

# Allen-Bradley ControlLogix Unsolicited 驱 动程序

© 2018, PTC Inc. 保留所有权利。

# 目录

|  |           |
|--|-----------|
| <b>Allen-Bradley ControlLogix Unsolicited 驱动程序</b> ..... | <b>1</b>  |
| <b>目录</b> .....  | <b>2</b>  |
| Allen-Bradley ControlLogix Unsolicited 驱动程序 .....        | 4         |
| <b>概述</b> .....  | <b>5</b>  |
| <b>设置</b> .....  | <b>6</b>  |
| <b>通道设置</b> .....  | <b>6</b>  |
| 信道属性 - 常规 .....  | 6         |
| 信道属性 - 以太网通信 .....                                       | 7         |
| 信道属性 - 写入优化 .....  | 7         |
| 信道属性 - 高级 .....  | 8         |
| 信道属性 - EtherNet/IP 模块 .....                              | 8         |
| 主设备配置 .....  | 10        |
| <b>设备设置</b> .....  | <b>10</b> |
| 设备属性 - 常规 .....  | 10        |
| 设备属性 - 扫描模式 .....  | 12        |
| 设备属性 - 标记生成 .....  | 12        |
| 设备属性 - 控制器模块 .....                                       | 13        |
| 设备属性 - 原生标记数据库 .....                                     | 14        |
| 标记层次 .....   | 15        |
| 原生标记数据库 CSV 导入 .....                                     | 17        |
| 设备属性 - 选项 .....  | 18        |
| <b>数据类型说明</b> .....                                      | <b>20</b> |
| <b>地址说明</b> .....  | <b>21</b> |
| 地址格式 .....   | 21        |
| 标记范围 .....   | 22        |
| 高级寻址 .....   | 22        |
| <b>错误代码</b> .....  | <b>25</b> |
| 以太网/IP 封装错误代码 .....                                      | 25        |
| CIP 错误代码 .....   | 25        |
| 0x01 扩展错误代码 .....  | 25        |
| 0xFF 扩展错误代码 .....  | 26        |
| <b>事件日志消息</b> .....                                      | <b>27</b> |
| 导入原生标记数据库时出错。无法打开文件。  OS 错误 = '<错误>'。 .....              | 27        |
| 导入原生标记数据库时出错。无法打开文件，常规读取失败。 .....                        | 27        |
| 导入原生标记数据库时出错。不支持文件编码。 .....                              | 27        |
| 导入原生标记数据库时出错。 .....                                      | 27        |
| 导入原生标记数据库时出错，字段名称未识别。  未识别字段名称 = '<字段名称>'。 .....         | 27        |
| 导入原生标记数据库时出错，字段名称重复。  重复字段名称 = '<字段>'。 .....             | 28        |
| 导入原生标记数据库时出错。标记字段标识记录缺失。 .....                           | 28        |
| 导入原生标记数据库时出错。标记字段标识记录不完整。 .....                          | 28        |
| 无法启动未经请求的 Logix 服务器。 .....                               | 28        |

|   |           |
|---|-----------|
| 标记地址无效。未导入原生标记。  无效地址 = '<地址>'。 .....   | 28        |
| 已将两个通道配置为使用相同的网络适配器 IP 和 TCP 端口。必须将每个通道绑定至唯一的本地 IP 和端口。  第一个通道 = '<通道>', 第二个通道 = '<通道>'。 .....            | 29        |
| 已将两台设备配置为使用来自 EtherNet/IP 模块的相同路径。每个通道必须具有唯一的来自 EtherNet/IP 模块的路径。  第一台设备 = '<地址>', 第二台设备 = '<地址>'。 ..... | 29        |
| 标记地址无效。不允许重复的标记地址。  无效地址 = '<地址>'。 .....  | 29        |
| 无法为标记分配内存。  标记地址 = '<地址>'。 .....  | 29        |
| 原生标记无效。单个标记大小限制为 128 kB。  标记地址 = '<地址>'。 .....  | 29        |
| 导入原生标记时出错。总数据库标记数据大小限制为 128 kB。  标记地址 = '<地址>'。 .....   | 30        |
| 指定的 TCP/IP 端口超出范围。正在使用默认端口。  有效范围 = <数字> 到 <数字>, 默认端口 = <数字>。 .....                                       | 30        |
| 通道内的另一台设备已注册为本地 CPU。  设备 = <设备>。 .....  | 30        |
| 原生标记已导入。  标记计数 = <计数>, 标记数据库路径 = '<路径>'。 .....  | 30        |
| 标记已生成。  标记计数 = <计数>, 标记层次模式 = '<模式>'。 .....   | 30        |
| 由于资源不足, 因此无法执行自动标记生成。 .....   | 30        |
| <b>索引</b> .....   | <b>31</b> |

## Allen-Bradley ControlLogix Unsolicited 驱动程序

---

帮助版本 1.022

### 目录

#### 概述

什么是 Allen-Bradley ControlLogix Unsolicited 驱动程序？

#### 通道设置

如何指定通道的唯一 TCP/IP 端口？

#### 设备设置

如何配置使用此驱动程序的设备？

#### 数据类型说明

此驱动程序支持哪些数据类型？

#### 地址说明

如何对 Allen-Bradley ControlLogix 设备上的标记进行寻址？

#### 自动标记数据库生成

如何配置 Allen-Bradley ControlLogix Unsolicited 驱动程序的标记？

#### 错误代码

Allen Bradley ControlLogix 错误代码是什么？

#### 事件日志消息

此驱动程序会产生哪些消息？

## 概述

---

Allen-Bradley ControlLogix Unsolicited 驱动程序 提供将 Allen-Bradley ControlLogix PLC 连接至客户端应用程序的可靠方式。它会模拟包含单个 EtherNet/IP 模块和多达 16 个 ControlLogix CPU 的 ControlLogix 5000 系列机架。ControlLogix 5000 系列 PLC 可配置为向带有 MSG 阶梯指令的驱动程序执行 CIP 数据表读取/写入操作。

● 有关配置 ControlLogix 5000 系列 PLC 与驱动程序进行通信的详细信息，请参阅 *Rockwell/Allen Bradley* 文档。

## 设置

Allen-Bradley ControlLogix Unsolicited 驱动程序用作包含单个 Ethernet/IP 模块的模拟 ControlLogix 5000 系列机架。机架最多可包含 16 个 ControlLogix CPU，其中一个构建到 EtherNet/IP 模块 (视为本地)，最多 15 个构建到单个 CPU 模块 (视为远程 EtherNet/IP 模块)。模拟 EtherNet/IP 模块最多可随时连接 256 个设备。

### 支持的设备

所有支持“CIP 数据表读/写 MSG 说明”并运行“固件修订版本 16 或更高版本”的 ControlLogix 5000 系列 PLC。同时支持已连接和未连接的 CIP 数据表的读取或写入操作。

### 通信协议

EtherNet/IP

### PLC 配置

必须对网络上的设备进行编程，以在驱动程序中使用“CIP 数据表读/写 MSG 说明”，并处理返回的数据。

● 有关配置 MSG 说明的详细信息，请参阅“ControlLogix 系统”中的“Allen-Bradley 编程消息”。

### 套接字

最多可同时运行 256 个传入连接。连接保持打开状态，直到由发起者将其关闭。

### ●另请参阅：

[通道设置](#)

[设备设置](#)

## 通道设置

信道设置包括以下属性组的配置：

[常规](#)

[以太网通信](#)

[写入优化](#)

[高级](#)

[EtherNet/IP 模块](#)

### 信道属性 - 常规

此服务器支持同时使用多个通信驱动程序。服务器项目中使用的各个协议或驱动程序称为信道。服务器项目可以由具有相同通信驱动程序或具有唯一通信驱动程序的多个信道组成。信道充当 OPC 链路的基础构建块。此组用于指定常规信道属性，如标识属性和操作模式。

| 属性组       | 标识        |           |
|-----------|-----------|-----------|
| <b>常规</b> | 名称        | 通道 1      |
| 写优化       | 说明        |           |
| 高级        | 驱动程序      | Simulator |
| 持久存储      | <b>诊断</b> |           |
|           | 诊断数据捕获    | 禁用        |

### “标识”

**“名称”**：此信道的用户定义标识。在每个服务器项目中，每个信道名称都必须是唯一的。尽管名称最多可包含 256 个字符，但在浏览 OPC 服务器的标记空间时，一些客户端应用程序的显示窗口可能不够大。信道名称是 OPC 浏览器信息的一部分。

● 有关保留字符的信息，请参阅服务器帮助中的“如何正确命名信道、设备、标记和标记组”。

**“说明”**：有关此信道的用户定义信息。

● 这些属性 (包括 Description) 当中有很多具有关联的系统标记。

**“驱动程序”(Driver):** 为该信道选择的协议/驱动程序。该属性指定在信道创建期间选择的设备驱动程序。它在信道属性中为禁用设置。

● **注意:** 服务器全天在线运行时,可以随时更改这些属性。其中包括更改信道名称以防止客户端向服务器注册数据。如果客户端在信道名称更改之前已从服务器中获取了项,那么这些项不会受到任何影响。如果客户端应用程序在信道名称更改之后发布项,并尝试通过原来的信道名称重新获取项,则该项将不被接受。考虑到这一点,一旦开发完成大型客户端应用程序,就不应对属性进行任何更改。利用“用户管理器”可防止操作员更改属性并限制对服务器功能的访问权限。

## 诊断

**“诊断数据捕获”(Diagnostics Capture):** 启用此选项后,信道的诊断信息即可用于 OPC 应用程序。由于服务器的诊断功能所需的开销处理量最少,因此建议在需要时使用这些功能,而在不需要时禁用这些功能。默认设置为禁用状态。

