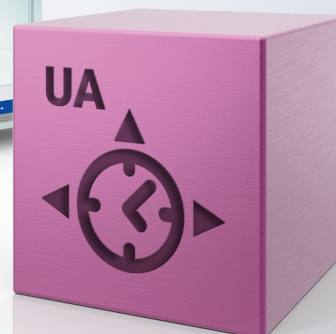
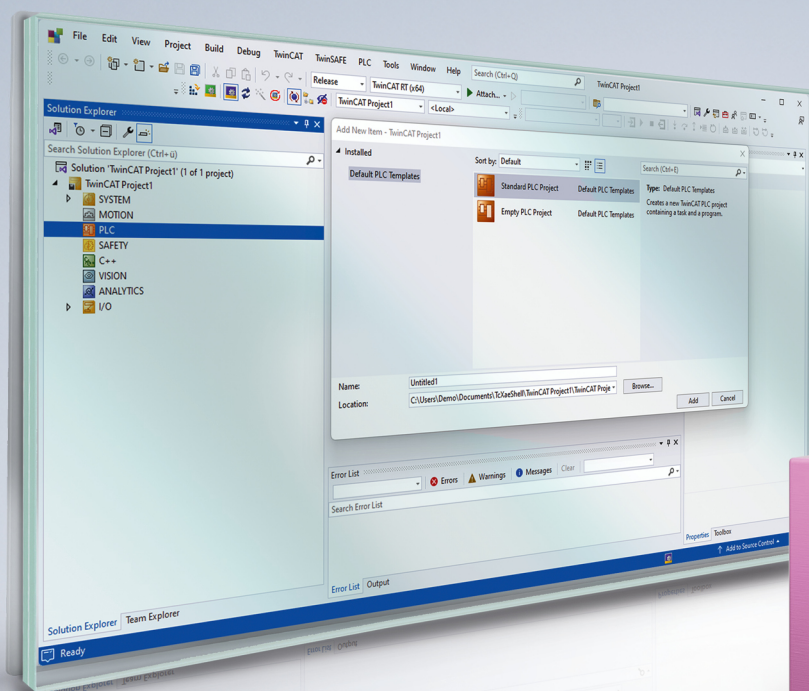


手册 | ZH

TF6105

TwinCAT 3 | OPC UA Pub/Sub



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
1.4 文档发行状态	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	10
3.4 容器	12
4 技术简介	15
4.1 快速入门	15
4.1.1 UDP	15
4.1.2 MQTT	19
4.2 支持的功能	23
4.3 协议概述	25
4.4 发布者/订阅者	27
4.5 数据集	28
4.6 传输协议	33
4.6.1 UDP	33
4.6.2 MQTT	33
4.7 报头布局	36
4.8 关键帧和增量帧	36
4.9 配置导入/导出	38
4.10 深入	40
4.10.1 发布间隔时间	40
4.10.2 KeyFrames、DeltaFrames、KeepAlive	41
4.10.3 MessageReceiveTimeout	42
5 示例	43
5.1 UDP	43
5.2 MQTT	44
5.3 EtherCAT 主站	45
6 教程	52
7 附录	53
7.1 诊断	53
7.2 故障排除	54

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。

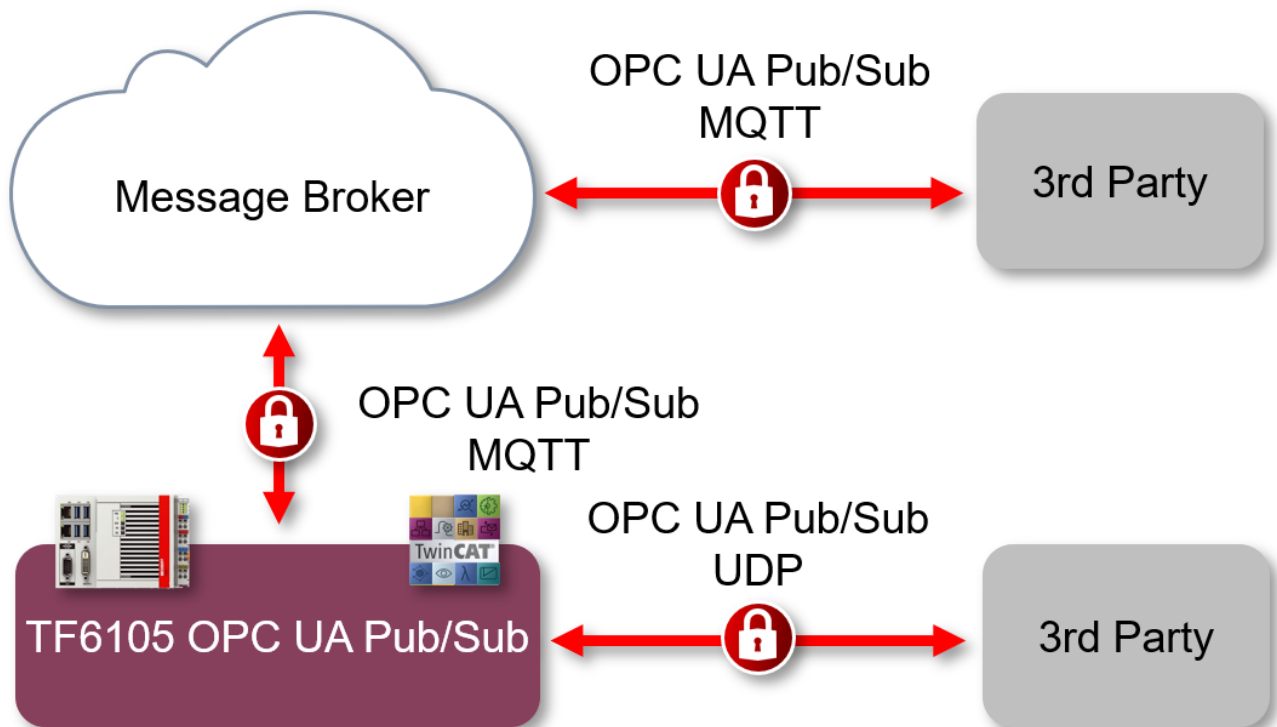
如需了解 Beckhoff 产品信息安全的信息，请订阅 <https://www.beckhoff.com/secinfo> 上的 RSS 源。

1.4 文档发行状态

日期	修改
2.0.x	文档已全面修订，各章节均有修改和新增内容

2 概述

TF6105 TC3 OPC UA Pub/Sub 产品支持通过 OPC UA 规范 (OPC10000-14) 中 Pub/Sub 扩展所规定的不同传输协议进行连接。规格书中列出的多种功能已实现，并可供使用。



本产品内含：

- 实时驱动程序，用于通过 UDP 单播或多播实现 OPC UA Pub/Sub 通信
- 实时驱动程序，用于通过 MQTT 实现 OPC UA Pub/Sub 通信
- 用于配置 OPC UA Pub/Sub 的 TwinCAT System Manager 扩展软件
- 用于扫描远程 OPC UA 服务器设备的 TwinCAT System Manager 扩展软件
- 可选配集成至 TwinCAT OPC UA 服务器 (TF6100 TC3 OPC UA) 中，以便使用基于标准化 OPC UA Pub/Sub 信息模型的配置界面。

3 安装

3.1 系统要求

技术数据	描述
操作系统	Windows 10 Windows 11 (TwinCAT 3.1 Build >= 4026.16) TwinCAT/BSD TwinCAT/Beckhoff RT Linux®
目标平台	PC 架构 (x86、x64)
最低 TwinCAT 版本	TwinCAT 3.1 Build 4026
需要的 TwinCAT 设置级别	TwinCAT 3 XAE、XAR
所需 TwinCAT 授权	TF6105 TC3 OPC UA Pub/Sub
硬件要求	本产品支持不同传输协议。如果要使用 UDP，则需要一块与 TwinCAT 实时以太网适配器驱动程序兼容的网络接口卡，详见下文。如果您想使用 MQTT，则无需专门的网络接口卡。

实时以太网适配器

对于 UDP 通信，本产品基于 TwinCAT 实时以太网适配器驱动程序。要检查您的网络接口卡是否适用于该驱动程序，请按照以下步骤操作。请注意，如果您仅想通过 MQTT 使用 OPC UA Pub/Sub 功能，则无需安装此驱动程序。

1. 右键点击 **Devices**（设备）并选择 **Add New Item**（添加新项目），创建 OPC UA RT 设备。
2. 展开“OPC UA”类别，然后选择 **Real-Time OPC UA Device**（实时 OPC UA 设备）。
3. 现在选择 **Compatible Devices**（兼容设备），以此选择适配器并查找相应的以太网接口。
4. 请尝试为您的网络接口卡安装 TwinCAT Realtime-Ethernet-Adapter 驱动程序。



“仅供演示使用”

如果您的网络接口卡列在“仅供演示使用”类别下，则说明已安装 TwinCAT Realtime-Ethernet-Adapter 驱动程序，但该驱动程序可能不具备实时功能。然而，您仍可将其用于评估和测试。

3.2 安装

如果在 Windows 操作系统上使用 TwinCAT 3.1 Build 4026（及更高版本），则可以通过 TwinCAT Package Manager 安装此功能，请参见安装文档。通常情况下，可通过相应的工作负载安装此功能。然而，您也可以单独安装工作负载中包含的软件包。本文档简要介绍了通过工作负载进行安装的过程。

命令行程序 TcPkg

您可以使用 TcPkg Command Line Interface（命令行界面，CLI）来显示系统上的可用工作负载：

```
tcpkg list -t workload
```

您可使用以下命令安装此功能的工作负载。

```
tcpkg install TF610x.OpcUaClientPubSub.XAE
tcpkg install TF610x.OpcUaClientPubSub.XAR
```

TwinCAT Package Manager UI

您可以使用 User Interface（用户界面，UI）显示所有可用的工作负载，并根据需要进行安装。

为此，请按照界面中的相应说明操作。

可以通过 TcPkg CLI 或 TcPkg UI 安装相应的软件包或工作负载。

● 未做好准备的 TwinCAT 重启会导致数据丢失

i

安装该功能可能会导致 TwinCAT 重启。

确保系统上没有运行重要的 TwinCAT 应用程序，或者先有序关闭这些应用程序。

Beckhoff RT Linux®

1. 直接在设备上使用 CLI。或者，您也可以通过 SSH 连接。
 2. 以管理员身份登录。
 3. 更新 Package Manager 中的软件包。
`sudo apt update`
 4. 搜索现有的 TF6100 软件包。
`sudo apt search tf610x`
 5. 通过 Package Manager 安装 TF6100 OPC UA 服务器。
`sudo apt install tf610x-opc-ua-client-pubsub`
- ⇒ TF6100 OPC UA 服务器已安装。

```
Administrator@PC-983B8A: ~$ sudo apt install tf6100-opc-ua-server
Installing:
  tf6100-opc-ua-server

Installing dependencies:
  libtceventloggeradsproxy

Summary:
  Upgrading: 0, Installing: 2, Removing: 0, Not Upgrading: 14
  Download size: 6386 kB
  Space needed: 26.8 MB / 18.2 GB available

Continue? [Y/n] y
Get:1 https://deb.beckhoff.com/debian trixie-stable/main amd64 libtceventloggeradsproxy amd64 2.9.1-1 [492 kB]
Get:2 https://deb.beckhoff.com/debian trixie-stable/main amd64 tf6100-opc-ua-server amd64 5.2.124-1 [5894 kB]
Fetched 6386 kB in 1s (9796 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/
perl5/Debconf/FrontEnd/Dialog.pm line 79, <STDIN> line 2.)
debconf: falling back to frontend: Readline
Selecting previously unselected package libtceventloggeradsproxy.
(Reading database ... 16296 files and directories currently installed.)
Preparing to unpack .../libtceventloggeradsproxy_2.9.1-1_amd64.deb ...
Unpacking libtceventloggeradsproxy (2.9.1-1) ...
Selecting previously unselected package tf6100-opc-ua-server.
Preparing to unpack .../tf6100-opc-ua-server_5.2.124-1_amd64.deb ...
Unpacking tf6100-opc-ua-server (5.2.124-1) ...
Setting up libtceventloggeradsproxy (2.9.1-1) ...
Setting up tf6100-opc-ua-server (5.2.124-1) ...
Processing triggers for libc-bin (2.41-12) ...
Administrator@PC-983B8A: ~$
```

TwinCAT/BSD

1. 直接在设备上使用 CLI。或者，您也可以通过 SSH 连接。
 2. 以管理员身份登录。
 3. 更新 Package Manager 中的软件包。
`doas pkg update`
 4. 搜索现有的 TF610x 软件包。
`doas pkg search TF610x`
 5. 通过 Package Manager 安装 TF610x OPC UA Client PubSub。
`doas pkg install TF610x-OPC-UA-Client-PubSub`
- ⇒ 已安装 TF6100 OPC UA 客户端。

```
Administrator: PowerShell
Administrator@PC-70E9BB:~ $ doas pkg install TF610x-OPC-UA-Client-PubSub
Password:
Updating TCBSD repository catalogue...
TCBSD repository is up to date.
All repositories are up to date.
The following 1 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
  TF610x-OPC-UA-Client-PubSub: 1.8.131

Number of packages to be installed: 1

The process will require 2 MiB more space.
540 KiB to be downloaded.

Proceed with this action? [y/N]: y
[1/1] Fetching TF610x-OPC-UA-Client-PubSub-1.8.131: 100% 540 KiB 552.9 k/s 00:01
Checking integrity... done (0 conflicting)
[1/1] Installing TF610x-OPC-UA-Client-PubSub-1.8.131...
[1/1] Extracting TF610x-OPC-UA-Client-PubSub-1.8.131: 100%
Administrator@PC-70E9BB:~ $
```

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

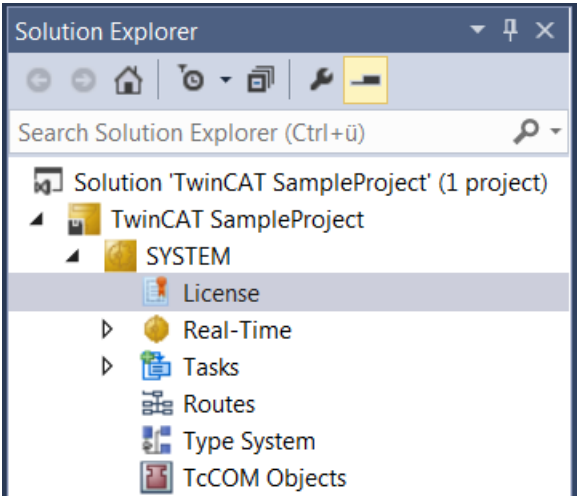
关于许可证完整版本的程序描述，可参见倍福信息系统的文档“[TwinCAT 3 授权](#)”。

授权一个 TwinCAT 3 功能的 7 天测试版本

i 对于TwinCAT 3 许可证加密狗无法启用 7 天测试版本。

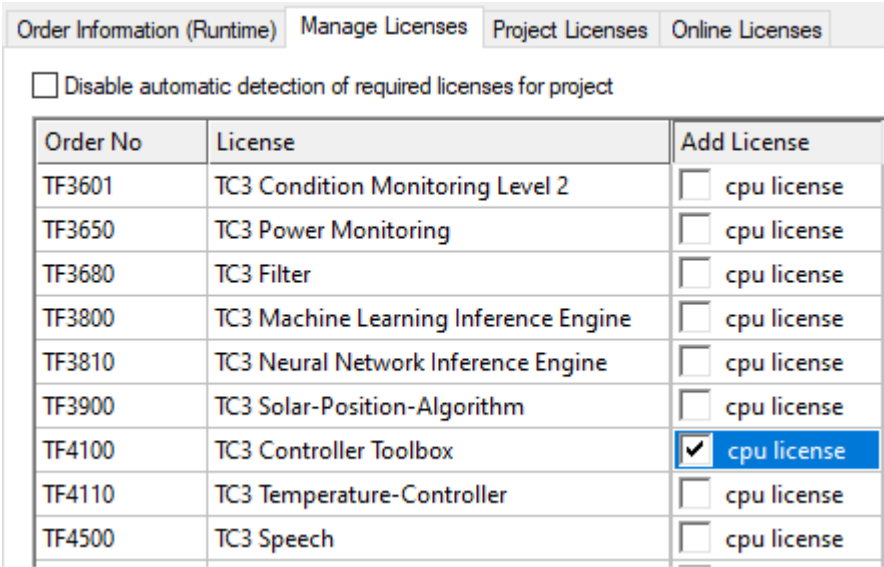
1. 启动 TwinCAT 3 开发环境 (XAE) 。
2. 打开一个现有的 TwinCAT 3 项目或创建一个新项目。
3. 如果想为一个远程设备激活授权，请设置所需的目标系统。为此，从工具栏的**选择目标系统**下拉列表中选择目标系统。
 - ⇒ 授权设置总是指选定的目标系统。在目标系统上激活项目时，自动将相应的 TwinCAT 3 许可证复制到该系统中。

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 容器

TwinCAT Runtime for Linux® 是一款功能强大的全新操作系统，专为倍福控制器打造，兼具 TwinCAT 的实时性与先进 Linux® 系统的灵活性。其核心创新之一是对 Docker® 等容器技术的原生支持，首次实现了 TwinCAT 功能组件以容器应用程序的形式进行安装和运行。

Docker® 容器的使用为 TwinCAT 功能组件的提供、版本化和维护开辟了新的可能性。用户可以对功能组件进行封装、重复部署，并独立于系统的其余部分进行更新。同时，用户无需放弃倍福控制平台的确定性实时特性，即可使用成熟的 Linux® 和 DevOps 工作流程。

本文将介绍如何在 Linux® 系统下，在 Docker® 容器中安装和运行此 TwinCAT 功能组件。其旨在以通俗易懂的方式介绍基本概念、前提条件和步骤，从而为实际使用基于容器的 TwinCAT 功能组件奠定了基础。

以下示例设置展示了在 Linux® 系统下以 Docker® 容器形式运行 TwinCAT 功能组件的典型系统架构。TwinCAT 功能组件以独立容器的形式运行，并通过指定的接口集成到 TwinCAT 系统中。

该控制器基于 TwinCAT Runtime for Linux® 运行，并提供了 TwinCAT 的实时环境和容器运行时环境（Docker®）。TwinCAT 功能组件在 Docker® 容器内运行，并使用系统提供的资源以及相应的 TwinCAT 接口与控制器和其他应用程序进行通信。在本例中，该功能组件与 TwinCAT PLC 运行时在同一容器中一起运行，但这些组件也可以独立运行。

可以使用 ADS-over-MQTT 进行多个容器之间的 TwinCAT 端通信。通过将 MQTT 作为中间传输层，各个 TwinCAT 组件之间实现了解耦，从而大大简化了 ADS 路由的设置和管理工作。

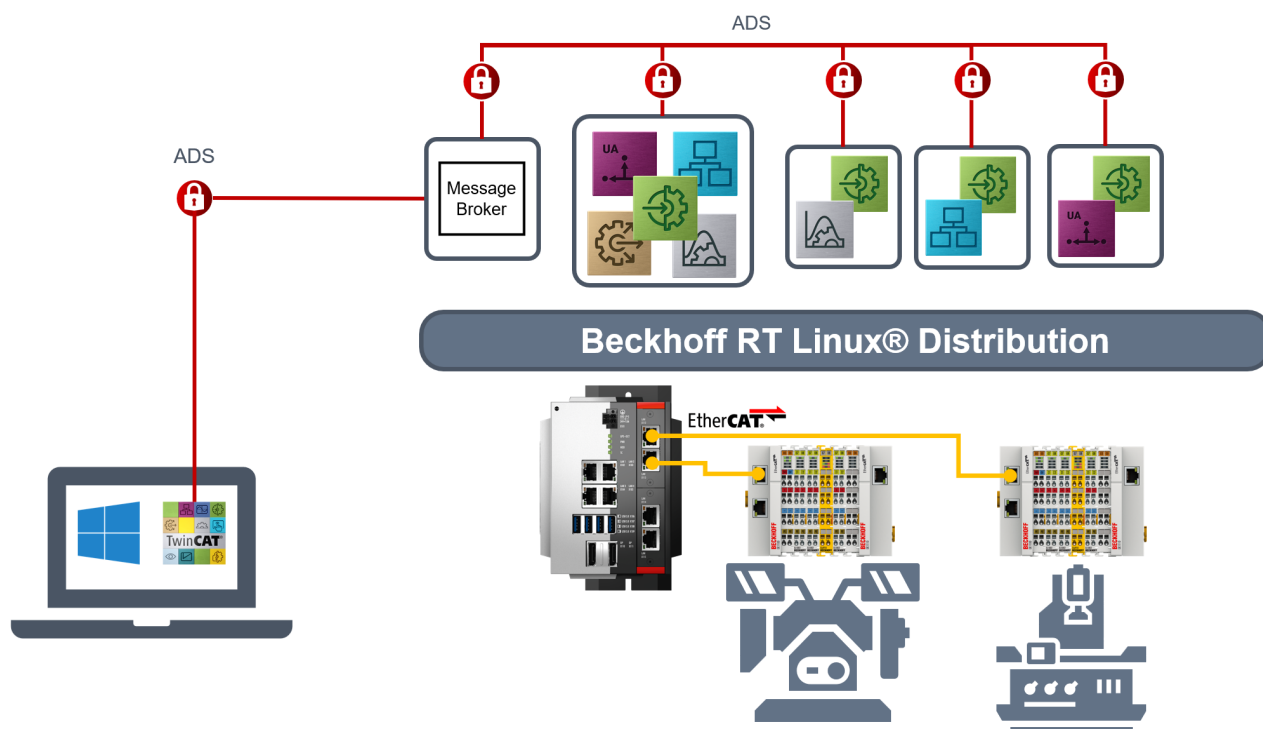
相关容器通过共享的 MQTT 消息代理进行通信，而不是通过特定的静态点对点 ADS 路由进行通信。因此，更改 IP 地址、容器名称或部署结构时，通常不需要对 ADS 路由作任何调整，这一点对于容器化和动态环境的优势尤为明显。

另外，ADS-over-MQTT 所需的消息代理也可以作为 Docker® 容器在倍福控制器或独立系统上运行，从而无缝集成到基于容器的系统结构中。



注意

下图为概念性概览，并不代表完整的系统配置。



设置概述

以下步骤概述了将 TwinCAT 功能组件作为 Docker® 容器进行安装和调试的基本程序。有关详细说明和完整配置示例，可访问链接中的 GitHub 存储库（参见下文）。

查看要求

确保控制器配备了 TwinCAT Runtime for Linux®, 并且安装和激活了 Docker®, 作为容器运行时使用。

创建容器图像

通过 Docker® 文件创建一个新容器图像，图像中包含该 TwinCAT 功能组件和所有附属组件。

配置容器

使用 Docker® Compose File 配置容器运行时所需的参数。此类参数的示例如下：

- 网络和通信接口的设置
- 访问权限和资源
- TwinCAT 专用设置（例如 ADS 路由）
- 用于存储用户专用配置文件的卷映射

启动容器

在倍福控制器上启动 Docker® 容器，并检查 TwinCAT 功能组件是否正常启动。

集成到 TwinCAT 系统中

将 TwinCAT 功能组件集成到现有的 TwinCAT 项目中，并对通信和功能进行测试。

监控和维护

监控容器状态，评估日志，并在必要时执行更新或重启。

下方的 GitHub 存储库提供了一个完整的可执行示例，其中包含了详细的文档、配置文件和源代码：

https://www.github.com/Beckhoff/TF6105_Samples

4 技术简介

4.1 快速入门

下一章将指导您在 TwinCAT 中完成 OPC UA Pub/Sub 配置的首次设置。在继续操作之前，请确保您已按照安装 [► 8] 说明进行操作，并且您的系统符合系统要求 [► 8]。提供以下快速入门教程：

教程	描述
UDP [► 15]	演示如何设置 PLC 项目、OPC UA RT 设备和发布者，该发布者通过 UDP 发送包含一个变量的数据集。该变量通过过程映像与 PLC 项目中的变量相关联。
MQTT [► 19]	演示如何设置 PLC 项目、OPC UA RT 设备和发布者，该发布者通过 MQTT 发送包含一个变量的数据集。该变量通过 TwinCAT Target Browser 与 PLC 项目中的变量相关联。

