



Versionshistorie

Getestet? Du willst es?

[Lizenzmodell](#) [Preise](#) [Angebot](#) [Jetzt bestellen](#)

Historie

2.5.0.0 (freigegeben am 23.01.2024)

NEU

- Verbesserte Evaluierungserfahrung.
- DnsDeviceEndPoint kann mit IP-Adress-Strings verwendet werden und kann auch mit IPv6-Hostnamen (für .NET Core 3.0 und höher) durch Verwendung eines Dual-Mode-Sockets verwendet werden.

VERÄNDERT

- Die Eigenschaften/Methoden DnsDeviceEndPoint.Address, DnsDeviceEndPoint.AddressFamily und DnsDeviceEndPoint.DiscardResolvedAddress() lösen eine NotSupportedException aus.

BEHOBEN

- Problem mit falschen Werten, die von PlcDeviceConnection.ReadInt64() und PlcDeviceConnection.ReadUInt64() zurückgegeben werden, oder wenn ein PlcInt64 oder PlcUInt64 verwendet wird (außer für Array-Werte).

2.4.0.0 (freigegeben am 11.10.2022)

NEU

- Produktpflege.

BEHOBEN

- Problem des ignorierten Kanaltyps SimaticChannelType.ProgrammerDevice, wenn der SimaticDeviceType S7300 oder S7400 ist.

2.3.2.0 (freigegeben am 14.01.2022)

BEHOBEN

- Problem mit falschen Werten, die von PlcDeviceConnection.ReadValues() zurückgegeben werden.
- Problem mit NullReferenceException bei der Ausführung des QueryDynamicData-Beispiels.
- Problem mit immer noch falsch zurückgegebenem NoData-Status, wenn mehrere PlcValues gelesen werden, aber einer von ihnen die Datenblockgröße überschreitet.
- Problem, das dazu führte, dass SimaticBlockInfo.Size in einigen Fällen eine falsche Zahl zurückgab.

2.3.1.0 (freigegeben am 02.12.2021)

NEU

- Einführung der neuen Eigenschaften IPDeviceEndPoint.Port und IPDeviceEndPoint.IsCustomPort einschließlich der Felder IPDeviceEndPoint.DefaultPort, IPDeviceEndPoint.MinPort und IPDeviceEndPoint.MaxPort.

BEHOBEN

- Problem mit der TypeLoadException, die beim Veröffentlichen einer .NET-Anwendung mit ReadyToRun ausgelöst wird.

2.3.0.1 (freigegeben am 03.08.2021)

BEHOBEN

- Problem mit unerwarteten/inkonsistenten Kodierungen, die beim Lesen eines String-Wertes auf verschiedenen Betriebssystemen verwendet werden.
- Problem mit unnötigen String-Allokationen beim Vergleich von PlcNames.

2.3.0.0 (freigegeben am 21.05.2021)

NEU

- Unterstützung für LDT mit PlcDateTimeLong eingeführt.
- Einheitliche PlcStatus-Informationen zwischen ReadValue- und ReadValues-Aufrufen in Bezug auf ihre PlcStatus.Type-Informationen.
- Unterstützung der PlcType-bezogenen PlcStatus-Evaluierung nach Abschluss eines Vorgangs. Dies bedeutet, dass PlcNotifications.EvaluateStatus auch für PlcType-bezogene Vorgänge aufgerufen wird.
- Logik zur korrekten Bestimmung, welche Datenbereiche mit PlcDeviceConnection.ReadValues nicht gelesen werden konnten. Dies bedeutet, dass die Verwendung mehrerer PlcValues zum Lesen von Daten, die möglicherweise die Größe eines Datenblocks überschreiten, jetzt den korrekten Status und Wert für alle Datenbereiche ergibt, auf die zugegriffen wird.

VERÄNDERT

- PlcDeviceConnection.Status spiegelt nicht länger den zuletzt verwendeten PlcType wider. Verwenden Sie PlcDeviceConnection.GetStatus, um den Status des stattdessen verwendeten PlcType zu bestimmen.

BEHOBEN

- Problem bei der Ermittlung der Größe von sehr kleinen Datenbausteinen.
- Problem mit hoher CPU-Auslastung, wenn die SPS die Verbindung mit einem FIN-Flag schließt.

2.2.0.3 (freigegeben am 01.03.2021)

BEHOBEN

- Problem mit dem Verweis auf System.ValueTuple.

2.2.0.2 (freigegeben am 18.11.2020)

BEHOBEN

- Problem bei der Ermittlung von PlcObject Eigenschaften mit den selben Namen wie diese durch PlcObject definiert werden.

