

Allen-Bradley ControlLogix Ethernet 驱动程序

© 2019, PTC Inc. 保留所有权利。

目录

Allen-Bradley ControlLogix Ethernet 驱动程序	1
目录	2
Allen-Bradley ControlLogix Ethernet 驱动程序	5
概述	6
设置	6
Configuration API Allen-Bradley ControlLogix 示例	9
通道属性 - 常规	10
通道属性 - 以太网通信	10
通道属性 - 写入优化	11
通道属性 - 高级	11
设备属性 - 常规	12
操作模式	12
设备属性 - 扫描模式	13
设备属性 - 定时	13
设备属性 - 自动降级	14
设备属性 - 标记生成	14
设备属性 - Logix 通信参数	16
设备属性 - Logix 选项	16
设备属性 - Logix 数据库设置	17
设备属性 - ENIDF1/DH+/CN 网关通信参数	19
块写入	20
设备属性 - SLC 500 插槽配置	20
Configuration API - 插槽配置	22
设备属性 - 冗余	22
SLC 500 模块化 I/O 选择指南	22
性能优化	24
优化通信	24
优化应用程序	26
性能统计信息和调优	27
性能调优示例	27
数据类型说明	37
默认数据类型条件	38
地址说明	38
Logix 寻址	39
MicroLogix 寻址	40
SLC 500 固定 I/O 寻址	42
SLC 500 模块化 I/O 寻址	43
PLC-5 系列寻址	44
Logix 基于标记的寻址	45
地址格式	46
标记范围	47

内部标记	48
预定义术语标记	48
原子型数据类型寻址	48
寻址结构数据类型	50
寻址字符串数据类型	50
Logix 数组数据的排序	51
Logix 高级寻址	52
高级寻址: BOOL	52
高级寻址: SINT	53
高级寻址: INT	55
高级寻址: DINT	57
高级寻址: LINT	59
高级寻址: REAL	60
高级寻址: USINT	62
高级寻址: UINT	63
高级寻址: UDINT	65
高级寻址: ULINT	66
高级寻址: LREAL	67
文件列表	68
输出文件	69
输入文件	72
状况文件	75
二进制文件	76
计时器文件	76
计数器文件	77
控制文件	78
整型文件	78
浮点型文件	79
ASCII 文件	80
字符串文件	80
BCD 文件	81
长整型文件	81
MicroLogix PID 文件	82
PID 文件	83
MicroLogix 消息文件	84
消息文件	85
块传输文件	86
功能文件	87
高速计数器文件 (HSC)	87
实时时钟文件 (RTC)	88
信道 0 通信状况文件 (CS0)	89
信道 1 通信状况文件 (CS1)	89
I/O 模块状况文件 (IOS)	90
自动生成标记数据库	90

标记层次	90
控制器到服务器的名称转换	92
自动生成标记数据库前准备	93
错误代码	95
封装错误代码	95
CIP 错误代码	95
0x0001 扩展错误代码	96
0x001F 扩展错误代码	97
0x00FF 扩展错误代码	97
事件日志消息	98
附录	99
附录 A - 通道属性	99
附录 B - 设备属性	99
附录 C - 标记属性	100
Logix 设备 ID	101
CompactLogix 5300 以太网设备 ID	101
1761-NET-ENI 设置	102
Data Highway Plus 网关设置	103
ControlNet™ 网关设置	104
EtherNet/IP 网关设置	105
串行网关设置	106
MicroLogix 1100 设置	107
通信路由	107
连接路径规范	107
路由示例	108
选择协议模式	111
检测“控制器项目”中所做的更改	112
SoftLogix 5800 连接注意事项	113
索引	114

Allen-Bradley ControlLogix Ethernet 驱动程序

帮助版本 1.167

目录

[概述](#)

什么是 Allen-Bradley ControlLogix Ethernet 驱动程序？

[设置](#)

如何配置使用此驱动程序的设备？

[快速入门](#)

如何使用 Configuration API 配置通道和设备？

[通信路由](#)

如何与远程 ControlLogix 5000 处理器或 1756-DHRIO/1756-CNB 接口模块进行通信？

[性能优化](#)

如何从 Allen-Bradley ControlLogix Ethernet 驱动程序 获得最佳性能？

[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 Ethernet 设备上的标记进行寻址？

[自动生成标记数据库](#)

如何轻松配置 Allen-Bradley ControlLogix Ethernet 驱动程序 的标记？

[事件日志消息](#)

此驱动程序会产生哪些消息？

[错误代码](#)

什么是 Allen-Bradley ControlLogix Ethernet 错误代码？

[附录](#)

在何处可以找到与 Allen-Bradley ControlLogix Ethernet 驱动程序 相关的其他信息？

概述

Allen-Bradley ControlLogix Ethernet 驱动程序 提供将 Allen-Bradley ControlLogix Ethernet 控制器连接至客户端应用程序的简单且可靠的方式；其中包括 HMI、SCADA、Historian、MES、ERP 和无数自定义应用程序。

支持 Allen-Bradley 控制器

ControlLogix® 5500 系列

可通过用于以太网通信的 EtherNet/IP 通信模块，或用于使用控制器串行端口进行以太网转串口通信的 1761-NET-ENI 模块，实现与 ControlLogix 的通信。

CompactLogix™ 5300 系列

与 CompactLogix 进行的以太网通信需要一个具有内置 EtherNet/IP 端口的处理器，例如 1769-L35E。否则，与 CompactLogix 的通信需要一个用于使用控制器串行端口进行以太网转串口通信的 1761-NET-ENI 模块。

FlexLogix 5400 系列

可以通过用于以太网通信的 1788-ENBT 子卡，或用于使用控制器串行端口进行以太网转串口通信的 1761-NET-ENI 模块，实现与 FlexLogix 的通信。

SoftLogix 5800

驱动程序支持 Allen-Bradley SoftLogix 5800 系列控制器，并要求 SoftLogix PC 中存在以太网卡。

Data Highway Plus 网关

驱动程序支持具有 Data Highway Plus 接口的 PLC-5 系列和 SLC 500 系列。这可以通过 DH+ 网关实现，并需要一个上述的 PLC、EtherNet/IP 通信模块以及 1756-DHRIO 接口模块 (两者皆位于 ControlLogix 机架中)。

ControlNet 网关

驱动程序支持 PLC-5C 系列。这可以通过 ControlNet 网关实现，并需要一个上述的 PLC、EtherNet/IP 通信模块以及 1756-CNBR 接口模块 (两者皆位于 ControlLogix 机架中)。

1761-NET-ENI

驱动程序支持与 1761-NET-ENI 设备进行通信。ENI 设备通过向全双工 DF1 控制器和 Logix 控制器提供以太网转串行接口，来为设备网络和通信增加更大的灵活性。此驱动程序与 ENI 设备结合使用，可支持以下功能：

- ControlLogix 5500 系列*
- CompactLogix 5300 系列*
- FlexLogix 5400 系列*
- MicroLogix 系列
- SLC 500 固定 I/O 处理器
- SLC 500 模块化 I/O 系列
- PLC-5 系列

*这些型号需要 1761-NET-ENI 系列 B 或更高版本。

MicroLogix 1100

驱动程序支持使用 EtherNet/IP 与 MicroLogix 1100 (CH1 以太网) 进行通信。

ControlLogix 是 Allen-Bradley Company, LLC 的注册商标。
CompactLogix 是 Rockwell Automation, Inc 的商标。
所有商标都归其各自所有者所有。

设置

支持的设备

设备系列	通信
ControlLogix	通过 1756-

设备系列	通信
5550/5553/5555/5561/5562/5563/5564/5565/5571/5572/557-3/5574/5575/5580 处理器	ENBT/ENET/EN2F/EN2T/EN2TR/EN3TR/E-WEB/EN2TXT 以太网模块 通过串行网关 通过使用通道 0 (串行) 的 1761-NET-ENI 系列 B 或更高版本
CompactLogix 5320 / 5323 / 5330 / 5331 / 5332 / 5335 / 5343 / 5345 / 5370 / 5380 / 5480	处理器上的带有 E 后缀* 的内置 Ethernet/IP 端口 通过串行网关 通过使用通道 0 (串行) 的 1761-NET-ENI 系列 B 或更高版本
FlexLogix 5433/5434 处理器	通过 1788-ENBT 以太网子卡 通过串行网关 通过使用通道 0 (串行) 的 1761-NET-ENI 系列 B 或更高版本
SoftLogix 5810/5830/5860 处理器	通过 SoftLogix Ethernet/IP 报文通信模块 通过串行网关
MicroLogix 1000/1200/1500	通过 1761-NET-ENI 通过 EtherNet/IP 网关
MicroLogix 1100/1400	通过 MicroLogix 1100/1400 通道 1 (以太网) 通过 1761-NET-ENI 通过 EtherNet/IP 网关
SLC 500 固定 I/O 处理器	通过 1761-NET-ENI 通过 EtherNet/IP 网关
SLC 500 模块化 I/O 处理器 (SLC 5/01、SLC 5/02、SLC 5/03、SLC 5/04、SLC 5/05)	通过 DH + 网关** 通过 1761-NET-ENI 通过 EtherNet/IP 网关
PLC-5 系列 (不包括 PLC5/250 系列)	通过 DH+ 网关 通过 1761-NET-ENI 通过 EtherNet/IP 网关
PLC-5/20、PLC-5/40C、PLC-5/80C	通过 ControlNet 网关 通过 1761-NET-ENI 通过 EtherNet/IP 网关

*例如, 1769-L35E。

**该驱动程序支持任何支持 DH+ 或可以与 DH+ 网络连接 (例如 KF2 接口模块) 的 SLC 500 系列 PLC。

固件版本

设备系列	版本
ControlLogix 5550 (1756-L1)	11.35 - 13.34
ControlLogix 5553 (1756-L53)	11.28
ControlLogix 5555 (1756-L55)	11.32 - 16.04
ControlLogix 5561 (1756-L61)	12.31 - 20.11
ControlLogix 5562 (1756-L62)	12.31 - 20.11
ControlLogix 5563 (1756-L63)	11.26 - 20.11
ControlLogix 5564 (1756-L64)	16.03 - 20.11
ControlLogix 5565 (1756-L65)	16.03 - 20.11
ControlLogix 5571 (1756-L71)	20.11 - 32.12
ControlLogix 5572 (1756-L72)	19.11 - 32.12
ControlLogix 5573 (1756-L73)	18.12 - 32.12
ControlLogix 5574 (1756-L74)	19.11 - 32.12
ControlLogix 5575 (1756-L75)	18.12 - 32.12
ControlLogix 5580 (1756-L8)	28.11 - 32.12
CompactLogix 5370 (1769-L1)	20.11 - 32.12
CompactLogix 5370 (1769-L2)	20.11 - 32.12
CompactLogix 5370 (1769-L3)	20.11 - 32.12
CompactLogix 5320 (1769-L20)	11.27 - 13.18
CompactLogix 5323 (1769-L23)	17.05 - 20.11
CompactLogix 5330 (1769-L30)	11.27 - 13.18
CompactLogix 5331 (1769-L31)	16.22 - 20.11
CompactLogix 5332 (1769-L32)	16.22 - 20.11
CompactLogix 5335 (1769-L35)	16.22 - 20.11
CompactLogix 5343 (1768-L43)	15.07 - 20.11
CompactLogix 5345 (1768-L45)	16.24 - 20.11
CompactLogix 5380 (5069-L3)	28.11 - 32.12
CompactLogix 5480 (5069-L4)	32.12
FlexLogix 5433 (1794-L33)	11.25 - 13.33
FlexLogix 5434 (1794-L34)	11.25 - 16.02
SoftLogix 5800 (1789-L60)	16.00 - 20.01
ControlLogix、CompactLogix 和 FlexLogix 串行通信	1761-NET-ENI 系列 B 或更高版本或串行网关
MicroLogix 1100 (1763-L16AWA/BWA/BBB)	1.1

通信协议

通信协议为使用 TCP/IP 的 EtherNet/IP (Ethernet 上的 CIP)。

支持的最大通道数量为 1024。

Logix 和网关型号

Logix 和网关型号支持以下内容：

- 连接的报文通信
- 符号读取
- 符号写入
- 符号实例读取 (V21 或更高版本)
- 物理 (DMA) 读取 (V20 或更低版本)
- 符号实例写入

ENI 型号

ENI 型号支持未连接的报文通信。

Configuration API Allen-Bradley ControlLogix 示例

有关通道及设备定义和枚举的列表，请通过 REST 客户端访问以下端点。

通道定义

端点 (GET):

```
https://<主机名_或_ip>:<端口>/config/v1/doc/drivers/Allen-Bradley%20ControlLogix%20Ethernet/channels
```

设备定义

端点 (GET):

```
https://<主机名_或_ip>:<端口>/config/v1/doc/drivers/Allen-Bradley%20ControlLogix%20Ethernet/devices
```

以下 API 命令是创建 AllenBradley ControlLogix Ethernet 通道、设备和标记所需的最低要求。

● 有关常规项目配置的详细信息，请参阅服务器帮助文件。

创建 Allen-Bradley ControlLogix 通道

端点 (POST):

```
https://<主机名_或_ip>:<端口>/config/v1/project/channels
```

Body:

```
{ "common.ALLTYPES_NAME": "MyChannel", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER":  
  "Allen-Bradley ControlLogix Ethernet" }
```

● 另请参阅: [附录 A](#) 查看通道属性列表。

创建 Allen-Bradley ControlLogix 设备

端点 (POST):

```
https://<主机名_或_ip>:<端口>/config/v1/project/channels/MyChannel/devices
```

Body:

```
{ "common.ALLTYPES_NAME": "MyDevice", "servermain.DEVICE_ID_STRING": "<IP>,0,1",  
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet",  
  "servermain.DEVICE_MODEL": <模型枚举> }
```

● 注意: servermain.DEVICE_ID_STRING 值的格式可能根据为 servermain.DEVICE_MODEL 指定的模型枚举而异。如上所示的设备 ID 字符串格式针对的是 ControlLogix 5500 模型。

● 另请参阅: [附录 B](#) 查看设备属性列表。

创建 Allen-Bradley ControlLogix 标记

端点 (POST):

```
https://<主机名_或_ip>:<端口>/config/v1/project/channels/MyChannel/devices/MyDevice/tags
```

Body:

```
[ { "common.ALLTYPES_NAME": "MyTag1", "servermain.TAG_ADDRESS": "40001" } {  
"common.ALLTYPES_NAME": "MyTag2", "servermain.TAG_ADDRESS": "40002" } ]
```

- 另请参阅: [附录 C](#) 查看标记属性列表。
- 有关通过 *Configuration API* 配置标记和标记组的详细信息, 请参阅服务器帮助。

通道属性 - 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通道。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通道组成。通道充当 OPC 链路的基础构建块。此组用于指定常规通道属性, 如标识属性和操作模式。

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

标识

“名称”: 此通道的用户定义标识。在每个服务器项目中, 每个通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。通道名称是 OPC 浏览器信息的一部分。该属性是创建通道所必需的。

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

“说明”: 有关此通道的用户定义信息。
● 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”: 为该通道选择的协议/驱动程序。该属性指定在通道创建期间选择的设备驱动程序。它在通道属性中为禁用设置。该属性是创建通道所必需的。

● **注意:** 服务器全天在线运行时, 可以随时更改这些属性。其中包括更改通道名称以防止客户端向服务器注册数据。如果客户端在通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通道名称更改之后发布项, 并尝试通过原来的通道名称重新获取项, 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”: 启用此选项后, 通道的诊断信息即可提供给 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, 则该属性不可用。

● 有关详细信息, 请参阅服务器帮助中的“通信诊断”和“统计信息标记”。

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。如果留空或选择“默认”，则操作系统将选择默认适配器。

通道属性 - 写入优化

与任何服务器一样，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	<input checked="" type="checkbox"/> 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”: 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”:** 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”:** 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”:** 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

此组用于指定高级通道属性。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 checked="" type="checkbox"/> 非规范浮点数处理	
常规	浮点值	替换为零
以太网通信	<input checked="" type="checkbox"/> 设备间延迟	
写优化	设备间延迟 (毫秒)	0
高级		
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，则此属性不可用。根据通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”: 指定在接收到同一通道上的当前设备发出的数据后，通信通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。

设备属性 - 常规

属性组	标识	
常规	名称	ENI: SLC 500 Modular I/O
扫描模式	说明	
定时	驱动程序	Allen-Bradley ControlLogix Ethernet
自动降级	型号	ENI: SLC 500 Modular I/O
标记生成	通道分配	Allen-Bradley ControlLogix Ethernet
Logix 通信参数	ID	<255.255.255.2>
Logix 选项	操作模式	
Logix 数据库设置	数据收集	启用
ENI DF1/DH+/CN 网关通信参...	模拟	否
槽配置		
冗余		

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。

ID: 输入设备的唯一网络地址，格式通常为 <IP 或主机名>,1,<路由路径>,<插槽>。

● 寻址约定因型号和路由而不同。有关详细信息，请参阅[参考资料](#)下的特定型号寻址主题。

操作模式

Property Groups	+ Identification	
General	- Operating Mode	
Scan Mode	Data Collection	Enable
	Simulated	No

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的。“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”：此模式可将该数值集指定为最大扫描速率。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 - 定时

设备的“定时”属性允许调整驱动程序对错误条件的响应，以满足应用程序的需要。在很多情况下，需要更改环境的此类属性，以便获得最佳性能。由电气原因产生的噪音、调制解调器延迟以及较差的物理连接等因素都会影响通信驱动程序遇到的错误数或超时次数。“定时”属性特定于每个配置的设备。

属性组	通信超时	
常规	连接超时 (秒)	3
扫描模式	请求超时 (毫秒)	1000
定时	重试次数	3
自动降级		
冗余		
属性组	定时	
	请求间延迟 (毫秒)	0

通信超时

“连接超时”(Connect Timeout): 此属性 (主要由基于驱动程序的以太网使用) 控制建立远程设备套接字连接所需的时间长度。设备的连接时间通常比针对同一设备的正常通信请求所花费时间更长。有效范围为 1 到 30 秒。默认值通常为 3 秒钟, 但可能会因驱动程序的具体性质而异。如果驱动程序不支持此设置, 则此设置将被禁用。

● **注意:** 鉴于 UDP 连接的性质, 当通过 UDP 进行通信时, 连接超时设置不适用。

“请求超时”(Request Timeout): 此属性可指定一个所有驱动程序使用的间隔来决定驱动程序等待目标设备完成响应的的时间。有效范围是 50 至 9,999,999 毫秒 (167.6667 分钟)。默认值通常是 1000 毫秒, 但可能会因驱动程序而异。大多数串行驱动程序的默认超时是基于 9600 波特或更高的波特率来确定的。当以较低的波特率使用驱动程序时, 请增加超时, 以补偿获取数据所需增加的时间。

“超时前的尝试次数”: 此属性用于指定在认定请求失败以及设备出错之前, 驱动程序发出通信请求的次数。有效范围为 1 到 10。默认值通常是 3, 但可能会因驱动程序的具体性质而异。为应用程序配置的尝试次数很大程度上取决于通信环境。此属性适用于连接尝试和请求尝试。

定时

“请求间延迟”: 此属性指定驱动程序在将下一个请求发送到目标设备之前等待的时间。它会覆盖设备关联标记的一般轮询频率, 以及一次性读取和写入次数。在处理周转时间慢的设备时, 以及担心网络负载问题时, 这种延迟很有用。为设备配置延迟会影响与通道上所有其他设备的通信。建议用户尽可能将所有需要请求间延迟的设备隔离至单独的通道。其他通信属性 (例如通信序列化) 可以延长此延迟。有效范围是 0 至 300,000 毫秒; 但是, 某些驱动程序可能因某项特别设计的功能而限制最大值。默认值为 0, 它表示对目标设备的请求之间没有延迟。

● **注意:** 不是所有的驱动程序都支持“请求间延迟”。如果不可用, 则此设置不会出现。

设备属性 - 自动降级

自动降级属性可以在设备未响应的情况下使设备暂时处于关闭扫描状态。通过将特定时间段内无响应的设备脱机, 驱动程序可以继续优化与同一通道上其他设备的通信。该时间段结束后, 驱动程序将重新尝试与无响应设备进行通信。如果设备响应, 则该设备会进入开启扫描状态; 否则, 设备将再次开始其关闭扫描时间段。

属性组	自动降级	
常规	故障时降级	启用
扫描模式	降级超时	3
定时	降级期间 (毫秒)	10000
自动降级	降级时放弃请求	禁用
标记生成		

“故障时降级”: 启用后, 将自动对设备取消扫描, 直到该设备再次响应。

● **提示:** 使用 _AutoDemoted 系统标记来监视设备的降级状态, 确定何时对设备取消扫描。

“降级超时”: 指定在对设备取消扫描之前, 请求超时和重试的连续周期数。有效范围是 1 到 30 次连续失败。默认值为 3。

“降级期间”: 指示当达到超时值时, 对设备取消扫描多长时间。在此期间, 读取请求不会被发送到设备, 与读取请求关联的所有数据都被设置为不良质量。当此期间到期时, 驱动程序将对设备进行扫描, 并允许进行通信尝试。有效范围为 100 至 3600000 毫秒。默认值为 10000 毫秒。

“降级时放弃请求”: 选择是否在取消扫描期间尝试写入请求。如果禁用, 则无论是否处于降级期间都始终发送写入请求。如果启用, 则放弃写入; 服务器自动将接收自客户端的写入请求视为失败, 且不会在事件日志中记录消息。

设备属性 - 标记生成

自动生成标记数据库功能可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生成标记数据库，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：自动生成标记数据库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”：启用自动生成标记数据库后，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 - Logix 通信参数

属性组	EtherNet/IP	
常规	TCP/IP 端口	44818
扫描模式	CIP	
定时	连接大小 (字节)	500
自动降级	非活动情况监视器	32
标记生成	Logix	
Logix 通信参数	数组块大小	120
Logix 选项		
Logix 数据库设置		
ENI DF1/DH+/CN 网关通信参...		
槽配置		
冗余		

EtherNet/IP

“TCP/IP 端口”(TCP/IP Port): 指定设备被配置使用的 TCP/IP 端口号。默认值为 44818。

CIP

“连接大小”(Connection Size): 指定 CIP 连接上可用于数据请求和响应的字节数。有效范围为 500 到 4000 个字节。默认值为 500 个字节。

● **注意:** 仅 ControlLogix 5500 和 CompactLogix 5300 设备型号支持此功能。要支持大于 500 个字节的连接大小, 设备必须支持固件版本 20 或更高版本的控制器, 和以太网网桥 EN3x、EN2x 或 En5.x。早期版本的以太网模块 (如 ENBT 和 ENET) 不支持此功能。不符合必需要求的设备会自动回退到 500 个字节的默认设置, 尽管在发生通信故障后重新尝试了所请求的大小。

● **“连接大小”值也可以通过系统标记 “_CIPConnectionSizeRequested” 来请求。有关详细信息, 请参阅[内部标记](#)。**

“非活动情况监视器”(Inactivity Watchdog): 指定连接在由控制器关闭之前保持空闲状态 (不进行读/写事务处理) 的时长 (以秒为单位)。此值越大, 控制器释放连接资源所需的时间越长, 反之亦然。默认值为 32 秒。

● **注意:** 如果在上传项目时频繁出现 CIP 连接超时的错误, 请增加“非活动情况监视器”值。否则, 建议使用默认值。

Logix

“数组块大小”(Array Block Size): 该属性指定在单个事务处理中要读取的最大数组元素数。该值可调, 其范围介于 30 到 3840 个元素之间。默认值为 120 个元素。

● **提示:** 对于布尔型数组, 单个元素将被视为 32 元素位数组。将块大小设置为 30 个元素, 则会转换为 960 个位元素, 而 3840 个元素则会转换为 122880 个位元素。

设备属性 - Logix 选项

属性组	协议选项	
常规	协议模式	逻辑非阻塞
扫描模式	在线编辑后同步	是
定时	离线编辑后同步	是
自动降级	遇 LEN 终止字符串数据	启用
标记生成	项目选项	
Logix 通信参数	默认数据类型	默认值
Logix 选项	性能统计信息	禁用
Logix 数据库设置		
ENI DF1/DH+/CN 网关通信参...		
槽配置		
冗余		

协议选项

“协议模式”: 选择从控制器读取 Logix 标记数据的方式: “逻辑未分块”、“逻辑分块”和“符号”。默认设置为“逻辑未分块”。此选项只能由希望提高客户端/服务器标记更新性能的高级用户来更改。

● 有关详细信息, 请参阅[选择协议模式](#)。

● **注意**: “逻辑未分块”和“逻辑分块”不适用于串行网关型号。

“在线编辑后同步”: 启用后, 当检测到在线项目编辑 (或从 RSLogix/Studio5000 下载项目) 时, 驱动程序会将其项目图像与控制器项目图像同步。此选项可防止在项目更改期间发生不必要的错误。此选项仅在所选“协议模式”为“逻辑未分块”或“逻辑分块”时可用。默认设置为“是”。

“离线编辑后同步”: 启用后, 当检测到离线项目编辑 (或从 RSLogix/Studio5000 下载项目) 时, 驱动程序会将其项目图像与控制器项目图像同步。此选项可防止在项目更改期间发生不必要的错误。此选项仅在所选协议为“逻辑未分块”或“逻辑分块”时可用。默认设置为“是”。

● 如果无法与项目更改同步, 会导致读取和写入错误的原生标记地址。

“LEN 处的终止字符串数据”: 启用后, 每当读取 DATA 成员时, 驱动程序便会自动读取 STRING 结构的 LEN 成员。DATA 字符串将在遇到的第一个空字符 (其位置等于 LEN 的值) 或 DATA 的最大字符串长度处终止 (以先发生者为准)。禁用后, 驱动程序会绕过 LEN 成员, 并在遇到的第一个空字符或 DATA 的最大字符串长度处终止 (以先发生者为准)。因此, 如果通过外部源缩短 LEN 而不修改 DATA, 则驱动程序不会根据此缩短后的长度终止 DATA。默认设置为“启用”。

“项目选项”

“默认数据类型”: 在添加/修改/导入标记期间选择默认类型的情况下, 选择分配给客户端/服务器标记的数据类型。默认设置为“默认”。

● 有关详细信息, 请参阅[默认数据类型条件](#)。

“性能统计信息”: Allen-Bradley ControlLogix Ethernet 驱动程序 能够收集通信统计信息, 以帮助确定驱动程序的性能。启用后, 此选项将被启用。驱动程序会跟踪客户端/服务器标记更新的数量和类型。重新启动服务器应用程序时, 结果会显示在服务器的“事件日志”中。默认值为“否”。

● **注意**: 将项目配置设定为以期达到最佳性能后, 建议用户禁用“性能统计信息”。由于统计信息会在关闭时写入“事件日志”, 因此必须重新启动服务器才能查看结果。

● 另请参阅: [检测“控制器项目”中所做的更改](#)

设备属性 - Logix 数据库设置

属性组	数据库导入方法	
常规	数据库导入方法	从设备创建
扫描模式	标记导入文件	*.l5k
定时	标记说明	启用
自动降级	Logix 数据库选项	
标记生成	限制名称长度	禁用
Logix 通信参数	标记层次	已扩展
Logix 选项	Logix 数据库筛选	
Logix 数据库设置	强制数组限制	禁用
ENI DF1/DH+/CN 网关通信参...	数组计数上限	2000
槽配置		
冗余		

数据库导入方法

“数据库导入方法”：选择应如何填充标记数据库：

- “从设备创建”：通过用于数据访问的以太网连接直接从控制器检索标记，这样可以快速导入大多数标记，但需要访问控制器且不会导入说明。未导入的标记包括附加指令 (AOI) InOut 属性。
● 注意：此功能不适用于串行网关型号。
- “从导入文件创建”：从所选的 RSLogix L5K/L5X 文件中检索标记。无需访问控制器即可导入说明，且用户可以离线工作；但是，此选项较慢，不会导入控制器中的所有标记。未导入的标记包括：
 - I/O 标记
 - 附加指令 (AOI) InOut 属性
 - 给定其他属性别名的 AOI 属性
 - 从其他“设备阶段”或“程序”给定属性别名的“设备阶段”属性
 - 从其他“程序”或“设备阶段”给定属性别名的“程序”属性。
 - 计时器/计数器 CTL 位

标记导入文件：单击“浏览 (...)”按钮查找并选择要导入标记的 L5K/L5X 文件。当系统指示“自动生成标记数据库”创建标记数据库时，将使用此文件。所有标记，包括“全局”和“程序”，将根据其各自的数据类型进行导入并展开。

标记说明：选择“启用”导入非结构、非数组标记的标记说明。如有必要，给定长名称标记的说明，以说明原始标记名称。

使用 API 生成标记

CLX 离线 ATG 的“配置 API”属性为：

```
"controllogix_ethernet.DEVICE_DATABASE_IMPORT_METHOD": 1, "controllogix_ethernet.DEVICE_TAG_IMPORT_FILE": "myFile.l5x", "controllogix_ethernet.DEVICE_DISPLAY_DESCRIPTIONS": true,
```

其中，导入方法的枚举为：
0 表示从设备创建；
以及
1 表示从文件创建

“Logix 数据库选项”

“限制名称长度”：设置为“启用”以将标记和组名称长度限制在 31 个字符内。默认设置为“禁用”。

- 在 OPC 服务器版本 4.70 之前，标记和组名称长度被限制在 31 个字符内。当前 256 个字符长度限制适合 Logix 40 字符 Logix 标记名称。

