

# PLCProject

## Users Guide

Last Update 2020/10/16

This document describes the PLCProject commandline arguments and the global XML structure. The Windows application WPLCProject is selfexplaining, just browse the projects and select ,Export...' from the menu.

## Contents

[Commandline Arguments](#)

[XML Structure](#)

[XML tags and their attributes](#)

[Examples](#)

## Commandline Arguments

For both the Windows and the Linux operating system all arguments have the same syntax and meaning.

All arguments are expected to have the format

<key>=<value>

If an argument contains empty characters it has to be enclosed in apostrophes („...“). The keyname is not case-sensitive.

Key	Type	Meaning
xmlfile	Filename	The name of the XML formatted textfile to create. An existing file is overwritten.
projectfile	Filename	The name of the TIA projectfile. It must have the extension .ap14, .ap15, .ap15_1 or .ap16.
device	String	Select the controller device having the given name. The devicename is case-sensitive. All other devices defined in the project are ignored. This argument is optional.
group	String	Select the group having the given name. The groupname is case-sensitive. All other groups defined in the device except subgroups are ignored. Use „/“ as delimiter to address a nested group. This argument is optional and must be used together with the device-argument.
block	String	Select the block having the given name. The blockname is case-sensitive. All other blocks belonging to the device are ignored. This argument is optional and must be used together with the device-argument.

extags	Number	<p>A set of XML-tags to exclude as the sum of the following values:</p> <ul style="list-style-type: none"> <li>1 &lt;Blocks&gt;</li> <li>2 &lt;TechnoObjects&gt;</li> <li>4 &lt;Symbols&gt;</li> <li>8 &lt;Constants&gt;</li> <li>16 &lt;Types&gt;</li> <li>32 &lt;Header&gt;</li> <li>64 &lt;Comments&gt;</li> <li>128 &lt;Decls&gt;</li> <li>256 &lt;Input&gt;</li> <li>512 &lt;Output&gt;</li> <li>1024 &lt;InOut&gt;</li> <li>2048 &lt;Static&gt;</li> <li>4096 &lt;Temp&gt;</li> <li>8192 &lt;Constant&gt;</li> <li>16382 &lt;DeclComments&gt;</li> <li>32768 &lt;DeclValues&gt;</li> <li>65536 &lt;Code&gt;</li> <li>131072 &lt;Tags&gt;</li> </ul> <p>By default the XML-file contains all tags. E.g. if you are not interested in Temp- and constant-declarations of codeblocks, this value should be 12288 (4096 + 8192). This argument is optional.</p>
exbtypes	Number	<p>A set of blocktypes to exclude as the sum of the following values:</p> <ul style="list-style-type: none"> <li>1 OB</li> <li>2 FB</li> <li>4 FC</li> <li>8 DB</li> <li>16 UDT</li> <li>32 SFB</li> <li>64 SFB</li> <li>128 SDT</li> </ul> <p>By default the XML-file contains all blocks. E.g. if you are only interested in DB's and UDT's this value should be 231. This argument is optional.</p>
language	Number	<p>The local language identifier as a sequential index. By default this value is 0. If there are multilingual comments defined in the project, you can select the language by using another index. This argument is optional.</p>
mnemonic	Number	<p>The formal mnemonic for instructions and operands.</p> <ul style="list-style-type: none"> <li>0 International (Default value)</li> <li>1 German</li> </ul> <p>This argument is optional.</p>
serno	String	<p>Use this argument only once for licensing the product. The value is the serialnumber you received.</p>

## XML Structure

Tags marked with \* can have 0..n instances.

```
<Project>  
  <Device> *
```

```
<Device>  
  <Blocks>  
  <TechnoObjects>  
  <Symbols>  
  <Constants>  
  <Types>  
  <Tags>
```

```
<Blocks>  
  <Block> *  
  <Group> *
```

```
<TechnoObjects>  
  <Block> *
```

```
<Symbols>  
  <Group> *  
  <Symbol> *
```

```
<Constants>  
  <Constant> *
```

```
<Types>  
  <Group> *  
  <Block> *
```

```
<Tags>  
  <Tagtable> *  
  <Group> *
```

```
<Tagtable>  
  <Tag> *
```

```
<Group>  
  <Block> *  
  <Tagtable> *  
  <Group> *
```

```
<Block>  
  <Header>  
  <Comment>  
  <Decls>  
  <Code>
```

<Comment>  
    <Line> \*

<Decls>  
    <Decl> \*  
    <Input>  
    <Output>  
    <InOut>  
    <Static>  
    <Temp>  
    <Return>  
    <Constant>  
    <DeclComments>  
    <DeclValues>

<Decl> <Input> <Output> <InOut> <Static> <Temp> <Return> <Constant>  
    <Decl> \*

<DeclComments>  
    <DeclComment> \*

<DeclValues>  
    <DeclValue> \*

<Code>  
    <Operands>  
    <Segment> \*

<Operands>  
    <Operand> \*

<Segment>  
    <Comment>  
    <Instr> \*  
    <Part> \*  
    <Steps>  
    <Transitions>

