



Dokumentation

PLCcom-Schnittstelle

für das File-System

PLCcom für S7 Version 10

Indi.An Systems GmbH

Flughafenallee 3

28199 Bremen

Deutschland

support@indi-systems.de

Tel + 49 421-989703-30

Fax + 49 421-989703-39

Inhaltsverzeichnis

Was ist die PLCcom-Schnittstelle in das Filesystem?	3
Wie muss ich die Schnittstelle vorbereiten?	4
Für welche Zwecke können Sie die Schnittstelle einsetzen?	5
Einstellungen vornehmen	6
Codebeispiel Einrichtung Dataserver mit Loggingfunktionalität	7
Datenarchivierung	10
Aktuelles Datenabbild	11
Aufbau Datenabbild im Textformat	11
Aufbau Datenabbild im XML-Format	12
Ver- und Entschlüsselung	13
Haben Sie Fragen?	15

Was ist die PLCcom-Schnittstelle in das Filesystem?

Die Schnittstelle für das Dateisystem wird von der PLCcom-Library in Verbindung mit dem PLCcom-Dataserver zur Verfügung gestellt.

Der PLCcom-Anwender kann innerhalb der Schnittstelle über folgende Funktionalitäten bereit verfügen:

1. Daten-Archivierer
Hier werden alle im PLCcom-Dataserver auflaufenden Daten der Endgeräte in fortlaufende Dateien tagesweise in das Filesystem geschrieben. Die maximale Dateigröße sowie die max. Anzahl der Dateien sind einstellbar.
2. Daten-Abbild
Komplettes Abbild der Variablen mit aktuellem Wert und Zeitpunkt der letzten Änderung in einer Datei pro Datenquelle
3. Optionale Verschlüsselung
Auflaufende Daten werden verschlüsselt abgelegt. Methoden und Funktionen zur einfachen Entschlüsselung sind beigefügt.
4. Automatische Reorganisation der Daten, wenn gewünscht

Zum Beispiel kann über diese Schnittstelle eine externe Applikation auf den Datenbestand der eingesetzten Software zugreifen, und auch auf historische Daten zugreifen.

Für einen optimalen Schutz können sämtliche Daten auch AES-verschlüsselt abgelegt werden. Hier hat dann nur der Anwender/Applikation Zugriff, der auch über den frei festlegbaren Verschlüsselungskey verfügt.

Wie muss ich die Schnittstelle vorbereiten?

Sie müssen dem PLCcom- System einen festen erreichbaren Ordner im Dateisystem bereitstellen. Der Ordner muss vorhanden sein, das System wird aber auch versuchen den Ziel-Ordner anzulegen wenn er nicht vorhanden ist.

Die Prozesse benötigen innerhalb dieses Ordners Lese-, Schreib- und Löschrchte. Stehen diese Rechte nicht zur Verfügung, können Fehlfunktionen bzgl. Der Schnittstellenprozesse auftreten.

Im Besonderen müssen die Löschrchte vorhanden sein, ansonsten können die Reorganisationsfunktionen alte Dateien nicht wegräumen, das kann dann durch Volllaufen des Datenträgers Probleme bis zum Betriebssystemabsturz nach sich ziehen.

Das gleich gilt, wenn die Dateien von externen Prozessen gesperrt werden. Bitte prüfen Sie auch während des Betriebes, dass ausreichend Speicherplatz auf dem Zieldatenträger zur Verfügung steht.

Bitte beachten Sie bei Vorbereitung des Zielsystems: Die Archivierung in das Dateisystem ist hoch performant und wurde mit handelsüblicher Hardware und einer Datenmenge von 1000 Variablenänderungen pro 100ms problemlos getestet.

Sollte aber aus unterschiedlichen Gründen das Problem auftreten, dass das PLCcom-System höhere Datenmengen bereitstellt, als in das Archiv geschrieben werden können, so speichert das System diese Daten im Arbeitsspeicher zwischen und arbeitet diese Daten nacheinander ab. Wenn allerdings diese intern gespeicherte Datenmenge einen Schwellwert von 50000 Datensätze pro Datenquelle übersteigt, werden keine neuen Datensätze angenommen, sondern verworfen.

Das geschilderte Problem kann z.B. bei anderweitig ausgelasteter Hardware oder zu wenig bereitgestellten Ressourcen (z.B. virtueller Server) auftreten.