2. 如果使用之前版本的服务器通过 L5K/L5X 导入了标记，请检查“事件日志”或扫描服务器项目以查看是否曾因字符限制而截断标记。如果是这样，请启用此特性以保留服务器标记名称。OPC 客户端标记参考不受影响。如果禁用，则会创建较长的标记名称，并且参考已修剪标记的客户端必须更改为参考新的标记名称。
3. 如果使用之前版本的 OPC 服务器通过 L5K/L5X 导入了标记，且未曾因 31 字符限制而截断任何标记，则保留禁用此选项。
4. 如果使用 4.70 版本或更高版本的服务器通过 L5K/L5X 导入了标记，则保留禁用此项。

●另请参阅：[控制器服务器名称转换](#)

“标记层次”：此属性用于指定标记层次的树组织。当处于“已压缩”模式时，自动生成标记所创建的服务器标记遵循的组/标记层次与标记地址一致。会为每个段（前面带有句点）创建组。当处于“已扩展”模式时，自动生成标记所创建的服务器标记遵循的组/标记层次与 RSLogix 5000 中的标记层次一致。会为每个段（前面带有句点）创建组，但也会创建组来表示逻辑分组。要使用此功能，请在[标记生成](#)属性中启用“允许子组”。

●有关所创建组的详细信息，请参阅[标记层次结构](#)和[控制器到服务器名称转换](#)。

Logix 数据库筛选

“强加数组限制”：选择“启用”来限制数组元素数。可以将控制器中的标记声明为具有很大的数组维度。默认情况下，数组在标记生成过程中会完全展开，这对大数组来说非常耗时。通过强加限制，只生成来自每个维度的指定数量的元素。限制仅在数组维度大小超出限制时生效。默认设置为“禁用”。

“数组计数上限”：指定数组计数限制。默认值为 2000。

设备属性 - ENI DF1/DH+/CN 网关通信参数

属性组	ENI DF1/DH+/CN 网关通信参数	
常规	TCP/IP 端口	44818
扫描模式	请求大小 [字节数]	232
定时	允许函数文件块写入	禁用
自动降级		
标记生成		
Logix 通信参数		
Logix 选项		
Logix 数据库设置		
ENI DF1/DH+/CN 网关通...		
槽配置		
冗余		

“TCP/IP 端口”：指定对远程设备进行配置后，该远程设备所使用的端口号（例如 1756-ENBT）。默认值为 44818。

“请求大小”：为优化性能，选择一次可以从属设备请求的字节数。选项为 32、64、128 或 232。默认值为 232 个字节。

“允许函数文件块写入”：函数文件是基于结构的文件（非常类似于 PD 和 MG 数据文件），并且是 MicroLogix 1100、1200 和 1500 所特有的。对于适用的功能文件，可以在单个操作中将数据写入设备。默认情况下，当数据写入功能文件子元素（功能文件结构中的字段）时，将立即对该标记进行写操作。对于 RTC 文件之类的文件，其子元素包括小时（HR）、分钟（MIN）和秒（SEC），并不是始终都能接受分别写入。由于这些子元素仅依赖于时间，因此必须在一个操作中写入值，以避免子元素写入之间的时间流逝。因此，提供了对这些子元素进行块写入的选项。默认设置为禁用状态。

●有关详细信息，请参阅[块写入](#)和[功能文件](#)。

块写入

块写入包括在单次写入操作中将功能文件中每个读/写子元素的值写入设备。不需要在执行块写入操作之前写入各个子元素。不受影响 (写入) 的子元素的当前值将写回到它们。例如, 如果当前 (上次读取) 日期和时间 为 1/1/2001, 12:00.00, DOW = 3 且小时更改为 1 点, 则写入到设备的值将为 1/1/2001, 1:00.00, DOW = 3。有关详细信息, 请参阅以下说明。

- 1. 首先, 在“设备属性”中找到“ENI DF1/DH+/CN 网关通信参数”。
- 2. 启用“允许功能文件块写入”, 以通知驱动程序对支持块写入的功能文件进行块写入。
- 3. 单击“确定”或“应用”。
- 4. 将所需值写入所讨论的子元素标记。子元素标记将立即采用写入其中的值。

● **注意:** 在块写入模式下至少写入一次子元素后, 标记的值将源自驱动程序的写入缓存而非控制器。块写入完成后, 所有子元素标记值均源自控制器。

- 5. 写入全部所需子元素后, 执行块写入操作以将这些值发送到控制器。要实例化块写入, 请引用标记地址 *RTC:<元素>_SET*。将此标记的值设置为 “true” 后, 将基于当前 (上次读取) 的子元素和受影响 (写入) 的子元素进行块写入。将标记设置为 “true” 后, 该值会立即自动重置为 “false”。这是默认状态, 不会执行任何操作。

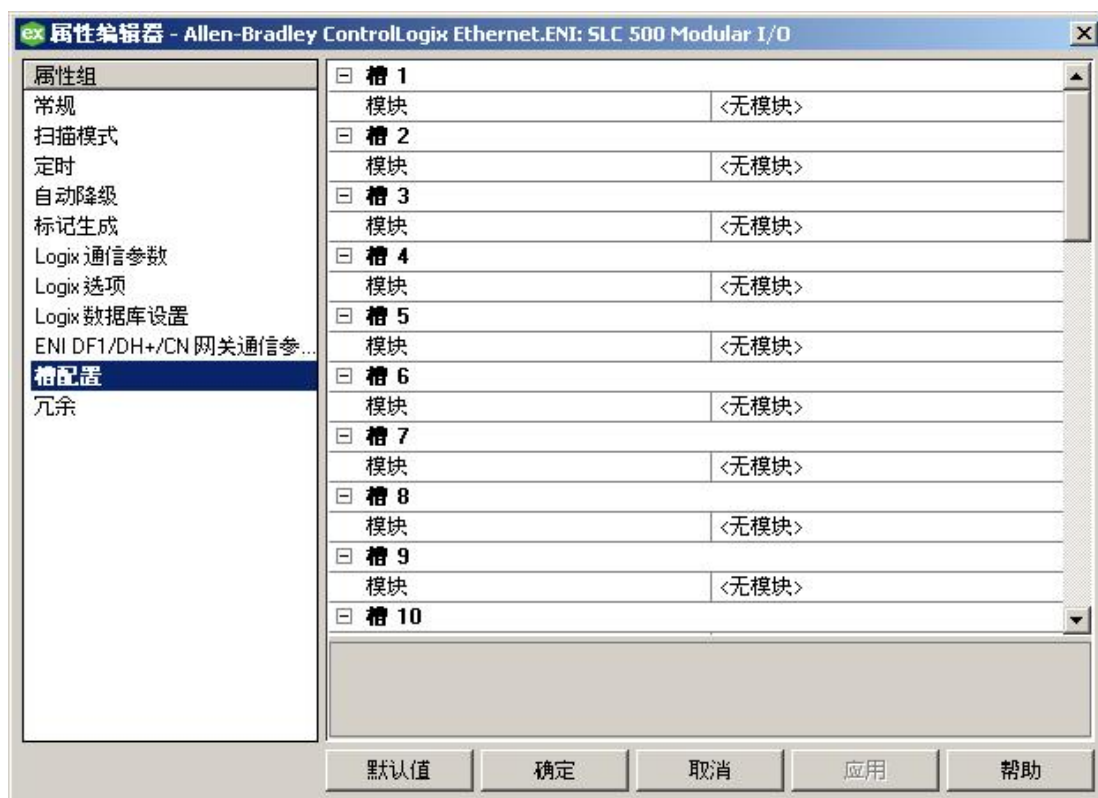
适用的功能文件/子元素

RTC	
年	YR
月	MON
日	DAY
星期	DOW
小时	HR
分钟	MIN
秒	SEC

● 另请参阅: [功能文件列表](#)

设备属性 - SLC 500 插槽配置

对于要访问的 I/O, 必须将 SLC5/01/02/03/04/05 型号 (模块化 I/O 机架) 配置为与 Allen-Bradley ControlLogix Ethernet 驱动程序 配合使用。每个设备最多可配置 30 个插槽。



插槽 n: 配置的物理插槽。使用加号图标展开属性。

“模块”: 从下拉列表的可用选项中选择插槽中的 模块类型。

要通过 [Configuration API 服务配置插槽](#)，请参阅本示例。

“输入字”(Input Words): 如果所选模块需要，则输入此模块的“输入字”(Input Words) 的最大数量。

“输出字”(Output Words): 如果所选模块需要，则输入此模块的“输出字”(Output Words) 的最大数量。

要使用插槽配置，请按照以下说明进行操作：

1. 通过单击模块列表框中的行，选择要配置的插槽。
2. 要选择模块，请从可用模块下拉列表中单击相应模块。
3. 如有必要，请配置“输入字”和“输出字”。
4. 要移除插槽/模块，请从可用模块下拉列表中选择“无模块”。
5. 完成后，单击“确定”。

提示：

- 使用“0000-类属模块”(0000-Generic Module) 可对未包含在“可用模块”列表中的 I/O 进行配置。
- 可用的模块选择与 Allen Bradley APS 软件中的相同。

注意：机架中通常具有不包含物理模块的开放插槽。要正确访问真正包含模块的各个插槽的数据，先前的模块必须具有正确的映射字数。例如，如果只对插槽 3 中的 I/O 感兴趣，但插槽 1 和 2 包含 I/O 模块，则必须从此插槽配置组中为插槽 1、2 和 3 选择正确的模块。

“0000-类属模块”

使用“类属模块”映射未在可用模块列表中显示的模块的“输入”和“输出”字。要正确使用“类属模块”，用户必须知道每个模块所需的“输入”和“输出”字数。

请参阅 [Allen-Bradley I/O 用户手册文档](#)，以确定输入和输出要求，并注意针对 1 类或 3 类操作的要求可能不同。

有关每个 I/O 模块可用的输入和输出字数的信息，请参阅 [模块化 I/O 选择指南](#)。

Configuration API - 插槽配置

本例中的插槽配置：

- 插槽 1: 1746-OV32 32 输出 [发送器](接收器) 10/50 V_{DC}
- 插槽 2: 1746-I*16 任意 16 点离散量输入模块
- 插槽 3: <无模块>
- 插槽 4: 0000-类属模块 [1 个输入字, 2 个输出字]
- 插槽 5: 1746-NT8 模拟 8 通道热电偶输入
- 插槽 6 - 30: <无模块>

使用 API 配置插槽：

```
"controllogix_ethernet.DEVICE_SLOT_CONFIGURATION": { "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_MODULE": [ 30, 1, 255, 62, 76, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255 ], "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_INPUT_WORDS": [ 0, 1, 0, 1, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ], "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_OUTPUT_WORDS": [ 2, 0, 0, 2, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] }
```

设备属性 - 冗余

属性组

常规

扫描模式

定时

自动降级

冗余

冗余

次级路径	...
操作模式	故障切换
监视器项目	
监视器间隔 (秒)	300
尽快返回至主要设备	是

Media-Level Redundancy 插件提供冗余。
有关详细信息，请参阅网站、向销售代表咨询或查阅用户手册。

SLC 500 模块化 I/O 选择指南

下表列出了“插槽配置”列表中每个 I/O 模块可用的输入和输出字数。

模块 ID	模块类型	输入字	输出字
0	1746-I*8 任意 8 点离散量输入模块	1	0
1	1746-I*16 任意 16 点离散量输入模块	1	0
2	1746-I*32 任意 32 点离散量输入模块	2	0
3	1746-O*8 任意 8 点离散量输出模块	0	1
4	1746-O*16 任意 16 点离散量输出模块	0	1
5	1746-O*32 任意 32 点离散量输出模块	0	2
6	1746-IA4 4 输入 100/120 VAC	1	0
7	1746-IA8 8 输入 100/120 VAC	1	0
8	1746-IA16 16 输入 100/120 VAC	1	0
9	1746-IB8 8 输入 (接收器) 24 VDC	1	0
10	1746-IB16 16 输入 (接收器) 24 VDC	1	0
11	1746-IB32 32 输入 (接收器) 24 VDC	2	0

模块 ID	模块类型	输入字	输出字
12	1746-IG16 16 输入 [TTL] (源) 5 VDC	1	0
13	1746-IM4 4 输入 200/240 VAC	1	0
14	1746-IM8 8 输入 200/240 VAC	1	0
15	1746-IM16 16 输入 200/240 VAC	1	0
16	1746-IN16 16 输入 24 VAC/VDC	1	0
17	1746-ITB16 16 输入 [快速] (接收器) 24 VDC	1	0
18	1746-ITV16 16 输入 [快速] (源) 24 VDC	1	0
19	1746-IV8 8 输入 (源) 24 VDC	1	0
20	1746-IV16 16 输入 (源) 24 VDC	1	0
21	1746-IV32 32 输入 (源) 24 VDC	2	0
22	1746-OA8 8 输出 (双向可控硅三极管) 100/240 VAC	0	1
23	1746-OA16 16 输出 (双向可控硅三极管) 100/240 VAC	0	1
24	1746-OB8 8 输出 [发送器] (源) 10/50 VDC	0	1
25	1746-OB16 16 输出 [发送器] (源) 10/50 VDC	0	1
26	1746-OB32 32 输出 [发送器] (源) 10/50 VDC	0	2
27	1746-OBP16 16 输出 [发送器 1 安培] (SRC) 24 VDC	0	1
28	1746-OV8 8 输出 [发送器] (接收器) 10/50 VDC	0	1
29	1746-OV16 16 输出 [发送器] (接收器) 10/50 VDC	0	1
30	1746-OV32 32 输出 [发送器] (接收器) 10/50 VDC	0	2
31	1746-OW4 4 输出 [继电器] VAC/VDC	0	1
32	1746-OW8 8 输出 [继电器] VAC/VDC	0	1
33	1746-OW16 16 输出 [继电器] VAC/VDC	0	1
34	1746-OX8 8 输出 [隔离继电器] VAC/VDC	0	1
35	1746-OVP16 16 输出 [发送器 1 安培] (接收器) 24 VDC3	0	1
36	1746-IO4 2 输入 100/120 VAC 2 输出 [继电器] VAC/VDC3	1	1
37	1746-IO8 4 输入 100/120 VAC 4 输出 [继电器] VAC/VDC4	1	1
38	1746-IO12 6 输入 100/120 VAC 6 输出 [继电器] VAC/VDC	1	1
39	1746-NI4 4 通道模拟输入	4	0
40	1746-NIO4I 模拟梳状 2 输入和 2 电流输出	2	2
41	1746-NIO4V 模拟梳状 2 输入和 2 电压输出	2	2
42	1746-NO4I 4 通道模拟电流输出	0	4
43	1746-NO4V 4 通道模拟电压输出	0	4
44	1746-NT4 4 通道热电偶输入模块	8	8
45	1746-NR4 4 通道 Rtd/电阻输入模块	8	8
46	1746-HSCE 高速计数器/编码器	8	1
47	1746-HS 单轴运动控制器	4	4
48	1746-OG16 16 输出 [TLL] (接收器) 5 VDC	0	1
49	1746-BAS 基本模块 500 5/01 配置	8	8
50	1746-BAS 基本模块 5/02 配置	8	8
51	1747-DCM 直接通信模块 (1/4 机架)	2	2
52	1747-DCM 直接通信模块 (1/2 机架)	4	4
53	1747-DCM 直接通信模块 (3/4 机架)	6	6
54	1747-DCM 直接通信模块 (整个机架)	8	8
55	1747-SN 远程 I/O 扫描器	32	32
56	1747-DSN 分布式 I/O 扫描器 7 块	8	8

模块 ID	模块类型	输入字	输出字
57	1747-DSN 分布式 I/O 扫描器 30 块	32	32
58	1747-KE 接口模块, 系列 A	1	0
59	1747-KE 接口模块, 系列 B	8	8
60	1746-NI8 8 通道模拟输入, 类 1	8	8
61	1746-NI8 8 通道模拟输入, 类 3	16	12
62	0000-类属模块	-	-
63	1746-IC16 16 输入 (接收器) 48 VDC	1	0
64	1746-IH16 16 输入 [发送器] (源) 125 VDC	1	0
65	1746-OAP12 12 输出 (双向可控硅三极管) 120/240 VDC	0	1
66	1746-OB6EI 6 输出 [发送器] (源) 24 VDC	0	1
67	1746-OB16E 16 输出 [发送器] (源) 受保护	0	1
68	1746-OB32E 32 输出 [发送器] (源) 10/50 VDC	0	2
69	1746-OBP8 8 输出 [发送器 2 安培] (源) 24 VDC	0	1
70	1746-IO12DC 6 输入 12 VDC, 6 输出 [继电器]	1	1
71	1746-INI4I 模拟 4 通道隔离电流输入	8	8
72	1746-INI4VI 模拟 4 通道隔离电压/电流输入	8	8
73	1746-INO4I 模拟 4 通道隔离电流输出	8	8
74	1746-INO4VI 模拟 4 通道隔离电压/电流输出	8	8
75	1746-INT4 4 通道隔离热电偶输入	8	8
76	1746-NT8 模拟 8 通道热电偶输入	8	8
77	1746-HSRV 运动控制模块	12	8
78	1746-HSTP1 步进控制器模块	8	8
79	1747-MNET MNET 网络通信模块	0	0
80	1746-QS 已同步的轴模块	32	32
81	1747-QV 开环速度控制	8	8
82	1747-RCIF 机器人控制接口模块	32	32
83	1747-SCNR ControlNet SLC 扫描器	32	32
84	1747-SDN DeviceNet 扫描器模块	32	32
85	1394-SJT GMC 涡轮系统	32	32
86	1203-SM1 SCANport 通信模块 - 基本	8	8
87	1203-SM1 SCANport 通信模块 - 增强型	32	32
88	AMCI-1561 AMCI 系列 1561 解析器模块	8	8

性能优化

有关通信和应用程序级别的优化的详细信息, 请从下表中选择一个链接。

[优化通信](#)

[优化应用程序](#)

[性能统计信息和调优](#)

[性能调优示例](#)

优化通信

与任何可编程控制器一样, 可使用多种方法来增强性能和系统通信。

协议模式

协议模式可确定如何从控制器访问 Logix 标记数据。存在三种类型的协议模式: “符号”、“逻辑非阻塞”和“逻辑阻塞”。说明如下:

- **“符号”模式**: 每个客户端/服务器标记地址在数据包中由其 ASCII 字符名称表示。
- **“逻辑非阻塞”模式**: 每个客户端/服务器标记由 PLC 中的逻辑内存地址表示。
- **“逻辑阻塞”模式**: Logix 标记将作为单个数据块进行访问。每个客户端/服务器标记 (如 MYTIMER.ACC) 都有相应的 Logix 标记 (MYTIMER)。对于结构, 多个客户端/服务器标记可属于同一 Logix 标记。在每个只读周期读取 Logix 标记后, 其块会在驱动程序缓存中更新, 并且所有客户端/服务器标记会在此缓存中更新。

通常建议使用“逻辑非阻塞”模式, 原因在于该模式是收集和处理 Logix 标记数据的最有效模式。想要获得向后兼容性, 建议使用“符号”模式, 而对于包含少量 UDT 引用项目和/或预定义结构 Logix 标记, 建议使用“逻辑非阻塞”模式。虽然“逻辑阻塞”模式非常高效, 但如果使用不正确, 也会降低性能。有关每种模式优势和劣势的详细信息, 请参阅[选择协议模式](#)。

标记划分提示

用户应针对“逻辑阻塞”目的指定一个或多个设备, 并针对“逻辑非阻塞”目的指定一个或多个设备。这有助于提高性能, 原因在于项目中的不同标记通常更适用于不同模式。利用标记划分时, 用户应执行以下操作:

1. 将引用原子型 Logix 标记的服务器标记 (数组或非数组) 标记分配给“逻辑非阻塞”设备。
2. 将引用结构 Logix 标记 (由三分之一* 或更小比例的结构标记组成) 的服务器标记分配给“逻辑非阻塞”设备。例如, 如果引用 PID_ENHANCED Logix 标记的成员标记达到或低于 55**, 则应将所有这些标记分配给“逻辑非阻塞”设备。
3. 将引用结构 Logix 标记 (由三分之一* 或更大比例的结构标记组成) 的服务器标记分配给“逻辑阻塞”设备。例如, 如果引用 PID_ENHANCED Logix 标记的成员标记超过 55**, 则应将所有这些标记分配给“逻辑阻塞”设备。

*三分之一并不是一个精确的限制值, 而是一个经大量研究验证过的数字。

**PID_ENHANCED 结构包含 165 个标记, 因此三分之一等于 55 个标记。

连接大小

增大“连接大小”可以增加每个数据包的读取/写入请求数量, 从而提高吞吐量。尽管它也会增加 CPU 载荷和响应周转时间, 但是可显著提高性能。只能在 ControlLogix 5500 和 CompactLogix 5300 设备型号中对“连接大小”属性进行修改。有关详细信息, 请参阅[Logix 通信参数](#)。

UDT 子结构别名使用

如果 UDT 包含大型子结构, 并且客户端中引用了三分之一或更大比例的子结构成员, 请参阅下面的说明以优化子结构的读取。

1. 在 RSLogix 5000 中创建子结构的别名。然后, 将引用其余 UDT 子结构的服务器标记分配给“逻辑阻塞”设备。
2. 接下来, 将引用其余 UDT (但非子结构) 的服务器标记分配给“逻辑非阻塞”设备。

系统开销时间片

系统开销时间片 (SOTS) 是为执行通信任务 (如 OPC 驱动程序通信) 分配的时间百分比, 该值在 RSLogix 5000 中进行设置。100% SOTS 是控制器任务 (如梯形图) 的时间百分比。默认 SOTS 为 10%。在发生的每个 10 ms 程序扫描过程中, 控制器花费 1 ms 来处理驱动程序请求 (如果控制器有连续任务)。SOTS 的值可定义任务的优先级。如果控制器任务的优先级较高, 则应将 SOTS 设置为小于 30% 的值。如果通信任务的优先级较高, 则应将 SOTS 设置为大于或等于 30% 的值。为了在通信性能与 CPU 使用率之间达到最佳平衡状态, 请将 SOTS 设置为介于 10% 至 40% 之间的值。

多请求数据包

Allen-Bradley ControlLogix Ethernet 驱动程序 旨在优化读取和写入操作。对于非数组、非字符串标记 (仅请求一个元素), 请求会被锁定到单个事务。与单个标记事务处理相比, 这可显著提高性能。唯一的限制是需要调整单个事务处理的数据字节数。

● **重要事项**: 在“符号”模式中, 系统会将每个标记的 ASCII 字符串值插入到请求包中, 直到不存在合适的标记请求为止。为了获得最佳性能, 用户应保持标记名称包含的字符数最少。标记名称包含的字符数越少, 适合单个事务的标记就越多, 并且用于处理所有标记的事务越少。

已分块的数组元素 (仅限“符号”和“逻辑非阻塞”模式)

要优化原子型数组元素的读取，请在单个请求中以块的形式读取数组，而不是单独读取。每个块中读取的元素越多，性能越好。由于事务处理费用较高且处理会消耗大多数时间，因此，在尽可能多地扫描所需标记的同时要尽可能少地进行事务处理。此为数组元素块的实质。

规定块大小为元素计数。块大小为 120 个元素，意味着在一个请求中最多能够读取 120 个数组元素。块大小最大为 3840 个元素。布尔型数组的处理方式会不同：在协议中，布尔型数组是一个 32 位数组。因此，请求元素 0 将请求位 0 至 31。为了保持讨论的一致性，会将布尔型数组元素视为单个位。可请求的数组元素最大数量（基于块大小 3840）汇总如下：122880 BOOL、3840 SINT、3840 INT、3840 DINT、3840 REAL、3840 LINT、3840 UINT、3840 USINT、3840 ULINT、3840 LREAL。

正如 [Logix 通信参数](#) 中所述，块大小是可调整的，且应该根据当前项目进行选择。例如，如果数组元素 0-26 和元素 3839 是要读取的标记，那么，使用 3840 的块大小不仅过剩，而且影响驱动程序的性能。这是因为：尽管只需要 28 个元素，但是会针对每个请求读取 0 和 3839 之间的所有元素。在这种情况下，块大小设为 30 更合适。在一个请求中读取元素 0-26，而在下一个请求中读取元素 3839。

优化字符串

在“逻辑寻址”模式下，写入 STRING.DATA 还将写入至具有合适长度值的 STRING.LEN。

LEN 处的终止字符串数据

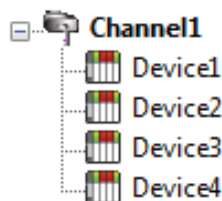
在此驱动程序中，字符串标记为包含单独字符数据和长度组件的结构。在这种情况下，驱动程序自动读取以下两个事务中的字符串标记：一个为字符串字符数据 (DATA) 在“逻辑协议模式”下的事务，另一个为字符串长度 (LEN) 在“符号”模式下的事务。当禁用“LEN 处的终止字符串数据”选项时，单个事务会读取字符串字符数据。在这种情况下，将跳过读取字符串长度的“符号”模式。在带有多个字符串标记的项目中，这种情况可显著减少读取所有标记所需的时间。

有关“遇 LEN 终止字符串数据”选项的详细信息，请参阅 [Logix 选项](#)。

优化应用程序

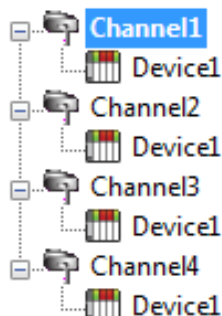
Allen-Bradley ControlLogix Ethernet 驱动程序旨在提供最佳性能，使得其对系统的整体性能影响最小。即使 Allen-Bradley ControlLogix Ethernet 驱动程序速度很快，也可以利用一系列指南来优化应用程序，并获得最佳性能。

服务器将诸如 Allen-Bradley ControlLogix Ethernet 等通信协议称为通道。应用程序中定义的每个通道都表示服务器中一个单独的执行路径。一旦定义了通道，则必须在该通道下定义一系列设备。每一个此类设备都代表一个可从中收集数据的 Allen-Bradley Logix CPU。虽然这种定义应用程序的方法提供了高水平的性能，但它不能充分利用 Allen-Bradley ControlLogix Ethernet 驱动程序或网络。以下是使用单一通道配置应用程序时的示例。



每个设备均出现在单个通道下(称为“Ethernet_Channel1”)。在此配置中，驱动程序需要尽快从一个设备移动到下一个设备，以便更高效地收集信息。随着更多设备的添加或从单个设备请求的信息的增加，整体更新速率会受到不利影响。

如果驱动程序只能定义一个单通道，则上述示例将是唯一可用的选项；但是，驱动程序最多可以定义 1024 个通道。使用多个通道，可通过同时向网络发出多个请求来分发数据集合工作载荷。下面显示了使用多个通道来提高性能时相同应用程序所呈现效果的示例。



当前，每个设备已在其自身的通道下定义。在这个新配置中，单个执行路径专用于从每个设备收集数据。如果应用程序的设备数较小，则可对其进行优化，如此处所示。

即使应用程序设备数较大，也可改善性能。虽然设备数较小可能是理想情况，但附加通道仍会对应用程序有益。尽管在全部通道上分散设备载荷会使服务器再次从一个设备移动到另一个设备，但是，它现在可以用极少的设备在单通道上进行处理。

性能统计信息和调优

“性能统计信息”功能提供了与应用程序性能有关的基准和统计信息。由于“性能统计信息”是处理的附加层，因此会影响服务器性能。在这种情况下，默认状态为“关闭”。要启用“性能统计信息”功能，请访问“设备属性”并设置**“Logix 选项”**以使用**“启用性能统计信息”**。

性能统计信息类型

性能统计信息提供了涵盖三个范围 (设备、通道和驱动程序) 的有意义的数值结果。类型说明如下：

- **设备：**这些统计信息提供了特定设备上的数据访问性能。
- **通道：**这些统计信息提供了启用性能统计信息的给定通道下所有设备的平均数据访问性能。
- **驱动程序：**这些统计信息提供了使用 Allen-Bradley ControlLogix Ethernet 驱动程序 且已启用性能统计信息的所有设备的平均数据访问性能。

选择统计信息类型

所需的统计信息类型取决于应用程序。通常情况下，驱动程序统计信息提供了应用程序性能的真实度量，而通道和设备统计信息在调优应用程序时最为相关。例如，将设备 A 中的 10 个特定标记移动到设备 B 中，一定会提高设备 A 的性能吗？将设备 A 从通道 1 移动至通道 2 会提高通道 1 的性能吗？这些问题都是使用设备和通道统计信息的情况的很好的示例。

找到统计信息

服务器统计信息在关闭时输出到服务器“事件日志”。要查看结果，请关闭服务器并重新启动。

服务器统计信息和性能统计信息之间的差异

性能统计信息提供已执行的读取类型组合 (例如符号型、符号实例与物理型，或者设备读取与缓存读取)，而服务器统计信息则提供一般读取计数值。

调优应用程序以提高性能

● 有关提高设备和通道统计信息结果的信息，请参阅下述说明。有关详细信息，请参阅[优化通信](#)。

- 应将引用原子型 Logix 标记的服务器标记 (数组或非数组) 的标记应分配给“逻辑非阻塞”设备。
- 应将引用结构 Logix 标记 (由三分之一* 或更小比例的结构标记组成) 的服务器标记分配给“逻辑非阻塞”设备。
- 应将引用结构 Logix 标记 (由三分之一* 或更大比例的结构标记组成) 的服务器标记分配给“逻辑阻塞”设备。
- 使用“符号模式”时，应保持 Logix 名称长度最小。
- 应尽可能经常使用 Logix 数组。
- 应仅为梯形图/FBD 分配必要量的系统开销时间片，以将其余部分用于驱动程序通信。
- 对于在“逻辑模式”下读取大量字符串标记的项目，禁用“设备属性”中“Logix 选项”下的“LEN 处的终止字符串数据”选项。