2.2.0.1 (freigegeben am 29.09.2020)

BEHOBEN

- Problem mit falscher Interpretation des Channel Types SimaticChannelType.Other.

2.2.0.0 (freigegeben am 09.09.2020)

NEU

- Die Licenser-Klasse bietet jetzt verschiedene Faillf...- and Throwf...-Methoden, um die verwendete Lizenz bei Bedarf anhand des Benutzercodes zu überprüfen. Verwenden Sie Faillf...-Methoden in Ihren DEBUG-Konfigurationen und Throwf...-Methoden in Ihren RELEASE-Konfigurationen, um Sie vor der nicht lizenzierten Verwendung des SDK zu schützen.
- Zusätzliche Konstruktoren in SimaticDevice zum Akzeptieren eines addressOrHostName und des entsprechenden SimaticDeviceType.

VERÄNDERT

- **Lizenzcodes, die vor Version 2.2 verwendet wurden, funktionieren nicht mehr. Kunden erhalten neue Lizenzschlüssel, die in allen unterstützten Framework funktionieren.**
- **Alle Klassen mit dem Präfix „Siemens“ wurden in „Simatic“ umbenannt, um die Produktbeziehung besser auszudrücken, insbesondere in Bezug auf andere Produkte, die mit Siemens-Produkten arbeiten.**

2.1.2.0

NEU

- Preview release.

2.1.1.0

NEU

- Implementierung der neunten SiemensDevice.Options-Eigenschaften zur weiteren Anpassung der Gerätekommunikationseigenschaften wie der Verwendung einer reduzierten PDU-Größe.

2.1.0.0

NEU

- Integrierte IPS7LnkNet-Codebasis direkt in die Assembly (=geringere Dateigröße).
- Das Framework wurde für die Verwendung unter .NET Standard und .NET Core 2.0 vorbereitet.
- Einführung der LicenseInfo-Eigenschaft in der Licenser-Klasse, um einige Informationen zu den verwendeten Lizenzbedingungen bereitzustellen.

2.0.1.5

BEHOBEN

- Probleme beim Lesen eines einzelnen Wertes, der mit ReadValues nicht adressierbar ist, dass immer noch eine verbindungsweite Ausnahme ausgelöst wird, obwohl die Statusbewertung im Falle einer Ausnahme bei PlcValues fortgesetzt wird.

2.0.1.4

BEHOBEN

- Problem beim Lesen von Datenbereichen, die die Grenzen eines Datenblocks erreichen.
- Problem beim Lesen von Zeichenfolge-Werten, die länger als 221 Zeichen sind.

2.0.1.3

BEHOBEN

- Problem beim Lesen eines PlcString, das einen allgemeinen Fehler verursachte, wenn das Lesen fehlschlug.

2.0.1.2

NEU

- Knotenüberquerung implementiert, um Knoten rekursiv korrekt zu initialisieren und zu aktualisieren.

BEHOBEN

- Problem mit falscher Behandlung der Ergebnisse des Knotenbetriebs, die zu falschen Statusinformationen des Knotens führen.
- Problem mit neunmaligen Verbindungsstatusbewertungen beim Abrufen von Geräteinformationen.

2.0.1.1

NEU

- Die ToString-Implementierung in IPDeviceEndPoint wurde korrigiert, um die virtuelle Adresseigenschaft zu verwenden.

BEHOBEN

- Problem mit der Auslösung von IPV6-Adressen, die nicht unterstützt werden.

2.0.1.0

NEU

- Implementierung der GetSupportedRawTypes-Methode in PlcType, um die unterstützten PlcRawTypes für die Adressierung eines bestimmten PlcTypeCode zu ermitteln.
- Implementierte IsSupported-Methode in PlcType, um zu bestimmen, ob ein bestimmter PlcRawType auf einen bestimmten PlcTypeCode zugreifen kann.
- Implementierte IsSupportedArreyType-Methode in PlcType, um zu bestimmen, ob ein PlcTypeCode zum Definieren eines Array-Typs des aktuellen Typs verwendet werden kann.
- Implementierung der MakeArrayType-Methode in PlcType zum Erstellen von PlcType-Objekten eines Array-Typs.
- Implementierte GetSupportedRawTypes-Methode in PlcOperand, um die unterstützten PlcRawTypes zu ermitteln, um den SPS-Speicher eines bestimmten PlcOperandType zu adressieren.
- Implementierte IsSupported-Methode in PlcOperand, um zu bestimmen, ob auf einen bestimmten PlcOperandType von einem bestimmten PlcRawType zugegriffen werden kann.

BEHOBEN

- Problem mit nicht eindeutigen PlcType-Hashcodes.

