den alten logischen Adressen wiederaufgenommen.

Hinweise zur SoftLogix 5800-Verbindung

Damit SoftLogix ordnungsgemäß funktioniert, sollten keine Ethernet-basierten Treiber (z.B. Ethernet-Geräte, Remote-Geräte via Gateway, etc.) in RSLinx auf dem SoftLogix Rechner installiert sein. Sind ein oder mehrere Ethernet-basierte Treiber installiert, werden Anfragen mit CIP-Fehler 0x5, Ext.-Fehler 0x1 und CIP-Fehler 0x8 zurückgegeben.

Verbindung zu einer SoftLogix Soft PLC auf demselben Rechner wie der OPC-Server

Befolgen Sie die nachfolgenden Anweisungen, um ControlLogix-Ethernet-Treiber von Allen-Bradley mit einer SoftLogix Soft PLC, die auf demselben Rechner wie der Server ausgeführt wird, zu verbinden.

1. Stellen Sie sicher, dass derzeit keine Ethernet-basierten Treiber in RSLinx auf dem Rechner ausgeführt werden.

2. Vergewissern Sie sich, dass das **EtherNet/IP Message Module** in der SoftLogix virtuellen Chassis installiert ist.
3. Suchen Sie in der Gruppe **Geräteigenschaften**| **Allgemein** nach dem Wert für die Geräte-ID. Der Wert sollte nicht "127.0.0.1, 1, <PLC_CPU_Steckplatz>" sein. Die Geräte-ID sollte auf "<Spezifische_PC_Adresse>, 1, <PLC_CPU_Steckplatz>" festgelegt sein.

   Beispiel: Die IP-Adresse des Rechners lautet 192.168.3.4 und die SoftLogix CPU befindet sich in Steckplatz 2 der virtuellen Chassis, so ist die korrekte Geräte-ID "192.168.3.4, 1, 2".

Index

-

---------- 145

%

%s | GERÄTESTATISTIK 145
%s | KANALSTATISTIK 145

0

0000-Generic Module 29
0x0001 Erweiterte Fehlercodes 113
0x001F Erweiterte Fehlercodes 114
0x00FF Erweiterte Fehlercodes 114
0x01 112
0x02 112
0x03 112
0x04 112
0x05 112
0x06 112
0x07 112
0x08 112
0x09 112
0x0A 112
0x0B 112
0x0C 112
0x0D 112
0x0E 112
0x0F 112
0x10 112
0x11 112
0x12 112
0x13 112
0x14 112
0x15 112
0x1A 112
0x1B 112
0x1C 112
0x1D 113
0x1E 113

0x1F 113
0x22 113
0x25 113
0x26 113
0x27 113

1

1761-NET-ENI 151

A

Adressbeschreibungen 48
Adressformate 57
Adressieren unteilbarer Datentypen 59
Adressierung der PLC-5-Reihe 54
Adressierung der PLC-5-Reihe für ControlNet 54
Adressierung der PLC-5-Reihe für EtherNet/IP-Gateway 55
Adressierung von Struktur-Tags 58
Änderung im Controller-Projekt erkennen 163
Anfangsaktualisierungen aus Cache 19
Anforderungs-Timeout 20
Anforderungsgröße 27
Anfragen verwerfen, wenn herabgestuft 21
Angeforderte CIP-Verbindungsgröße wird von diesem Gerät nicht unterstützt. Es wird automatisch auf die maximale Größe ausgewichen. | Angeforderte Größe = <Anzahl> (Byte), max. Größe = <Anzahl> (Byte). 141
Anhänge 148
Anwendung optimieren 34
Array-Anzahl-Grenze 27
Array-Blockgröße 24
Array-Grenze festlegen 26
Array-Tags 57, 107
ASCII-Dateien 95
Auf Logix-Tags basierte Adressierung 56
Ausgabedateien 84
Ausgabewörter 29
Automatische Herabstufung 20
Automatische Tag-Datenbankgenerierung 106
Automatische Tag-Datenbankgenerierung wird vorbereitet 110