● 有关提高驱动程序统计信息结果的信息，请参阅下述说明。有关详细信息，请参阅[优化应用程序](#)。

- 设备应分布在不同的通道上。除非有必要，否则不应将多个设备放在一个通道上。
- 载荷应均匀分布在设备上。除非有必要，否则单个设备不应过载。
- 不同设备之间不应引用相同的 Logix 标记。

● **注意：**尽管这些一般规则可以优化性能，但最终还要取决于应用程序。扫描速率会使结果模糊：如果标记请求为轻量级，则在出现下一个请求之前就可以完成读取和写入事务。在这种情况下，“逻辑阻塞”和“逻辑非阻塞”的“性能统计信息”结果相同。如果标记请求为重量级 (很多标记或扫描速率高)，则完成事务的时间可能会延长。此时，“逻辑阻塞”和“逻辑非阻塞”的优势和劣势皆很明显。性能统计信息可以帮助优化应用程序以实现性能最大化。有关示例，请参阅[性能调优示例](#)。

性能调优示例

统计信息可应用于任何应用程序。在下列示例中，Quick Client 用于性能调优过程中。其目的是使得项目中使用的**所有标记都能够以很快的扫描速率同时进行读取**。虽然这不现实，但却为服务器中的项目布局提供了准确的基准 (标记属于特定设备，设备属于特定通道等)。

收集的统计信息是相对的。用户应先进行服务器项目布局，收集统计信息，然后再进行调优。建议进行多次试用来正确评估给定布局的结果。确定最有效的布局后，再次确认服务器已处于最佳状态即可构建客户端应用程序。

● 通过 **Quick Client** 获得的性能结果不能等同于通过客户端应用程序获得的性能结果：多种因素会产生差异。虽然使用客户端应用程序进行性能调优比使用 **Quick Client** 更为准确，但所需的调优不仅会影响到服务项目，还会影响到客户端应用程序。建议在开发客户端应用程序之前先使用 **Quick Client** 对应用程序进行调优。

● **注意：**下述调优过程假定所有标记都以很快的扫描速率进行读取。写入操作会影响性能。

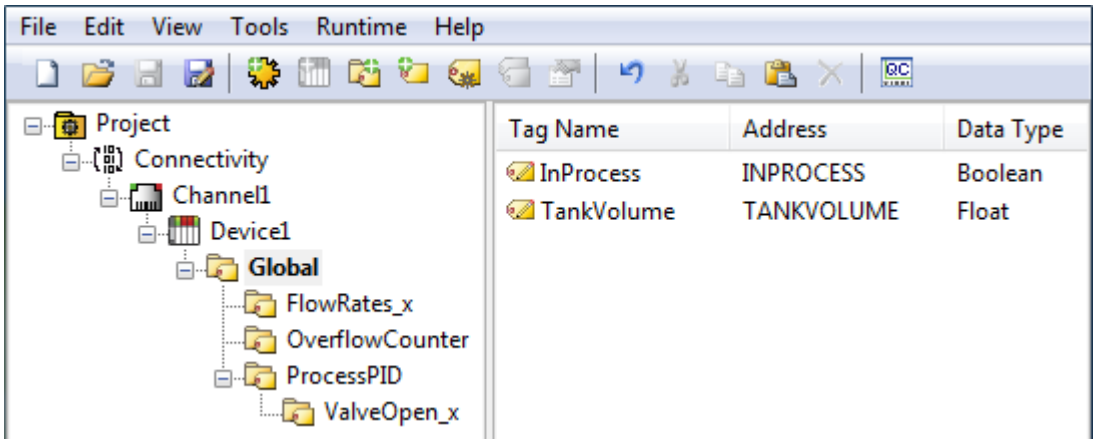
1. 下方显示的控制器项目中包含以下内容：

- 2 个原子型
- 1 个原子型数组
- 1 个 UDT
- 1 个 UDT 数组
- 1 个预定义类型

● **注意：**开销时间片 (OTS) = 10%。

	P	Tag Name	Alias For	Base Tag	Type
		InProcess			BOOL
	<input type="checkbox"/>	+OverflowCounter			COUNTER
	<input type="checkbox"/>	+ValveOpen			DINT[10]
	<input type="checkbox"/>	+ProcessPID			PIDLoop
	<input type="checkbox"/>	+FlowRates			ProcessVariable[2]
	<input type="checkbox"/>	TankVolume			REAL
*	<input type="checkbox"/>				

2. 在此控制器中执行完“自动标记数据库生成”后，服务器会生成下列项目。



Tag Name	Address	Data Type
InProcess	INPROCESS	Boolean
TankVolume	TANKVOLUME	Float

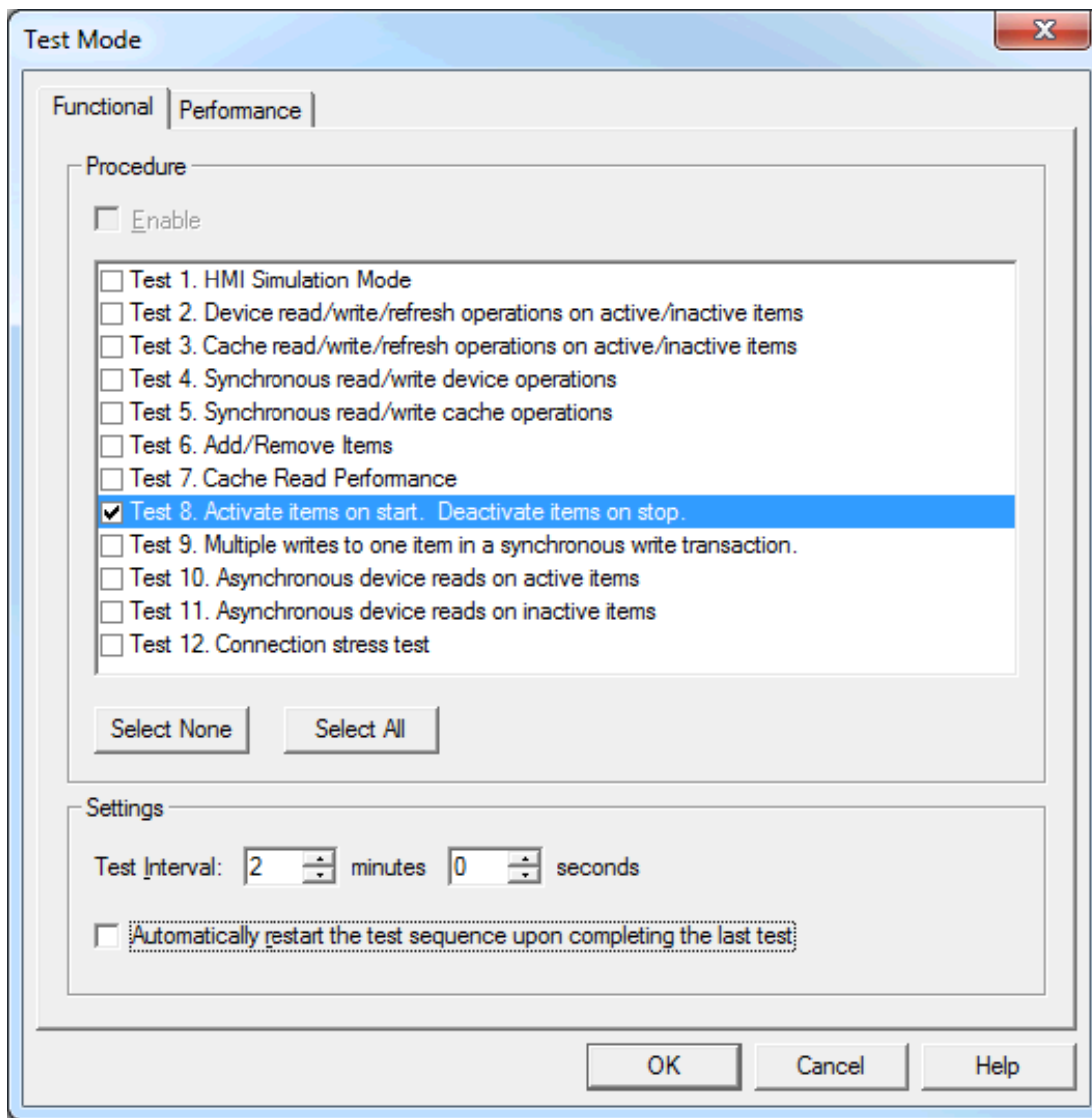
● **注意：**“全局”标记组包含 130 个标记。

- 为说明标记划分的优势，本示例未引用所有标记。引用了多于三分之一的 **ProcessPID** 标记、少于三分之一的 **FlowRates** 标记以及所有其他标记。在这种情况下，新的标记计数是 105。
- 准备客户端以进行测试。为此，请单击 **QuickClient** 图标以从服务器中启动 **Quick Client**。
- 项目加载后，除了包含需要用到的标记的组外，移除其他所有组。例如，不需要“统计信息”和“系统”标记。

● **注意：**对于小项目，将组的“更新速率”设置为 0-10 毫秒。对于大型项目，将速率设置为 10-50 毫秒。

6. 选择“工具”|“测试模式”。
7. 启用测试 8。启动时激活项目。停止时禁用项目，然后设置测试间隔。

● **注意：**由于此项目相当小，因此已将间隔设置为 2 分钟。对于较大的项目，应增加间隔以使读取更加准确。



8. 返回至“工具”|“测试模式”，然后禁用测试模式。应禁用所有标记。
9. 断开 Quick Client，以开始试用时间。
10. 关闭服务器。
11. 启动服务器并在“设备属性”下将“协议模式”设置为“逻辑分块”。
12. 将“性能统计信息”设置为“是”。



13. 使用 Quick Client 连接至服务器。选择“工具”|“测试模式”(Test Mode)。启用测试模式。
 ● **注意：**数据读取开始。测试间隔到期后，所有标记都将被禁用，而驱动程序将停止统计信息收集。随后即可查看结果。
14. 断开 Quick Client 与服务器的连接，然后关闭服务器。
15. 重新启动服务器并搜索其事件日志以获取统计信息。下图显示了对设备使用“逻辑分块”的第一次试用。

```

DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5972.26
    
```

- **注意：**下图显示了对通道和驱动程序使用“逻辑分块”的第一次试用。

```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 5972.26
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 5972.26

```

● **注意:** 此控制的设置目的是用于比较。

16. 在服务器中，将“协议模式”设置为“逻辑未分块”。

17. 使用 Quick Client 连接至服务器。选择“工具”|“测试工具”，然后启用测试模式。

● **注意:** 数据读取开始。测试间隔到期后，所有标记都将被禁用，而驱动程序将停止统计信息收集。随后即可查看结果。

18. 断开 Quick Client 与服务器的连接，然后关闭服务器。

19. 重新启动服务器并搜索其“事件日志”以获取统计信息。下图显示了对设备使用“逻辑未分块”的第二次试用。

```

DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 3705.23

```

● **注意:** 下图显示了对通道和驱动程序使用“逻辑未分块”的第二次试用。

```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 3705.23
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 3705.23

```

20. 在服务器中，将“协议模式”设置为“符号”以查看 Allen-Bradley ControlLogix Ethernet 驱动程序 版本 4.6.0.xx 之前版本的性能状况。
21. 使用 Quick Client 连接至服务器。单击“工具”|“测试工具”，然后启用测试模式。
 - **注意：**数据读取开始。测试间隔到期后，所有标记都将被禁用，而驱动程序将停止统计信息收集。随后即可查看结果。
22. 断开 Quick Client 与服务器的连接，然后关闭服务器。
23. 重新启动服务器并搜索其事件日志以获取统计信息。下图显示了对设备使用“符号”的第三次试用。

```

DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 777.442

```

下图显示了对通道和驱动程序使用“符号”的第三次试用。


```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 777.442
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 777.442

```

● **注意:** 如此看来,“逻辑分块”更适合给定的应用程序。

优化通道通信

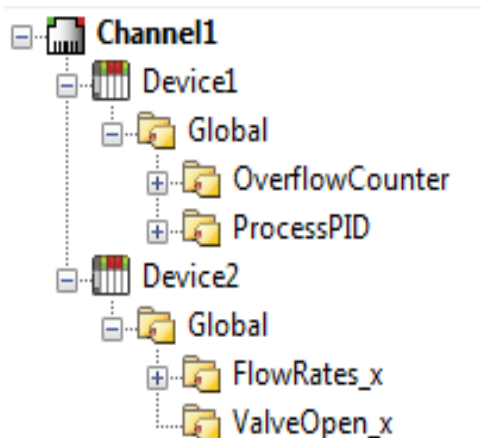
可以通过在一个设备中移动用于“逻辑分块”的标记,并在另一个设备中移动用于“逻辑未分块”的标记来优化通道通信。这种操作称为标记划分。

逻辑分块 (设备 1)

ProcessPID
OverflowCounter

逻辑未分块 (设备 2)

FlowRate
ValveOpen
InProcess
Tank Volume



1. 重复步骤 4 到步骤 15。在步骤 11 中，确保设备 1 为“逻辑分块”，设备 2 为“逻辑未分块”。
2. 启动服务器并搜索服务器事件日志以获取统计信息。下图显示了对设备使用标记划分的第四次试用。

```

DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 6933
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 76
Total tags read = 630979, Elapsed Time = 119782 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5268.43
DEVICE Channel1:Device2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 27732
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 104044, Elapsed Time = 119969 ms
DEVICE Channel1:Device2 PERFORMANCE: AvgTagReadPerSec = 867.373

```

● **注意：**下图显示了对通道和驱动程序使用标记划分的第四次试用。

```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 34665
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 735023, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 6126.77
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 34665
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 735023, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 6126.77

```

● **注意：**由于两个设备运行在独立的统计信息计数器上，因此单个的设备统计信息并不显眼。本测试的关键在于，通道和驱动程序统计信息比 (6126) 使用具有“逻辑分块”(5972) 或“逻辑未分块”(3705) 的一个通道/一个设备时更好。

优化应用程序

可通过将设备 1 移动到一个通道并将设备 2 移动到另一个通道来优化应用程序。

逻辑分块 (Channel1.Device 1)

ProcessPID

OverflowCounter

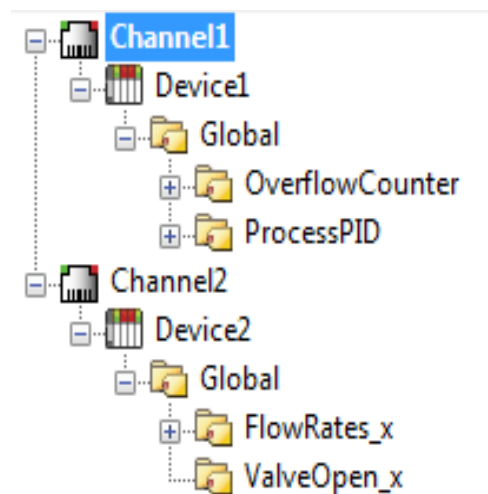
逻辑未分块 (Channel2.Device 2)

FlowRate

ValveOpen

InProcess

Tank Volume



1. 重复步骤 4 到步骤 15。在步骤 11 中，确保 Channel1.Device 1 为“逻辑分块”，Channel2.Device 2 为“逻辑未分块”。
2. 启动服务器并搜索服务器事件日志以获取统计信息。下图显示了针对 Channel 1.Device1 将 Logix 标记和多个通道配合使用的第五次试用。

```

CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 7271
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 661741, Elapsed Time = 119968 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 5517.4
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 7271
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 661741, Elapsed Time = 119968 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5517.4

```

● **注意：**下图显示了对 Channel2.Device2 使用 Logix 标记的第四次试用。

```

CHANNEL Channel2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 29120
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 109205, Elapsed Time = 119968 ms
CHANNEL Channel2 PERFORMANCE: AvgTagReadPerSec = 910.52
DEVICE Channel2:Device2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 29120
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 109205, Elapsed Time = 119968 ms
DEVICE Channel2:Device2 PERFORMANCE: AvgTagReadPerSec = 910.52

```

● **注意：**下图显示了对驱动程序使用标记划分的第四次试用。

```
DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 36391
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 84
Total tags read = 770946, Elapsed Time = 119968 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 6426.26
```

结果

服务器项目布局	驱动程序性能 (读取次数/秒)	相对于“符号”的改进
单通道 具有“逻辑分块”的单个设备	5972	768%
单通道 具有“逻辑未分块”的单个设备	3705	476%
单通道 具有“符号”的单个设备	777	不适用
单通道 带标记划分的多个设备	6126	788%
多个通道 带标记划分的多个设备	6426	827%

结论

项目开始时具有单个通道和单个设备，这是单个控制器的默认设置。所有标记均从此控制器中导入到此 **channel.device**。随后会对所有三种协议模式进行测试以确定哪一种能够实现最佳性能。在这种情况下，“逻辑分块协议”可以实现最佳性能。最佳协议取决于当前的应用程序。当性能至关重要时，可以进行“逻辑分块”和“逻辑未分块”试用，以确定哪一种是最适合此应用程序的协议模式。不需要符号协议，因为这种协议达不到其他任何一种协议模式的性能水平。显示于此处仅用作示例。

已根据 [优化通信](#) 中所述的提示采取措施来优化通信。最值得一提的是，已使用标记划分将“逻辑分块”类型标记置于分配了“逻辑分块”的设置中，将“逻辑未分块”类型标记置于分配了“逻辑未分块”的设备中。另外，两台设备均位于同一通道上。结果显示了相对于对单个设备使用“逻辑分块”所实现的改进。这是因为相对于其他协议模式，某些标记本身更适用于某一种协议模式。例如，由于将计数器作为块读取要比将其作为单个成员读取更快，所以从“逻辑分块”读取整个计数器要好于从“逻辑未分块”读取整个计数器。

也已采取措施通过将设备置于其自己的通道来优化应用程序。使用先前试用中所创建的设备，将“逻辑分块”设备置于一个通道中，将“逻辑未分块”设备置于另一个通道中。结果显示了相对于先前试用中单通道/多设备场景得到改进。这再次肯定了可以通过尽可能减少每个通道的设备数并尽可能增加总通道数来提高性能的想法。

使用了这三种优化方法后，项目的性能相对于 **Allen-Bradley ControlLogix Ethernet 驱动程序 4.6.0.xx** 之前的版本提高了 **827%**。标记划分和多通道使性能提高了 **107%**。项目越大，性能提高越明显。

数据类型说明

数据类型	说明
布尔型	单个位
字节	无符号 8 位值
字符	有符号 8 位值
字	无符号 16 位值
短整型	有符号 16 位值

数据类型	说明
双字型	无符号 32 位值
长整型	有符号 32 位值
BCD	两个字节封装的 BCD，四位十进制数字
LBCD	四个字节封装的 BCD，八位十进制数字
浮点型	32 位 IEEE 浮点
双精度	64 位 IEEE 浮点
日期	64 位日期/时间
字符串	空终止字符数组

有关 Logix 特定平台 数据类型的说明，请参阅 [Logix 高级寻址](#)。

有关对固件 V30 中的布尔型数组所做更改的具体信息，请参阅此 [知识库文章](#)。

默认数据类型条件

当发生以下任意情况时，会向客户端/服务器标记分配默认数据类型：

- 在客户端中创建动态标记，并将“原生”作为其分配的数据类型。
- 在服务器中创建静态标记，并将“默认”作为其分配的数据类型。
- 在生成离线自动标记的过程中，当未知数据类型的 UDT 成员和别名标记出现在 5K/L5X 文件中时。
- 在生成离线自动标记的过程中，当以下类型的别名出现在 L5K/L5X 中时：
 - 别名的别名。
 - 非字/双字内位 I/O 模块标记的别名。例如，如果标记 "AliasTag" 参考 I/O 模块标记 "Local:5:C.ProgToFaultEn" @ BOOL，则无法解析 "AliasTag" 的数据类型，所以会为其分配此默认类型。另一方面，如果 "AliasTag" 参考 I/O 模块标记 "Local:5:C.Ch0Config.RangeType.0" @ BOOL，则可以解析该数据类型，因为 .(dot) BIT 将其定义为字/双字内位。布尔型数据类型将自动分配给字/双字内位 I/O 模块标记的别名。

注意：

- 如果选择“默认”，当客户端动态访问标记时，驱动程序会从控制器检索 Logix 标记的数据类型，而不会将数据类型显式分配给该项。例如，一个标记存在于名为 "MyTag" 的控制器中，数据类型为实数。相应客户端项将指定为 "Channel1.Device1.MyTag"，且未分配数据类型。当选择“默认”作为服务器中的默认数据类型时，驱动程序会从控制器读取 "MyTag"，并在响应中确定其为实数，同时向客户端提供浮点型数据类型。
- 由于大多数 I/O 模块标记都不是字/双字内位标记，建议将默认类型设置为 .ACD 项目中常见的数据类型。例如，如果 75% 的别名 I/O 模块标记是整型标记，则将默认类型设置为整型。

地址说明

地址规范因所使用的型号而异。关于相关型号的地址信息，请参阅下表。

型号	Output	Input	Status	Binary	Timer	Counter	Control	Integer	Float	ASCII	String	BCD	Long	PID	Message	Block Transfer	Function
MicroLogix	X	X	X	X	X	X	X	X	X	X	X	X		X	X		
PLC5	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	
SLC 5/05	X	X	X	X	X	X	X	X	X	X	X						

另请参阅：
[Logix 寻址](#)

[MicroLogix 寻址](#)
[PLC-5 系列寻址](#)
[SLC 500 Modular I/O 寻址](#)

协议类	型号	帮助链接
Logix 以太网	ControlLogix 5500 以太网、CompactLogix 5300 以太网、FlexLogix 5400 以太网、SoftLogix 5800	Logix 寻址
DH+ 网关	DH+ 网关: PLC-5 DH+ 网关: SLC 5/04	PLC-5 系列寻址 SLC 500 Modular I/O 寻址
ControlNet 网关	ControlNet 网关: PLC-5C	PLC-5 系列寻址
1761-NET-ENI	ENI: ControlLogix 5500 ENI: CompactLogix 5300 ENI: FlexLogix 5400 ENI: MicroLogix ENI: SLC 500 Fixed I/O ENI: SLC 500 Modular I/O ENI: PLC-5	Logix 寻址 MicroLogix 寻址 SLC 500 Fixed I/O 寻址 SLC 500 Modular I/O 寻址 PLC-5 系列寻址
MicroLogix 1100 以太网	MicroLogix 1100	MicroLogix 寻址
MicroLogix 1400 以太网	MicroLogix 1400	MicroLogix 寻址

有关控制器的预定义数据类型的详细信息，请参阅设备的文档。

Logix 寻址

有关这些模型基于标记的寻址和与 *Allen-Bradley ControlLogix Ethernet 驱动程序* 的关系的详细信息，请参阅[Logix 基于标记的寻址](#)。

以太网的 ControlLogix 5500 寻址

ControlLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

ENI 的 ControlLogix 5500 寻址

ControlLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

串行网关的 ControlLogix 5500 寻址

ControlLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

以太网的 CompactLogix 5300 寻址

CompactLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

ENI 的 CompactLogix 5300 寻址

CompactLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

串行网关的 CompactLogix 5300 寻址

CompactLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

以太网的 FlexLogix 5400 寻址

FlexLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

ENI 的 FlexLogix 5400 寻址

FlexLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

串行网关的 FlexLogix 5400 寻址

FlexLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

SoftLogix 5800 寻址

SoftLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

串行网关的 SoftLogix 5800 寻址

SoftLogix 是 Logix 系列的成员，也是 Rockwell 自动化集成体系架构的一部分。这意味着它使用基于标记或符号的寻址结构。Logix 标记与常规 PLC 数据项的区别在于，标记名称本身是地址，而不是物理或逻辑地址。

MicroLogix 寻址

EtherNet/IP 网关的 MicroLogix 寻址

可用的实际地址数取决于 PLC 的模型。范围是开放的，以便为未来的模型提供最大的灵活性。如果驱动程序在运行时发现设备中不存在某一地址，它将发布一条错误消息，然后从其扫描列表中移除该标记。有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)

[输入文件](#)

[状况文件](#)

[二进制文件](#)

[计时器文件](#)

[计数器文件](#)

[控制文件](#)

[整型文件](#)

[浮点型文件](#)

[ASCII 文件](#)

[字符串文件](#)

[长整型文件](#)

[MicroLogix PID 文件](#)

[MicroLogix 消息文件](#)

有关功能文件的信息，请从下表选择一个链接。

[高速计数器文件 \(HSC\)](#)

[实时时钟文件 \(RTC\)](#)

[通道 0 通信状况文件 \(CS0\)](#)

[通道 1 通信状况文件 \(CS1\)](#)

[I/O 模块状况文件 \(IOS\)](#)

ENI 的 MicroLogix 寻址

可用的实际地址数取决于 PLC 的模型。范围是开放的，以便为未来的模型提供最大的灵活性。如果驱动程序在运行时发现设备中不存在某一地址，它将发布一条错误消息，然后从其扫描列表中移除该标记。有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)
[字符串文件](#)
[长整型文件](#)
[MicroLogix PID 文件](#)
[MicroLogix 消息文件](#)

有关功能文件的信息，请从下表选择一个链接。

[高速计数器文件 \(HSC\)](#)
[实时时钟文件 \(RTC\)](#)
[通道 0 通信状况文件 \(CS0\)](#)
[通道 1 通信状况文件 \(CS1\)](#)
[I/O 模块状况文件 \(IOS\)](#)

MicroLogix 1100 寻址

可用的实际地址数取决于 PLC 的模型。范围是开放的，以便为未来的模型提供最大的灵活性。如果驱动程序在运行时发现设备中不存在某一地址，它将发布一条错误消息，然后从其扫描列表中移除该标记。有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[字符串文件](#)
[长整型文件](#)
[MicroLogix PID 文件](#)
[MicroLogix 消息文件](#)

有关功能文件的信息，请从下表选择一个链接。

[高速计数器文件 \(HSC\)](#)
[实时时钟文件 \(RTC\)](#)
[通道 0 通信状况文件 \(CS0\)](#)
[通道 1 通信状况文件 \(CS1\)](#)
[I/O 模块状况文件 \(IOS\)](#)

MicroLogix 1400 寻址

可用的实际地址数取决于 PLC 的模型。范围是开放的，以便为未来的模型提供最大的灵活性。如果驱动程序在运行时发现设备中不存在某一地址，它将发布一条错误消息，然后从其扫描列表中移除该标记。有关文件特定寻址的详细信息，请从下表中选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)
[字符串文件](#)
[长整型文件](#)
[MicroLogix PID 文件](#)
[MicroLogix 消息文件](#)

有关功能文件的信息，请从下表中选择一个链接。

[高速计数器文件 \(HSC\)](#)
[实时时钟文件 \(RTC\)](#)
[通道 0 通信状况文件 \(CS0\)](#)
[通道 1 通信状况文件 \(CS1\)](#)
[I/O 模块状况文件 \(IOS\)](#)

SLC 500 固定 I/O 寻址

EtherNet/IP 网关的 SLC 500 固定 I/O 寻址

有关文件特定寻址的详细信息，请从下表中选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)

ENI 的 SLC 500 固定 I/O 寻址

有关文件特定寻址的详细信息，请从下表中选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)

SLC 500 模块化 I/O 寻址

DH+ 的 SLC 500 模块化 I/O 寻址

可用的实际地址数取决于 PLC 的模型。范围是开放的，以便为未来的模型提供最大的灵活性。如果驱动程序在运行时发现设备中不存在某一地址，它将发布一条错误消息，然后从其扫描列表中移除该标记。有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)
[字符串文件](#)

EtherNet/IP 网关的 SLC 500 模块化 I/O 寻址

可用的实际地址数取决于 PLC 的模型。范围是开放的，以便为未来的模型提供最大的灵活性。如果驱动程序在运行时发现设备中不存在某一地址，它将发布一条错误消息，然后从其扫描列表中移除该标记。有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)
[字符串文件](#)

ENI 的 SLC 500 模块化 I/O 寻址

可用的实际地址数取决于 PLC 的模型。范围是开放的，以便为未来的模型提供最大的灵活性。如果驱动程序在运行时发现设备中不存在某一地址，它将发布一条错误消息，然后从其扫描列表中移除该标记。有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)
[字符串文件](#)

PLC-5 系列寻址

用于 ControlNet 的 PLC-5 系列寻址

有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)
[字符串文件](#)
[BCD 文件](#)
[PID 文件](#)
[消息文件](#)
[块传输文件](#)

用于 DH+ 的 PLC-5 系列寻址

有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)
[字符串文件](#)
[BCD 文件](#)
[PID 文件](#)
[消息文件](#)
[块传输文件](#)

用于 EtherNet/IP 网关的 PLC-5 系列寻址

有关文件特定寻址的详细信息，请从下表选择一个链接。

[输出文件](#)
[输入文件](#)
[状况文件](#)
[二进制文件](#)
[计时器文件](#)
[计数器文件](#)
[控制文件](#)
[整型文件](#)
[浮点型文件](#)
[ASCII 文件](#)

- [字符串文件](#)
- [BCD 文件](#)
- [PID 文件](#)
- [消息文件](#)
- [块传输文件](#)

用于 ENI 的 PLC-5 系列寻址

有关文件特定寻址的详细信息，请从下表中选择一个链接。

- [输出文件](#)
- [输入文件](#)
- [状况文件](#)
- [二进制文件](#)
- [计时器文件](#)
- [计数器文件](#)
- [控制文件](#)
- [整型文件](#)
- [浮点型文件](#)
- [ASCII 文件](#)
- [字符串文件](#)
- [BCD 文件](#)
- [PID 文件](#)
- [消息文件](#)
- [块传输文件](#)