2.0.0.0

NEU

- Implementierung einer neuen Name-Eigenschaft in IPlcDevice und Hinzufügen spezifischer

Gerätetypen zu SiemensDeviceType.

- Implizierte Zeichenfolge zum PlcOperand-Cast-Operator hinzugefügt.
- Unterstützung für die UDT-Operanden (UserDefinedType) implementiert.
- Neue Symbols-Eigenschaft in IPlcDevice implementiert.
- Verbesserte Unterstützung für Timer- und Zähleroperanden zur automatischen Verwendung von PlcRawType.Byte für den Fall, dass die Adresszeichenfolge keinen Rohtyp enthält.
- Erweiterte Implementierung von PlcValue und PlcArray mit der Eigenschaft Name und Implementierung eines neuen PlcIdentity-Systems zur Unterstützung verschiedener Identitäten wie Adressen, Operanden und Namen.
- Alle PlcValue- und PlcArray-Klassen können jetzt als Symbole verwendet werden.
- Der PlcType wurde hinsichtlich der neuen Basisklasse PlcName und PlcIdentity von PlcOperand und PlcAddress vollständig überprüft.
- Aktualisierte Dokumentation der PlcAddress-Klasse.
- Weitere Adressbeschränkungen für Operandentypen Zähler und Timer implementiert.
- Neue Adressableitungen wurden implementiert, um explizite Bit-, Bite-, Wort- und Wortadressen darzustellen.
- Verschiedene operandentypspezifische Operandenfelder wurden zu Eigenschaften überarbeitet.
- Implementierte spezialisierte Unterklassen, um jeden möglichen Operandentyp durch seinen eigenen Klassentyp darzustellen.
- Unterstützung für UserDefinedType-Operanden implementiert.
- Vollständig überarbeitete PlcType-Implementierung mit PlcTypeMetadata.
- Den gesamten Satz möglicher Of-Methodenüberladungen implementiert, um die möglichen Typmetadatenkombinationen darzustellen.
- Die Ausnahmemeldungen beim Erstellen eines neuen PlcType wurden angepasst.
- Implementierung einer ganzen Reihe von Unterklassen des PlcType, um alle verschiedenen Typzuordnungen (vom PC-Typ zum SPS-Typ) als einen einzigen eindeutigen PlcType darzustellen, der implementiert wurde, um die internen Typdarstellungen nicht mehr zu kennen.
- Implementiert für jede PlcType-Unterklasse die entsprechende SPS-Mitgliedsinformationsklasse, um verschiedene nachfolgende Mitglieder nach Name / Typ-Paaren für andere Typen wie den neuen PlcObjectType zu definieren.
- Implementierung der neuen SPS-typspezifischen Unterklassen.
- Es wurden zusätzliche Ctors implementiert, um Instanzen mit vordefinierten Typen in Wert-/ Array-Klassen zu erstellen.
- Einführung von Typen und Werten in IPlcDevice.
- DnsDeviceEndPoint wurden implementiert, um mithilfe von DNS-Informationen auf ein PlcDevice zuzugreifen.
- Implementierte IPlcEntity-basierte Lese- / Schreibvorschläge einschließlich Ergebnisinstanzen in PlcDeviceConnection.
- Implementierung einer Tag-Eigenschaft für PlcObject und PlcMember.
- Die GetHashCode-Implementierung in PlcAddress und PlcOperand wurde überarbeitet.

VERÄNDERT

- Überarbeitete ToString-Implementierung in SPS-Wert, Typ- und Infoklassen.
- Read<T> wurde zu ReadObject<T> umbenannt.
- Write<T>(T instance) wurde zu WriteObject<T>(T instance) umbenannt.
- WriteValuesCore<T>(PlcAddress, T[]) existiert nicht mehr. Ersetzt durch WriteValuesCore<T>(PlcValueType|PlcArrayType, value(s))

BEHOBEN

- Problem mit Informationen zur Endadresse des Bitarrays.

1.9.1.1 (freigegeben am 13.09.2019)

BEHOBEN

- Leistungsproblem mit `PlcDeviceConnection.ReadValues(IEnumerable<PlcType>)`.

1.9.1.0 (freigegeben am 24.06.2019)

VERÄNDERT

- Die TSAP-Codierung verwendet jetzt ISO-8859-1 anstelle von Windows-1252 für eine bessere Unterstützung des .NET-Standards.

1.9.0.0 (freigegeben am 15.05.2019)

NEU

- Das Framework wurde für die Verwendung unter .NET Standard 2.0 und .NET Core 2.0 oder höher vorbereitet.

1.8.2.3 (freigegeben am 13.05.2019)

BEHOBEN

- Exception beim direkten Zugriff auf die IPS7-Klasse.