● **注意:** 如果驱动程序不支持诊断,则该属性将被禁用。

● 有关详细信息,请参阅服务器帮助中的“通信诊断”。

## 通道属性 - 以太网通信

以太网通信可用于与设备进行通信。

|       |       |     |
|-------|-------|-----|
| 属性组   | 以太网设置 |     |
| 常规    | 网络适配器 | 默认值 |
| 以太网通信 |       |     |
| 写优化   |       |     |
| 高级    |       |     |
| 通信序列化 |       |     |

### 以太网设置

**“网络适配器”(Network Adapter):** 指定要绑定的网络适配器。如果选择“默认”(Default),则操作系统将选择默认适配器。

## 通道属性 - 写入优化

与任何 OPC 服务器一样,将数据写入设备可能是应用程序应具备的最重要的功能。服务器旨在确保从客户端应用程序写入的数据能够准时发送到设备。为了达到此目标,服务器提供了可用来满足特定需求以提高应用程序响应能力的优化属性。

|      |      |             |
|------|------|-------------|
| 属性组  | 写优化  |             |
| 常规   | 优化方法 | 仅写入所有标记的最新值 |
| 写优化  | 占空比  | 10          |
| 高级   |      |             |
| 持久存储 |      |             |

### 写入优化

**“优化方法”(Optimization Method):** 控制如何将写入数据传递至底层通信驱动程序。选项包括:

- **“写入所有标记的所有值”(Write All Values for All Tags):** 此选项可强制服务器尝试将每个值均写入控制器。在此模式下,服务器将持续收集写入请求并将它们添加到服务器的内部写入队列。服务器将对写入队列进行处理并尝试通过将数据尽快写入设备来将其清空。此模式可确保从客户端应用程序写入的所有数据均可发送至目标设备。如果写入操作顺序或写入项的内容必须且仅能显示于目标设备上,则应选择此模式。
- **“写入非布尔标记的最新值”(Write Only Latest Value for Non-Boolean Tags):** 由于将数据实际发送至设备需要一段时间,因此对同一个值的多次连续写入会存留于写入队列中。如果服务器要更新已位于写入队列中的某个写入值,则需要大大减少写入操作才能获得相同的最终输出值。这样一来,便不

会再有额外的写入数据存留于服务器队列中。几乎就在用户停止移动滑动开关时，设备中的值达到其正确值。根据此模式的规定，任何非布尔值都会在服务器的内部写入队列中更新，并在下一个可能的时机发送至设备。这可以大大提高应用性能。

● **注意**：该选项不会尝试优化布尔值的写入。它允许用户在不影响布尔运算的情况下优化 HMI 数据的操作，例如瞬时型按钮等。

- **“写入所有标记的最新值”(Write Only Latest Value for All Tags)**：该选项采用的是第二优化模式背后的理论并将其应用至所有标记。如果应用程序只需向设备发送最新值，则该选项尤为适用。此模式会通过当前写入队列中的标记发送前对其进行更新来优化所有的写入操作。此为默认模式。

**“占空比”(Duty Cycle)**：用于控制写操作与读操作的比率。该比率始终基于每一到十次写入操作对应一次读取操作。占空比的默认设置为 10，这意味着每次读取操作对应十次写入操作。即使在应用程序执行大量的连续写入操作时，也必须确保足够的读取数据处理时间。如果将占空比设置为 1，则每次读取操作对应一次写入操作。如果未执行任何写入操作，则会连续处理读取操作。相对于更加均衡的读写数据流而言，该特点使得应用程序的优化可通过连续的写入操作来实现。

● **注意**：建议在将应用程序投入生产环境前使其与写入优化增强功能相兼容。

## 通道属性 - 高级

此组用于指定高级信道属性。并非所有驱动程序都支持所有属性，因此不会针对不支持的设备显示“高级”组。

|           |  |      |
|-----------|--|------|
| 属性组       | <input type="checkbox"/> <b>非规范浮点数处理</b> |      |
| 常规        | 浮点值                                      | 替换为零 |
| 以太网通信     | <input type="checkbox"/> <b>设备间延迟</b>    |      |
| 写优化       | 设备间延迟 (毫秒)                               | 0    |
| <b>高级</b> |  |      |
| 通信序列化     |  |      |

**“非规范浮点数处理”**：非规范值定义为无穷大、非数字 (NaN) 或不正规编号。默认值为“替换为零”。具有原生浮点数处理功能的驱动程序可能会默认设置为“未修改”。通过非规范浮点数处理，用户可以指定驱动程序处理非规范 IEEE-754 浮点数据的方式。选项说明如下：

- **“替换为零”**：此选项允许驱动程序在将非规范 IEEE-754 浮点值传输到客户端之前，将其替换为零。
- **“未修改”**：此选项允许驱动程序向客户端传输 IEEE-754 不正规、规范、非数字和无穷大值，而不进行任何转换或更改。

● **注意**：如果驱动程序不支持浮点值或仅支持所显示的选项，则将禁用此属性。根据信道的浮点规范化设置，将仅对实时驱动程序标记 (如值和数组) 进行浮点规范化。例如，此设置不会影响 EFM 数据。

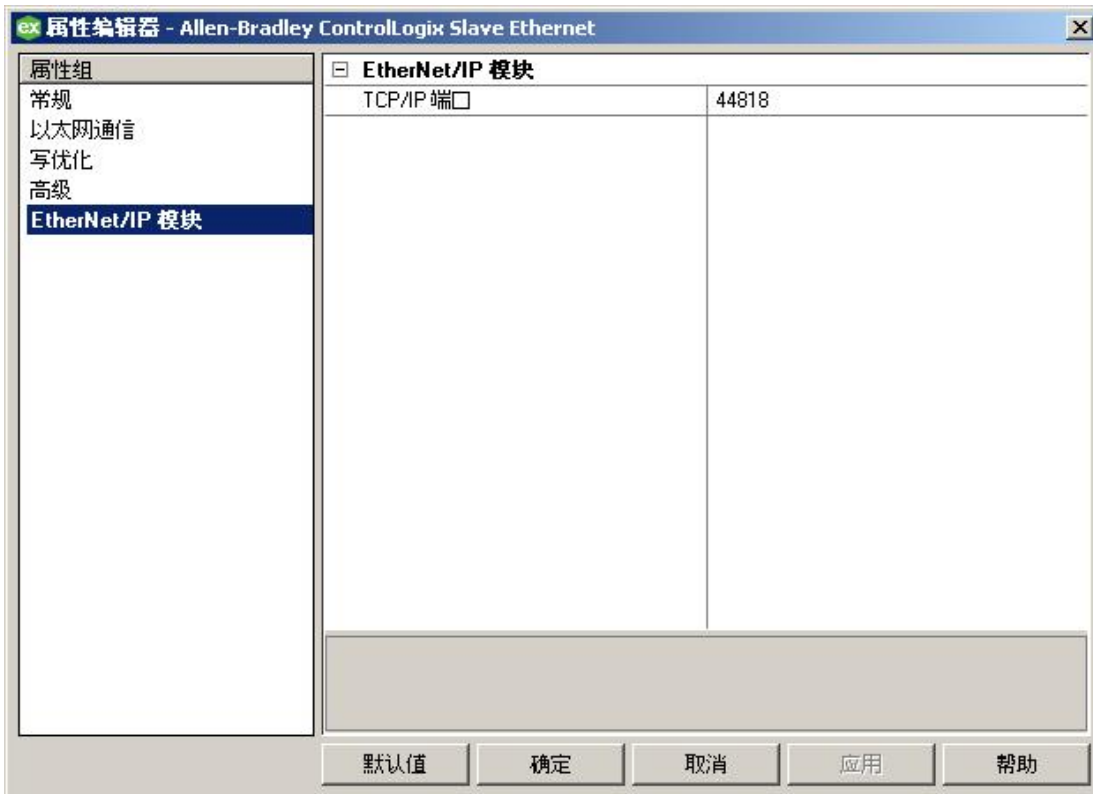
● 有关浮点值的详细信息，请参阅服务器帮助中的“如何使用非规范化浮点值”。

**“设备间延迟”**：指定在接收到同一信道上的当前设备发出的数据后，通信信道向下一设备发送新请求前等待的时间。设置为零 (0) 将禁用延迟。

● **注意**：此属性并不适用于所有驱动程序、型号和相关设置。

## 信道属性 - EtherNet/IP 模块





**“TCP/IP 端口”**: 用于指定为 EtherNet/IP 模块提供唯一通信信道的 TCP/IP 和 UDP 端口。有效范围为 1 到 65535。默认值为 44818。

- **注意**: Allen-Bradley ControlLogix Unsolicited 驱动程序 当前将通道数限制为一个。如果网络适配器和端口与主机上的另一个应用程序冲突, 则驱动程序无法接受入站 EtherNet/IP 连接。
- 有关详细信息, 请参阅 [事件日志消息](#)。

## 主设备配置

---

必须对 Allen-Bradley ControlLogix PLC 进行编程，以使用“MSG 梯形图指令”向此驱动程序发出“CIP 数据表读/写”消息。应使用代表驱动程序配置的路由路径，其中包括 IP 地址、插槽编号及可选端口。有关“MSG 梯形图指令”的详细信息，请参阅 *Rockwell/Allen-Bradley PLC 编程文档*。在“设备属性”下的“控制器模块”中，提供了与特定从属设备相关联的路由路径。有关详细信息，请参阅[设备属性 - 控制器模块](#)。

### 支持的服务

非零碎读取  
零碎读取  
非零碎写入  
零碎写入  
读取/修改/写入

● **注意：**“ControlLogix MSG 梯形图指令”会根据请求大小自动决定使用是“零碎”还是“非零碎”服务。用户不可配置此选项。

### 支持的 Logix 类型

BOOL  
DWORD (BOOL 数组)  
SINT  
INT  
DINT  
LINT  
REAL

### 错误代码

Allen-Bradley ControlLogix Unsolicited 驱动程序 会对接收到的所有格式正确的消息作出响应。如果无法完成请求，会返回响应消息，其中在 MESSAGE 结构的 ERR 及 EXERR 标记中会包含一个非零错误状态和一个可选扩展错误状态。应编写梯形图程序处理这些错误。