Logix 基于标记的寻址

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

● **注意：**除非另行说明，否则在本帮助文件中，假定 **Logix** 标记本质上为全局标记。

Allen-Bradley ControlLogix Ethernet 驱动程序 允许用户访问控制器的原子数据类型：**BOOL**、**SINT**、**INT**、**DINT**、**LINT**、**REAL**、**LREAL**、**USINT**、**UINT**、**UDINT** 和 **ULINT**。尽管某些预定义类型为结构，但归根结底，它们基于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，可以将 **TIMER** 的原子型成员分配给标记 (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	0、1
SINT	有符号 8 位值	VT_UI1	-128 到 127
INT	有符号 16 位值	VT_I2	-32,768 到 32,767
DINT	有符号 32 位值	VT_I4	-2,147,483,648 到 2,147,483,647
LINT	有符号 64 位值	VT_I8	-9,223,372,036,854,775,808 到 9,223,372,036,854,775,807
USINT	无符号 8 位值	VT_UI1	0 到 255
UINT	无符号 16 位值	VT_UI2	0 到 65535

原子型数据类型	说明		范围
UDINT	无符号 32 位值	VT_UI4	0 到 4294967295
ULINT	无符号 64 位值	VT_I8	0 到 18,446,744,073,709,551,615
REAL	32 位 IEEE 浮点	VT_R4	±1.1754943508222875E-38 到 ±3.4028234663852886E+38 (规范化) 0 ±1.4012984643248170E-45 到 ±1.1754942106924411E-38 (去规范化)
LREAL	有符号 64 位值	VT_R8	±2.2250738585072014E-308 到 ±1.7976931348623157E+308 (规范化) 0, ±4.9406564584124654E-324 到 ±2.2250738585072010E-308 (去规范化)

● 另请参阅: [Logix 高级寻址](#)

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。
- 最多可以包含 40 个字符。
- 不能有连续的下划线。
- 不区分大小写。

客户端/服务器标记名称规则

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。

● 注意: 为获得最佳性能, Logix 标记名称应尽可能保持最短。名称越短, 在单个事务中适合的请求就越多。

● “符号模式”用户应将客户端/服务器标记地址保持在 400 个字符以内。例如, tagarray [1,2,4].somestruct.substruct_array[3].basetag.[4] 的长度为 57 个字符。由于数据包只能包含 500 个数据字节, 需要添加到数据包中的任何开销字节会极大地减少字符本身的可用空间。通过将地址长度限制在 400 个字符以内, 可以保持标记请求的完整性和有效性。

● 另请参阅: [性能优化](#)

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。所用格式取决于标记的类型及其使用情况。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信息, 请参阅下表。

● 注意: 除数组和字符串以外的所有格式均为 RSLogix5000 所固有。因此, 参考原子型数据类型时, 可将 RSLogix 5000 标记名称复制并粘贴到服务器的标记地址字段, 并可供使用。

格式	语法	示例	注解
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] tag_1 [0, 3]	维度范围 = 1 至 3 元素范围 = 0 至 65535
无偏	<Logix 数组标记名称> {列数}	tag_1 {8} tag_1 {2}{4}	维度范围 = 1 至 2 元素范围 = 1 至 65535

格式	语法	示例	注解
移数组*	<Logix 数组标记名称> {行数}{列数}		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。 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> [偏移 1, 偏移 2]{列数} <logix 数组元素标记> [偏移 1, 偏移 2]{行数}{列数}	tag_1 [2, 3] {10} 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。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 <logix 标记名称> .[位]	tag_1.0 tag_1.[0]	位范围 = 0 至 31 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> /<最大字符串长度>	tag_1.Data/4 Stringtag_1.Data/82 SINTarraytag_1/16	长度范围 = 1 至 65535 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 = 4 且控制器标记为 3X3 元素数组，则所参考的元素采用 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。

要了解如何引用 1 维、2 维和 3 维数组的元素，请参阅[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，但程序标记的范围局限于定义该标记的程序内。程序标记遵循与全局标记相同的寻址规则和限制，但其使用下列符号作为前缀：

Program: <程序名称>。

例如，将在客户端/服务器标记地址中以 "Program:prog_1.tag_1" 的形式对程序 "prog_1" 中的 Logix 标记 "tag_1" 进行寻址。

结构标记寻址

Logix 结构标记 (全局或程序) 是带有一个或多个成员标记的标记。成员标记的属性可以是原子型或结构型。

<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：**上述示例仅列举出几种涉及结构的寻址可能性，所示示例仅用于介绍结构寻址。有关详细信息，请参阅 Allen-Bradley 或 Rockwell 文档。

内部标记

尽管内部标记在服务器配置中不可见，但可通过 OPC 客户端进行浏览并在 <通道名称> 下找到。<设备名称> 组。_CIPConnectionSizeRequested 标记反映了所请求的 CIP 连接大小。_CIPConnectionSizeActual 标记反映了正在使用的实际 CIP 连接大小。如果设备不支持所请求的值，则该值与 _CIPConnectionSizeRequested 标记不同。有关连接大小的详细信息，请参阅 [Logix 通信参数](#)。

● **注意：**下表所描述的标记仅对于 ControlLogix 5500 和 CompactLogix 5300 设备型号有效。

类型	标记名称	支持	访问
系统标记	_CIPConnectionSizeRequested	Logix 模型，串行网关模型除外。	读/写*
状况标记	_CIPConnectionSizeActual	Logix 模型，串行网关模型除外。	只读

*对于 ENI Logix 模型，此标记为只读形式。

更改 CIP 连接大小

_CIPConnectionSizeRequested 标记允许用户实时更改 CIP 连接大小属性。连接客户端后，连接大小属性 (位于“设备属性”中的“**Logix 通信参数**”下) 和系统标记均可配置。在执行下一个读/写请求之前，应先应用更改。

预定义术语标记

下表中显示的标记可用于从运行 13 或更高版本固件的 PLC 中获取常规处理器信息。

标记名称	说明
#MODE	PLC 当前密钥切换模式的说明。可能的字符串值包括程序、运行、远程程序、远程运行和远程调试。支持的数据类型包括字符串。
#PLCTYPE	与 PLC 的 EDS 文件中指定的 "ProdType" 属性对应的整数数值。支持的数据类型包括除字符串以外的所有类型。
#REVISION	固件修订显示为"<主要>.<次要>"。支持的数据类型包括字符串。
#PROCESSORNAME	与 PLC 的 EDS 文件中指定的 "ProdName" 属性对应的处理器名称。支持的数据类型包括字符串。
#STATUS	指示 PLC 的状态。可能的值包括确定 (1) 和故障 (0)。支持的数据类型包括除日期以外的所有类型。
#PRODUCTCODE	与 PLC 的 EDS 文件中指定的 "ProdCode" 属性对应的整数数值。支持的数据类型包括除字符串以外的所有类型。
#VENDORID	与 PLC 的 EDS 文件中指定的 "VendCode" 属性对应的整数数值。支持的数据类型包括除字符串以外的所有类型。

原子型数据类型寻址

如下所列是在给定可用地址格式的情况下，Logix 数据类型的建议使用情况和寻址可能性。同时提供了示例以强化理解。单击“**高级**”可获得给定原子型数据类型的高级寻址可能性。

● **注意：**空白单元不一定表示缺少支持。

原子型数据类型	标准	数组元素	带偏移/无偏移数组	位	字符串
BOOL					
客户端/服务器数据类型	布尔型	布尔型	布尔型数组		
高级		(BOOL 1 维数组)	(BOOL 1 维数组)		
客户端/服务器标记示例	BOOLTAG	BOOLARR[0]	BOOLARR[0]{32}		

原子型数据类型	标准	数组元素	带偏移/无偏移数组	位	字符串
SINT					
客户端/服务器数据类型 高级	字节、字符	字节、字符	字节数组、字符数组 (SINT 1/2/3 维数组)	布尔型 (SINT 中的位)	字符串 (SINT 1/2/3 维数组)
客户端/服务器标记示例	SINTTAG	SINTARR[0]	SINTARR[0]{4}	SINTTAG.0	SINTARR/4
INT					
客户端/服务器数据类型 高级	字、短整型	字、短整型	字数组、短整型数组 (INT 1/2/3 维数组)	布尔型 (INT 中的位)	请参阅 高级寻址 - INT
客户端/服务器标记示例	INTTAG	INTARR[0]	INTARR[0]{4}	INTTAG.0	
DINT					
客户端/服务器数据类型 高级	双字型、长整型	双字型、长整型	双字型数组、长整型数组	布尔型 (DINT 中的位)	请参阅 高级寻址 - DINT
客户端/服务器标记示例	DINTTAG	DINTARR[0]	DINTARR[0]{4}	DINTTAG.0	
LINT					
客户端/服务器数据类型 高级	双精度、日期	双精度、日期	双精度数组		
客户端/服务器标记示例	LINTTAG	LINTARR[0]	LINTARR[0]{4}		
REAL					
客户端/服务器数据类型 高级	浮点型	浮点型	浮点型数组		请参阅 高级寻址 - REAL
客户端/服务器标记示例	REALTAG	REALARR[0]	REALARR[0]{4}		
USINT					
客户端/服务器数据类型 高级	字节	字节	字节数组	布尔型 (USINT 中的位)	请参阅 高级寻址 - USINT
客户端/服务器数据类型	USINTTAG	USINTTARR[0]	USINTTARR[0]{4}	USINTTAG.0	
UINT					
客户端/服务器数据类型 高级	字、BCD	字、BCD	字数组、BCD 数组	布尔型 (UINT 中的位)	请参阅 高级寻址 - UINT
客户端/服务器标记示例	UINTTAG	UINTARR[0]	UINTARR[0]{4}	UINTTAG.0	
UDINT					
客户端/服务器数据类型 高级	双字型、LBCD	双字型、LBCD	双字型数组、LBCD 数组	布尔型	请参阅 高级寻址 - UDINT
客户端/服务器标记示例	UDINTTAG	UDINTARR[0]	UDINTARR[0]{4}	UDINTAG.0	
ULINT					
客户端/服务器数据类型 高级	双精度	双精度	双精度数组		请参阅 高级寻址 - ULINT
客户端/服务器标记示例	ULINTTAG	ULINTARR[0]	ULINTARR[0]{4}		

原子型数据类型	标准	数组元素	带偏移/无偏移数组	位	字符串
LREAL					
客户端/服务器数据类型 高级	双精度	双精度	双精度数组		请参阅 高级寻址 - LREAL
客户端/服务器标记示例	LREALTAG	LREALARR[0]	LREALARR[0]{4}		

寻址结构数据类型

只有原子型结构成员可在结构级别进行寻址。有关详细信息，请参阅以下示例。

Logix 标记

MyTimer @ TIMER

客户端/服务器标记

1. 无效

TimerTag 地址 = MyTimer

TimerTag 数据类型 = ??

2. 有效

TimerTag 地址 = MyTimer.ACC

TimerTag 数据类型 = 双字型

寻址字符串数据类型

“字符串”是一种预定义的 Logix 数据类型，其结构中包含两个成员：DATA 和 LEN。DATA 是“短整型”数组，可存储“字符串”的字符。LEN 是一种双整数，表示数据中要显示在客户端的字符数。

由于 LEN 和 DATA 是原子型成员，因此必须独立于客户端/服务器对其进行引用。语法如下所示。

说明	语法	示例
字符串值	DATA/<最大字符串长度>	MYSTRING.DATA/82
实际字符串长度	LEN	MYSTRING.LEN

读取

读取自 DATA 的“字符串”将在以下情况下终止：

- 第一个空终止符出现。
- LEN 中的值未先出现 (如果 a) 成立)。
- <最大字符串长度> 未先出现 (如果 a) 或 b) 成立)。

示例

MYSTRING.DATA 在 PLC 中包含 "Hello World"，但 LEN 手动设置为 5。读取 MYSTRING.DATA/82 时将显示 "Hello"。如果将 LEN 设置为 20，则 MYSTRING.DATA/82 将显示 "Hello World"。

写入

将“字符串”值写入数据时，驱动程序也会以所写入数据的长度写入 LEN。如果写入 LEN 的操作由于任何原因而失败，则写入数据的操作也将被视为失败 (尽管数据成功写入控制器)。

注意：此行为专用于字符串类型的 Logix 标记或自定义导数。以下注意事项适用于要在 UDT 中实施自己的“字符串”的用户。

- 如果 UDT 存在且其中含有作为“字符串”引用的 DATA 成员和作为“双整型”引用的 LEN 成员，则无论给定 UDT 的 LEN 意图如何，写入 LEN 的操作均将成功。如果不希望 LEN 与数据的长度相同，则设计 UDT 时必须谨慎，以避免这种可能性。

- 如果 UDT 存在且其中含有作为“字符串”引用的 DATA 成员，但不含 LEN 成员，则写入 LEN 的操作将静默失败，而不对 DATA 造成任何影响。

示例

MYSTRING.DATA/82 的值为 "Hello World"。MYSTRING.LEN 的值为 11。如果将值 "Alarm Triggered" 写入 MYSTRING.DATA/82，则会将 15 写入 MYSTRING.LEN。如果写入 MYSTRING.LEN 的操作失败，则 MYSTRING.LEN 将保留原有值 11，而 MYSTRING.DATA/82 将显示前 11 个字符 ("Alarm Trigg")。如果写入 MYSTRING.DATA/82 的操作失败，则两个标记均不受影响。

LEN 处的终止字符串数据

在逻辑寻址模式下，读取 STRING.DATA 时会自动读取符号模式下的 STRING.LEN。可通过禁用“LEN 处的终止字符串数据”选项跳过此操作。有关详细信息，请参阅 [Logix 选项](#)。

Logix 数组数据的排序

一维数组 - 数组 [1 维]

一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]

二维数组 - 数组 [1 维, 2 维]

二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]

三维数组 - 数组 [1 维, 2 维, 3 维]

三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]

Logix 高级寻址

“高级寻址”适用于以下原子型数据类型。有关特定数据类型的详细信息，请从下面的列表选择一个链接。

[BOOL](#)
[SINT](#)
[INT](#)
[DINT](#)
[LINT](#)
[REAL](#)
[USINT](#)
[UINT](#)
[UDINT](#)
[ULINT](#)
[LREAL](#)

高级寻址: BOOL

格式	支持的数据类型	注解
标准	布尔型 字节、字符 字、短整型、BCD 双字型、长整型、LBCD 浮点型*	无
	布尔型	控制器标记必须是 1 维数组。
无偏移数组	布尔型数组	1. 控制器标记必须是 1 维数组。 2. 元素的数目必须是 8 的因子。
无偏移数组	字节数组、字符数组 字数组、短整型数组、BCD 数组 双字型数组、长整型数组、LBCD 数组 浮点型数组*	不支持。
带偏移数组	布尔型数组	1. 控制器标记必须是 1 维数组。 2. 偏移必须位于 32 位边界处。 3. 元素的数目必须是 8 的因子。
位	布尔型	1. 控制器标记必须是 1 维数组。 2. 范围限制为 0 至 31。
字符串	字符串	不支持。

*浮点值等于浮点形式 (非 IEEE 浮点数) 的控制器标记的面值。

示例

[突出显示](#) 的示例表示常见用例。

BOOL 控制器标记 - booltag = true

服务器标记地址	格式	数据类型	注解
booltag	标准	布尔型	值 = 真
booltag	标准	字节	值 = 1
booltag	标准	Word	值 = 1
booltag	标准	双字型	值 = 1
booltag	标准	浮点型	值 = 1.0
booltag [3]	数组元素	布尔型	无效: 标记不是数组。
booltag [3]	数组元素	Word	无效: 标记不是数组。
booltag {1}	无偏移数组	Word	无效: 不受支持。
booltag {1}	无偏移数组	布尔型	无效: 不受支持。
booltag [3] {32}	带偏移数组	布尔型	无效: 标记不是数组。
booltag .3	位	布尔型	无效: 标记不是数组。
booltag / 1	字符串	字符串	无效: 不受支持。
booltag / 4	字符串	字符串	无效: 不受支持。

BOOL 数组控制器标记 - bitarraytag = [0,1,0,1]

服务器标记地址	格式	数据类型	注解
bitarraytag	标准	布尔型	无效: 标记不能是数组。
bitarraytag	标准	字节	无效: 标记不能是数组。
bitarraytag	标准	Word	无效: 标记不能是数组。
bitarraytag	标准	双字型	无效: 标记不能是数组。
bitarraytag	标准	浮点型	无效: 标记不能是数组。
bitarraytag [3]	数组元素	布尔型	值 = 真
bitarraytag [3]	数组元素	Word	无效: 数据类型不正确。
bitarraytag {3}	无偏移数组	Word	无效: 标记不能是数组。
bitarraytag {1}	无偏移数组	Word	无效: 标记不能是数组。
bitarraytag {1}	无偏移数组	布尔型	无效: 数组大小必须为 8 的因子。
bitarraytag {32}	无偏移数组	布尔型	值 = [0,1,0,1,...]
bitarraytag [3] {32}	带偏移数组	布尔型	偏移必须从 32 位边界处开始。
bitarraytag[0]{32}	带偏移数组	布尔型	值 = [0,1,0,1,...]
bitarraytag[32]{64}	带偏移数组	布尔型	值 = [...] 上方未提供的值
bitarraytag .3	位	布尔型	值 = 真
bitarraytag / 1	字符串	字符串	无效: 不受支持。
bitarraytag / 4	字符串	字符串	无效: 不受支持。

高级寻址: SINT

格式	支持的数据类型	注释
标准	布尔型*、字节、字符、字、短整型、BCD、双字型、长整型、LBCD、浮点型***	无
数组元素	字节、字符、字、短整型、BCD、双字型、长整型、LBCD、浮点型***	控制器标记必须是数组。
无偏移	布尔型数组	1. 使用此例, 以数组形式获得 SINT 中的位。 ● 注意: 这不是使用布尔型符号的 SINT 数组。

格式	支持的数据类型	注释
<u>数组</u>		2. 仅应用于 SINT 中的位。示例: tag_1.0{8}。 3. .bit 与数组大小的总和不能超过 8 位。示例: tag_1.1{8} 超出 SINT, tag_1.0{8} 未超出。 4. 数组大小必须为八(8)的倍数。
<u>无偏移数组</u>	字节数组、字符数组、字数组、短整型数组、BCD 数组**、双字型数组、长整型数组、LBCD 数组**、浮点型数组**	如果访问多个元素,则控制器标记必须是数组。
<u>带偏移数组</u>	字节数组、字符数组、字数组、短整型数组、BCD 数组**、双字型数组、长整型数组、LBCD 数组**、浮点型数组**	控制器标记必须是数组。
<u>位</u>	布尔型	1. 范围限制为 0 至 7。 2. 如果控制器标记是数组,则位类引用必须以数组元素类引用为前缀。示例: tag_1 [2,2,3].0。
<u>字符串</u>	字符串	1. 如果访问单个元素,则控制器标记无需是数组。 ● 注意: 字符串的值是 SINT 值的 ASCII 对等值。 示例: SINT = 65 (十进制) = "A"。 2. 如果访问多个元素,则控制器标记必须是数组。 字符串的值是字符串中所有“短整数”的 ASCII 对等值 (以空值终止)。字符串中的 1 个字符 = 1 SINT。

*非零值限制为“真”。

**数组的每个元素与 SINT 数组中的一个元素对应。不封装数组。

***浮点值等于浮点形式 (非 IEEE 浮点数) 的控制器标记的面值。

示例

突出显示的示例表示常见用例。

SINT 控制器标记 - sinttag = 122 (十进制)

服务器标记地址	格式	数据类型	注释
sinttag	标准	布尔型	值 = 真
sinttag	标准	字节	值 = 122
sinttag	标准	字	值 = 122
sinttag	标准	双字型	值 = 122
sinttag	标准	浮点型	值 = 122.0
sinttag [3]	数组元素	布尔型	无效: 标记不是数组。此外, 布尔型无效。
sinttag [3]	数组元素	字节	无效: 标记不是数组。
sinttag {3}	无偏移数组	字节	无效: 标记不是数组。
sinttag {1}	无偏移数组	字节	值 = [122]
sinttag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
sinttag [3] {1}	带偏移数组	字节	无效: 标记不是数组。
sinttag . 3	位	布尔型	值 = 真
sinttag . 0 {8}	无偏移数组	布尔型	值 = [0,1,0,1,1,1,1,0] 位值为 122
sinttag / 1	字符串	字符串	值 = "z"

服务器标记地址	格式	数据类型	注释
sinttag / 4	字符串	字符串	无效: 标记不是数组。

SINT 数组控制器标记 - sintarraytag [4,4] = [[83,73,78,84],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

服务器标记地址	格式	数据类型	注释
sintarraytag	标准	布尔型	无效: 标记不能是数组。
sintarraytag	标准	字节	无效: 标记不能是数组。
sintarraytag	标准	Word	无效: 标记不能是数组。
sintarraytag	标准	双字型	无效: 标记不能是数组。
sintarraytag	标准	浮点型	无效: 标记不能是数组。
sintarraytag [3]	数组元素	字节	无效: 服务器标记缺少 2 维地址。
sintarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
sintarraytag [1,3]	数组元素	字节	值 = 8
sintarraytag {10}	无偏移数组	字节	值 = [83,73,78,84,5,6,7,8,9,10]
sintarraytag {2} {5}	无偏移数组	Word	值 = [83,73,78,84,5] [6,7,8,9,10]
sintarraytag {1}	无偏移数组	字节	值 = 83
sintarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
sintarraytag [1,3] {4}	带偏移数组	字节	值 = [8,9,10,11]
sintarraytag .3	位	布尔型	无效: 标记必须引用原子型位置。
sintarraytag [1,3] .3	位	布尔型	值 = 1
sintarraytag [1,3] .0 {8}	无偏移数组	布尔型	值 = [0,0,0,1,0,0,0,0]
sintarraytag / 1	字符串	字符串	值 = "S"
sintarraytag / 4	字符串	字符串	值 = "SINT"

高级寻址: INT

格式	支持的数据类型	注解
<u>标准</u>	布尔型*、字节、字符**、字、短整型、BCD、双字型、长整型、LBCD、浮点型****	无
<u>数组元素</u>	字节、字符**、字、短整型、BCD、双字型、长整型、LBCD、浮点型****	控制器标记必须是数组。
<u>无偏移数组</u>	布尔型数组	<ol style="list-style-type: none"> 使用此例，以数组形式获得 INT 中的位。 注意: 这不是使用布尔型符号的 INT 数组。 仅应用于 INT 中的位。示例: tag_1.0{16}。 .bit 与数组大小的总和不能超过 16 位。示例: tag_1.1{16} 超出 INT, tag_1.0{16} 未超出。 数组大小必须为 八(8) 的倍数。
<u>无偏移数组</u>	字节数组、字符数组**、字数组、短整型数组、BCD 数组、双字型数组、长整型数组、LBCD 数组***浮点型数组***、****	如果访问多个元素，则控制器标记必须是数组。
<u>带</u>	字节数组、字符数组**、字数组、短整型数组、	控制器标记必须是数组。

格式	支持的数据类型	注解
偏移数组	BCD 数组、双字型数组、长整型数组、LBCD 数组***、浮点型数组***、****	
位	布尔型	<ol style="list-style-type: none"> 范围限制为 0 至 15。 如果控制器标记是数组，则位类引用必须以数组元素类引用为前缀。示例：tag_1 [2,2,3].0。
字符串	字符串	<ol style="list-style-type: none"> 如果访问单个元素，则控制器标记无需是数组。 ● 注意：字符串的值是 INT 值的 ASCII 对等值 (限制到 255)。示例：INT = 65 (十进制) = "A"。 如果访问多个元素，则控制器标记必须是数组。字符串的值是字符串中所有整数的 ASCII 对等值 (以空值终止，限制到 255)。 <p>字符串中的 1 个字符 = 1 INT (限制到 255)</p> <p>不压缩 INT 字符串。为提高效率，请改用 SINT 字符串或 STRING 结构。</p>

*非零值限制为“真”。

**超过 255 的值限制到 255。

***数组的每个元素对应于 INT 数组中的一个元素。不封装数组。

****浮点值等于浮点形式 (非 IEEE 浮点数) 的控制器标记的面值。

示例

突出显示的示例表示常见用例。

INT 控制器标记 - inttag = 65534 (十进制)

服务器标记地址	类	数据类型	注释
inttag	标准	布尔型	值 = 真
inttag	标准	字节	值 = 255
inttag	标准	字	值 = 65534
inttag	标准	双字型	值 = 65534
inttag	标准	浮点型	值 = 65534.0
inttag [3]	数组元素	布尔型	无效：标记不是数组。布尔型无效。
inttag [3]	数组元素	Word	无效：标记不是数组。
inttag {3}	无偏移数组	Word	无效：标记不是数组。
inttag {1}	无偏移数组	Word	值 = [65534]
inttag {1}	无偏移数组	布尔型	无效：数据类型不正确。
inttag [3] {1}	带偏移数组	字	无效：标记不是数组。
inttag . 3	位	布尔型	值 = 真
inttag . 0 {16}	无偏移数组	布尔型	值 = [0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] 位值为 65534
inttag / 1	字符串	字符串	值 = 不可打印字符 = 255 (十进制)。
inttag / 4	字符串	字符串	无效：标记不是数组。

INT 数组控制器标记 - intarraytag [4,4] = [[73,78,84,255],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

服务器标记地址	类	数据类型	注释
intarraytag	标准	布尔型	无效: 标记不能是数组。
intarraytag	标准	字节	无效: 标记不能是数组。
intarraytag	标准	Word	无效: 标记不能是数组。
intarraytag	标准	双字型	无效: 标记不能是数组。
intarraytag	标准	浮点型	无效: 标记不能是数组。
intarraytag [3]	数组元素	Word	无效: 服务器标记缺少 2 维地址。
intarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
intarraytag [1,3]	数组元素	Word	值 = 259
intarraytag {10}	无偏移数组	字节	值 = [73,78,84,255,255,255,255,255,9,10]
intarraytag {2} {5}	无偏移数组	Word	值 = [73,78,84,255,256] [257,258,259,9,10]
intarraytag {1}	无偏移数组	Word	值 = 73
intarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
intarraytag [1,3] {4}	带偏移数组	字	值 = [259,9,10,11]
intarraytag .3	位	布尔型	无效: 标记必须引用原子型位置。
intarraytag [1,3] .3	位	布尔型	值 = 0
intarraytag [1,3] .0 {16}	无偏移数组	布尔型	值 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] 位值为 259
intarraytag / 1	字符串	字符串	值 = "I"
intarraytag / 3	字符串	字符串	值 = "INT"

高级寻址: DINT

格式	支持的数据类型	注解
<u>标准</u>	布尔型*、字节、字符**、字、短整型、BCD***、双字型、长整型、LBCD、浮点型****	无
<u>数组元素</u>	字节、字符**、字、短整型、BCD***、双字型、长整型、LBCD、浮点型****	控制器标记必须是数组。
<u>无偏移数组</u>	布尔型数组	<ol style="list-style-type: none"> 1. 使用此例, 以数组形式获得 DINT 中的位。 ● 注意: 这不是使用布尔型符号的 DINT 数组。 2. 仅应用于 DINT 中的位。示例: tag_1.0{32}。 3. .bit 与数组大小的总和不能超过 32 位。示例: tag_1.1{32} 超出 DINT, tag_1.0{32} 未超出。 4. 数组大小必须是八 (8) 的倍数。
<u>无偏移数组</u>	字节数组、字符数组**、字数组、短整型数组、BCD 数组***、双字数组、长整型数组、LBCD 数组、浮点型数组****	如果访问多个元素, 则控制器标记必须是数组。
<u>带偏移数组</u>	字节数组、字符数组**、字数组、短整型数组、BCD 数组***、双字数组、长整型数组、LBCD 数组、浮点型数组****	控制器标记必须是数组。
<u>位</u>	布尔型	<ol style="list-style-type: none"> 1. 范围限制为 0 至 31。

格式	支持的数据类型	注解
		2. 如果控制器标记是数组，则位类引用必须以数组元素类引用为前缀。示例： <code>tag_1[2,2,3].0</code> 。
字符串	字符串	1. 如果访问单个元素，则控制器标记无需是数组。 ● 注意：字符串的值是 DINT 值的 ASCII 对等值 (限制到 255)。示例： <code>SINT = 65 (十进制) = "A"</code> 。 2. 如果访问多个元素，则控制器标记必须是数组。字符串的值是字符串中所有双整数的 ASCII 对等值 (以空值终止，限制到 255)。字符串中的 1 个字符 = 1 个双整数 (限制到 255) ● 注意：不封装 DINT 字符串。为提高效率，请改用 SINT 字符串或 STRING 结构。

*非零值限制为“真”。

**超过 255 的值限制到 255。

***超过 65535 的值限制到 65535。

****浮点值等于浮点形式 (非 IEEE 浮点数) 的控制器标记的面值。

示例

突出显示的示例表示常见用例。

DINT 控制器标记 - `dinttag = 70000` (十进制)

服务器标记地址	格式	数据类型	注解
<code>dinttag</code>	标准	布尔型	值 = 真
<code>dinttag</code>	标准	字节	值 = 255
<code>dinttag</code>	标准	Word	值 = 65535
<code>dinttag</code>	标准	双字型	值 = 70000
<code>dinttag</code>	标准	浮点型	值 = 70000.0
<code>dinttag [3]</code>	数组元素	布尔型	无效：标记不是数组。布尔型无效。
<code>dinttag [3]</code>	数组元素	双字型	无效：标记不是数组。
<code>dinttag {3}</code>	无偏移数组	双字型	无效：标记不是数组。
<code>dinttag {1}</code>	无偏移数组	双字型	值 = [70000]
<code>dinttag {1}</code>	无偏移数组	布尔型	无效：数据类型不正确
<code>dinttag [3]{1}</code>	带偏移数组	双字型	无效：标记不是数组。
<code>dinttag . 3</code>	位	布尔型	值 = 假
<code>dinttag . 0 {32}</code>	无偏移数组	布尔型	值 = [0,0,0,0,1,1,1,0,1,0,0,0,1,0,0,0,1,0,...0] 位值为 70000
<code>dinttag / 1</code>	字符串	字符串	值 = 不可打印字符 = 255 (十进制)
<code>dinttag / 4</code>	字符串	字符串	无效：标记不是数组。