Bitte überprüfen Sie aus diesem Grunde vor produktiver Nutzung die Auslastung des Zielsystems. Außerdem ist auf dem Zieldatenträger eine ausreichende Menge an freiem Speicherplatz zur Verfügung zu stellen. Eine allgemeine Empfehlung kann nur insoweit getroffen werden, dass 20GB freier Speicherplatz zur Verfügung stehen sollte, eine genaue Angabe hängt von der Anzahl benutzter Variablen sowie der Intensität und Menge von Wechselvorgängen der Variablenwerte ab.

Für welche Zwecke können Sie die Schnittstelle einsetzen?

Für die Benutzung der Schnittstelle gibt es unzählige Einsatzmöglichkeiten. Sie steht Ihnen immer dann zur Verfügung, wenn Sie von ‚Außerhalb‘ auf das PLCcom-System zugreifen möchten.

Außerdem können Sie die Daten für externe Auswertungen, Charts, Reports oder einfach für Archivierungszwecke nutzen.

Zum Beispiel kann über diese Schnittstelle eine selbst programmierte Applikation auf den Datenbestand des Systems zugreifen und auswerten.

Einstellungen vornehmen

Sie können für die Files-System-Schnittstelle folgende Einstellungen vornehmen:

1. Zielpfad
2. Ein- oder Ausschalten der Archiv-Funktionalität
3. (Option) Maximale Anzahl Archiv-Dateien innerhalb des Zielordners
Wird die maximale Anzahl überschritten, werden die ältesten Archiv-Dateien automatisch gelöscht
4. (Option) Das maximale Alter der Archiv-Dateien nach der Erstellung in Stunden
Wird das angegebene Alter der Archiv-Dateien überschritten, so werden diese Dateien automatisch gelöscht
5. Maximale Größe der Archiv-Dateien in MB.
Wenn die angegebene Größe erreicht wurde, wird eine neue Datei mit nachfolgender Sequence-Nummer geschrieben. Sollte keine Größe angegeben werden, liegt die maximale Dateigröße bei 2GB (nicht bei Windows x64).
6. Ein- oder Ausschalten der automatischen Reorganisation innerhalb des Archivs.
Bei Deaktivierung werden die aufgeführten Löschfunktionen nicht ausgeführt. In diesem Falle muss der Benutzer die Dateien selbst löschen.
7. Ein- oder Ausschalten der Bereitstellung eines Variablenabbildes (Image)
Wahlweise im Flat (.csv)- oder XML-Format.
8. Ein- oder Ausschalten der Dateiverschlüsselung
9. (Option) Übergabe eines Passwortes zur Verschlüsselung. Bei Einschalten der Verschlüsselung ist die Passwort-Übergabe Pflicht. Bitte heben Sie das Passwort für andere unzugänglich auf und geben Sie es nicht weiter.

