



KEPServerEX 中 MQTT 应用

北京东方鼎晨科技有限公司

2020 年 3 月 24 日

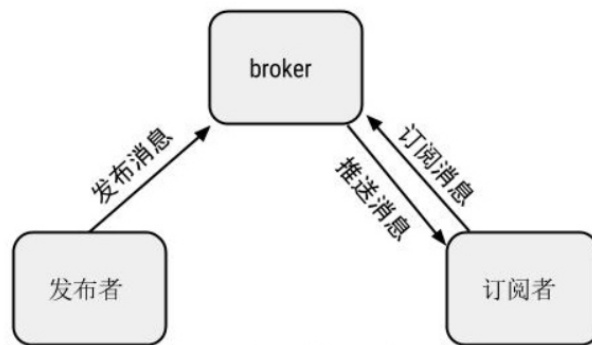
目录

1. MQTT 简介.....	2
2. KEPServerEX 中的 MQTT 应用	2
2.1. MQTT Client Driver 应用.....	2
2.2. IOT Gateway Plug-in 应用	3
3. KEPServerEX 中的 MQTT 配置	5
3.1. MQTT Client Driver 配置.....	5
3.1.1. 建立通道.....	5
3.1.2. 建立设备.....	6
3.1.3. 建立标签.....	6
3.1.4. Quick Client 查看数据	8
3.2. IOT Gateway Plug-in 中的 MQTT Client 配置.....	8
3.2.1. 建立 Agent.....	8
3.2.2. 建立 Item	9

1. MQTT 简介

物联网最近曝光率越来越高。虽然 HTTP 是网页的事实标准，不过机器之间的大规模沟通需要不同的通讯模式，之前的请求/回答 模式不再合适，取而代之的是发布/订阅 模式。这就是轻量级、可扩展的 MQTT 可以施展拳脚的舞台。

MQTT 是一个轻型协议，使用基于 TCP/IP 协议的发布/订阅消息转发模式。MQTT 协议的中心是 MQTT 服务器或代理 (broker)，支持发布程序和订阅程序进行访问，如下图所示：



2. KEPServerEX 中的 MQTT 应用

在 KEPServerEX 软件中 MQTT 有两种应用，一个是 Driver 层面的应用：MQTT Client Driver，一个是 Plug-in 层面的应用：IoT Gateway Plug-in。

2.1. MQTT Client Driver 应用

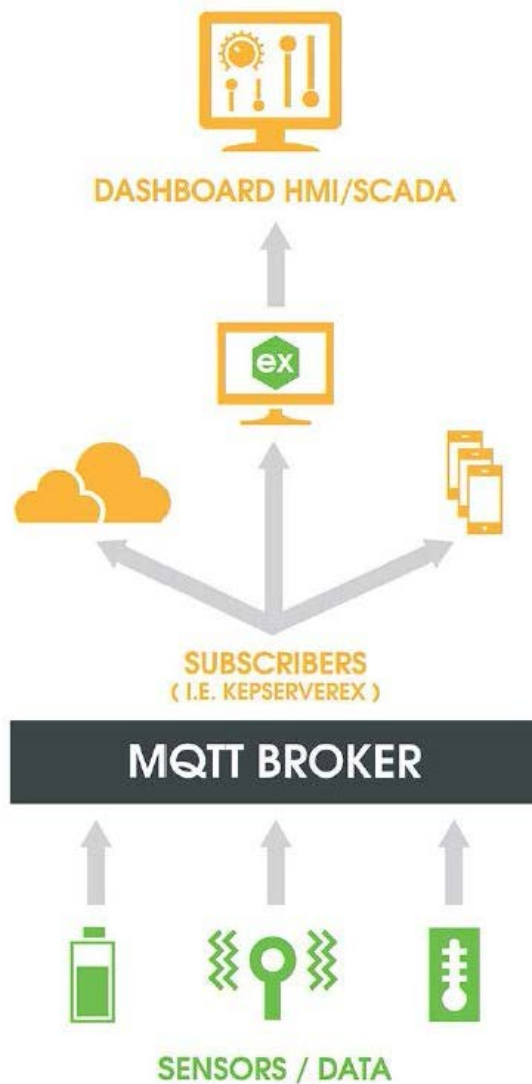
MQTT Client Driver 提供了一种获取设备数据的方法，这些来自设备的数据通过 MQTT 协议发送。许多设备和传感器使用不同的或专有的通信协议。使用 MQTT 可以统一数据收集和发布。此驱动程序连接到 MQTT 服务器（代理）并订阅指定的主题。当接收到关于这些主题的更新时，将解析有效负载并将这些数据给到 OPC 标签，然后通过 OPC 客户端访问这些标签，从而构建有意义的数据库和仪表盘。

