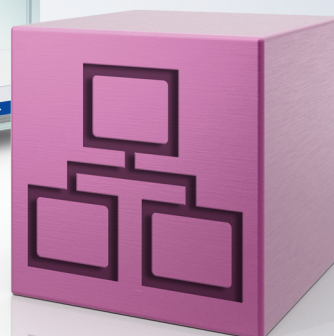
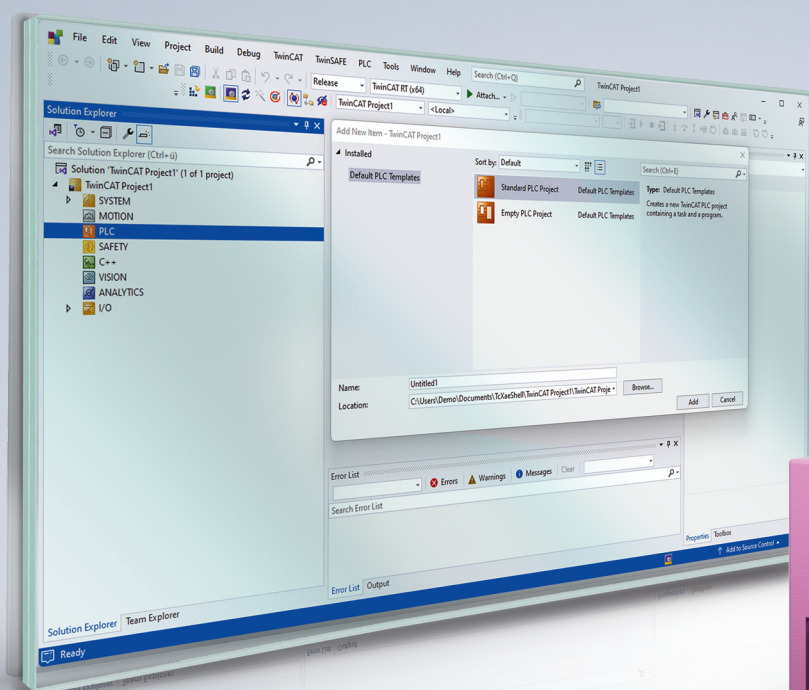


手册 | ZH

TF6421

TwinCAT 3 | XML Server



目录

1 前言	5
1.1 文档说明	5
1.2 安全信息	5
1.3 信息安全说明	6
2 概述	7
3 安装	8
3.1 系统要求	8
3.2 安装	8
3.3 授权	11
3.4 Windows CE 安装	13
4 PLC 功能库	16
4.1 概述	16
4.2 功能块	17
4.2.1 FB_XmlSrvRead	17
4.2.2 FB_XmlSrvWrite	18
4.2.3 FB_XmlSrvReadByName	19
4.2.4 FB_XmlSrvWriteByName	20
4.2.5 FB_XmlSrvEnableObfuscation	21
4.2.6 FB_XmlSrvDisableObfuscation	22
4.2.7 FB_XmlSrvIsObfuscationEnabled	23
4.2.8 FB_XmlSrvObfuscationExport	24
4.3 全局常量	25
4.3.1 全局变量	25
4.3.2 功能库版本	25
5 示例	26
5.1 入门	26
5.2 功能块	27
5.3 更多示例	29
5.4 生产示例	31
6 附录	34
6.1 TC3 XML Server 错误代码概述	34
6.2 ADS 返回代码	34
6.3 TwinCAT 3 XML Server 的内部错误代码	37
6.4 其他注册表项	38
6.5 FAQ - 常见问题及其解答	38
7 技术支持和服务	40

1 前言

1.1 文档说明

本说明仅适用于熟悉国家标准且经过培训的控制和自动化工程专家。

在安装和调试组件时，必须遵循文档和以下说明及解释。

操作人员应具备相关资质，并始终使用最新的生效文档。

相关负责人员必须确保所述产品的应用或使用符合所有安全要求，包括所有相关法律、法规、准则和标准。

免责声明

本文档经过精心准备。然而，所述产品正在不断开发中。

我们保留随时修改和更改本文档的权利，恕不另行通知。

不得依据本文档中的数据、图表和说明对已供货产品的修改提出赔偿。

商标

Beckhoff®、ATRO®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、MX-System®、Safety over EtherCAT®、TC/BSD®、TwinCAT®、TwinCAT/BSD®、TwinSAFE®、XFC®、XPlanar® 和 XTS® 是 Beckhoff Automation GmbH

的注册商标并由其授权使用。本出版物中所使用的其它名称可能是商标名称，任何第三方出于其自身目的使用它们可能会侵犯商标所有者的权利。



EtherCAT® 是注册商标和专利技术，由 Beckhoff Automation GmbH 授权使用。

版权所有

© Beckhoff Automation GmbH。

未经明确授权，不得复制、分发、使用和传播本文档内容。

违者将被追究赔偿责任。Beckhoff Automation GmbH 保留所有发明、实用新型和外观设计专利权。

第三方商标

本文档可能使用了第三方商标。有关商标信息，可以访问：<https://www.beckhoff.com/trademarks>。

1.2 安全信息

安全规范

为了确保您的使用安全，请务必仔细阅读并遵守本文档中每个产品的安全使用说明。

责任免除

所有组件在供货时都配有适合应用的特定硬件和软件配置。严禁未按文档所述修改硬件或软件配置，否则，德国倍福自动化有限公司对由此产生的后果不承担责任。

人员资格

本说明仅供熟悉适用国家标准的控制、自动化和驱动工程专家使用。

警示性词语

文档中使用的警示信号词分类如下。为避免人身伤害和财产损失，请阅读并遵守安全和警告注意事项。

人身伤害警告

⚠ 危险

存在死亡或重伤的高度风险。

⚠ 警告

存在死亡或重伤的中度风险。

⚠ 谨慎

存在可能导致中度或轻度伤害的低度风险。

财产或环境损害警告

注意

可能会损坏环境、设备或数据。

操作产品的信息



这些信息包括：
有关产品的操作、帮助或进一步信息的建议。

1.3 信息安全说明

Beckhoff Automation GmbH & Co.KG (简称 Beckhoff) 的产品，只要可以在线访问，都配备了安全功能，支持工厂、系统、机器和网络的安全运行。尽管配备了安全功能，但为了保护相应的工厂、系统、机器和网络免受网络威胁，必须建立、实施和不断更新整个操作安全概念。Beckhoff 所销售的产品只是整个安全概念的一部分。客户有责任防止第三方未经授权访问其设备、系统、机器和网络。它们只有在采取了适当的保护措施的情况下，方可与公司网络或互联网连接。

此外，还应遵守 Beckhoff 关于采取适当保护措施的建议。关于信息安全和工业安全的更多信息，请访问本公司网站 <https://www.beckhoff.com/secguide>。

Beckhoff 的产品和解决方案持续进行改进。这也适用于安全功能。鉴于持续进行改进，Beckhoff 明确建议始终保持产品的最新状态，并在产品更新可用后马上进行安装。使用过时的或不支持的产品版本可能会增加网络威胁的风险。

如需了解 Beckhoff 产品信息安全的信息，请订阅 <https://www.beckhoff.com/secinfo> 上的 RSS 源。

2 概述

TwinCAT XML Server 提供了一个 PLC 功能库，可用于实现对 XML 文件的写入和读取访问。该 XML Server 具有易于使用的特点。例如，特别适用于加载初始化数据，在设备启动时经常需要执行此类操作。此外，它还可以将包含 PLC 注释的 PLC 变量以 XML 文件格式保存下来。XML 文件中的变量结构与 PLC 中的变量结构相匹配。这样可以直接访问变量的各个子元素。仅传输那些在 XML 文件中也已定义的子元素（结构体或数组中的元素）。在写入 PLC 变量时，可以选择性地添加缺失的元素。

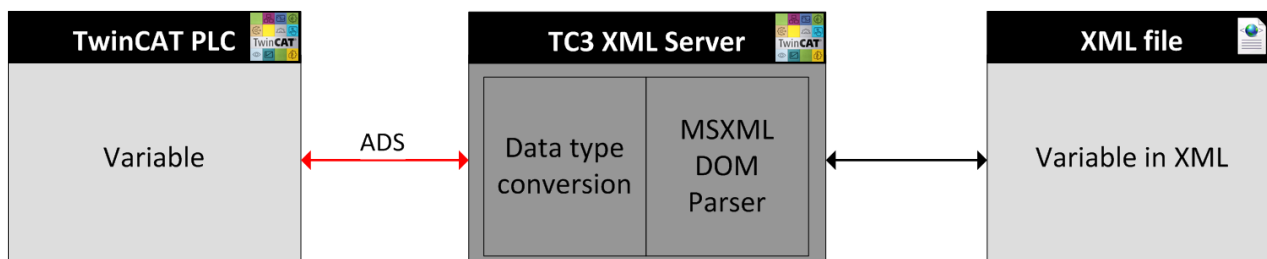
组件

- TwinCAT XML Server：是一项与 TwinCAT 同时启动和停止的服务。它是 TwinCAT 与 XML 文件之间的连接纽带。
- PLC 功能库 [► 16]：PLC 功能库提供了 4 种不同的功能块，可用于将 PLC 功能库中的数据写入 XML 文件或从 XML 文件中读取数据。因此，您可以以 XML 文件格式存储变量，也可以在 TwinCAT 中对 XML 文件中的变量进行初始化。

工作原理

XML Server 通过 ADS 与 TwinCAT PLC 进行通信。在写入过程中，它会将所有待写入的变量转换为文本，并通过 MSXML DOM 解析器写入 XML 文件。XML 文件不包含任何有关数据类型的信息，只包含以标签名称显示的变量名称和值：<variable name> value </variable name>。

在读取过程中，它会通过 ADS 确定待读取变量的数据类型，并对 XML 文件中的文本进行相应的转换。



3 安装

3.1 系统要求

技术数据	TF6421 TwinCAT 3 XML Server
目标系统	Windows 10、Windows 11、Windows CE、TwinCAT/BSD、Beckhoff RT Linux®、PC (x86、x64 和 Arm®)
最低 TwinCAT 版本	3.1 Build 4011
最低 TwinCAT 等级	TC1200 TwinCAT 3 PLC

3.2 安装

安装 (TwinCAT 3.1 Build 4024)