1.8.2.2 (freigegeben am 09.05.2019)

BEHOBEN

- Probleme beim Herstellen einer Verbindung zu S7-200.

1.8.2.1 (freigegeben am 03.04.2019)

BEHOBEN

- Problem beim Laden von IPS7LnkNet-Advanced als eingebettete Ressource.
- Exception beim Zugriff auf die LicenseInfo-Eigenschaft in der Licenser-Klasse.

VERÄNDERT

- In den Beispielen wurden die angepassten Methoden `Read<T>()` und `Write<T>(T instance)` in `ReadObject<T>()` und `WriteObject<T>(T instance)` und das Zielframework auf .NET Framework 4.0 angepasst.

1.8.2.0 (freigegeben am 20.03.2019)

NEU

- Unterstützung für S7-1500 zur Bestimmung der Größe von Datenblöcken.
- Auf .NET Framework 4.0 aktualisiert.
- Einführung der LicenseInfo-Eigenschaft für alle Licenser-Klassen, um einige Informationen zu den verwendeten Lizenzbedingungen bereitzustellen.

1.8.1.1 (freigegeben am 07.04.2018)

BEHOBEN

- Problem mit dem Anbieter eines virtuellen Geräts, der NoData zurückgegeben hat, falls noch keine Daten geschrieben wurden.
- Problem mit falscher Zuweisung gepolter Verbindungskanäle, wenn dort ein neuer Kanal erstellt werden soll, obwohl ein vorhandener Kanal dem Verbindungsaufbau entsprechen würde.

1.8.1.0 (freigegeben am 06.04.2018)

NEU

- Implementiertes Verbindungspooling mithilfe der neuen kanalbasierten Gerätekommunikationsschicht. Dies ermöglicht das Erstellen mehrerer Verbindungsobjekte unter Verwendung derselben Kommunikationsschicht auf der Basis des unteren Kanals, falls der Endpunkt und die Gerätekonfiguration einer oder mehrerer Verbindungen übereinstimmen. Um diese Funktion zu aktivieren, muss PlcDeviceConnection.UsePool auf den Wert true gesetzt werden. Der Standardwert ist false, um das vorherige Verhalten beizubehalten und zusätzlich mehrere unabhängige Verbindungen zu unterstützen, z.B. eine für Lese- und eine für Schreiboperationen.

BEHOBEN

- Problem mit falsch weitergeleiteter innerer Exception, die zu einer ArgumentNullException anstelle einer erwarteten PlcException mit der entsprechenden InnerException führte.
- Problem mit manchmal nicht korrekt aktualisierten Statusinformationen, insbesondere bei PlcType-basierten Statusinformationen.

1.8.0.2 (freigegeben am 20.02.2018)

BEHOBEN

- Problem mit der implizierten Verwendung von LReal, wenn „double“ als Type eines Elements in einem PlcObject verwendet wird.

1.8.0.1 (freigegeben am 17.12.2017)

BEHOBEN

- Problem mit falschem WriteLReal.

1.8.0.0 (freigegeben am 17.12.2017)

NEU

- Unterstützung für LInt-, LWord- und LReal-Datentypen implementiert.
- Unterstützung für lokale und Remote-TSAP in IPDeviceEndPoint implementiert.
- Implementierung zusätzlicher Ctors in SiemensDevice zur Vereinfachung der Objekterstellung.
- Unterstützung für Hostnamen-basierte SPS-Adressierung implementiert, siehe DnsDeviceEndPoint.

1.7.3.0 (freigegeben am 14.11.2017)

NEU

- Zusätzliche Überladung der Read<T>() implementiert, um Daten in ein vorhandenes PlcObject

einzulesen, falls das konkrete PlcObject dort keinen Standardkonstruktor bereitstellt.

VERÄNDERT

- Umbenannte Methoden Read<T>() und Write<T>(T instance) in ReadObject<T>() und WriteObject<T>(T instance), um parameterisierte Leseüberladungen bereitzustellen, die nicht mit anderen Lesemethoden kollidieren.

1.7.2.4 (freigegeben am 28.10.2017)

BEHOBEN

- Problem beim Herstellen einer Verbindung zur Logo 8.x-SPS.

1.7.2.3 (freigegeben am 04.09.2017)

BEHOBEN

- Problem beim Lesen der maximalen Anzahl von Zeichen bei Verwendung von PlcString.GetValue.

1.7.2.2 (freigegeben am 30.05.2017)

BEHOBEN

- Problem beim Lesen von Datenbereichen, die die Grenzen eines Datenblocks erreichen.
- Problem beim Lesen von Zeichenfolgenwerten, die länger als 221 Zeichen sind.