MQTT 客户机驱动程序支持：

- 1) 分析有效的 JSON 数据格式
- 2) 从 MQTT 服务器读取数据（当前不支持写入）
- 3) 身份验证和 TLS/SSL 加密
- 4) 使用 MQTT 3.1 和 3.1.1 协议连接到 MQTT 服务器



- 5) 与 ThingWorx 平台和其他 HMI、SCADA、物联网和基于云的解决方案集成
- 6) 基于用户管理器和安全策略插件的用户级访问



2.2. IOT Gateway Plug-in 应用

物联网网关是一个可选功能，允许通过基于行业标准 IP 的协议将系统和设备标签发布到第三方端点。当已配置标签的值发生更改或满足发布速率时，将以标准 JSON 格式向相应的第三方端点发送更新，更新的有效负载为标记 ID、值、质量和时间戳。

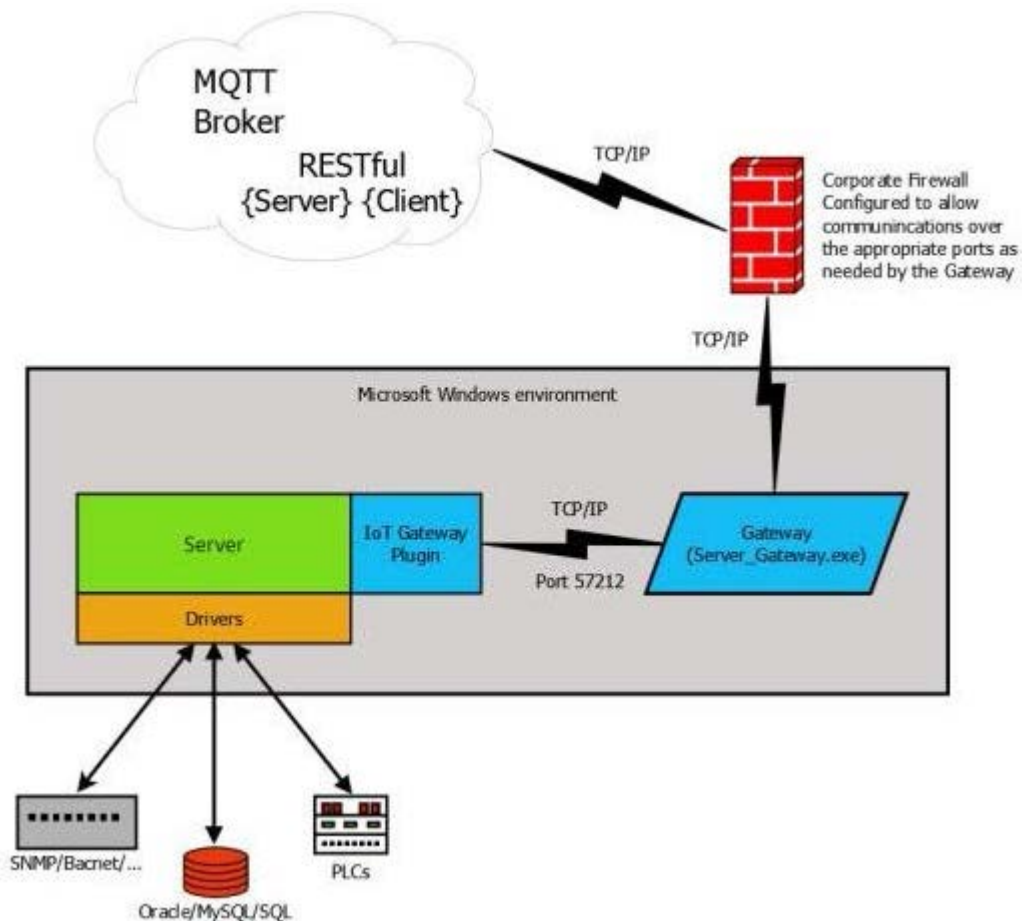
物联网网关提供以下功能：

- 1) 能够从服务器中的任何数据源（例如驱动程序、插件或系统标记）发布由名称、值、质量和时间戳组成的数据
- 2) 标准的可读 JSON 数据格式



- 3) 通过 MQTT 和 REST 客户机代理支持发布
- 4) 支持从 MQTT 客户机和 REST 服务器代理读取数据
- 5) 可配置的数据收集速率, REST 和 MQTT 客户机的频率为 10 毫秒, 最多为每 27.77 小时 (999999900 毫秒) 一次
- 6) 可配置的数据发布速率, REST 和 MQTT 客户机的频率为 10 毫秒, 最多为每 27.77 小时一次 (999999900 毫秒)
- 7) 在所有代理上支持身份验证和 TLS/SSL 加密
- 8) 支持基于用户管理器和安全策略插件的用户级访问
- 9) 用于与不同第三方端点集成的可配置头和负载信息