下文将介绍如何在基于 Windows 的操作系统上安装 TwinCAT 3 功能组件。

✓ 从倍福官网下载 TwinCAT 3 功能组件的安装文件。

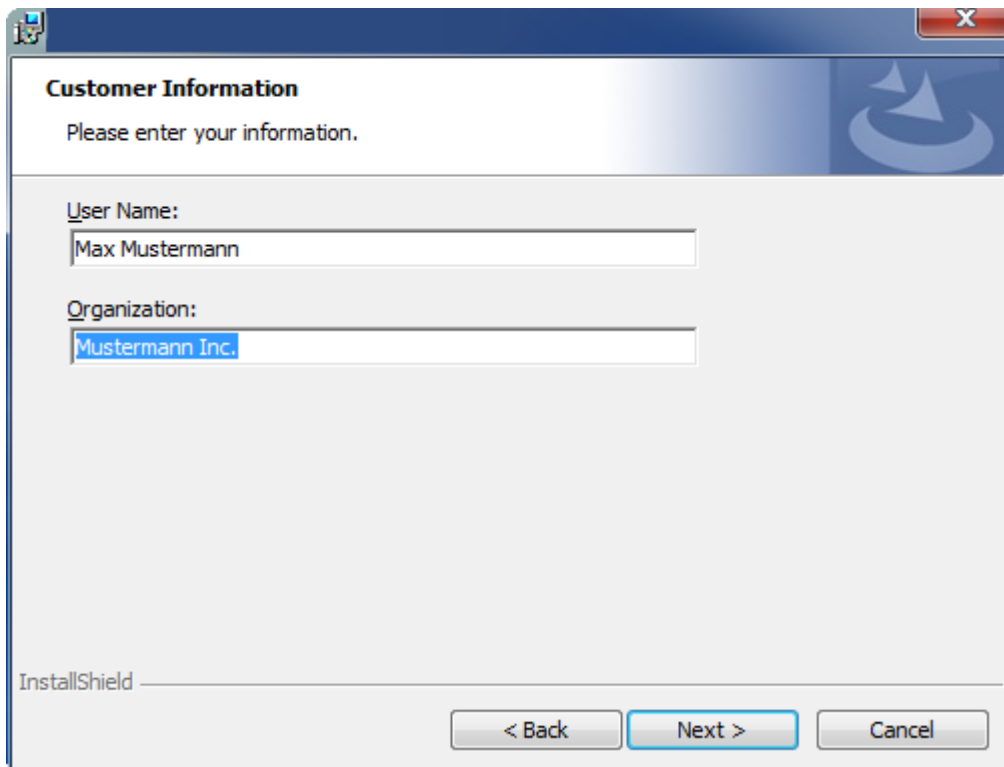
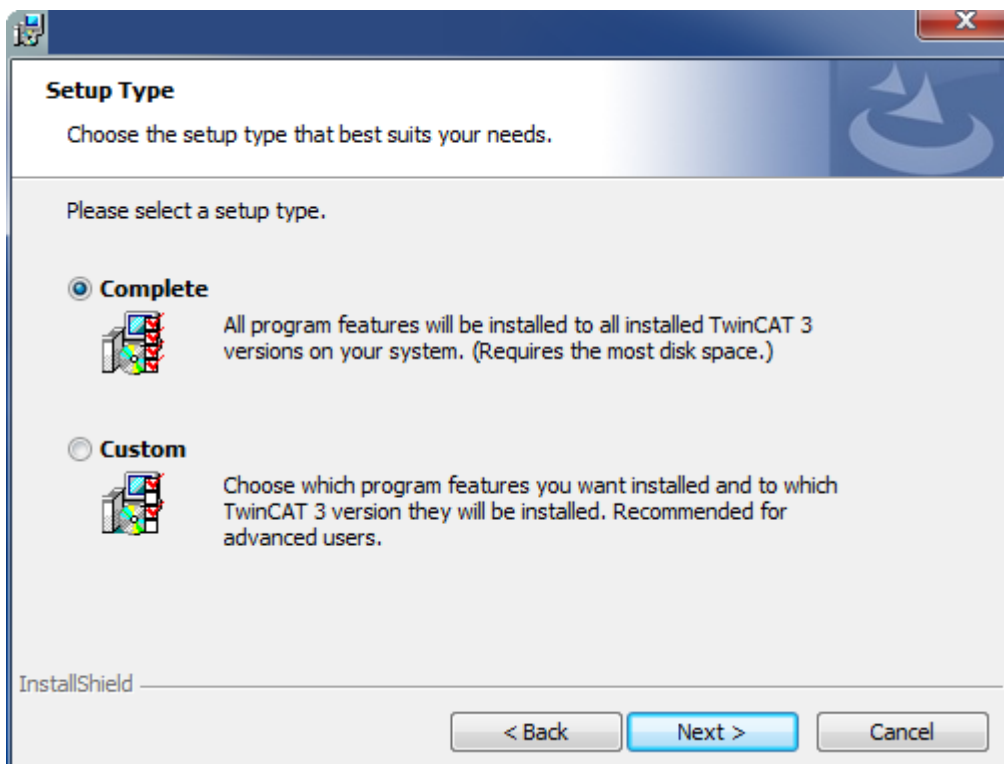
1. 以管理员身份运行安装文件。为此，请在该文件的上下文菜单中选择“**Run As Admin（以管理员身份运行）**”命令。

⇒ 安装对话框打开。

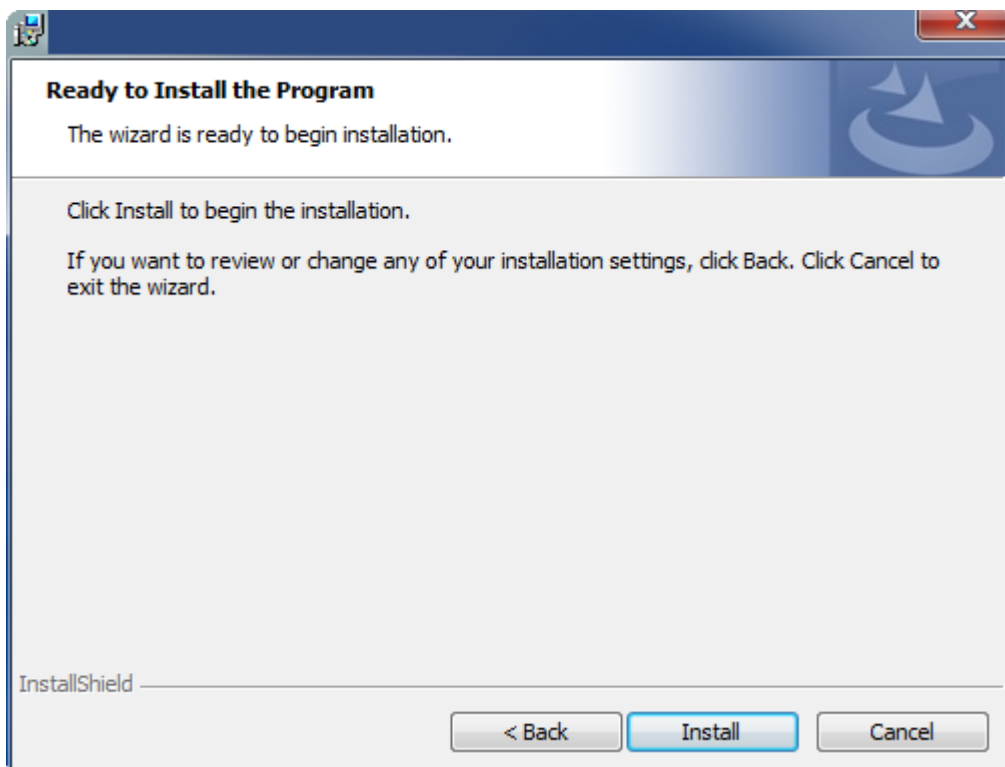
2. 接受最终用户许可协议，然后点击“**Next（下一步）**”。



3. 输入用户数据。

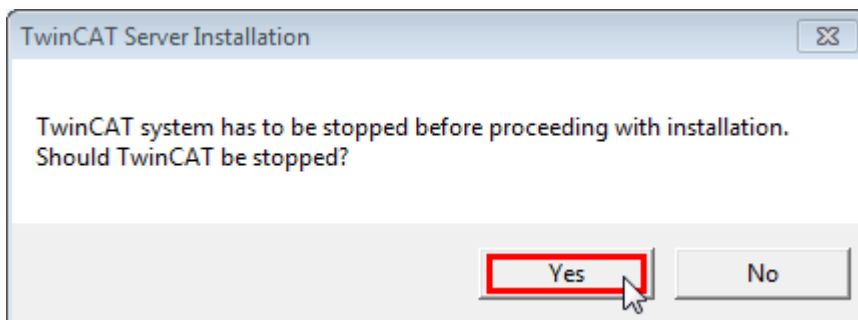
4. 若要安装完整版 TwinCAT 3 功能组件，请选择“**Complete（完整版）**”作为安装类型。若要单独安装 TwinCAT 3 功能组件，请选择“**Custom（自定义）**”。