● 有关可能返回到主设备的错误代码的详细信息，请参阅[错误代码](#)。

● **注意：**此驱动程序支持上面所列“Logix 原子类型”的“CIP 数据表读/写”。虽然不支持结构化类型，但是可使用“MSG 梯形图指令”在结构化类型中对单个“Logix 原子类型”进行读/写。例如，如果必须将标记“MyString @STRING”写入驱动程序，必须对“MyString.DATA”和“MyString.LEN”分别执行“CIP 数据表读”操作。

## 设备设置

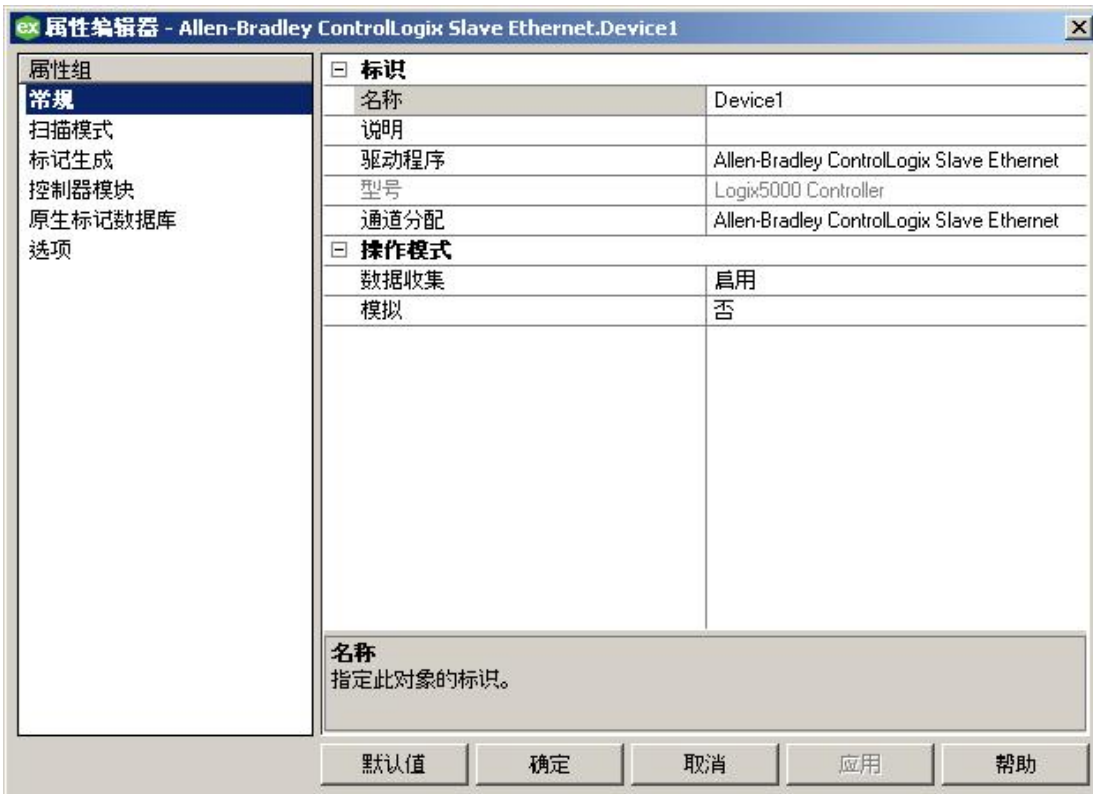
---

设备设置包括以下属性组的配置：

[常规](#)  
[扫描模式](#)  
[自动降级](#)  
[控制器模块](#)  
[原生标记数据库](#)  
[选件](#)

### 设备属性 - 常规

---



## 标识

**“名称”**: 此设备的用户定义标识。

**“说明”**: 有关此设备的用户定义信息。

**“通道分配”**: 该设备当前所属通道的用户定义名称。

**驱动程序**: 为该设备选择的协议驱动程序。

**“型号”**: 设备的特定版本。

## 操作模式

**“数据收集”**: 此属性控制设备的活动状态。尽管默认情况下会启用设备通信，但可使用此属性禁用物理设备。设备处于禁用状态时，不会尝试进行通信。从客户端的角度来看，数据将标记为无效，且不接受写入操作。通过此属性或设备系统标记可随时更改此属性。

**“模拟”**: 此选项可将设备置于模拟模式。在此模式下，驱动程序不会尝试与物理设备进行通信，但服务器将继续返回有效的 OPC 数据。模拟停止与设备的物理通信，但允许 OPC 数据作为有效数据返回到 OPC 客户端。在“模拟模式”下，服务器将所有设备数据处理为反射型：无论向模拟设备写入什么内容，都会读取回来，而且会单独处理每个 OPC 项。项的内存映射取决于组更新速率。如果服务器移除了项（如服务器重新初始化时），则不保存数据。默认值为“否”。

### ● 注意：

1. “系统”标记 (`_Simulated`) 为只读且无法写入，从而达到运行时保护的目。 “系统”标记允许从客户端监控此属性。
2. 在“模拟”模式下，项的内存映射取决于客户端更新速率 (OPC 客户端的“组更新速率”或本机和 DDE 接口的扫描速率)。这意味着，参考相同项、而采用不同更新速率的两个客户端会返回不同的数据。

● “模拟模式”仅用于测试和模拟目的。该模式永远不能用于生产环境。

## 设备属性 - 扫描模式

“扫描模式”为需要设备通信的标记指定订阅客户端请求的扫描速率。同步和异步设备的读取和写入会尽快处理；不受“扫描模式”属性的影响。

|      |           |              |
|------|-----------|--------------|
| 属性组  | ☐ 扫描模式    |              |
| 常规   | 扫描模式      | 遵循客户端指定的扫描速率 |
| 扫描模式 | 来自缓存的初始更新 | 禁用           |
| 定时   |           |              |

“扫描模式”：为发送到订阅客户端的更新指定在设备中扫描标记的方式。选项说明如下：

- “遵循客户端指定的扫描速率”：此模式可使用客户端请求的扫描速率。
- “不超过扫描速率请求数据”：此模式可指定要使用的最大扫描速率。有效范围为 10 至 99999990 毫秒。默认值为 1000 毫秒。  
 ● 注意：当服务器有活动的客户端和设备项目且扫描速率值有所提高时，更改会立即生效。当扫描速率值减小时，只有所有客户端应用程序都断开连接，更改才会生效。
- “以扫描速率请求所有数据”：此模式将以订阅客户端的指定速率强制扫描标记。有效范围为 10 至 99999990 毫秒。默认值为 1000 毫秒。
- “不扫描，仅按需求轮询”：此模式不会定期轮询属于设备的标签，也不会在一个项变为活动状态后为获得项的初始值而执行读取操作。客户端负责轮询以便更新，方法为写入 `_DemandPoll` 标记或为各项发出显式设备读取。有关详细信息，请参阅服务器帮助中的“设备需求轮询”。
- “遵循标签指定的扫描速率”：此模式将以静态配置标记属性中指定的速率强制扫描静态标记。以客户端指定的扫描速率扫描动态标记。

“来自缓存的初始更新”：启用后，此选项允许服务器为存储（缓存）数据的新激活标签参考提供第一批更新。只有新项参考共用相同的地址、扫描速率、数据类型、客户端访问和缩放属性时，才能提供缓存更新。设备读取仅用于第一个客户端参考的初始更新。默认设置为禁用；只要客户端激活标记参考，服务器就会尝试从设备读取初始值。

## 设备属性 - 标记生成

自动标记数据库生成功能使设置应用程序成为一项即插即用操作。选择可以配置为自动构建标记列表的通信驱动程序（标记与特定于设备的数据相对应）。可以从客户端浏览这些自动生成的标记（这取决于支持驱动程序的性质）。

●并非所有设备和驱动程序都支持全自动标记数据库生成，也并非所有都支持相同的数据类型。有关具体信息，请参阅各驱动程序的数据类型说明或支持的数据类型列表。

如果目标设备支持其自身的本地标记数据库，则驱动程序会读取设备的标记信息，并使用该数据来在服务器中生成标记。如果该设备本身不支持已命名的标记，则驱动程序会根据特定于驱动程序的信息来创建标记列表。这两个条件的示例如下：

1. 如果数据采集系统支持其自身的本地标记数据库，则通信驱动程序将使用在设备中发现的标记名称来构建服务器的标记。
2. 如果以太网 I/O 系统支持其自身可用 I/O 模块类型的检测，则通信驱动程序会基于插入以太网 I/O 机架的 I/O 模块类型在服务器中自动生成标记。

●注意：自动标记数据库生成的操作模式可进行完全配置。有关详细信息，请参阅下方的属性说明。

|      |           |        |
|------|-----------|--------|
| 属性组  | ☐ 标记生成    |        |
| 常规   | 设备启动时     | 启动时不生成 |
| 扫描模式 | 对于重复标记    | 创建时删除  |
| 定时   | 父组        |        |
| 自动降级 | 允许自动生成的子组 | 启用     |
| 标记生成 | 创建        | 创建标记   |
| 冗余   |           |        |

**“属性更改时”**: 如果设备支持在特定属性更改时自动生成标记, 系统会显示**“属性更改时”**选项。默认情况下, 该选项设置为**“是”**, 但可以将其设置为**“否”**来控制何时生成标记。在此情况下, 必须手动执行**“创建标记”**操作以执行标记生成。