DINT 数组控制器标记 - `dintarraytag [4,4] = [[68,73,78,84],[256,257,258,259],[9,10,11,12],[13,14,15,16]]`

服务器标记地址	格式	数据类型	注解
<code>dintarraytag</code>	标准	布尔型	无效：标记不能是数组。
<code>dintarraytag</code>	标准	字节	无效：标记不能是数组。
<code>dintarraytag</code>	标准	Word	无效：标记不能是数组。
<code>dintarraytag</code>	标准	双字型	无效：标记不能是数组。

服务器标记地址	格式	数据类型	注解
dintarraytag	标准	浮点型	无效: 标记不能是数组。
dintarraytag [3]	数组元素	双字型	无效: 服务器标记缺少 2 维地址。
dintarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
dintarraytag [1,3]	数组元素	双字型	值 = 259
dintarraytag {10}	无偏移数组	字节	值 = [68,73,78,84,255,255,255,255,9,10]
dintarraytag {2}{5}	无偏移数组	双字型	值 = [68,73,78,84,256] [257,258,259,9,10]
dintarraytag {1}	无偏移数组	双字型	值 = 68
dintarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
dintarraytag [1,3]{4}	带偏移数组	双字型	值 = [259,9,10,11]
dintarraytag .3	位	布尔型	无效: 标记必须引用原子型位置。
dintarraytag [1,3].3	位	布尔型	值 = 0
dintarraytag [1,3].0 {32}	无偏移数组	布尔型	值 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0] 位值为 259
dintarraytag / 1	字符串	字符串	值 = "D"
dintarraytag / 3	字符串	字符串	值 = "DINT"

高级寻址: LINT

格式	支持的数据类型	注解
<u>标准</u>	双精度*、日期**	无
<u>数组元素</u>	双精度*、日期**	控制器标记必须是数组。
<u>无偏移数组</u>	双精度、数组*	如果访问多个元素, 则控制器标记必须是数组。
<u>带偏移数组</u>	双精度、数组*	控制器标记必须是数组。
<u>位</u>	不适用	不支持。
<u>字符串</u>	不适用	不支持。

*双精度值等于浮点形式 (非 IEEE 浮点数) 的控制器标记的面值。

**日期值采用世界协调时间 (UTC), 而不是本地时间。

示例

突出显示 的示例表示常见用例。

LINT 控制器标记 - linttag = 2007-01-01T16:46:40.000 (日期) == 1.16767E+15 (十进制)

服务器标记地址	格式	数据类型	注解
linttag	标准	布尔型	无效: 布尔型不受支持。
linttag	标准	字节	无效: 字节类型不受支持。
linttag	标准	字	无效: 字类型不受支持。
linttag	标准	双精度	值 = 1.16767E+15
linttag	标准	日期	值 = 2007-01-01T16:46:40.000*
linttag [3]	数组元素	布尔型	无效: 标记不是数组。布尔型无效。
linttag [3]	数组元素	双精度	无效: 标记不是数组。
linttag {3}	无偏移数组	双精度	无效: 标记不是数组。
linttag {1}	无偏移数组	双精度	值 = [1.16767E+15]
linttag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
linttag [3]{1}	带偏移数组	双精度	无效: 标记不是数组。
linttag .3	位	布尔型	无效: 语法/数据类型不受支持。
linttag / 1	字符串	字符串	无效: 语法/数据类型不受支持。

*日期值采用世界协调时间 (UTC), 而不是本地时间。

LINT 数组控制器标记 -

lintarraytag [2,2] = [0, 1.16767E+15],[9.4666E+14, 9.46746E+14], 其中:

1.16767E+15 == 2007-01-01T16:46:40.000 (日期)

9.4666E+14 == 1999-12-31T17:06:40.000

9.46746E+14 == 2000-01-1T17:00:00.000

0 == 1970-01-01T00:00:00.000

服务器标记地址	格式	数据类型	注解
lintarraytag	标准	布尔型	无效: 布尔型不受支持。
lintarraytag	标准	字节	无效: 字节类型不受支持。
lintarraytag	标准	字	无效: 字类型不受支持。
lintarraytag	标准	双精度	无效: 标记不能是数组。
lintarraytag	标准	日期	无效: 标记不能是数组。
lintarraytag [1]	数组元素	双精度	无效: 服务器标记缺少 2 维地址。
lintarraytag [1,1]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
lintarraytag [1,1]	数组元素	双精度	值 = 9.46746E+14
lintarraytag [1,1]	数组元素	日期	值 = 2000-01-01T17:00:00.000*
lintarraytag {4}	无偏移数组	双精度	值 = [0, 1.16767E+15, 9.4666E+14, 9.46746E+14]
lintarraytag {2} {2}	无偏移数组	双精度	值 = [0, 1.16767E+15][9.4666E+14, 9.46746E+14]
lintarraytag {4}	无偏移数组	日期	无效: 日期数组不受支持。
lintarraytag {1}	无偏移数组	双精度	值 = 0
lintarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
lintarraytag [0,1] {2}	带偏移数组	双精度	值 = [1.16767E+15, 9.4666E+14]
lintarraytag . 3	位	布尔型	无效: 语法/数据类型不受支持。
lintarraytag / 1	字符串	字符串	无效: 语法/数据类型不受支持。

*日期值采用世界协调时间 (UTC), 而不是本地时间。

高级寻址: REAL

格式	支持的数据类型	注解
标准	布尔型*、字节、字符**、字、短整型、BCD***、双字型、长整型、LBCD、浮点型****	无
数组元素	字节、字符**、字、短整型、BCD***、双字型、长整型、LBCD、浮点型****	控制器标记必须是数组。
无偏移数组	布尔型数组	<ol style="list-style-type: none"> 使用此例, 以数组形式获得 REAL 中的位。 ● 注意: 这不是使用布尔型符号的 REAL 数组。 仅应用于 REAL 中的位。示例: tag_1.0{32}。 .bit 与数组大小的总和不能超过 32 位。示例: tag_1.1{32} 超出 REAL, tag_1.0{32} 未超出。 数组大小必须为 八(8) 的倍数。
无偏移数组	字节数组、字符数组**、字数组、短整型数组、BCD 数组***、双字型数组、长整型数组、LBCD 数组、浮点型数组****	如果访问多个元素, 则控制器标记必须是数组。
带偏移	字节数组、字符数组**、字数组、短整型数组、BCD 数组***、双字型数组、长	控制器标记必须是数组。

格式	支持的数据类型	注解
移数组	整型数组、LBCD 数组、浮点型数组****	
位	布尔型	<ol style="list-style-type: none"> 范围限制为 0 至 31。 如果控制器标记是数组，则位类引用必须以数组元素类引用为前缀。示例: tag_1 [2,2,3].0。 ● 注意: 浮点型将转换为双字型，以允许引用位。
字符串	字符串	<ol style="list-style-type: none"> 如果访问单个元素，则控制器标记无需是数组。 ● 注意: 字符串的值是 REAL 值的 ASCII 对等值 (限制到 255)。示例: SINT = 65 (十进制) = "A"。 如果访问多个元素，则控制器标记必须是数组。字符串的值是字符串中所有实数的 ASCII 对等值 (以空值终止，限制到 255)。字符串中的 1 个字符 = 1 个实数 (限制到 255)。 ● 注意: 不压缩 REAL 字符串。为提高效率，请改用 SINT 字符串或 STRING 结构。

*非零值限制为“真”。

**超过 255 的值限制到 255。

***超过 65535 的值限制到 65535。

****浮点值是有效的 IEEE 单精度浮点数。

示例

突出显示的示例表示常见用例。

“实型”控制器标记 - realtag = 512.5 (十进制)

服务器标记地址	格式	数据类型	注解
realtag	标准	布尔型	值 = 真
realtag	标准	字节	值 = 255
realtag	标准	字	值 = 512
realtag	标准	双字型	值 = 512
realtag	标准	浮点型	值 = 512.5
realtag [3]	数组元素	布尔型	无效: 标记不是数组。此外，布尔型无效。
realtag [3]	数组元素	双字型	无效: 标记不是数组。
realtag {3}	无偏移数组	双字型	无效: 标记不是数组。
realtag {1}	无偏移数组	浮点型	值 = [512.5]
realtag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
realtag [3]{1}	带偏移数组	浮点数	无效: 标记不是数组。
realtag . 3	位	布尔型	值 = 真
realtag . 0 {32}	无偏移数组	布尔型	值 = [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,...0] 位值为 512
realtag / 1	字符串	字符串	值 = 不可打印字符 = 255 (十进制)
realtag / 4	字符串	字符串	无效: 标记不是数组。

“实型”数组控制器标记 - realarraytag [4,4] = [[82.1,69.2,65.3,76.4],[256.5,257.6,258.7,259.8],[9.0,10.0,11.0,12.0],[13.0,14.0,15.0,16.0]]

服务器标记地址	格式	数据类型	注解
realarraytag	标准	布尔型	无效: 标记不能是数组。

服务器标记地址	格式	数据类型	注解
realarraytag	标准	字节	无效: 标记不能是数组。
realarraytag	标准	Word	无效: 标记不能是数组。
realarraytag	标准	双字型	无效: 标记不能是数组。
realarraytag	标准	浮点型	无效: 标记不能是数组。
realarraytag [3]	数组元素	浮点型	无效: 服务器标记缺少 2 维地址。
realarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
realarraytag [1,3]	数组元素	浮点型	值 = 259.8
realarraytag {10}	无偏移数组	字节	值 = [82,69,65,76,255,255,255,255,9,10]
realarraytag {2} {5}	无偏移数组	浮点型	值 = [82.1,69.2,65.3,76.4,256.5] [257.6,258.7,259.8,9,10]
realarraytag {1}	无偏移数组	浮点型	值 = 82.1
realarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
realarraytag [1,3] {4}	带偏移数组	浮点数	值 = [259,8,9,0,10.0,11,0]
realarraytag .3	位	布尔型	无效: 标记必须引用原子型位置。
realarraytag [1,3] .3	位	布尔型	值 = 0
realarraytag [1,3] .0 {32}	无偏移数组	布尔型	值 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] 位值为 259
realarraytag / 1	字符串	字符串	值 = "R"
realarraytag / 3	字符串	字符串	值 = "REAL"

高级寻址: USINT

格式	支持的数据类型	注解
<u>标准</u>	字节	无
<u>数组元素</u>	字节	控制器标记必须是数组。
<u>无偏移数组</u>	布尔型数组	<ol style="list-style-type: none"> 使用此例, 以数组形式获得 USINT 中的位。 ● 注意: 这不是使用布尔型符号的 USINT 数组。 仅应用于 USINT 中的位。示例: tag_1.0{8}。 .bit 与数组大小的总和不能超过 8 位。示例: tag_1.1{8} 超出 USINT, tag_1.0{8} 未超出。
<u>无偏移数组</u>	字节数组	如果访问多个元素, 则控制器标记必须是数组。
<u>带偏移数组</u>	字节数组	控制器标记必须是数组。
<u>位</u>	布尔型	<ol style="list-style-type: none"> 范围限制为 0 至 7。 如果控制器标记是数组, 则位类引用必须以数组元素类引用为前缀。示例: tag_1 [2,2,3].0。
<u>字符串</u>	不适用	不支持

示例

突出显示 的示例表示常见用例。

USINT 控制器标记 - usinttag = 122(十进制)

服务器标记地址	格式	数据类型	注解
usinttag	标准	布尔型	值 = 真
usinttag	标准	字节	值 = 122
usinttag	标准	字	无效: 字类型不受支持。
usinttag	标准	双字型	无效: 双字型不受支持。
usinttag	标准	浮点型	无效: 浮点型不受支持。
usinttag [3]	数组元素	布尔型	无效: 标记不是数组。此外, 布尔型无效。
usinttag [3]	数组元素	字节	无效: 标记不是数组。
usinttag {3}	无偏移数组	字节	无效: 标记不是数组。
usinttag {1}	无偏移数组	字节	值 = [122]
usinttag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
usinttag [3] {1}	带偏移数组	字节	无效: 标记不是数组。
usinttag . 3	位	布尔型	值 = 真
usinttag . 0 {8}	无偏移数组	布尔型	值 = [0,1,0,1,1,1,0] 位值为 122
usinttag / 1	字符串	字符串	值 = "z"
usinttag / 4	字符串	字符串	无效: 标记不是数组。

USINT 数组控制器标记 - usintarraytag [4,4] = [[83,73,78,84],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

服务器标记地址	格式	数据类型	注解
usintarraytag	标准	布尔型	无效: 标记不能是数组。
usintarraytag	标准	字节	无效: 标记不能是数组。
usintarraytag	标准	Word	无效: 标记不能是数组。
usintarraytag	标准	双字型	无效: 标记不能是数组。
usintarraytag	标准	浮点型	无效: 标记不能是数组。
usintarraytag [3]	数组元素	字节	无效: 服务器标记缺少 2 维地址。
usintarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
usintarraytag [1,3]	数组元素	字节	值 = 8
usintarraytag {10}	无偏移数组	字节	值 = [83,73,78,84,5,6,7,8,9,10]
usintarraytag {2} {5}	无偏移数组	Word	值 = [83,73,78,84,5] [6,7,8,9,10]
usintarraytag {1}	无偏移数组	字节	值 = 83
usintarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
usintarraytag [1,3] {4}	带偏移数组	字节	值 = [8,9,10,11]
usintarraytag . 3	位	布尔型	无效: 标记必须引用原子型位置。
usintarraytag [1,3] . 3	位	布尔型	值 = 1
usintarraytag [1,3] . 0 {8}	无偏移数组	布尔型	值 = [0,0,0,1,0,0,0,0]
usintarraytag / 1	字符串	字符串	无效: 语法/数据类型不受支持。
usintarraytag / 4	字符串	字符串	无效: 语法/数据类型不受支持。

高级寻址: UINT

格式	支持的数据类型	注解
标准	字、BCD	无
数组元素	字、BCD	控制器标记必须是数组。
无偏移	布尔型数组	1. 使用此例, 以数组形式获得 UINT 中的位。

格式	支持的数据类型	注解
<u>数组</u>		<p>● 注意: 这不是使用布尔型符号的 UINT 数组。</p> <p>2. 仅应用于 UINT 中的位。示例: tag_1.0{16}。</p> <p>3. .bit 与数组大小的总和不能超过 8 位。示例: tag_1.1{32} 超出 UINT, tag_1.0{32} 未超出。</p> <p>4. 数组大小必须为八(8)的倍数。</p>
<u>无偏移数组</u>	字数组、BCD 数组	如果访问多个元素,则控制器标记必须是数组。
<u>带偏移数组</u>	字数组、BCD 数组	控制器标记必须是数组。
<u>位</u>	布尔型	<p>1. 范围限制为 0 至 15。</p> <p>2. 如果控制器标记是数组,则位类引用必须以数组元素类引用为前缀。示例: tag_1 [2,2,3].0。</p>
<u>字符串</u>	不适用	不支持

示例

突出显示 的示例表示常见用例。

UINT 控制器标记 - uinttag = 65534(十进制)

服务器标记地址	类	数据类型	注解
uinttag	标准	布尔型	无效: 布尔型不受支持。
uinttag	标准	字节	无效: 字节类型不受支持。
uinttag	标准	字	值 = 65534
uinttag	标准	双字型	无效: 双字型不受支持。
uinttag	标准	浮点型	无效: 浮点型不受支持。
uinttag [3]	数组元素	布尔型	无效: 标记不是数组。布尔型无效。
uinttag [3]	数组元素	字节	无效: 标记不是数组。
uinttag {3}	无偏移数组	字节	无效: 标记不是数组。
uinttag {1}	无偏移数组	字节	值 = [65534]
uinttag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
uinttag [3] {1}	带偏移数组	字	无效: 标记不是数组。
uinttag . 3	位	布尔型	值 = 真
uinttag . 0 {16}	无偏移数组	布尔型	值 = [0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] 位值为 65534
uinttag / 1	字符串	字符串	无效: 语法/数据类型不受支持。
uinttag / 4	字符串	字符串	无效: 语法/数据类型不受支持。

UINT 数组控制器标记 - uintarraytag [4,4] = [[73,78,84,255],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

服务器标记地址	格式	数据类型	注解
uintarraytag	标准	布尔型	无效: 标记不能是数组。
uintarraytag	标准	字节	无效: 标记不能是数组。
uintarraytag	标准	Word	无效: 标记不能是数组。
uintarraytag	标准	双字型	无效: 标记不能是数组。
uintarraytag	标准	浮点型	无效: 标记不能是数组。
uintarraytag [3]	数组元素	Word	无效: 服务器标记缺少 2 维地址。
uintarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。

服务器标记地址	格式	数据类型	注解
uintarraytag [1,3]	数组元素	Word	值 = 259
uintarraytag {10}	无偏移数组	BCD	值 = [49,54,54,165,100,101,102,103,9,10]
uintarraytag {2} {5}	无偏移数组	Word	值 = [73,78,84,255,256] [257,258,259,9,10]
uintarraytag {1}	无偏移数组	Word	值 = 73
uintarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
uintarraytag [1,3] {4}	带偏移数组	字节	值 = [259,9,10,11]
uintarraytag . 3	位	布尔型	无效: 标记必须引用原子型位置。
uintarraytag [1,3] . 3	位	布尔型	值 = 0
uintarraytag [1,3] . 0 {16}	无偏移数组	布尔型	值 = [0,0,0,1,0,0,0,0] 值 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] 位值为 259
uintarraytag / 1	字符串	字符串	无效: 语法/数据类型不受支持。
uintarraytag / 3	字符串	字符串	无效: 语法/数据类型不受支持。

高级寻址: UDINT

格式	支持的数据类型	注解
<u>标准</u>	双字型、LBCD	无
<u>数组元素</u>	双字型、LBCD	控制器标记必须是数组。
<u>无偏移数组</u>	布尔型数组	<ol style="list-style-type: none"> 使用此例, 以数组形式获得 UDINT 中的位。 ● 注意: 这不是使用布尔型符号的 UDINT 数组。 仅应用于 UDINT 中的位。示例: tag_1.0{32}。 .bit 与数组大小的总和不能超过 32 位。示例: tag_1.1{32} 超出 UINT, tag_1.0{32} 未超出。 数组大小必须为八(8)的倍数。
<u>无偏移数组</u>	双字型数组、LBCD 数组	如果访问多个元素, 则控制器标记必须是数组。
<u>带偏移数组</u>	双字型数组、LBCD 数组	控制器标记必须是数组。
<u>位</u>	布尔型	<ol style="list-style-type: none"> 范围限制为 0 至 31。 如果控制器标记是数组, 则位类引用必须以数组元素类引用为前缀。示例: tag_1 [2,2,3].0
<u>字符串</u>	不适用	不支持

示例

突出显示 的示例表示常见用例。

UDINT 控制器标记 - uinttag = 70000(十进制)

服务器标记地址	格式	数据类型	注解
uinttag	标准	布尔型	无效: 布尔型不受支持。
uinttag	标准	字节	无效: 字节类型不受支持。
uinttag	标准	字	无效: 字类型不受支持。

服务器标记地址	格式	数据类型	注解
udinttag	标准	双字型	值 = 70000
udinttag	标准	LBCD	值 = 11170
udinttag [3]	数组元素	布尔型	无效: 标记不是数组。布尔型无效。
udinttag [3]	数组元素	双字型	无效: 标记不是数组。
udinttag {3}	无偏移数组	双字型	无效: 标记不是数组。
udinttag {1}	无偏移数组	双字型	值 = [70000]
udinttag {1}	无偏移数组	布尔型	无效: 布尔型数组不受支持。
udinttag [3] {1}	带偏移数组	双字型	无效: 标记不是数组。
udinttag . 3	位	布尔型	值 = 假
udinttag . 0 {32}	无偏移数组	布尔型	值 = [0,0,0,0,1,1,1,0,1,0,0,0,1,0,0,0,1,0,...0] 位值为 70000
udinttag / 1	字符串	字符串	无效: 语法/数据类型不受支持。
udinttag / 4	字符串	字符串	无效: 语法/数据类型不受支持。

UDINT 数组控制器标记 - `udintarraytag [4,4] = [[68,73,78,84],[256,257,258,259],[9,10,11,12],[13,14,15,16]]`

服务器标记地址	格式	数据类型	注解
udintarraytag	标准	布尔型	无效: 布尔型不受支持。
udintarraytag	标准	字节	无效: 字节类型不受支持。
udintarraytag	标准	字	无效: 字类型不受支持。
udintarraytag	标准	双字型	无效: 标记不能是数组。
udintarraytag	标准	浮点型	无效: 浮点型不受支持。
udintarraytag [3]	数组元素	双字型	无效: 服务器标记缺少 2 维地址。
udintarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
udintarraytag [1,3]	数组元素	双字型	值 = 259
udintarraytag {10}	无偏移数组	LCBD	值 = [44,49,54,54,100,101,102,103,9,10]
udintarraytag {2}{5}	无偏移数组	双字型	值 = [68,73,78,84,256] [257,258,259,9,10]
udintarraytag {1}	无偏移数组	双字型	值 = 68
udintarraytag {1}	无偏移数组	布尔型	无效: 数据类型不正确。
udintarraytag [1,3]{4}	带偏移数组	双字型	值 = [259,9,10,11]
udintarraytag . 3	位	布尔型	无效: 标记必须引用原子型位置。
udintarraytag [1,3] . 3	位	布尔型	值 = 假
udintarraytag [1,3] . 0 {32}	无偏移数组	布尔型	值 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] 位值为 259
udintarraytag / 1	字符串	字符串	无效: 语法/数据类型不受支持。
udintarraytag / 3	字符串	字符串	无效: 语法/数据类型不受支持。

高级寻址: ULINT

格式	支持的数据类型	注解
<u>标准</u>	双精度*	无
<u>数组元素</u>	双精度*	控制器标记必须是数组。
<u>无偏移数组</u>	双精度、数组*	如果访问多个元素, 则控制器标记必须是数组。
<u>带偏移数组</u>	双精度、数组*	控制器标记必须是数组。
<u>位</u>	不适用	不支持
<u>字符串</u>	不适用	不支持

*双精度值等于浮点形式 (非 IEEE 浮点数) 的控制器标记的面值。

示例

突出显示 的示例表示常见用例。

ULINT 控制器标记 - ulinttag = 1.8446744073709560e+19 (十进制)

服务器标记地址	格式	数据类型	注解
ulinttag	标准	布尔型	无效: 布尔型不受支持。
ulinttag	标准	字节	无效: 字节类型不受支持。
ulinttag	标准	字	无效: 字类型不受支持。
ulinttag	标准	双精度	值 = 1.8446744073709560e+19
ulinttag [3]	数组元素	布尔型	无效: 标记不是数组。布尔型无效。
ulinttag [3]	数组元素	双精度	无效: 标记不是数组。
ulinttag {3}	无偏移数组	双精度	无效: 标记不是数组。
ulinttag {1}	无偏移数组	双精度	值 = [1.8446744073709560e+19]
ulinttag {1}	无偏移数组	布尔型	无效: 布尔型数组不受支持。
ulinttag [3] {1}	带偏移数组	双精度	无效: 标记不是数组。
ulinttag . 3	位	布尔型	无效: 语法/数据类型不受支持。
ulinttag / 1	字符串	字符串	无效: 语法/数据类型不受支持。

ULINT 数组控制器标记 -

ulintarraytag [2,2] = [0, 1.16767E+15],[9.4666E+14, 1.8446744073709560e+19]

服务器标记地址	格式	数据类型	注解
ulintarraytag	标准	布尔型	无效: 布尔型不受支持。
ulintarraytag	标准	字节	无效: 字节类型不受支持。
ulintarraytag	标准	字	无效: 字类型不受支持。
ulintarraytag	标准	双精度	无效: 标记不能是数组。
ulintarraytag	标准	日期	无效: 日期类型不受支持。
ulintarraytag [1]	数组元素	双精度	无效: 服务器标记缺少 2 维地址。
ulintarraytag [1,1]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
ulintarraytag [1,1]	数组元素	双精度	值 = 1.8446744073709560e+19
ulintarraytag {4}	无偏移数组	双精度	值 = [0, 1.16767E+15, 9.4666E+14, 1.8446744073709560e+19]
ulintarraytag {2} {2}	无偏移数组	双精度	值 = [0, 1.16767E+15][9.4666E+14, 1.8446744073709560e+19]
ulintarraytag {4}	无偏移数组	日期	无效: 日期数组不受支持。
ulintarraytag {1}	无偏移数组	双精度	值 = 0
ulintarraytag {1}	无偏移数组	布尔型	无效: 布尔型数组不受支持。
ulintarraytag [0,1] {2}	带偏移数组	双精度	值 = [1.16767E+15, 9.4666E+14]
ulintarraytag . 3	位	布尔型	无效: 语法/数据类型不受支持。
ulintarraytag / 1	字符串	字符串	无效: 语法/数据类型不受支持。

高级寻址: LREAL

格式	支持的数据类型	注解
标准	双精度	无
数组元素	双精度	控制器标记必须是数组。
无偏移数组	双精度数组	如果访问多个元素，则控制器标记必须是数组。
带偏移数组	双精度数组	控制器标记必须是数组。
位	不适用	不支持
字符串	不适用	不支持

示例

[突出显示](#) 的示例表示常见用例。

LREAL 控制器标记 - lrealtag = 1.7976931348623157E+308(十进制)

服务器标记地址	格式	数据类型	注释
lrealtag	标准	布尔型	无效: 布尔型不受支持。
lrealtag	标准	字节	无效: 字节类型不受支持。
lrealtag	标准	字	无效: 字类型不受支持。
lrealtag	标准	双精度	值 = 1.7976931348623157E+308
lrealtag [3]	数组元素	布尔型	无效: 标记不是数组。此外, 布尔型无效。
lrealtag {1}	无偏移数组	双精度	值 = [1.7976931348623157E+308]
lrealtag {1}	无偏移数组	布尔型	无效: 布尔型数组不受支持。
lrealtag [3] {1}	带偏移数组	双精度	无效: 标记不是数组。
lrealtag . 3	位	布尔型	无效: 语法/数据类型不受支持。
lrealtag . 0 {32}	无偏移数组	布尔型	无效: 语法/数据类型不受支持。
lrealtag / 1	字符串	字符串	无效: 语法/数据类型不受支持。
lrealtag / 4	字符串	字符串	无效: 语法/数据类型不受支持。

LREAL 数组控制器标记 - lrealarraytag [4,4] = [[82.1,69.2,65.3,76.4],[256.5,257.6,258.7,259.8],[9.0,10.0,11.0,12.0],[13.0,14.0,15.0,16.0]]

服务器标记地址	格式	数据类型	注释
lrealarraytag	标准	布尔型	无效: 标记不能是数组。
lrealarraytag	标准	字节	无效: 标记不能是数组。
lrealarraytag	标准	Word	无效: 标记不能是数组。
lrealarraytag	标准	双精度	无效: 标记不能是数组。
lrealarraytag [3]	数组元素	双精度	无效: 服务器标记缺少 2 维地址。
lrealarraytag [1,3]	数组元素	布尔型	无效: 数组元素不允许使用布尔型。
lrealarraytag [1,1]	数组元素	双精度	值 = 257.6
lrealarraytag {2} {5}	无偏移数组	双精度	值 = [82.1,69.2,65.3,76.4,256.5] [257.6,258.7,259.8,9,10]
lrealarraytag {1}	无偏移数组	双精度	值 = 82.1
lrealarraytag {1}	无偏移数组	布尔型	无效: 布尔型数组不受支持。
lrealarraytag [1,3] {4}	带偏移数组	双精度	值 = [259,8,9,0,10.0,11.0]
lrealarraytag . 3	位	布尔型	无效: 语法/数据类型不受支持。
lrealarraytag / 1	字符串	字符串	无效: 语法/数据类型不受支持。

文件列表

有关各种设备型号支持的特定文件的信息, 请从下面的列表选择一个链接。

- [输出文件](#)
- [输入文件](#)
- [状况文件](#)
- [二进制文件](#)
- [计时器文件](#)
- [计数器文件](#)
- [控制文件](#)
- [整数文件](#)
- [浮点数文件](#)
- [ASCII 文件](#)
- [字符串文件](#)
- [BCD 文件](#)
- [长整数文件](#)
- [MicroLogix PID 文件](#)
- [PID 文件](#)
- [MicroLogix 消息文件](#)
- [消息文件](#)
- [块传输文件](#)

功能文件列表

- [高速计数器文件 \(HSC\)](#)
- [实时时钟文件 \(RTC\)](#)
- [通道 0 通信状况文件 \(CS0\)](#)
- [通道 1 通信状况文件 \(CS1\)](#)
- [I/O 模块状况文件 \(IOS\)](#)

有关设备型号及其所支持文件的详细信息，请参阅[地址说明](#)。

输出文件

用于访问输出文件中数据的语法因 PLC 模型而异。输出文件不支持数组。默认数据类型以**粗体**显示。

PLC-5 语法

语法	数据类型	访问
O:<字>	短整型、字、BCD	读/写
O:<字>/<位>	布尔型	读/写
O/位	布尔型	读/写

注意：PLC-5 模型的字和位地址信息采用八进制形式。这遵循了编程软件的惯例。

MicroLogix 语法

语法	数据类型	访问
O:<字>	短整型、字、BCD	读/写
O:<字>/<位>	布尔型	读/写
O/位	布尔型	读/写

MicroLogix 模型有两种类型的 I/O：嵌入式 I/O 和扩展 I/O (不适用于 MicroLogix 1000)。嵌入式 I/O 置于 CPU 基本单元内，而扩展 I/O 为插入 CPU 基本单元。下表列出了每种 MicroLogix 模型的 I/O 功能。