离线配置交互

OPC UA Pub/Sub 定义了一种基于 UA 二进制格式的离线配置交互文件格式（见 OPC UA 规范第 5.2 章）。如果您想在 2 台不同的设备之间建立 OPC UA Pub/Sub 配置，我们强烈建议使用此通用交换文件。配置导入/导出 [► 38] 一章中包含有关此主题的更多信息。

4.1.1 UDP

本快速入门教程演示了如何设置 PLC 项目、OPC UA RT 设备和发布者，该发布者通过 UDP 发送包含一个变量的数据集。该变量通过过程映像与 PLC 项目中的变量相关联。

创建 PLC 项目

创建 PLC 项目，其中定义一个数据类型为 INT 的输出变量，例如：

```
PROGRAM MAIN
VAR
  nCounter AT%Q* : INT;
END_VAR
nCounter := nCounter + 1;
```

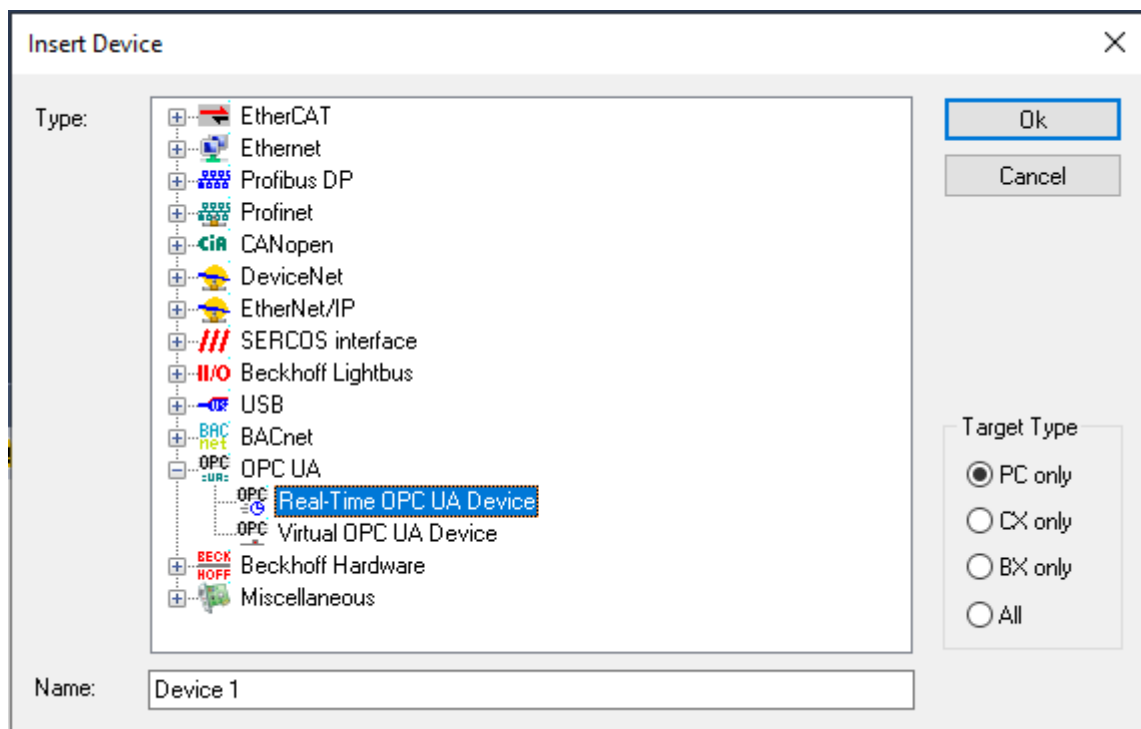
编译 PLC 项目，以生成 PLC 过程映像。

配置发布者

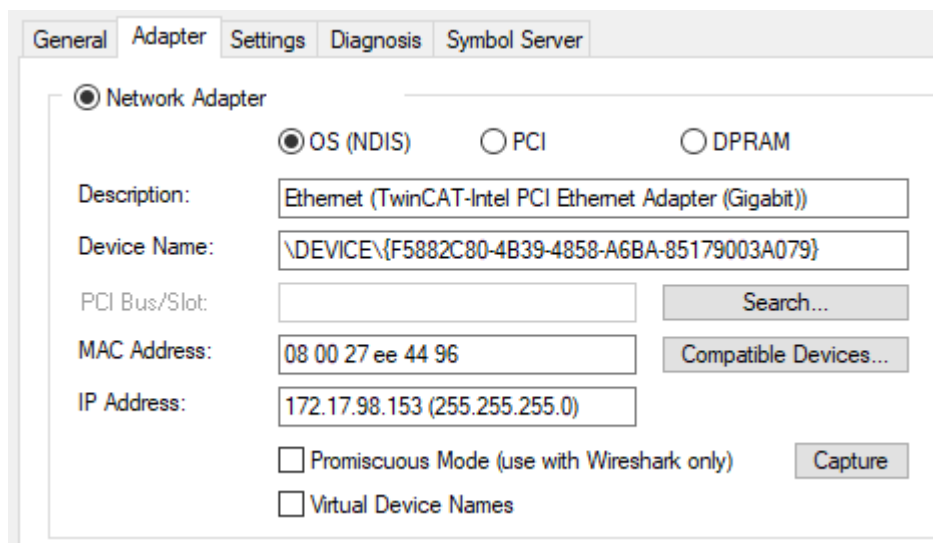
请按照以下配置步骤设置 OPC UA Publisher 设备：

1. 确保 TwinCAT 处于配置模式。
2. 在 Solution Explorer（解决方案资源管理器）中，展开 I/O 节点，然后选择其子项 Devices（设备）。
3. 右键单击 Devices（设备），然后选择 **Add new element...**（添加新元素.....）

4. 展开 **OPC UA** 类别，选择 **Real-Time OPC UA Device**（实时 OPC UA 设备），然后点击 **OK**（确定）。

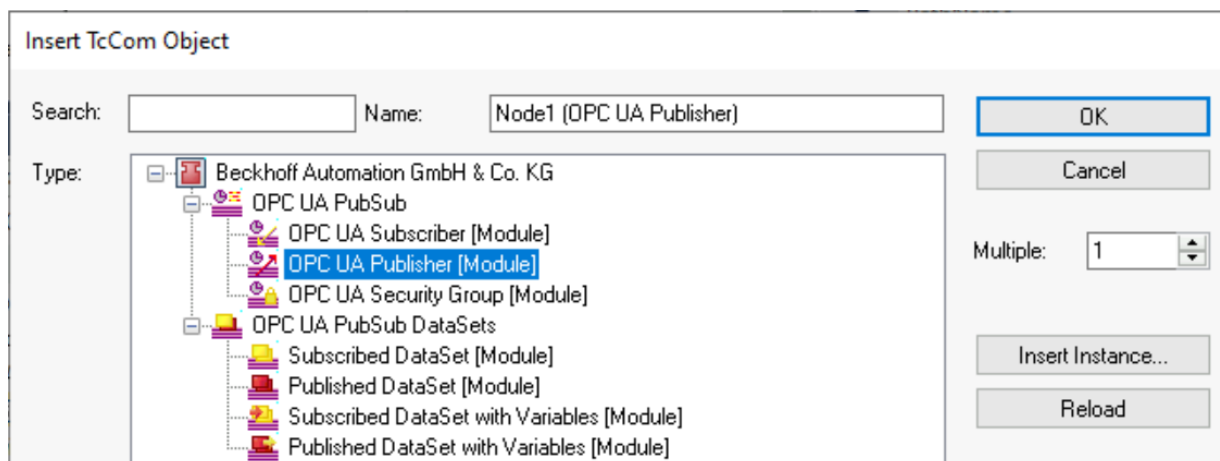


5. 双击已添加的设备，然后转至 **Adapter**（适配器）选项卡。

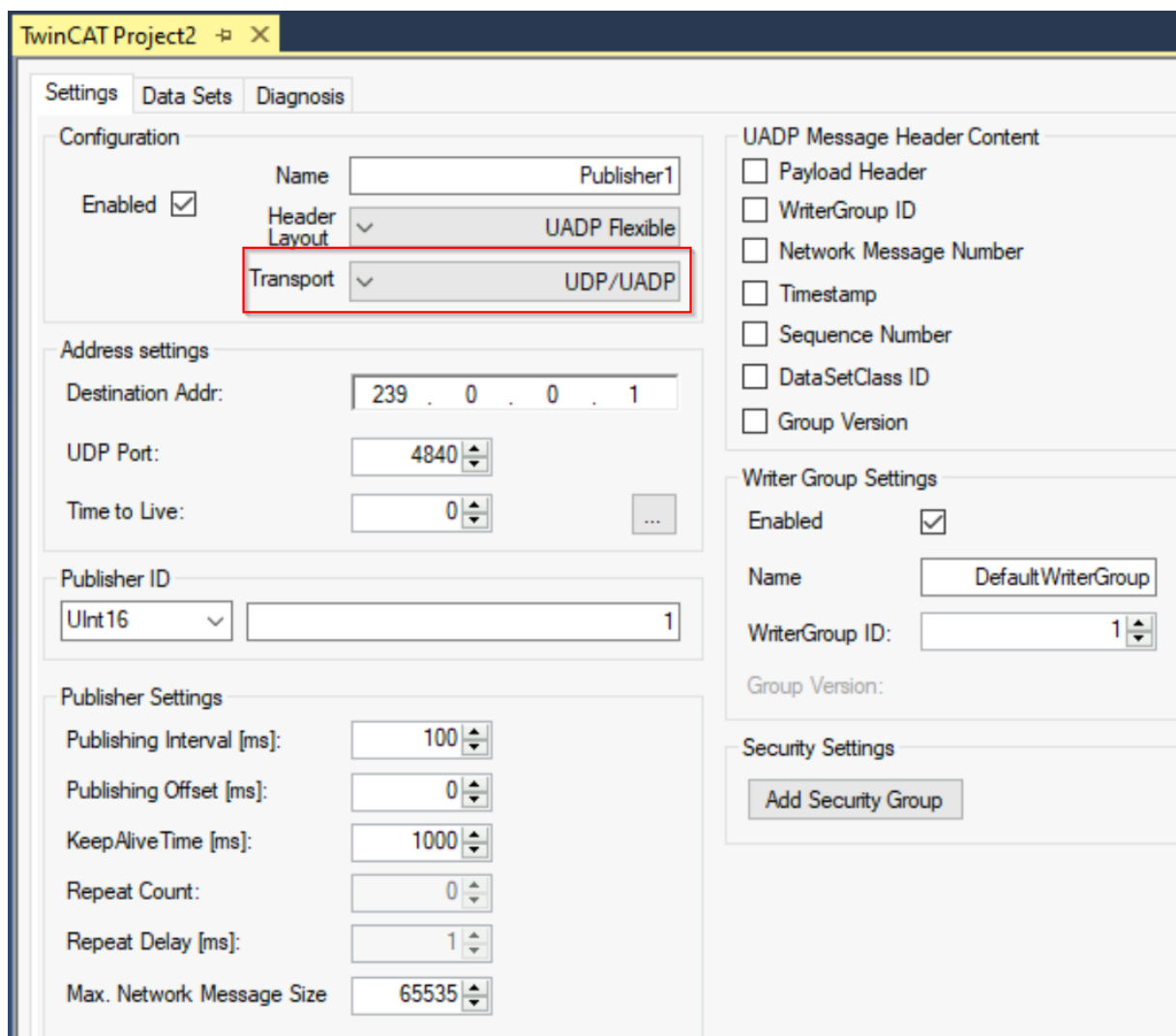


6. 将该设备绑定至您的其中一张网络接口卡。确保已在该 NIC 上安装 TwinCAT-Intel PCI Ethernet Adapter 驱动程序。
7. 右键单击该设备，然后选择 **Add new element...**（添加新元素.....）

8. 选择 **OPC UA Publisher (Module)** (OPC UA 发布者 (模块)) 条目, 然后点击 **OK** (确定)。

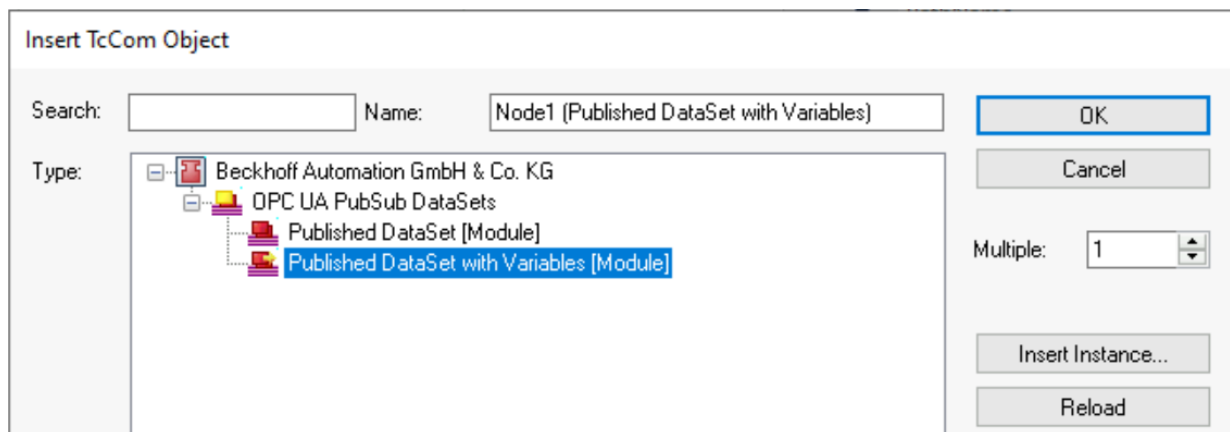


9. 要配置已添加的 **Publisher** (发布者) 节点, 只需在树形视图中双击该节点即可。Publisher (发布者) 端提供许多配置选项, 例如, 要使用的传输协议 (UDP、MQTT)、数据格式 (二进制、JSON)、可选标头字段等。我们希望暂时将这些设置保留默认值。只需确保已选择 **UDP/UADP** 传输协议即可。

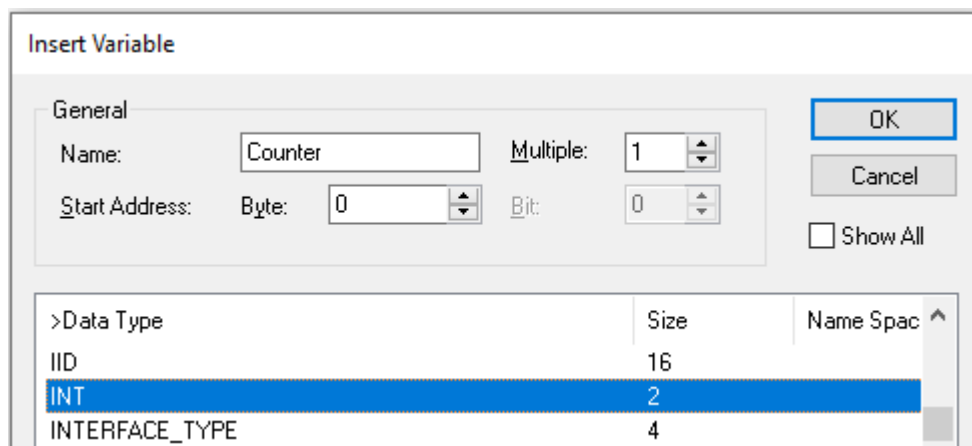


10. 下一步, 您希望配置所谓的 Dataset (数据集), 以便定义 Publisher (发布者) 应发送的变量。右键单击 **Publisher** (发布者) 节点, 然后选择 **Add new element....** (添加新元素……)

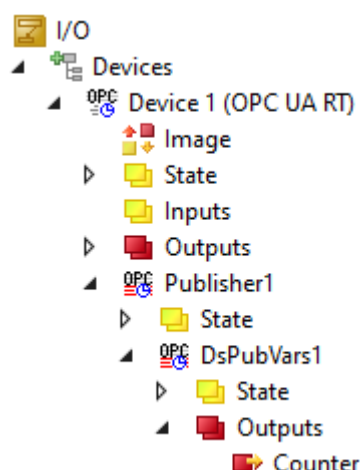
11. 选择 **Published DataSet with Variables (Module)**（带变量的已发布数据集（模块））条目，然后点击 **OK**（确定）。



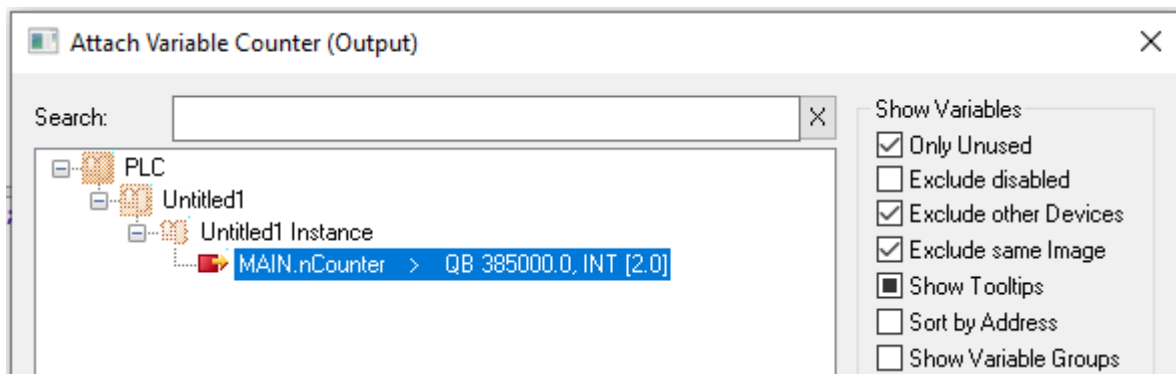
12. 在已添加的数据集上，右键单击 **Outputs**（输出）节点，然后选择 **Add new element...**（添加新元素.....）
13. 为过程变量设置名称，例如 **Counter**（计数器），并将数据类型选为 **INT**。点击 **OK**（确定）可将变量添加到数据集的过程映像。



⇒ 现在，您的配置应如下所示：



14. 右键单击已添加的变量，然后选择 **Change link...**（更改链接.....）。选择您早先创建的 PLC 变量，然后点击 **OK**（确定）。



激活项目

点击 TwinCAT XAE 工具栏上的 **Activate**（激活）按钮，激活 TwinCAT 项目。确保您已选择正确的目标设备（我们假定您正在使用本地设备），并且正在使用正确的实时设置，以便在设备上激活 TwinCAT Runtime。有关如何激活 TwinCAT 配置以及将 TwinCAT 切换至 Run mode（运行模式）的更多信息，请参阅常规 TwinCAT 文档。

在本快速入门教程中，我们使用了 UDP 作为传输协议。现在，您可以将 OPC UA Pub/Sub 订阅者连接到数据，也可以使用 Wireshark® 查看数据。如果您想连接订阅者，我们强烈建议您使用 [配置导入/导出 \[► 38\]](#)。

4.1.2 MQTT

本快速入门教程演示了如何设置 PLC 项目、OPC UA RT 设备和发布者，该发布者通过 MQTT 发送包含一个变量的数据集。该变量通过过程映像与 PLC 项目中的变量相关联。

创建 PLC 项目

创建 PLC 项目，其中定义一个数据类型为 INT 的变量，例如：

```
PROGRAM MAIN
VAR
  nCounter : INT;
END_VAR
nCounter := nCounter + 1;
```

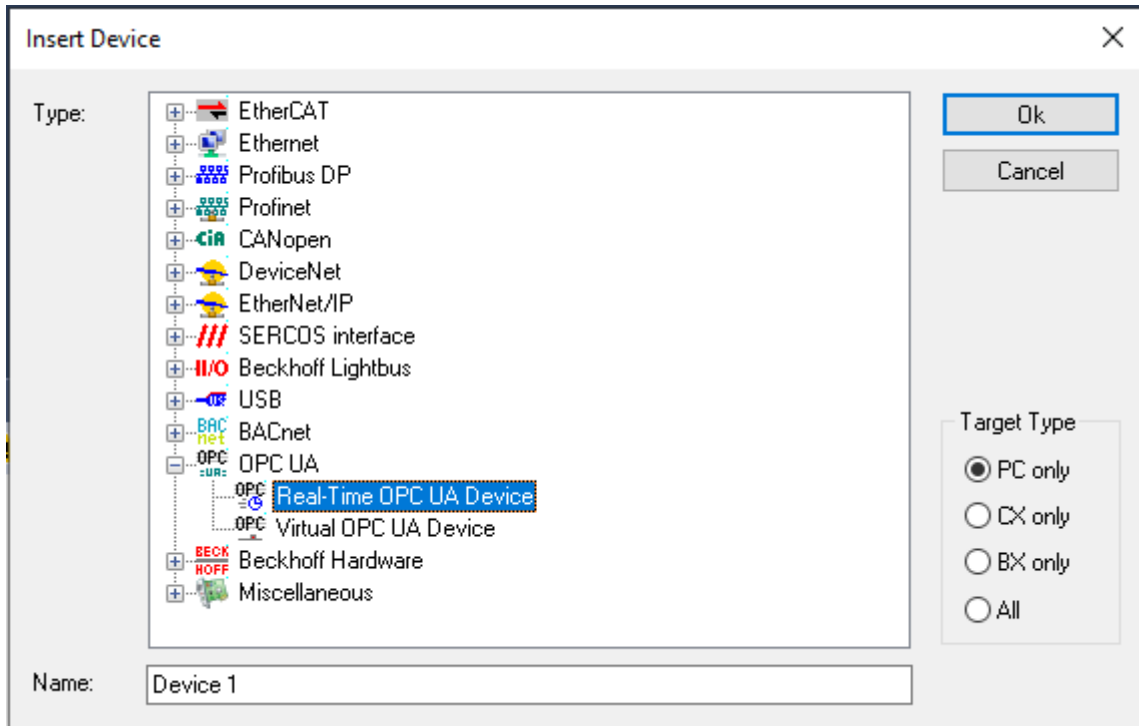
请注意，您无需设置任何输入/输出编译指令，因为我们不会使用过程映像来关联变量。激活该配置，启动 PLC 程序。

配置发布者

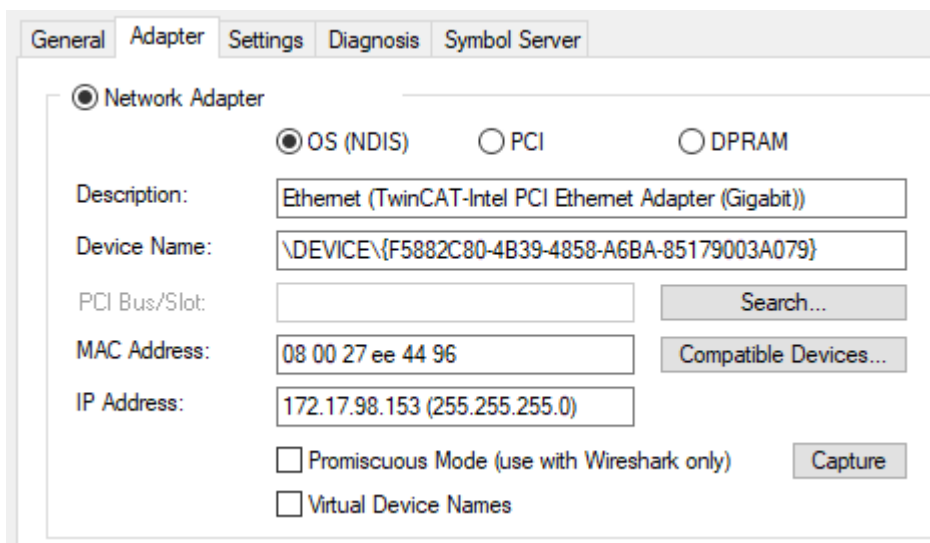
请按照以下配置步骤设置 OPC UA Publisher 设备：

1. 确保 TwinCAT 处于配置模式。
2. 在 Solution Explorer（解决方案资源管理器）中，展开 I/O 节点，然后选择其子项 **Devices**（设备）。
3. 右键单击 **Devices**（设备），然后选择 **Add new element...**（添加新元素.....）