**“设备启动时”**: 此属性指定何时自动生成 OPC 标记。选项说明如下:

- **“启动时不生成”**: 此选项可防止驱动程序向服务器的标记空间添加任何 OPC 标记。这是默认设置。
  - **“始终在启动时生成”**: 此选项可使驱动程序评估设备, 以便获得标记信息。每次启动服务器时, 它还会向服务器的标记空间添加标记。
  - **“首次启动时生成”**: 此选项可使驱动程序在首次运行项目时评估目标设备, 以便获得标记信息。它还可以根据需要向服务器标记空间添加任何 OPC 标记。
- **注意**: 如果选择自动生成 OPC 标记的选项, 添加到服务器标记空间的任何标记都必须随项目保存。用户可以在**“工具”|“选项”**菜单中将项目配置为自动保存。

**“对于重复标记”**: 启用自动标记数据库生成后, 服务器需要了解如何处理先前已添加的标记, 或在初始创建通信驱动程序后添加或修改的标记。此设置可控制服务器处理自动生成的以及当前存在于项目中的 OPC 标记的方式。它还可以防止自动生成的标记在服务器中累积。

例如, 如果用户更改机架中的 I/O 模块, 并且服务器配置为**“始终在启动时生成”**, 则每当通信驱动程序检测到新的 I/O 模块时, 新标记就会添加到服务器。如果未移除旧标记, 则许多未使用的标记可能会在服务器的标记空间中累积。选项包括:

- **“创建时删除”**: 此选项可在添加任何新标记之前, 将先前添加到标记空间的任何标记删除。这是默认设置。
  - **“根据需要覆盖”**: 此选项可以指示服务器仅移除通信驱动程序要用新标记替换掉的标记。所有未被覆盖的标记仍将保留在服务器的标记空间中。
  - **“不覆盖”**: 此选项可以防止服务器移除任何之前生成的标记或服务器中已存在的标记。通信驱动程序只能添加全新的标记。
  - **“不覆盖, 记录错误”**: 此选项与前一选项有相同效果, 并且在发生标记覆盖时, 也会将错误消息发布到服务器的事件日志。
- **注意**: 删除 OPC 标记会影响通信驱动程序已自动生成的标记以及使用匹配已生成标记的名称添加的任何标记。如果标记所使用的名称可能与驱动程序自动生成的标记相匹配, 则用户应避免将此类标记添加到服务器。

**“父组”**: 此属性通过指定将要用于自动生成标记的组, 来防止自动生成的标记与已手动输入的标记发生混淆。组名称最多可包含 256 个字符。此父组具有一个根分支, 可将所有自动生成的标记添加到其中。

**“允许自动生成的子组”**: 此属性用于控制服务器是否为自动生成的标记自动创建子组。这是默认设置。如果禁用, 则服务器会在没有任何分组的简单列表中生成设备标记。在服务器项目中, 生成的标记使用地址值命名。例如, 生成过程中不会保留标记名称。

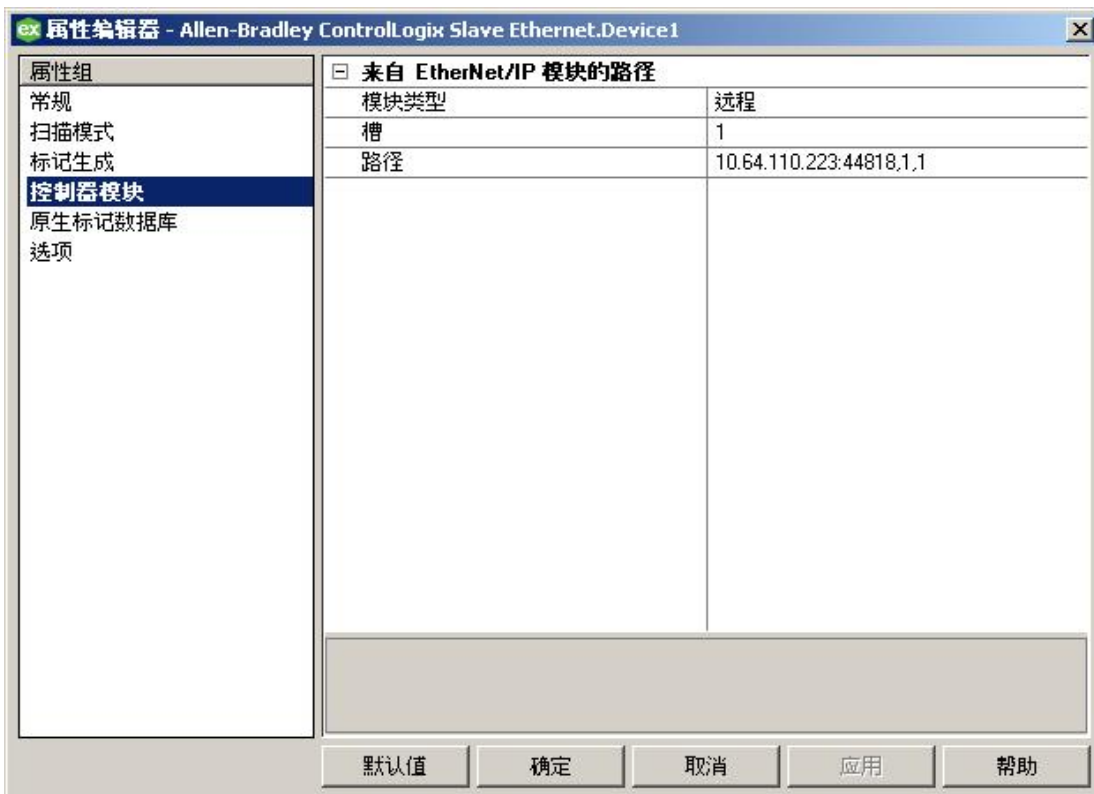
● **注意**: 如果在服务器生成标记的过程中, 分配给标记的名称与现有标记的名称相同, 则系统会自动递增到下一个最高数字, 以免标记名称发生重复。例如, 如果生成过程中创建了名为 **“AI22”** 的标记且该名称已存在, 则会将标记创建为 **“AI23”**。

**“创建”**: 开始创建自动生成的 OPC 标记。如果已修改设备的配置, 则**“创建标记”**可强制驱动程序重新评估设备以发现可能的标记更改。由于该选项可以通过系统标记进行访问, 这使得客户端应用程序能够启动标记数据库创建。

● **注意**: 当**“配置”**对项目进行离线编辑时, 会禁用**“创建标记”**。

## 设备属性 - 控制器模块

---



**“模块类型”**: 用于验证设备为“本地”(模拟 EtherNet/IP 模块的一部分) 还是“远程”(这需要为 EtherNet/IP 路由提供一个插槽编号)。可以存在一个本地 CPU 和最多 15 个远程 CPU。

- **“本地”**: 若为“本地”，“控制器模块”会被视为模拟 EtherNet/IP 模块的本地 CPU。每个信道只能有一个“本地控制器模块”。默认设置为已启用。
- **“远程”**: 若为“远程”，“控制器模块”会被视为独立于模拟 EtherNet/IP 模块的 CPU。每个信道最多可以有 15 个远程控制器模块。启用此选项时，还必须指定一个插槽。默认设置为禁用状态。

● **注意:**

1. 必须将每个从属设备配置为表示 ControlLogix 5000 系列控制器。
2. 如果另一台设备已配置为“本地”，则此设备将被禁用，并强制使用“远程 (1)”。

**“插槽”**: 此属性是“控制器模块”路由路径的一部分。其中仅包含当前可用于配置信道/设备的插槽。如果指定新插槽，则先前的插槽即可供另一台设备使用。

**“路径”**: 此属性表示到主设备“控制器模块”的路由路径。在 PLC 的主设备配置期间应使用此路径。

● 有关配置主设备的详细信息，请参阅[主设备配置](#)。

## 设备属性 - 原生标记数据库

Allen-Bradley ControlLogix Unsolicited 驱动程序 可配置为自动生成与原生标记数据库相对应的服务器标记列表。“原生标记数据库”标记必须是预定义的 Logix 原子型，但也可以是结构化类型的一部分。

驱动程序会为每个在“原生标记数据库”中定义的原子型标记生成一个服务器标记。对于数组类型，已为数组的各个元素定义服务器标记。数组标记可以快速增加导入标记以及服务器中可用标记的数目。自动生成的标记始终配置为具有“读/写”客户端访问权限。

● 有关详细信息，请参阅[原生标记数据库 CSV 导入](#)。





## 原生标记数据库

除了自动生成服务器标记外，“原生标记数据库”对话框还用于收集模拟的“控制器模块”所表示的数据。数据库将作为逗号分隔值 (CSV) 文件导入。

● 有关详细信息，请参阅[原生标记数据库 CSV 导入](#)。

**“导入数据库”(Import Database):** 单击“导入标记...”(Import tags...) 查找并选择用于“原生标记数据库”导入的 CSV 文件。当服务器的客户端连接仍在使用时，无法导入原生标记。

● **注意:** 导入原生标记数据库后，将使用服务器项目文件进行维护。远程部署不需要 CSV 文件。有关详细信息，请参阅自动标记数据库生成。

## 自动标记生成

**“标记层次”(Tag Hierarchy):** 选择标记显示“树”组织是否应为“已扩展”(Expanded) 或“已压缩”(Condensed)。默认值为“已扩展”(Expanded)。

- **“已扩展”(Expanded):** 生成的客户端标记的分组与 RSLogix 类似，其中包括为每个段 (标记地址后跟一个句点)、结构、子结构以及数组创建组。
- **“已压缩”(Condensed):** 对于生成的客户端标记，其分组与标记寻址类似，包括为前面带有句点的每个段创建组。

● 有关导入所需的“原生标记数据库”CSV 格式的详细信息，请参阅[原生标记数据库 CSV 导入](#)。

## 标记层次

自动生成的服务器标记可以遵循两个层次之一：“已扩展”(Expanded) 或“已压缩”(Condensed)。默认值为“已扩展模式”。

### 已扩展模式

在“已扩展模式”下，自动生成的服务器标记遵循的“组/标记”层次与 RSLogix5000 中的标记层次一致。会为每个段 (前面带有句点) 创建组，并且还会以逻辑分组的形式进行创建。创建的组包括以下内容：