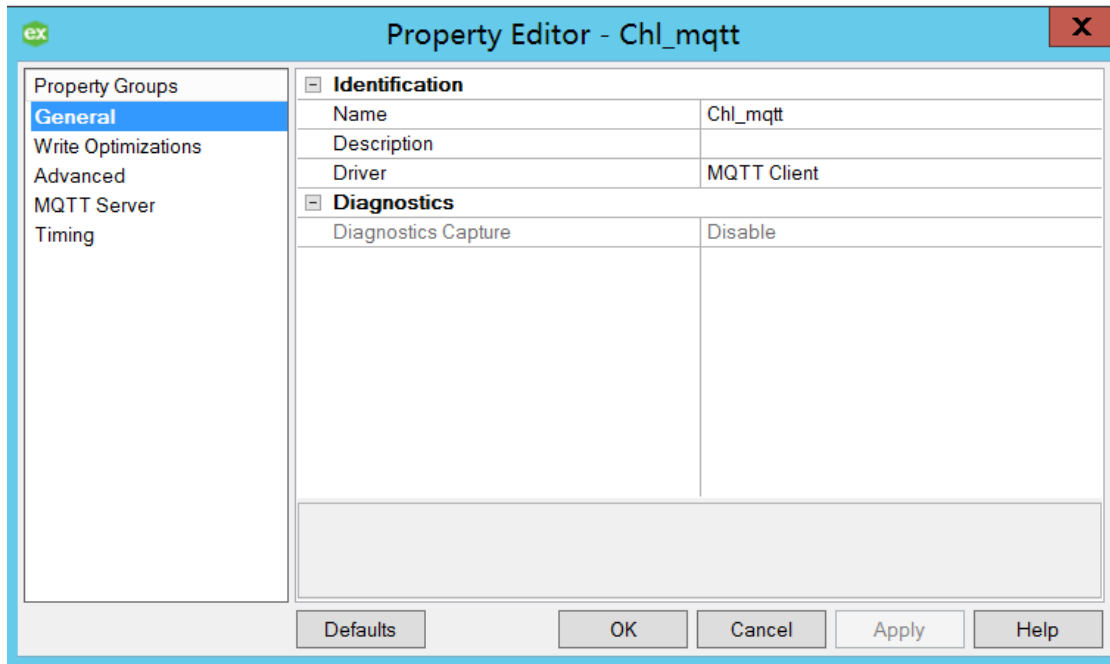
架构如下图：



3. KEPServerEX 中的 MQTT 配置

3.1. MQTT Client Driver 配置

3.1.1. 建立通道

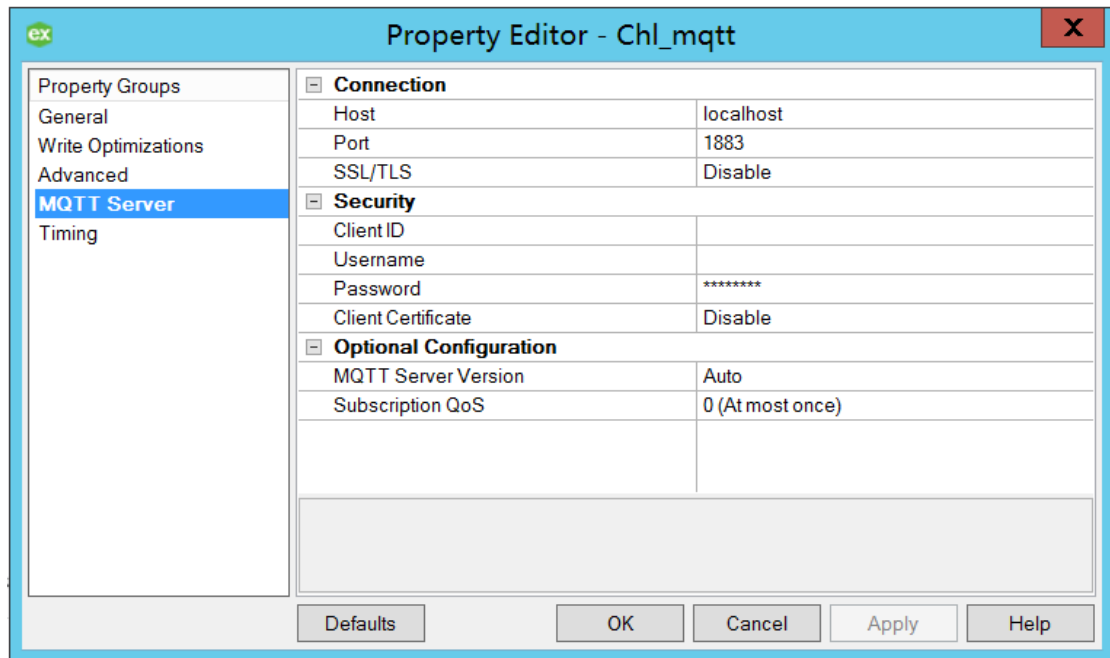


The screenshot shows the 'Property Editor - Chl_mqtt' dialog box. The 'Property Groups' list on the left includes 'General', 'Write Optimizations', 'Advanced', 'MQTT Server', and 'Timing'. The 'General' group is selected, and the 'Identification' section is expanded. The 'Diagnostics' section is also expanded.

Identification	
Name	Chl_mqtt
Description	
Driver	MQTT Client

Diagnostics	
Diagnostics Capture	Disable

Buttons at the bottom: Defaults, OK, Cancel, Apply, Help.



The screenshot shows the 'Property Editor - Chl_mqtt' dialog box. The 'Property Groups' list on the left includes 'General', 'Write Optimizations', 'Advanced', 'MQTT Server', and 'Timing'. The 'MQTT Server' group is selected, and the 'Connection', 'Security', and 'Optional Configuration' sections are expanded.

Connection	
Host	localhost
Port	1883
SSL/TLS	Disable

Security	
Client ID	
Username	
Password	*****
Client Certificate	Disable

Optional Configuration	
MQTT Server Version	Auto
Subscription QoS	0 (At most once)

Buttons at the bottom: Defaults, OK, Cancel, Apply, Help.

3.1.2. 建立设备

Identification	
Name	Dev1
Description	
Driver	MQTT Client
Model	MQTT
Channel Assignment	Chl_mqtt

Operating Mode	
Data Collection	Enable
Simulated	No

Name
Specify the identity of this object.