4. 展开 **OPC UA** 类别，选择 **Real-Time OPC UA Device**（实时 OPC UA 设备），然后点击 **OK**（确定）。

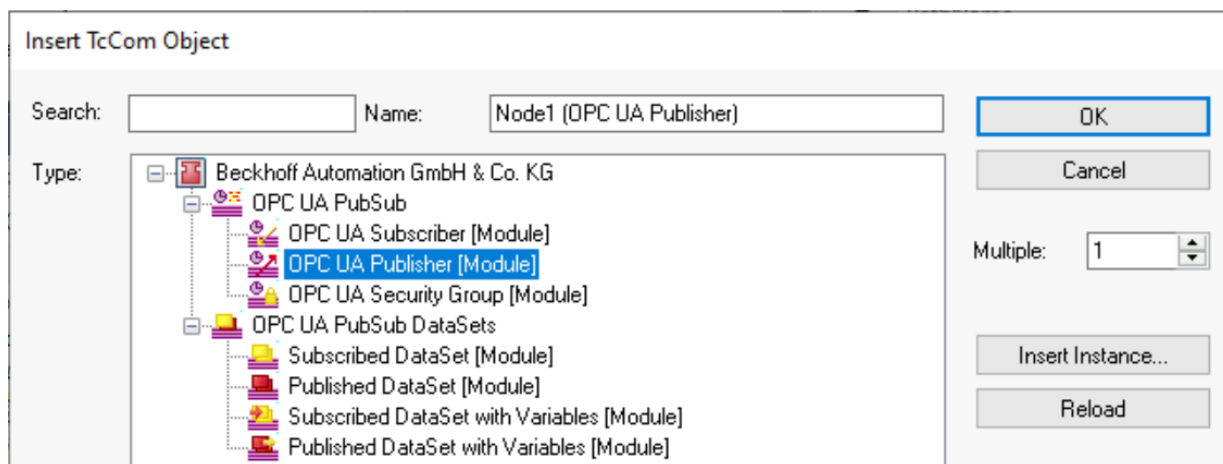


5. 双击已添加的设备，然后转至 **Adapter**（适配器）选项卡。

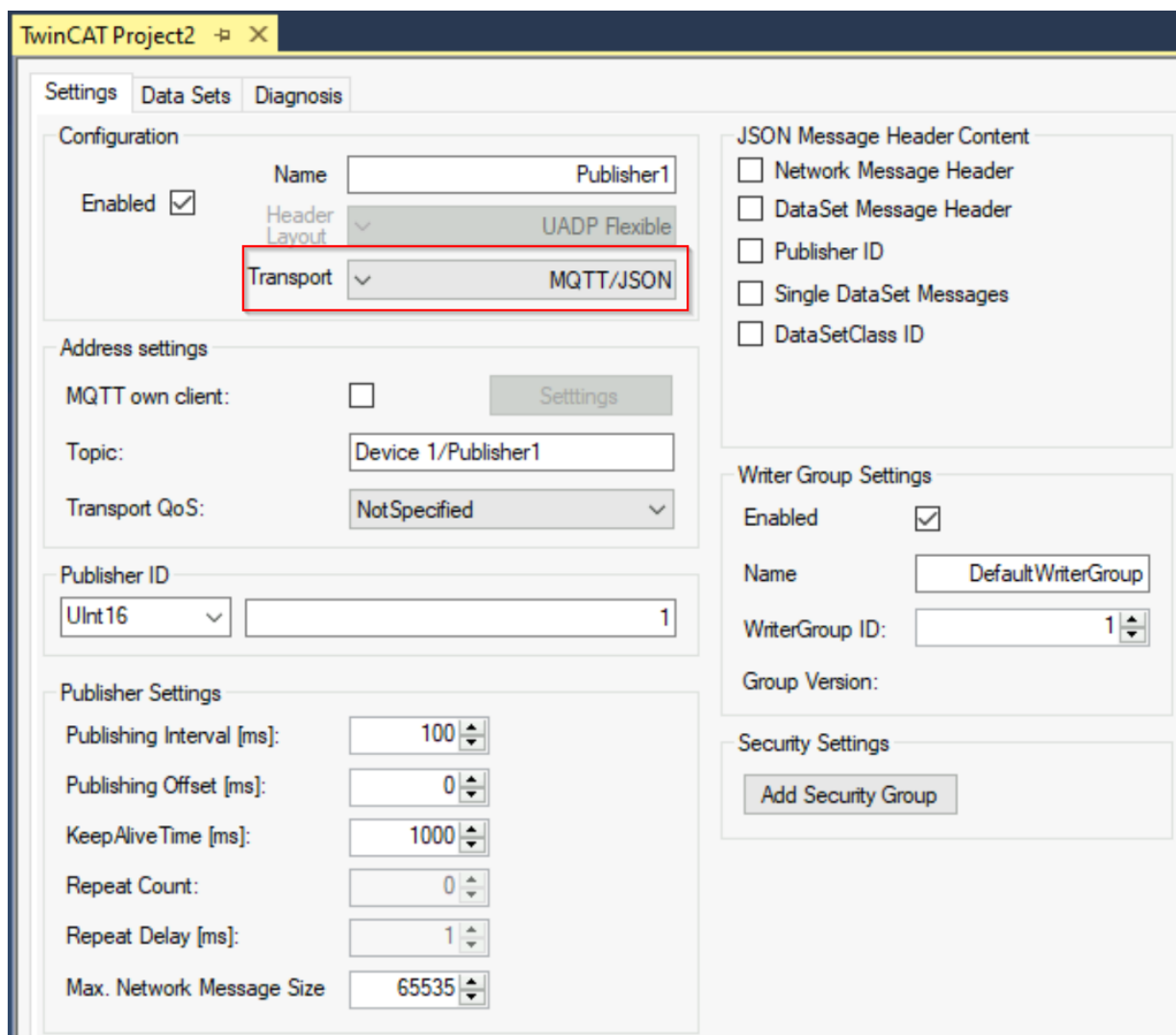


6. 将该设备绑定至您的其中一张网络接口卡。确保已在该 NIC 上安装 TwinCAT-Intel PCI Ethernet Adapter 驱动程序。
7. 右键单击该设备，然后选择 **Add new element...**（添加新元素.....）

8. 选择 **OPC UA Publisher (Module)** (OPC UA 发布者 (模块)) 条目, 然后点击 **OK** (确定)。

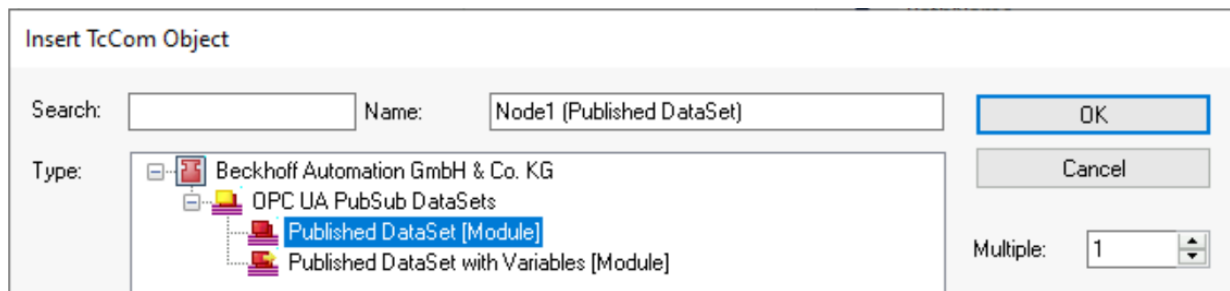


9. 要配置已添加的 **Publisher** (发布者) 节点, 只需在树形视图中双击该节点即可。Publisher (发布者) 端提供许多配置选项, 例如, 要使用的传输协议 (UDP、MQTT)、数据格式 (二进制、JSON)、可选标头字段等。我们希望暂时将这些设置保留为默认值, 仅将传输设置更改为 **MQTT/JSON**。

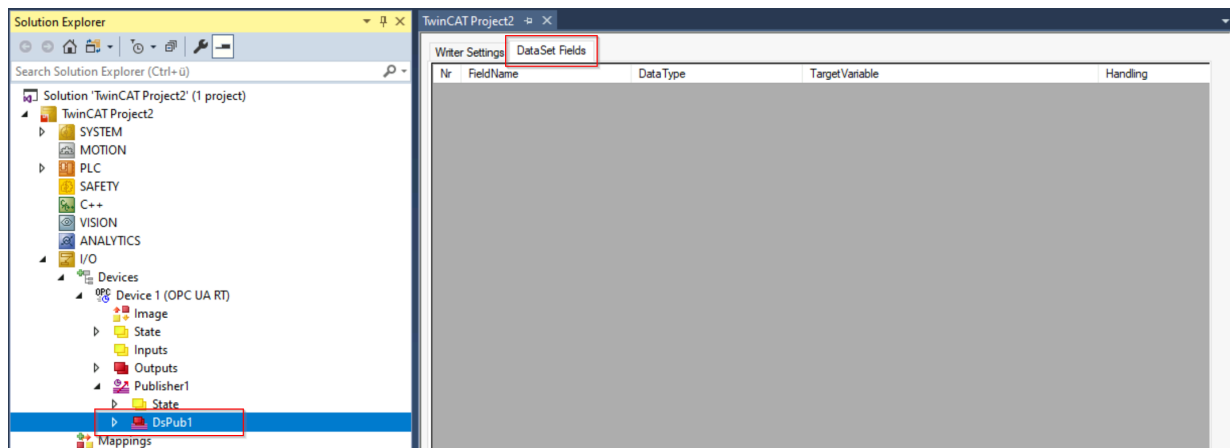


10. 下一步, 您希望配置所谓的 Dataset (数据集), 以便定义 Publisher (发布者) 应发送的变量。右键单击 **Publisher** (发布者) 节点, 然后选择 **Add new element....** (添加新元素……)

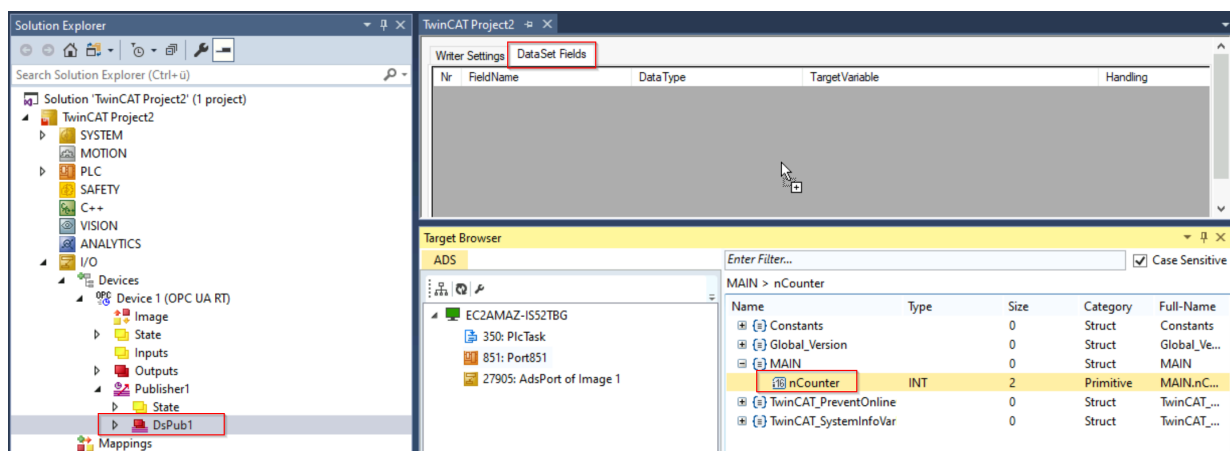
11. 选择 **Published DataSet (Module)**（已发布数据集（模块））条目，然后点击 **OK**（确定）。



12. 双击该数据集，打开 **Data Set fields**（数据集字段）选项卡。



13. 打开 TwinCAT Target Browser，转至 PLC 项目的 MAIN 程序，并将变量 **nCounter** 拖放到数据集字段列表中。



激活项目

点击 TwinCAT XAE 工具栏上的 **Activate**（激活）按钮，激活 TwinCAT 项目。确保您已选择正确的目标设备（我们假定您正在使用本地设备），并且正在使用正确的实时设置，以便在设备上激活 TwinCAT Runtime。有关如何激活 TwinCAT 配置以及将 TwinCAT 切换至 Run mode（运行模式）的更多信息，请参阅常规 TwinCAT 文档。

在本快速入门教程中，我们使用 MQTT 作为传输协议。现在，您可以使用任何能够访问消息代理的 MQTT 客户端来订阅已发布的数据集。下方截图展示了已订阅该数据的命令行工具 **mosquitto_sub.exe**。该工具是 Mosquitto 软件应用的一部分。

```
Administrator: PowerShell 7 (x64)
PS C:\Program Files\mosquitto> .\mosquitto_sub.exe -t "Device 1/Publisher1"
{
  "MessageId": "{00000002-FFFE-FFFF-FEFF-FFFFFFEFFFFFFF}",
  "MessageType": "ua-data",
  "Messages": [
    {
      "Payload": {
        "nCounter": {
          "Type": 4,
          "Body": 0
        }
      }
    }
  ]
}
{
  "MessageId": "{00000002-FFFE-FFFF-FEFF-FFFFFFEFFFFFFF}",
  "MessageType": "ua-data",
  "Messages": [
    {
      "Payload": {
        "nCounter": {
          "Type": 4,
          "Body": 0
        }
      }
    }
  ]
}
```

4.2 支持的功能

OPC UA Pub/Sub 具备丰富的功能。本产品初始阶段可能无法支持所有功能。随着时间的推移，我们将通过产品更新逐步增加更多功能。下表列出了目前支持的所有功能。

- **与 OPC UA Pub/Sub 规范兼容**
i 本产品的当前版本与第 1.05.03 版 OPC UA Pub/Sub 规范兼容。

报头布局

名称	支持
灵活性高	是
UADP Dynamic	是
UADP PeriodicFixed	是
JSON DataSetMessage	是
JSON Minimal	是
JSON NetworkMessage	是

UDP 传输

功能	支持
发布者	是
订阅者	是

功能	支持
UADP 分块	是
IP 数据分片	是
DataKey-Frames	是
Delta-Frames	是
KeepAlive	是
消息超时处理	是
UADP 原始编码	是
UADP 变体编码	是
Pub/Sub 安全消息签名	尚不支持
Pub/Sub 安全消息加密	尚不支持
支持原始数据类型（INT、REAL 等）	是
支持原始数据类型（INT、REAL 等）的数组	是
支持数据结构	是
支持数据结构的数组	是
支持联合体	尚不支持
支持变体	尚不支持
支持本地数据集	是
支持预配置（全局）数据集	是
支持单播/多播	是
支持配置 UDP 端口	是
支持事件消息	尚不支持
支持数据集中的字段特定数据 （Delta-Deadband、Deadband-Type、Substitute Value）	尚不支持
支持以太网 VLAN 传输	尚不支持
支持元数据服务	尚不支持

MQTT 传输

功能	支持
发布者	是
订阅者	是
Key-Frames	是
Delta-Frames UADP	是
Delta-Frames JSON	尚不支持
KeepAlive	尚不支持
支持 MQTT 3.1.1 版本	是
支持 MQTT 5.0 版本	是
支持 MQTT/TLS 1.2 和 1.3 版本	是
JSON 编码（可逆）	是
JSON 编码（不可逆）	是
UADP 编码	是
支持事件消息	尚不支持
支持原始数据类型（INT、REAL 等）	是
支持原始数据类型（INT、REAL 等）的数组	是
支持数据结构	是
支持数据结构的数组	是

功能	支持
支持联合体	尚不支持
支持变体	尚不支持
支持本地数据集	是
支持全局数据集	是
支持元数据服务	尚不支持

支持的消息代理

我们的 TwinCAT MQTT 协议驱动程序支持 MQTT 3.1.1 和 5.0 版本。因此，支持所有同样支持这些版本的消息代理。我们已成功测试与 Mosquitto、HiveMQ 和 AWS IoT Core 之间的通信。

OPC UA Pub/Sub 安全

功能	支持
预共享密钥 (PSK)	尚不支持
本地 SKS	尚不支持
远程 SKS	尚不支持

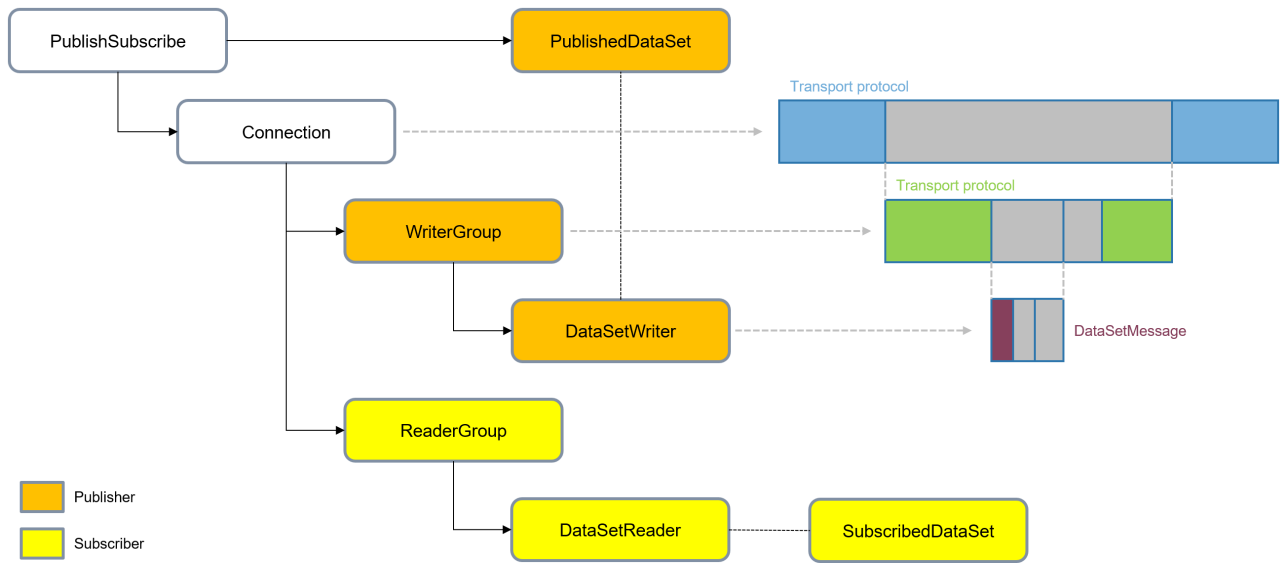
配置处理

功能	支持
离线配置交换 (UA-Binary)	是
在线配置交换 (读取)	尚不支持
在线配置交换 (写入)	尚不支持

4.3 协议概述

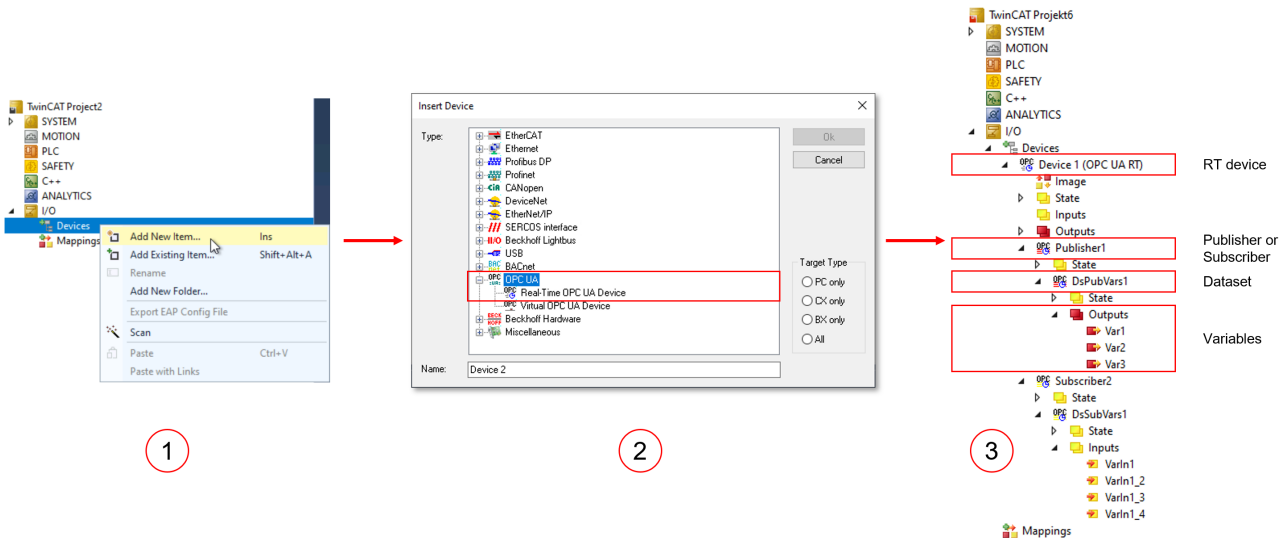
OPC UA Pub/Sub 为各种组件定义了不同的配置参数。它们定义了发布者和订阅者的行为。可通过 PubSub 的 OPC UA 信息模型或供应商特定机构配置这些参数，对于 TF6105 TC3 OPC UA Pub/Sub 产品而言，即 TwinCAT XAE 环境。

下图展示了不同的配置组件及其相互关系。WriterGroup、DataSetWriter 和 PublishedDataSet 组件定义了发布者，而 ReaderGroup、DataSetReader 和 SubscribedDataSet 则定义了订阅者。



如需了解各个组件的详细信息，建议查阅 OPC UA 规范第 14 部分 (Pub/Sub)。

在使用 TwinCAT XAE 环境进行设置配置时，工程工作流程可如下所示：



步骤	描述
1	TwinCAT I/O 配置中添加了新设备。
2	在设备列表中，您可以选择“Real-Time OPC UA Device”（实时 OPC UA 设备），该设备将作为所有 OPC UA Pub/Sub 设置的入口。
3	添加了发布者、订阅者、数据集和变量，及其对应的参数。

以下各章节对各个组件进行了概述，并提供了更详细的文档文章链接。

设备

OPC UA RT (Realtime) Device（OPC UA RT（实时）设备）是 OPC UA Pub/Sub 配置的入口，并将通过基于 TwinCAT Realtime-Ethernet-Adapter 驱动程序的网络接口卡进行关联。该设备可能包含一个或多个发布者和/或订阅者，以及（全局）数据集。

发布者/订阅者

发布者或订阅者节点用于定义所连接的数据集是发送（发布）还是接收（订阅）。每个发布者/订阅者 [► 27] 节点都包含应使用的传输协议 [► 33] 的地址信息。

数据集

数据集可以在 2 个不同的范围中进行指定。您既可以在发布者或订阅者上对数据集进行本地定义，也可以在设备上进行全局定义。全局数据集可由多个发布者或订阅者共享。有 2 种不同类型的数据集，其定义的是变量的创建和处理方式：

- 带变量的数据集：该数据集类型可配置为包含出现在数据集过程映像中的变量。然后，这些变量可以与其他过程映像变量建立关联。
- 不带变量的数据集：该数据集类型可通过 TwinCAT Target Browser 配置变量。这些变量随后会在内部自动建立关联，且不会显示在过程映像上。

