Documentation PLCcom filesystem interface

Version 2.0

Indi.Systems GmbH

Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de

Tel + 49 421-989703-30
Fax + 49 421-989703-39

Table of contents

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

What is the PLCcom filesystem interface?

The filesystem interface is provided by the PLCcom-Library in conjunction with the PLCcom-Dataserver.

With this interface the user can access following functionalities:

1. Data logging
   Log all data from devices connected to the dataserver into daily files in your filesystem. The maximum file size as well as number of files overall can be configured.

2. Data image
   A complete image of all variables with current values and timestamp of last change will be saved into one file per datasource.

3. Optional encryption
   Accumulating data will be saved encrypted. Methods and functions for easy decryption are enclosed.

4. Automatic reorganization of data, if desired.

For example, an external application can access data (historic data as well) of the software in use via this interface.

For best security all information can be saved AES-encrypted. In this case, only applications/users with knowledge of the encryption key can access said data.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

How do I prepare the interface?

You have to provide a fixed accessible folder for the PLCcom-system within your filesystem. If the folder can't be found, PLCcom will however try to create it in the desired location.

The process need read, write and delete permissions within the defined folder. If this permissions aren't granted the interface process might malfunction.

Especially delete permissions should be granted. Without this rights the process can't reorganize old files and therefore free disk space. This can lead to your system running out of space, and even crash your operating system. The same applies to other processes blocking these files.

Please check the remaining disk space continually, even during operation.

Important note:

The archiving processes are highly performant and were tested with standard hardware and a data volume of 1000 variable changes every 100ms without any problems.

Should however, for some reason, the PLCcom system provide more data than the archiving process can handle, it will buffer data into the main memory. The process will then work through these records one by one. This will only happen until a threshold of 50.000 records is reached. After this, no new records will be saved, but discarded instead.

This problem may, for example, occur in otherwise used hardware or too little resources provided for the archiving process (e.g. virtual server).

For this reason, we recommend checking the load of the system before going into productive use. You should also provide enough free disk space. The needed space highly depends on the number of variables as well as the update interval. For this reason it's difficult to give an accurate recommendation, but 20GB should be a good start.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Possible use cases for the filesystem interface

There are countless possible applications where you could use this interface. You can always use the interface if you need to access the PLCcom system from external applications.

Furthermore you can use the provided data for external analysis, charts, reports, or just for archiving purposes.

This way, an external application could access and analyze data from PLCcom.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Configure the filesystem interface

You can make the following settings for the filesystem interface:

1. Target path

2. Enable or disable archiving

3. (Optional) Maximum number of archive files within the target folder
   When this amount is reached, the oldest files will be deleted.

4. (Optional) „Time to live“ for archive files in hours
   When archive files exceed their time to live, they’ll be deleted.

5. Maximum size of archive files in MB
   When archive files reach this size, a new file with consecutive sequence number will be created. If no size is specified the maximum file size is 2GB (does not apply for Windows x64).

6. Enable or disable the automatic reorganization of archive files
   If deactivated, above mentioned delete functions won’t be executed. In this case the user has to delete the files himself.

7. Enable or disable the creation of a complete variable image
   You can choose between XML and CSV format.

8. Enable or disable encryption

9. (Optional) Set an encryption password
   If encryption is activated, the setting of a password is compulsory. Keep this password in a safe place.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Code example for dataserver with logging functionality

CSharp

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

```
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,                          //startAdress
                                                   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,                          //startAdress
                                                   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 immage 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 TCPIP 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
'-startAdress
'-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
'-startAdress
'-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 immage 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()
```