Codebeispiel Einrichtung Dataserver mit Loggingfunktionalität

CSharp

```

PLCcom.PLCComDataServer.PLCComDataServer myDataServer = null;
PLCcomDevice Device = null;

//Enter your Username + Serial here
Console.WriteLine("Please enter your user name");
authentication.User = Console.ReadLine();
Console.WriteLine("Please enter your user serial key");
authentication.Serial = Console.ReadLine();

Console.WriteLine("Start Connect to TCPIP device..");
//Create an PLCcom-Device instance
Device = new TCP_ISO_Device("192.168.1.100", 0, 2, ePLCType.S7_300_400_compatibel);

//set autoconnect to true and idle time till disconnect to 10000 milliseconds
Device.setAutoConnect(true, 10000);

//Create an instance depending on the device type
Console.WriteLine("Create DataServer PLCDataServerTCP1..");
myDataServer = new PLCcom.PLCComDataServer.PLCComDataServer_TCP("PLCDataServerTCP1", (TCP_ISO_Device)Device, 500);

//register incoming events
//register events
myDataServer.OnConnectionStateChange += new
PLCcom.PLCComDataServer.PLCComDataServer.ConnectionStateChangeEventHandler(myDataServer_OnConnectionStateChange);
myDataServer.OnReadDataResultChange += new
PLCcom.PLCComDataServer.PLCComDataServer.ReadDataResultChangeEventHandler(myDataServer_OnReadDataResultChange);
myDataServer.OnIncomingLogEntry += new
PLCcom.PLCComDataServer.PLCComDataServer.OnIncomingLogEntryDelegate(myDataServer_OnIncomingLogEntry);

//define new request
Console.WriteLine("Create new Request Read 4 Bytes from DB1 at address 0 ...");
ReadDataRequest RequestItem1 = new ReadDataRequest(eRegion.DataBlock, //Region
1, //datablock
0, //startAddress
eDataType.BYTE, //target data type
4); //Quantity

//add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem1, "Request1");

//define new request
Console.WriteLine("Create new Request Read 10 DWord from Flags_Markers at address 4 ...");
ReadDataRequest RequestItem2 = new ReadDataRequest(eRegion.Flags_Markers, //Region
0, //datablock
4, //startAddress
eDataType.DWORD, //target data type
4); //Quantity

//add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem2, "Request2");

//add on or more Loggingkonnektoren with logging and writing of a data image into filesystem or database
//in this case create a new FileSystemConnector instanc
PLCcom.ExternalLogging.LoggingConnector con = new
    PLCcom.ExternalLogging.FileSystemConnector(System.Threading.Thread.GetDomain().BaseDirectory, //Target folder
    "FileSystemConnector1", //unique connector name
    ';', //text separator recommendation ';'
    true, //activate progressive logging
    true, //activate image writing
    PLCcom.ExternalLogging.eImageOutputFormat.dat, //output format .dat or .xml
    10, //restrict the maximum number of files. -1 = Disabled.
    24, //restrict the maximum age of files. -1 = Disabled.
    30, //You can restrict the maximum size of files. -1 = Disabled.
    string.Empty); //If you enter an encryption password, the data is stored in encrypted form.
//add Connector to Dataserver
myDataServer.AddOrReplaceLoggingConnector(con);

//start PLCcom data server
Console.ReadLine();

//stop PLCcom data server
myDataServer.StopServer();

```

Visual Basic

```

Dim myDataServer As PLCcom.PLCComDataServer.PLCComDataServer = Nothing
Dim Device As PLCcomDevice = Nothing
'Enter your Username + Serial here
Console.WriteLine("Please enter your user name")
authentication.User = Console.ReadLine()
Console.WriteLine("Please enter your user serial key")
authentication.Serial = Console.ReadLine()

Console.WriteLine("Start Connect to TCP/IP device...")
'Create an PLCcom-Device instance
Device = New TCP_ISO_Device("192.168.1.2", 0, 2, ePLCType.S7_300_400_compatibel)

'set autoconnect to true and idle time till disconnect to 10000 milliseconds
Device.setAutoConnect(True, 10000)

'Create an instance depending on the device type
Console.WriteLine("Create DataServer PLCDataServerTCP1...")
myDataServer = New PLCcom.PLCComDataServer.PLCComDataServer_TCP("DataServerTCP1", DirectCast(Device, TCP_ISO_Device), 500)

'register incoming events
AddHandler myDataServer.OnConnectionStateChange, AddressOf myDataServer_OnConnectionStateChange
AddHandler myDataServer.OnReadDataResultChange, AddressOf myDataServer_OnReadDataResultChange
AddHandler myDataServer.OnIncomingLogEntry, AddressOf myDataServer_OnIncomingLogEntry

Console.WriteLine("Create new Request Read 4 Bytes from DB1 at address 0 ...")
'Parameter:
'-Region
'-datablock
'-startAddress
'-target data type
'-Quantity
Dim RequestItem1 As New ReadDataRequest(eRegion.DataBlock, 1, 0, eDataType.BYTE, 4)
'add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem1, "Request1")

Console.WriteLine("Create new Request Read 10 DWord from Flags_Markers at address 4 ...")
'Parameter:
'-Region
'-datablock
'-startAddress
'-target data type
'-Quantity
Dim RequestItem2 As New ReadDataRequest(eRegion.Flags_Markers, 0, 4, eDataType.DWORD, 4)
'add new request to plccom data server
myDataServer.AddReadDataRequest(RequestItem2, "Request2")

'add on or more Loggingkonnektoren with logging and writing of a data image into filesystem or database
'Parameter:
'-Target folder
'-unique connector name
'-text separator recommendation ';'
'-activate progressive logging
'-activate image writing
'-output format .dat or .xml
'-restrict the maximum number of files. When the value is exceeded the old files are automatically deleted. -1 = Disabled.
'-restrict the maximum age of files. When the value is exceeded the old files are automatically deleted. -1 = Disabled.
'-You can restrict the maximum size of files. When the value is exceeded the old files are deleted. -1 = Disabled.
'-If you enter an encryption password, the data is stored in encrypted form.
Dim con As PLCcom.ExternalLogging.LoggingConnector = New
PLCcom.ExternalLogging.FileSystemConnector(System.Threading.Thread.GetDomain().BaseDirectory,
    "FileSystemConnector1", _
        ";"c, _
        True, _
        True, _
        PLCcom.ExternalLogging.eImageOutputFormat.dat, _
        10, _
        24, _
        30, _
        String.Empty)

'add Connector to Dataserver
myDataServer.AddOrReplaceLoggingConnector(con)

'start PLCcom data server
myDataServer.StartServer()
Console.ReadLine()
'stop PLCcom data server
myDataServer.StopServer()