1.7.2.1 (freigegeben am 08.04.2017)

BEHOBEN

- Problem beim Lesen von Daten mit ReadValues(PlcType[]|IEnumerable<PlcType>) : object[], was zu einem „verkleinerten“ Ergebnis führte. Dieses „verkleinerte“ Ergebnis wurde erzeugt, wenn dort derselbe PlcType mehrmals verwendet wird und die Daten nur für einen von ihnen bereitgestellt wurden.

1.7.2.0 (freigegeben am 12.10.2016)

NEU

- Geänderte Sequenz / Logik innerhalb dieses IPlcValue. Bisher wurden geänderte Ergebnisse angesprochen. Bis zu dieser Version wurde beim Lesen der Werte ein geändertes Ergebnis ausgelöst, solange sich die Verbindung im synchronisierten Zustand befindet. Dies bedeutet so weit, dass eine nachfolgende Operation, die dieselbe Verbindung innerhalb des Handels für geänderte Ergebnisse + separaten Thread verwendet, zu einem Deadlock führt. Ab dieser Version (V1.7.2.0) werden die geänderten Ereignisse ausgelöst, nachdem der synchronisierte Status der Verbindung beendet wurde.

1.7.1.1 (freigegeben am 28.09.2016)

BEHOBEN

- Problem mit fehlenden Wochentagsanteilen in PlcDataTime.

1.7.1.0 (freigegeben am 28.09.2016)

NEU

- Die Überprüfung von Änderungen mit tiefem Wert wurde in PlcArray-Instanzen implementiert, um auch das Ergebnis „Geändert“ auszulösen, wenn mindestens ein Element im Array während eines Wertelesevorgangs geändert wird.

1.7.0.0 (freigegeben am 11.03.2016)

NEU

- Optimiertes Multi-Write (WriteValues) implementiert, um jeden Wert innerhalb einer Transaktion länger zu schreiben. Dies beschleunigt jetzt die Leistung beim gleichzeitigen Schreiben von Multiple Werten.

1.6.9.2 (freigegeben am 24.02.2016)

BEHOBEN

- Interner gepufferter Leseanforderungscache entfernt, um die Treiberoptimierung auf niedriger Ebene zu verbessern. Dies behebt auch Probleme beim gleichzeitigen Lesen derselben Zeichenfolgewerte.

1.6.9.1 (freigegeben am 20.01.2016)

NEU

- Auf IPS7LnkNet v1.1.77.6 aktualisiert

BEHOBEN

- MultiRead mit PlcString seit Version 1.1.77.5: Wenn Sie MultiRead verwenden, um denselben PlcString mehrmals zu lesen, hat der Treiber die Länge der Zeichenfolge verringert. Sobald die Länge einen Wert unter Null erreichte, löste der MultiRead eine Exception aus.

1.6.9.0 (freigegeben am 27.12.2015)

NEU

- Implementierte Methode zum Abrufen von SPS-Blockinformationen wie das Vorhandensein oder die Größe eines in einem SPS-Gerät definierten DataBlocks.
- Implementierung von PlcOperand mithilfe von Zeichenfolgenwerten.

1.6.8.1 (freigegeben am 16.12.2015)

BEHOBEN

- Problem mit ReadValues mithilfe von PlcString-Daten, die Datenbereiche adressieren, die das Ende eines Datenblocks erreichen.

1.6.8.0 (freigegeben am 15.09.2015)

BEHOBEN

- Problem mit ungelösten Sperrsituationen in Umgebungen mit mehreren Threads, wenn während der

Verbindung eine Ausnahme aufgetreten ist, die darin besteht, von mehreren Threads gleichzeitig verwendet zu werden.

1.6.7.2 (freigegeben am 04.09.2015)

NEU

- Neues Produktsymbol.

1.6.7.1 (freigegeben am 03.03.2015)

BEHOBEN

- InvalidCastException beim Lesen eines Bool[] mit ReadBoolean.

1.6.7.0 (freigegeben am 26.02.2015)

NEU

- Implementierung der SiemensChannelsType-Enumeration von SiemensDevice, um den Kanaltyp zu verwalten, der für eine von SiemensDevice erstellte Verbindung verwendet werden soll, anstatt ihn über die generische PlcDeviceConnection mit dem PlcDeviceConnectionType zu verwalten. Im Allgemeinen ist ein Kanaltyp nur für den Siemens-Anbieter relevant. Daher wird der PlcDeviceConnectionType in zukünftigen Versionen veraltet sein.
- Die implementierte IsDisposed-Eigenschaft für intern zeigt an, dass eine PlcDevice-Verbindung entsorgt wurde.
- Verbesserte GetInfo ()-Methode für SiemensDeviceConnection, um keine wiederkehrende Verbindungsstatusänderungen und Statusauswertungen mehr zu erzeugen.
- Die Überprüfung in PlcType wurde implementiert, um die Verwendung einer anderen PlcAddress.RawType als Bit für PlcType-Instanzen mit PlcTypeCode.Bool zu verbieten.