MicroLogix 模型	嵌入式 I/O	扩展 I/O
1000	插槽 0	不适用

MicroLogix 模型	嵌入式 I/O	扩展 I/O
1100	插槽 0	插槽 1-4
1200	插槽 0	插槽 1-6
1400	插槽 0	插槽 1-7
1500	插槽 0	插槽 1-16

MicroLogix I/O 的地址语法引用从零开始的字偏移，而不是插槽。用户必须确定到特定插槽的字偏移。这需要掌握模块及各自字大小的相关信息。下表指定了一些可用模块的大小；但是，建议用户同时参考 MicroLogix 文档和控制器项目来确定模块的真实字大小。

MicroLogix 嵌入式 I/O 字大小

MicroLogix 模型	输入字数量	输出字数量
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

MicroLogix 扩展 I/O 字大小

模块	输入字数量	输出字数量
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12

模块	输入字数量	输出字数量
1769-SM2	7	7
1769-ASCI	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OF8	0	1

计算

插槽 x 的输出字偏移 = 通过插槽 (x-1) 的插槽 0 中的输出字数量。

注意：

1. 在向扩展 I/O 进行偏移时，需要考虑嵌入式 I/O。
2. 输入字的数量不会计入到输出字偏移量的计算中。

I/O 示例

让

插槽 0 = MicroLogix 1500 LRP 系列 C = 4 输出字

插槽 1 = 1769-OF2 = 2 输出字

插槽 2 = 1769-OW8 = 1 输出字

插槽 3 = 1769-IA16 = 0 输出字

插槽 4 = 1769-OF8V = 9 输出字

插槽 4 的位 5 = 4 + 2 + 1 = 7 字 = O:7/5

SLC 500 语法

默认数据类型以**粗体**显示。

语法	数据类型	访问
O:<插槽>	短整型、 字 、BCD	只读
O:<插槽>.<字>	短整型、 字 、BCD	只读
O:<插槽>/<位>	布尔型	只读
O:<插槽>.<字>/<位>	布尔型	只读

范围

PLC 型号	最小插槽	最大插槽	最大字
MicroLogix	不适用	不适用	2047
SLC 500 固定 I/O	不适用	不适用	1
SLC 500 模块化 I/O	1	30	*
PLC-5 系列	不适用	不适用	277 (八进制)

*可在 [SLC 500 模块化 I/O 选择指南](#) 中找到每个 I/O 模块可用的输入或输出字数量。

示例

MicroLogix	说明
O:0	字 0
O/2	位 2
O:0/5	位 5

SLC 500 固定 I/O	说明
O:0	字 0
O:1	字 1
O/16	位 16
O:1/0	位 0 字 1 (与 O/16 相同)

PLC5*	说明
O:0	字 0
O:37	字 31 (八进制 37 = 十进制 31)
O/42	位 34 (八进制 42 = 十进制 34)
O:2/2	位 2 字 2 (与 O/42 相同)

*地址采用八进制形式。

SLC 500 模块化 I/O	说明
O:1	字 0 槽 1
O:1.0	字 0 槽 1 (与 O:1 相同)
O:12	字 0 槽 12
O:12.2	字 2 槽 12
O:4.0/0	位 0 字 0 槽 4
O:4/0	位 0 槽 4 (与 O:4.0/0 相同)
O:4.2/0	位 0 字 2 槽 4
O:4/32	位 32 槽 4 (与 O:4.2/0 相同)

输入文件

用于访问输入文件中数据的语法因 PLC 型号而异。输入文件不支持数组。默认数据类型以**粗体**显示。

PLC-5 语法

语法	数据类型	访问
I:<字>	短整型、 字 、BCD	读/写
I:<字>/<位>	布尔型	读/写
I/位	布尔型	读/写

● **注意：**PLC-5 模型的字和位地址信息采用八进制形式。这遵循了编程软件的惯例。

MicroLogix 语法

语法	数据类型	访问
I:<字>	短整型、 字 、BCD	读/写
I:<字>/<位>	布尔型	读/写
I/位	布尔型	读/写

MicroLogix 模型有两种类型的 I/O: 嵌入式 I/O 和扩展 I/O (不适用于 MicroLogix 1000)。嵌入式 I/O 置于 CPU 基本单元内, 而扩展 I/O 为插入 CPU 基本单元。下表列出了每种 MicroLogix 模型的 I/O 功能。

MicroLogix 模型	嵌入式 I/O	扩展 I/O
1000	插槽 0	不适用
1100	插槽 0	插槽 1-4
1200	插槽 0	插槽 1-6
1400	插槽 0	插槽 1-7
1500	插槽 0	插槽 1-16

MicroLogix I/O 的地址语法引用从零开始的字偏移, 而不是插槽。用户必须确定到特定插槽的字偏移。这需要掌握模块及各自字大小的相关信息。下表指定了一些可用模块的大小; 但是, 建议参考 MicroLogix 文档和控制器项目来确定模块的真实字大小。

MicroLogix 嵌入式 I/O 字大小

MicroLogix 模型	输入字数量	输出字数量
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

MicroLogix 扩展 I/O 字大小

模块	输入字数量	输出字数量
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1

模块	输入字数量	输出字数量
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

计算

插槽 x 的输入字偏移 = 通过插槽 (x-1) 的插槽 0 中的输入字数量。

注意：

1. 在向扩展 I/O 进行偏移时，需要考虑嵌入式 I/O。
2. 输出字的数量不会计入到输入字偏移量的计算中。

I/O 示例

让

插槽 0 = Micrologix 1500 LRP 系列 C = 4 个输入字

插槽 1 = 1769-OF2 = 2 个输入字

插槽 2 = 1769-OW8 = 0 个输入字

插槽 3 = 1769-IA16 = 1 个输入字

插槽 4 = 1769-OF8V = 11 个输入字

插槽 3 的位 5 = 4 + 2 = 6 个字 = I:6/5

SLC 500 语法

语法	数据类型	访问
I:<插槽>	短整型、字、BCD	只读
I:<插槽>.<字>	短整型、字、BCD	只读
I:<插槽>.<位>	布尔型	只读
I:<插槽>.<字>.<位>	布尔型	只读

范围

PLC 型号	最小插槽	最大插槽	最大字
MicroLogix	不适用	不适用	2047

PLC 型号	最小插槽	最大插槽	最大字
SLC 500 固定 I/O	不适用	不适用	1
SLC 500 模块化 I/O	1	30	*
PLC-5 系列	不适用	不适用	277 (八进制)

*可在 [SLC 500 模块化 I/O 选择指南](#) 中找到每个 I/O 模块可用的输入或输出字数量。

示例

MicroLogix	说明
I:0	字 0
I/2	位 2
I:1/5	位 5 字 1

SLC 500 固定 I/O	说明
I:0	字 0
I:1	字 1
I/16	位 16
I:1/0	位 0 字 1 (与 I/16 相同)

PLC5*	说明
I:0	字 0
I:10	字 8 (八进制 10 = 十进制 8)
I/20	位 16 (八进制 20 = 十进制 16)
I:1/0	位 0 字 1 (与 I/20 相同)

*地址采用八进制形式。

SLC 500 模块化 I/O	说明
I:1	字 0 槽 1
I:1.0	字 0 插槽 1 (与 I:1 相同)
I:12	字 0 槽 12
I:12.2	字 2 槽 12
I:4.0/0	位 0 字 0 槽 4
I:4/0	位 0 槽 4 (与 I:4.0/0 相同)
I:4.2/0	位 0 字 2 槽 4
I:4/32	位 32 槽 4 (与 I:4.2/0 相同)

状况文件

要访问状况文件，请指定字和字中可选的位。默认数据类型以**粗体**显示。

语法	数据类型	访问
S:<字>	短整型、 字 、BCD、双字型、长整型、LBCD	读/写
S:<字> [行数][列数]	短整型、 字 、BCD、双字型、长整型、LBCD (数组类型)	读/写
S:<字> [列数]	短整型、 字 、BCD、双字型、长整型、LBCD (数组类型)	读/写
S:<字> /<位>	布尔型	读/写
S/位	布尔型	读/写

数组元素的数量 (以字节为单位) 不能超过指定的块请求大小。这意味着，在块请求大小为 32 字节的情况下，数组大小不能超过 16 个字。

范围

PLC 模型	最大字
MicroLogix	999
SLC 500 固定 I/O	96
SLC 500 模块化 I/O	999
PLC-5 系列	999

在作为 32 位数据类型 (如长整型、双字型或长整型 BCD) 进行访问时, 最大字位置会减一。

示例

示例	说明
S:0	字 0
S/26	位 26
S:4/15	位 15 字 4
S:10 [16]	从字 10 开始的 16 元素数组
S:0 [4] [8]	从字 0 开始的 4x8 元素数组

二进制文件

要访问二进制文件, 请指定文件编号、字和字中的位 (可选)。默认数据类型以**粗体**显示。

语法	数据类型	访问
B<文件>:<字>	短整型、 字 、BCD、双字型、长整型、LBCD	读/写
B<文件>:<字> [行数][列数]	短整型、 字 、BCD、双字型、长整型、LBCD (数组类型)	读/写
B<文件>:<字> [列]	短整型、 字 、BCD、双字型、长整型、LBCD (数组类型)	读/写
B<文件>:<字>/<位>	布尔型	读/写
B<文件>/位	布尔型	读/写

数组元素的数量 (以字节为单位) 不能超过指定的块请求大小。这意味着, 在块请求大小为 32 字节的情况下, 数组大小不能超过 16 个字。

范围

PLC 型号	文件编号	最大字
MicroLogix	3, 9-999	999
SLC 500 固定 I/O	3, 9-255	255
SLC 500 模块化 I/O	3, 9-999	999
PLC-5 系列	3-999	1999

在作为 32 位数据类型 (如长整型、双字型或长整型 BCD) 进行访问时, 最大字位置会减一。

示例

示例	说明
B3:0	字 0
B3/26	位 26
B12:4/15	位 15 字 4
B3:10 [20]	从字 10 开始的 20 元素数组
B15:0 [6] [6]	从字 0 开始的 6x6 元素数组

计时器文件

计时器文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

语法	数据类型	访问
T<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段的含义，请参阅 PLC 文档。

元素字段	数据类型	访问
ACC	短整型、字	读/写
PRE	短整型、字	读/写
DN	布尔型	只读
TT	布尔型	只读
EN	布尔型	只读

范围

PLC 型号	文件编号	最大元素
MicroLogix	4、9-999	999
SLC 500 固定 I/O	4、9-255	255
SLC 500 模块化 I/O	4、9-999	999
PLC-5 系列	3-999	1999

示例

示例	说明
T4:0.ACC	计时器 0 文件 4 的累加器
T4:10.DN	计时器 10 文件 4 的完成位
T15:0.PRE	计时器 0 文件 15 的预设

计数器文件

计数器文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

语法	数据类型	访问
C<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段的含义，请参阅 PLC 文档。

元素字段	数据类型	访问
ACC	字、短整型	读/写
PRE	字、短整型	读/写
UA	布尔型	只读
UN	布尔型	只读
OV	布尔型	只读
DN	布尔型	只读
CD	布尔型	只读
CU	布尔型	只读

范围

PLC 型号	文件编号	最大元素
MicroLogix	5、9-999	999
SLC 500 固定 I/O	5、9-255	255

PLC 型号	文件编号	最大元素
SLC 500 模块化 I/O	5, 9-999	999
PLC-5 系列	3-999	1999

示例

示例	说明
C5:0.ACC	计数器 0 文件 5 的累加器
C5:10.DN	计数器 10 文件 5 的完成位
C15:0.PRE	计数器 0 文件 15 的预设

控制文件

控制文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

语法	数据类型	访问
R<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段的含义，请参阅 PLC 文档。

元素字段	数据类型	访问
LEN	字、短整型	读/写
POS	字、短整型	读/写
FD	布尔型	只读
IN	布尔型	只读
UL	布尔型	只读
ER	布尔型	只读
EM	布尔型	只读
DN	布尔型	只读
EU	布尔型	只读
EN	布尔型	只读

范围

PLC 型号	文件编号	最大元素
MicroLogix	6, 9-999	999
SLC 500 固定 I/O	6, 9-255	255
SLC 500 模块化 I/O	6, 9-999	999
PLC-5 系列	3-999	1999

示例

示例	说明
R6:0.LEN	控制 0 文件 6 的长度字段
R6:10.DN	控制 10 文件 6 的完成位
R15:18.POS	控制 18 文件 15 的位置字段

整型文件

要访问整型文件，请指定文件编号、字和字中的某个位 (可选)。默认数据类型以**粗体**显示。

语法	数据类型	访问
N<文件>:<字>	短整型、 字 、BCD、双字型、长整型、LBCD	读/写

语法	数据类型	访问
N<文件>:<字> [行数][列数]	短整型、字、BCD、双字型、长整型、LBCD (数组类型)	读/写
N<文件>:<字> [列数]	短整型、字、BCD、双字型、长整型、LBCD (数组类型)	读/写
N<文件>:<字>/<位>	布尔型	读/写
N<文件>/位	布尔型	读/写

数组元素的数量 (以字节为单位) 不能超过指定的块请求大小。这意味着, 在块请求大小为 32 字节的情况下, 数组大小不能超过 16 个字。

范围

PLC 型号	文件编号	最大字
MicroLogix	7, 9-999	999
SLC 500 固定 I/O	7, 9-255	255
SLC 500 模块化 I/O	7, 9-999	999
PLC-5 系列	3-999	1999

在作为 32 位数据类型 (如长整型、双字型或长整型 BCD) 进行访问时, 最大字位置会减一。

示例

示例	说明
N7:0	字 0
N7/26	位 26
N12:4/15	位 15 字 4
N7:10 [8]	从字 10 开始的 8 元素数组
N15:0[4] [5]	从字 0 开始的 4x5 元素数组

浮点型文件

要访问浮点型文件, 请指定文件编号和元素。默认数据类型以**粗体**显示。

语法	数据类型	访问
F<文件>:<元素>	浮点型	读/写
F<文件>:<元素> [行数][列数]	浮点型 (数组类型)	读/写
F<文件>:<元素> [列数]	浮点型 (数组类型)	读/写

数组元素的数量 (以字节为单位) 不能超过指定的块请求大小。这意味着, 在块请求大小为 32 字节的情况下, 数组大小不能超过 8 位浮点型值。

范围

PLC 型号	文件编号	最大字
MicroLogix	8-999	999
SLC 500 固定 I/O	不适用	不适用
SLC 500 模块化 I/O	8-999	999
PLC-5 系列	3-999	1999

示例

示例	说明
F8:0	浮点数 0
F8:10 [16]	从字 10 开始的 16 元素数组
F15:0 [4] [4]	从字 0 开始的 16 元素数组

ASCII 文件

要访问 ASCII 文件数据，请指定文件编号和字符位置。默认数据类型以**粗体**显示。

语法	数据类型	访问
A<文件>:<字符>	字符、字节*	读/写
A<文件>:<字符> [行数][列数]	字符、字节*	读/写
A<文件>:<字符> [列数]	字符、字节*	读/写
A<文件>:<字偏移>/长度	字符串**	读/写

*数组元素的数量不能超过指定的块请求大小。PLC 在文件中每个字封装两个字符，其中高字节包含第一个字符，低字节包含第二个字符。PLC 编程软件允许字级或双字符级访问。Allen-Bradley ControlLogix Ethernet 驱动程序 允许访问字符级。

使用编程软件 "A10:0 = AB" 时，将导致 "A" 存储在 A10:0 的高字节中，而 "B" 存储在低字节中。使用 Allen-Bradley ControlLogix Ethernet 驱动程序 时，将进行以下两个分配："A10:0 = A" 和 "A10:1 = B"。这将导致存储在 PLC 内存中的数据相同。

**将此文件作为字符串数据引用时，可像编程软件一样访问字符串边界处的数据。长度最多可为 232 个字符。如果发送到设备的字符串长度小于地址指定的长度，则在将字符串发送到控制器之前，驱动程序会使用空值终止该字符串。

范围

PLC 型号	文件编号	最大字符数
MicroLogix	3-255	511
SLC 500 固定 I/O	不适用	不适用
SLC 500 模块化 I/O	9-999	1999
PLC-5 系列	3-999	1999

● **注意：**并非所有 MicroLogix 和 SLC 500 PLC 设备都支持 ASCII 文件类型。有关详细信息，请参阅 PLC 文档。

示例

示例	说明
A9:0	字符 0 (字 0 的高字节)
A27:10 [80]	从字符 10 开始的 80 字符数组
A15:0 [4] [16]	从字符 0 开始的 4x16 字符数组
A62:0/32	从字偏移 0 开始的 32 字符串

字符串文件

要访问字符串文件，请指定文件编号和元素。字符串是 82 个字符的空终止数组。驱动程序根据 PLC 返回的字符串长度放置空终止符。默认数据类型以**粗体**显示。

● **注意：**字符串文件不支持数组。

语法	数据类型	访问
ST<文件>:<元素>.<字段>	字符串	读/写

范围

PLC 型号	文件编号	最大字
MicroLogix	9-999	999
SLC 500 固定 I/O	不适用	不适用

PLC 型号	文件编号	最大字
SLC 500 模块化 I/O	9-999	999
PLC-5 系列	3-999	999

示例

示例	说明
ST9:0	字符串 0
ST18:10	字符串 10

BCD 文件

要访问 BCD 文件，请指定文件编号和字。默认数据类型以**粗体**显示。

PLC-5 语法

语法	数据类型	访问
D<文件>:<字>	BCD , LBCD	读/写
D<文件>:<字> [行数][列数]	BCD , LBCD (数组类型)	读/写
D<文件>:<字> [列]	BCD , LBCD (数组类型)	读/写

数组元素的数量 (以字节为单位) 不能超过指定的块请求大小。这意味着，在块请求大小为 32 字节的情况下，数组大小不能超过 16 (BCD 值)。

范围

PLC 型号	文件编号	最大字
MicroLogix	不适用	不适用
SLC 500 固定 I/O	不适用	不适用
SLC 500 模块化 I/O	不适用	不适用
PLC-5 系列	3-999	999

示例

示例	说明
D9:0	字 0
D27:10 [16]	从字 10 开始的 16 元素数组
D15:0 [4][8]	从字 0 开始的 32 元素数组

长整型文件

要访问长整型文件，请指定文件编号和元素。默认数据类型以**粗体**显示。

语法	数据类型	访问
L<文件>:<双字型>	长整型、 双字型 、LBCD	读/写
L<文件>:<双字型> [行数][列数]	长整型、 双字型 、LBCD (数组类型)	读/写
L<文件>:<双字型> [列数]	长整型、 双字型 、LBCD (数组类型)	读/写

数组元素的数量 (以字节为单位) 不能超过指定的块请求大小。这意味着，在块请求大小为 32 字节的情况下，数组大小不能超过 8 位长整型值。

范围

PLC 型号	文件编号	最大字
MicroLogix	9-999	999
SLC 500 固定 I/O	不适用	不适用

PLC 型号	文件编号	最大字
SLC 500 模块化 I/O	不适用	不适用
PLC-5 系列	不适用	不适用

示例

示例	说明
L9:0	字 0
L9:10 [8]	从字 10 开始的 8 元素数组
L15:0 [4] [5]	从字 0 开始的 4x5 元素数组

MicroLogix PID 文件

PID 文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

语法	数据类型	访问
PD<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关每个字段的含义，请参阅 PLC 文档。

元素字段	数据类型	访问
SPS	字、短整型	读/写
KC	字、短整型	读/写
TI	字、短整型	读/写
TD	字、短整型	读/写
MAXS	字、短整型	读/写
MINS	字、短整型	读/写
ZCD	字、短整型	读/写
CVH	字、短整型	读/写
CVL	字、短整型	读/写
LUT	字、短整型	读/写
SPV	字、短整型	读/写
CVP	字、短整型	读/写
TM	布尔型	读/写
AM	布尔型	读/写
CM	布尔型	读/写
OL	布尔型	读/写
RG	布尔型	读/写
SC	布尔型	读/写
TF	布尔型	读/写
DA	布尔型	读/写
DB	布尔型	读/写
UL	布尔型	读/写
LL	布尔型	读/写
SP	布尔型	读/写
PV	布尔型	读/写
DN	布尔型	读/写
EN	布尔型	读/写

范围

PLC 型号	文件编号	最大元素
MicroLogix	3-255	255
所有 SLC	不适用	不适用
PLC-5	PID 文件	PID 文件

示例

示例	说明
PD14:0.KC	PD 0 文件 14 的比例增益
PD18:6.EN	PD 6 文件 18 的 PID 启用位

PID 文件

PID 文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

PLC-5 语法

语法	数据类型	访问
PD<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段的含义，请参阅 PLC 文档。

元素字段	数据类型	访问
SP	实型	读/写
KP	实型	读/写
KI	实型	读/写
KD	实型	读/写
BIAS	实型	读/写
MAXS	实型	读/写
MINS	实型	读/写
DB	实型	读/写
SO	实型	读/写
MAXO	实型	读/写
MINO	实型	读/写
UPD	实型	读/写
PV	实型	读/写
ERR	实型	读/写
OUT	实型	读/写
PVH	实型	读/写
PVL	实型	读/写
DVP	实型	读/写
DVN	实型	读/写
PVDB	实型	读/写
DVDB	实型	读/写
MAXI	实型	读/写
MINI	实型	读/写
TIE	实型	读/写
FILE	字、短整型	读/写
ELEM	字、短整型	读/写
EN	布尔型	读/写

元素字段	数据类型	访问
CT	布尔型	读/写
CL	布尔型	读/写
PVT	布尔型	读/写
DO	布尔型	读/写
SWM	布尔型	读/写
CA	布尔型	读/写
MO	布尔型	读/写
PE,	布尔型	读/写
INI	布尔型	读/写
SPOR	布尔型	读/写
OLL	布尔型	读/写
OLH	布尔型	读/写
EWD	布尔型	读/写
DVNA	布尔型	读/写
DVHA	布尔型	读/写
PVLA	布尔型	读/写
PVHA	布尔型	读/写

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	不适用
SLC 500 固定 I/O	不适用	不适用
SLC 500 模块化 I/O	不适用	不适用
PLC-5 系列	3-999	999

示例

示例	说明
PD14:0.SP	PD 0 文件 14 的设定点字段
PD18:6.EN	PD 6 文件 18 的状态启用位

MicroLogix 消息文件

消息文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

语法	数据类型	访问
MG<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段的含义，请参阅 PLC 文档。

元素字段	数据类型	访问
IA	字、短整型	读/写
RBL	字、短整型	读/写
LBN	字、短整型	读/写
RBN	字、短整型	读/写
CHN	字、短整型	读/写
NOD	字、短整型	读/写
MTO	字、短整型	读/写
NB	字、短整型	读/写

元素字段	数据类型	访问
TFT	字、短整型	读/写
TFN	字、短整型	读/写
ELE	字、短整型	读/写
SEL	字、短整型	读/写
TO	布尔型	读/写
CO	布尔型	读/写
EN	布尔型	读/写
RN	布尔型	读/写
EW	布尔型	读/写
ER	布尔型	读/写
DN	布尔型	读/写
ST	布尔型	读/写
BK	布尔型	读/写

范围

PLC 型号	文件编号	最大元素
MicroLogix	3-255	255
所有 SLC	不适用	不适用
PLC5	消息文件	消息文件

示例

示例	说明
MG14:0.TO	数据文件 14 中 MSG 元素 0 的超时位
MG18:6.CO	数据文件 18 中 MSG 元素 6 的继续扫描位

消息文件

消息文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

PLC-5 语法

语法	数据类型	访问
MG<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段的含义，请参阅 PLC 文档。

元素字段	数据类型	访问
ERR	短整型 、字	读/写
RLEN	短整型 、字	读/写
DLEN	短整型 、字	读/写
EN	布尔型	读/写
ST	布尔型	读/写
DN	布尔型	读/写
ER	布尔型	读/写
CO	布尔型	读/写
EW	布尔型	读/写
NR	布尔型	读/写
TO	布尔型	读/写

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	不适用
SLC 500 固定 I/O	不适用	不适用
SLC 500 模块化 I/O	不适用	不适用
PLC-5 系列	3-999	999

示例

示例	说明
MG14:0.RLEN	MG 0 文件 14 的所请求的长度字段
MG18:6.CO	MG 6 文件 18 的继续扫描位

块传输文件

块传送文件是一种通过指定文件编号、元素和字段来访问其数据的结构化类型。默认数据类型以**粗体**显示。

PLC-5 语法

语法	数据类型	访问
BT<文件>:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段含义的详细信息，请参阅 PLC 文档。

元素字段	数据类型	访问
RLEN	字、短整型	读/写
DLEN	字、短整型	读/写
FILE	字、短整型	读/写
ELEM	字、短整型	读/写
RW	布尔型	读/写
ST	布尔型	读/写
DN	布尔型	读/写
ER	布尔型	读/写
CO	布尔型	读/写
EW	布尔型	读/写
NR	布尔型	读/写
TO	布尔型	读/写

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	不适用
SLC 500 固定 I/O	不适用	不适用
SLC 500 模块化 I/O	不适用	不适用
PLC-5 系列	3-999	1999

示例

示例	说明
BT14:0.RLEN	BT 0 文件 14 的所请求的长度字段
BT18:6.CO	BT 6 文件 18 的继续扫描位

功能文件

有关 ENI MicroLogix 和 MicroLogix 1100 设备模型所支持文件的信息，请从下表选择一个链接。

[高速计数器文件 \(HSC\)](#)

[实时时钟文件 \(RTC\)](#)

[信道 0 通信状况文件 \(CS0\)](#)

[通道 1 通信状况文件 \(CS1\)](#)

[I/O 模块状况文件 \(IOS\)](#)

有关设备模型及其所支持文件的详细信息，请参阅[地址说明](#)。

高速计数器文件 (HSC)

HSC 文件是一种通过指定元素和字段来访问数据的结构化类型。默认数据类型以**粗体**显示。

另请参阅：[ENI DF1/DH+/ControlNet 网关通信参数](#)

语法	数据类型	访问
HSC:<元素>.<字段>	取决于字段。	取决于字段。

每个元素允许的字段如下。有关各字段的含义，请参阅 PLC 文档。

元素字段	默认类型	访问
ACC	双字型 、长整型	只读
HIP	双字型 、长整型	读/写
LOP	双字型 、长整型	读/写
OVF	双字型 、长整型	读/写
UNF	双字型 、长整型	读/写
PFN	字 、短整型	只读
ER	字 、短整型	只读
MOD	字 、短整型	只读
OMB	字 、短整型	只读
HPO	字 、短整型	读/写
LPO	字 、短整型	读/写
UIX	布尔型	只读
UIP	布尔型	只读
AS	布尔型	只读
ED	布尔型	只读
SP	布尔型	只读
LPR	布尔型	只读
HPR	布尔型	只读
DIR	布尔型	只读
CD	布尔型	只读
CU	布尔型	只读
UIE	布尔型	读/写
UIL	布尔型	读/写
FE	布尔型	读/写
CE	布尔型	读/写
LPM	布尔型	读/写
HPM	布尔型	读/写
UFM	布尔型	读/写

元素字段	默认类型	访问
OFM	布尔型	读/写
LPI	布尔型	读/写
HPI	布尔型	读/写
UFI	布尔型	读/写
OFI	布尔型	读/写
UF	布尔型	读/写
OF	布尔型	读/写
MD	布尔型	读/写

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	254
所有 SLC	不适用	不适用
PLC5	不适用	不适用

示例

示例	说明
HSC:0.OMB	输出高速计数器 0 的掩码设置。
HSC:1.ED	高速计数器 1 中检测到指示器错误。

实时时钟文件 (RTC)

RTC 文件是一种通过指定元素和字段来访问数据的结构化类型。默认数据类型以**粗体**显示。

● 另请参阅: [ENI DF1/DH+/ControlNet 网关通信参数](#)

语法	数据类型	访问
RTC:<元素>.<字段>	取决于字段	取决于字段

每个元素允许的字段如下。有关各字段的含义, 请参阅 PLC 文档。

元素字段	数据类型	访问
YR	字、短整型	读/写
MON	字、短整型	读/写
DAY	字、短整型	读/写
HR	字、短整型	读/写
MIN	字、短整型	读/写
SEC	字、短整型	读/写
DOW	字、短整型	读/写
DS	布尔型	只读
BL	布尔型	只读
_SET (用于块写入)	布尔型	读/写

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	254
所有 SLC	不适用	不适用
PLC5	不适用	不适用

示例

示例	说明
RTC:0.YR	实时时钟 0 的年设置。
RTC:0.BL	实时时钟 0 的电池电量低指示器。

信道 0 通信状况文件 (CS0)

要访问信道 0 的通信状况文件，请指定一个字 (或者字中的某位)。默认数据类型以**粗体**显示。

● 另请参阅: [ENI DF1/DH+/ControlNet 网关通信参数](#)

语法	数据类型	访问
CS0:<字>	短整型、 字 、BCD、双字型、长整型、LBCD	取决于 <字> 和 <位>
CS0:<字>/<位>	布尔型	取决于 <字> 和 <位>
CS0/位	布尔型	取决于 <字> 和 <位>

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	254
所有 SLC	不适用	不适用
PLC5	不适用	不适用

示例

示例	说明
CS0:0	字 0。
CS0:4/2	位 2 字 4 = MCP。

● 有关 CS0 字/位含义的详细信息，请参阅 *Rockwell* 文档。

信道 1 通信状况文件 (CS1)