5. 点击“Next（下一步）”，然后点击“Install（安装）”开始进行安装。

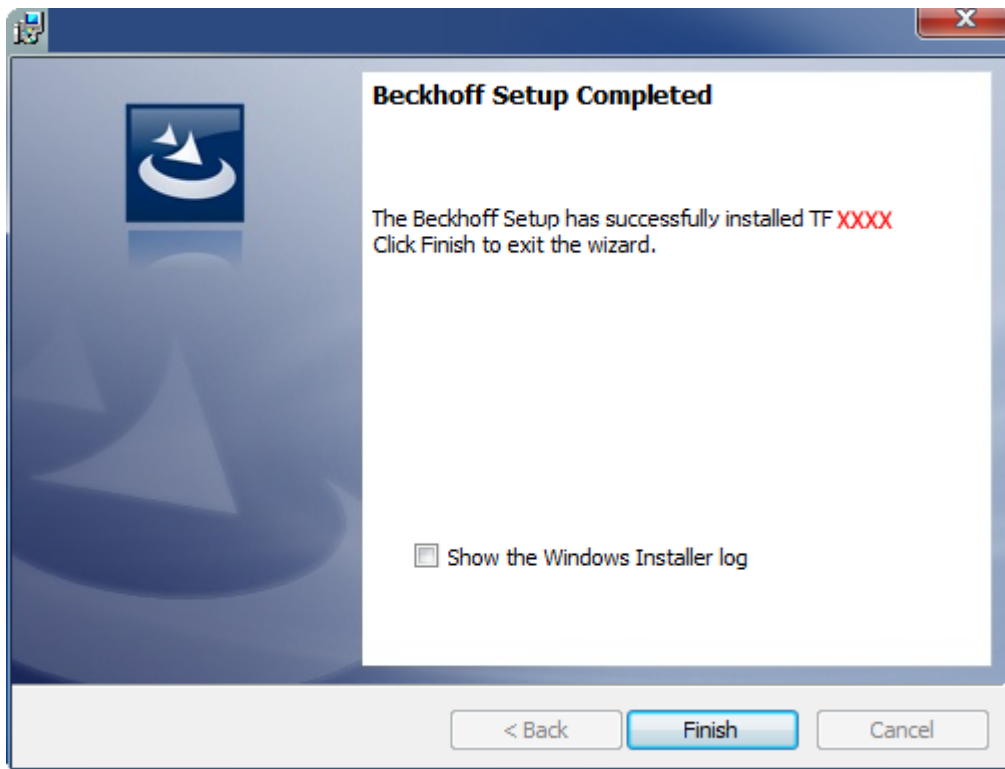


⇒ 将显示一个对话框，提示您必须关闭 TwinCAT 系统才能继续安装。

6. 点击“Yes（是）”进行确认。



7. 点击“**Finish（完成）**”退出安装。



⇒ TwinCAT 3 功能组件安装成功。

3.3 授权

TwinCAT 3 功能可以作为完整版或 7 天测试版激活。这两种许可证类型都可以通过 TwinCAT 3 开发环境（XAE）激活。

授权使用 TwinCAT 3 功能的完整版本

关于许可证完整版本的程序描述，可参见倍福信息系统的文档“[TwinCAT 3 授权](#)”。

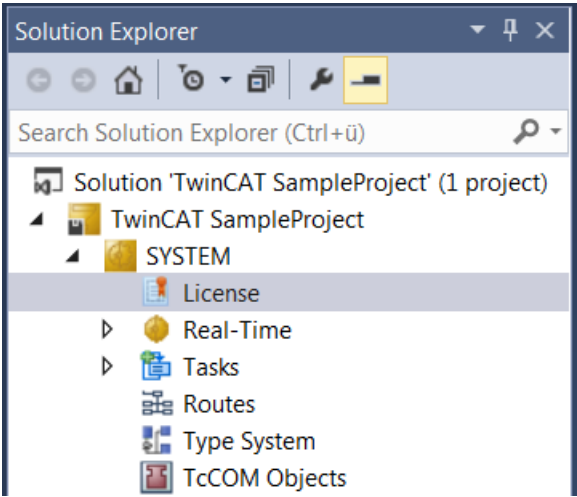
授权一个 TwinCAT 3 功能的 7 天测试版本



对于TwinCAT 3 许可证加密狗无法启用 7 天测试版本。

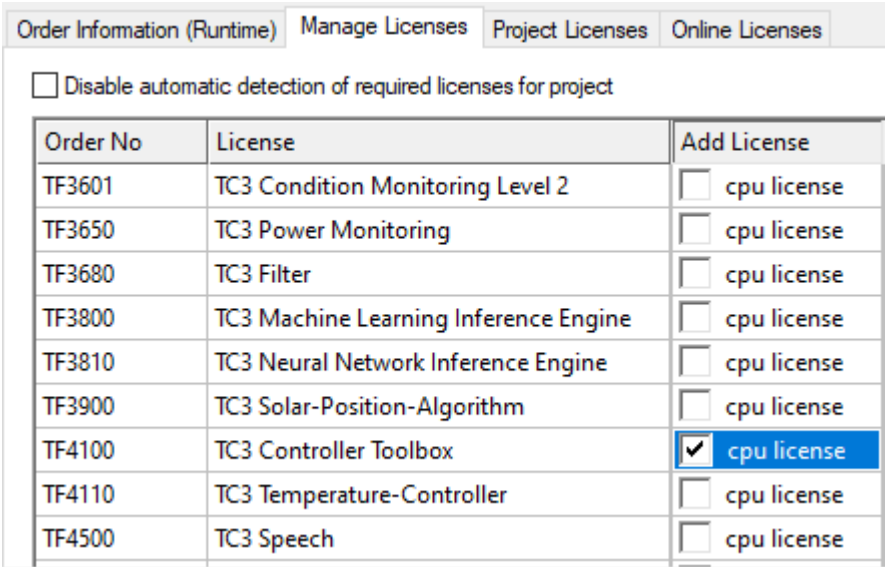
1. 启动 TwinCAT 3 开发环境（XAE）。
2. 打开一个现有的 TwinCAT 3 项目或创建一个新项目。
3. 如果想为一个远程设备激活授权，请设置所需的目标系统。为此，从工具栏的**选择目标系统**下拉列表中选择目标系统。
 - ⇒ 授权设置总是指选定的目标系统。在目标系统上激活项目时，自动将相应的 TwinCAT 3 许可证复制到该系统中。

4. 在**解决方案资源管理器**中，双击**系统**子目录的许可证。



⇒ TwinCAT 3 许可证管理器打开。

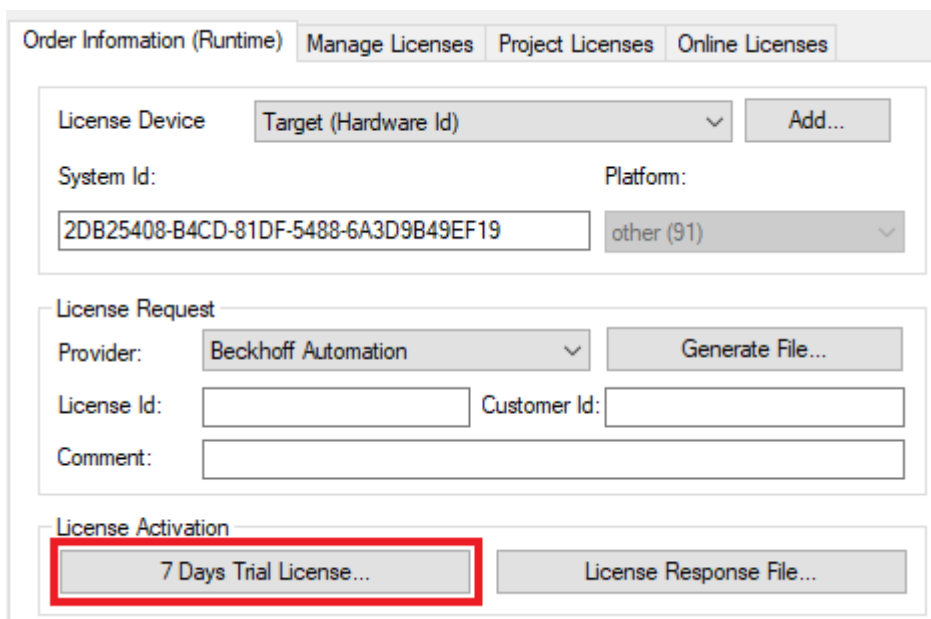
5. 打开**管理许可证**选项卡。在**添加许可证**栏中，选中希望添加到项目的许可证的复选框（例如“TF4100 TC3 控制器工具箱”）。



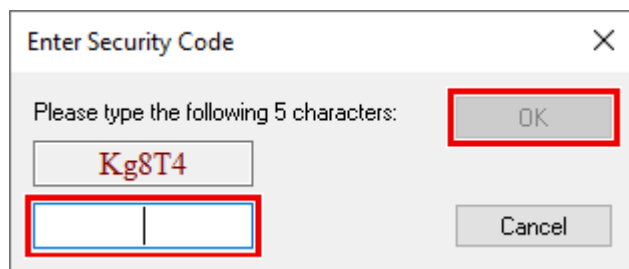
6. 打开**订单信息（运行时间）**选项卡。

⇒ 在许可证的表格概览中，以前选择的许可证显示为“缺失”状态。

7. 点击**7 天试用许可证...**，以激活 7 天试用许可证。



⇒ 一个对话框打开，提示输入对话框中显示的安全代码。



8. 准确输入显示的代码并确认输入。

9. 确认随后的对话框，表明激活成功。

⇒ 在许可证的表格概览中，许可证状态现在显示许可证的到期日。

10. 重新启动 TwinCAT 系统。

⇒ 7 天试用版启用。

3.4 Windows CE 安装

下文将介绍如何在装有 Windows CE 的倍福嵌入式控制器上安装 TwinCAT 3 功能组件 (TFxxx)

1. [下载并安装设置文件 \[► 13\]](#)
2. [将 CAB 文件传输到 Windows CE 设备上 \[► 14\]](#)
3. [在 Windows CE 设备上运行 CAB 文件 \[► 14\]](#)

如果 Windows CE 设备上已经安装了旧版本的 TFxxx，则可以对其进行更新：

- [软件升级 \[► 14\]](#)

下载并安装设置文件

Windows CE 的 CAB 安装文件是 TFxxx 设置的一部分。此文件可在倍福网站 www.beckhoff.com 上获取，并自动包含适用于 Windows XP、Windows 7 和 Windows CE (x86 和 ARM) 的所有版本。

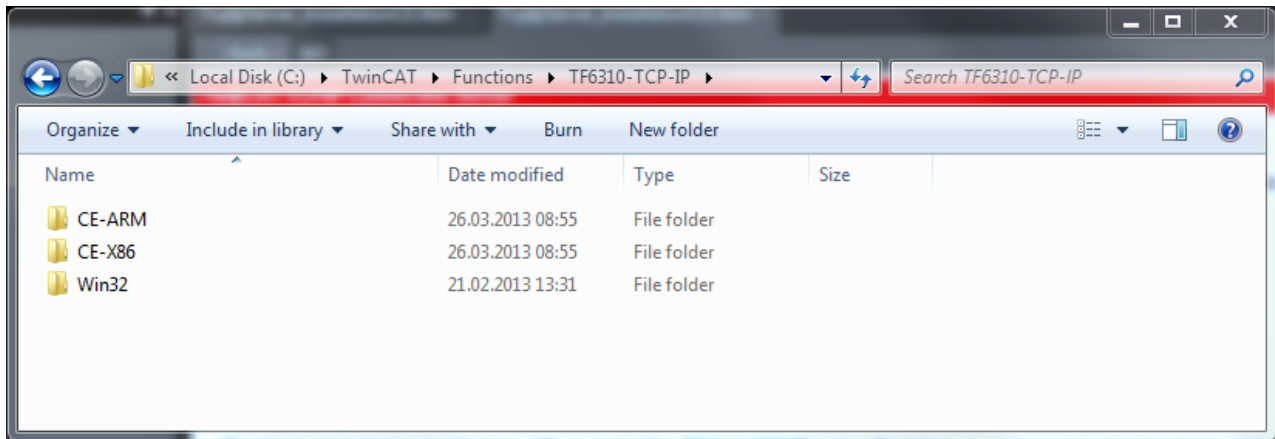
下载 TFxxx 设置文件，并按照[安装 \[► 8\]](#)部分的说明安装 TwinCAT 3 功能组件。

安装完成后，安装文件夹包含三个目录（每个硬件平台一个目录）：

- **CE-ARM:** 基于 ARM 的嵌入式控制器，运行 Windows CE，例如 CX8090、CX9020
- **CE-X86:** 基于 X86 的嵌入式控制器，运行 Windows CE，例如 CX50xx、CX20x0
- **Win32:** 嵌入式控制器，运行 Windows XP、Windows 7 或 Windows Embedded Standard

CE-ARM 和 CE-X86 目录包含与 Windows CE 设备各自硬件平台相关的 Windows CE TwinCAT 3 功能组件的 CAB 文件。

例如：安装文件夹“TF6310”



将 CAB 文件传输到 Windows CE 设备上

将相应的 CAB 文件传输到 Windows CE 设备上。

传输可执行文件有多种选择：

- 通过网络共享
- 通过集成的 FTP 服务器
- 通过 ActiveSync
- 通过 CF/SD 卡

更多信息请参见倍福信息系统的“操作系统”文档（嵌入式控制器 > 操作系统 > CE）。

在 Windows CE 设备上运行 CAB 文件

将 CAB 文件传输到 Windows CE 设备后，双击该文件。选择 **OK**（确定），确认安装对话框。然后重启 Windows CE 设备。

重启设备后，TwinCAT 3 功能组件 (TFxxxx) 的文件将在后台自动加载，然后可以使用。

软件安装在 Windows CE 设备的以下目录中：

\Hard Disk\TwinCAT\Functions\TFxxxx

软件升级

如果 Windows CE 设备上已经安装了旧版本的 TwinCAT 3 功能组件，请在 Windows CE 设备上执行以下步骤升级至新版本：

1. 单击**开始 > 运行**，输入"Explorer"，打开 CE 资源管理器。
2. 导航至 \Hard Disk\TwinCAT\Functions\TFxxx\xxxx。
3. 将文件 Tc*.exe 重命名为 Tc*.old。
4. 重启 Windows CE 设备。
5. 将新的 CAB 文件传输至 Windows CE 设备。
6. 在 Windows CE 设备上运行 CAB 文件并安装新版本。
7. 删除 Tc*.old 文件。