如需了解更多信息，请查阅我们关于数据集 [► 28] 的文档文章。

变量

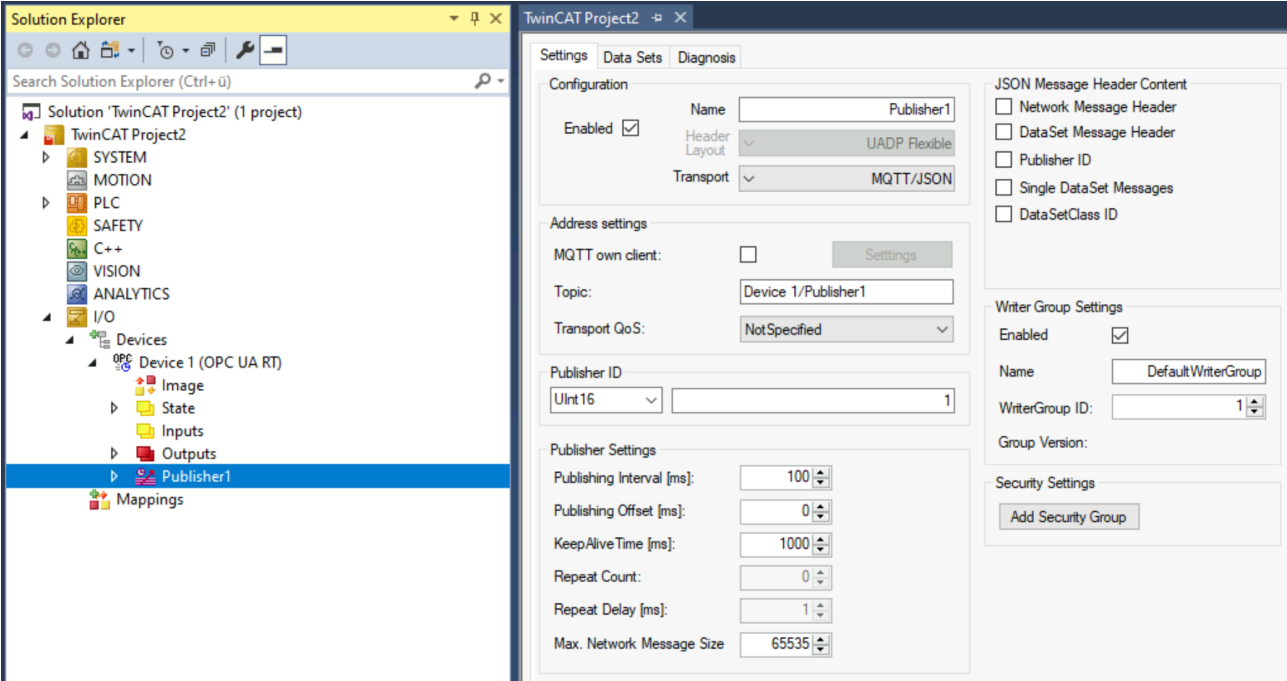
变量以所谓的“数据集字段”的形式添加到数据集中。它们既可以与其他过程映像变量建立关联，也可以通过 TwinCAT Target Browser 直接与其他变量连接。有关更多信息，请参见我们关于数据集 [► 28] 的文档文章。

4.4 发布者/订阅者

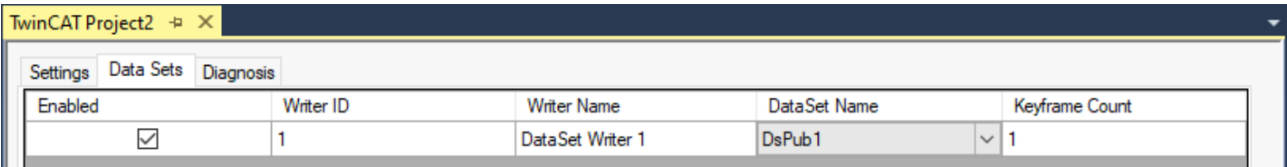
OPC UA Pub/Sub 以发布者/订阅者通信模式为基础。TF6105 TC3 OPC UA Pub/Sub 包含不同的配置机构，用于设置发布者或订阅者及其对应的数据集 [► 28]。我们还建议您查阅我们的协议概述 [► 25] 章节，以了解不同配置组件的概况。

发布者

可以利用发布者来设置用于发送数据的通信。可将其直接添加到 OPC UA RT 设备中。“设置”页面包含用于定义传输协议 [► 33] 的不同配置参数，以及进一步定义数据通信的其他设置。



Data Sets (数据集) 选项卡包含有关所有已关联的本地或全局数据集的信息。



Diagnosis (诊断) 选项卡包含有关数据通信的诊断信息。

TwinCAT Project2

SettingsData SetsDiagnosis

OPC UA Real-Time Device Statistics

	Name	Value
▶	NetworkMessages	(166;0;0;0)
	nSendCnt	166 (10.20 /s)
	nSendFailCnt	0 (0.00 /s)
	nRecvCnt	0 (0.00 /s)
	nRecvFailCnt	0 (0.00 /s)
	+ DataKeyMessages	(166;0;0;0)
	+ DeltaFrameMessages	(0;0;0;0)
	+ EventMessages	(0;0;0;0)
	+ KeepAliveMessages	(0;0;0;0)
	+ MetadataMessages	(0;0;0;0)
	+ EncryptedFrames	(0;0;0;0)
	+ SignedFrames	(0;0;0;0)
	+ DiscoveryRequests	(0;0;0;0)
	+ DiscoveryResponses	(0;0;0;0)
	+ ChunkMessages	(0;0;0;0)
	+ StateInfo	(Operational;Operational)

订阅者

可以利用发布者来设置用于接收数据的通信。可将其直接添加到 OPC UA RT 设备中。“设置”页面包含用于定义传输协议 [▶ 33] 的不同配置参数，以及进一步定义数据通信的其他设置。

Solution Explorer

Search Solution Explorer (Ctrl+u)

Solution 'TwinCAT Project2' (1 project)

- TwinCAT Project2
 - SYSTEM
 - MOTION
 - PLC
 - SAFETY
 - C++
 - VISION
 - ANALYTICS
 - I/O
 - Devices
 - OPC Device 1 (OPC UA RT)
 - Image
 - State
 - Inputs
 - Outputs
 - Subscriber1

Mappings

TwinCAT Project2

SettingsData SetsDiagnosis

Configuration

Enabled☒

NameSubscriber1

TransportUDP/UADP

Reader Group settings

Enabled☒

NameDefaultWriterGroup

Address settings

Multicast Addr:239 . 0 . 0 . 1

UDP Port:4840

Subscriber

Max. Network Message Size1

☐ Check Group Version0

Status Process Data

☐ Timestamp

☐ Pico seconds

☐ Sequence number

与“发布者”类似，**Data Sets（数据集）** 选项卡包含有关所有已关联的本地或全局数据集的信息，而 **Diagnosis（诊断）** 选项卡则包含有关数据通信的诊断信息。

4.5 数据集

可以将数据集添加到发布者和订阅者中，并定义一组名称和值对，用以表示一组变量值。此类变量随后由所谓的“数据集字段”表示。可以将元数据添加至数据集，这些元数据提供了有关数据集结构和内容的补充信息。

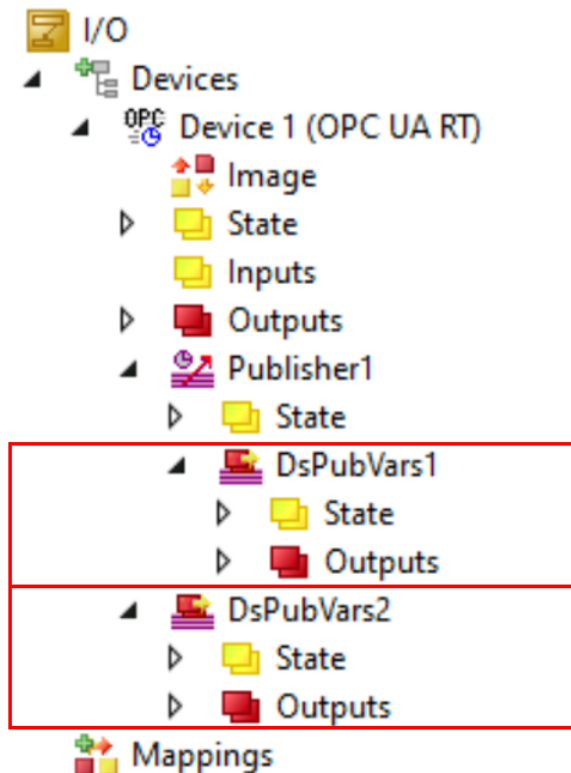
TF6105 TC3 OPC UA Pub/Sub 提供了 2 种不同类型的数据集，这些数据集定义了变量的配置和处理方式。

- 带变量的数据集
- 不带变量的数据集

这些数据集类型会在下文详细说明。

范围

可以在 2 个不同的范围内配置数据集：本地或全局。本地数据集专为单个发布者或订阅者配置，而全局数据集则在设备级别添加，可由发布者或订阅者共享，例如，如果您想在两个不同的发布者上发布相同的数据点。



Local data set on a publisher

Global data set on device level

可以通过 **Data Sets**（数据集）选项卡将全局数据集添加至发布者/订阅者：

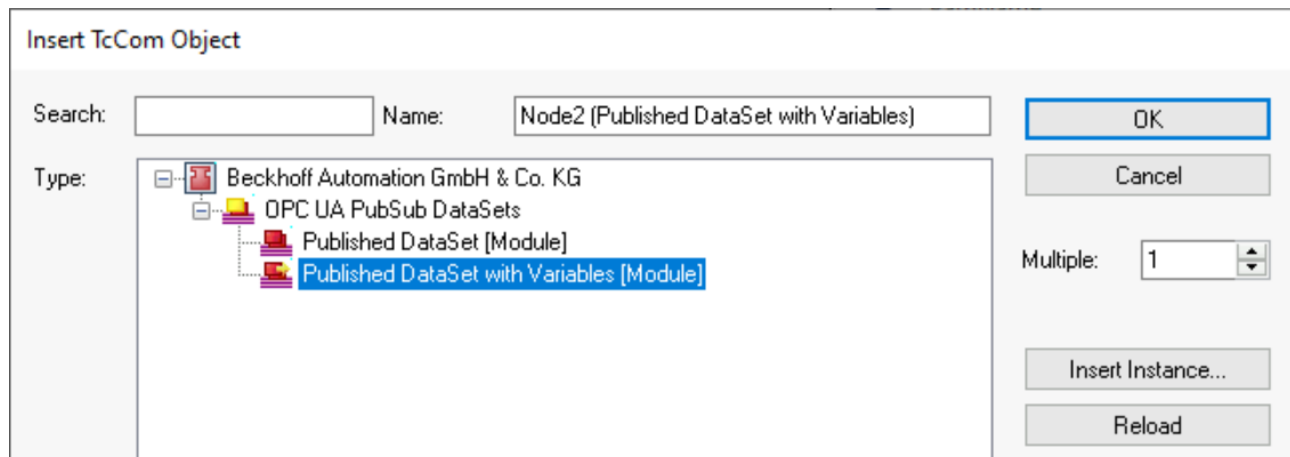
Enabled	Writer ID	Writer Name	DataSet Name	Keyframe Count
☒	1	DataSet Writer 1	DsPubVars1	1
☒	2	DataSet Writer 2	DsPubVars2	1

Add DataSet
Delete DataSet

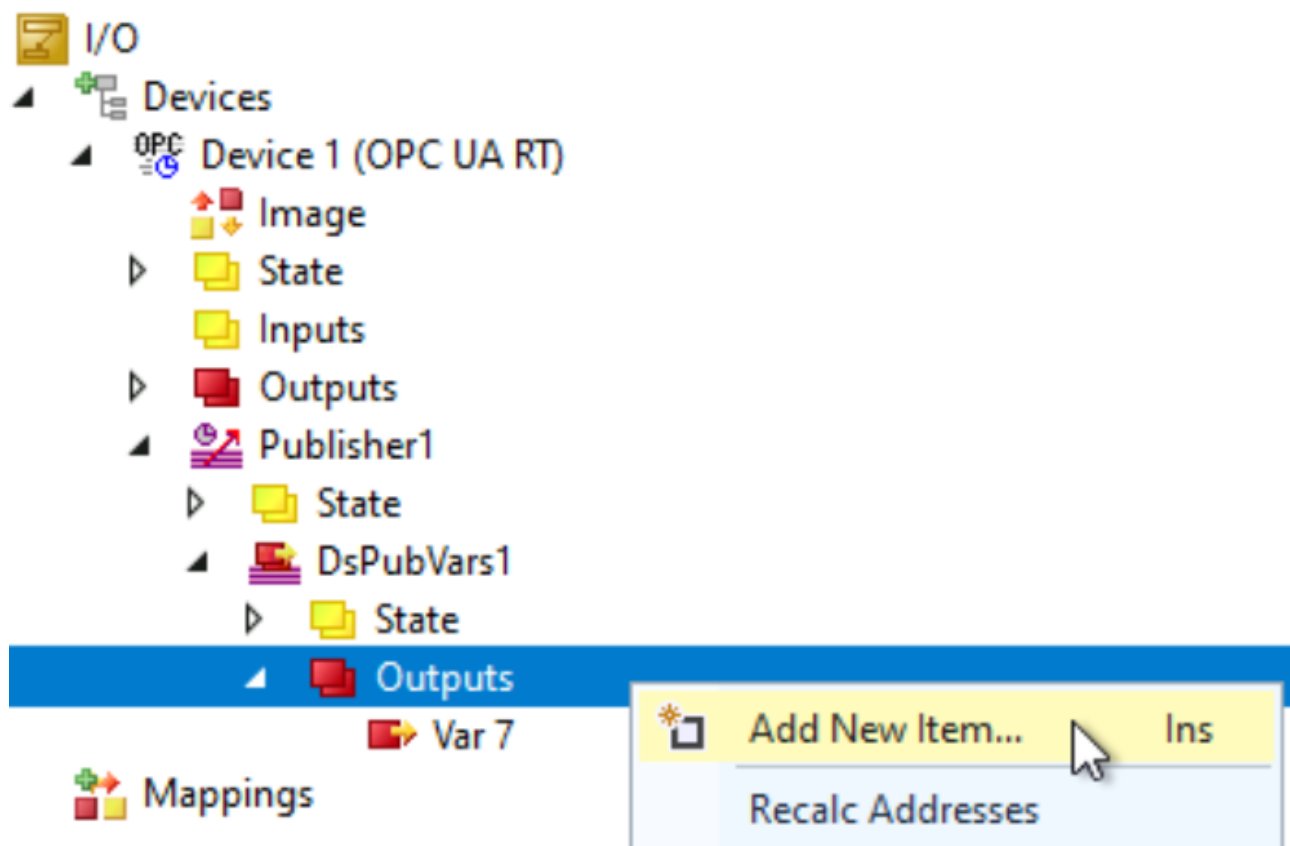
在上例中，已为发布者配置名为“DsPubVars1”的本地数据集和名为“DsPubVars2”的全局数据集。

带变量的数据集

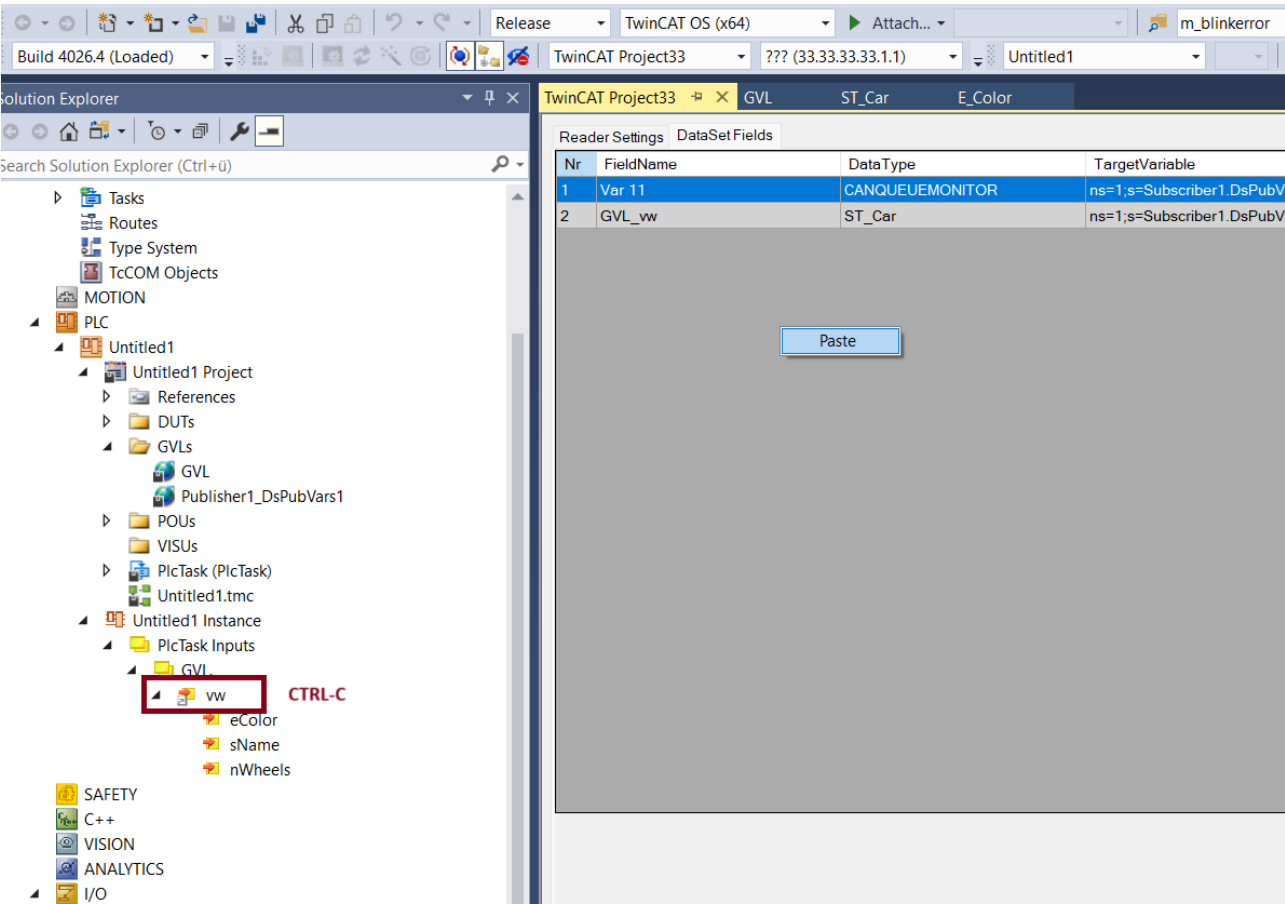
带变量的数据集会将所有变量作为过程映像的一部分进行配置。这些数据集可以进行全局或本地添加，例如在发布者上：



然后，可以将这些变量直接添加到数据集的过程映像上：



此外，您还可以通过 Copy&Paste（复制并粘贴）操作，将 PLC 过程映像中的变量添加到数据集，如下方截图所示。只需按下 STRG+C 将变量复制到剪贴板，然后通过上下文菜单将其粘贴到目标数据集即可。



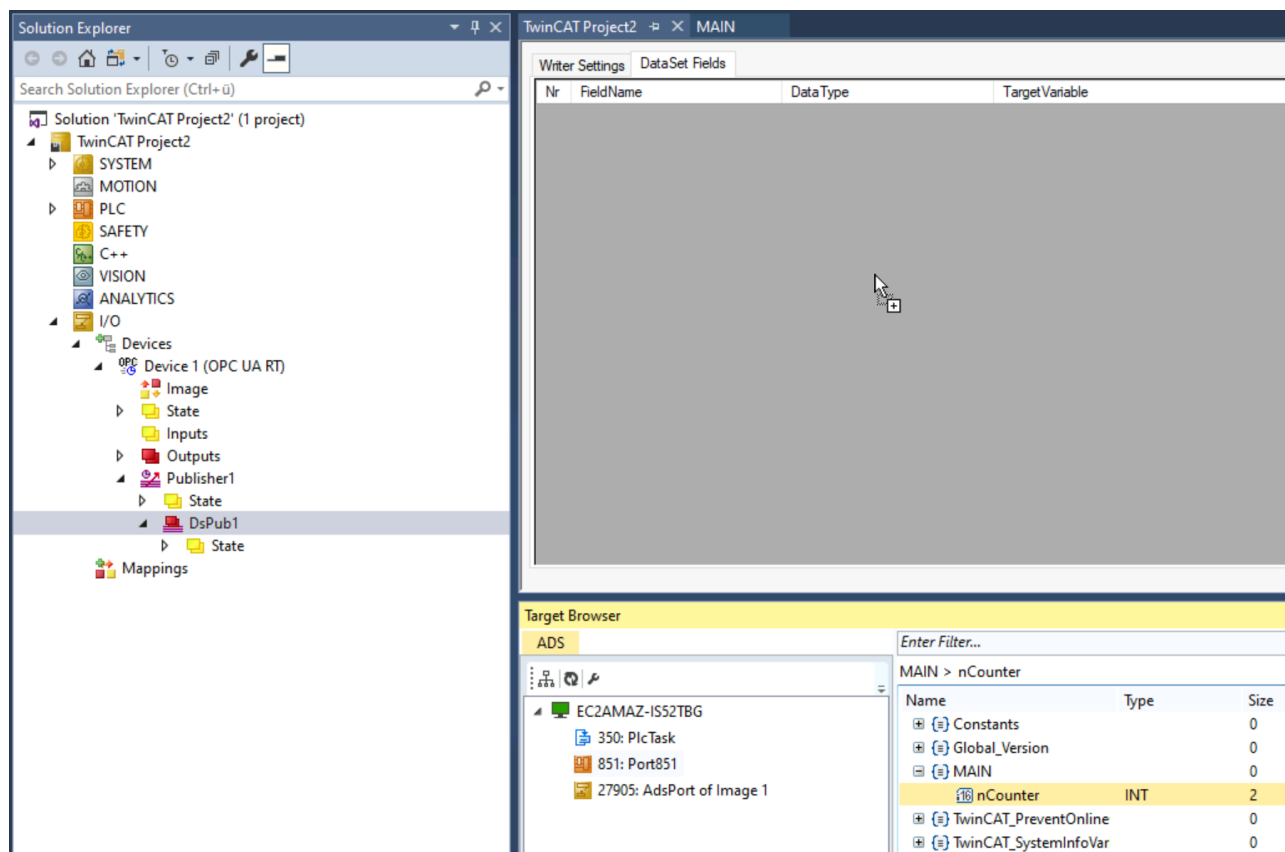
不带变量的数据集

不带变量的数据集不会将这些变量作为过程映像的一部分进行配置。相反，可以通过 TwinCAT Target Browser 或直接从 PLC 过程映像向数据集添加变量。这些变量随后会自动与相应的符号建立关联。

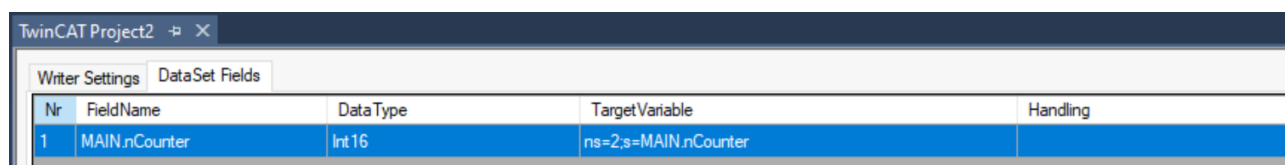
● 符号变更

请注意，如果相关联的符号发生变化，则需要先从数据集中删除这些变量，然后通过 TwinCAT Target Browser 重新添加，因为符号信息是在 Drag&Drop（拖放）过程中获取的。

若要向此类数据集添加变量，请打开 TwinCAT Target Browser，并将变量拖放到该数据集的 **Data Set Fields**（数据集字段）选项卡中，如下方截图所示。

