```
FUNCTION_BLOCK write_to_XML_Event
VAR_INPUT
      LogInput   : ARRAY[1..20] OF Datensatz;
      bExecute   : BOOL := FALSE;
      sFilePath  : T_MaxString := 'C:\temp_Yizhen\Test.xml';
//disable for CE
  //sFilePath    : T_MaxString := '\Hard Disk\Test.xml';
      //enable for CE
END_VAR
VAR_OUTPUT
      WriteFinishFlag : BOOL;
END_VAR
VAR
      bBusy : BOOL;
      bExecuteValue         : BOOL;
      XmlSrvWriteValue : FB_XmlSrvWrite;
     sXPathData        : T_MaxString := '/datafromroom/datapart';
      INPUT: int;
      Error: BOOL;
      ErrID: UDINT;
      MyTimer: getsystemtime;
      DownTrigDetect: F_TRIG;
      DownTrigDetectHeader: F_TRIG;
      Header: XMLHeader;
      XmlSrvWriteHeader: FB_XmlSrvWrite;
      sXPathHeader: T_MaxString := '/datafromroom/Header';
END_VAR

(*    it creates an XML-file under the path C:\Test.xml and writes
value1 to it.
      Existing file will be overwritten.
      FUNCTIONBLOCK: FB_XmlSrvWrite *)


//reset buffer reading flag
WriteFinishFlag := FALSE;

//detect the trigger:true to false
DownTrigDetect(CLK:=XmlSrvWriteValue.bBusy);
DownTrigDetectHeader(CLK:=XmlSrvWriteHeader.bBusy);


//write the xml header
XmlSrvWriteHeader(
      nMode := XMLSRV_ADDMISSING,
  pSymAddr := ADR(Header),
  cbSymSize      := SIZEOF(Header),
  sFilePath      := sFilePath,
  sXPath    := sXPathHeader,
  bExecute := bExecute
);
```

```
//when xml header is done, start to write the values
IF DownTrigDetectHeader.Q THEN
     bExecuteValue := TRUE;
END_IF

//write values to xml file
XmlSrvWriteValue(
  nMode         := XMLSRV_ADDMISSING,
  pSymAddr := ADR(LogInput),
  cbSymSize     := SIZEOF(LogInput),
  sFilePath     := sFilePath,
  sXPath   := sXPathData,
  bExecute := bExecuteValue
);

Error := XmlSrvWriteValue.bError;
ErrID := XmlSrvWriteValue.nErrId;
bBusy := XmlSrvWriteValue.bBusy;


//reset parameters
IF DownTrigDetect.Q THEN
     WriteFinishFlag := TRUE;
     bExecute := FALSE;
     bExecuteValue := FALSE;
END_IF
```