- 全局 (控制器) 范围
- 结构和子结构

- 数组

● **注意:** 不会为 .bit 地址创建组。

**基本全局标记**

基本全局标记 (或非结构、非数组标记) 位于“全局”组之下。针对每个结构和数组的标记, 提供父组中其自身的子组。以此方式组织数据后, 服务器的“标记视图”会模拟 RSLogix5000。

● **注意:** “结构/数组”子组的名称同样提供说明。例如, 在控制器中定义的数组 "tag1 [1,6]" 具有一个子组名称 "tag1 [x,y]"。在此示例中, x 表示存在 1 维, 而 y 表示存在 2 维。此外, 数组子组内的标记为该数组中的元素, 除非明确限定。结构子组中的标记是结构成员本身。包含数组的结构也会创建该结构组的数组子组。

**数组标记**

为包含数组元素的每个数组创建一个组。组名称带有符号 <数组名称>[x,y,z], 其中:

- [x,y,z]: 3 维数组。
- [x,y]: 2 维数组
- [x]: 1 维数组

● **注意:** “数组标记”带有符号 <标记元素>\_XXXXX\_YYYYY\_ZZZZ。例如, 元素 "tag1[12,2,987]" 具有标记名称 "tag1\_12\_2\_987"。

**简单示例**

| Name             | Value | Force Mask | Style   | Data Type  |
|------------------|-------|------------|---------|------------|
| MyTag            | {...} | {...}      |         | MyDataType |
| MyTag.Member1    | {...} | {...}      | Decimal | DINT[10]   |
| MyTag.Member1[0] | 0     |            | Decimal | DINT       |
| MyTag.Member1[1] | 0     |            | Decimal | DINT       |
| MyTag.Member1[2] | 0     |            | Decimal | DINT       |
| MyTag.Member1[3] | 0     |            | Decimal | DINT       |

| Tag Name  | Address          |
|-----------|------------------|
| MEMBER1_0 | MYTAG.MEMBER1[0] |
| MEMBER1_1 | MYTAG.MEMBER1[1] |
| MEMBER1_2 | MYTAG.MEMBER1[2] |
| MEMBER1_3 | MYTAG.MEMBER1[3] |
| MEMBER1_4 | MYTAG.MEMBER1[4] |
| MEMBER1_5 | MYTAG.MEMBER1[5] |
| MEMBER1_6 | MYTAG.MEMBER1[6] |
| MEMBER1_7 | MYTAG.MEMBER1[7] |
| MEMBER1_8 | MYTAG.MEMBER1[8] |
| MEMBER1_9 | MYTAG.MEMBER1[9] |

**复杂示例**

使用地址 "MyStructArray[0].MySubStruct.Data" 定义的“Logix 标记”将在以下各组中表示: "Global"、"MYSTRUCTARRAY[x]"、"MYSTRUCTARRAY[0]" 和 "MYSUBSTRUCT"。标记 "DATA" 将在最后一组中。"DATA" 的静态参考为 "Channel1.Device1.Global.MYSTRUCTARRAY[x].MYSTRUCTARRAY [0].MYSUBSTRUCT.DATA"。动态参考为 "Channel1.Device1. MyStructArray[0].MySubStruct.Data"。

● 有关详细信息, 请参阅服务器帮助文件中的“静态标记 (用户定义)”和“动态标记”。

**已压缩模式**

在“已压缩模式”下, 自动生成的服务器标记遵循组/标记层次与标记的地址一致。会为每个段 (前面带有句点) 创建组。创建的组包括以下内容:

- 程序范围
- 结构和子结构

● **注意:**



1. 不会为数组或 .bit 地址创建组。
2. 以下划线开头的标记或结构成员名称会转换为 "U"。这是必需的，因为服务器不支持标记名称字段中存在前导下划线。

### 简单示例

| Name             | Value | Force Mask | Style   | Data Type  |
|------------------|-------|------------|---------|------------|
| MyTag            | {...} | {...}      |         | MyDataType |
| MyTag.Member1    | {...} | {...}      | Decimal | DINT[10]   |
| MyTag.Member1[0] | 0     |            | Decimal | DINT       |
| MyTag.Member1[1] | 0     |            | Decimal | DINT       |
| MyTag.Member1[2] | 0     |            | Decimal | DINT       |
| MyTag.Member1[3] | 0     |            | Decimal | DINT       |

| Tag Name  | Address          |
|-----------|------------------|
| MEMBER1_0 | MYTAG.MEMBER1[0] |
| MEMBER1_1 | MYTAG.MEMBER1[1] |
| MEMBER1_2 | MYTAG.MEMBER1[2] |
| MEMBER1_3 | MYTAG.MEMBER1[3] |
| MEMBER1_4 | MYTAG.MEMBER1[4] |
| MEMBER1_5 | MYTAG.MEMBER1[5] |
| MEMBER1_6 | MYTAG.MEMBER1[6] |
| MEMBER1_7 | MYTAG.MEMBER1[7] |
| MEMBER1_8 | MYTAG.MEMBER1[8] |
| MEMBER1_9 | MYTAG.MEMBER1[9] |

### 复杂示例

使用地址 "MyStructArray[0].MySubStruct.Data" 定义的“Logix 标记”将在以下各组中表示：“MYSTRUCTARRAY [0]”和“MYSUBSTRUCT”。标记“DATA”将在最后一组中。“DATA”的静态参考为“Channel1.Device1.MYSTRUCTARRAY[0].MYSUBSTRUCT.DATA”，动态参考为“Channel1.Device1.MyStructArray[0].MySubStruct.Data”。

## 原生标记数据库 CSV 导入

CSV 文件指定每个设备可以表示的原生标记。它在标记导入时会被使用一次，但在自动标记数据库生成或远程部署时不需要。必须使用以下 CSV 标题进行“原生标记”导入：

| Logix 地址 | Logix 数据类型 | 外部访问 | 说明 |
|----------|------------|------|----|
|----------|------------|------|----|

**提示：**在 <Server Installation Directory>/Drivers/controllogix\_unsolicited\_ethernet/import\_template.csv</Server> 中包含“原生标记数据库”CSV 文件模板以供参考。

### 将原生标记作为短整型、整型和双整型数组导入

作为短整型、整型和双整型导入的原生标记还具有一个已定义的字符串标记，该标记使用标记地址中相应数组的元素数目。示例如下：

- 如果导入了名为 "MySINTarray @ SINT[100]" 的原生标记，则会生成地址为 "MYSINTARRAY / 100" 的静态标记和字符串数据类型。
- 如果导入了名为 "MyINTarray @ INT[100]" 的原生标记，则会生成地址为 "MYINTARRAY / 100" 的静态标记和字符串数据类型。
- 如果导入了名为 "MyDINTarray @ DINT[100]" 的原生标记，则会生成地址为 "MYDINTARRAY / 100" 的静态标记和字符串数据类型。

**提示：**要导入 RSLogix5000 预定义字符串，则在执行导入前，应在“原生标记数据库”CSV 文件中定义字符串类型 ("STRING.DATA" 和 "STRING.LEN") 中包含的两个元素。

## Logix 地址

对“Logix 地址”的限制符合 RSLogix5000 要求，与以下 IEC 1131-3 标识符规则对应：

- 必须以字母字符 (A-Z、a-z) 或下划线开头。
- 只能包含字母数字字符和下划线。
- 每段最多包含 40 个字符。
- 不能有连续的下划线。
- 不区分大小写。

如果标记因没有唯一的 Logix 地址或不满足上述标识符要求而无法导入，会导致将具有指定 Logix 地址的消息发布到服务器的“事件日志”。

## Logix 数据类型

支持以下预定义的“Logix 原子型”：

| Logix 数据类型 | 支持的数据类型 |
|------------|---------|
| BOOL       | 布尔型     |
| SINT       | 字符型、字节型 |
| 整型         | 短整型、字   |
| DINT       | 长整型、双字型 |
| LINT       | 双精度、日期  |
| REAL       | 浮点数     |

● **注意：**不支持其他预定义或用户定义的复杂 (结构化) 数据类型。可以通过将“Logix 地址”限定为原子型来导入结构化数据。例如，存在称为 TIME 的结构化类型，其描述如下：

```
TIME
{
  HOUR @ SINT
  MIN @ SINT
  SEC @ SINT
}
```

结构可以分解，其中原子型成员导入为 "TIME.HOUR"、"TIME.MIN" 和 "TIME.SEC"，其关联的“Logix 数据类型”、“外部访问”和“说明”均遵循上述 CSV 格式。在 CSV 导入中指定的所有不受支持的“Logix 数据类型”值均默认为双整型，以便导入成功。