BEHOBEN

- Falsche Berechnung der Endadresse von PlcType-Instanzen, die einen Bool[] darzustellen.

1.6.6.1 (freigegeben am 24.02.2015)

NEU

- Die Implementierung wurde überarbeitet, um die Eigenschaft für die maximale Zeichenfolgelänge (Length) besser nutzen zu können. Achten Sie außerdem auf PlcString.MinLength und PlcString.MaxLength und füllen Sie den Wert nicht mehr mit Leerzeichen auf.
- Problem mit veralteten Rückgabedaten von ReadValues behoben und einige Lesemethoden vereinfacht, um den EnumerableHelper besser nutzen zu können.

BEHOBEN

- Problem mit falschem Pufferarray-Typ (Kurzpuffer in Float-Puffer geändert).

1.6.6.0 (freigegeben am 23.02.2015)

NEU

- PlcStatus wurde auf die neuesten LatestXXX Eigenschaften überarbeitet, um die aktuellen Statusinformationen direkt abzurufen / festzulegen, ohne dass die Initialisierung beendet werden

muss, um die neuesten Statusinformationen während der Verarbeitung in einem SPS-Betriebskontext abzurufen.

- Bei Multi-Read (ReadValues) und Multi-Write (WriteValues) spiegelt die Verbindung keine Datenstatuscodes mehr wider. Datenstatuscodes werden jetzt in der Verbindung mit dem verwendeten PlcType und auf jedem IPlcValue-instanzspezifischen PlcStatus gespeichert.

1.6.5.0 (freigegeben am 19.02.2015)

NEU

- Verbesserte Exception- und Statusinformationen.
- Verbesserte Behandlung von Status- und Wertänderungen auf transaktionale Weise.

BEHOBEN

- Multi-Threaded Problem in PlcType.Of(...)

1.6.4.0 (freigegeben am 09.02.2015)

NEU

- Neue Überladungen für GetValue und SetValue implementiert. Zusätzlich implementierter ValidateValue. Darüber hinaus wurde der Mechanismus zum Abrufen und Festlegen der Value-Eigenschaft sowie die Implementierung von GetValue und SetValue überarbeitet, um eine bessere Integration in neue Lösungen zu ermöglichen und aussagekräftigere Codeflows zum Ableiten von Klassen bereitzustellen.
- Auf IPS7LnkNet v1.0.74.0 aktualisiert

BEHOBEN

- Problem beim Lesen und Schreiben von Zeichenwerten behoben.
- Die Verbindung zur SPS hat sich nicht automatisch repariert.
- Verwenden von Threads mit Verbindungen: Die Lizenz ist mehrmals abgelaufen.

1.6.3.4 (freigegeben am 15.12.2014)

BEHOBEN

- Problem mit doppelten PlcOperand-Instanzen in Umgebungen mit mehreren Threads.

1.6.3.3 (freigegeben am 09.12.2014)

BEHOBEN

- Problem beim Lesen von Referenzdaten (Array-Daten) mit ReadValues auf einer PlcDeviceConnection-Instanz, das zu überschriebenen Daten beim Lesen derselben Adressbereiche durch denselben PlcType und zu einem Verweis auf den internen Datenpuffer führt. Darüber hinaus wurden benutzerspezifische Referenzdaten mit einem Verweis auf den internen Datenpuffer überschrieben.

1.6.3.2 (freigegeben am 28.11.2014)

NEU

- IPS7LnkNet (v1.73) aktualisiert: Logo/S7-200 Lizenzverwaltung implementiert

BEHOBEN

- Problem mit fehlgeschlagenen automatischen Verbindungsversuchen, bei denen eine Kommunikation gestartet wird, während das Netzwerkgerät beim Start deaktiviert und bei weiteren Verbindungs- / Zugriffsversuchen aktiviert wird.

1.6.2.1 (freigegeben am 23.11.2014)

BEHOBEN

- Problem mit Standardwerten beim Lesen von Daten mit ReadValues.

1.6.2.0 (freigegeben am 07.11.2014)

BEHOBEN

- Die Basisklasse von PlcException wurde von ExternalException in Exception geändert, da die nicht verlegte SPS-Schicht nicht mehr zu einer externen (nicht verwalteten) API gehört. Darüber hinaus führt diese Änderung zu einer vollständigen Unterstützung der Verwendung in Windows Store Apps.