8. 重启 Windows CE 设备。

⇒ 重启后，新版本生效。

4 PLC 功能库

4.1 概述

PLC 功能库 **Tc2_TcXmlDataSrv** 随 TC3 XML Server 一起提供，在安装过程中会将其复制到文件夹 ...C:\TwinCAT\3.1\Components\Plc\Managed Libraries\Beckhoff Automation GmbH 中。

该功能库有 2 个用于从 XML 文件读取变量的功能块：

- FB_XmlSrvRead
- FB_XmlSrvReadByName

以及 2 个用于将 PLC 变量写入 XML 文件的功能块：

- FB_XmlSrvWrite
- FB_XmlSrvWriteByName

第一个版本（FB_XMLSrvRead、FB_XMLSrvWrite）使用 PLC 变量的地址和大小来指定变量。第二个版本（FB_XMLSrvReadByName、FB_XMLSrvWriteByName）使用符号名称来指定变量。第一个版本具有更高的性能。此外，还必须将 XML 文件的路径和 XML 文档中变量的位置作为输入参数传输至功能块。

原始数据类型

支持以下原始数据类型：

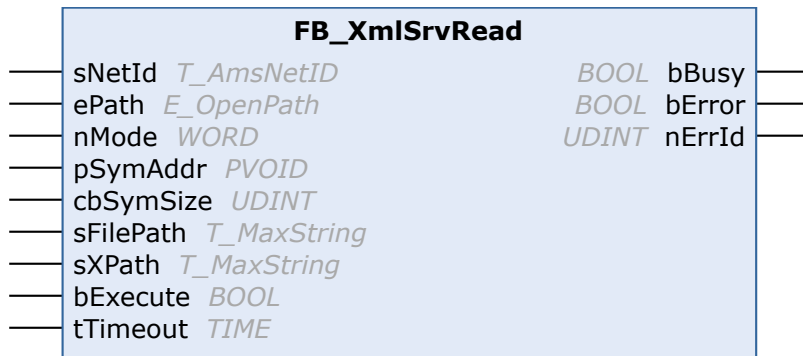
数据类型	PLC 示例	XML 示例
UDINT、DINT、UINT、INT、USINT、SINT、DWORD、WORD、BYTE	value1 : DINT := -1; value2 : UDINT := 65535;	<dataentry> <MAIN.value1>-1</MAIN.value1> <MAIN.value2>65535</MAIN.value2> </dataentry>
LREAL、REAL	value1 : LREAL = 1.2;	<dataentry> <MAIN.value1>1.2</MAIN.value1> </dataentry>
STRING	str1 : STRING = 'hallo';	<dataentry> <MAIN.str1>hallo</MAIN.str1> </dataentry>
TIME、DATE、TOD、DT	date1:DATE :=D#2005-05-04; (* 时间类型以 DWORD 格式存储在 XML 文件中*)	<dataentry> <MAIN.date1>1115164800</MAIN.date1> </dataentry>
BOOL	bool1:BOOL := TRUE; bool2:BOOL := FALSE;	<dataentry> <MAIN.bool1>>true</MAIN.bool1> <MAIN.bool2>>false</MAIN.bool2> </dataentry>

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2 功能块

4.2.1 FB_XmlSrvRead



FB_XmlSrvRead 功能块可用于使用 XML 文件中的数据初始化 PLC 变量。输入变量 sXPath 必须指向通过 sFilePath 指定的 XML 文件中的一个有效节点。待初始化的符号可通过其符号地址和大小唯一确定。

输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  ePath       : E_OpenPath := PATH_GENERIC;
  nMode       : WORD := XMLSRV_SKIPMISSING;
  pSymAddr    : DWORD;
  cbSymSize   : UDINT;
  sFilePath   : T_MaxString;
  sXPath      : T_MaxString;
  bExecute    : BOOL;
  tTimeout    : TIME := T#60s;
END_VAR
```

名称	类型	描述
sNetId	<u>T_AmsNetId</u>	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
ePath	<u>E_OpenPath</u>	此类变量可用于在目标设备上选择通用路径或 TwinCAT 系统路径，以打开文件。默认值为 PATH_GENERIC。
nMode	WORD	该输入可用于控制 XML 文件的解析方式。对于 XmlSrvRead 命令，最初仅支持 XMLSRV_SKIPMISSING 模式。
pSymAddr	DWORD	待使用 XML 文件中的数据写入的 PLC 变量地址。
cbSymSize	UDINT	待使用 XML 文件中的数据写入的 PLC 变量的大小。
sFilePath	<u>T_MaxString</u>	包含要打开的文件路径和文件名。 路径只能指向计算机的本地文件系统。这意味着在这里不能使用网络路径。
sXPath	<u>T_MaxString</u>	包含 XML 文档中待写入数据的标签的地址。地址必须是有效的 XPath 指令。标签名称不得与符号名称相同。
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

输出

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
END_VAR
```

名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。

名称	类型	描述
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 <u>TwinCAT 3 XML Server</u> 的错误编号 [► 34]。

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2.2 FB_XmlSrvWrite



FB_XmlSrvWrite 功能块可用于将 PLC 变量的值写入 XML 文件。输入变量 sXPath 必须指向通过 sFilePath 指定的 XML 文件中的一个有效节点。待写入的符号可通过其地址和大小唯一确定。

🔧 输入

```

VAR_INPUT
  sNetId      : T_AmsNetId;
  ePath       : E_OpenPath := PATH_GENERIC;
  nMode       : WORD := XMLSRV_SKIPMISSING;
  pSymAddr    : DWORD;
  cbSymSize   : UDINT;
  sFilePath   : T_MaxString;
  sXPath      : T_MaxString;
  bExecute    : BOOL;
  tTimeout    : TIME := T#60s;
END_VAR

```

名称	类型	描述
sNetId	<u>T_AmsNetId</u>	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
ePath	<u>E_OpenPath</u>	此类变量可用于在目标设备上选择通用路径或 TwinCAT 系统路径，以打开文件。默认值为 PATH_GENERIC。
nMode	WORD	该输入可用于控制将哪些数据写入 XML 文件。XMLSRV_SKIPMISSING 和 XMLSRV_ADDMISSING 模式适用于 XmlSrvWrite 命令。在 XMLSRV_SKIPMISSING 模式下，只会将 XML 文件中已存在的 PLC 符号的子元素写入 XML 文件。在 XMLSRV_ADDMISSING 模式下，会将缺失的子元素添加到 XML 文件中。从第 3.2.31.0 版产品开始，可以通过参数 XMLSRV_SERIALIZESYMCOMMENT 和 XMLSRV_SERIALIZETYPECOMMENT 将声明部分中的注释写入 XML 文件。
pSymAddr	DWORD	待写入 XML 文件的 PLC 变量地址。
cbSymSize	UDINT	待写入 XML 文件的 PLC 变量的大小。
sFilePath	<u>T_MaxString</u>	包含要打开的文件路径和文件名。 路径只能指向计算机的本地文件系统。这意味着在这里不能使用网络路径。

名称	类型	描述
sXPath	T_MaxString	包含 XML 文档中待写入数据的标签的地址。地址必须是有效的 XPath 指令。标签名称不得与符号名称相同。
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

输出

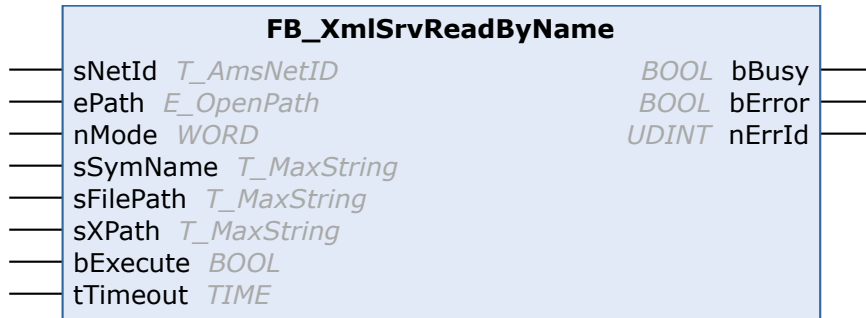
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
END_VAR
```

名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 TwinCAT 3 XML Server 的错误编号 [► 34]。

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2.3 FB_XmlSrvReadByName



FB_XmlSrvReadByName 功能块可用于使用 XML 文件中的数据初始化 PLC 变量。输入变量 sXPath 必须指向通过 sFilePath 指定的 XML 文件中的一个有效节点。待初始化的符号可通过其符号名唯一标识。

输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  ePath       : E_OpenPath := PATH_GENERIC;
  nMode       : WORD := XMLSRV_SKIPMISSING;
  sSymName    : T_MaxString;
  sFilePath   : T_MaxString;
  sXPath      : T_MaxString;
  bExecute    : BOOL;
  tTimeout    : TIME := T#60s;
END_VAR
```

名称	类型	描述
sNetId	T_AmsNetId	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
ePath	E_OpenPath	此类变量可用于在目标设备上选择通用路径或 TwinCAT 系统路径，以打开文件。默认值为 PATH_GENERIC。

名称	类型	描述
nMode	WORD	该输入可用于控制 XML 文件的解析方式。XmlSrvReadByName 命令目前仅支持 XMLSRV_SKIPMISSING 模式。
sSymName	T_MaxString	待使用 XML 文件中的数据写入的 PLC 符号的名称。
sFilePath	T_MaxString	包含要打开的文件路径和文件名。 路径只能指向计算机的本地文件系统。这意味着在这里不能使用网络路径。
sXPath	T_MaxString	包含 XML 文档中待写入数据的标签的地址。地址必须是有效的 XPath 指令。标签名称不得与符号名称相同。
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

输出

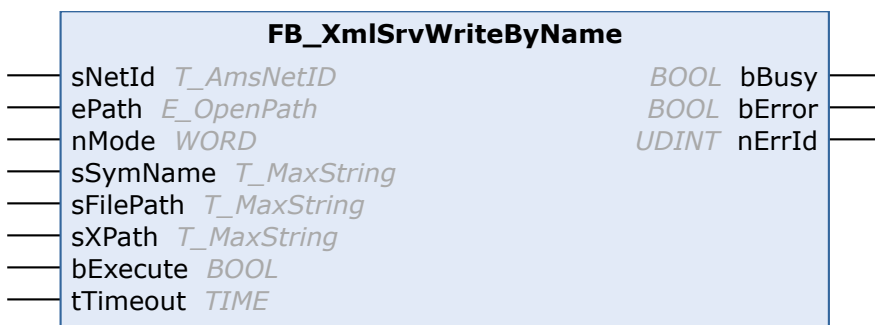
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
END_VAR
```

名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 TwinCAT 3 XML Server 的错误编号 [► 34]。

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2.4 FB_XmlSrvWriteByName



FB_XmlSrvWriteByName 功能块可用于将 PLC 变量的值写入 XML 文件。输入变量 sXPath 必须指向通过 sFilePath 指定的 XML 文件中的一个有效节点。待写入的符号可通过其符号名唯一标识。

输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  ePath       : E_OpenPath := PATH_GENERIC;
  nMode       : WORD := XMLSRV_SKIPMISSING;
  sSymName    : T_MaxString;
  sFilePath   : T_MaxString;
  sXPath      : T_MaxString;
  bExecute    : BOOL;
  tTimeout    : TIME := T#60s;
END_VAR
```