```


Java

```

PLCComDataServer myDataServer = null;
PLCComDevice Device = null;

BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
//Enter your Username + Serial here
System.out.println("Please enter your user name");
authentication.User(input.readLine());
System.out.println("Please enter your user serial key");
authentication.Serial(input.readLine());

System.out.println("Start Connect to TCPIP device...");
//Create an PLCCom-Device instance
Device = new TCP_ISO_Device("192.168.1.100", 0, 2, ePLCType.S7_300_400_compatibel);

//set autoconnect to true and idle time till disconnect to 10000 milliseconds
Device.setAutoConnect(true, 10000);

//Create an instance depending on the device type
System.out.println("Create DataServer PLCDataServerTCP1...");
myDataServer = new PLCComDataServer_TCP("PLCDataServerTCP1", (TCP_ISO_Device)Device, 500);

//register incoming events
// register Connection state change event
myDataServer.connectionStateChangeNotifier = new ConnectionStateChangeNotifier(this);
// register incoming log event
myDataServer.incomingLogEntryEventNotifier = new IncomingLogEntryEventNotifier(this);
// register change ReadDataResult event
myDataServer.readDataResultChangeNotifier = new ReadDataResultChangeNotifier(this);

//define new request
System.out.println("Create new Request Read 4 Bytes from DB1 at address 0 ...");
ReadDataRequest RequestItem1 = new ReadDataRequest(eRegion.DataBlock, //Region
1, //datablock
0, //startAddress
eDataType.BYTE, //target data type
4); //Quantity

//add new request to plccom data server
myDataServer.addReadDataRequest(RequestItem1, "Request1");

//define new request
System.out.println("Create new Request Read 10 DWord from Flags_Markers at address 4 ...");
ReadDataRequest RequestItem2 = new ReadDataRequest(eRegion.Flags_Markers, //Region
0, //datablock
4, //startAddress
eDataType.DWORD, //target data type
4); //Quantity

//add new request to plccom data server
myDataServer.addReadDataRequest(RequestItem2, "Request2");

//add on or more Loggingkonnektoren with logging and writing of a data image into filesystem or database
//in this case create a new FileSystemConnector instance
LoggingConnector con = new FileSystemConnector(new File(".").getAbsolutePath(), //Target folder
"FileSystemConnector1", //unique connector name
';', //text separator recommendation ';'
true, //activate progressive logging
true, //activate image writing
eImageOutputFormat.dat, //output format .dat or .xml
10, //restrict the maximum number of files. -1 = Disabled.
24, //restrict the maximum age of files. -1 = Disabled.
30, //You can restrict the maximum size of files. -1 = Disabled.
"");//If you enter an encryption password, the data is stored in encrypted form

//add Connector to Dataserver
myDataServer.addOrReplaceLoggingConnector(con);

//start PLCcom data server
myDataServer.startServer();
input.readLine();

//stop PLCcom data server
myDataServer.stopServer();

```

Datenarchivierung

Mit der Daten-Archivierer-Funktionalität werden alle Datenänderungen in das Dateisystem protokolliert. Sie können diese Daten für beliebige Zwecke nutzen.