B

BCD 47

BCD-Dateien 96

Bedingungen für Standard-Datentyp 48

Bei doppeltem Tag 22

Bei Eigenschaftsänderung 22

Bei Gerätestart 22

Beim Abrufen des Controller-Projekts sind Fehler aufgetreten. 146

Beispiel für Leistungsoptimierung 36

Beschreibungen anzeigen 26

Betriebsmodus 18

Binäre Dateien 91

Block-Leseanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Blockgröße = <Anzahl> (Byte), Blockname = '<Name>'. 124

Block-Leseanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 123

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Byte), Tag-Name = '<Tag>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 125

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 124

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>. 133

Block kann nicht gelesen werden. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>. 136

Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Byte), Tag-Name = '<Tag>'. 131

Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 138

Block kann nicht gelesen werden. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 130

Block kann nicht gelesen werden. Block unterstützt keine Arrays mit mehreren Elementen. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 128

Block kann nicht gelesen werden. Controller-Tag-Datentyp ist unbekannt. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = <Typ>. 125

Block kann nicht gelesen werden. Datentyp für diesen Block ist unzulässig. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = '<Typ>'. 127

Block kann nicht gelesen werden. Datentyp wird nicht unterstützt. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>', Datentyp = '<Typ>'. 126

Block kann nicht gelesen werden. Empfangener Frame enthält Fehler. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>'. 132

Block kann nicht gelesen werden. Interner Speicher ist ungültig. Block deaktiviert. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 138

Block kann nicht gelesen werden. Interner Speicher ist ungültig. Tag deaktiviert. | Tag-Adresse = '<Adresse>'. 138

Block kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Blockgröße = <Anzahl> (Byte), Blockname = '<Name>'. 129

Block kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Blockgröße = <Anzahl> (Elemente), Blockstartadresse = '<Adresse>'. 129

Block kann nicht gelesen werden. Tags deaktiviert. | Blockgröße = <Anzahl> (Elemente), Startadresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>. 132, 134

Blocktransferdateien 102

Boolean 47
Byte 47

C

Char 47
CIP-Fehlercodes 112
CompactLogix 5300-Adressierung für ENI 49
CompactLogix 5300-Adressierung für Ethernet 49
CompactLogix 5300-Adressierung für Serial Gateway 50
Controller wird nicht unterstützt. | Händler-ID = <ID>, Produkttyp = <Typ>, Produktcode = <Code>, Produktname = '<Name>'. 122
ControlLogix-Datenbankeinstellungen 25
ControlLogix-Kommunikationsparameter 23
ControlLogix-Optionen 24
ControlLogix 5000-Adressierung 56
ControlLogix 5500-Adressierung für ENI 49
ControlLogix 5500-Adressierung für Ethernet 49
ControlLogix 5500-Adressierung für Serial Gateway 49
ControlLogix 5500-Ethernet 150
ControlNet-Gateway 153

D

DataHighwayPlus(TM)-Gateway-Setup 152
Dateiauflistung 83
Datenbank-Importmethode 25
Datenbankfehler. Array-Tags umbenannt, da sie maximale Zeichenlänge überschreiten. | Array-Tags = '<Tags>', max. Länge = <Anzahl>, neue Array-Tags = '<Tags>'. 142
Datenbankfehler. CIP-Datentyp für Tag konnte nicht aufgelöst werden. Standardtyp wird verwendet. | CIP-Datentyp = <Typ>, Tag-Name = '<Tag>', Standard-Datentyp = '<Typ>'. 119
Datenbankfehler. Datentyp für Mitglied in Tag-Importdatei wurde nicht gefunden. Datentyp wird auf Standardwert gesetzt. | Datentyp für Mitglied = '<Typ>', UDT = '<Typ>', Standard-Datentyp '<Typ>'. 116
Datenbankfehler. Datentyp für Referenz-Tag ist unbekannt. Datentyp für Alias-Tag wird auf den Standardwert gesetzt. | Referenz-Tag = '<Tag>', Alias-Tag = '<Tag>', Standard-Datentyp = '<Typ>'. 115
Datenbankfehler. Datentyp wurde in Tag-Importdatei nicht gefunden. Tag nicht hinzugefügt. | Datentyp = '<Typ>', Tag-Name = '<Tag>'. 116
Datenbankfehler. Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 119
Datenbankfehler. Fehler beim Hochladen der Projektinformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 118
Datenbankfehler. Fehler beim Verarbeiten des Alias-Tags. Tag wurde nicht hinzugefügt. | Alias-Tag = '<Tag>'. 116
Datenbankfehler. Fehler beim Weiterleiten der offenen Anforderung. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 117
Datenbankfehler. Framing-Fehler beim Hochladen der Programminformationen. | Programmname =