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

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,                          //startAdress
                                                   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,                          //startAdress
                                                   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();
```

Data archiving

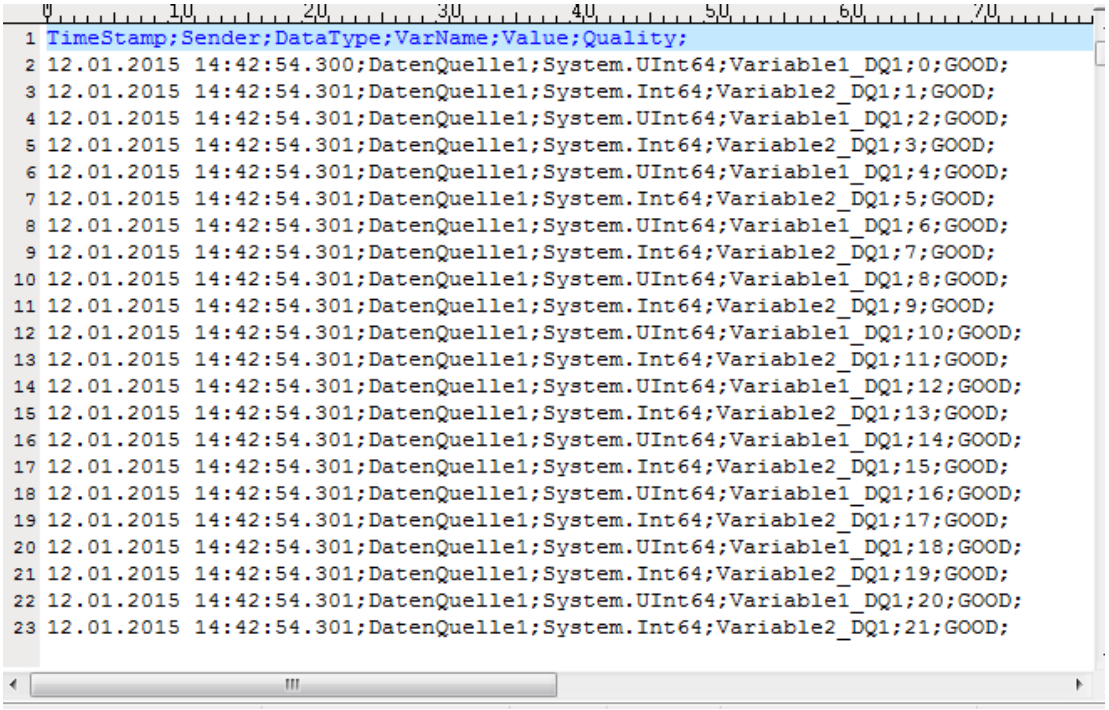
All changes of data can be logged into the filesystem with the data archiving option. You can use this data for any purpose.

For this purpose files with the following naming convention will be created:

<date>_<sequence_number_5_digits>_<datasourcename>.log
e.g. 20140112_00001_DataSource1.log

Encrypted archive files have the file extension *.logx

Datasets within an archive are separated by semicolons. The first row contains the column headers.



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

The following data will be archived:

1. Timestamp with milliseconds
2. Sender (datasource name)
3. Datatype
4. Variable name
5. Current value
6. Quality (GOOD if the communication between PLCcom and device works fine or BAD if problems occurred (e.g. timeouts)

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Data images

With the aid of an data image, you can access the variable pool of the PLCcom system from an external application.

There are two ways to export the image. A flat text file (.csv) or as a XML document.

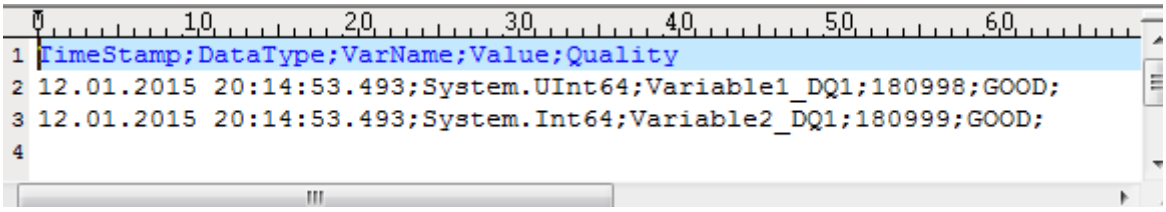
Structure of an CSV image

For this purpose files with the following naming convention will be created:

<datasourcename>.dat
e.g. DataSource1.dat

Encrypted archive files have the file extension *.datx

Datasets within an image are separated by semicolons. The first row contains the column headers.



The following data will be written for every variable:

1. Timestamp with milliseconds
2. Sender (datasource name)
3. Datatype
4. Variable name
5. Current value
6. Quality (GOOD if the communication between PLCcom and device works fine or BAD if problems occurred (e.g. timeouts)

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Structure of an XML image

For this purpose files with the following naming convention will be created:

<datasourcename>.xml
e.g. DataSource1.xml

For xml files there's no special extension for encrypted files, these also have the *.xml extension.

```
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>
```

For every variable there's a variable tag within the image tag. The attributes of every variable tag holds the data for the corresponding variable:

1. Flag to show if data is encrypted
2. Timestamp with milliseconds
3. Sender (datasource name)
4. Datatype
5. Variable name
6. Current value
7. Quality (GOOD if the communication between PLCcom and device works fine or BAD if problems occurred (e.g. timeouts)

For encrypted XML files, the XML structure was preserved. Just the data within each variable has been encrypted.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Encryption and decryption

To provide the best security every PLCcom interface is equipped with an optional AES-encryption.

The encryption is enabled as soon as you make the needed settings for the datasource, as well as set a password.

Please note:

You'll have to provide a password to enable encryption. Keep this password in a safe place. Without this password you won't be able to read your data. We also won't be able to help you in this case.

The encryption routines need CPU resources. If you enable encryption ensure that you have enough hardware resources. To estimate the performance impact of encryption, we recommend you to test your project in editor runtime mode first.

For encryption and decryption we provide you with suitable methods and functions in the PLCcom library.

Example:

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);
}
```

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

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);
}
```

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39

Any questions?

Please write an email to support@indi-systems.de.

We will process your request promptly or respond to you directly.

Indi.Systems GmbH
Universitätsallee 23
28359 Bremen
Germany
info@indi-systems.de
Tel + 49 421-989703-30
Fax + 49 421-989703-39