名称	类型	描述
sNetId	T_AmsNetId	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
ePath	E_OpenPath	此类变量可用于在目标设备上选择通用路径或 TwinCAT 系统路径，以打开文件。默认值为 PATH_GENERIC。
nMode	WORD	该输入可用于控制将哪些数据写入 XML 文件。XMLSRV_SKIPMISSING 和 XMLSRV_ADDMISSING 模式适用于 XmlSrvWriteByte 命令。在 XMLSRV_SKIPMISSING 模式下，只会将 XML 文件中已存在的 PLC 符号的子元素写入 XML 文件。在 XMLSRV_ADDMISSING 模式下，会将缺失的子元素添加到 XML 文件中。从第 3.2.31.0 版产品开始，可以通过参数 XMLSRV_SERIALIZESYMCOMMENT 和 XMLSRV_SERIALIZETYPECOMMENT 将声明部分中的注释写入 XML 文件。
sSymName	T_MaxString	待写入 XML 文件的 PLC 符号的名称。
sFilePath	T_MaxString	包含要打开的文件路径和文件名。 路径只能指向计算机的本地文件系统。这意味着在这里不能使用网络路径。
sXPath	T_MaxString	包含 XML 文档中待写入数据的标签的地址。地址必须是有效的 XPath 指令。标签名称不得与符号名称相同。
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
END_VAR
```

名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 TwinCAT 3 XML Server 的错误编号 [► 34]。

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2.5 FB_XmlSrvEnableObfuscation

FB_XmlSrvEnableObfuscation		
sNetId	T_AmsNetId	BOOL bBusy
sUserName	T_MaxString	BOOL bError
sPassword	T_MaxString	UDINT nErrId
bExecute	BOOL	
tTimeout	TIME := TIME#1m0s0ms	

FB_XmlSrvEnableObfuscation 功能块可启用混淆模式。启用后，XML 文件将仅以 AES-256 加密方式存储在硬盘上。

 输入

```

VAR_INPUT
    sNetId      : T_AmsNetId;
    sUserName   : T_MaxString;
    sPassword   : T_MaxString;
    bExecute    : BOOL;
    tTimeout    : TIME := T#1m0s0ms;
END_VAR

```

名称	类型	描述
sNetId	T_AmsNetId	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
sUserName	T_MaxString	用户名
sPassword	T_MaxString	密码
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

 输出

```

VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR

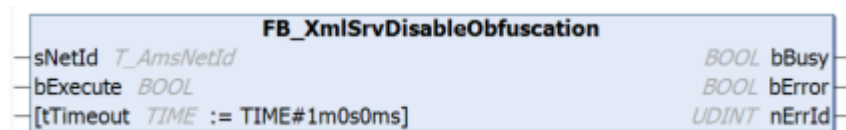
```

名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 <u>TwinCAT 3 XML Server 的错误编号</u> [► 34]。

要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.1 Build 4024.55	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2.6 FB_XmlSrvDisableObfuscation



FB_XmlSrvDisableObfuscation 功能块可禁用混淆模式。

 输入

```

VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := T#1m0s0ms;
END_VAR

```

名称	类型	描述
sNetId	T_AmsNetId	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

输出

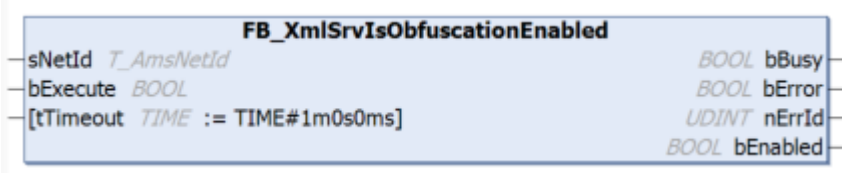
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
END_VAR
```

名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 <u>TwinCAT 3 XML Server</u> 的错误编号 [► 34]。

要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.1 Build 4024.55	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2.7 FB_XmlSrvIsObfuscationEnabled



FB_XmlSrvIsObfuscationEnabled 功能块可查询混淆模式是否已启用或禁用。

输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  bExecute    : BOOL;
  tTimeout    : TIME := T#1m0s0ms;
END_VAR
```

名称	类型	描述
sNetId	<u>T_AmsNetId</u>	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

输出

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
  bEnabled   : BOOL;
END_VAR
```

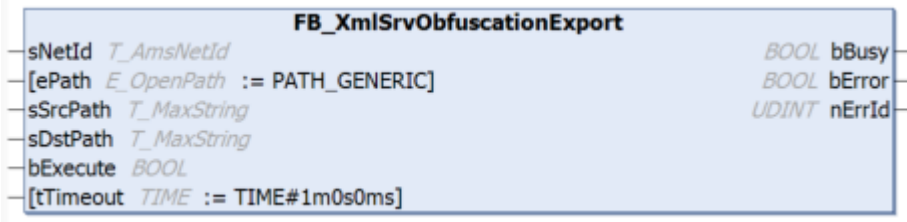
名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 <u>TwinCAT 3 XML Server</u> 的错误编号 [► 34]。

名称	类型	描述
bEnabled	BOOL	该输出的值若为 TRUE，表示混淆模式已启用。FALSE 表示已禁用。

要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.1 Build 4024.55	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.2.8 FB_XmlSrvObfuscationExport



FB_XmlSrvObfuscationExport 功能块可解密经过混淆处理的 XML 文件，并将其以未加密的形式存储在硬盘上。

输入

```

VAR_INPUT
    sNetId      : T_AmsNetId;
    ePath       : E_OpenPath := PATH_GENERIC;
    sSrcPath    : T_MaxString;
    sDstPath    : T_MaxString;
    bExecute    : BOOL;
    tTimeout    : TIME := T#1m0s0ms;
END_VAR

```

名称	类型	描述
sNetId	<u>T_AmsNetId</u>	包含 TwinCAT 3 XML Server 网络地址的字符串。对于本地计算机（默认），可以指定一个空字符串。
ePath	<u>E_OpenPath</u>	此类变量可用于在目标设备上选择通用路径或 TwinCAT 系统路径，以打开文件。默认值为 PATH_GENERIC。
sSrcPath	<u>T_MaxString</u>	加密文件的路径。
sDstPath	<u>T_MaxString</u>	未加密文件的目标路径。
bExecute	BOOL	功能块由该输入端的上升沿触发激活。
tTimeout	TIME	允许执行功能块的最长时间。

输出

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
END_VAR

```

名称	类型	描述
bBusy	BOOL	在激活功能块时设置该输出，并保持该状态，直至收到反馈。
bError	BOOL	如果在命令传输过程中发生错误，则在 bBusy 输出重置后设置该输出。
nErrId	UDINT	如果已设置 bError 输出，该参数将返回 <u>TwinCAT 3 XML Server</u> 的错误编号 [► 34]。

要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.1 Build 4024.55	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.3 全局常量

4.3.1 全局变量

VAR_GLOBAL 常量

```
VAR_GLOBAL CONSTANT
XMLSRV_AMSPORT :UINT :=10600;

XMLSRV_IGR_CLOSE :UDINT := 121;
XMLSRV_IGR_READ :UDINT := 122;
XMLSRV_IGR_WRITE :UDINT := 123;
XMLSRV_IGR_OPENREAD :UDINT := 124;
XMLSRV_IGR_OPENWRITE :UDINT := 125;

XMLSRV_SKIPMISSING :WORD := 0;
XMLSRV_ADDMISSING :WORD := 1; (*for write commands*)

XMLSRV_MAX_FRAGSIZE :UDINT := 16#40000;

XMLSRVERROR_INTERNAL :UDINT:= 16#8000;
XMLSRVERROR_NOTFOUND :UDINT:= 16#8001;
XMLSRVERROR_PARSERERROR :UDINT:= 16#8002;
XMLSRVERROR_INCOMPATIBLE :UDINT:= 16#8003;
XMLSRVERROR_NOMEMORY :UDINT:= 16#8004;
XMLSRVERROR_ADDNODE :UDINT:= 16#8005;
XMLSRVERROR_INVALIDDXPATH :UDINT:= 16#8006;

END_VAR
```

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

4.3.2 功能库版本

所有功能库都有特定的版本。该版本信息会显示在 PLC 功能库的资源库中。
功能库版本号将存储在全局常量中。

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_TcXmlDataSrv : ST_LibVersion;
END_VAR
```

名称	类型	描述
stLibVersion_Tc2_TcXmlDataSrv	ST_LibVersion	Tc2_ModbusSrv 功能库的版本号。

函数 F_CmpLibVersion (在 Tc2_System 功能库中) 可用于对现有版本和所需版本进行比较。



与 TwinCAT 2 兼容
不再提供 TwinCAT 2 功能库的查询选项。

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

5 示例

以下页面包含一些示例（以下称为“示例”），旨在向您介绍 TC3 XML Server 的工作原理。这些示例按顺序进行编号，可在此处以 PLC 项目的形式下载这些示例：https://infosys.beckhoff.com/content/1033/TF6421_Tc3_XML_Server/Resources/1635013643.zip

- [入门 \[▶ 26\]](#)
本节将介绍 TwinCAT XML Data Server 的一般工作原理、结构体和数组的功能以及各个节点的访问权限。
- [功能块 \[▶ 27\]](#)
我们通过 4 个示例对 PLC 功能库 TcXmlDataSrv.Lib 的功能块的应用和配置进行了说明。（示例 1–4）
- [更多示例 \[▶ 29\]](#)
本文档包含更多示例，例如描述了 PLC 启动时的一次性初始化和循环写入等。（示例 5–6）
- [生产示例 \[▶ 31\]](#)
生产示例演示了小型生产订单的处理流程。它使用的是 FB_XmlSrvReadByName 和 FB_XmlSrvWriteByName。（示例 7）

5.1 入门

下文将通过几个简单的示例介绍 TC3 XML Server 的基本工作原理。

我们在 PLC 中定义了 DINT 类型的全局变量 .Var1。应将其保存在一个 XML 文件中，如下所示：

```
<dataentry>
  <Var1>10</Var1>
</dataentry>
```

为此，必须对功能块 FB_XmlSrvWrite 的输入变量进行如下设置：

```
fbRead.pSymAddr    := ADR(value1);
fbRead.cbSymSize    := SIZEOF(value1);
fbRead.sFilePath    := 'C:\Test.xml'; (*Pfad zur XML-Datei*)
fbRead.sXPath       := '/dataentry/Var1';
```

可自由选择 XML 文件中变量的根元素名称。只需调整输入变量 sXPath 中的路径即可。一个 XML 文件可能包含多个变量的定义：

```
<dataentry>
  <Var1>10</Var1>
  <Var2>100</Var2>
  <Var3>
    <a>100</a>
    <b>10</b>
  </Var3>
</dataentry>
```

例如，若要访问符号 .Var3.a，必须将 sXPath 设置为 “/dataentry/Var3.a”。

结构体

XML 文件的层次结构与 PLC 相同。可跳过 XML 文件的个别子元素：结构体的子元素必须与 PLC 中的名称相同，否则将作跳过处理。如果 XML 文件的子元素无法转换为正确的数据类型，也会作跳过处理。