'<Name>'. 119

Datenbankfehler. Framing-Fehler beim Hochladen der Projektinformationen. 118

Datenbankfehler. Framing-Fehler beim Registrieren der Sitzungsanforderung. 117

Datenbankfehler. Framing-Fehler beim Weiterleiten der offenen Anforderung. 117

Datenbankfehler. Interner Fehler aufgetreten. 118

Datenbankfehler. Kapselungsfehler beim Hochladen der Programminformationen. | Programmname = '<Name>', Kapselungsfehler = <Code>. 118

Datenbankfehler. Kapselungsfehler beim Hochladen der Projektinformationen. | Kapselungsfehler = <Code>. 117

Datenbankfehler. Kapselungsfehler beim Registrieren der Sitzungsanforderung. | Kapselungsfehler = <Code>. 117

Datenbankfehler. Kapselungsfehler beim Weiterleiten der Anforderung zum Öffnen. | Kapselungsfehler = <Code>. 117

Datenbankfehler. Keine Verbindungen mehr verfügbar zum Weiterleiten offener Anforderungen. 122

Datenbankfehler. PLC5/SLC/MicroLogix-Geräte unterstützen diese Funktion nicht. 146

Datenbankfehler. Programmgruppenname überschreitet maximale Zeichenlänge. Programmgruppe umbenannt. | Gruppenname = '<Name>', max. Länge = <Anzahl>, neuer Gruppenname = '<Name>'. 142

Datenbankfehler. Tag umbenannt, da maximale Zeichenlänge überschritten wurde. | Tag-Name = '<Tag>', max. Länge = <Anzahl>, neuer Tag-Name = '<Tag>'. 142

Datenbankfehler. Timeout von CIP-Verbindung, während Projektinformationen hochgeladen werden. 122

Datenbankstatus. | Programmanzahl = <Anzahl>, Datentypanzahl = <Anzahl>, Anzahl importierter Tags = <Anzahl>. 143

Datenbankstatus. Controller-Projekt wird abgerufen. 143

Datenbankstatus. OPC-Tags werden generiert. 143

Datenbankstatus. Tag-Projekte werden erstellt. Bitte warten. | Anzahl der Tag-Projekte = <Anzahl>. 142

Datenbankstatus. Tags mit Alias werden importiert. 142

Datenbankstatus. Tags ohne Alias werden importiert. 142

Datenbankstatus. Tags werden aus L5X-Datei importiert. | Schemarevision = '<Wert>', Softwarerevision = '<Wert>'. 143

Datensammlung 18

Datentypbeschreibung 47

Datum 47

Der Dateiname für den Tag-Import ist ungültig. Dateipfade sind nicht erlaubt. 141

Der vom Gerät empfangene Frame enthält Fehler. 122

Details. | IP = '<Adresse>', Händler-ID = <Händler>, Produkttyp = <Typ>, Produktcode = <Code>, Revision= <Wert>, Produktname = '<Name>', Produkt-Seriennummer = <Anzahl>. 143

Die folgenden Fehler sind beim Hochladen des Controller-Projekts vom Gerät aufgetreten. Das symbolische Protokoll wird stattdessen verwendet. 115, 140