Hierzu werden Dateien nach folgender Namenskonvention angelegt:

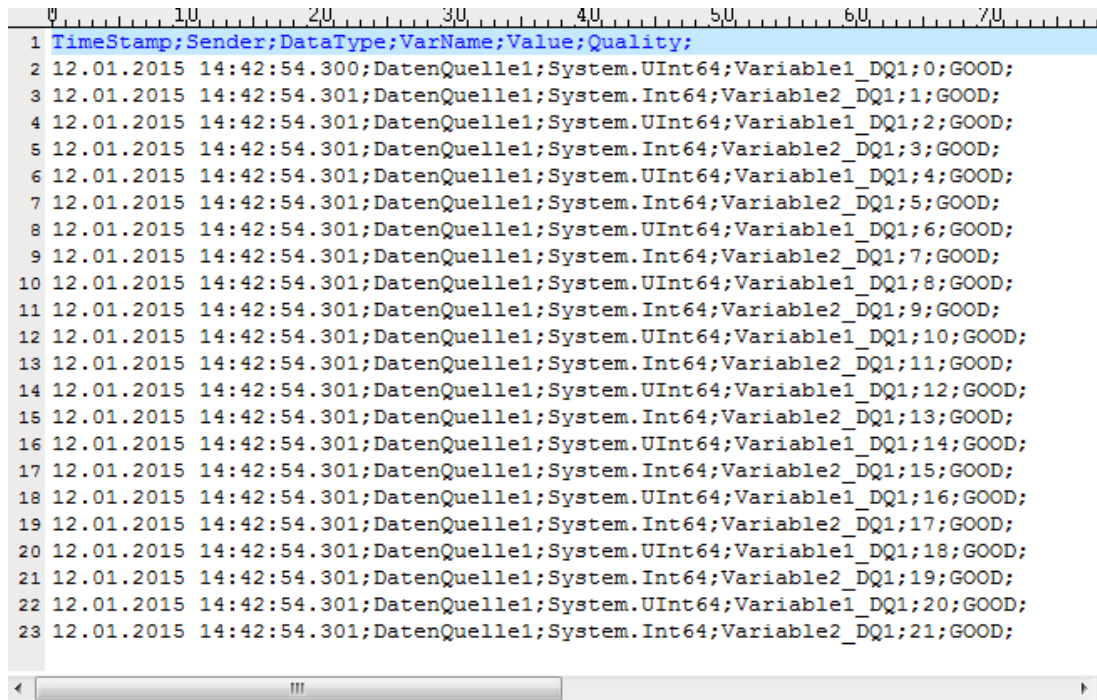
<datum>_<zähler 5-stellig>_<Datenquellenname>.log

z.B. 20140112_00001_DataSource1.log

Optional verschlüsselte Archivfiles tragen den gleichen Dateinamen mit der Endung *.logx

Wenn die Datei die festgelegte max. Größe oder 2GB (nicht bei Windows x64) erreicht, wird eine neue Datei mit nachfolgendem Zähler angelegt.

Die Felder in den Dateien sind mit Semikolon separiert, in der ersten Zeile befinden die Spaltenbezeichnungen.



```
1 TimeStamp;Sender;DataType;VarName;Value;Quality;
2 12.01.2015 14:42:54.300;DatenQuelle1;System.UInt64;Variable1_DQ1;0;GOOD;
3 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;1;GOOD;
4 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;2;GOOD;
5 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;3;GOOD;
6 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;4;GOOD;
7 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;5;GOOD;
8 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;6;GOOD;
9 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;7;GOOD;
10 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;8;GOOD;
11 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;9;GOOD;
12 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;10;GOOD;
13 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;11;GOOD;
14 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;12;GOOD;
15 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;13;GOOD;
16 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;14;GOOD;
17 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;15;GOOD;
18 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;16;GOOD;
19 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;17;GOOD;
20 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;18;GOOD;
21 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;19;GOOD;
22 12.01.2015 14:42:54.301;DatenQuelle1;System.UInt64;Variable1_DQ1;20;GOOD;
23 12.01.2015 14:42:54.301;DatenQuelle1;System.Int64;Variable2_DQ1;21;GOOD;
```

Es werden folgende Informationen geschrieben:

1. Zeitpunkt mit Millisekunden
2. Sender (Name der Datenquelle)
3. Datentyp
4. Name der Variable
5. Aktueller Wert
6. Qualität (GOOD wenn die PLCcom-Anbindung zum Zielgerät vorschriftsmäßig arbeitet oder BAD wenn Probleme (z.B. Timeouts) auftreten)

Aktuelles Datenabbild

Mithilfe des Datenabbildes können Sie über externe Applikationen auf den Variablenvorrat des PLCcom-Systems zugreifen und aktuelle Zustände auslesen.