1.6.1.0 (freigegeben am 12.09.2014)

NEU

- Neuer Verbindungsstatus hinzugefügt Disconnected.
- Erweiterte globale Benachrichtigungen mit neuem Ereignis ConnectionDisconnected.
- Erweiterte abstrakte Basisklasse PlcDeviceConnection mit neuem Ereignis Disconnected.
- Überarbeitete Behandlungen des Verbindungsstatus, um während jeder Interaktion mit einer Verbindung festzustellen, ob sich eine Verbindung entweder im verbundenen oder im getrennten Zustand befindet.

1.6.0.0 (freigegeben am 28.08.2014)

NEU

- Setter zur generischen IPlcValue<T> Schnittstelle hinzugefügt, um auch das Festlegen der Value-Eigenschaft zu unterstützen.
- Neues Ereignis ConnectionCreated in PlcNotifications, das immer dann auftritt, wenn eine neue PlcDeviceConnection erstellt wurde.
- Neue Ereignisse Connecting und Connected auf PlcDeviceConnection, die darüber informieren, dass eine Verbindung zu einem Gerät besteht und vollständig hergestellt wurde.
- Neues Ereignis StateChanged in PlcDeviceConnection, das über jede Änderung der State-Eigenschaft von PlcDeviceConnection informiert.
- Neues Ereignis ConnectionStateChanged in PlcNotifications, das über jede Änderung der State-Eigenschaft einer PlcDeviceConnection informiert.
- Neue Eigenschaft IsConnected in PlcDeviceConnection, die angibt, ob eine Verbindung zu einem Gerät vollständig hergestellt wurde.
- Neue Methode Connect () auf PlcDeviceConnection, um manuell eine vollständige Verbindung zu einem Gerät herzustellen.
- Der Datenstatus-Cache einer PlcDeviceConnection enthält nicht immer dieselbe PlcStatus-Instanz. Dies ermöglicht das Abonnieren von Statusänderungen für PlcType-spezifische PlcStatus-Objekte.
- Die PlcConnectionState-Enumeration definiert jetzt die neuen Zustände Connecting and Connected.
- Die PlcConnectionStateChangedEventArgs-Klasse wird vom neuen stateChanged-Ereignis in

PlcDeviceConnection verwendet und bietet OldState- und NewState-Eigenschaften.

- In VirtualDeviceConnection wurde die Exception-Behandlung implementiert, um sicherzustellen, dass sich alle Implementierungen des PlcDevice-Anbieters gleich verhalten.
- Die Exception-Behandlung wurde in WebDeviceConnection implementiert, um sicherzustellen, dass alle PlcDevice-Provider-Implementierungen auf dieselbe Weise ausgeführt werden. Die vorhandene Exception-Behandlung wurde zusätzlich an das erwartete Verhalten angepasst.
- Implementierte Exception-Eigenschaften in PlcStatus, die die entsprechende PlcException-Instanz bereitstellt, die mit den von einer PlcStatus-Instanz bereitgestellten Statusinformation initialisiert wurde.
- Implementierte InnerException-Eigenschaft in PlcStatus, die die letzte von einer internen Operation ausgelöste Exception-Instanz bereitstellt. Diese Eigenschaft wird zusätzlich mit der SocketCode-Eigenschaft synchronisiert.
- Implementierte TimeStamp-Eigenschaft in PlcStatus, die die DateTime der letzten Aktualisierung der von PlcStatus bereitgestellten Status-Metadaten angibt.
- Implementierte Type-Eigenschaft in PlcStatus, die die PlcType-Informationen des PlcType bereitstellt, der von der letzten Operation verwendet wurde.
- Alle abstrakten Lese- / Schreibmethoden in PlcDeviceConnection werden jetzt mit mindestens einem PlcType aufgerufen. Diese Instanz macht nun frühere Parameter wie Adresse: PlcAddress, count:int or length:int überflüssig.

VERÄNDERT

- Die Statusänderung in „Faulted“ wurde entfernt, nachdem eine PlcDeviceConnection entsorgt wurde.
- Die vorherige PlcDeviceConnectionEventArgs-Klasse ist jetzt eine Unterklasse von PlcNotifications. Da es spezifisch für die von der PlcNotifications-Klasse bereitgestellten Ereignisse ist und daher einerseits die betroffene PlcDeviceConnection bereitstellt, weil von den Ereignissen kein Absender bereitgestellt wird und andererseits mit dem neuen PlcDeviceConnectionEventArgs in Konflikt steht, das dem Ereignis keine Quellverbindung bereitstellt.
- ValidateStatusDelegate wurde in EvaluateStatusDelegate umbenannt, um dem Konzept, das hinter der Verwendung des Rückrufs steht, besser zu entsprechen. Zusätzlich wurde ein Bool als Rückgabotyp hinzugefügt, um anzugeben, dass eine Framework-spezifische Bewertung durchgeführt werden soll.

