



Wichtige Hinweise für die Migration von PLCcom Version 8 oder älter

Indi.An Systems GmbH

Flughafenallee 3

28199 Bremen

Deutschland

support@indi-systems.de

Tel + 49 421-989703-30

Fax + 49 421-989703-39

Inhaltsverzeichnis

Über dieses Dokument?	3
Welche Programmierplattformen unterstützt PLCcom?	4
Optimiertes Lesen oder Schreiben von Datenbereichen	5
Sonstige Änderungen	6
Entfernte und ersetzte Funktionalitäten.....	6
Veraltete Funktionalitäten	7
Haben Sie Fragen?	8

Über dieses Dokument?

Das vorliegende Dokument soll Ihnen einen ersten Überblick über die empfohlenen Änderungen für die Migration von PLCcom V8 oder älter auf die aktuelle Version liefern. Es handelt sich nicht um eine komplette Dokumentation. Weitergehende Informationen entnehmen Sie bitte

- den Code-Beispielen im Auslieferungspaket,
- den Code-Beispielen und der Onlinehilfe auf unserer Website <http://www.plccom.net/code-examples/plccom/s7.html>
- sowie der jeweiligen Onlinehilfe im Auslieferungspaket (index.html)

Alle Angaben in diesem Dokument werden ohne Gewähr veröffentlicht. Änderungen und alle Rechte vorbehalten. Der Inhalt dieses Dokumentes ist urheberrechtlich geschützt und darf ohne unsere Zustimmung nicht (auch nicht in Teilen) vervielfältigt, reproduziert, übertragen, in Medien verarbeitet und gespeichert oder übersetzt werden.

Hinweis:

Alle Produktnamen oder andere Namen oder Marken auf die in diesem Dokument Bezug genommen wird, sind Warenzeichen oder eingetragene Warenzeichen und Eigentum ihrer jeweiligen Inhaber. Es besteht keinerlei Verbindung zwischen der genannten Marke oder dem Markeninhaber und der Fa. Indi.Systems GmbH. Jegliche Nennung von Marken dient ausschließlich als Hinweis zur Nutzung und Verwendungszweck.

Welche Programmierplattformen unterstützt PLCcom?

Abweichend zu den vorherigen Versionen unterstützt PLCcom in aktueller Version eine Reihe weiterer Programmierplattformen wie **Android, Xamarin, oder .Net-Core**

PLCcom für S7 wird in drei Versionen zur Verfügung gestellt:

1. .Net Version

Die .Net-Version unterstützt die klassische .Net Framework-Programmierung.

Des Weiteren enthält das Auslieferungspaket eine Version für **.Net-Standard Version 2.1**

Diese Komponente kann zur Entwicklung von .Net-Core-, Xamarin-, UWP- oder Unity-Applikationen eingesetzt werden.

2. Windows CE-Version

Es wird die Entwicklung von Applikationen unter Windows CE ab Version 5 unterstützt.

3. Java-Version

Die Java-Version bietet Entwicklern eine Java-Komponente zur Java-Applikationsentwicklung z.B. mit Eclipse- oder Netbeans.

Seit Version 9 wird zusätzlich auch die Entwicklung von Android-Apps unterstützt.

Konkrete Hinweise zur Anwendung finden Sie im Handbuch „Erste Schritte“ ab Kapitel „Welche System-Komponenten sind Voraussetzung für den Betrieb von PLCcom?“

Optimiertes Lesen oder Schreiben von Datenbereichen

In der nun aktuellen Version der PLCcom-Library wurde die Funktion

ReadData (ReadDataRequestCollection) als deprecated markiert.

Dies betrifft nur das Lesen mittels der ReadDataRequestCollection, welche durch die neue Funktion ReadWriteData abgelöst wird, das direkte (unoptimiert) lesen oder schreiben von einzelnen Requests ist nicht von dieser Änderung betroffen..

Die nun empfohlene Vorgehensweise besteht aus der Benutzung der neuen Funktion ReadWriteData (ReadWriteRequestSet).