要访问信道 1 的通信状况文件，请指定一个字 (或者字中的某位)。默认数据类型以**粗体**显示。

● 另请参阅: [ENI DF1/DH+/ControlNet 网关通信参数](#)

语法	数据类型	访问
CS1:<字>	短整型、 字 、BCD、双字型、长整型、LBCD	取决于 <字> 和 <位>
CS1:<字>/<位>	布尔型	取决于 <字> 和 <位>
CS1/位	布尔型	取决于 <字> 和 <位>

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	254
所有 SLC	不适用	不适用
PLC5	不适用	不适用

示例

示例	说明
CS1:0	字 0。
CS1:4/2	位 2 字 4 = MCP。

● 有关 CS1 字/位含义的详细信息，请参阅 *Rockwell* 文档。

I/O 模块状况文件 (IOS)

要访问 I/O 模块状况文件，请指定字和位 (后者可选)。默认数据类型以**粗体**显示。

● 另请参阅: [ENI DF1/DH+/ControlNet 网关通信参数](#)

语法	数据类型	访问
IOS:<字>	短整型、字、BCD、双字型、长整型、LBCD	取决于 <字> 和 <位>
IOS:<字>/<位>	布尔型	取决于 <字> 和 <位>
IOS/位	布尔型	取决于 <字> 和 <位>

范围

PLC 型号	文件编号	最大元素
MicroLogix	不适用	254
所有 SLC	不适用	不适用
PLC5	不适用	不适用

示例

示例	说明
IOS:0	字 0。
IOS:4/2	位 2 字 4。

● **注意:** 有关 1769 扩展 I/O 状况代码的列表，请参阅说明手册。

自动生成标记数据库

Allen-Bradley ControlLogix Ethernet 驱动程序 可配置为在服务器内自动生成服务器标记列表 (标记与特定于设备的数据相对应)。自动生成的标记基于在 Logix 设备中定义的 Logix 标记，并可通过 OPC 客户端浏览。Logix 标记可以是原子型或结构化类型。结构和数组标记可以快速增加导入标记的数目 (从而增加服务器中可用标记的数目)。

● **注意:** ENI/DH+、ControlNet 网关以及 MicroLogix 型号不支持“自动生成标记数据库”: 仅 ENI ControlLogix、CompactLogix 和 FlexLogix 型号支持。

原子型标记 -> 一对一 -> 服务器标记
结构标记 -> 一对多 -> 服务器标记
数组标记 -> 一对多 -> 服务器标记

● 有关数据库创建设置的详细信息，请参阅服务器帮助文件。

● **注意:** 对于 RSLogix5000 编程环境中出现的控制器标记，必须将要读取的标记的“外部访问”属性配置为“只读”或“读/写”。默认情况下，自动生成的标记可能会将外部访问设置为“无”。要读取控制器标记，请根据需要在 RSLogix 的附加指令参数中重新配置外部访问。请参阅制造商文档。

标记层次

自动标记生成所创建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)。要使用此功能，请启用设备属性中的“允许子组”(Allow Sub Groups)。

已扩展模式

当处于“已扩展”(Condensed) 模式时，自动标记生成所创建的服务器标记遵循的组/标记层次与 RSLogix 5000 中的标记层次一致。如在“已压缩”(Condensed) 模式下一样，会为每个段 (前面带有句点) 创建组，但还会以逻辑分组的形式创建这些组。创建的组包括以下内容：

- 全局 (控制器) 范围
- 程序范围

- 结构和子结构
- 数组

● **注意:** 不会为 .bit 地址创建组。

根级组 (或“父组”(Parent Group) 中指定组的子组级别) 是 "Prgm_<程序名称>" 和 "Global"。控制器中的每个程序都有自己的 "Prgm_<程序名称>" 组。驱动程序将此识别为第一组级别。

基本全局标记 (或非结构、非数组标记) 位于“全局”组下; 基本“程序”标记位于其各自的程序组下。以父组中各自子组的形式提供每个结构和数组标记。以此方式组织数据后, 服务器的标记视图会模拟 RSLogix5000。

结构/数组子组的名称同样提供对结构/数组的说明。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"; 其中 x 表示存在 1 维, y 表示存在 2 维。数组子组内的标记都是该数组的元素 (除非明确限定)。结构子组中的标记是结构成员本身。如果结构中包含数组, 则也会在结构组中创建数组子组。

处理复杂项目时, 标记层次可能需要大量组级别。通过自动标记生成创建的组级别最大为七级。其中不包括在“将生成的标记添加到以下组”中指定的组。当需要多于七个级别时, 标记被放置在第七组中 (使层次趋于稳定)。

数组标记

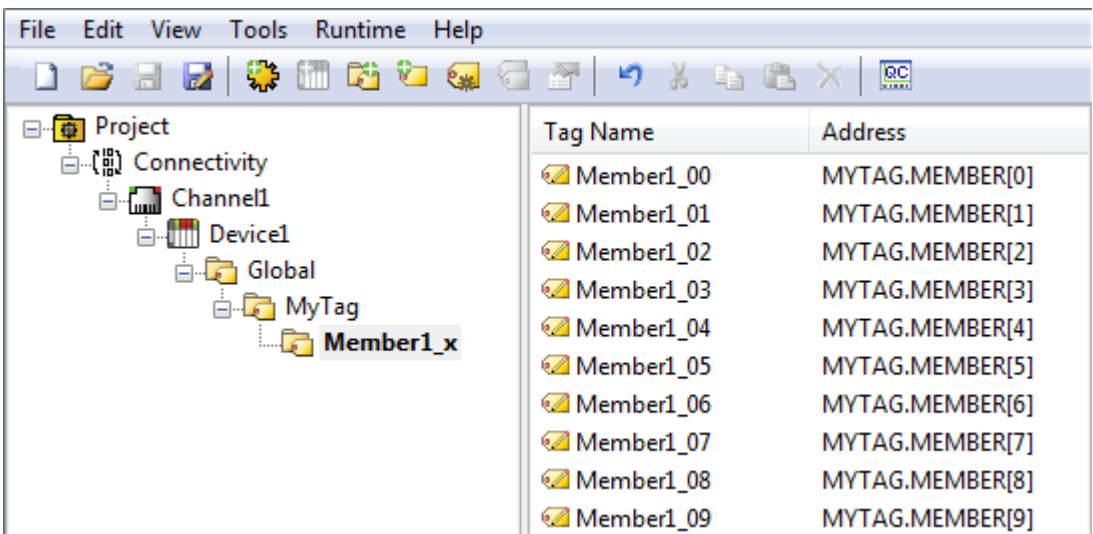
为包含数组元素的每个数组创建一个组。组名称带有符号: <数组名称>_x_y_z, 其中:

x_y_z = 3 维数组
x_y = 2 维数组
x = 1 维数组

数组标记带有符号: <标记元素>_XXXXX_YYYYY_ZZZZZ。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



复杂示例

使用地址 "Local:1:O.Slot[9].Data" 定义了一个 Logix 标记。它会分别在组 "Global" - "Local_1_O" - "Slot_x" - "Slot_09" 中表示。最后一组中将为标记 "Data"。

"Data" 的静态参考为 "Channel1.Device1.Global.Local_1_O.Slot_x.Slot_09.Data"。"Data" 的动态参考为 "Channel1.Device1.Local:1:O.Slot[9].Data"。

已压缩模式

在“已压缩”(Condensed) 模式下，自动标记生成所创建的服务器标记遵循的组/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：**不会为数组或 .bit 地址创建组。

处理复杂项目时，易于了解标记层次如何需要大量组级别。通过自动标记生成创建的组级别最大为七级。其中不包括在“将生成的标记添加到以下组”中指定的组。当需要多于七个级别时，标记被放置在第七组中 (使层次趋于稳定)。

● **注意：**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，因为服务器不支持前导下划线。有关详细信息，请参阅[控制器到服务器的名称转换](#)。

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

File Edit View Tools Runtime Help																							
	<table><tr><th>Tag Name</th><th>Address</th></tr><tr><td>Member1_[0]</td><td>MYTAG.MEMBER[0]</td></tr><tr><td>Member1_[1]</td><td>MYTAG.MEMBER[1]</td></tr><tr><td>Member1_[2]</td><td>MYTAG.MEMBER[2]</td></tr><tr><td>Member1_[3]</td><td>MYTAG.MEMBER[3]</td></tr><tr><td>Member1_[4]</td><td>MYTAG.MEMBER[4]</td></tr><tr><td>Member1_[5]</td><td>MYTAG.MEMBER[5]</td></tr><tr><td>Member1_[6]</td><td>MYTAG.MEMBER[6]</td></tr><tr><td>Member1_[7]</td><td>MYTAG.MEMBER[7]</td></tr><tr><td>Member1_[8]</td><td>MYTAG.MEMBER[8]</td></tr><tr><td>Member1_[9]</td><td>MYTAG.MEMBER[9]</td></tr></table>	Tag Name	Address	Member1_[0]	MYTAG.MEMBER[0]	Member1_[1]	MYTAG.MEMBER[1]	Member1_[2]	MYTAG.MEMBER[2]	Member1_[3]	MYTAG.MEMBER[3]	Member1_[4]	MYTAG.MEMBER[4]	Member1_[5]	MYTAG.MEMBER[5]	Member1_[6]	MYTAG.MEMBER[6]	Member1_[7]	MYTAG.MEMBER[7]	Member1_[8]	MYTAG.MEMBER[8]	Member1_[9]	MYTAG.MEMBER[9]
Tag Name	Address																						
Member1_[0]	MYTAG.MEMBER[0]																						
Member1_[1]	MYTAG.MEMBER[1]																						
Member1_[2]	MYTAG.MEMBER[2]																						
Member1_[3]	MYTAG.MEMBER[3]																						
Member1_[4]	MYTAG.MEMBER[4]																						
Member1_[5]	MYTAG.MEMBER[5]																						
Member1_[6]	MYTAG.MEMBER[6]																						
Member1_[7]	MYTAG.MEMBER[7]																						
Member1_[8]	MYTAG.MEMBER[8]																						
Member1_[9]	MYTAG.MEMBER[9]																						

复杂示例

使用地址 "Local:1:O.Slot[9].Data" 定义 Logix 标记。它会在组 "Local:1:O" -> "Slot[9]" 中表示。最后一组中将为标记 "Data"。

"Data" 的静态参考为 "Channel1.Device1.Local:1:O.Slot[9].Data"。动态参考为 "Channel1.Device1.Local:1:O.Slot[9].Data"。

● **注意：**无法在“离线”模式下直接导入 I/O 模块标记。因为可以导入别名，所以建议针对 RSLogix5000 中感兴趣的 I/O 模块标记创建别名。

控制器到服务器的名称转换

前导下划线

标记或项目群名称中的前导下划线 "_" 将替换为 "U_". 必需如此, 因为服务器不接受以下划线开头的标记或组名称。

长名称 (OPC 服务器 4.64 及以下版本)

在较旧的 OPC 服务器版本下, Allen-Bradley ControlLogix Ethernet 驱动程序 的组和标记名称限制为 31 个字符。因此, 如果控制器项目群或标记名称超出 31 个字符, 必须对其进行修剪。OPC 服务器 4.70 及以上版本限制为 256 个字符, 因此, 这些规则不适用。名称修剪示例如下:

非数组

1. 为此标记确定 5 位唯一 ID。
2. 指定标记名称: ThisIsALongTagNameAndProbablyExceeds31
3. 修剪到 31 个字符: ThisIsALongTagNameAndProbablyEx
4. 为唯一 ID 留出空间: ThisIsALongTagNameAndProba#####
5. 插入此 ID: ThisIsALongTagNameAndProba00000

数组

1. 为此数组确定 5 位唯一 ID。
2. 指定数组标记名称: ThisIsALongTagNameAndProbablyExceeds31_23_45_8
3. 在保留元素值的同时将标记修剪到 31 个字符: ThisIsALongTagNameAndPr_23_45_8
4. 为唯一 ID 留出空间: ThisIsALongTagName#####_23_45_8
5. 插入此 ID: ThisIsALongTagName00001_23_45_8

修剪长项目群名称的方法与修剪长非数组标记名称的方法相同。对于修剪的每个标记或项目群名称, 唯一 ID 都是递增的。已修剪数组名称的数组标记名称 (元素) 均具有相同的唯一 ID。这可支持 100000 个唯一标记/项目群名称。

● **注意:** 如果启用了“限制名称长度”, 即使支持 256 个字符的名称, 这些规则也仍然适用。有关详细信息, 请参阅 [Logix 数据库设置](#)。

自动生成标记数据库前准备

有关使用“自动生成标记数据库”的信息, 请按以下说明进行操作。

在线

建议在数据库创建过程中, 停止所关注的 Logix CPU 的所有通信。

在 RSLogix5000 中

将项目设置为“离线”。

在服务器中

1. 查看要为其生成标记的设备的设备属性。
2. 在“Logix 数据库设置”的“数据库导入方法”中选择“从设备创建”。
3. 在“Logix 数据库选项”中, 执行任何所需的更改, 然后单击“确定”。
4. 在“Logix 数据库筛选”中, 执行任何所需的更改, 然后单击“确定”。
5. 选择“标记生成”, 并在“创建”下单击蓝色链接以“创建标记”。

● **注意:** 在“Logix 选项”中, 将“协议模式”设置为“符号”并将“默认数据类型”设置为“默认”, 以便以控制器中使用的数据类型导入标记。

离线

Allen-Bradley ControlLogix Ethernet 驱动程序 使用从 RSLogix5000 中生成的名为 L5K/L5X 导入/导出文件的文件来生成标记数据库。

在 RSLogix5000 中

1. 打开包含将移入 OPC 服务器的标记的项目。
2. 单击“文件”|“另存为”。
3. 选择 **L5K/L5X Import/Export File**, 然后指定一个名称。RSLogix 会将该项目的内容导出到此 L5K/L5X 文件中。

在 OPC 服务器中

1. 打开要为其生成标记的设备的设备属性。
2. 选择“Logix 数据库设置”，然后在“数据库导入方法”中选择“从导入文件创建”。
3. 输入或浏览至先前创建的文件位置。
4. 在“Logix 数据库选项”中，执行任何所需的更改，然后单击“确定”。
5. 在“Logix 数据库筛选”中，执行任何所需的更改，然后单击“确定”。
6. 选择“标记生成”，并在“创建”下单击蓝色链接以“[创建标记](#)”。

● **注意：**导入的预定义标记类型基于驱动程序支持的最新版本。有关详细信息，请参阅固件版本。

错误代码

以下部分定义可能在服务器的事件日志中遇到的错误代码。有关特定错误代码类型的详细信息，请从下表中选择一个链接。

[封装错误代码](#)

[CIP 错误代码](#)

封装错误代码

以下错误代码为十六进制。

错误代码	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	无效的会话 id。
0065	标题长度无效。
0069	请求的协议版本不受支持。
0070	无效的目标 id。

CIP 错误代码

以下错误代码为十六进制。

错误代码	日志代码	说明
0001	0x01	连接失败。*
0002	0x02	资源不足。
0003	0x03	值无效。
0004	0x04	IOI 无法被解密或标记不存在。
0005	0x05	未知目标。
0006	0x06	请求的数据不适合响应数据包。
0007	0x07	失去连接。
0008	0x08	不支持的服务。
0009	0x09	数据段错误或属性值无效。
000A	0x0A	属性列表错误。
000B	0x0B	状态已经存在。
000C	0x0C	对象模型冲突。
000D	0x0D	对象已经存在。
000E	0x0E	属性不可设置。
000F	0x0F	权限被拒绝。
0010	0x10	设备状态冲突。
0011	0x11	回复不适合。
0012	0x12	片段原型。
0013	0x13	为执行服务指定的命令数据/参数不足。
0014	0x14	不支持属性。
0015	0x15	指定的数据过多。
001A	0x1A	桥接请求过大。
001B	0x1B	桥接响应过大。
001C	0x1C	属性列表短缺。
001D	0x1D	属性列表无效。

错误代码	日志代码	说明
001E	0x1E	嵌入式服务错误。
001F	0x1F	连接时失败。**
0022	0x22	收到的回复无效。
0025	0x25	关键段错误。
0026	0x26	指定的 IOI 字数与 IOI 字数统计不匹配。
0027	0x27	列表中存在意外的属性。

• *另请参阅：[0x0001 扩展错误代码](#)

• **另请参阅：[0x001F 扩展错误代码](#)

Logix5000 特定 (1756-L1) 错误代码

以下错误代码为十六进制。

错误代码	说明
00FF	一般错误。*

• *另请参阅：[0x00FF 扩展错误代码](#)

• 另请参阅：对于未列出的错误代码，请参阅 Rockwell 文档。

0x0001 扩展错误代码

以下错误代码为十六进制。

错误代码	说明
0100	正在使用连接。
0103	不支持传输。
0106	所有权冲突。
0107	未找到连接。
0108	连接类型无效。
0109	连接大小无效。
0110	未配置模块。
0111	不支持 EPR。
0114	模块错误。
0115	设备类型错误。
0116	修订版本错误。
0118	配置格式无效。
011A	应用程序超出连接数。
0203	连接超时。
0204	未连接消息超时。
0205	未连接发送参数错误。
0206	消息过大。
0301	无缓冲区内存。
0302	带宽不可用。
0303	无可用的筛选器。
0305	签名匹配。
0311	端口不可用。
0312	链路地址不可用。
0315	段类型无效。
0317	未计划连接。

错误代码	说明
0318	至自身的链路地址无效。

● **注意：**对于未列出的错误代码，请参阅 Rockwell 文档。

0x001F 扩展错误代码

以下错误代码为十六进制。

错误代码	说明
0203	连接超时。

● **注意：**对于未列出的错误代码，请参阅 Rockwell 文档。

0x00FF 扩展错误代码

以下错误代码为十六进制。

错误代码	说明
2104	地址超出范围。
2105	尝试进行超出数据对象末端的访问。
2106	正在使用数据。
2107	数据类型无效或不受支持。

● **注意：**对于未列出的错误代码，请参阅 Rockwell 文档。

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。

附录

有关特定主题的详细信息，请从下表选择一个链接。

- [通道属性](#)
- [设备属性](#)
- [标记属性](#)

- [Logix 设置](#)
- [1761-NET-ENI 设置](#)
- [Data Highway Plus 网关设置](#)
- [通信路由](#)
- [串行网关设置](#)
- [Data Highway Plus 网关](#)
- [ControlNet 网关](#)
- [EtherNet/IP 网关设置](#)
- [MicroLogix 1100 设置](#)
- [选择协议模式](#)
- [检测“控制器项目”中所做的更改](#)
- [SoftLogix 5800 连接注意事项](#)
- [术语表](#)

附录 A - 通道属性

以下是所有 Allen-Bradley ControlLogix Ethernet 通道级属性的列表。

```
{ "common.ALLTYPES_NAME": "MyChannel", "common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet",
"servermain.CHANNEL_DIAGNOSTICS_CAPTURE": false, "servermain.CHANNEL_UNIQUE_ID":
4126021724, "servermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING":
"", "servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2, "servermain.CHANNEL_WRITE_
OPTIMIZATIONS_DUTY_CYCLE": 10, "servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_
HANDLING": 0, }
```

附录 B - 设备属性

以下是所有 Allen-Bradley ControlLogix Ethernet 设备级属性的列表。

```
{ "common.ALLTYPES_NAME": "MyDevice", "common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet",
"servermain.DEVICE_MODEL": 0, "servermain.DEVICE_UNIQUE_ID": 1286734974,
"servermain.DEVICE_CHANNEL_ASSIGNMENT": "Channel1", "servermain.DEVICE_ID_FORMAT":
0, "servermain.DEVICE_ID_STRING": "<10.10.110.15>,1,0", "servermain.DEVICE_ID_
HEXADECIMAL": 0, "servermain.DEVICE_ID_DECIMAL": 0, "servermain.DEVICE_ID_OCTAL":
0, "servermain.DEVICE_DATA_COLLECTION": true, "servermain.DEVICE_SIMULATED":
false, "servermain.DEVICE_SCAN_MODE": 0, "servermain.DEVICE_SCAN_MODE_RATE_MS":
1000, "servermain.DEVICE_SCAN_MODE_PROVIDE_INITIAL_UPDATES_FROM_CACHE": false,
"servermain.DEVICE_CONNECTION_TIMEOUT_SECONDS": 3, "servermain.DEVICE_REQUEST_
TIMEOUT_MILLISECONDS": 1000, "servermain.DEVICE_RETRY_ATTEMPTS": 3,
"servermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS": 0, "servermain.DEVICE_AUTO_
DEMOTION_ENABLE_ON_COMMUNICATIONS_FAILURES": false, "servermain.DEVICE_AUTO_
DEMOTION_DEMOTE_AFTER_SUCCESIVE_TIMEOUTS": 3, "servermain.DEVICE_AUTO_DEMOTION_
PERIOD_MS": 10000, "servermain.DEVICE_AUTO_DEMOTION_DISCARD_WRITES": false,
"servermain.DEVICE_TAG_GENERATION_ON_STARTUP": 0, "servermain.DEVICE_TAG_
GENERATION_DUPLICATE_HANDLING": 0, "servermain.DEVICE_TAG_GENERATION_GROUP": "",
"servermain.DEVICE_TAG_GENERATION_ALLOW_SUB_GROUPS": true, "controllogix_
```

```

ethernet.DEVICE_PORT_NUMBER": 44818, "controllogix_ethernet.DEVICE_CONNECTION_
SIZE_BYTES": 500, "controllogix_ethernet.DEVICE_INACTIVITY_WATCHDOG_SECONDS": 32,
"controllogix_ethernet.DEVICE_ARRAY_BLOCK_SIZE_ELEMENTS": 120, "controllogix_
ethernet.DEVICE_PROTOCOL_MODE": 1, "controllogix_ethernet.DEVICE_ONLINE_EDITS":
true, "controllogix_ethernet.DEVICE_OFFLINE_EDITS": true, "controllogix_
ethernet.DEVICE_AUTOMATICALLY_READ_STRING_LENGTH": true, "controllogix_
ethernet.DEVICE_DEFAULT_DATA_TYPE": -1, "controllogix_ethernet.DEVICE_ENABLE_
PERFORMANCE_STATISTICS": false, "controllogix_ethernet.DEVICE_DATABASE_IMPORT_
METHOD": 1, "controllogix_ethernet.DEVICE_TAG_IMPORT_FILE": "Veracode2.L5X",
"controllogix_ethernet.DEVICE_DISPLAY_DESCRIPTIONS": true, "controllogix_
ethernet.DEVICE_LIMIT_TAG_NAMES": false, "controllogix_ethernet.DEVICE_TAG_
HIERARCHY": 1, "controllogix_ethernet.DEVICE_IMPOSE_ARRAY_ELEMENT_COUNT_LIMIT":
false, "controllogix_ethernet.DEVICE_ARRAY_ELEMENT_COUNT_LIMIT": 2000,
"controllogix_ethernet.DEVICE_CL_ENET_PORT_NUMBER": 44818, "controllogix_
ethernet.DEVICE_REQUEST_SIZE": 232, "controllogix_ethernet.DEVICE_PERFORM_BLOCK_
WRITES": false, "controllogix_ethernet.DEVICE_SLOT_CONFIGURATION": {
"controllogix_ethernet.DEVICE_SLOT_CONFIGURATION_MODULE": [ 255, 255, 255,
255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255, 255, 255, 255, 255, 255, 255, 255, 255 ], "controllogix_ethernet.DEVICE_
SLOT_CONFIGURATION_INPUT_WORDS": [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ], "controllogix_ethernet.DEVICE_SLOT_
CONFIGURATION_OUTPUT_WORDS": [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] } }

```

附录 C - 标记属性

以下是所有 Allen-Bradley ControlLogix Ethernet 标记属性的列表。

```

{ "common.ALLTYPES_NAME": "MyTag1", "common.ALLTYPES_DESCRIPTION": "",
"servermain.TAG_ADDRESS": "MyTag1", "servermain.TAG_DATA_TYPE": 6,
"servermain.TAG_READ_WRITE_ACCESS": 1, "servermain.TAG_SCAN_RATE_MILLISECONDS":
100, "servermain.TAG_AUTOGENERATED": true, "servermain.TAG_SCALING_TYPE": 0,
"servermain.TAG_SCALING_RAW_LOW": 0, "servermain.TAG_SCALING_RAW_HIGH": 1000,
"servermain.TAG_SCALING_SCALED_DATA_TYPE": 9, "servermain.TAG_SCALING_SCALED_LOW":
0, "servermain.TAG_SCALING_SCALED_HIGH": 1000, "servermain.TAG_SCALING_CLAMP_LOW":
false, "servermain.TAG_SCALING_CLAMP_HIGH": false, "servermain.TAG_SCALING_NEGATE_
VALUE": false, "servermain.TAG_SCALING_UNITS": "" }

```

Logix 设备 ID

有关 ENI 设备 ID 设置的信息，请参阅 [1761-NET-ENI 设置](#)。

MicroLogix 5500 以太网

设备 ID 指定了设备 IP 地址和控制器 CPU 的插槽编号。设备 ID 指定如下：

<IP 或主机名>,1,<可选路由路径>,<CPU 插槽>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	IP 地址或主机名。	十进制	0-255
1	端口 ID	底面端口。	十进制	1
可选的路由路径	多个链接、端口对	在 EtherNet/IP 接口模块之外指定一个通路，且必须等于 1 (底面端口)。	十进制	*
CPU 插槽	链路地址	ControlLogix 处理器的插槽编号。	十进制	0-255

● *有关详细信息，请参阅[连接路径规范](#)。

示例

123.123.123.123,1,0

这相当于 123.123.123.123 Ethernet/IP。端口 ID 为 1，CPU 插槽编号为 0。

CompactLogix 5300 以太网设备 ID

设备 ID 指定了设备 IP 地址和控制器 CPU 的插槽编号。设备 ID 指定如下：

<IP 或主机名>,1,<可选路由路径>,<CPU 插槽>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	CompactLogix 以太网 IP 地址或主机名。	十进制	0-255
1	端口 ID	底面端口。	十进制	1
可选的路由路径	多个链接、端口对	在以太网端口之外指定一个通路，且必须等于 1 (底面端口)。	十进制	*
CPU 插槽	链路地址	CompactLogix 处理器的插槽编号。	十进制	0-255

● *有关详细信息，请参阅[连接路径规范](#)。

示例

123.123.123.123,1,0

这相当于 123.123.123.123 CompactLogix IP。端口 ID 为 1，CPU 插槽编号为 0。

FlexLogix 5400 以太网设备 ID

设备 ID 指定了设备 IP 地址和控制器 CPU 的插槽编号。设备 ID 指定如下：

<IP 或主机名>,1,<可选路由路径>,<CPU 插槽>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	1788-ENBT IP 地址或主机名。	十进制	0-255
1	端口 ID	底面端口。	十进制	1
可选的路由路径	多个链接、端口对	在 1788-ENBT 接口模块之外指定一个通路，且必须等于 1 (底面端口)。	十进制	*
CPU 插槽	链路地址	FlexLogix 处理器的插槽编号。	十进制	0-255

● *有关详细信息，请参阅[连接路径规范](#)。

示例

123.123.123.123,1,0

这相当于 123.123.123.123 1788-ENBT IP。端口 ID 为 1，CPU 插槽编号为 0。

SoftLogix 5800 设备 ID

设备 ID 指定了 SoftLogix PC IP 地址和控制器 CPU 的虚拟插槽编号。设备 ID 指定如下：

<IP 或主机名>,1,<可选路由路径>,<CPU 插槽>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	SoftLogix PC NIC IP 地址或主机名。	十进制	0-255
1	端口 ID	底面端口。	十进制	1
可选的路由路径	多个链接、端口对	在 EtherNet/IP 消息模块之外指定一个通路，且必须等于 1 (虚拟底面端口)。	十进制	*
CPU 插槽	链路地址	虚拟背板内的 SoftLogix 处理器的插槽编号。	十进制	0-255

● *有关详细信息，请参阅[连接路径规范](#)。

示例

123.123.123.123,1,1

这相当于 123.123.123.123 SoftLogix PC IP 地址。端口 ID 为 1，CPU 插槽编号为 1。

● 有关利用远程背板的路由路径来补充设备 ID 的信息，请参阅[路由通信](#)。

● 另请参阅：[SoftLogix 5800 连接注意事项](#)

1761-NET-ENI 设置

1761-NET-ENI 提供了一种使用 Allen-Bradley ControlLogix Ethernet 驱动程序在以太网上与 ControlLogix、CompactLogix、FlexLogix、MicroLogix、SLC 500 和 PLC 5 系列 PLC 进行通信的方法。

要求

支持全双工 DF1 且利用 CH0 RS232 信道的 MicroLogix、SLC 500 或 PLC 5 系列 PLC。
1761-NET-ENI 设备系列 A、B、C 或 D。

利用 CH0 RS232 信道的 ControlLogix、CompactLogix 或 FlexLogix PLC。
1761-NET-ENI 设备系列 B 及更高版本。

● 注意：

- 1. 有关通信参数、数据库设置和项目/协议选项的信息，ENI ControlLogix、CompactLogix 和 FlexLogix 用户应参阅目录中的“Logix 设置”部分。
- 2. 要开启“CompactLogix 路由”选项 (位于实用工具的“ENI IP 地址”选项卡中)，请使用由 Allen-Bradley 提供的 ENI/ENIW 实用工具。已在固件版本 2.31 的 ENI 模块上对此进行测试。