1.5.8.2 (freigegeben am 12.08.2014)

NEU

- Internes geändertes Ereignis in PlcDeviceEndPoint das bei Änderungen an einem Endpunkt auftritt.
- Allgemeine Exception-Behandlung bei jedem API-Aufruf an den Siemens-Anbieter.

BEHOBEN

- Setzen Sie die zwischengespeicherte Verbindung in PlcDevice zurück, falls sich der Endpunkt ändert.

1.5.8.1 (freigegeben am 21.07.2014)

BEHOBEN

- Falsche Berechnung der S5TIME-Werte (Ursache: High- und Low-Byte mussten umgekehrt werden).

Inhaltsverzeichnis

Getestet? Du willst es?	1
Historie	2
2.5.0.0 (freigegeben am 23.01.2024)	2
2.4.0.0 (freigegeben am 11.10.2022)	2
2.3.2.0 (freigegeben am 14.01.2022)	2
2.3.1.0 (freigegeben am 02.12.2021)	2
2.3.0.1 (freigegeben am 03.08.2021)	3
2.3.0.0 (freigegeben am 21.05.2021)	3
2.2.0.3 (freigegeben am 01.03.2021)	3
2.2.0.2 (freigegeben am 18.11.2020)	3
2.2.0.1 (freigegeben am 29.09.2020)	3
2.2.0.0 (freigegeben am 09.09.2020)	4
2.1.2.0	4
2.1.1.0	4
2.1.0.0	4
2.0.1.5	4
2.0.1.4	4
2.0.1.3	5
2.0.1.2	5
2.0.1.1	5
2.0.1.0	5
2.0.0.0	5
1.9.1.1 (freigegeben am 13.09.2019)	7
1.9.1.0 (freigegeben am 24.06.2019)	7
1.9.0.0 (freigegeben am 15.05.2019)	7
1.8.2.3 (freigegeben am 13.05.2019)	7
1.8.2.2 (freigegeben am 09.05.2019)	7
1.8.2.1 (freigegeben am 03.04.2019)	7
1.8.2.0 (freigegeben am 20.03.2019)	7
1.8.1.1 (freigegeben am 07.04.2018)	8
1.8.1.0 (freigegeben am 06.04.2018)	8
1.8.0.2 (freigegeben am 20.02.2018)	8
1.8.0.1 (freigegeben am 17.12.2017)	8
1.8.0.0 (freigegeben am 17.12.2017)	8
1.7.3.0 (freigegeben am 14.11.2017)	8
1.7.2.4 (freigegeben am 28.10.2017)	9
1.7.2.3 (freigegeben am 04.09.2017)	9
1.7.2.2 (freigegeben am 30.05.2017)	9
1.7.2.1 (freigegeben am 08.04.2017)	9
1.7.2.0 (freigegeben am 12.10.2016)	9
1.7.1.1 (freigegeben am 28.09.2016)	9
1.7.1.0 (freigegeben am 28.09.2016)	10
1.7.0.0 (freigegeben am 11.03.2016)	10
1.6.9.2 (freigegeben am 24.02.2016)	10
1.6.9.1 (freigegeben am 20.01.2016)	10
1.6.9.0 (freigegeben am 27.12.2015)	10
1.6.8.1 (freigegeben am 16.12.2015)	10
1.6.8.0 (freigegeben am 15.09.2015)	10
1.6.7.2 (freigegeben am 04.09.2015)	11
1.6.7.1 (freigegeben am 03.03.2015)	11
1.6.7.0 (freigegeben am 26.02.2015)	11

1.6.6.1 (freigegeben am 24.02.2015)	11
1.6.6.0 (freigegeben am 23.02.2015)	11
1.6.5.0 (freigegeben am 19.02.2015)	12
1.6.4.0 (freigegeben am 09.02.2015)	12
1.6.3.4 (freigegeben am 15.12.2014)	12
1.6.3.3 (freigegeben am 09.12.2014)	12
1.6.3.2 (freigegeben am 28.11.2014)	12
1.6.2.1 (freigegeben am 23.11.2014)	13
1.6.2.0 (freigegeben am 07.11.2014)	13
1.6.1.0 (freigegeben am 12.09.2014)	13
1.6.0.0 (freigegeben am 28.08.2014)	13
1.5.8.2 (freigegeben am 12.08.2014)	14
1.5.8.1 (freigegeben am 21.07.2014)	14