## 外部访问

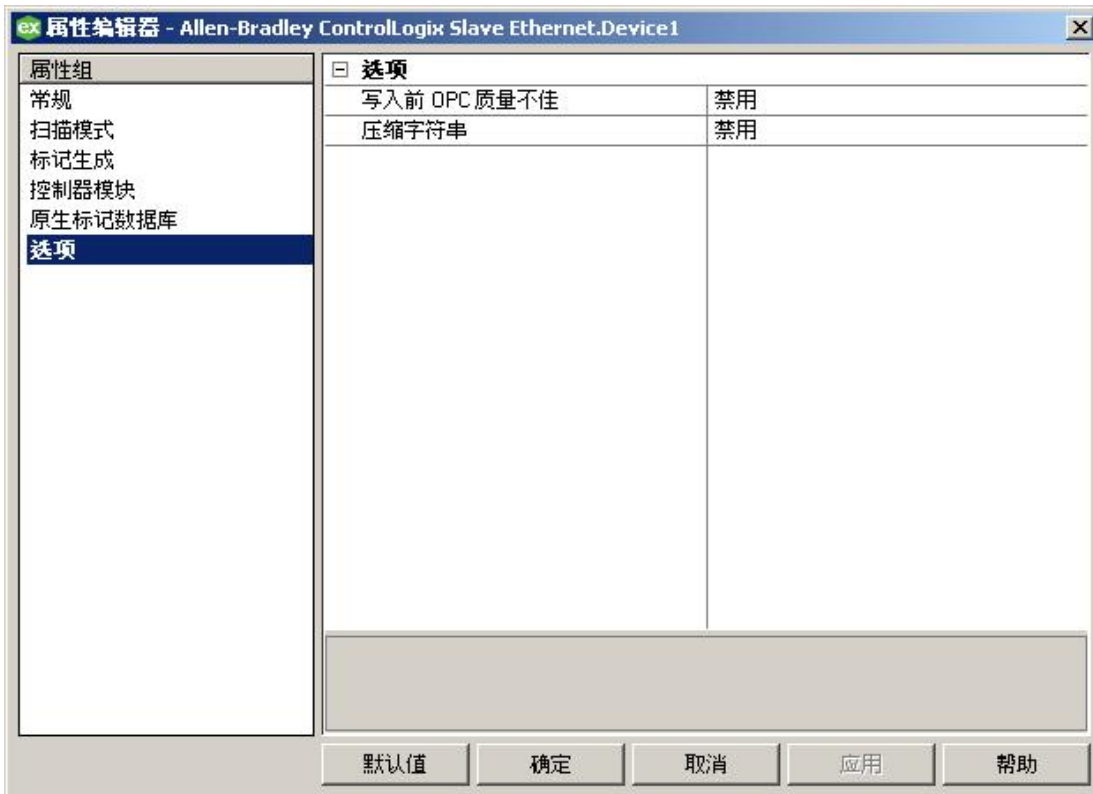
“外部访问”指定主设备的“读/写”权限。此访问不适用于客户端标记，客户端标记默认情况下始终具有“读/写”客户端访问权限。支持以下外部访问类型：指定的所有其他值都设置为“读/写”。

- **R/W:** 主设备对“原生标记”具有“读/写”权限。
- **RO:** 主设备具有“只读”权限。所有写入尝试均失败，并出现相应的错误 (CIP 错误 0x0F)。

## 说明

“说明”在自动标记数据库生成过程中使用，且被截断为 64 个字符。“说明”字段必须存在，但可以留空。

## 设备属性 - 选项



**“写入前 OPC 质量不佳”**: 选择“启用”可在写入原生标记前强制驱动程序返回“质量不佳”消息。可能会从客户端接口 (如 OPC) 或从主设备 (如 ControlLogix 5000 系列 PLC) 写入。当向数组的单个项目写入时, 整个数组会进行初始化, 并会返回“质量良好”消息。默认设置为禁用状态。启动时, Allen-Bradley ControlLogix Unsolicited 驱动程序 会将整型/数字数据类型值初始化为零 (0), 并将字符串初始化为空。默认情况下, 客户端接收具有“良好质量”的初始更新; 但是, 可以针对每个设备修改此行为。

**“压缩字符串”**: 对于字符串标记中显示的字符串, 启用此选项可包括压缩格式的数组元素的所有字节。禁用时, 字符串标记中显示的字符串为非压缩格式, 其中, 仅显示每个元素的低位字节。

## 数据类型说明

| 数据类型 | 说明  |
|------|---|
| 布尔型  | 单个位   |
| 字符   | 有符号 8 位值<br>位 0 是低位<br>位 6 是高位<br>位 7 是符号位    |
| 字节   | 无符号 8 位值<br>位 0 是低位<br>位 7 是高位                |
| 短整型  | 有符号 16 位值<br>位 0 是低位<br>位 14 是高位<br>位 15 是符号位 |
| 字    | 无符号 16 位值<br>位 0 是低位<br>位 15 是高位              |
| 长整型  | 有符号 32 位值<br>位 0 是低位<br>位 30 是高位<br>位 31 是符号位 |
| 双字型  | 无符号 32 位值<br>位 0 是低位<br>位 31 是高位              |
| 浮点型  | 32 位浮点值<br>位 0 是低位<br>位 31 是高位                |
| 双精度  | 64 位浮点值<br>位 0 是低位<br>位 63 是高位                |
| 字符串  | 通常为 null 终止、null 填充或空白填充的 ASCII 字符串。          |
| 日期   | 64 位浮点值。                                      |

● 有关特定于 Logix 平台的数据类型的说明，请参阅[地址说明](#)。

## 地址说明

Allen-Bradley ControlLogix Unsolicited 驱动程序支持基于符号标记的寻址。

### Logix 基于标记的寻址

此驱动程序使用基于标记或符号的寻址结构，通常称为 **Logix** 或原生标记 (与 Rockwell 自动化集成体系架构一致)。这些标记与常规 PLC 数据项的区别在于，标记名称是地址，而不是物理或逻辑地址。

驱动程序允许用户访问控制器的原子型数据类型：**BOOL**, **SINT**, **INT**, **DINT**, **LINT**, **REAL**。尽管某些预定义类型为结构，但归根结底，它们基于这些原子型数据类型。因此，结构的所有非结构 (原子型) 成员均可供访问。例如，不能将 **TIMER** 分配给服务器标记，但是，可以分配 **TIMER** 的原子型成员 (例如 **TIMER.EN**、**TIMER.ACC** 等)。如果结构成员为结构本身，则必须展开这两个结构，以便访问子结构的原子型成员。对于用户和模块定义的类型，这种情况更为常见，但是，在任何预定义类型中不会出现这种情况。

| 原子型数据类型 | 说明           | 范围   |
|---------|--------------|--|
| BOOL    | 单个位值         | VT_BOOL<br>0、1   |
| SINT    | 有符号 8 位值     | VT_I1<br>-128 到 127  |
| INT     | 有符号 16 位值    | VT_I2<br>-32,768 到 32,767                                    |
| DINT    | 有符号 32 位值    | VT_I4<br>-2,147,483,648 到 2,147,483,647                      |
| LINT    | 有符号 64 位值    | VT_R8<br>-9.22337E18 到 9.22336E18                            |
| REAL    | 32 位 IEEE 浮点 | VT_R4<br>1.1755 E-38 到 3.403E38<br>0<br>-3.403E-38 到 -1.1755 |

### 客户端/服务器标记地址规则

**Logix** 标记名称与客户端/服务器标记地址对应。**Logix** 标记名称 (通过 RSLogix5000 输入) 遵循 IEC 1131-3 标识符规则。客户端/服务器标记地址遵循这些规则。它们是：

- 必须以字母 (A-Z、a-z) 字符或下划线开头。
- 只能包含字母数字字符和下划线。
- 每段最多包含 40 个字符。
- 不能有连续的下划线。
- 不区分大小写。

#### 注意：

1. 服务器中的标记名称分配与地址分配不同，因为名称不能以下划线开头。
2. 要正确验证标记，用于表示静态客户端标记的原生标记必须存在于原生标记数据库中。

## 地址格式

进行 **Logix** 标记寻址有多种方法：可以在服务器中静态寻址，也可以从客户端动态寻址。所选格式取决于标记的类型及其使用方式。例如，在“短整型”标记内访问位时，将使用位格式。

有关地址格式和语法的详细信息，请参阅下表。

注意：对于 RSLogix5000 来说，除数组和字符串之外的所有格式均为原生格式。引用原子型数据类型时，用户可以将 RSLogix5000 标记名称复制并粘贴到服务器的标记地址参数：只要原生标记数据库中存在相应的原生标记，该名称就有效。

| 格式   | 语法                             | 示例                                 | 注释                             |
|------|--------------------------------|------------------------------------|--------------------------------|
| 标准   | <Logix 标记名称>                   | tag_1                              | 标记不能是数组。                       |
| 数组元素 | <Logix 数组标记名称> [1 维, 2 维, 3 维] | tag_1 [2, 58, 547]<br>tag_1 [0, 3] | 维度范围 = 1 至 3。元素范围 = 0 至 65535。 |

| 格式     | 语法   | 示例                                    | 注释  |
|--------|--|---------------------------------------|---|
| 无偏移数组* | <Logix 数组标记名称> {列数}<br><Logix 数组标记名称> {行数}{列数} | tag_1 {8}<br>tag_1 {2}{4}             | 维度范围 = 1 至 2。<br>元素范围 = 1 至 65535。<br><br>要读/写的元素数等于行数乘以列数。如果未指定任何行，则行数默认为 1。<br><br>数组起点处具有零偏移 (所有维度的数组索引均等于 0)。 |
| 带偏移数组* | <Logix 数组元素标记> {列数}<br><Logix 数组元素标记> {行数}{列数} | tag_1 [2, 3]{10}<br>tag_1 [2, 3] 2{5} | 数组起点处的偏移由数组元素标记中的维度指定。数组始终涵盖最高维度。因此, "tag_1[2,3]{10}" 将产生由元素 tag_1[2,3] -> tag_1 [2,13] 组成的数组。                    |
| 位      | <Logix 标记名称> .位<br><Logix 标记名称> . [位]          | tag_1.0<br>tag_1.[0]                  | 位范围 = 0 至 31。<br><br>如果标记为数组, 则它必须是布尔型数组; 否则, 标记不能为数组。  |
| 字符串    | <Logix 标记名称> .数据 / <最大字符串长度>                   | tag_1.Data/4                          | 长度范围 = 1 至 65535。<br><br>可以读/写到字符串的最大字符数。   |

\*由于此格式可以请求多个元素, 因此数组数据的传递顺序取决于 Logix 数组标记的维度。例如, 如果行数乘以列数的结果为 4 且控制器标记为 3X3 元素数组, 则所引用的元素采用 "array\_tag [0,0]"、"array\_tag [0,1]"、"array\_tag [0,2]" 和 "array\_tag [1,0]" 的顺序。如果控制器标记为 2X10 元素数组, 则结果可能有所不同。