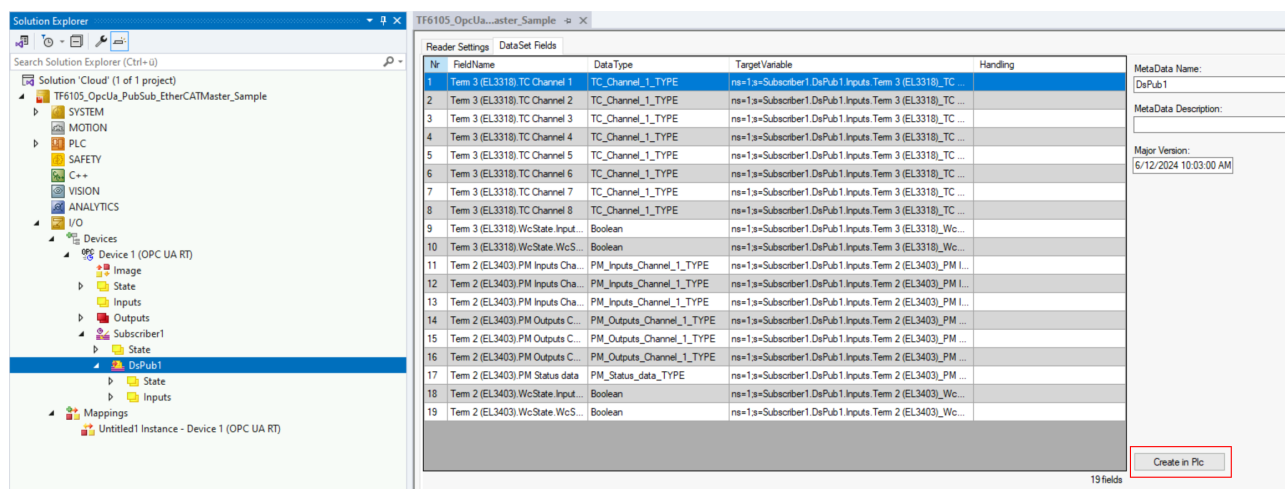


该数据集的字段配置现已包含该变量，配置一经激活，即会发布该变量。

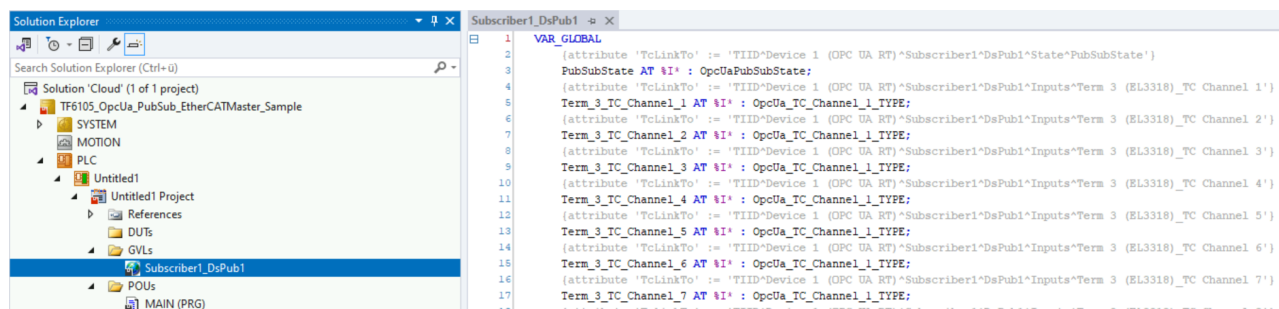


代码生成

订阅者数据集包含自动代码生成功能，可减少创建和链接匹配 PLC 变量所需的工程时间。可通过订阅者数据集上的相应按钮访问代码生成功能。



该功能会生成全局变量列表，其中包含订阅者数据集中的所有变量，这些变量不仅数据类型匹配，还具有“TcLinkTo”属性，该属性会自动将它们与其对应项建立关联。



4.6 传输协议

OPC UA Pub/Sub 扩展软件定义了不同的可用于传输数据的传输协议。这些传输协议包括：

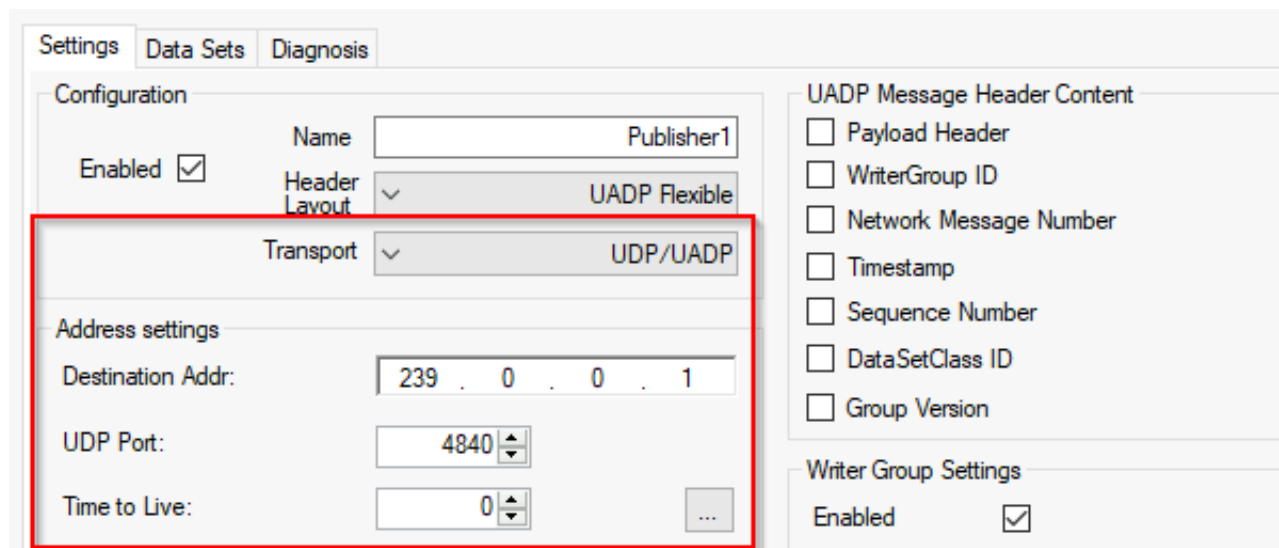
- 适用于高速机器对机器通信的 UDP
- 用于机器到云端通信的 MQTT

每种协议都有其优缺点，具体应选用哪种传输协议，取决于应用项目本身。尽管 MQTT 被视为“IoT”协议，且常与云服务配合使用，但它并不局限于与云相关的应用场景，因为 MQTT 通信连接也可以在私有网络内建立。

TF6105 TC3 OPC UA Pub/Sub 产品同时支持这两种传输协议。以下各章将介绍每种传输协议的具体配置参数。

4.6.1 UDP

要为“发布者”或“订阅者”配置 UDP 传输协议，只需打“发布者/订阅者”配置页面，然后选择 UDP 传输协议即可。UDP 支持（二进制）UADP 编码以及不同的报头布局[► 36]。提供不同地址设置选项，用于配置连接详细信息，例如目标地址、UDP 端口和生存时间。

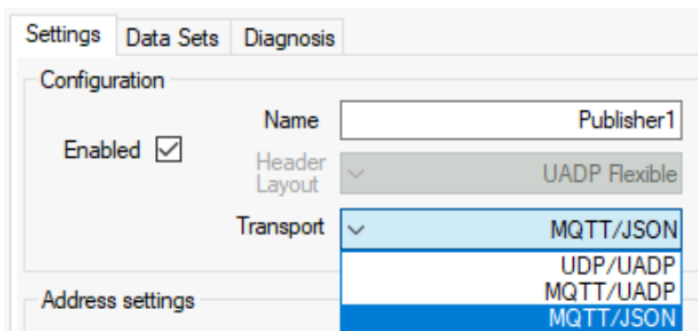


4.6.2 MQTT

要配置“发布者”或“订阅者”以使用 MQTT 传输协议，只需打开“发布者/订阅者”配置页面，然后选择 MQTT 传输协议即可。MQTT 支持（二进制）UADP 编码以及 JSON，并可配置为使用指定的报头布局[► 36]。

Encoding (编码)

有两种编码可供选择：MQTT/UADP 和 MQTT/JSON。第一种数据格式使用二进制 UADP 编码对消息进行编码，而最后一种则使用 JSON 格式。



根据所选报头布局，MQTT/JSON 编码可能如下所示：

```
{
  "MessageId": "{0CF7F27A-56FF-1F85-CFC2-12C886365C1D}",
  "MessageType": "ua-data",
  "Messages": [
    {
      "DataSetWriterId": 1,
      "Payload": {
        "nCounter": {
          "Type": 4,
          "Body": 18
        },
        "bToggle": {
          "Type": 1,
          "Body": false
        }
      }
    },
    {
      "DataSetWriterId": 2,
      "Payload": {
        "myNewVariable": {
          "Type": 4,
          "Body": 42
        }
      }
    }
  ]
}
```

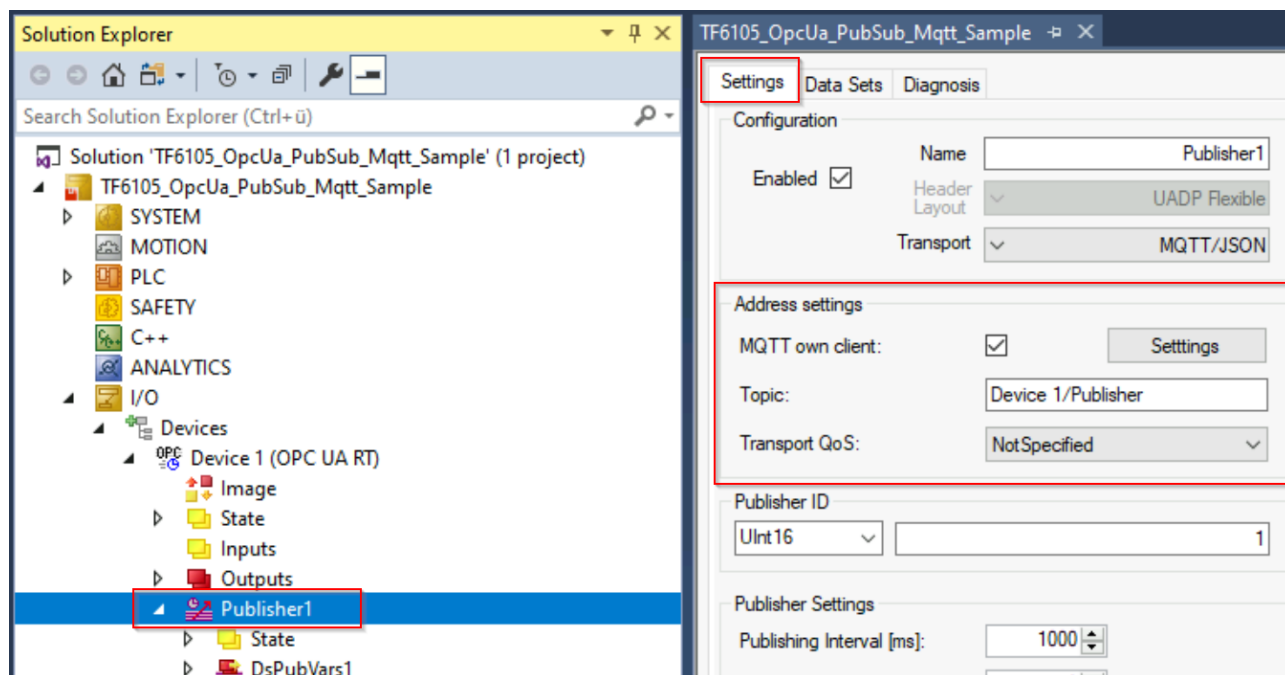
JSON 格式已在 OPC UA Pub/Sub 规范中进行标准化配置。有关 JSON 方案及各个字段的完整说明，请参阅规范文档。

连接参数

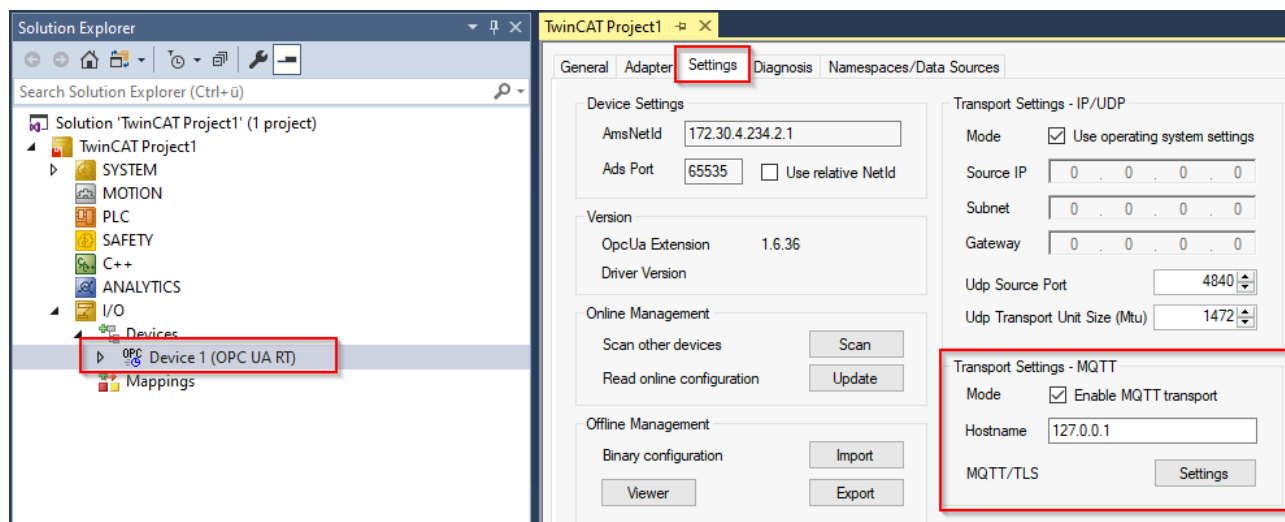
每当使用 MQTT 作为传输协议时，您还需要定义与 MQTT 消息代理的连接参数。根据具体用例，可以在以下两个不同位置指定这些参数：

- 您可以在每个“发布者”或“订阅者”上分别配置 MQTT 连接
- 或者，您可以在 OPC UA RT 设备设置中配置一个“全局”MQTT 连接。在这种情况下，所有“发布者”和“订阅者”都共用一条与 MQTT 消息代理的连接。

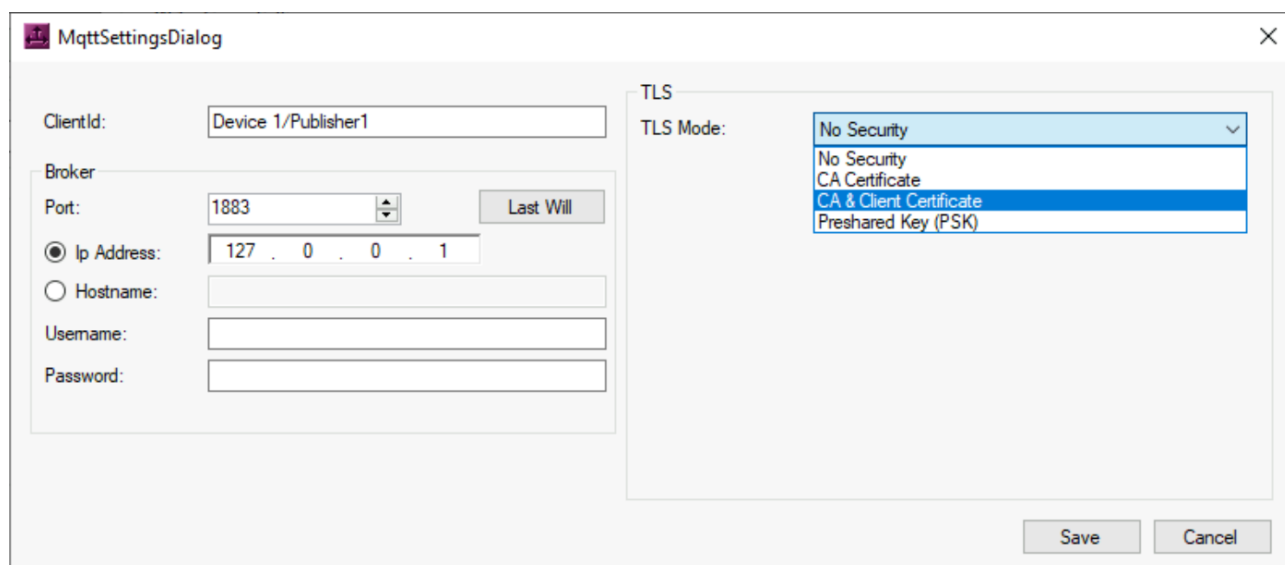
下方截图显示了“发布者”上的 MQTT 连接参数：



下方截图显示了在 OPC UA RT 设备上设置全局 MQTT 连接的位置：



在这两种情况下，用于输入连接参数的对话框都是相同的，如下所示：



主题

MQTT 消息代理上的数据按所谓的“主题”进行分类管理。从 OPC UA Pub/Sub 的角度来看，您可以在 2 个层面上指定主题：

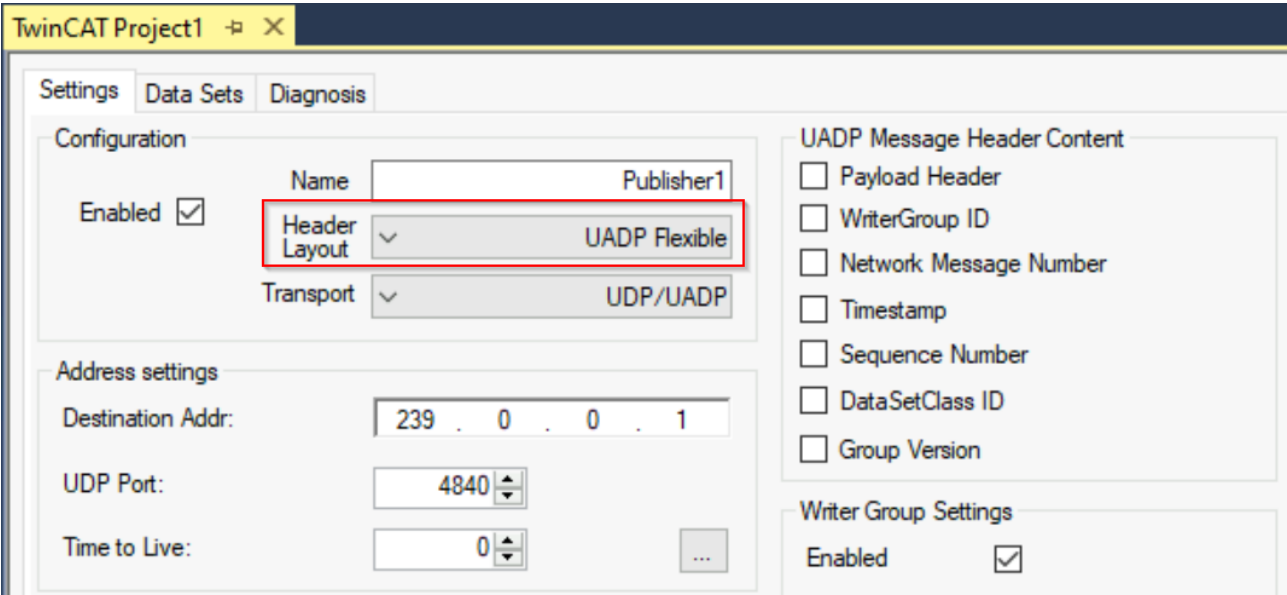
- “发布者”的主题是在“发布者”设置”上配置的。
- 每个数据集都会针对“订阅者”主题进行单独配置。

4.7 报头布局

报头内容和消息布局的设计具有灵活性，可通过启用或禁用报头中的各个字段来支持不同的用例。虽然这种灵活性使得 Pub/Sub 能够支持许多不同的用例，但可用的报头字段组合数量的增加，也使得实现和验证所需的工作量随之增加。另一方面，在特定的应用领域内或针对不同的用例，某些配置可能比其他配置更合适。

可以使用自定义报头布局配置（也称为“灵活”报头布局），但应仅限于不属于这些布局适用场景的应用。

如下方截图所示，可以在“发布者”或“订阅者”上配置报头布局。



支持以下报头布局。报头布局名称源自 OPC UA 规范第 14 部分附录 A。

名称	支持
灵活性高	是
UADP Dynamic	是
UADP PeriodicFixed	是
JSON DataSetMessage	是
JSON Minimal	是
JSON NetworkMessage	是

4.8 关键帧和增量帧

在 TwinCAT 3 OPC UA Pub/Sub 中，数据记录既可作为关键帧传输，也可作为增量帧传输。这些机构旨在优化网络带宽，并确保过程数据的高效传输。

关键帧

关键帧始终包含 DataSetMessage 中所有已配置字段的完整数据记录。接收端可以利用关键帧重建所有传输变量的完整状态。

关键帧的属性：

- 始终包含 DataSet 中的所有字段。
- 启用完整状态的同步功能。
- 这些数据包会定期发送，以确保订阅者即使在发生数据包丢失后，也能收到一致性状态信息。

因此，关键帧构成后续增量帧的参考基准。

增量帧

增量帧仅包含自上次发送帧以来内容发生变化的字段的值。从而可以减小传输消息的大小，特别是当 DataSet 中的变量变化很少时。

增量帧的属性：

- 仅传输已更改的字段。
- 减轻网络负载并降低带宽利用率。
- 只有在事先已接收有效关键帧的情况下，才能进行解析。

因此，订阅者必须先接收关键帧，才能正确应用增量帧。

关键帧与增量帧的交互作用

当增量帧功能激活时，发布者通常会发送一系列增量帧，其中会定期穿插关键帧。

这样可以确保：

- 丢失的数据包可以得到补偿。
- 订阅者可以随时重新进行同步。

关键帧的频率由“KeyFrameCount”参数决定。

对于 UADP 而言，关键帧与增量帧之间的交互具有特殊规则：如果数据记录的所有字段都发生了变化，则必须发送关键帧。即使增量帧比关键帧大（可能由于编码原因造成），也必须发送关键帧。

参数：KeyFrameCount

KeyFrameCount 参数用于定义在发送了多少条消息后再次传输关键帧。

示例：

```
KeyFrameCount = 3
```

在这种情况下：

1. 已发送关键帧。
2. 其后最多附带 2 个增量帧。
3. 然后再次设置关键帧。

因此，该序列如下所示：

```
Key, Delta, Delta, Key, Delta, ...
```

“KeyFrameCount”的数值越小，意味着完整数据集的传输频率越高，这虽然提高了鲁棒性，但也增加了网络负载。

参数：KeepAliveInterval

KeepAliveInterval 参数可确保即使 DataSet 中的值未发生任何变化，消息仍会发送。

在使用增量帧时，这一点尤为重要，否则在没有变化时将不会发送任何消息。

当达到 KeepAlive 间隔时，发布者会发送一条消息，即使字段没有发生变化也是如此。本消息的目的是：

- 以保持连接处于活动状态。
- 以向订阅者表明发布者仍处于活跃状态。

- 以避免超时或启动 Watchdog（看门狗）机制。

根据具体实现情况，该消息可以作为关键帧传输，也可以作为无场变化帧传输。

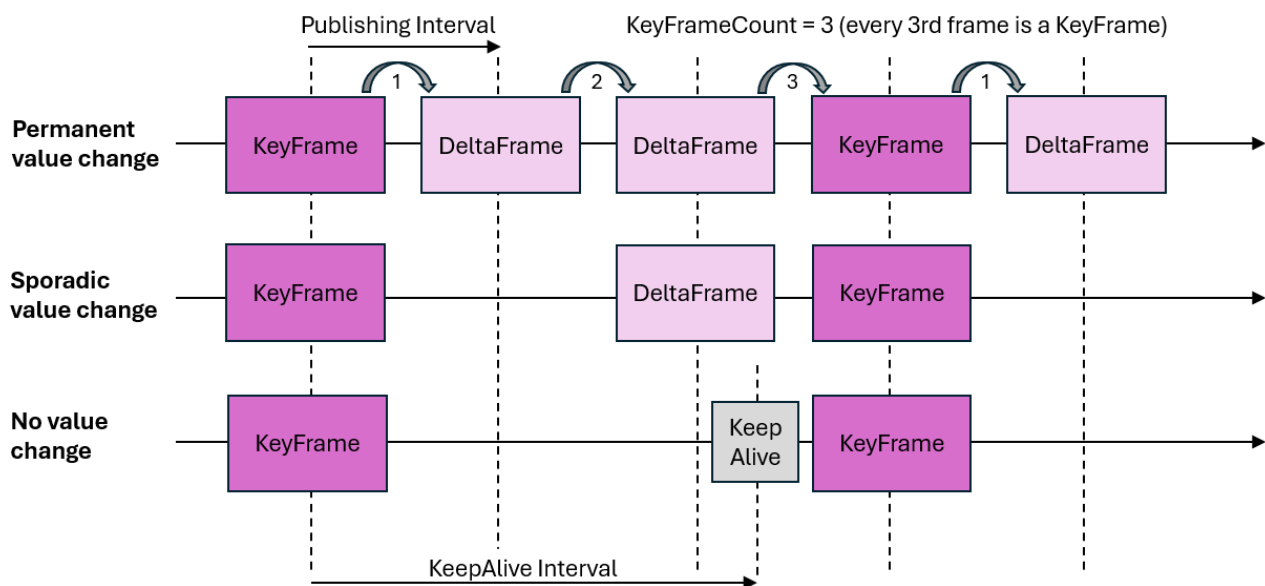
总结

关键帧、增量帧、KeyFrameCount 和 KeepAliveInterval 的结合，能够实现高效且可靠的数据传输：

- 关键帧可传输完整状态。
- 增量帧仅传输更改内容。
- KeyFrameCount 定义了 2 次完整状态传输之间增量帧的最大数量。
- KeepAliveInterval 可确保即使数据未发生变化，消息仍会定期发送。