3.1.3. 建立标签

标记地址的格式为 mqtt_topic+payload_item。

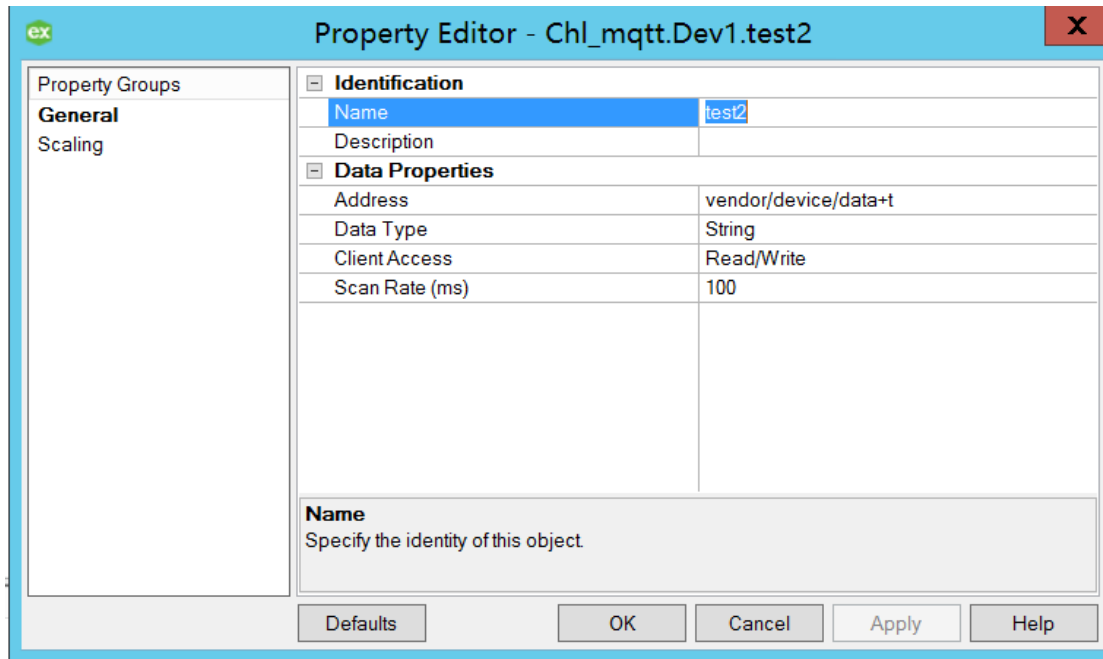
mqtt_主题是要发布所需数据值的主题。

payload_item 是有效负载中指向需要数据的特定项的路径。

Identification	
Name	test1
Description	

Data Properties	
Address	vendor/device/data+s
Data Type	String
Client Access	Read/Write
Scan Rate (ms)	100

Name
Specify the identity of this object.