Double 47

Durch Tag angegebenes Scan-Intervall berücksichtigen 19

Durchschnittl. Lesevorgänge/Sekunde für Tag = <Anzahl>. 145

Durchschnittl. Tags/Transaktion = <Anzahl>. 145

Durchschnittliche Verarbeitungszeit für Gerät = <Anzahl> (Millisekunden) 145

DWord 47

E

Eingabedateien 87
Eingabewörter 29
Elternteilgruppe 23
ENI-Geräte-ID 152
Ereignisprotokollmeldungen 115
Erstellen 23
Erweiterte Adressierung: BOOL 64
Erweiterte Adressierung: INT 67
Erweiterte Adressierung: LINT 72
Erweiterte Adressierung: REAL 73
Erweiterte Adressierung: SINT 65
Erweiterte Adressierung: UDINT 79
Erweiterte Adressierung: UINT 77
Erweiterte Adressierung: ULINT 81
Erweiterte Adressierung: USINT 75
Erweiterte Adressierung:DINT 69
Erweiterte Adressierung:LREAL 82
Erweiterte Logix-Adressierung 63
EtherNet/IP-Gateway-Setup 154

F

Fehler bei XML-Element in Post-Schema-Validierung. Modell unterstützt kein Importieren der Tags von Gerät. Verwenden Sie ein alternatives Element. | XML-Element = '{<Namespace>}<Element>', nicht unterstütztes Modell = '<Modell>', alternatives XML-Element = '{<Namespace>}<Element>'. 147
Fehler beim Hochladen der Controller-Programminformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 121
Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 121
Fehler beim Hochladen der Projektinformationen. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 120
Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Code>'. 122
Fehler während Anfrage an Gerät. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 131
Fehlercodes 112
Feste E/A-Adressierung für SLC 500 52
Feste E/A-Adressierung für SLC 500 für EtherNet/IP-Gateway 52
Feste E/A-Adressierung SLC 500 für ENI 53
Filterung 26
FlexLogix 5400-Adressierung für ENI 50
FlexLogix 5400-Adressierung für Ethernet 50
FlexLogix 5400-Adressierung für Serial Gateway 50
Float 47, 94
Float-Dateien 94
Framing-Fehler beim Hochladen der Controller-Programminformationen. 121

Framing-Fehler beim Hochladen der Programminformationen. | Programmname = '<Name>'. 121

Framing-Fehler beim Hochladen der Projektinformationen. 120

Funktionsdatei-Blockschreibvorgänge zulassen 27

Funktionsdatei kann nicht gelesen werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>. 134

Funktionsdatei kann nicht gelesen werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 136

Funktionsdatei kann nicht gelesen werden. Tags deaktiviert. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 135

Funktionsdatei kann nicht von Gerät gelesen werden. Empfangener Frame enthält Fehler. | Funktionsdatei = '<Name>'. 132

Funktionsdatei kann nicht von Gerät gelesen werden. Tags deaktiviert. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>. 133

Funktionsdateien 102

G

Ganzzahldateien 94

Generieren 22

Gerät antwortet nicht. Lokaler Knoten hat mit Fehler geantwortet. | DF1-Status = <Code>. 138

Geräte-ID für den ControlNet-Gateway 153

Geräte-ID für DH+-Gateway 152, 154

Geräte-ID für MicroLogix 1100 156

Geräte-ID für Serial Gateway 155

Geräteeigenschaften - Tag-Generierung 21

Globale Tags 58

H

Herabstufen bei Fehler 21

Herabstufungszeitraum 21

Herunterladen von Projekt abgeschlossen. 140

Herunterladen von Projekt in Bearbeitung oder kein Projekt vorhanden. 140

Hinweise zur SoftLogix Soft PLC-Verbindung 164

HSC-Datei (High Speed Counter, Hochgeschwindigkeitszähler) 103

I

ID 18

ID für Gerät kann nicht abgerufen werden. Alle Tags verwenden stattdessen das symbolische Protokoll. | CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 140

ID für Gerät kann nicht abgerufen werden. Alle Tags verwenden stattdessen das symbolische Protokoll. | Kapselungsfehler = <Code>. 140

ID für Gerät kann nicht abgerufen werden. Empfangener Frame enthält Fehler. Alle Tags verwenden stattdessen das symbolische Protokoll. 141