Es gibt zwei Möglichkeiten das Datenabbild ausgeben zu lassen, im Textformat oder als XML-Datei.

Aufbau Datenabbild im Textformat

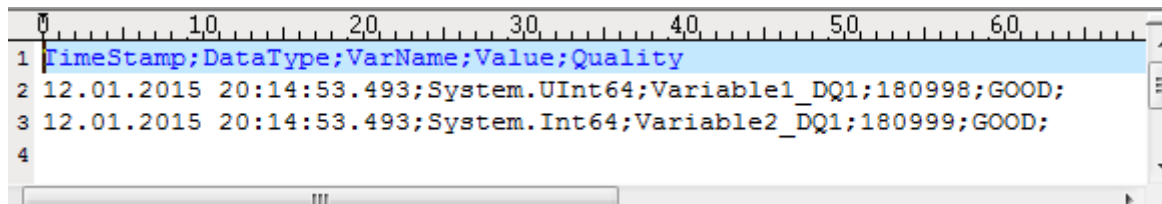
Hierzu werden Dateien nach folgender Namenskonvention angelegt:

<Datenquellenname>.dat

z.B. DataSource1.dat

Optional verschlüsselte Abbilddateien tragen den gleichen Namen mit der Endung *.datx

Die Felder in den Dateien sind mit Semikolon separiert, in der ersten Zeile befinden die Spaltenbezeichnungen.



```
0 10 20 30 40 50 60
1 TimeStamp;DataType;VarName;Value;Quality
2 12.01.2015 20:14:53.493;System.UInt64;Variable1_DQ1;180998;GOOD;
3 12.01.2015 20:14:53.493;System.Int64;Variable2_DQ1;180999;GOOD;
4
```

Für jede Variable existiert eine Zeile innerhalb der Datei. Es werden folgende Informationen geschrieben:

1. Zeitpunkt mit Millisekunden
2. Sender (Name der Datenquelle)
3. Datentyp
4. Name der Variable
5. Aktueller Wert
6. Qualität (GOOD wenn die PLCcom-Anbindung zum Zielgerät vorschriftsmäßig arbeitet oder BAD wenn Probleme (z.B. Timeouts) auftreten)

Aufbau Datenabbild im XML-Format

Hierzu werden Dateien nach folgender Namenskonvention angelegt:

<Datenquellenname>.xml

z.B. DataSource1.xml

Im xml-Format tragen optional verschlüsselte Abbilddateien tragen den gleichen Dateinamen mit gleicher Endung *.xml.

```

1 <?xml version="1.0" encoding="utf-16" standalone="yes"?>
2 <Image>
3   <variable isEncrypted="False" TimeStamp="12.01.2015 20:31:07.398" VarName="Variable1_DQ1" Value="261998" Quality="GOOD" />
4   <variable isEncrypted="False" TimeStamp="12.01.2015 20:31:07.398" VarName="Variable2_DQ1" Value="261999" Quality="GOOD" />
5 </Image>
    
```

Für jede Variable wird durch ein eigenes variable-Tag innerhalb des Image-Blocks dargestellt. Die Attribute des variable-Tag enthält folgende Informationen zu jeder Variable:

1. Flag, ob Daten verschlüsselt sind
2. Zeitpunkt mit Millisekunden
3. Sender (Name der Datenquelle)
4. Datentyp
5. Name der Variable
6. Aktueller Wert
7. Qualität (GOOD wenn die PLCcom-Anbindung zum Zielgerät vorschriftsmäßig arbeitet oder BAD wenn Probleme (z.B. Timeouts) auftreten).

Bei verschlüsselten Dateien ist das XML-Format weiterhin erhalten geblieben, lediglich die Feldinhalte sind verschlüsselt.

Ver- und Entschlüsselung

Um eine optimale Datensicherheit bereitstellen zu können, sind die PLCcom-Dateischnittstellen mit optionaler Verschlüsselung ausgerüstet.

Die Verschlüsselung wird direkt durchgeführt, wenn Sie die notwendigen Einstellungen innerhalb der Datenquellen eingerichtet und ein Passwort vergeben haben.