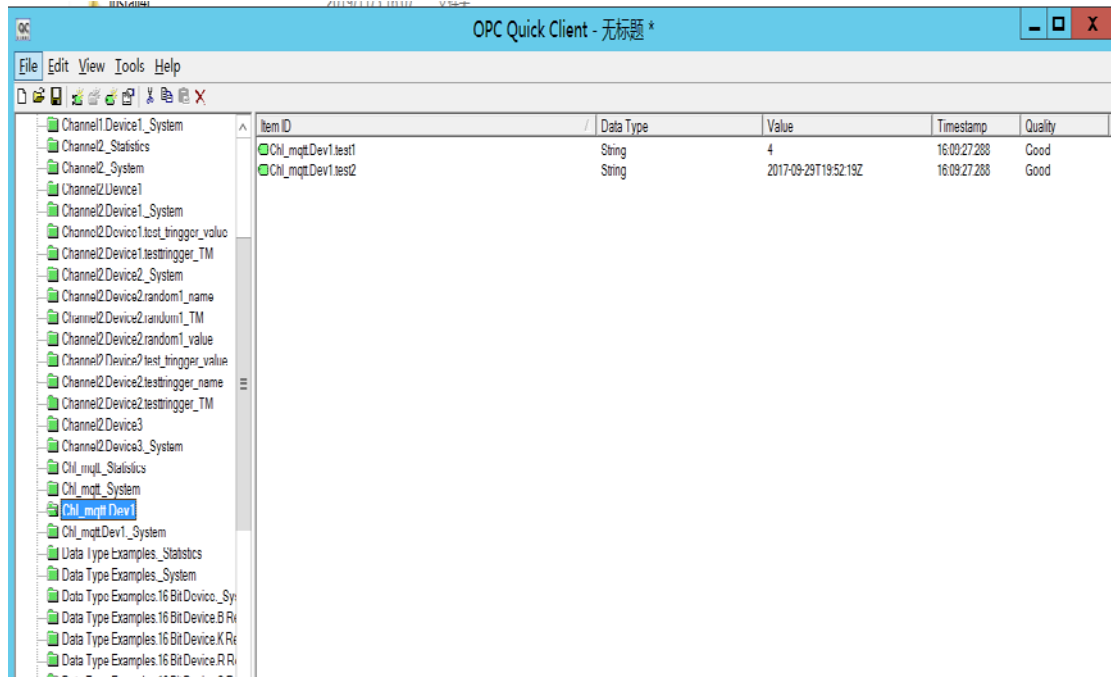
上述图例中的标签地址，接收的是如下 JSON 发送过来的数据：

Topic: vendor/device/data

Sample Payload:

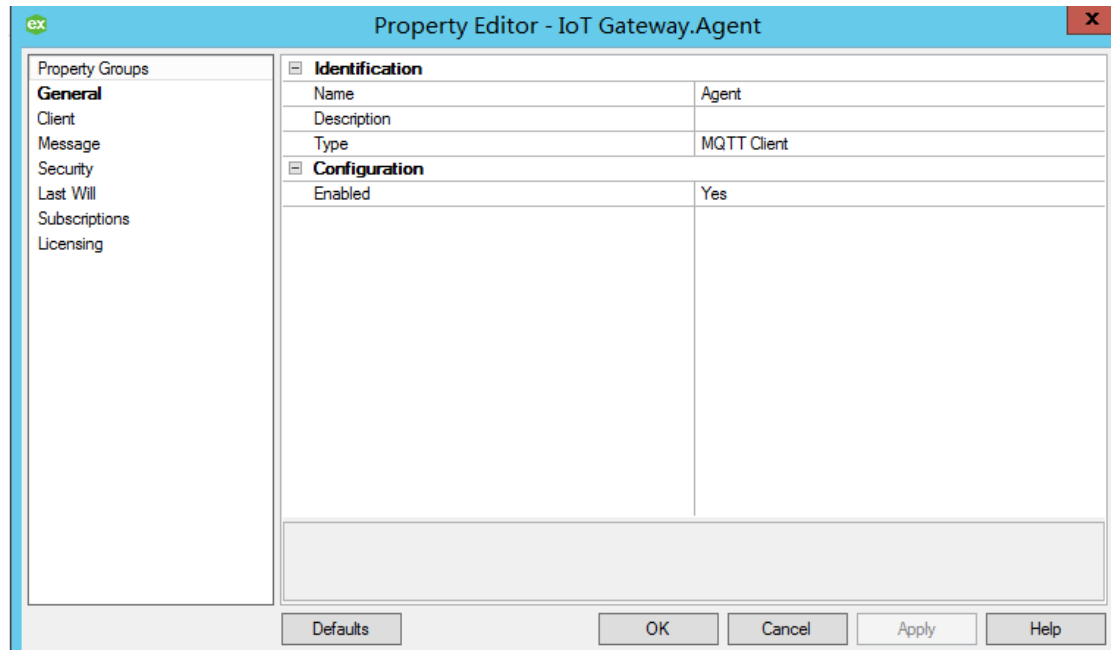
```
{
  "s":4,
  "t":"2017-09-29T19:52:19Z",
  "q":192,
  "c":6,
  "tempint":67.1,
  "vbatt":3.28,
  "ai1":8.92,
  "ai2":0.03,
  "temp1":46.4,
  "temp2":68.0
}
```