示例：

我们在 PLC 中定义了 ST_MYSTRUCT 类型的全局变量 .Var2：

```
TYPE ST_MYSTRUCT:
STRUCT
  a: UINT;
  b: DINT;
  c: LREAL;
  d: STRING;
END_STRUCT
END_TYPE
```

XML 文件内容如下所示：

```
<variables>
  <Var1>10</Var1>
  <Var2>      <!-- sXPath := '/variables/Var2' -->
```

```

    <a>100</a>
    <b>-10</b>
    <c>1.2</c>
    <d>Hallo</d>
  </Var2>
</variables>

```

本例中所有子元素都已被完整且正确地定义，因此变量已完全初始化。而在下面的示例中，仅对子元素 c 进行了序列化：

```

<variables>
  <Var1>10</Var1>
  <Variable2> <!-- sXPath := '/variables/Variable2' -->
    <Info>dies ist ein Test</Info>
    <a>-100</a>
    <c>1.2</c>
  </Variable2>
</variables>

```

子元素 a 无法转换，因为它是负数，要求使用 UINT 类型。子元素 b 完全缺失。将会跳过标签 <Info>，因为未在 PLC 文件中定义该标签。

数组

若要指定数组索引，各个数组元素必须使用属性“index”。可以省略数组中的单个元素，若省略，这些元素将被跳过。

示例：

我们在 PLC 中定义了 ARRAY[1..4] OF DINT 类型的变量 .array1。XML 文件内容如下所示：

```

<dataentry>
  <array1 index="1">10</array1>
  <array1 index="2">10</array1>
  <array1 index="3">10</array1>
  <array1 index="4">10</array1>
</dataentry>

```

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

5.2 功能块

以下几个示例展示了如何处理 Tc2_XmlDataSrv 功能库的功能块。可在此处下载包含这几个示例的 PLC 项目：https://infosys.beckhoff.com/content/1033/TF6421_Tc3_XML_Server/Resources/1635013643.zip

所有示例均使用 ST_MYSTRUCT 结构体，而 ST_MYSTRUCT 结构体又包括 ST_INNTERSTRUCT。这两个结构体如下所示：

结构体

```

TYPE ST_MYSTRUCT:
STRUCT
  fReal      : REAL;
  bBool      : ARRAY [0..2] OF BOOL;
  stInner    : ST_INNTERSTRUCT;
END_STRUCT
END_TYPE

TYPE ST_INNTERSTRUCT:
STRUCT
  nInteger   : INT;
  sString    : STRING;
END_STRUCT
END_TYPE

```

示例 1：使用 FB_XmlSrvWrite 进行写入操作

第一步是将 ST_MyStruct 结构写入 XML 文件。将模式设置为 XMLSRV_ADMISSING，以便自动生成 XML 文件并在其中创建该结构。系统不会自动创建文件夹！对于较大的结构，如果文件仅供后续读取，也建议采用此方法。这样便无需手动创建 XML 文件，从而避免了错误。

```
(* Sample1 creates an XML-file under the path C:\Test.xml and writes value1 to it.
FUNCTIONBLOCK: FB_XmlSrvWrite *)

PROGRAM Sample1
VAR
    value1      : ST_MyStruct;
    fbXmlSrvWrite : FB_XmlSrvWrite;
    bExecute     : BOOL;
    sFilePath    : T_MaxString := 'C:\Test.xml'; (* CE: '\Hard Disk\Test.xml' *)
    sXPath       : T_MaxString := '/dataentry/MAIN.value1';
END_VAR

fbXmlSrvWrite(
    nMode      := XLSRV_ADDMISSING,
    pSymAddr   := ADR(value1),
    cbSymSize  := SIZEOF(value1),
    sFilePath  := sFilePath,
    sXPath     := sXPath,
    bExecute   := bExecute
);
bExecute := TRUE;
```

示例 2 : 使用 FB_XmlSrvWriteByName 进行写入操作

示例 2 的结果与示例 1 相同，但使用的是 FB_XmlSrvWriteByName 功能块。示例 1 具有更高的性能。

```
(* Sample2 creates an XML-file under the path C:\Test.xml and writes value1 to it.
FUNCTIONBLOCK: FB_XmlSrvWriteByName *)

PROGRAM Sample2
VAR
    value1      : ST_MyStruct;
    fbXmlSrvWrite : FB_XmlSrvWriteByName;
    bExecute     : BOOL;
    sSymName     : T_MaxString := 'Sample2.value1';
    sFilePath    : T_MaxString := 'C:\Test.xml'; (* CE: '\Hard Disk\Test.xml' *)
    sXPath       : T_MaxString := '/dataentry/MAIN.value1';
END_VAR

fbXmlSrvWrite(
    nMode      := XLSRV_ADDMISSING,
    sSymName   := sSymName,
    sFilePath  := sFilePath,
    sXPath     := sXPath,
    bExecute   := bExecute
);
bExecute := TRUE;
```

XML 文件

在这两个示例中，XML 文件“Test.XML”均创建在 C:\ 下。为了确保两个示例中的 XML 文件具有相同的内容，我们使用了相同的 sXPath，即“/dataentry/MAIN.value”，但 value1 并未存储在 Main 中，而是直接存储在相应的程序中。sSymName（示例 2）指定了变量在 TwinCAT 中的位置：“Sample2.value1”！

```
<dataentry>
  <MAIN.value1>
    <fReal>0</fReal>
    <bBool index="0">false</bBool>
    <bBool index="1">false</bBool>
    <bBool index="2">false</bBool>
    <stInner>
      <nInteger>0</nInteger>
      <sString></sString>
    </stInner>
  </MAIN.value1>
</dataentry>
```

示例 3 : 使用 FB_XmlSrvRead 进行读取操作

下文描述了读取在示例 1 和/或示例 2 中创建的 XML 文件结构的过程。示例 3 使用 FB_XmlSrvRead 功能块来实现此目的。

```
(* Sample3 reads an XML-file (C:\Test.xml)
FUNCTIONBLOCK: FB_XmlSrvRead *)

PROGRAM Sample3
VAR
    value1      : ST_MyStruct;
    fbXmlSrvRead : FB_XmlSrvRead;
    bExecute     : BOOL;
    sFilePath    : T_MaxString := 'C:\Test.xml'; (* CE: '\Hard Disk\Test.xml' *)
    sXPath       : T_MaxString := '/dataentry/MAIN.value1';
END_VAR

fbXmlSrvRead(
    pSymAddr := ADR(value1),
```

```

cbSymSize := SIZEOF(value1),
sFilePath := sFilePath,
sXPath    := sXPath,
bExecute  := bExecute
);
bExecute:= TRUE;

```

示例 4：使用 FB_XmlSrvReadByName 进行读取操作

示例 4 显示了使用功能块 FB_XmlSrvReadByName 进行读取操作的过程。

```

(* Sample4 reads an XML-file (C:\Test.xml)
FUNCTIONBLOCK: FB_XmlSrvReadByName *)

PROGRAM Sample4
VAR
    value1      : ST_MyStruct;
    fbXmlSrvRead : FB_XmlSrvReadByName;
    bExecute     : BOOL;
    sSymName     : T_MaxString := 'Sample4.value1';
    sFilePath    : T_MaxString := 'C:\Test.xml'; (* CE: '\Hard Disk\Test.xml' *)
    sXPath       : T_MaxString := '/dataentry/MAIN.value1';
END_VAR

fbXmlSrvRead(
    sSymName := sSymName,
    sFilePath := sFilePath,
    sXPath    := sXPath,
    bExecute  := bExecute
);
bExecute:= TRUE;

```

与示例 2 一样，请注意 sSymName 与 sXPath 的不同之处：sXPath 表示 XML 文件中的路径。示例 1 和/或示例 2 已指定该路径。sSymName 表示 TwinCAT 变量的符号名称，因此为“Sample4.value1”。

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

5.3 更多示例

以下示例展示了 TC3 XML Server 不同类型的应用。可在此处下载包含这几个示例的 PLC 项目：https://infosys.beckhoff.com/content/1033/TF6421_Tc3_XML_Server/Resources/1635013643.zip

示例 5 展示了程序启动时的一次性初始化，示例 6 展示了循环和事件驱动型打印程序。这两个示例使用的同样是 ST_MYSTRUCT 结构，而 ST_MYSTRUCT 结构又包括 ST_ININTERSTRUCT。这两个结构体如下所示：

结构体

```

TYPE ST_MYSTRUCT:
STRUCT
    fReal      : REAL;
    bBool      : ARRAY [0..2] OF BOOL;
    stInner    : ST_ININTERSTRUCT;
END_STRUCT
END_TYPE

TYPE ST_ININTERSTRUCT:
STRUCT
    nInteger : INT;
    sString  : STRING;
END_STRUCT
END_TYPE

```

示例 5：程序启动时的一次性初始化

示例 5 展示了程序启动时 value1 的一次性初始化。使用了功能块 FB_XmlSrvRead。

```

(* Sample5 reads and initializes value1 when the PLC is started FUNCTIONBLOCK: FB_XmlSrvRead *)

PROGRAM Sample5
VAR
    value1      : ST_MyStruct;
    fbXmlSrvRead : FB_XmlSrvRead;
    bExecute     : BOOL;
    sFilePath    : T_MaxString := 'C:\Test.xml'; (* CE: '\Hard Disk\Test.xml' *)
    sXPath       : T_MaxString := '/dataentry/MAIN.value1';
    nState       : INT := 0;

```

```

END_VAR

CASE nState OF
0: (* initialize *)
    fbXmlSrvRead(
        pSymAddr    := ADR(value1),
        cbSymSize    := SIZEOF(value1),
        sFilePath    := sFilePath,
        sXPath       := sXPath,
        bExecute     := bExecute
    );
    fbXmlSrvRead(bExecute:= TRUE);
    nState:= 1;

1: (* wait for read operation *)
    fbXmlSrvRead(bExecute:= FALSE);
    IF NOT fbXmlSrvRead.bBusy AND NOT fbXmlSrvRead.bError THEN
        nState:= 2;
    ELSIF fbXmlSrvRead.bError THEN
        nState:= 100;
    END_IF

2: (* operations *)
    ;

100: (* errorState *)
    ;

END_CASE

```

示例 6：循环和事件驱动型写入

下方的示例每 20 秒生成一个新的 XML 文件，并将熟悉的结构写入其中。与往常一样，系统会根据当前的 Windows 日期和时间以及一个字符串生成文件名。也可以通过按下开关（或设置开关变量 bButton）来启动写入进程。如果在 1 秒钟内多次触发写入进程，当前文件将被覆盖。