通过适当的参数设置，可以在带宽消耗与系统鲁棒性之间实现最佳平衡。

下图再次说明了这些关系。

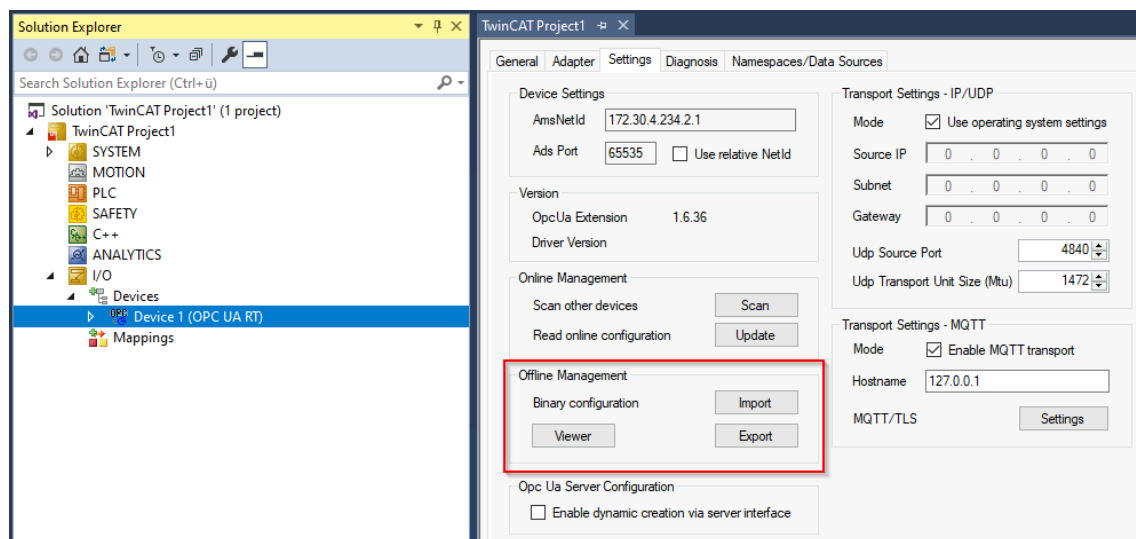


4.9 配置导入/导出

在项目的开发阶段，配置交换是非常重要的步骤，特别是当需要在不同设备和/或供应商之间交换 Pub/Sub 配置时。OPC UA Pub/Sub 内含用于交换配置的基于文件的标准化格式，称为“UA 二进制导入/导出”，或简称为“配置导入/导出”。工作流程通常如下所示：

- 在设备 A 上设置您的发布者。
- 导出配置文件。
- 在其他设备上导入配置文件，以设置相应的订阅者。

在 TwinCAT XAE 中的 OPC UA RT 设备上，有一个配置对话框可用于访问此导入/导出功能。



i 更改配置

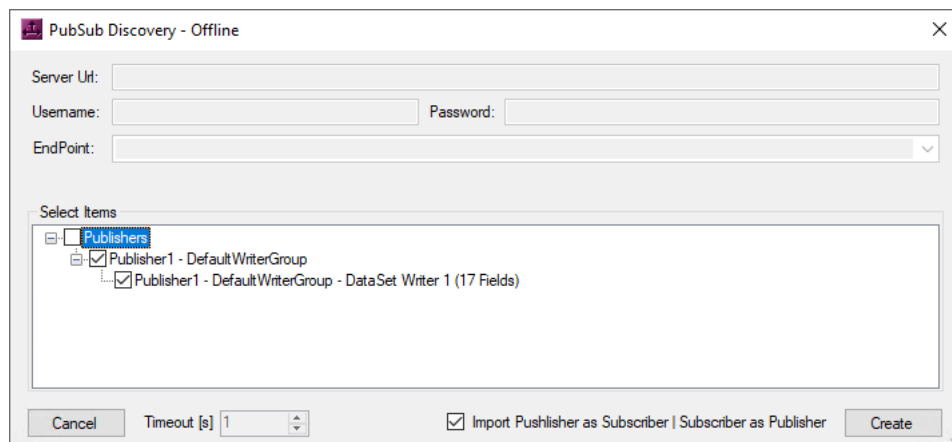
请注意，每次更改配置时，都需要创建新的配置文件。

导出

导出配置文件时，会弹出对话框，您可选择文件的保存位置。然后，您可以将生成的文件分享给相关方。

导入

导入配置文件时，会弹出对话框，您可选择应从该文件中导入的发布者/订阅者。默认情况下，发布者会导入为订阅者，反之亦然。



i MQTT 连接设置

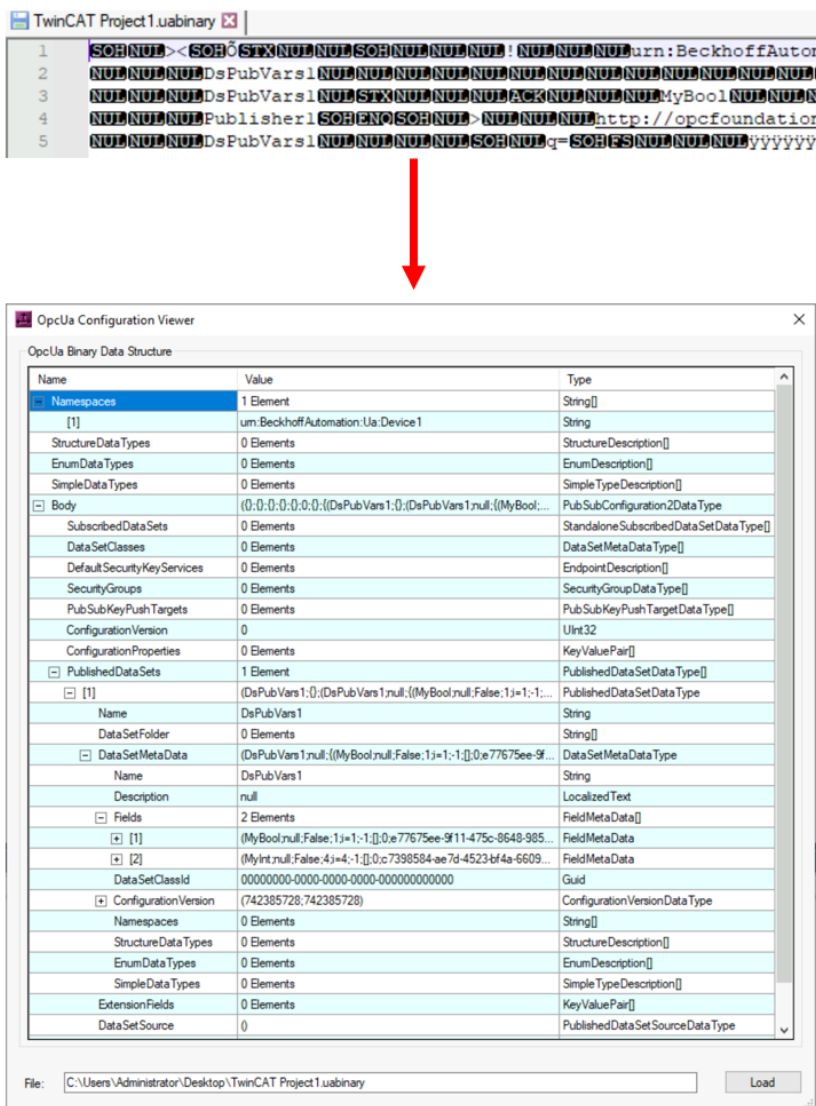
导入包含 MQTT 连接设置的配置文件时，连接详细信息会按以下方式导入：

如果所有配置项的 MQTT 连接设置均相同，则这些设置会导入并添加至 OPC UA RT 设备中。

如果某些配置项上的 MQTT 连接设置不同，则这些设置会被导入到发布者/订阅者节点上。

配置查看器

该设置区域还包含所谓的“配置查看器”，可用于预览二进制 Pub/Sub 配置文件的内容。

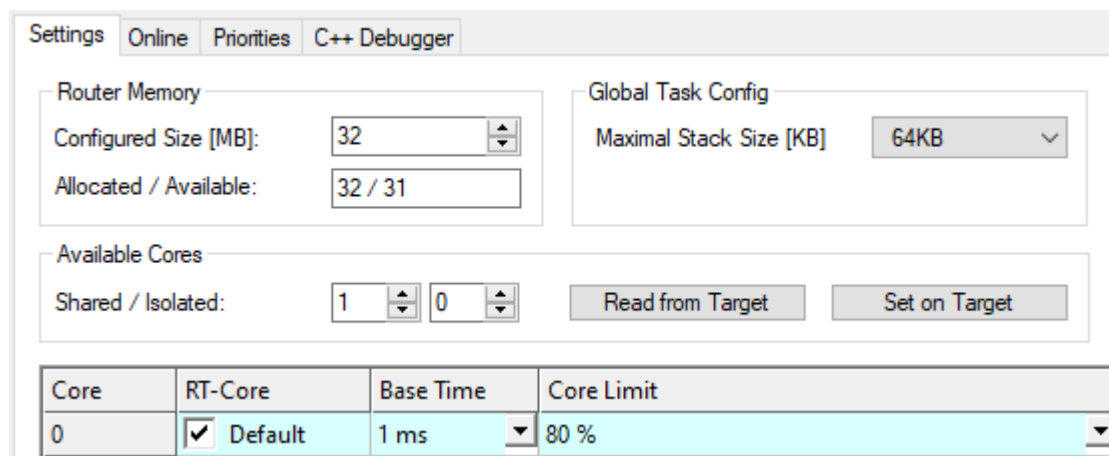


4.10 深入

4.10.1 发布间隔时间

PublishingInterval 作为“发布者”配置的一部分进行配置。指定各帧应发布的间隔时间，并与在 TwinCAT Real-Time 设置中配置的 BaseTime 具有相关性。

示例：下方截图展示了具有已在 TwinCAT Real-Time 设置中配置 1 ms BaseTime 的系统。



在配置“发布者”时，PublishingInterval 与该 BaseTime 具有相关性。下方截图展示的是 PublishingInterval 已配置为 100 ms（100 个刻度 * 1 ms BaseTime）。

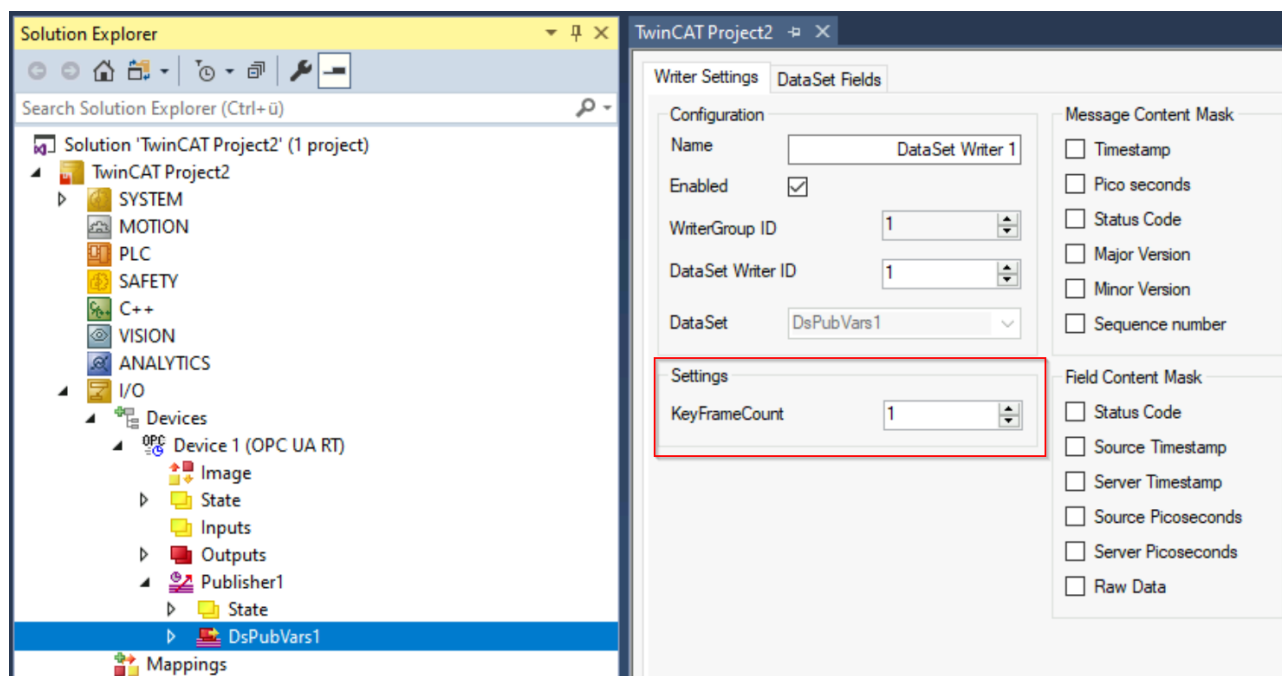


4.10.2 KeyFrames、DeltaFrames、KeepAlive

本文档介绍了在发布者或发布者数据集上配置 KeyFrames、DeltaFrames 和 KeepAlive 时，彼此之间建立相关性的方式。

关键帧和增量帧

关键帧始终包含完整数据集，而增量帧仅包含增量更新（已发生变化的字段）。在配置“发布者”数据集时，KeyFrameCount 指定了相对于发布间隔时间，关键帧应以何种频率进行发布。



当设置为“1”时，每个“发布者”循环都会发布关键帧。当设置为大于“1”的值时，每 n 个循环发布一个关键帧；否则发布一个增量帧。请注意，增量帧只会在数据发生变化时发布，这意味着有时可能不会发布任何帧。

KeepAlive

在发布者上配置的 KeepAlive 时间用于在较长时间内没有数据变化（因此也没有增量帧）时发送帧。这样可以确保订阅者仍然知道发布者在线。请注意，在订阅者上配置 KeepAlive 时，同样与 MessageReceiveTimeout 具有相关性。

4.10.3 MessageReceiveTimeout

MessageReceiveTimeout 作为“订阅者”配置的一部分进行配置。指定订阅者数据集预计收到新消息所需的时间，否则将发生超时，且 PubSubState 进程变量将进入错误状态。MessageReceiveTimeout 可能会被 [KeepAlive](#) [▶ 41](#) 重置。

5 示例

本产品的示例代码和配置可从 GitHub 上的相应存储库获取：https://github.com/Beckhoff/TF6105_Samples。您可以选择克隆存储库或下载包含示例的 ZIP 文件。目前可用的示例如下：

示例	描述
UDP [► 43]	演示如何根据 OPC UA Pub/Sub UDP 设置发布者和订阅者示例。
MQTT [► 44]	演示如何根据 OPC UA Pub/Sub MQTT 设置发布者和订阅者示例。
发布来自 EtherCAT 主站的数据 [► 45]	说明如何设置 EtherCAT 主站 ADS 符号服务器，并配置 OPC UA 发布者以发布其数据点。本示例无法下载，但本文档中将详细说明所有必要步骤。

同时，我们建议您查看我们的“快速入门”教程 [► 15]，其中针对各种配置用例提供了分步指导。

5.1 UDP

本示例演示了如何根据 OPC UA Pub/Sub UDP 设置发布者和订阅者。解决方案示例中包含 TwinCAT 项目，其中包括以下 PLC 配置：

- PLC 项目，用于生成随机数据，并将这些数据与 OPC UA RT 设备发布者上的数据集 [► 28] 建立关联。
- PLC 项目，用于将订阅者接收到的数据存储到 OPC UA RT 设备上。

该 I/O 配置内含 OPC UA RT 设备，配有 2 个发布者和 2 个订阅者：

- 已配置为包括带变量的数据集的发布者。这些变量通过过程映像与第一个 PLC 项目的输出变量相关联。
- 已配置为包括不带变量的数据集的发布者。这些变量通过其 ADS 符号名称与第一个 PLC 项目中的变量在内部相关联。
- 已配置为订阅由第一个发布者发布的数据的订阅者。输入变量与第二个 PLC 项目的变量相关联。
- 已配置为订阅由第二个发布者发布的数据的订阅者。输入变量与第二个 PLC 项目的变量相关联。

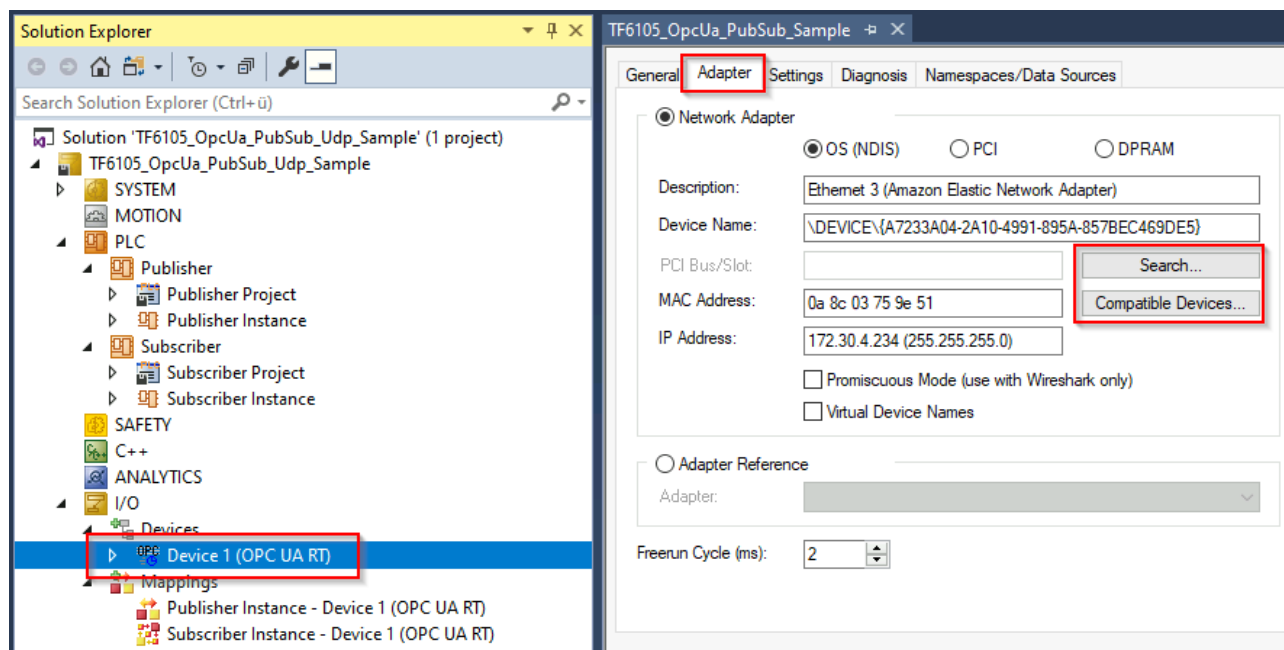


创建订阅者

这些订阅者是根据 [配置导入/导出 \[► 38\]](#) 自动创建。我们强烈建议在可能的情况下使用这种交换格式。

如何启动示例程序

在激活示例项目之前，需要将 OPC UA RT 设备设置为使用已安装 TwinCAT Realtime-Ethernet-Adapter 驱动程序的网络接口卡。可以通过 OPC UA RT 设备设置对话框来完成此操作。



此外，请确保您的系统符合系统要求 [► 8]。

5.2 MQTT

本示例演示了如何根据 OPC UA Pub/Sub MQTT 设置发布者和订阅者。解决方案示例中包含 TwinCAT 项目，其中包括以下 PLC 配置：

- PLC 项目，用于生成随机数据，并将这些数据与 OPC UA RT 设备发布者上的数据集 [► 28] 建立关联。
- PLC 项目，用于将订阅者接收到的数据存储在 OPC UA RT 设备上。

该 I/O 配置内含 OPC UA RT 设备，配有 2 个发布者和 2 个订阅者：

- 已配置为包括带变量的数据集的发布者。这些变量通过过程映像与第一个 PLC 项目的输出变量相关联。
- 已配置为包括不带变量的数据集的发布者。这些变量通过其 ADS 符号名称与第一个 PLC 项目中的变量在内部相关联。
- 已配置为订阅由第一个发布者发布的数据的订阅者。输入变量与第二个 PLC 项目的变量相关联。
- 已配置为订阅由第二个发布者发布的数据的订阅者。输入变量与第二个 PLC 项目的变量相关联。

● 创建订阅者

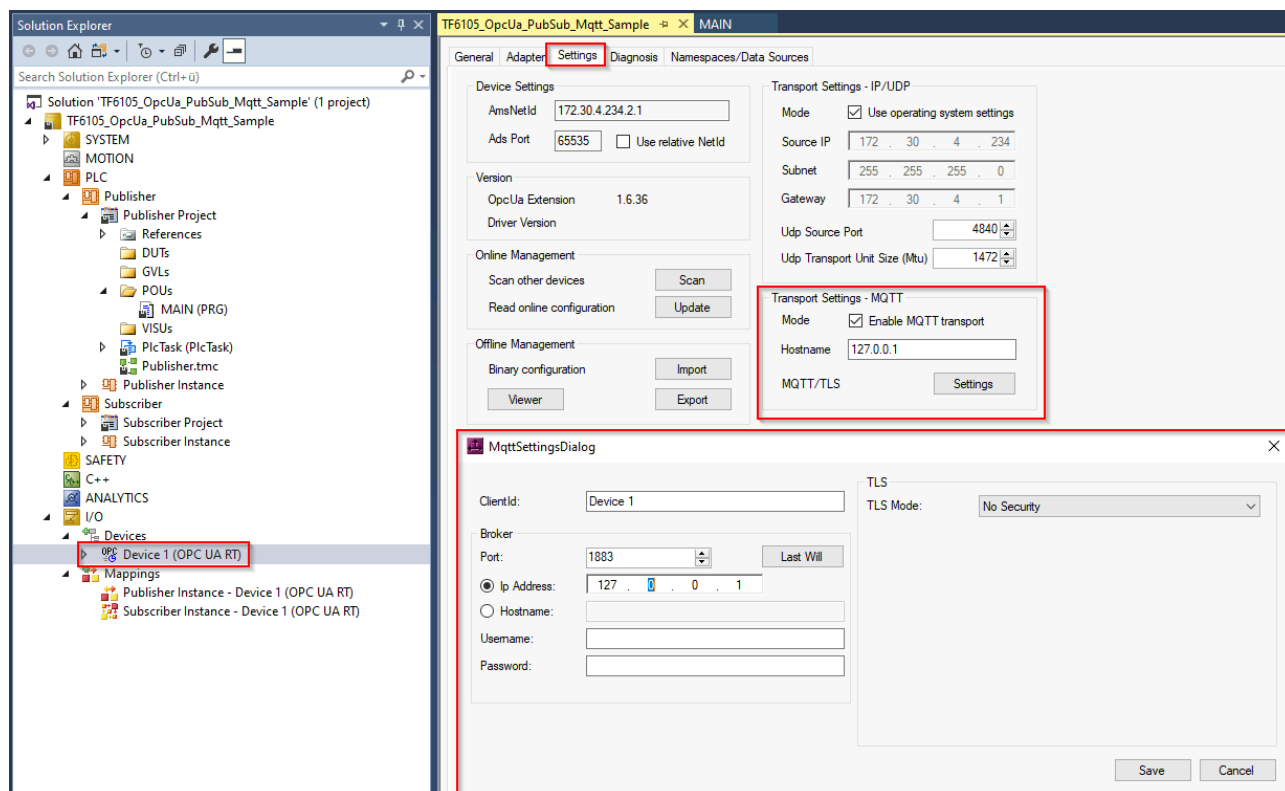
i

这些订阅者是根据 配置导入/导出 [► 38] 自动创建。我们强烈建议在可能的情况下使用这种交换格式。

如何启动示例程序

请确保您的系统符合系统要求 [► 8]。

在激活 TwinCAT 项目之前，请确保同一系统上已有 MQTT 消息代理在本地运行。消息代理应监听 1883/tcp 端口，且不进行任何加密或用户身份验证。请注意，这仅用于演示目的！请根据您的环境调整此配置。可在 OPC UA RT 设备上查阅 MQTT 设置。