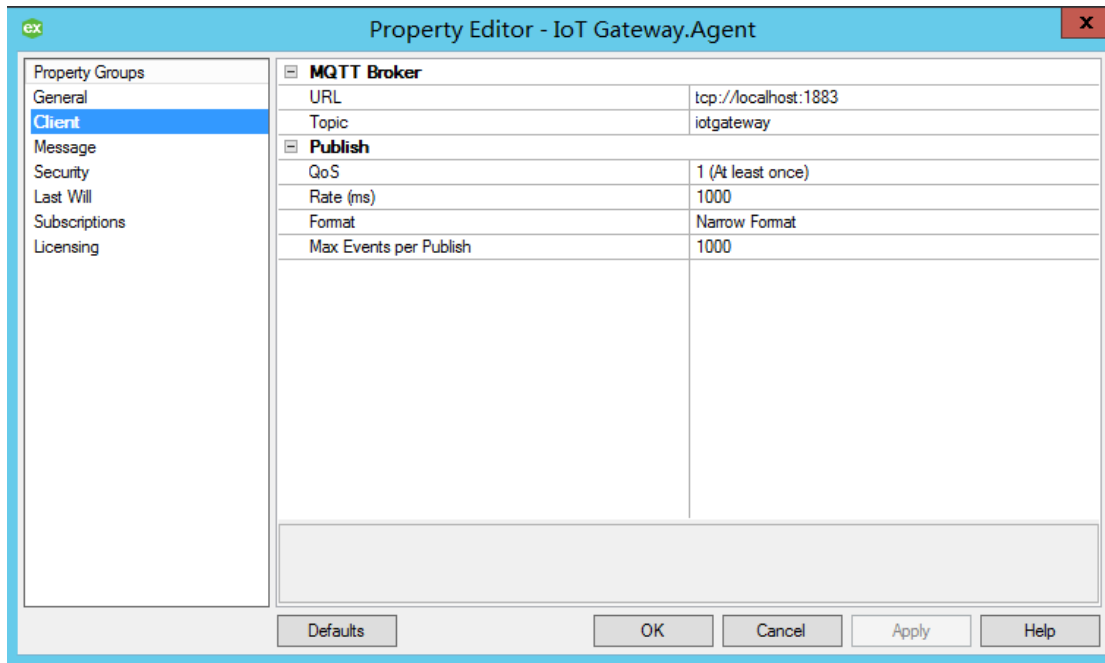

3.1.4. Quick Client 查看数据



3.2. IOT Gateway Plug-in 中的 MQTT Client 配置

3.2.1. 建立 Agent





1) QoS

QoS (Quality of Service , QoS) 为服务质量，MQTT 支持三种不同级别的服务质量为不同场景提供消息可靠性。

选项包括：0 (最多一次)、1 (至少一次)、2 (正好一次)。

2) Format

宽格式：生成一个输出，该输出包含 Agent 中所有启用的标记。此格式保证每次发布时数据格式一致。宽格式只发送每个标记的最新值，没有数据缓存。如果在使用宽格式时发布失败，则下一次发布是每个标记的最新扫描值。

窄格式：根据已更改值或质量的标记生成输出。此格式有缓存数据，并将所有标记数据更改发布到 Endpoint。

3.2.2. 建立 Item

新建 Item 时，可以通过 Tag Browser 界面浏览选择驱动、插件的标签和系统标签。

The image displays three screenshots from the KEPServerEX software interface:

- Tag Browser:** A window showing a tree view of tags on the left and a table on the right. The table has columns for 'Tag...', 'Data Ty...', and 'Description'. The tree view includes folders like Scheduler, _SNMP Agent, _System, ThingWorx, 5H, 6032, 632, Channel1, test, 模拟器示例, and _Statistics.
- Project Tree:** A tree view on the left showing project components like Connectivity, Aliases, Advanced Tags, Alarms & Events, Data Logger, IoT Gateway, 5KHS, 5KHS1, and Agent.
- Server Tag List:** A table listing server tags and their states:

Server Tag	State
模拟器示例函数.Ramp7	Enabled
模拟器示例函数.Ramp8	Enabled
模拟器示例函数.Random1	Enabled
模拟器示例函数.Random2	Enabled
- IoT Item Configuration Dialog:** A dialog box for configuring an IoT item. It includes:
 - Server Tag: 模拟器示例函数.Ramp7
 - Scan Rate (ms): 1000
 - Publish options: Only on Data Changes (with Deadband (%) set to 0) and Every scan.
 - Enabled checkbox:
 - Buttons: OK, Cancel, Help.

Item 属性里有两种发布方式。

1) Only on Data Changes(仅在数据更改时)：仅在值更改时发布此标记的数据。

死区：定义触发发布的更改阈值的值更改百分比。

2) 每次扫描：这将强制将此标签的数据发布到 Endpoint，即使标签值没有更改。

注意：质量为“坏”的标记发送一个具有该质量的更新，然后在质量恢复为“好”之前不发送更新。