(* Sample6: Every 20s value1 will be written into a new XML-File named after the current date and time. Furthermore you can activate the printing procedure by pressing a button (or setting the corresponding variable *)

```

PROGRAM Sample6
VAR
    value1          : ST_MyStruct;
    fbXmlSrvWrite    : FB_XmlSrvWrite;

    sFileFolder      : T_MaxString := 'C:\'; (* CE: '\Hard Disk\' *)
    sFileName        : T_MaxString := '_test.xml';
    sFilePathWrite    : T_MaxString
    (*sFilePathWrite = sFileFolder + time + sFileName*)

    sXPathWrite      : T_MaxString := '/dataentry/MAIN.value1';
    ntGetTime        : NT_GetTime;
    stMyTimeStruct    : TIMESTRUCT;
    iState           : INT := 1;
    bTwentySec       : BOOL:= FALSE;
    bButton          : BOOL:= FALSE;
    bTwentySecOver    : BOOL;
    triggerWrite      : R_TRIG;
    triggerButton     : R_TRIG;
END_VAR

triggerButton(CLK:= bButton);

CASE iState OF
0: (* idle state *)
    ;

1: (* initialize *)
    fbXmlSrvWrite(nMode:=XMLSRV_ADDMISSING, pSymAddr:= ADR(value1),
        cbSymSize:= SIZEOF(value1));
    ntGetTime(START:= TRUE, TIMESTR=>stMyTimeStruct); (* get Windows time *)
    IF NOT ntGetTime.BUSY AND NOT ntGetTime.ERR THEN
        iState:= 2;
    ELSIF ntGetTime.ERR THEN
        iState:= 100;
    END_IF

2: (* working state *)
    (* change some values - replace with production-process *)
    value1.stInner.nInteger:= value1.stInner.nInteger + 1;
    IF value1.stInner.nInteger = 32767 THEN
        value1.stInner.nInteger:= 0;
    END_IF(* get Windows time *)
    ntGetTime(START:= FALSE);
    IF NOT ntGetTime.BUSY AND NOT ntGetTime.ERR THEN
        ntGetTime(START:= TRUE, TIMESTR=>stMyTimeStruct);
    ELSIF ntGetTime.ERR THEN
        iState:= 100;
    END_IF

```

```
(* check if 20s have passed*)
IF stMyTimeStruct.wSecond = 0 OR stMyTimeStruct.wSecond = 20
    OR stMyTimeStruct.wSecond = 40 THEN
    bTwentySecOver:= TRUE;
ELSE
    bTwentySecOver:= FALSE;
END_IF

(* if 20s have passed => trigger writing-process *)
triggerWrite(CLK:=bTwentySecOver);
IF (triggerWrite.Q OR triggerButton.Q) AND NOT fbXmlSrvWrite.bBusy AND NOT fbXmlSrvWrite.bError THEN
    (* create filename *)
    sFilePathWrite:= CONCAT(sFileFolder, SYSTEMTIME_TO_STRING(stMyTimeStruct)); (* set folder + time *)
    sFilePathWrite:= DELETE(STR:= sFilePathWrite, LEN:= 4 , POS:= LEN(STR:=sFilePathWrite)-3); (* delete milliseconds *)
    sFilePathWrite:= REPLACE(STR1:= sFilePathWrite , STR2:= '.' , L:= 1,
        P:= LEN(STR:=sFilePathWrite)-2); (* replace colon with point *)
    sFilePathWrite:= REPLACE(STR1:= sFilePathWrite , STR2:= '.' , L:= 1,
        P:= LEN(STR:=sFilePathWrite)-5); (* replace colon with point *)
    sFilePathWrite:= CONCAT(sFilePathWrite, sFileName); (* add filename (default: test) *)

    (*change value 1*)
    value1.stInner.sString := sFilePathWrite;
    value1.stInner.nInteger := stMyTimeStruct.wSecond;

    (* write *)
    fbXmlSrvWrite(sFilePath:=sFilePathWrite, sXPath:=sXPathWrite, bExecute:= TRUE);

ELSIF fbXmlSrvWrite.bError THEN
    iState:= 100;
END_IF

(* reset fbXmlSrvWrite *)
IF fbXmlSrvWrite.bBusy AND NOT tGetTime.ERR THEN
    fbXmlSrvWrite(bExecute:= FALSE);
ELSIF ntGetTime.ERR THEN
    iState:= 100;
END_IF

100: (* error state*)
;

END_CASE
```

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

5.4 生产示例

示例 7 (生产示例)

本节以某个生产订单为例进行了说明。生产数据（如部件长度、部件宽度等）会被读取到一个结构体中以进行初始化（XML 读取），然后根据相应的数量进行生产，最后通过在 XML 文件中添加一条记录来完成生产订单（写入）。若要启动该程序，首先必须将 XML 文件保存到与文件路径相符的位置，并且在 PLC 程序中，变量 bStart 必须设置为 TRUE。

可在此处下载包含这些示例的 PLC 项目：https://infosys.beckhoff.com/content/1033/TF6421_Tc3_XML_Server/Resources/1635013643.zip

变量声明

```
PROGRAM Sample7
VAR
    fbXmlSrvReadByName      : FB_XmlSrvReadByName;
    fbXmlSrvWriteByName     : FB_XmlSrvWriteByName;
    value                   : ST_MyProductionStruct;
    state                   : INT := 0;
    R_Edge                  : R_TRIG;
    bStart                  : BOOL;
    bError                  : BOOL;
    nErrId                  : UDINT;
END_VAR
```

结构 ST_MyProductionStruct

```

TYPE ST_MyProductionStruct :
STRUCT
    rLength      : REAL;
    rWidth       : REAL;
    rHeight      : REAL;
    iQuantity    : INT;
    iCounter     : INT;
    bReady       : BOOL;
    stInfo       : STRING;
END_STRUCT
END_TYPE

```

PLC 程序

```

(* The production data is read, the production is carried out and, according to the
   quantity and the production order, completed with an entry in the XML file.
   To start the program the XML file needs to be stored in the corresponding folder (sFilePath) and the
   variable bStart needs to be set TRUE in the PLC program. *)

R Edge (CLK := bStart);
IF R Edge.Q THEN
    state := 1;
END_IF

CASE state OF
0: (* idle state *)
    ;

1: (* init state *)
    fbXmlSrvReadByName( sNetId := '',
                        sSymName := 'Sample7.value',
                        sFilePath := 'C:\Production1.xml',
                        sXPath := '/dataentry/MAIN.value',
                        bExecute := TRUE,
                        tTimeout := t#10s,
                        bError => bError,
                        nErrId => nErrId);
    state := 2;

2:
    fbXmlSrvReadByName(bExecute := FALSE);
    IF NOT fbXmlSrvReadByName.bBusy AND NOT fbXmlSrvReadByName.bError THEN
        state := 3;
    ELSIF fbXmlSrvReadByName.bError THEN
        state := 100;
    END_IF

3: (* working state *)
    IF value.bReady = TRUE THEN
        value.stInfo := 'The order was already processed!';
        (* replace your production XML file! *)
        state := 4;
        RETURN;
    END_IF

    (* call production program with
       new length, width and height here *)
    value.iCounter := value.iCounter + 1;
    IF value.iCounter = value.iQuantity THEN
        value.bReady := TRUE;
        state := 4;
    END_IF

4: (* documentation state *)
    fbXmlSrvWriteByName( sNetId := '',
                        nMode := XMLSRV_SKIPMISSING,
                        sSymName := 'Sample7.value',
                        sFilePath := 'C:\Production1.xml',
                        sXPath := '/dataentry/MAIN.value',
                        bExecute := TRUE,
                        tTimeout := t#10s,
                        bError => bError,
                        nErrId => nErrId);
    state := 5;

5:
    fbXmlSrvWriteByName(bExecute := FALSE);
    IF NOT fbXmlSrvWriteByName.bBusy AND NOT fbXmlSrvWriteByName.bError THEN
        state := 0;
    ELSIF fbXmlSrvWriteByName.bError THEN
        state := 100;
    END_IF

100: (* error state *)
    ;

```

XML 文件

```
<dataentry>
  <MAIN.value>
    <rLength>65.85</rLength>
    <rWidth>30</rWidth>
    <rHeight>2.5</rHeight>
    <iQuantity>500</iQuantity>
    <iCounter>0</iCounter>
    <bReady>false</bReady>
    <stInfo></stInfo>
  </MAIN.value>
</dataentry>
```

要求

开发环境	目标系统类型	要链接的 PLC 功能库
TwinCAT v3.1 Build 4011	PC 或 CX (x86、x64、Arm®)	Tc2_XmlDataSrv

6 附录

6.1 TC3 XML Server 错误代码概述

偏移 + 错误代码	范围	描述
0x00000000 + TwinCAT 系统错误代码 [► 34]	0x00000000-0x00007800	TwinCAT 系统错误 (包括 ADS 错误代码)
0x00008000 + TC3 XML Server 内部错误 [► 37]	0x00008000-0x000080FF	TC3 XML Server 内部错误代码

6.2 ADS 返回代码

错误代码分组：

全局错误代码：0x0000 [► 34]... (0x9811_0000 ...)

路由器错误代码：0x0500 [► 35]... (0x9811_0500 ...)

一般 ADS 错误：0x0700 [► 35]... (0x9811_0700 ...)

RTime 错误代码：0x1000 [► 36]... (0x9811_1000 ...)

全局错误代码

Hex	Dec	HRESULT	名称	描述
0x0	0	0x98110000	ERR_NOERROR	无错误。
0x1	1	0x98110001	ERR_INTERNAL	内部错误。
0x2	2	0x98110002	ERR_NORTIME	不具有实时性。
0x3	3	0x98110003	ERR_ALLOCCLOCKEDMEM	分配已锁定 – 内存错误。
0x4	4	0x98110004	ERR_INSERTMAILBOX	邮箱已满 – 无法发送 ADS 消息。减少每个周期的 ADS 消息数量将有所帮助。
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	HMSG 错误。
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	未找到目标端口 – ADS 服务器未启动或无法访问。
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	未找到目标计算机 – 未找到 AMS 路由。
0x8	8	0x98110008	ERR_UNKNOWNCMDID	未知命令 ID。
0x9	9	0x98110009	ERR_BADTASKID	任务 ID 无效。
0xA	10	0x9811000A	ERR_NOIO	No IO。
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	未知 AMS 命令。
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 错误。
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	端口未连接。
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	AMS 长度无效。
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	AMS Net ID 无效。
0x10	16	0x98110010	ERR_LOWINSTLEVEL	安装级别 – TwinCAT 2 授权错误。
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	无调试可用。
0x12	18	0x98110012	ERR_PORTDISABLED	端口已禁用 – TwinCAT 系统服务未启动。
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	端口已连接。
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 错误。
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync 超时。
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync 错误。
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	不存在适用于 AMS Sync 的索引映射。
0x18	24	0x98110018	ERR_INVALIDAMSPORT	AMS 端口无效。
0x19	25	0x98110019	ERR_NOMEMORY	无内存。
0x1A	26	0x9811001A	ERR_TCPSEND	TCP 发送错误。
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	主机无法访问。
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	AMS 片段无效。
0x1D	29	0x9811001D	ERR_TLSEND	TLS 发送错误 – ADS 安全连接失败。
0x1E	30	0x9811001E	ERR_ACCESSDENIED	拒绝访问 – 拒绝 ADS 安全访问。

路由器错误代码

Hex	Dec	HRESULT	名称	描述
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	无法分配锁定的内存。
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	路由器内存大小无法更改。
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	邮箱已达到最大消息数。
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	调试邮箱已达到最大消息数。
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	端口类型未知。
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	路由器未初始化。
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	端口号已分配。
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	端口未注册。
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	已达到最大端口数。
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	端口无效。
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	路由未激活。
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	邮箱已达到最大片段消息数。
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	发生片段超时。
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	端口已移除。