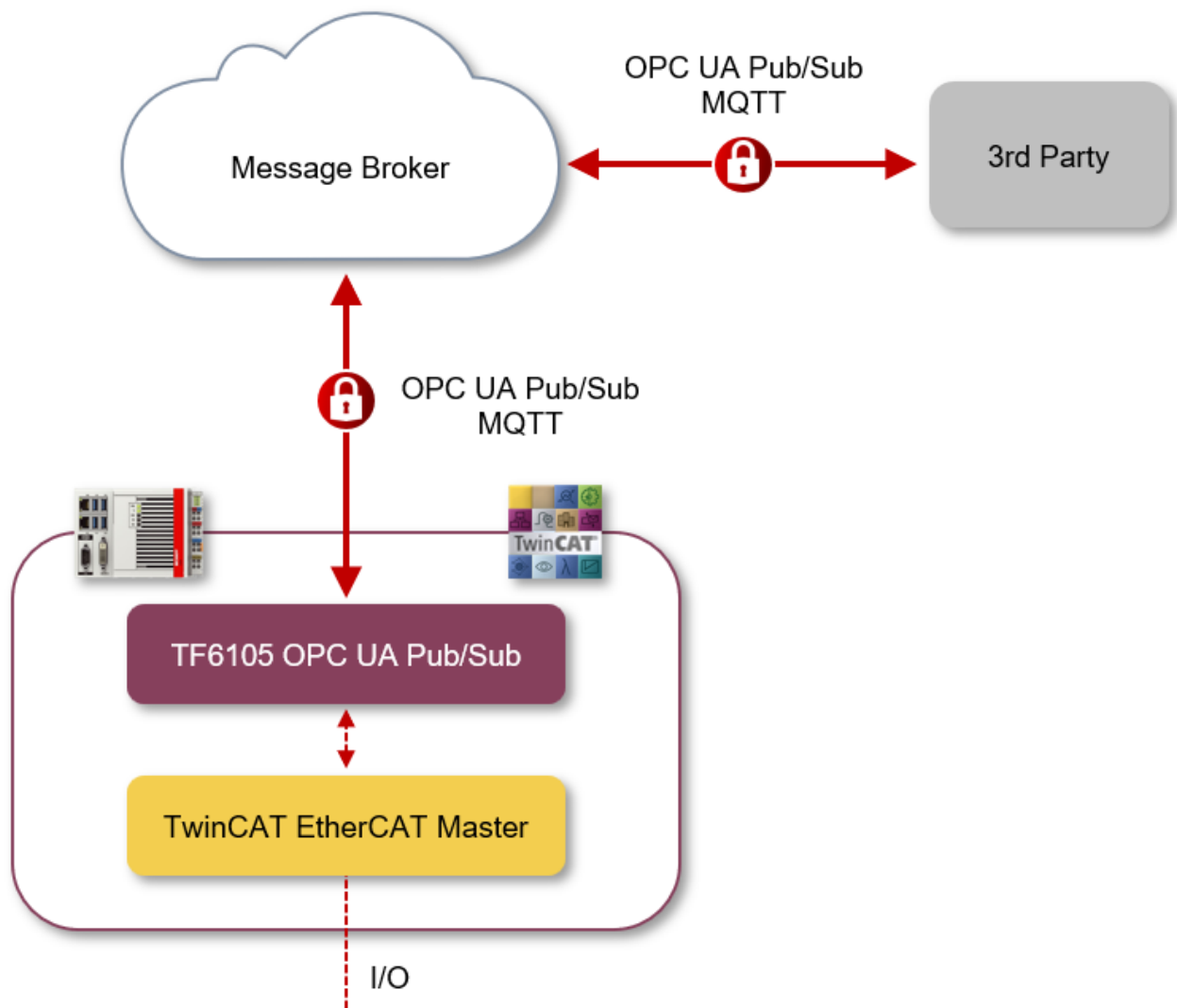
然后，您可以激活 TwinCAT 项目，并登录 PLC 项目以调试 PLC 运行时。您应该能看到，订阅者 PLC 项目接收到了数据（该数据由另一个 PLC 项目发布）。

5.3 EtherCAT 主站

您可以从 EtherCAT 主站图像轻松地发布数据点。虽然不需要执行中间 PLC 项目，但如果您需要进行数据转换或变换，也可以添加该项目。以下文档介绍了设置此用例所需的所有步骤。这些步骤包括：

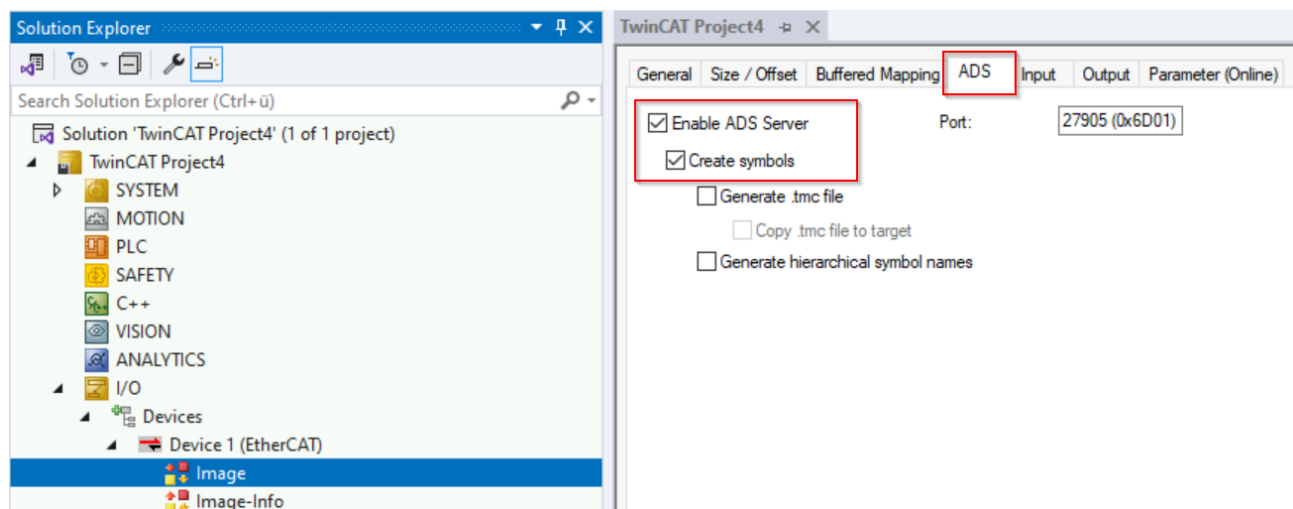
1. 在 EtherCAT 主站上激活 ADS 符号服务器
2. 设置 OPC UA RT 设备
3. 使用数据集配置发布者
4. 将来自 EtherCAT 主站的数据点添加到发布者数据集
5. [可选] 设置 TwinCAT 订阅者
6. [可选] 设置通用 MQTT 客户端订阅者

本示例基于用例演示该设置方案，该用例要求将来自 EtherCAT 主站的 I/O 数据点发布到云端的消息代理。随后，应将第三方应用连接到同一消息代理并订阅数据。本示例可根据应用需求，适应该用例的不同变体。



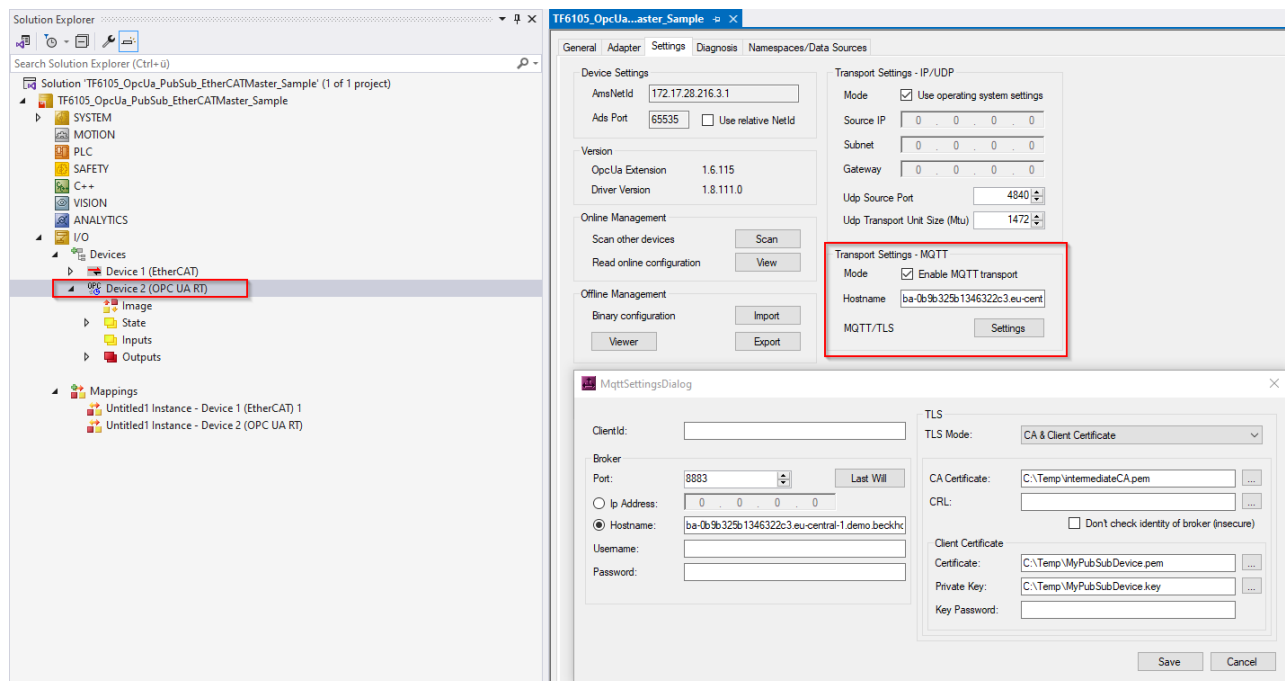
在 EtherCAT 主站上激活 ADS 符号服务器

TwinCAT EtherCAT 主站可提供 ADS 符号服务器的各个功能。随后，EtherCAT 主站上的所有端子模块和变量都会通过 ADS 提供。ADS 客户端可以连接至 ADS 符号服务器，以便浏览其命名空间并读写数据。您可以在 EtherCAT 主站图像的 ADS 设置页面上激活此功能。



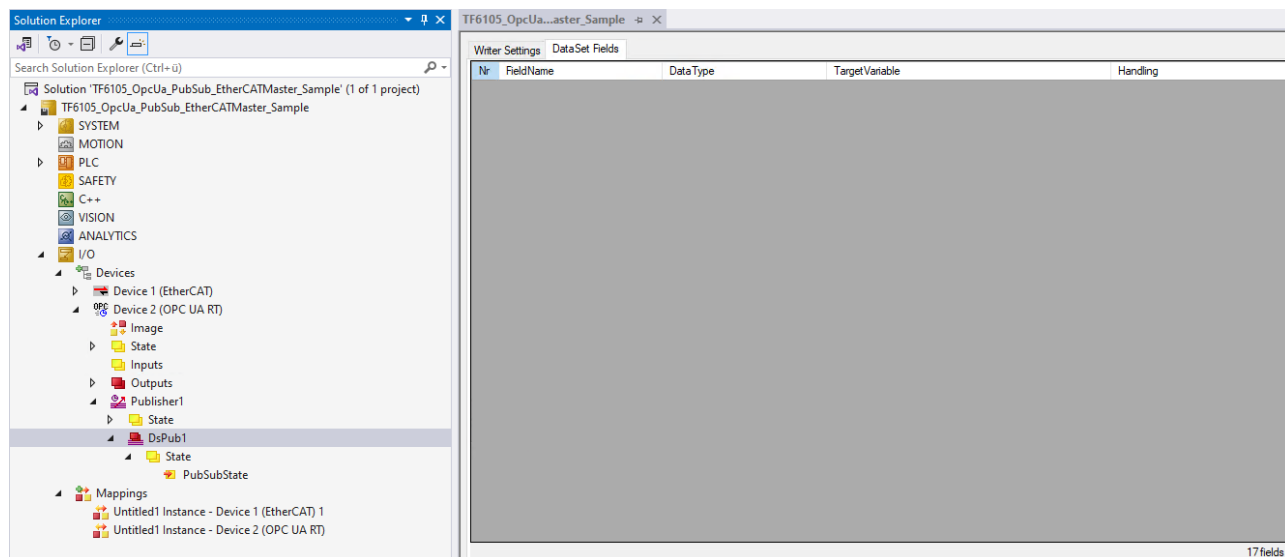
设置 OPC UA RT 设备

按照我们的快速入门 [▶ 15] 教程中的说明，配置 OPC UA RT 设备。在本示例中，将配置 OPC UA RT 设备，使其通过安全 MQTT 连接与运行在云端的远程消息代理建立连接，但您也可以在本地运行本示例。请根据您的环境输入 MQTT 连接详情。



使用数据集配置发布者

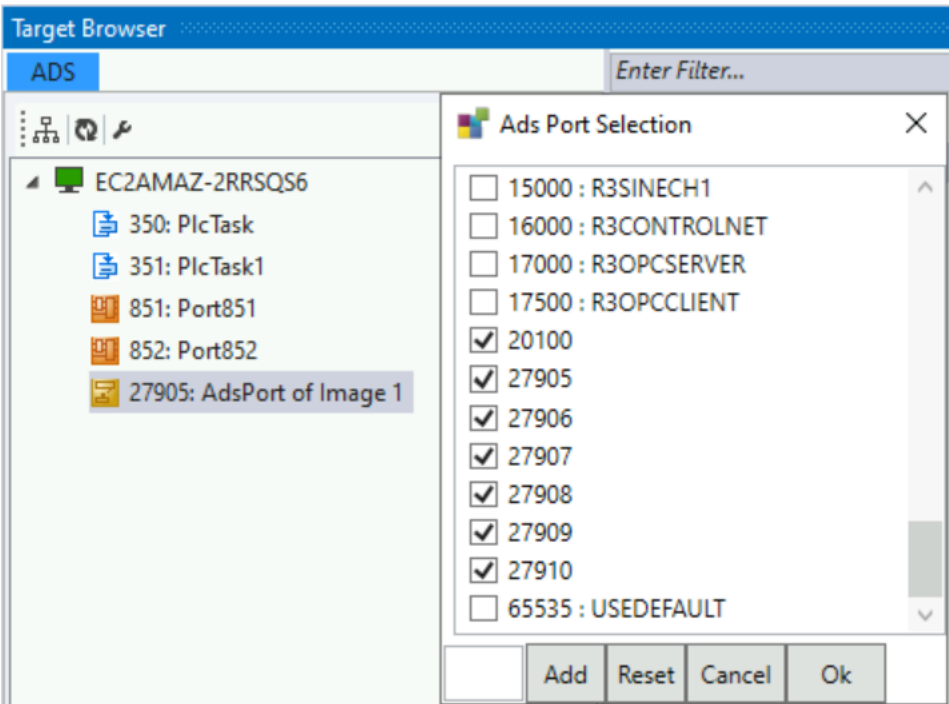
接下来，您需要创建发布者，其中包含数据集，内含与 EtherCAT 主站图像的数据点相关联的变量。右键单击 OPC UA RT 设备，选择 **Add New Item**（添加新项目），然后选择 **OPC UA Publisher**（OPC UA 发布者）。然后右键单击已添加的发布者，选择 **Add New Item**（添加新项目），再选择 **Published DataSet**（已发布的数据集）。请注意，如下方截图所示，该数据集目前尚未包含任何字段。



现在，您可以使用 TwinCAT Target Browser 从 EtherCAT 主站添加数据点。

将来自 EtherCAT 主站的数据点添加到发布者数据集

打开 TwinCAT Target Browser，并导航至 EtherCAT 主站的 ADS 符号服务器。如果尚未显示此 ADS 服务器，您可以通过“ADS 端口选择”对话框添加其端口，如下方截图所示。

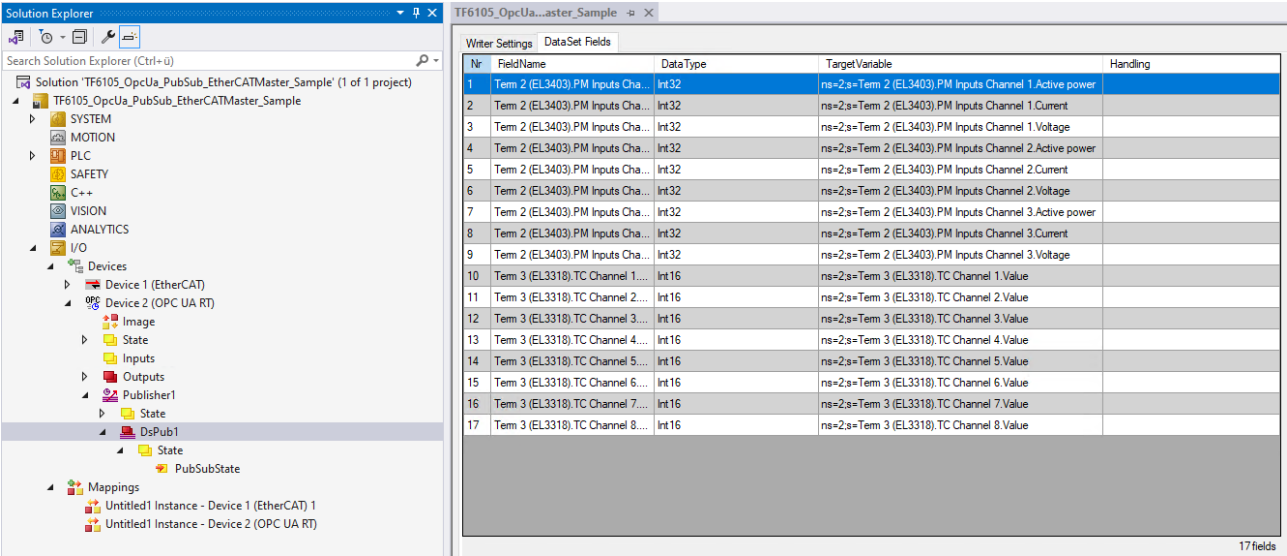


EtherCAT 主站图像的 ADS 端口可在第 1 步的设置页面中找到。

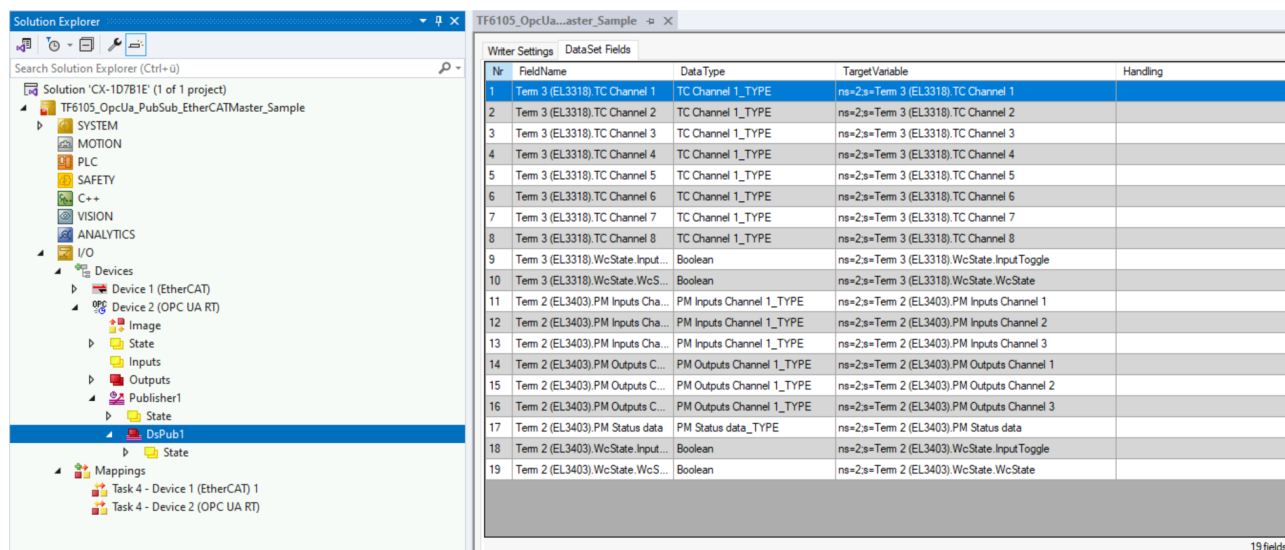
现在，您可以浏览 EtherCAT 主站图像的命名空间，并选择要发布的数据点。添加数据点时，您有 4 种选择：

1. 请逐一选择每个数据点，然后使用 Drag&Drop（拖放）功能将其添加到数据集中。
2. 选中整个过程映像，然后使用 Drag&Drop（拖放）功能将其添加到数据集中。在这种情况下，图像中的每个端子模块都将作为结构化数据类型添加到数据集中。
3. 选中整个过程映像，然后按住 SHIFT + Drag&Drop（拖放），将其添加到数据集。在这种情况下，只有各端子模块上的变量会被添加到数据集中（作为简单数据类型）。如果 OPC UA Pub/Sub 订阅者不支持结构化类型，则可能需要使用此选项。
4. 选择端子模块，并使用 Drag&Drop（拖放）功能将其添加到数据集。

下方截图展示了针对 2 个端子模块（EL3403 和 EL3318）执行 Drag&Drop（拖放）操作后的结果（如选项 1 所述）。



下方截图展示了针对 2 个端子模块（EL3403 和 EL3318）执行 Drag&Drop（拖放）操作后的结果（如选项 4 所述）。



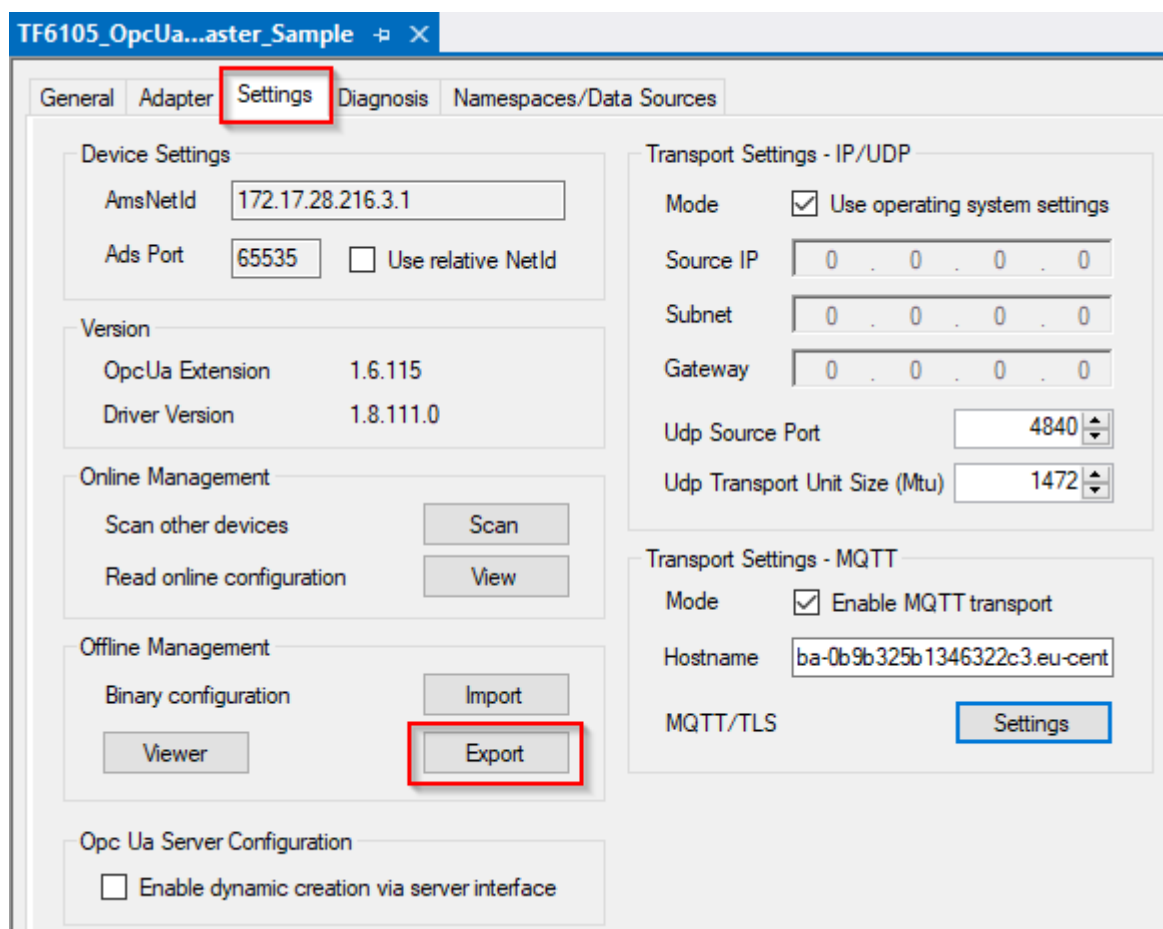
The screenshot shows the TwinCAT Solution Explorer on the left, displaying a project structure for 'TF6105_OpcUa...aster_Sample'. The main window displays a table of 'Data Set Fields'.