● 有关 1 维、2 维和 3 维数组如何参考元素的详细信息, 请参阅 [Logix 数组数据排序](#)。

## 标记范围

### 全局标记

全局标记是在控制器中具有全局范围的 Logix 标记。任何程序或任务都可以访问“全局标记”; 但是, “全局标记”的引用方式数量取决于其 Logix 数据类型和使用的地址格式。

### 程序标记

程序标记与全局标记相同, 但前者的范围局限于定义该标记的程序内。驱动程序目前不支持导入带有程序标识的“原生标记”。

### 结构标记寻址

Logix 结构标记是带有一个或多个成员标记的标记 (可为原子型或结构型)。

<结构名称> . <原子型标记>

这意味着将按照以下语法对子结构进行寻址:

<结构名称> . <子结构名称> . <原子型标记>

将按照以下语法对结构数组进行寻址:

<结构数组名称> [1 维, 2 维, 3 维]. <原子型标记>

这意味着将按照以下语法对子结构数组进行寻址:

<结构名称> . <子结构数组名称> [1 维, 2 维, 3 维]. <原子型标记>

● **注意:** 上述示例显示了几种涉及结构的寻址可能性, 这些示例仅用于介绍结构寻址。有关详细信息, 请参阅 *Rockwell/Allen-Bradley* 文档。

## 高级寻址

用户具有多个可包括在符号标记地址中的符号寻址选项。位和数组寻址语法的数据类型具有以下限制:

- 对于位语法, 索引不得超过数据类型的位大小。例如, "MyDint @ Dint" 作为原生标记导入。位索引不得超过 31 位 (因为双整数为有符号 32 位值)。

- 对于数组语法，数组中的偏移量和元素数不得超过关联原生标记中的元素数。例如，"MyDintArray @ DINT[10]" 作为原生标记导入。由于数组中包括原生标记的前 5 个和后 5 个元素，因此地址为 "MYDINTARRAY[0]{5}" 和 "MYDINTARRAY[4]{5}" 的静态标记有效。地址为 "MYDINTARRAY[5]{10}" 的静态标记无效，这是因为标记正在请求从偏移 5 开始的 10 个双整数，且原生标记数组不够大。

有关高级主题的详细信息，请参阅下表。

| 元素    | 语法                         | 示例                       | 注释  |
|-------|----------------------------|--------------------------|---|
| 标准    | <标记名称>                     | tag_1                    | 不适用   |
| 无偏移数组 | <数组标记名称> {列数}              | tag_1 {8}                | 要读/写的元素数等于行数乘以列数。如果未指定任何行，则行数默认设置为 1。必须为至少一个数组元素寻址。<br><br>数组起点处具有零偏移 (所有维度的数组索引均等于 0)。 |
|       | <数组标记名称> {行数} {列数}         | tag_1 {2}{4}             |   |
| 带偏移数组 | <数组元素标记> [偏移] {列数}         | tag_1 [5]{8}             | 要读/写的元素数等于行数乘以列数。如果未指定任何行，则行数默认设置为 1。必须为至少一个数组元素寻址。<br><br>数组起点处具有零偏移 (所有维度的数组索引均等于 0)。 |
|       | <数组元素标记> [偏移] {行数} {列数}    | tag_1 [5]{2}{4}          |   |
| 位     | <标记名称> . 位<br><标记名称> . [位] | tag_1 . 0<br>tag_1 . [0] | 不适用<br>不适用  |
| 字符串   | <标记名称> / <元素计数>            | tag_1 / 4                | 元素计数必须至少为 1。要读/写的字符数取决于“压缩字符串”属性。*  |

\*启用时，字符数等于元素计数乘以元素大小 (“整型”数组的 4 个元素表示 8 个字符)。\*禁用时，字符数等于元素计数 (“整型”数组的 4 个元素表示 4 个字符)。

有关详细信息，请参阅[选项](#)。

## Logix 数组数据的排序

由于原生标记最多支持三维数组，因此 Logix 数组数据的排序将映射至 2 维 OPC 数组。

### 1. 维数组 - 数组 [1 维]

1 维数组数据按升序传递至控制器并从中传出。  
适用于 (维度 1 = 0; 维度 1 < 维度 1\_max; 维度 1++)

示例: 3 元素数组

数组 [0]

数组 [1]

数组 [2]

### 2. 维数组 - 数组 [1 维, 2 维]

2 维数组数据按升序传递至控制器并从中传出。  
适用于 (维度 1 = 0; 维度 1 < 维度 1\_max; 维度 1++)  
适用于 (维度 2 = 0; 维度 2 < 维度 2\_max; 维度 2++)

**示例: 3X3 元素数组**

数组 [0, 0]  
数组 [0, 1]  
数组 [0, 2]  
数组 [1, 0]  
数组 [1, 1]  
数组 [1, 2]  
数组 [2, 0]  
数组 [2, 1]  
数组 [2, 2]

**3. 维数组 - 数组 [1 维, 2 维, 3 维]**

3 维数组数据按升序传递至控制器并从中传出。  
适用于 (维度 1 = 0; 维度 1 < 维度 1\_max; 维度 1++)  
适用于 (维度 2 = 0; 维度 2 < 维度 2\_max; 维度 2++)  
适用于 (维度 3 = 0; 维度 3 < 维度 3\_max; 维度 3++)

**示例: 3X3X3 元素数组**

数组 [0, 0, 0]  
数组 [0, 0, 1]  
数组 [0, 0, 2]  
数组 [0, 1, 0]  
数组 [0, 1, 1]  
数组 [0, 1, 2]  
数组 [0, 2, 0]  
数组 [0, 2, 1]  
数组 [0, 2, 2]  
数组 [1, 0, 0]  
数组 [1, 0, 1]  
数组 [1, 0, 2]  
数组 [1, 1, 0]  
数组 [1, 1, 1]  
数组 [1, 1, 2]  
数组 [1, 2, 0]  
数组 [1, 2, 1]  
数组 [1, 2, 2]  
数组 [2, 0, 0]  
数组 [2, 0, 1]  
数组 [2, 0, 2]  
数组 [2, 1, 0]  
数组 [2, 1, 1]  
数组 [2, 1, 2]  
数组 [2, 2, 0]  
数组 [2, 2, 1]  
数组 [2, 2, 2]



## 错误代码

Allen-Bradley ControlLogix Unsolicited 驱动程序 可能会返回下列错误代码。

● 有关特定类型的错误代码的详细信息，请参见下面列表中的链接。

[以太网/IP 封装错误代码](#)

[CIP 错误代码](#)

[0x01 扩展错误代码](#)

[0xFF 扩展错误代码](#)

## 以太网/IP 封装错误代码

Allen-Bradley ControlLogix Unsolicited 驱动程序 可能会返回下列错误代码。

● **注意：**错误代码为十六进制。

| 错误   | 说明           |
|------|--------------|
| 0001 | 命令未处理。       |
| 0002 | 命令内存不可用。     |
| 0003 | 数据不正确或不完整。   |
| 0064 | 会话 ID 无效。    |
| 0065 | 标题长度无效。      |
| 0069 | 请求的协议版本不受支持。 |

## CIP 错误代码

错误代码为十六进制。

| 错误 | 说明                      |
|----|-------------------------|
| 01 | 连接失败*                   |
| 02 | 资源不足                    |
| 03 | 参数值无效                   |
| 04 | IOI 无法被解密或标记不存在         |
| 05 | 未知目标                    |
| 06 | 请求的数据不适合响应数据包           |
| 08 | 不支持的服务                  |
| 0F | 权限被拒绝                   |
| 13 | 为执行服务指定的命令数据/参数不足。      |
| 26 | 指定的 IOI 字数与 IOI 字数统计不匹配 |
| FF | 一般错误**                  |

● \*另请参阅：[0x01 扩展错误代码](#)

● \*\*另请参阅：[0xFF 扩展错误代码](#)

## 0x01 扩展错误代码

Allen-Bradley ControlLogix Unsolicited 驱动程序 可能会返回 CIP 错误 0x01 的以下扩展错误。

● **注意：**错误代码为十六进制。

| 错误     | 说明          |
|--------|-------------|
| 0x0205 | 未连接发送参数错误。  |
| 0x0312 | 链路地址不可用。    |
| 0x0318 | 至自身的链路地址无效。 |

## 0xFF 扩展错误代码

Allen-Bradley ControlLogix Unsolicited 驱动程序 可能会返回 CIP 错误 0xFF 的以下扩展错误。

● **注意：**错误代码为十六进制。

| 错误   | 说明             |
|------|----------------|
| 2104 | 地址超出范围。        |
| 2105 | 尝试的访问超出数据对象末端。 |
| 2107 | 数据类型无效或不受支持。   |

## 事件日志消息

以下信息涉及发布到主要用户界面中“事件日志”窗格的消息。请参阅有关筛选和排序“事件日志”详细信息视图的服务器帮助。服务器帮助包含许多常见的消息，因此也应对其进行搜索。通常，其中会尽可能提供消息的类型（信息、警告）和故障排除信息。