一般性 ADS 错误代码

Hex	Dec	HRESULT	名称	描述
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	一般性设备错误。
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	服务器不支持该服务。
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	索引组无效。
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	索引偏移量无效。
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	不允许读取或写入。 可能有几种原因。例如，创建路由时输入了错误的密码。
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	参数大小不正确。
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	无效数据值。
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	设备尚未准备好运行。
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	设备正忙。
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	操作系统上下文无效。这可能是由于在不同的任务中使用 ADS 函数块造成的。可以通过在 PLC 中进行多任务同步解决这个问题。
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	内存不足。
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	参数值无效。
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	未找到（文件…）。
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	文件或命令中存在语法错误。
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	对象不匹配。
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	对象已存在。
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	符号未找到。
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	符号版本无效。这可能是由于联机更改造成的。创建新句柄。
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	设备（服务器）处于无效状态。
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	不支持 AdsTransMode。
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	通知句柄无效。
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	通知客户端未注册。
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLs	没有更多的句柄可用。
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	通知大小过大。
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	设备未初始化。
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	设备超时。
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	接口查询失败。
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	请求的接口错误。

Hex	Dec	HRESULT	名称	描述
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID 无效。
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID 无效。
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	请求待定。
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	请求已中止。
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	警告信号。
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	数组索引无效。
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	符号未激活。
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	拒绝访问。 有几种可能的原因。例如，单向 ADS 路由用于相反方向。
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	缺少授权。
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	授权已过期。
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	超出授权。
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	授权无效。
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	授权问题：系统 ID 无效。
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	授权不受时间限制。
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	授权问题：未来的时间。
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	授权期限太长。
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	系统启动异常。
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	授权文件读取了两次。
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	签名无效。
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	证书无效。
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	OEM未知公钥。
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	此系统 ID 授权无效。
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	演示授权已禁止。
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCD	无效函数 ID。
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	超出有效范围。
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	无效对齐。
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	无效平台级别。
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	上下文 – 转到被动级别。
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	上下文 – 转到调度级别。
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	上下文 – 转到实时。
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	客户端错误。
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	服务包含无效参数。
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	轮询列表为空。
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var 连接已在使用中。
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	调用的 ID 已在使用中。
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	已发生超时 – 远程终端在指定的 ADS 超时中未响应。远程终端的路由设置可能配置不正确。
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Win32 子系统中发生错误。
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	客户端超时值无效。
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	端口未打开。
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	无 AMS 地址。
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Ads sync 中发生内部错误。
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	哈希表溢出。
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	未在表格中找到密钥。
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESVM	缓存中没有符号。
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	收到无效响应。
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	同步端口已锁定。
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	请求被取消。

RTime 错误代码

Hex	Dec	HRESULT	名称	描述
0x1000	4096	0x98111000	RTERR_INTERNAL	实时系统中发生内部错误。
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	计时器值无效。

Hex	Dec	HRESULT	名称	描述
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	任务指针提供了无效值 0（零）。
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	堆栈指针提供了无效值 0（零）。
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	请求任务优先级已分配。
0x1005	4101	0x98111005	RTERR_NOMORETCB	没有可用的空闲 TCB（任务控制块）。TCB 的最大数量为 64。
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	没有可用的空闲信号。信号的最大数量为 64。
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	队列中没有可用的空闲空间。队列中的最大位置数为 64。
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	已应用外部同步中断。
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	未应用外部同步中断。
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	外部同步中断应用失败。
0x1010	4112	0x98111010	RTERR_IRQNOTLESSOREQUAL	在错误的上下文中调用服务函数
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	不支持 Intel® VT-x 扩展。
0x1018	4120	0x98111018	RTERR_VMXDISABLED	BIOS 中未启用 Intel® VT-x 扩展。
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Intel® VT-x 扩展中缺少函数。
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Intel® VT-x 激活失败。

具体的正向 HRESULT 返回代码：

HRESULT	名称	描述
0x0000_0000	S_OK	无错误。
0x0000_0001	S_FALSE	无错误。 示例：处理成功，但有一个负面或不完整的结果。
0x0000_0203	S_PENDING	无错误。 示例：处理成功，但还没有结果。
0x0000_0256	S_WATCHDOG_TIMEOUT	无错误。 示例：处理成功，但发生了超时。

TCP Winsock 错误代码

Hex	Dec	名称	描述
0x274C	10060	WSAETIMEDOUT	发生连接超时 - 在创建连接时发生错误，因为远程终端在特定时间后未正确响应，或者所建立连接因所连接主机未响应而无法维持。
0x274D	10061	WSAECONNREFUSED	连接遭到拒绝 - 无法建立连接，因为目标计算机明确拒绝了该连接。此错误通常由尝试连接到外部主机上处于非活动状态的服务（即没有运行服务器应用程序的服务）导致。
0x2751	10065	WSAHOSTUNREACH	不存在至主机的路由 - 套接字操作引用了不可用的主机。
更多 Winsock 错误代码： Win32 错误代码			

6.3 TwinCAT 3 XML Server 的内部错误代码

编码	描述	符号名称
0x00008000	内部错误	XMLSRERROR_INTERNAL
0x00008001	未找到句柄。	XMLSRERROR_NOTFOUND
0x00008002	在解析 XML 文件时出错	XMLSRERROR_PARSERERROR
0x00008003	数据类型不兼容。	XMLSRERROR_INCOMPATIBLE
0x00008004	在分配内存时出错	XMLSRERROR_NOMEMORY
0x00008005	在添加 XML 节点时出错	XMLSRERROR_ADDNODE
0x00008006	sXPath 无效	XMLSRERROR_INVALIDXPath
0x00008007	TC 中的字符串无效	XMLSRERROR_INVALIDSTRING
0x00008800	客户端句柄无效	XMLSRERROR_INVALIDCLIENTHANDLE
0x00008900	使用了错误的 TcXmlDataSrv.exe（适用于 TC2 而非 TC3）	XMLSRERROR_INVALIDTWINCATVERSION

6.4 其他注册表项

本节概述了可用于静态影响 TwinCAT 3 XML Server 行为的注册表项。因此，在运行时无法进行切换；该设置是唯一的，只有在重启后才会生效。

系统将这些项列于一个表格中，并且所有项均应创建在以下目录中：

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Beckhoff\TwinCAT3 Functions\TF6421 XML Server]

注册表项名称	类型	默认值（十进制）	描述
Priority	DWORD	1	进程的 Windows 优先级（1（正常）、2（高）或 3（实时）） - 只有当优先级与运行进程的默认优先级不同时才会生效。
NumWorkerThreads	DWORD	10	指定了工作线程的数量。
AdsBlockSize	DWORD	262,144	从 PLC 读取当前数据的功能块的大小（以字节为单位）。
XmlIdentSize	DWORD	0	每个 xml 等级的制表符/空格数量。
XmlUseSpaces	DWORD	0	若为 1，则将空格用作制表符。
LogLevel	DWORD	1	指定了日志记录等级：1、2 或 3。所有类型的消息均记录在第 3 级。
LogMaxFileSize	DWORD	10,000	指定了日志文件占用的最大空间（以字节为单位）。
LogWriteThrough	DWORD	0	如果 !=0，则绕过内部缓存，并尝试直接写入存储介质。0=关闭。
XmlTrimStrings	DWORD	0	如果 !=0，读取 STRING/WSTRING 时将会删除其左右两侧的空格。

6.5 FAQ - 常见问题及其解答

本节将对常见问题进行解答，以便于您使用 XML Server。

如果您还有任何其他问题，请联系我们的技术支持部门：+49(0)5246/963-157)

XML 文件中存储了哪些信息？ [► 38]

是否可以同时向 XML 文件写入多个变量？ [► 38]

是否可以在网络上访问 XML 文件？ [► 39]

XML Server 能否自动生成 XML 文件？ [► 39]

如果 XML Server 无法转换某个变量（例如，XML 文件中的整型变量的值为“hello”），会发生什么情况？ [► 39]

是否支持别名数据类型？ [► 39]

是否可以将结构体中的某些变量从写入进程中排除？ [► 39]

? XML 文件中存储了哪些信息？

! XML 文件中只会存储变量的名称及其值，而不会存储其数据类型。此外，自第 3.2.31.0 版产品起，还可以将 PLC 注释写入 XML 文件。有关更多信息，请参见：“[入门 \[► 26\]](#)”。

? 是否可以同时将多个变量写入 XML 文件？

! 是的，可以。因此，必须用一个结构体来封装您的变量。然后即可写入此类型的变量。[示例 \[► 26\]](#)中对此进行了演示。

? 是否可以在网络上访问 XML 文件?

! 不可以。sPath 只能指向本地文件系统。

? XML-Server 是否具备自动创建文件的功能?

! 是的，具备此功能。若要使用此功能，必须设置 nMode:=1 (XMLSRV_ADDMISSING)。XML Server 将自动创建文件和文件中的各个部分。它不会创建文件夹!

? 如果 XML Server 因类型不匹配而无法转换某个变量（例如，INT 在 XML 文件中的值为“hallo”），会发生什么情况?

! 在这种情况下，它将跳过该变量。这样不会导致出错。

? 是否支持别名数据类型?

! 是的，从第 3.2.31.0 版产品开始，将会支持别名数据类型。

? 是否可以将结构体中的某些变量从写入进程中排除?

! 可以，从第 3.2.37.0 | 32.37.0 版开始，可以借助属性实现这一点。必须将属性写在声明中相应变量的前面，内容如下：

```
{attribute 'TcXmlServer' := 'exclude'}
```

7 技术支持和服务

倍福公司及其合作伙伴在世界各地提供全面的技术支持和服务，对与倍福产品和系统解决方案相关的所有问题提供快速有效的帮助。

下载搜索器

我们的下载搜索器包含我们供您下载的所有文件。您可以通过它搜索我们的应用案例、技术文档、技术图纸、配置文件等等。

可供下载的文件格式多种多样。

倍福分公司和代表处

若需要倍福产品的本地支持和服务，请联系倍福分公司或代表处！

倍福遍布世界各地的分公司和代表处地址可在倍福官网上找到：<http://www.beckhoff.com.cn>

该网页还提供更多倍福产品组件的文档。

倍福技术支持

技术支持部门为您提供全面的技术援助，不仅帮助您应用各种倍福产品，还提供其他广泛的服务：

- 技术支持
- 复杂自动化系统的设计、编程和调试
- 以及倍福系统组件的各种培训课程

热线电话：+49 5246 963-157

电子邮箱：support@beckhoff.com

倍福售后服务

倍福服务中心提供所有售后服务：

- 现场服务
- 维修服务
- 备件服务
- 热线服务

热线电话：+49 5246 963-460

电子邮箱：service@beckhoff.com

倍福公司总部

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

电话：+49 5246 963-0

电子邮箱：info@beckhoff.com

网址：www.beckhoff.com

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

DALI, DALI-2, D4i, DALI+, and DiiA are trademarks in various countries in the exclusive use of the Digital Illumination Interface Alliance.

Excel, IntelliSense, Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

MAX®, Stratix®, Cyclone®, Altera®, Agilex™, Arria®, Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

更多信息:
www.beckhoff.com/tf6421

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
电话号码: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