Durch diese neue Funktionalität wurden neue interne Optimierungsmethoden eingeführt, welche höchst performante Lese/Schreib-Operationen ermöglicht.

Es ist nun möglich zwischen mehreren Optimierungsmöglichkeiten zu wählen bzw. mit der Expert-Version PLCcom automatisch die beste Optimierung anzuwenden zu lassen. Verfügbar ist jetzt das performante Zusammenfassen von Datenbereichen oder auch das gleichzeitige Schreiben und/oder Lesen in verschiedenen Bereichen.

Die bislang benutzten Funktion ReadData(ReadDataRequestCollection) bleibt aber weiterhin innerhalb der nächsten Versionen in der Library verfügbar, eine Abwärtskompatibilität ist also sichergestellt. Trotzdem sollten bei Gelegenheit die als deprecated markierten Funktionalitäten abgelöst werden.

Konkrete Hinweise zur Anwendung finden Sie im Handbuch „Erste Schritte“ ab Kapitel „Optimiertes Lesen und Schreiben von Daten“

Sonstige Änderungen

Entfernte und ersetzte Funktionalitäten

Im Zuge der Bereinigung wurden diverse Methoden und Funktionen entfernt. Diese Funktionalitäten waren bereits innerhalb der Vorversionen als deprecated markiert und sollten dementsprechend schon längere Zeit nicht mehr benutzt werden.

Version	Type	Entfernte Funktion	Stattdessen zu benutzende Funktionalität
Alle	enum	ePLCType.Logo_compatibel	ePLCType.Logo0BA7_compatibel oder ePLCType.Logo0BA8_compatibel
Alle	enum	eDataType.UNICODECHAR	eDataType.STRING oder eDataType.S7_STRING
Alle	enum	eDataType.BCD	eDataType.BCD16
Alle	object	ReadItemRequest	ReadDataRequest
Alle	object	ReadItemRequestCollection	ReadWriteRequestSet
Alle	object	ReadRequest	ReadDataRequest
Alle	object	WriteRequest	WriteDataRequest
Alle	object	ReadItemResult	ReadDataResult
Alle	object	ReadItemResultCollection	ReadWriteResultSet
Alle	object	ReadResult	ReadDataResult
Alle	object	WriteResult	WriteDataResult
.Net	method	PLCcomDevice.SetPLCTime(DateTime)	PLCcomDevice.SetPLCClockTime (DateTime)
Java	method	PLCcomDevice.setPLCTime(Calendar)	PLCcomDevice.setPLCClockTime (Calendar)
.Net	function	PLCcomDevice.GetPLCTime()	PLCcomDevice.GetPLCClockTime()
Java	function	PLCcomDevice.getPLCTime()	PLCcomDevice.getPLCClockTime()
.Net	function	PLCComDataServer.AddReadDataRequest (ReadDataRequest value, String ItemKey)	PLCComDataServer.AddReadDataRequest (ReadDataRequest value)
Java	function	PLCComDataServer.addReadDataRequest (ReadDataRequest value, String ItemKey)	PLCComDataServer.addReadDataRequest (ReadDataRequest value)
.Net	function	PLCcomDevice.GetLocal_MPI()	PLCcomDevice.GetBUS_ADRESS_LOCAL()
Java	function	PLCcomDevice.getLocal_MPI()	PLCcomDevice.getBUS_ADRESS_LOCAL()
.Net	method	PLCcomDevice.SetLocal_MPI(int)	PLCcomDevice.SetBUS_ADRESS_LOCAL(int)
Java	method	PLCcomDevice.setLocal_MPI(int)	PLCcomDevice.setBUS_ADRESS_LOCAL(int)
.Net	function	PLCcomDevice.GetPLC_MPI()	PLCcomDevice.GetBUS_ADRESS_PLC()
Java	function	PLCcomDevice.getPLC_MPI()	PLCcomDevice.getBUS_ADRESS_PLC()
.Net	method	PLCcomDevice.SetPLC_MPI(int)	PLCcomDevice.SetBUS_ADRESS_PLC(int)
Java	method	PLCcomDevice.setPLC_MPI(int)	PLCcomDevice.setBUS_ADRESS_PLC(int)
.Net	function	PLCcomDevice.GetLocal_PPI ()	PLCcomDevice.GetBUS_ADRESS_LOCAL()
Java	function	PLCcomDevice.getLocal_PPI ()	PLCcomDevice.getBUS_ADRESS_LOCAL()
.Net	method	PLCcomDevice.SetLocal_PPI(int)	PLCcomDevice.SetBUS_ADRESS_LOCAL(int)
Java	method	PLCcomDevice.setLocal_PPI(int)	PLCcomDevice.setBUS_ADRESS_LOCAL(int)
.Net	function	PLCcomDevice.GetPLC_PPI()	PLCcomDevice.GetBUS_ADRESS_PLC()
Java	function	PLCcomDevice.getPLC_PPI()	PLCcomDevice.getBUS_ADRESS_PLC()
.Net	function	ConnectResult.HasConnected	OperationResult.getQuality