Identifikation 18

Import-Datei '%s' nicht gefunden unter Pfad '<empty>'. 147

Import-Datei '<empty>' nicht gefunden unter Pfad '%s'. 147

Import-Datei '<empty>' nicht gefunden unter Pfad '<empty>'. 146

Importdatei '%s' nicht gefunden unter Pfad '%s'. 146

Importieren von Geräte-Tag abgebrochen. 146

In Adresse kann nicht geschrieben werden. | Adresse = '<Adresse>', DF1-Status = <Code>, erweiterter Status = <Code>. 135

In Adresse kann nicht geschrieben werden. | Adresse = '<Adresse>', DF1-Status = <Code>. 137

In Adresse kann nicht geschrieben werden. Empfangener Frame enthält Fehler. | Adresse = '<Adresse>'. 133

In Adresse kann nicht geschrieben werden. Interner Speicher ist ungültig. | Tag-Adresse = '<Adresse>'. 138

In Adresse kann nicht geschrieben werden. Lokaler Knoten hat mit Fehler geantwortet. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 139

In Funktionsdatei kann nicht geschrieben werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>, erweiterter Status = <Code>. 135

In Funktionsdatei kann nicht geschrieben werden. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 137

In Funktionsdatei kann nicht geschrieben werden. Empfangener Frame enthält Fehler. | Funktionsdatei = '<Name>'. 133

In Funktionsdatei kann nicht geschrieben werden. Lokaler Knoten hat mit Fehler geantwortet. | Funktionsdatei = '<Name>', DF1-Status = <Code>. 139

In Tag kann nicht geschrieben werden. | Tag-Adresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 124

In Tag kann nicht geschrieben werden. | Tag-Adresse = '<Adresse>'. 129

In Tag kann nicht geschrieben werden. Controller-Tag-Datentyp ist unbekannt. | Tag-Adresse = '<Adresse>', Datentyp = <Typ>. 125

In Tag kann nicht geschrieben werden. Datentyp für dieses Tag ist unzulässig. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 126

In Tag kann nicht geschrieben werden. Datentyp wird nicht unterstützt. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 126

In Tag kann nicht geschrieben werden. Keine Größenübereinstimmung bei nativem Tag. | Tag-Adresse = '<Adresse>'. 128

In Tag kann nicht geschrieben werden. Tag unterstützt keine Arrays mit mehreren Elementen. | Tag-Adresse = '<Adresse>'. 127

Inaktivitätsüberwachung 23

Inhalt der Hilfe 10

Initialisierungstransaktionen = <Anzahl>. 145

Interne Tags 59

Interner Treiberfehler aufgetreten. 146

IOS-Datei (I/O Module Status, E/A-Modulstatus) 106

K

Kanalzuweisung 18

Kapselungsfehler beim Hochladen der Controller-Programminformationen. | Kapselungsfehler = <Code>. 121

Kapselungsfehler beim Hochladen der Programminformationen. | Programmname = '<Name>', Kapselungsfehler = <Code>. 121

Kapselungsfehler beim Hochladen der Projektinformationen. | Kapselungsfehler = <Code>. 120

Kapselungsfehler während Anfrage an Gerät. | Kapselungsfehler = <Code>. 131
Kapselungsfehlercodes 112
Kommunikation optimieren 32
Kommunikations-Routing 156
Kommunikations-Timeouts 20
Kommunikationsparameter für ENI DF1-/DH+-/ControlNet-Gateway 27
Kommunikationsprotokoll 13
Kommunikationsstatusdatei für Kanal 0 (Channel 0 Communication Status File) 105
Kommunikationsstatusdatei für Kanal 1 (Channel 1 Communication Status File) 105