---

### 导入原生标记数据库时出错。无法打开文件。| OS 错误 = '<错误>'。

**错误类型：**

错误

---

### 导入原生标记数据库时出错。无法打开文件，常规读取失败。

**错误类型：**

错误

**可能的原因：**

文件缺失、损坏或格式不正确。

**可能的解决方案：**

找到标记数据库文件，并在再试一次之前验证标题行、格式和位置。

---

### 导入原生标记数据库时出错。不支持文件编码。

**错误类型：**

错误

**可能的原因：**

正在导入的 CSV 文件使用了不支持的文件编码。

**可能的解决方案：**

将 CSV 文件更新为使用 ANSI 或 UTF-8 编码方法。

---

### 导入原生标记数据库时出错。

**错误类型：**

错误

**可能的原因：**

已遇到意外错误。

**可能的解决方案：**

验证正在导入的 CSV 文件的格式是否正确。

---

### 导入原生标记数据库时出错，字段名称未识别。| 未识别字段名称 = '<字段名称>'。

**错误类型：**

错误

**可能的原因：**

CSV 文件定义了一个原生标记数据库导入不支持的字段。

**可能的解决方案：**

验证 CSV 文件中不存在意外字段。

**● 注意：**

支持的字段名称包括“Logix 地址”、“Logix 数据类型”、“外部访问”和“说明”。

---

**导入原生标记数据库时出错，字段名称重复。| 重复字段名称 = '<字段>'。**

---

**错误类型：**

错误

**可能的原因：**

正在导入的 CSV 文件中包含同一字段名称的多种定义。

**可能的解决方案：**

移除或更正 CSV 文件中的重复字段。支持的字段名称包括 Logix 地址、Logix 数据类型、外部访问和说明。

---

**导入原生标记数据库时出错。标记字段标识记录缺失。**

---

**错误类型：**

错误

**可能的原因：**

正在导入的 CSV 文件中不包含标题。

**可能的解决方案：**

验证 CSV 文件中是否缺失字段，并确认或更正标题行。

**● 注意：**

支持的字段名称包括 Logix 地址、Logix 数据类型、外部访问和说明。

---

**导入原生标记数据库时出错。标记字段标识记录不完整。**

---

**错误类型：**

错误

**可能的原因：**

正在导入的 CSV 文件不包含完整且有效的标题。

**可能的解决方案：**

验证 CSV 文件中是否缺失字段，并确认或更正标题行。

**● 注意：**

支持的字段名称包括 Logix 地址、Logix 数据类型、外部访问和说明。

---

**无法启动未经请求的 Logix 服务器。**

---

**错误类型：**

错误

**可能的原因：**

驱动程序无法绑定和侦听指定的 IP/端口。

**可能的解决方案：**

验证指定端口 (TCP 或 UDP) 是否正在由另一个应用程序使用，然后解决所有冲突。

---

**标记地址无效。未导入原生标记。| 无效地址 = '<地址>'。**

---

**错误类型：**

警告

已将两个通道配置为使用相同的网络适配器 IP 和 TCP 端口。必须将每个通道绑定至唯一的本地 IP 和端口。| 第一个通道 = '<通道>', 第二个通道 = '<通道>'。

**错误类型:**

警告

已将两台设备配置为使用来自 EtherNet/IP 模块的相同路径。每个通道必须具有唯一的来自 EtherNet/IP 模块的路径。| 第一台设备 = '<地址>', 第二台设备 = '<地址>'。

**错误类型:**

警告

**可能的原因:**

为设备配置了已经使用的 CPU 类型和插槽编号。

**可能的解决方案:**

1. 请选择其他 CPU 类型 (本地或远程)。
2. 如果将设备配置为远程控制器模块, 请选择其他插槽编号。

标记地址无效。不允许重复的标记地址。| 无效地址 = '<地址>'。

**错误类型:**

警告

**可能的原因:**

正在导入的 CSV 文件中包含使用同一 Logix 地址的一个或多个标记。

**可能的解决方案:**

在正在导入的原生标记数据库 CSV 文件中, 消除包含重复 Logix 地址的原生标记。

无法为标记分配内存。| 标记地址 = '<地址>'。

**错误类型:**

警告

**可能的原因:**

无法分配构建标记所需的资源。未将标记添加到项目。

**可能的解决方案:**

关闭任何未使用的应用程序和/或增加虚拟内存量, 然后再试一次。

原生标记无效。单个标记大小限制为 128 kB。| 标记地址 = '<地址>'。

**错误类型:**

警告

**可能的原因:**

正在导入的 CSV 文件中包含需要 128 kB 以上内存才能表示的单个标记定义, 例如在大型数组中。无法定义需要 128 kB 以上内存的原生标记。

**可能的解决方案:**

减小数组大小或从待导入的 CSV 文件中移除标记。

导入原生标记时出错。总数据库标记数据大小限制为 128 kB。| 标记地址 = '<地址>'。

---

**错误类型：**

警告

**可能的原因：**

正在导入的 CSV 文件中包含需要 128 kB 以上内存的标记定义，例如在大型数组中。

**可能的解决方案：**

如果需要 128 kB 以上内存，请创建新设备并在多台设备之间拆分标记数据库。

指定的 TCP/IP 端口超出范围。正在使用默认端口。| 有效范围 = <数字> 到 <数字>，默认端口 = <数字>。

---

**错误类型：**

警告

**可能的原因：**

已加载将 TCP/IP 端口指定为 0 的项目，但不支持此选项。

**可能的解决方案：**

请使用默认端口 (44818) 或选择处于有效范围 (1 到 65535) 内的端口。

通道内的另一台设备已注册为本地 CPU。| 设备 = <设备>。

---

**错误类型：**

警告

原生标记已导入。| 标记计数 = <计数>，标记数据库路径 = '<路径>'。

---

**错误类型：**

信息化

标记已生成。| 标记计数 = <计数>，标记层次模式 = '<模式>'。

---

**错误类型：**

信息化

由于资源不足，因此无法执行自动标记生成。

---

**错误类型：**

信息化

# 索引

## 0

0x01 扩展错误代码 25

0xFF 扩展错误代码 26

## C

CIP 错误代码 25

CSV 15

## E

EtherNet/IP 模块 8

## I

IEEE-754 浮点 8

## T

TCP/IP 端口 9

## 竣

帮助内容 4

## 墨

本地 14

## 椴

标记层次 15

标记地址无效。不允许重复的标记地址。| 无效地址 = '<地址>'。 29

标记地址无效。未导入原生标记。| 无效地址 = '<地址>'。 28

标记范围 22

标记生成 12

标记已生成。| 标记计数 = <计数>, 标记层次模式 = '<模式>'。 30

标识 10

## 尪

不超过扫描速率请求数据 12

不扫描, 仅按需求轮询 12

## 焮

布尔型 20

## 捲

插槽 14

## 冻

创建 13

## 钹

错误代码 25

## 宜

导入标记... 15

导入数据库 15

导入原生标记时出错。总数据库标记数据大小限制为 128 kB。| 标记地址 = '<地址>'。 30

导入原生标记数据库时出错, 字段名称未识别。| 未识别字段名称 = '<字段名称>'。 27

导入原生标记数据库时出错, 字段名称重复。| 重复字段名称 = '<字段>'。 28

导入原生标记数据库时出错。 27

导入原生标记数据库时出错。标记字段标识记录不完整。 28

导入原生标记数据库时出错。标记字段标识记录缺失。 28

导入原生标记数据库时出错。不支持文件编码。 27

导入原生标记数据库时出错。无法打开文件, 常规读取失败。 27

导入原生标记数据库时出错。无法打开文件。| OS 错误 = '<错误>'。 27

## 嗽

地址格式 21

地址说明 21



## 亡

逗号分隔值 15

## 瞍

短整型 20

## 宙

对于重复标记 13

## 雾

非规范浮点数处理 8

## 洎

浮点型 20

## 燭

父组 13

## 褌

覆盖 13

## 栖

概述 5

## 駉

高级信道属性 8

高级寻址 22

## 括

控制器模块 13

## 竭

来自缓存的初始更新 12

## 趟

路径 14

## 稿

模块类型 14

模拟 11

## 駢

驱动 7, 11

## 敦

日期 20

## 戈

扫描模式 12

## 净

删除 13

## 讓

设备启动时 13

设备设置 10

设备属性 - 标记生成 12

设置 6

## 璿

生成 13

## 丫

事件日志消息 27

## 对

属性更改时 13

## 攘

数据类型说明 20

数据收集 11

## 印

双精度 20

双字型 20

## 辺

通道分配 11

通道内的另一台设备已注册为本地 CPU。| 设备 = <设备>。 30

通道设置 6

通道属性 - 常规 6

通道属性 - 写入优化 7

通道属性 - 以太网通信 7

## 缙

网络适配器 7

## 馊

无法启动未经请求的 Logix 服务器。 28

无法为标记分配内存。| 标记地址 = '<地址>'。 29

## 償

- 写入非布尔标记的最新值 7
- 写入前 OPC 质量不佳 19
- 写入所有标记的所有值 7
- 写入所有标记的最新值 8
- 写入优化 7

## 坵

- 型号 11

## 辯

- 选项 18

## 匱

- 压缩字符串 19

## 嶒

- 已将两个通道配置为使用相同的网络适配器 IP 和 TCP 端口。必须将每个通道绑定至唯一的本地 IP 和端口。| 第一个通道 = '<通道>', 第二个通道 = '<通道>'。 29
- 已将两台设备配置为使用来自 EtherNet/IP 模块的相同路径。每个通道必须具有唯一的来自 EtherNet/IP 模块的路径。| 第一台设备 = '<地址>', 第二台设备 = '<地址>'。 29
- 已扩展 15
- 已压缩 15

## J

- 以扫描速率请求所有数据 12
- 以太网/IP 封装错误代码 25

## 躡

- 优化方法 7

## 璣

由于资源不足, 因此无法执行自动标记生成。 30

## 匿

原生标记数据库 14

原生标记数据库 CSV 导入 17

原生标记无效。单个标记大小限制为 128 kB。| 标记地址 = '<地址>'。 29

原生标记已导入。| 标记计数 = <计数>, 标记数据库路径 = '<路径>'。 30

## 軾

远程 14

## 億

允许子组 13

## 勻

占空比 8

## 錕

长整型 20

## 藹

诊断 7

## 摺

指定的 TCP/IP 端口超出范围。正在使用默认端口。| 有效范围 = <数字> 到 <数字>, 默认端口 = <数字>。  
30

主设备配置 10

## 婿

字 20

字符 20

字符串 20

字节 20

## 逦

遵循标签指定的扫描速率 12

遵循客户端指定的扫描速率 12