Java	function	ConnectResult.hasConnected()	OperationResult.getQuality()
.Net	function	ConnectResult.HasWorked	OperationResult.getQuality
Java	function	ConnectResult.HasWorked ()	OperationResult.Quality()
.Net	function	BasicInfoResult.Ordernummer	BasicInfoResult.Ordernumber
Java	function	BasicInfoResult.Ordernummer()	BasicInfoResult.Ordernumber()

Veraltete Funktionalitäten

Innerhalb der neuen Version wurden folgende Funktionalitäten als Deprecated markiert:

Version	Type	Deprecated	Stattdessen zu benutzende Funktionalität
.Net	function	ReadData(ReadDataRequestCollection)	ReadWriteData(ReadWriteRequestSet)
Java	function	readData(ReadDataRequestCollection)	readWriteData(ReadWriteRequestSet)
Java	function	DeviceInfo()	getDeviceInfo()
Java	function	DisConnect()	disconnect()
Java	function	BeginConnect()	beginConnect()
Java	function	StartPLC()	startPLC()
Java	function	GetPLCClockTime()	getPLCClockTime()
Java	function	SetPLCClockTime(Calendar)	setPLCClockTime(Calendar)
Java	function	GetLEDInfo()	getLEDInfo()
Java	function	GetBasicInfo()	getBasicInfo()
Java	function	GetCPUMode()	getCPUMode()
Java	function	GetBlockList()	getBlockList()
Java	function	GetBlockLenght(eBlockType, int)	getBlockLenght(eBlockType, int)
Java	function	ReadPLCBlock_MC7(eBlockType, int)	readPLCBlock_MC7(eBlockType, int)
Java	function	WritePLCBlock_MC7(WritePLCBlockRequest)	writePLCBlock_MC7(WritePLCBlockRequest)
Java	function	DeleteBlock(eBlockType, int)	deleteBlock(eBlockType, int)
Java	function	GetDiagnosticInfo()	getDiagnosticInfo()
Java	function	GetSystemStatusList(int, int)	getSystemStatusList(int, int)
		BasicInfoResult.Ordernummer()	BasicInfoResult.getOrdernumber()
		BasicInfoResult.ModuleVersion ()	BasicInfoResult.getModuleVersion ()
		BasicInfoResult.FirmwareVersion ()	BasicInfoResult.getFirmwareVersion()
		BasicInfoResult.Name ()	BasicInfoResult.getName()
		CPUModeInfoResult.CPUModeInfo()	CPUModeInfoResult.getCPUModeInfo()
		CPUModeInfoResult.CPUStateInfo()	CPUModeInfoResult.getCPUStateInfo()
		LEDInfoResult.LEDInfo()	LEDInfoResult.getLEDInfo()
		OperationResult.InnerException()	OperationResult.getInnerException()
		OperationResult.Message()	OperationResult.getMessage()
		OperationResult.Quality ()	OperationResult.getQuality()

Haben Sie Fragen?

Rufen Sie uns bitte unter der Hotline +49 421 98970330 an, oder senden Sie Ihre Frage an support@indi-systems.de.

Wir werden Ihr Anliegen in kürzester Zeit bearbeiten oder direkt beantworten.