L

L5K-Datei ist ungültig oder fehlerhaft. 146
L5X-Datei ist ungültig oder fehlerhaft. 146
lange Controller-Programm- und Tag-Namen 106
LBCD 47
Leistungsoptimierungen 32
Leistungsstatistik 25
Leistungsstatistik und -optimierung 35
Lese-/Schreibanfragen an Gerät beendet. Aktualisierung logischer Adressen vom Geräteprojekt läuft. 142
Lese-/Schreibanfragen an Gerät wurden fortgesetzt. Aktualisierung logischer Adressen vom Geräteprojekt abgeschlossen. Zurzeit wird logische Adressierung verwendet. 142
Lese-/Schreibtransaktionen = <Anzahl>. 145
Leseanforderung für Tag ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Tag-Adresse = '<Adresse>'. 123
Lesevorgänge für Tags = <Anzahl>. 144
Logix-Adressierung 49, 56
Logix-Datenbankeinstellungen 25
Logix-Geräte-IDs 150
Logix-K 23
Logix-Optionen 24
Long 47
Long-Dateien 97
Löschen 22

M

Meldungsdateien 101
MicroLogix-Adressierung 50
MicroLogix-Adressierung für ENI 51
MicroLogix-Adressierung für EtherNet/IP-Gateway 50
MicroLogix-Meldungsdateien 100
MicroLogix-PID-Dateien 97
MicroLogix 1100-Setup 156

Modell 18
Modul 29
Modulare E/A-Adressierung für SLC 500 53
Modulare E/A-Adressierung für SLC 500 für DH+ 53
Modulare E/A-Adressierung für SLC 500 für ENI 54
Modulare E/A-Adressierung für SLC 500 für EtherNet/IP-Gateway 53

N

Nach Offline-Bearbeitungen synchronisieren 24
Nach Online-Bearbeitungen synchronisieren 24
Namenlänge begrenzen 26
Namenskonvertierungen von Controller in Server 109
Nicht scannen, nur Abruf anfordern 19

O

Offline-Bearbeitung des Projekts festgestellt. Zurzeit wird symbolische Adressierung verwendet. 140
Online-Bearbeitung des Projekts festgestellt. Zurzeit wird symbolische Adressierung verwendet. 140

P

Pakete durchschnittl. empfangen/Sekunde = <Anzahl>. 145
Pakete durchschnittl. gesendet/Sekunde = <Anzahl>. 145
Pakete empfangen = <Anzahl>. 145
Pakete gesendet = <Anzahl>. 144
Physisch blockierend - Lesevorgänge von Cache = <Anzahl>. 144
Physisch blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Physisch nicht blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Physisch nicht blockierend, Array-Block - Lesevorgänge von Cache = <Anzahl>. 144
Physisch nicht blockierend, Array-Block - Lesevorgänge von Gerät = <Anzahl>. 144
PID-Dateien 98
Port-ID 157
Programm-Tags 58
Projektoptionen 25
Protokoll 24
Protokollmodus 24
Protokollmodus auswählen 161

R

Redundanz 30
Reihenfolge von Logix-Array-Daten 62

Richtlinien für modulare E/A-Auswahl für SLC 500 30
Routing-Beispiele 158
RTC-Datei (Real Time Clock, Echtzeituhr) 104

S

Scan-Modus 19
Schreibanforderung ist aufgrund eines Framing-Fehlers fehlgeschlagen. | Tag-Adresse = '<Adresse>'. 122
Serial Gateway-Setup 155
Setup 12
Short 47
Simuliert 18
SLC 500-Steckplatzkonfiguration 28
SoftLogix-Datenbankeinstellungen 25
SoftLogix-Kommunikationsparameter 23
SoftLogix-Optionen 24
SoftLogix 5800-Adressierung 50
SoftLogix 5800-Adressierung für Serial Gateway 50
Speicherplatz für Tag konnte nicht zugeordnet werden. | Tag-Adresse = '<Adresse>'. 132
Statistik 35
Statistiktyp 35
Statusdateien 90
Steckplatz 29
Steuerdateien 93
String-Dateien 96
STRING-Datentyp adressieren 61
Strukturdatentypen adressieren 61
Symbolinstanz blockierend - Lesevorgänge von Cache = <Anzahl>. 144
Symbolinstanz blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Symbolinstanz nicht blockierend - Lesevorgänge von Gerät = <Anzahl>. 144
Symbolinstanz nicht blockierend, Array-Block - Lesevorgänge von Cache = <Anzahl>. 144
Symbolinstanz nicht blockierend, Array-Block - Lesevorgänge von Gerät = <Anzahl>. 144
Symbolisch - Lesevorgänge von Gerät = <Anzahl>. 143
Symbolisch, Array-Block - Lesevorgänge von Cache = <Anzahl>. 143
Symbolisch, Array-Block - Lesevorgänge von Gerät = <Anzahl>. 143