<Part>  
    <Pin> \*

<Steps>  
    <Step> \*

<Transitions>  
    <Transition> \*

<Step>  
    <Transitions>  
    <Actions>

<Transition>  
    <Steps>  
    <Condition>

<Actions>  
    <Action> \*

<Action>  
    <Param> \*

## XML tags and their attributes

Tagname	Attributename	Meaning														
<Project>	Name Version	Short projectname Projectversion, Value is 1400, 1500, 1501 or 1600														
<Device>	Name Type Family	Name of the device Devicetype Devicefamily, possible values are 'Controller' or 'HMI'														
<Block>	Name Address Title OptAccess Created Modified InstanceOf Protection	Symbolic name of the block, e.g. 'Main' Blockaddress, e.g. 'OB 1' Blocktitle True, if the block has attribute ,Optimized Access' Date and time the block was created Date and time the block was last modified The symbolic blockname of the instantiated block Possible values are 'KnowHow' or 'Write'														
<Decl>	Name Type Id  Address Size Flags  Remanence	Name of the declaration Datatype including arraybounds Numerical id, referred by Path-attribute of comment and value Bitaddress of the declaration Used memorysize of the declaration in bit Set of flags indicating specific boolean attributes like ,HMI visible' Possible values are 'Retain' or 'Classic'														
<DeclComment>	Path Text	Commentaddress including id's and arrayindex, e.g. 51:54[5] Commenttext														
<DeclValue>	Path Value	Valueaddress including id's and arrayindex, e.g. 51:54[5] Default value of the declaration														
<Symbol>	Name Address Type Comment	Symbolname, e.g. 'Temperature' Symboladdress, e.g. '%IW100' Datatype of the symbol Symbolcomment														
<Constant>	Name Value Type Comment	Constantname Constantvalue Datatype of the constant Comment for the constant														
<Header>	Author Family Name Version	The author of a block Blockfamily Userdefined blockname Blockversion, default is 0.1														
<Operand>	Id Type	Numerical identifier referred by instructions Kind of operand <table style="margin-left: 40px;"> <tr><td>1</td><td>Symbol</td></tr> <tr><td>2</td><td>Global PLC-Address</td></tr> <tr><td>3</td><td>Local variable</td></tr> <tr><td>4</td><td>Constant</td></tr> <tr><td>5</td><td>Label</td></tr> <tr><td>6</td><td>Blockname</td></tr> <tr><td>7</td><td>Blockaddress</td></tr> </table>	1	Symbol	2	Global PLC-Address	3	Local variable	4	Constant	5	Label	6	Blockname	7	Blockaddress
1	Symbol															
2	Global PLC-Address															
3	Local variable															
4	Constant															
5	Label															
6	Blockname															
7	Blockaddress															

	Value	9 Variable of a datablock The operand's value depends on its type. For Variables this string contains their identifiers concatenated with ':'. The index of an arrayelement in [...] refers to another <Operand>, it is not the index itself.
	DBName	Name of the datablock for Type 9
<Segment>	Number Title Type	Number 1..n Segment-Title Programminglanguage STL LAD FBD SCL GRAPH
<Part>	Name Id BlockId DBId	Functionname of a part in LAD or FBD Numerical identifier of the part referred by <Pin> Reference to a called block (if Name is 'CALL') Reference to an instance DB (if Name is 'CALL')
<Pin>	Name Type  ConnType  OpId PartId PinName Negated	Visible name of the pin inside of <Part> Location (access) of the pin In Left side Out Right side Type of the pin's connection 0 No connection 1 Operand connected 2 Part connected Reference to an operand (see Id in <OpList>) Reference to a part (see Id in <Part>) Name of the connected pin Indicates if the pin is negated (true or false)
<Instr>	Code Op OpId Comment	Instructioncode Instructionoperand Reference to the operand (see Id in <OpList>) Instructioncomment
<Action>	Interlock Event Operation OpId SOpld Function	Boolean interlockstate Kind of the action's event Kind of the action's operation Reference to an operand Reference to a second operand for assignments Type of function 1 Simple Assignment 2 CALL
<Condition>	Type	See <Segment>
<Tagtable>	Name	Name of the HMI tagtable
<Tag>	Name Type PLCName PLCAddress Comment	Name of the HMI tag Datatype of the HMI tag Name of the connected controller operand (optional) Address of the connected controller operand (optional) Optional comment

## Examples

The following command exports all data of a project to the file c:\xml\project01.xml.

```
PLCProject    xmlfile=c:\xml\project01.xml
              projectfile=c:\tia\project01\project01.ap16
```

The following command exports only the symbols on device ,PLC 1'.

```
PLCProject    xmlfile=c:\xml\project01.xml
              projectfile=c:\tia\project01\project01.ap16
              „device=PLC 1“
              extags=27
```

The following command exports all datablocks of group ,opc ua' for device ,PLC 1'.

```
PLCProject    xmlfile=c:\xml\project01.xml
              projectfile=c:\tia\project01\project01.ap16
              „device=PLC 1“
              „group=opc ua“
              exbtypes=247
```

The following command exports the block ,My function' on device ,PLC 1'.

```
PLCProject    xmlfile=c:\xml\project01.xml
              projectfile=c:\tia\project01\project01.ap16
              „device=PLC 1“
              „block=My function“
```