Bitte beachten Sie: Sie müssen bei der Einrichtung zur Verschlüsselung ein Passwort vergeben. Bitte heben Sie dieses Passwort gut auf! Sollte es abhandenkommen, können die verschlüsselten Dateien nicht mehr geöffnet werden.

Auch wir können in diesem Fall nicht mehr helfen.

Die Verschlüsselungsroutinen benötigen CPU-Ressourcen! Planen Sie bei Verwendung die Bereitstellung etwas höherer Rechenleistung ein, besonders wenn in sehr kurzen Abständen viele Variablen geändert werden. Sie können die Auslastung aber vorab im Editor während des Laufzeitmodus testen.

Zur Ver- und Entschlüsselung liegen Ihrer PLCcom-Library Methoden und Funktionen bei.

Beispiel:

CSharp

```
try
{
    // create a decryptor instance
    Decrypter dc = new Decrypter(txtInsertPassword.Text);

    // xml or dat file
    if (txtSourceFile.Text.ToLower().EndsWith(".xml"))
    {
        dc.DecryptPLCcomXmlFile(<SourceFile>, <TargetFile>);
    }
    else
    {
        dc.DecryptPLCcomDataFile(<SourceFile>, <TargetFile>);
    }
}
catch (Exception ex )
{
    MessageBox.Show(ex.Message + " Please check your Key or File", "", MessageBoxButtons.OK, MessageBoxIcon.Hand);
}
```

Visual Basic

```

Try
    ' create a decryptor instance
    Dim dc As New Decrypter(txtInsertPassword.Text)

    ' xml or dat file
    If txtSourceFile.Text.ToLower().EndsWith(".xml") Then
        dc.DecryptPLCcomXmlFile(<SourceFile>, <TargetFile>)
    Else
        dc.DecryptPLCcomDataFile(<SourceFile>, <TargetFile>)
    End If
Catch ex As Exception
    MessageBox.Show(ex.Message + " Please check your Key or File", "", MessageBoxButtons.OK, MessageBoxIcon.Hand)
End Try

```

Java

```

try {
    create a decryptor instance
    PLCCom.Decryptor dc = new Decryptor(txtInsertPassword.getPassword().toString());

    // xml or dat file
    if (txtSourceFile.getText().toLowerCase().endsWith(".xml")) {
        dc.decryptPLCcomXmlFile(txtSourceFile.getText(), txtTargetFile.getText());
    } else {
        dc.decryptPLCcomDataFile(txtSourceFile.getText(), txtTargetFile.getText());
    }
    try {
        JOptionPane.showMessageDialog(this,
            ResourceBundle.getBundle("logFileDecrypter_app.resources.resources")
                .getString("successful_saved") + System.getProperty("line.separator") + "File: "
                + txtTargetFile.getText(),
            "", JOptionPane.INFORMATION_MESSAGE);
    } finally {
        txtSourceFile.setText("");
        txtTargetFile.setText("");
        btnDefineTargetFile.setEnabled(false);
        btnDecrypt.setEnabled(false);
    }
} catch (InvalidKeyException eik) {
    // Invalid key => occurs usually by using of wrong key
    JOptionPane.showMessageDialog(this, eik.getClass().getName() + " " + eik.getMessage(), "",
        JOptionPane.ERROR_MESSAGE);
} catch (InvalidAlgorithmParameterException eia) {
    // Invalid or inappropriate algorithm parameters for => internal
    // Error
    JOptionPane.showMessageDialog(this, eia.getClass().getName() + " " + eia.getMessage(), "",
        JOptionPane.ERROR_MESSAGE);
} catch (IllegalBlockSizeException eib) {
    // The length of data provided to a block cipher is incorrect =>
    // internal Error
    JOptionPane.showMessageDialog(this, eib.getClass().getName() + " " + eib.getMessage(), "",
        JOptionPane.ERROR_MESSAGE);
} catch (BadPaddingException ep) {
    // The input data but the data is not padded properly => occurs
    // usually by using of wrong key
    JOptionPane.showMessageDialog(this,
        "Please check your Key " + ep.getClass().getName() + " " + ep.getMessage(), "",
        JOptionPane.ERROR_MESSAGE);
} catch (Exception ex) {
    JOptionPane.showMessageDialog(this, ex.getClass().getName() + " " + ex.getMessage(), "",
        JOptionPane.ERROR_MESSAGE);
}
}

```

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.