T

Tag-Generierung 21
Tag-Hierarchie 106
Tag-Importdatei 25
Tag-Umfang 58
Tag kann nicht gelesen werden. | Tag-Adresse = '<Adresse>', CIP-Fehler = <Code>, erweiterter Fehler = <Code>. 124

Tag kann nicht gelesen werden. Controller-Tag-Datentyp ist unbekannt. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = <Typ>. 125

Tag kann nicht gelesen werden. Datentyp für dieses Tag ist unzulässig. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 138

Tag kann nicht gelesen werden. Datentyp für dieses Tag ist unzulässig. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 127

Tag kann nicht gelesen werden. Datentyp wird nicht unterstützt. Tag deaktiviert. | Tag-Adresse = '<Adresse>', Datentyp = '<Typ>'. 126

Tag kann nicht gelesen werden. Interner Speicher ist ungültig. | Tag-Adresse = '<Adresse>'. 137

Tag kann nicht gelesen werden. Keine Größenübereinstimmung bei nativem Tag. | Tag-Adresse = '<Adresse>'. 128

Tag kann nicht gelesen werden. Tag deaktiviert. | Tag-Adresse = '<Adresse>'. 130

Tag kann nicht gelesen werden. Tag unterstützt keine Arrays mit mehreren Elementen. Tag deaktiviert. | Tag-Adresse = '<Adresse>'. 128

TCP/IP-Port 23, 27

Timeout bis zum Herabstufen 21

Timeout von CIP-Verbindung, während Projektinformationen hochgeladen werden. 121

Treiber 18

TREIBERSTATISTIK 145

U

Überschreiben 22

Übersicht 11

Unbekannter Fehler aufgetreten. 143, 146

Unerwarteter Offset bzw. Spanne für Tag festgestellt. | Tag-Adresse = '<Adresse>'. 140

Unerwarteter Offset für Tag festgestellt. | Tag-Adresse = '<Adresse>'. 139

Unerwarteter Offset für Tag festgestellt. Tag verwendet stattdessen symbolisches Protokoll. | Tag-Adresse = '<Adresse>'. 139

Ungültiges oder fehlerhaftes Controller-Projekt wurde beim Synchronisieren festgestellt. Die Synchronisierung wird in Kürze erneut versucht. 115

Ungültiges oder fehlerhaftes Controller-Projekt wurde beim Synchronisieren festgestellt. Versuchen Sie es später erneut. 146

Untergruppen zulassen 23

Unterstützte Geräte 12

V

Verbindungs-Timeout 20

Verbindungsgröße 23

Verbindungspfadspezifikation 157

Verknüpfungsadresse 157

Verstrichene Zeit = <Anzahl> (Sekunden). 143

Versuche vor Timeout 20

Verzögerung zwischen Anfragen 20

Von Gerät erstellen 25

Von Importdatei erstellen 25
Vorangestellte Unterstriche 109
Vordefinierte Ausdrucks-Tags 59

W

Während der Synchronisierung wurde ein Herunterladen des Projekts festgestellt. Die Synchronisierung wird in Kürze erneut versucht. 115
Während der Synchronisierung wurde ein Herunterladen des Projekts festgestellt. Versuchen Sie es später erneut. 146
Wert wird für ein XML-Element in diesem Modell nicht unterstützt. Ein neuer Wert wird automatisch festgelegt. | Wert = '<Wert>', XML-Element = '{<Namespace>}<Element>', Modell = '<Modell>', neuer Wert = '<Wert>'. 147
Word 47

Z

Zählerdateien 92
Zeichenfolge 47
Zeichenfolgendaten bei LEN beenden 24
Zeitgeberdateien 92
Zu wenig Speicherressourcen. 143, 146