Nr	FieldName	Data Type	Target Variable	Handling
1	Term 3 (EL3318).TC Channel 1	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 1	
2	Term 3 (EL3318).TC Channel 2	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 2	
3	Term 3 (EL3318).TC Channel 3	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 3	
4	Term 3 (EL3318).TC Channel 4	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 4	
5	Term 3 (EL3318).TC Channel 5	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 5	
6	Term 3 (EL3318).TC Channel 6	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 6	
7	Term 3 (EL3318).TC Channel 7	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 7	
8	Term 3 (EL3318).TC Channel 8	TC Channel 1_TYPE	ns=2;s=Term 3 (EL3318).TC Channel 8	
9	Term 3 (EL3318).WcState.Input...	Boolean	ns=2;s=Term 3 (EL3318).WcState.InputToggle	
10	Term 3 (EL3318).WcState.WcS...	Boolean	ns=2;s=Term 3 (EL3318).WcState.WcState	
11	Term 2 (EL3403).PM Inputs Cha...	PM Inputs Channel 1_TYPE	ns=2;s=Term 2 (EL3403).PM Inputs Channel 1	
12	Term 2 (EL3403).PM Inputs Cha...	PM Inputs Channel 1_TYPE	ns=2;s=Term 2 (EL3403).PM Inputs Channel 2	
13	Term 2 (EL3403).PM Inputs Cha...	PM Inputs Channel 1_TYPE	ns=2;s=Term 2 (EL3403).PM Inputs Channel 3	
14	Term 2 (EL3403).PM Outputs C...	PM Outputs Channel 1_TYPE	ns=2;s=Term 2 (EL3403).PM Outputs Channel 1	
15	Term 2 (EL3403).PM Outputs C...	PM Outputs Channel 1_TYPE	ns=2;s=Term 2 (EL3403).PM Outputs Channel 2	
16	Term 2 (EL3403).PM Outputs C...	PM Outputs Channel 1_TYPE	ns=2;s=Term 2 (EL3403).PM Outputs Channel 3	
17	Term 2 (EL3403).PM Status data	PM Status data_TYPE	ns=2;s=Term 2 (EL3403).PM Status data	
18	Term 2 (EL3403).WcState.Input...	Boolean	ns=2;s=Term 2 (EL3403).WcState.InputToggle	
19	Term 2 (EL3403).WcState.WcS...	Boolean	ns=2;s=Term 2 (EL3403).WcState.WcState	

[可选] 设置 TwinCAT 订阅者

您可以通过将配置导出 [► 38] 为 OPC UA 二进制交换文件，轻松为已发布的数据点设置订阅者。随后，该文件可在另一端导入，以自动创建适配发布者配置的订阅者。

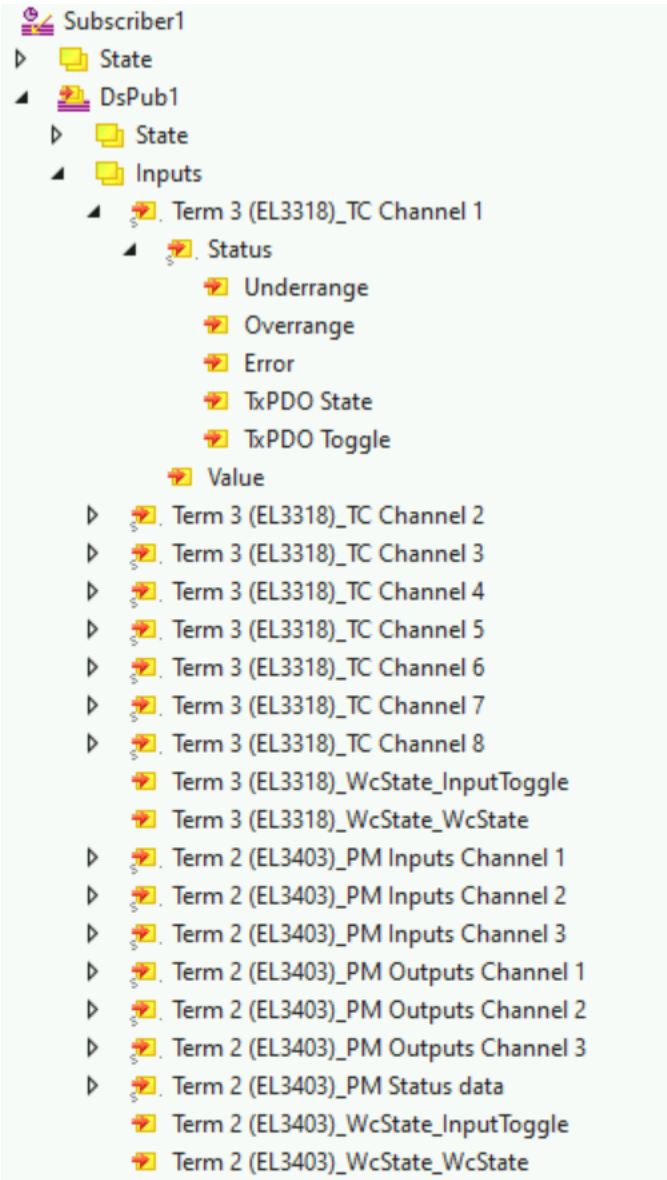
要导出配置，请双击 OPC UA RT 设备，然后选择 **Settings (设置)**。点击 **Export (导出)** 按钮，将配置保存到文件中。



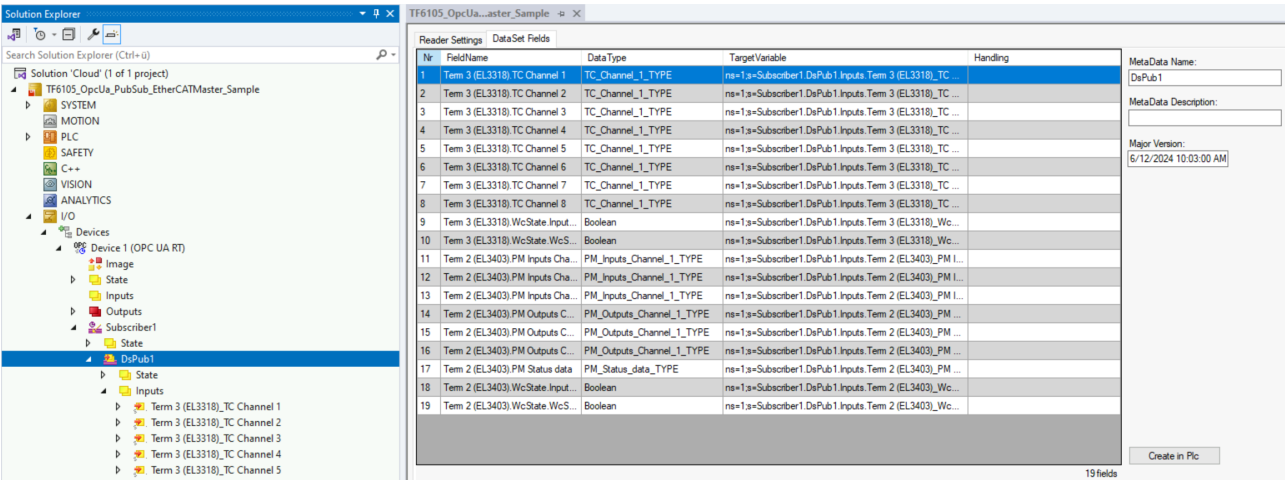
在应托管订阅者的设备上导入此配置文件。如果该设备同时也是运行 TwinCAT 的系统，则可以通过 OPC UA RT 设备上的 **Import (导入)** 按钮，将文件导入到相应的 TwinCAT 项目中。

您还可以在运行发布者的同一项目中创建订阅者。因此，该控制器将同时充当发布者和订阅者。

导入配置文件后，订阅者将自动配置为使用包含发布者字段的数据集。



然后，您可以创建包含变量的映射（例如来自 PLC 应用），或者通过数据集配置上的 **Create PLC code**（**创建 PLC 代码**）按钮自动生成 PLC 代码。



此操作将创建 GVL，其中包含通过“TcLinkTo”属性与过程映像变量关联的变量，例如：


```
Subscriber1_DsPub1  ▸ ×
1  VAR_GLOBAL
2  {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^State^PubSubState'}
3  PubSubState AT %I* : OpcUaPubSubState;
4  {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 1'}
5  Term_3_TC_Channel_1 AT %I* : OpcUa_TC_Channel_1_TYPE;
6  {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 2'}
7  Term_3_TC_Channel_2 AT %I* : OpcUa_TC_Channel_1_TYPE;
8  {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 3'}
9  Term_3_TC_Channel_3 AT %I* : OpcUa_TC_Channel_1_TYPE;
10 {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 4'}
11 Term_3_TC_Channel_4 AT %I* : OpcUa_TC_Channel_1_TYPE;
12 {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 5'}
13 Term_3_TC_Channel_5 AT %I* : OpcUa_TC_Channel_1_TYPE;
14 {attribute 'TcLinkTo' := 'TIID^Device 1 (OPC UA RT)^Subscriber1^DsPub1^Inputs^Term 3 (EL3318)_TC Channel 6'}
15 Term_3_TC_Channel_6 AT %I* : OpcUa_TC_Channel_1_TYPE;
```

[可选] 设置通用 MQTT 客户端订阅者

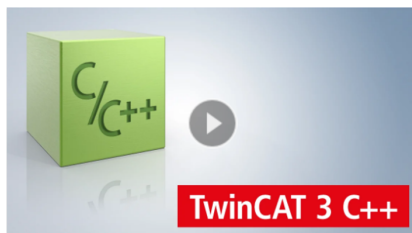
任意 MQTT 客户端均可订阅已发布至 MQTT 消息代理的数据。在本示例中，我们采用 MQTT/JSON 数据格式，该格式以 ASCII 为基础，便于对数据进行解析。下文展示输出的 JSON 格式示例。

```
{
  "DataSetWriterId": 1,
  "PublisherId": "1",
  "SequenceNumber": 5228,
  "MinorVersion": 771501780,
  "Timestamp": "2024-06-12T10:14:04.734Z",
  "Status": 0,
  "Payload": {
    "Term 3 (EL3318).TC Channel 1": {
      "Status": {
        "Underrange": false,
        "Overrange": false,
        "Error": false,
        "TxPDO State": false,
        "TxPDO Toggle": false
      },
      "Value": 233
    },
    "Term 3 (EL3318).TC Channel 2": {
      "Status": {
        "Underrange": true,
        "Overrange": false,
        "Error": false,
        "TxPDO State": false,
        "TxPDO Toggle": false
      },
      "Value": 13720
    },
    ...
  },
  ...
}
```

6 教程

可在我们的网站 <https://www.beckhoff.com/tutorials> 上查看本产品的视频教程。该视频教程提供了对产品

和产品各个方面的快速入门简介。



14.03.2024 | Multimedia

Tutorial: TwinCAT 3 C++ overview

Get an overview of the C++ integration into TwinCAT.

[Learn more →](#)



12.02.2024 | Multimedia

Tutorial: Chart synchronization in TwinCAT Scope

Learn how to dock and synchronize charts in TwinCAT Scope.

[Learn more →](#)



12.02.2024 | Multimedia

Tutorial: Getting started with TF6100 TwinCAT 3 OPC UA Server

Learn how to configure the TwinCAT OPC UA Server.

[Learn more →](#)

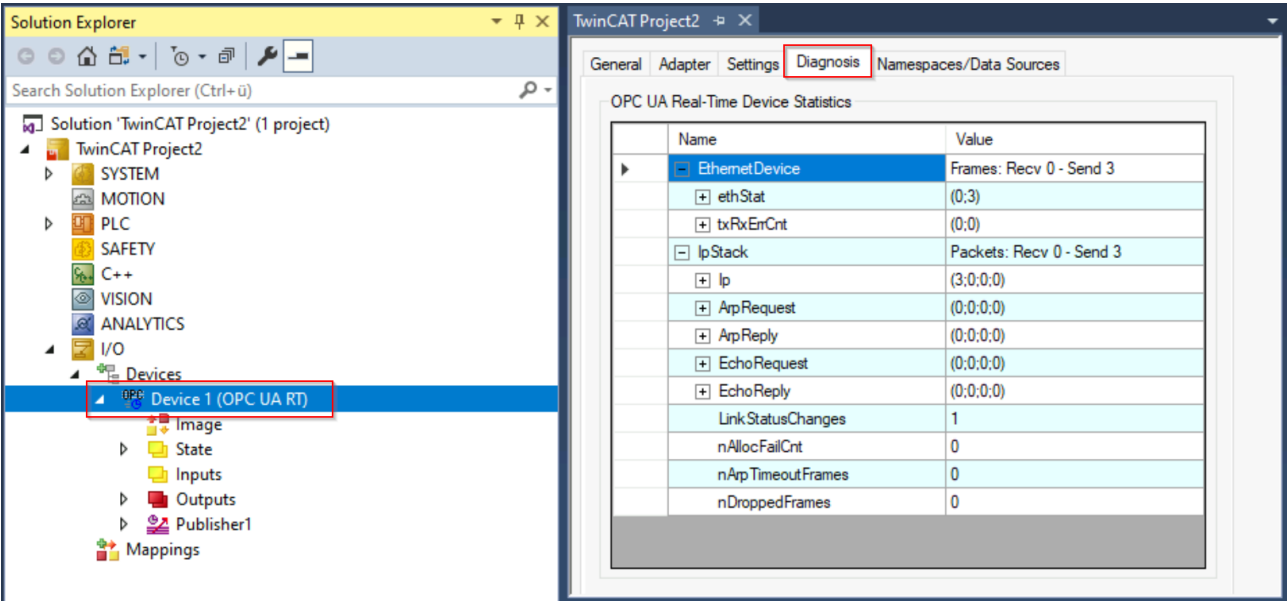
7 附录

7.1 诊断

您可以在不同的配置级别上查看诊断信息。可在相应配置项的单独选项卡页面上查看相关信息。

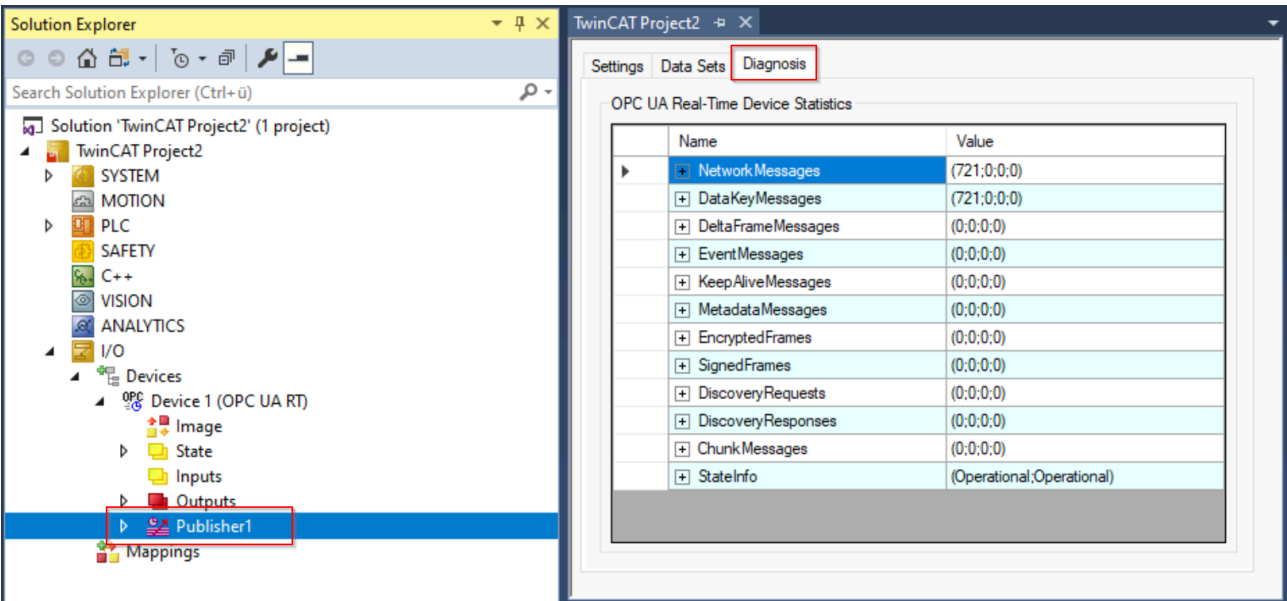
设备

OPC UA RT 设备中包含 **Diagnosis (诊断)** 选项卡，其中提供了不同指标，用于查看底层 IP 协议栈是否已发送或接收数据包及其数量。



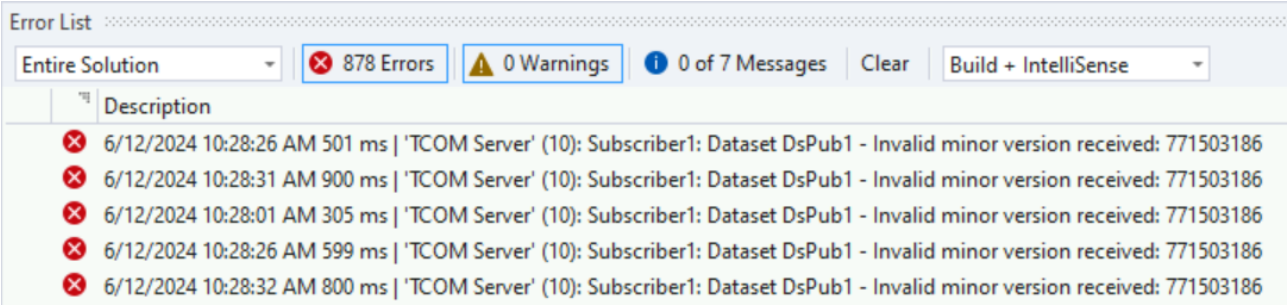
发布者/订阅者

每个发布者或订阅者都设有 **Diagnosis (诊断)** 选项卡，其中列出了该特定发布者或订阅者的不同指标。



TwinCAT 错误列表

TwinCAT OPC UA Pub/Sub 驱动程序还会将日志记录信息发送至 TwinCAT Error List 窗口，该窗口可通过 TwinCAT XAE 访问。



Wireshark®

Wireshark® 是一款适用于高级调试和诊断的优秀软件工具。您还可以使用 Wireshark® 来诊断 OPC UA Pub/Sub 数据包。

示例：下方 Wireshark® 捕获记录显示了由已配置的“发布者”产生的流量，该发布者每隔 1,000 ms（第一列显示的时间戳）向 UDP 多播组 224.0.0.1 发送数据包。

16	2.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
20	3.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
24	4.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
27	5.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
30	6.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
37	7.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
46	8.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
49	9.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10
54	10.740000000	172.17.98.153	224.0.0.1	OPCUA	UADP	52	4840	→	4840	Len=10

每个数据包都包含已配置的数据集及其数据类型为 INT（在本例中为“计数器”）的变量（该类型等同于 Int16）。

UADP Network Message

> NetworkMessage Header

Payload Header

> DataSet Payload

> DataSetMessage[0] (Data Key Frame Data)

> DataSetFlags1

> Data Key Frame

Field Count: 1

> DataSetField[0] (Variant Encoding)

> Variant Encoding Mask: 4

> Value (Int16)

18889

i

Wireshark® 插件/解析器

如果您希望以易读方式追踪和解析 Pub/Sub 消息，可能需要使用 OPC UA Pub/Sub Wireshark® 解析器。

7.2 故障排除

行为	注意
我无法接收订阅者的任何数据，且 TwinCAT Error List 中显示多条错误信息，内容均为“接收到的次版本号无效”。	当源数据发生变化时，通常会出现此错误，例如当发布者数据集中添加新字段时。在这种情况下，OPC UA Pub/Sub 数据包中的版本信息会递增，因此需要将新配置同步至订阅者。 实现这一目标最简单的方法是将更新后的发布者配置导出到配置文件 [▶ 38] 中，然后在订阅者端将其导入。

行为	注意
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_NOT_FOUND”。	请确保您使用了 MQTT 消息代理的正确主机名或 IP 地址。
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_CONN_TIMEDOUT”。	请确保您使用了 MQTT 消息代理的正确主机名或 IP 地址，并且可以通过该地址访问该代理。此外，请确保已指定消息代理的正确 TCP/IP 端口，例如 1883 或 8883。
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_TLS_VERIFY_FAIL”。	当 Pub/Sub 驱动程序不接受消息代理证书时，通常会弹出此错误消息，这通常与使用了错误的 CA 证书有关。 请确保您已指定正确的 CA 证书。
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_TLS_HANDSHAKE_FAIL”。	当消息代理不接受您的客户端证书时，通常会弹出此错误消息。如果您完全未指定任何客户端证书，也可能弹出此错误消息。 请确保您已指定正确的客户端证书，并且该证书已被消息代理视为可信。
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_TLS_KEY_NOTFOUND”。	当无法找到客户端密钥时，通常会弹出此错误信息。 请确保客户端密钥位于您在 MQTT 连接设置中指定的路径下。
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_TLS_KEY_INVALID”。	当无法打开客户端密钥时，通常会弹出此错误消息。 请确保已在 MQTT 连接设置中指定有效的客户端密钥文件，并且未将客户端密钥与客户端证书混淆。
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_TLS_CERT_NOTFOUND”。	当无法找到客户端证书时，通常会弹出此错误消息。 请确保客户端证书位于您在 MQTT 连接设置中指定的路径下。
无法与消息代理建立连接，且 TwinCAT Error List 中显示多条错误信息，内容均为“连接 MQTT 客户端时出错 MQTT_ERR_TLS_CERT_INVALID”。	当无法打开客户端证书时，通常会弹出此错误消息。 请确保已在 MQTT 连接设置中指定有效的客户端证书文件，并且未将客户端证书与客户端密钥混淆。
当我将 TwinCAT 设为 Config Mode（配置模式），然后在 Load I/O Devices（加载 I/O 设备） 对话框中确认时，会出现错误消息：“SafeOP 到 OP 的操作……失败。参数无效 0x9811070B。”	由于 TwinCAT 状态发生变化，OPC UA Pub/Sub 驱动程序也会发生上下文切换。然而，由于切换到 Config Mode（配置模式），PLC 停止运行，因此会显示这条错误消息。当我点击 TwinCAT XAE 工具栏中的 Reload Devices（重新加载设备） 按钮时，也会出现同样的情况。这种行为是按设计构建的。

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

Docker is a trademark or registered trademark of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

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.

PCI Express®, PCIe®, PCI™ and PCI HOT PLUG™ are trademarks or registered trademarks and/or service marks of PCI-SIG.

Wireshark is a registered trademark of Sysdig, Inc.

OPC UA is a registered trademark of the OPC Foundation.

更多信息:
www.beckhoff.com/tf6105

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