● ENI 模块的 TCP 连接数有限。因此，用户应避免应用程序与模块 (如 RSLinx/RWho) 进行通信，以使连接可用于驱动程序。

ENI 设备 ID

设备 ID 指定了 1761-NET-ENI 的 IP 地址。设备 ID 指定如下：

<IP 地址>

指示符	指示符类型	说明	格式	范围
IP 地址	不适用	1761-NET-ENI IP 地址	十进制	0-255

示例
123.123.123.123

这相当于 123.123.123.123 的 ENI IP。由于设备仅支持全双工 DF1，因此无需节点 ID。

● 有关通信参数的详细信息，请参阅 [Logix 通信参数](#)。

Data Highway Plus 网关设置

DH+ 网关提供了一种通过 Allen-Bradley ControlLogix Ethernet 驱动程序与 DH+ 上的 SLC 500 和 PLC-5 系列 PLC 进行通信的方法。

要求

EtherNet/IP 接口模块。
为 DH+ 配置具有相应信道的 1756-DHRIO 接口模块。
DH+ 网络上的 SLC500 或 PLC-5 系列 PLC。

● 注意：DH+ 网关模型不支持自动生成标记数据库。

DH+ 网关设备 ID

设备 ID 指定设备 IP 地址，以及进行连接所需的 DH+ 参数。设备 ID 指定如下：

<IP 或主机名>,1,<可选路由路径>,<DHRIO 插槽>.<DHRIO 信道>.<DH+ 节点 ID (十进制)>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	IP 地址或主机名。	十进制	0-255
1	端口 ID	移植到底面。	十进制	1
可选的路由路径	多个链接、端口对	指定一种除 EtherNet/IP 接口模块之外的途径，且必须等于 1 (移植到底面)。	十进制	*
DHRIO 插槽	链路地址	1756-DHRIO 接口模块的插槽编号。	十进制	0-255
DHRIO 信道		要使用的 DH+ 信道。	Alpha	A 和 B
DH+ 节点 ID		目标 PLC 的 DH+ 节点 ID (十进制格式)。**	十进制	0-99

● *有关详细信息，请参阅[连接路径规范](#)。

● **有关详细信息，请参阅下方的“节点 ID 八进制寻址”。

示例

123.123.123.123,1,2.A.3

这相当于 Ethernet/IP 123.123.123.123。DH+ 卡插在插槽 2 中：使用 DH+ 信道 A 和寻址目标 DH+ 节点 ID 3 (十进制)。

节点 ID 八进制寻址

DH+ 节点 ID 在 PLC 中以八进制格式指定，并且需要转换为十进制格式，以便用于 DH+ 网关设备 ID。可以在 RSLinx 内的 RSWho 中找到节点 ID。它以八进制格式显示。

示例

RSWho 中的 DH+ 节点 10 (八进制) = DH+ 网关设备 ID 中的 DH+ 节点 8 (十进制)。

请务必确定与正确控制器进行通信。在以上示例中，如果将 10 作为 DH+ 节点 ID 输入到 DH+ 网关设备 ID 中，则将通过节点 12 (八进制的 12 相当于十进制的 10) 而不是节点 10 (八进制) 进行通信。如果节点 12 (八进制) 不存在，则 DHRIO 模块将返回 DF1 STS 0x02。这意味着链路层不能保证数据包的传送。简单来说，就是在 DH+ 网络上找不到 DH+ 节点。

- 有关利用远程 DH+ 节点的路由路径来补充设备 ID 的信息，请参阅[路由通信](#)。
- 有关通信参数的详细信息，请参阅[ENI DF1/DH+/ControlNet 网关通信参数](#)。

ControlNet™ 网关设置

ControlNet 网关提供了一种通过 Allen-Bradley ControlLogix Ethernet 驱动程序与 ControlNet 上的 PLC-5C 系列 PLC 进行通信的方法。

要求

EtherNet/IP 接口模块。
1756-CNB 或 1756-CNBR 接口模块。
ControlNet 网络上的 PLC-5C 系列 PLC。

● 注意：ControlNet 网关模型不支持自动生成标记数据库。

ControlNet 网关设备 ID

设备 ID 用于指定设备 IP 地址，以及进行连接所需的 ControlNet 参数。设备 ID 指定如下：

<IP 或主机名>,1,[<可选路由路径>],<CNB 插槽>.<CNB 通道>.<ControlNet 节点 ID (十进制)>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	IP 地址或主机名。	十进制	0-255
1	端口 ID	底面端口。	十进制	1
可选的路由路径	多个链接、端口对	在 EtherNet/IP 通信模块之外指定一个通路，且必须等于 1 (背板端口)。	十进制	*
CNB 插槽	链路地址	1756-CNB/CNBR 接口模块的插槽编号。	十进制	0-255
CNB 通道	端口 ID	要使用的 ControlNet 通道。	Alpha	A 和 B
ControlNet 节点 ID	链路地址	目标 PLC 的 ControlNet 节点 ID (十进制格式)。**	十进制	0-99

- *有关详细信息，请参阅[连接路径规范](#)。
- **有关详细信息，请参阅下方的“节点 ID 八进制寻址”。

示例

123.123.123.123,1,2.A.3

这相当于 Ethernet/IP 123.123.123.123。ControlNet 卡插在插槽 2 中：使用 ControlNet 通道 A 和寻址目标 ControlNet 节点 ID 3。

节点 ID 八进制寻址

ControlNet 节点 ID 在 PLC 中以八进制格式指定，并且需要转换为十进制格式，以便用于 ControlNet 网关设备 ID。可以在 RSLinx 内的 RSWho 中找到节点 ID。它以八进制格式显示。

示例

RSWho 中的 CN 节点 10 (八进制) = ControlNet 网关设备 ID 中的 CN 节点 8 (十进制)。

请务必确定与正确控制器进行通信。在以上示例中，如果将 10 作为 ControlNet 节点 ID 输入到 ControlNet 网关设备 ID 中，则将通过节点 12 (八进制的 12 相当于十进制的 10) 而不是节点 10 (八进制) 进行通信。如果节点 12 (八进制) 不存在，则 CNB 模块将返回 DF1 STS 0x02。这意味着链路层不能保证数据包的传送。简单来说，就是在 ControlNet 网络上找不到 ControlNet 节点。

注意：

- 1. 有关利用远程 ControlNet 节点的路由路径来补充设备 ID 的详细信息，请参阅[通信路由](#)。
- 2. 有关通信参数的详细信息，请参阅 [ENI DF1/DH+/ControlNet 网关通信参数](#)。

EtherNet/IP 网关设置

EtherNet/IP 网关通过 Allen-Bradley ControlLogix Ethernet 驱动程序 为与 Ethernet/IP 上的 MicroLogix、SLC 500 和 PLC-5 系列 PLC 建立通信提供了一种渠道。

要求

两个或多个 Ethernet/IP 接口模块 (如 1756-ENBT)。
MicroLogix、SLC500 或 PLC 5 系列 PLC，带 Ethernet/IP 连接性。

注意：EthernetIP 网关模型不支持自动生成标记数据库。

EtherNet/IP 网关设备 ID

设备 ID 指定本地设备 IP 地址，以及进行连接所需的远程 Ethernet/IP 地址。设备 ID 指定如下：

<IP 或主机名>,1,<可选路由路径>,<ENBT 插槽>.<ENBT 信道>.<远程 IP>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	本地 Ethernet/IP 接口模块的 IP 地址或主机名。	十进制	0-255
1	端口 ID	底面端口。	十进制	1
可选的路由路径	多个链接、端口对	在 Ethernet/IP 接口模块之外指定一个通路，且必须等于 1 (底面端口)。	十进制	*
ENBT 槽	链路地址	第二个 Ethernet/IP 接口模块的插槽编号。	十进制	0-255
ENBT 信道	端口 ID	要使用的 Ethernet/IP 端口。	Alpha	A 和 B
远程 IP 地址	链路地址	目标 PLC 远程 IP 地址。	十进制	0-255

*有关详细信息，请参阅[连接路径规范](#)。

示例

123.123.123.123,1,2.A.192.168.1.10

这相当于本地 IP 为 123.123.123.123。第二个 Ethernet/IP 卡位于插槽 2: 使用端口 A 和 IP 为 192.168.1.10 的寻址目标设备。

● **注意:**

1. 有关利用远程 EtherNet/IP 设备的路由路径来补充设备 ID 的信息, 请参阅[通信路由](#)。
2. 有关通信参数的详细信息, 请参阅 [ENI DF1/DH+/ControlNet 网关通信参数](#)。
3. 在配置设备 ID 时, 用户应验证是否可以使用通过 RSLinx 的相同路由对设备进行检测。

串行网关设置

串行网关提供了一种在串行网络上使用 Allen-Bradley ControlLogix Ethernet 驱动程序与 ControlLogix、CompactLogix、FlexLogix 和 SoftLogix PLC 进行通信的方式。

要求

EtherNet/IP 接口模块。
带串行端口的本地 CPU。
带串行端口的远程 ControlLogix、CompactLogix、FlexLogix 或 SoftLogix CPU。

● **注意:**

1. 本地和远程 CPU 必须在同一串行网络上。
2. 串行网关模型不支持自动标记数据库的生成。

串行网关设备 ID

设备 ID 指定本地设备 IP 地址, 以及进行连接所需的远程设备站 ID。设备 ID 指定如下:

<IP 或主机名>,1,<可选路由路径>,<CPU 插槽>.<串行端口信道>.<站 ID (十进制)>

指示符	指示符类型*	说明	格式	范围
IP/主机名	不适用	IP 地址或主机名。	十进制	0-255
1	端口 ID	底面端口。	十进制	1
可选的路由路径	多个链接、端口对	在 EtherNet/IP 接口模块之外指定一个通路, 且必须等于 1 (底面端口)。	十进制	*
CPU 插槽	链路地址	CPU 模块的插槽编号包括用于通信的串行端口。	十进制	0-255
串行端口信道		要使用的串行端口信道。	Alpha	A 和 B
站 ID		目标 PLC 的 DF1 站 ID (十进制格式)**	十进制	0-255

● *有关详细信息, 请参阅[连接路径规范](#)。

示例

123.123.123.123,1,0.A.3

这相当于 123.123.123.123 Ethernet/IP。CPU 卡插在插槽 0 中: 使用信道 A (串行端口) 和寻址目标站 ID 3 (十进制)。

● **注意:**

1. 有关利用远程串行节点的路由路径来补充设备 ID 的信息, 请参阅[通信路由](#)。
2. 有关通信参数的详细信息, 请参阅 [Logix 通信参数](#)。

3. 在配置设备 ID 时，用户应验证是否可以使用通过 RSLinx 的相同路由对设备进行检测。

MicroLogix 1100 设置

MicroLogix 1100 设备 ID

设备 ID 用于指定 MicroLogix 1100 的 IP 地址。设备 ID 指定如下：

<IP 或主机名>

指示符	指示符类型	说明	格式	范围
IP/主机名	不适用	IP 地址或主机名。	十进制	0-255

示例
123.123.123.123

这相当于 IP 为 123.123.123.123。

有关通信参数的详细信息，请参阅 [ENI DF1/DH+/ControlNet 网关通信参数](#)。

通信路由

路由可用于与多种网络上的远程设备进行通信。可将其视为本地设备和远程设备之间的网桥，即使其位于两个不同的现场总线网络上亦如此。访问远程 (目标) 背板能够与位于此背板的受支持模块进行直接通信。支持的模块包括：

- ControlLogix 应用程序的 ControlLogix 5500 处理器。
- SoftLogix 应用程序的 SoftLogix 5800 处理器。
- DH+ 网关应用程序的 1756-DHRIO 接口模块。
- ControlNet 网关应用程序的 1756-CNB 或 1756-CNBR 接口模块。

路由路径为一系列背板跃点，其最后一个跃点指向目标背板。每个跃点均需要一个 Logix 背板 (而不是 Logix 处理器)。单个跃点可以利用以下网络之一作为其媒介：

- ControlNet
- DH+
- TCP/IP (Ethernet/IP)

重要事项：ENI 和 MicroLogix 1100 型号不支持路由。

连接路径规范

路由路径在设备 ID 中进行指定。与非路由应用程序一样，通信由 PC 上的 Allen-Bradley ControlLogix Ethernet 驱动程序 发起，然后在本地以太网模块中进行定向。与非路由应用程序一样，到达该本地以太网模块后，设备 ID 便会指定一条从该模块发出并到背板上的路径。路由路径会将消息定向到所需的 Logix 背板。设备 ID 还可用于确定与之通信的设备 (例如 ControlLogix 处理器、SoftLogix 处理器、DH+ 节点或 ControlNet 节点)。

路由路径规范分别以左括号和右括号开始和结束 ([])。路径本身为一系列端口/链接地址对，与“RSLogix 5000 消息配置”对话框中的通信路径语法相同。

指示符类型	说明	格式	范围
端口 ID	用于指定从所讨论接口模块发出的路径。*	十进制	0-65535
链接地址	如果相应端口为背板，则链接地址为输出接口模块的插槽编号。 如果相应端口为接口模块端口，则链接地址指定如下所示目标节点。 - DH+/ControlNet: 节点 ID - EtherNet/IP 通信模块: IP 地址 - SoftLogix EtherNet/IP 模块: IP 地址	十进制	0-255

*有关详细信息，请参阅以下“端口引用”部分。

单跃点

IP 地址, 端口 ID0, [链接地址0, 端口 ID1, 链接地址1, 端口 ID2], 链接地址2。

多跃点 (N 个跃点)

IP 地址, 端口 ID0, [链接地址0, 端口 ID1, 链接地址1, 端口 ID2, 链接地址2, ... 端口 ID(N+1), 链接地址(N+1), 端口 ID(N+2)], 链接地址(N+2)。

注意:

1. 路径中的最后一个端口 ID (单跃点和多跃点分别为端口 ID2 和端口 ID(N+2)) 必须为 1 (背板端口)。
2. 端口 ID0 必须为 1 (背板端口)。链接地址2 和链接地址 (N+2) 为远程 Logix 处理器/1756-DHRIO 模块 /1756-CNB 模块的插槽编号。

端口引用

接口模块	端口 1	端口 2	端口 3
EtherNet/IP 通信模块	背板	以太网	不适用
SoftLogix EtherNet/IP 报文通信模块	虚拟背板	以太网	不适用
1756-DHRIO	背板	DH+ 网络 - 位于通道A	DH+ 网络 - 位于通道B
1756-CNB	背板	ControlNet 网络	不适用

应用程序说明

1. 消息不得通过该路径中的同一接口模块通道进行多次路由。否则会导致 CIP 错误 0x01 Ext. 错误 0x100B。
2. 对于多通道接口模块，无论消息是先定向到背板还是完全避开背板，均无法将消息路由到某个模块并随后立即路由出同一模块 (使用不同的通道)。如前所述，后者不受支持，因为每个跃点均需要一个 ControlLogix 背板。例如，通过一个 1756-DHRIO 接口模块将 DH+ 消息从一个 DH+ 链路 (例如 1756-DHRIO 的通道 A) 路由到另一个 DH+ 链路 (例如同一个 1756-DHRIO 的通道 B)。这通常称为“远程 DH+ 报文通信”且不受支持。

路由示例

下列路由示例中包括排除本地 1756-ENBT 的 IP 的完整设备 ID。设备 ID/路由路径针对本地 1756-ENBT 模块。跃点说明采用以下形式：

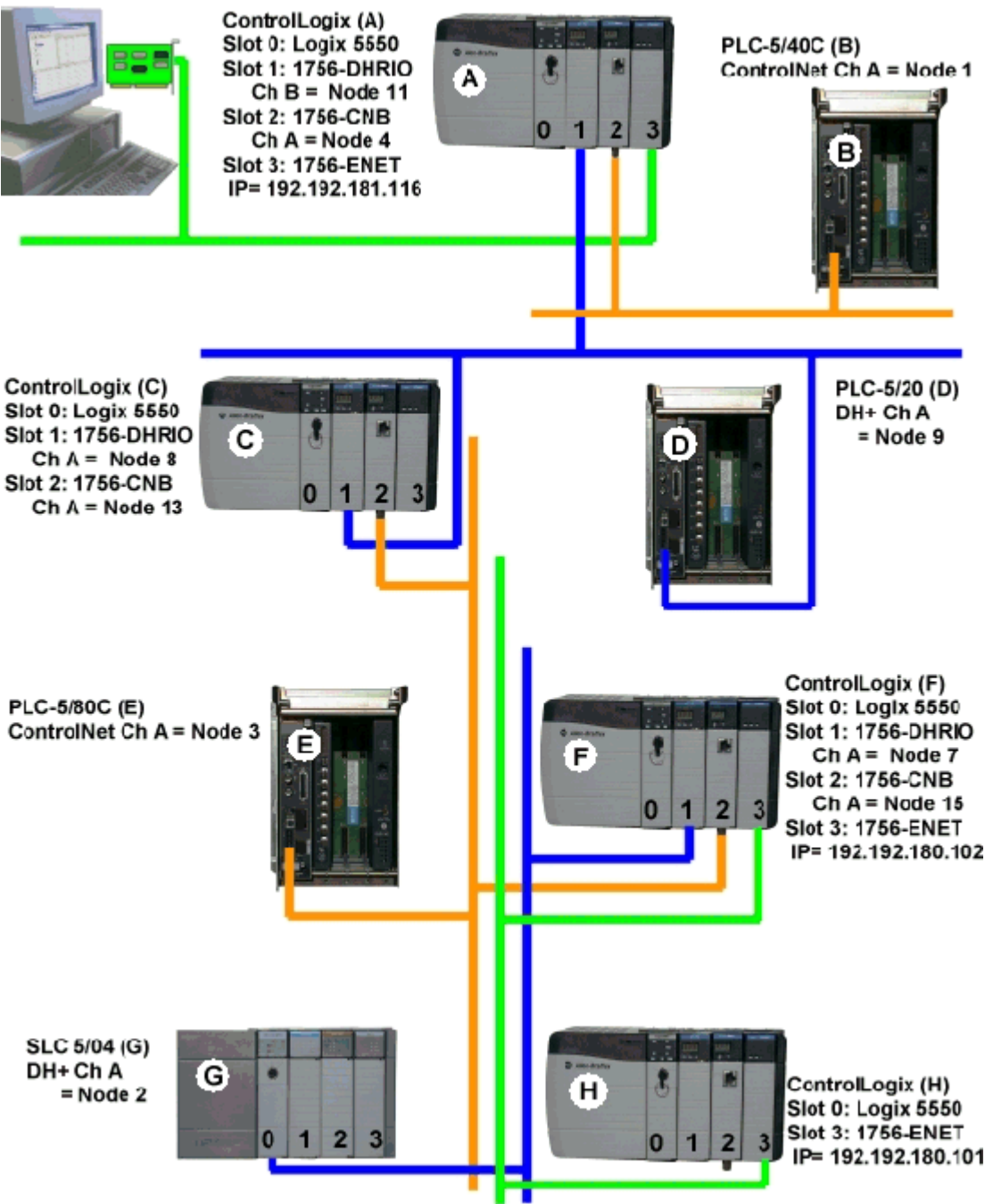
链接地址 (N), 端口 ID (N+1), 链接地址 (N+1), 端口 ID (N+2)

有关详细信息，请参阅[连接路径规范](#)。有关构建连接/路由路径的详细信息，请参阅 Allen-Bradley 出版物 1756-6.5.14, 第 4-5 页到第 4-8 页。

在下图中，所有 DH+/ControlNet 节点 ID 都被指定为十进制格式。在 PLC 中指定并在 RSWho 中显示的节点 ID 为八进制格式。颜色说明如下：

- 绿色 = 以太网
- 蓝色 = DH+
- 橙色 = ControlNet

有关详细信息，请参阅 [Data Highway Plus 网关设置](#) 和 [ControlNet 网关设置](#)。



示例 1
Logix5550 到 PLC-5 (通过 DH+ 网关)。

目标节点	型号	路由	设备 ID 短 IP
PLC-5/20 (D)	DH+ 网关	否	1,1.B.9

示例 2
Logix5550 到 PLC-5C (通过 CN 网关)。

目标节点	型号	路由	设备 ID 短 IP
PLC-5/40C (B)	CN 网关	否	1,2.A.1

示例 3

Logix5550 到 Logix5550 (通过 DH+ 路由)。

目标节点	型号	路由	设备 ID 短 IP
Logix5550 (C)	ControlLogix 5550	是	1,[1,2,8,1],0

示例 3 的路由路径细目。

跃点	段	说明
1	1,2,8,1	插槽 1 (DHRIO) -> 端口 2 (DH+ Ch A) -> DH+ 节点 8 -> Logix C 背板

示例 4

Logix5550 到 PLC-5C (通过 CN 网关), 通过 DH+ 路由。

目标节点	型号	路由	设备 ID 短 IP
PLC-5/80C (E)	CN 网关	是	1,[1,2,8,1],2.A.3

示例 4 的路由路径细目。

跃点	段	说明
1	1,2,8,1	插槽 1 (DHRIO) -> 端口 2 (DH+ Ch A) -> DH+ 节点 8 -> Logix C 背板

示例 5

Logix5550 到 Logix5550 (通过 DH+、ControlNet 路由)

目标节点	型号	路由	设备 ID 短 IP
Logix5550 (F)	ControlLogix 5550	是	1,[1,2,8,1,2,2,15,1],0

示例 5 的路由路径细目。

跃点	段	说明
1	1,2,8,1	插槽 1 (DHRIO) -> 端口 2 (DH+ Ch A) -> DH+ 节点 8 -> Logix C 背板
2	2,2,15,1	插槽 2 (CNB) -> 端口 2 (CN Ch A) -> CN 节点 15 -> Logix F 底面

示例 6

Logix5550 到 SLC 5/04 (通过 DH+、ControlNet 路由)。

目标节点	型号	路由	设备 ID 短 IP
SLC 5/04 (G)	DH+ 网关	是	1,[1,2,8,1,2,2,15,1],1.A.2

示例 6 的路由路径细目。

跃点	段	说明
1	1,2,8,1	插槽 1 (DHRIO) -> 端口 2 (DH+ Ch A) -> DH+ 节点 8 -> Logix C 背板
2	2,2,15,1	插槽 2 (CNB) -> 端口 2 (CN Ch A) -> CN 节点 15 -> Logix F 底面

示例 7

Logix5550 到 Logix5550 (通过 DH+、ControlNet、以太网路由)。

目标节点	型号	路由	设备 ID 短 IP
Logix5550 (H)	ControlLogix 5550	是	1,[1,2,8,1,2,2,15,1,3,2,192.192.180.101,1],0

示例 7 的路由路径细目。

跃点	段	说明
1	1,2,8,1	插槽 1 (DHRIO) -> 端口 2 (DH+ Ch A) -> DH+ 节点 8 -> Logix C 背板
2	2,2,15,1	插槽 2 (CNB) -> 端口 2 (CN Ch A) -> CN 节点 15 -> Logix F 底面
3	3,2,192.192.180.101,1	插槽 3 (ENBT) -> 端口 2 -> 远程 1756-ENBT IP -> Logix H 背板

选择协议模式

符号模式

在“符号模式”下，每个客户端/服务器标记地址在数据包中由其 ASCII 字符名称表示。

优势	劣势
<ol style="list-style-type: none"> 1. 发送数据请求时所需的全部信息都位于客户端/服务器标记的地址中。 2. 仅会从 PLC 请求客户端/服务器标记中正被访问的数据。 3. 向后兼容。 	<ol style="list-style-type: none"> 1. 处理符号地址时的设备周转时间较长。 2. 由于各个请求的大小不同，每个多请求数据包的请求较少。

● 注意：

1. 为有效利用多请求数据包优化，应尽可能在单个数据包中使用更多的表示标记。由于数据包中的标记地址由其 ASCII 字符名称表示，因此标记地址应尽可能缩短。例如，优先选用“MyTag”，而非“MyVeryLongTagNameThatContains36Chars”。
2. 当默认数据类型属性设置为“默认”时，自动生成标记会创建数据类型与控制器中类型相匹配的标记。

逻辑模式

“逻辑非阻塞”和“逻辑阻塞”封装了两种读取协议。驱动程序会根据控制器版本自动确定应使用的协议。下表汇总了它们所映射的模式和协议。

协议模式	使用的读取协议		使用的写入协议
	FRN V21 及更高版本	FRN V20 及更低版本	所有 FRN
符号	符号 (非阻塞)	符号 (非阻塞)	符号
逻辑非阻塞	符号实例非阻塞	物理非阻塞*	符号实例
逻辑阻塞	符号实例阻塞	物理阻塞*	符号实例

*V21 中已弃用。

执行逻辑读取时所需的信息由驱动程序按照控制器项目上载序列自动检索。出于简洁考虑，使用术语“逻辑地址”来表示“符号实例 ID”或“物理地址”，具体取决于所用的协议。“逻辑模式”避免了每个符号请求所需的比较耗时的地址解析和查找。

● 注意：这些逻辑模式不适用于串行网关模型。

逻辑非阻塞模式

“逻辑非阻塞模式”会以固定大小分别请求所有客户端/服务器标记。

优势	劣势
<ol style="list-style-type: none"> 1. 由于每个请求均为固定大小，每个多请求数据包中可包含的请求最多。 2. 由于数据包中的客户端/服务器标记均指定了逻辑地址，设备周转时间较短。 3. 仅会从 PLC 请求客户端/服务器标记中正被访问的数据。 	上载项目时需要初始化开销以确定逻辑地址。

● 注意：此模式首选用于只有少数结构标记成员被客户端/服务器引用的情形。

逻辑阻塞模式

“逻辑阻塞”会检索单个请求中可能仅由一个客户端/服务器标记启动的 Logix 标记的所有数据。收到数据块后，其会置于驱动程序的缓存中，然后加盖时间戳。属于给定 Logix 标记的连续客户端/服务器标记随后从此缓存中获取其数据。更新所有标记后，如果缓存不过期，则会启动新请求。如果当前时间 > 缓存时间戳 + 标记扫描速率，则缓存过期。如果这种情况成立，则向设备发出另一个块请求，刷新缓存，并重复该循环。

优势	劣势
<ol style="list-style-type: none">1. 每次读取时都会检索内容。2. 由于数据包中的客户端/服务器标记均指定了逻辑地址，设备周转时间较短。3. 由于每个请求均为固定大小，每个多请求数据包中可包含的请求最多。	<ol style="list-style-type: none">1. 上载项目时需要初始化开销以确定逻辑地址。2. 如果引用的 Logix 标记较少，则会比“逻辑非阻塞模式”慢 (因为 PLC 中正被访问的数据要多于客户端/服务器标记中所引用的数据)。

● **注意:** 此模式首选用于大多数结构标记成员被客户端/服务器引用的情形。

● **另请参阅:** [性能统计信息和调优](#)

符号实例与物理协议

“符号实例”读取为 CIP 请求，其中“CIP 实例 ID”用于指定读取请求中的原生标记。在“非阻塞模式”下，可能需要“CIP 成员 ID”来完全限定指向结构成员和数组元素的路径。例如，“CIP 实例 ID”表示结构，而“CIP 成员 ID”则表示结构中的成员。由于添加了完全限定客户端/服务器标记所需的“CIP 成员 ID”，请求的大小可能各有不同。结构有嵌套越深，指定标记时所需的“CIP 成员 ID”就越多，单个数据包中可包含的请求数就越少。FRN V21 中引入了符号实例读取。

物理读取为 CIP 请求，其中“DMA 地址”用于指定读取请求中的原生标记。在“非阻塞模式”下，可能需要字节偏移来完全限定指向结构成员和数组元素的路径。例如，DMA 起始地址表示结构，而字节偏移则表示结构中的成员。最终，起始 + 偏移即为请求中所指定的 DMA 地址；所有请求均为固定大小 (与符号实例读取不同)。无论结构的嵌套深度如何，数据包中的请求均大小相同。物理读取自 FRN V21 起弃用。

检测“控制器项目”中所做的更改

Allen-Bradley ControlLogix Ethernet 驱动程序可以监视项目更改，并可检测正在进行的下载、在线和离线编辑等。当协议设置为“逻辑”时，用户可以选择将驱动程序项目图像与控制器项目图像同步。同步可确保驱动程序在对每个原生标记读取和写入时均使用当前的逻辑地址。

- **“正在进行的下载”:** 驱动程序会监视每个请求中的在线和离线编辑。它在主动读取或写入原生标记时会检测是否发生了下载，然后根据其模式执行项目更改程序。要启用此同步，请右键单击设备并选择“属性”。在“Logix 选项”组中，找到“在线编辑后同步”或“离线编辑后同步”，然后选择“是”。
- **“在线编辑后同步”:** 驱动程序会监视每个请求中的在线编辑。它在以下读取或写入操作中会检测控制器是否发生了在线编辑，然后根据其模式执行项目更改程序。要启用此同步，请右键单击设备并选择“属性...”。在“Logix 选项”组中，找到“在线编辑后同步”，然后选择“是”。
- **“离线编辑后同步”:** 驱动程序会监视每个请求中的离线编辑。它在以下读取或写入操作中会检测控制器是否发生了离线编辑，然后根据其模式执行项目更改程序。要启用此同步，请右键单击设备并选择“属性”。在“Logix 选项”组中，找到“离线编辑后同步”，然后选择“是”。

项目更改程序 (符号模式)

1. 检测到项目更改。
2. 向“事件日志”发布一条消息，表明检测到更改。
3. 在项目更改期间，发生下载的情景如下：
 - 所有进行的读取和写入操作都会停止并失败。
 - 对控制器每 2 秒轮询一次以监视项目更改完成情况。

- 再未检测到项目更改。
 - 向“事件日志”发布一条消息，表明再未检测到更改。
4. 在项目更改期间，在线和离线编辑的情景如下：
 - 响应数据被忽略。
 - 所有进行的读取和写入操作被重试。
 5. 读取和写入操作恢复使用“符号模式”。

项目更改程序 (逻辑模式)

1. 检测到项目更改。
2. 向“事件日志”发布一条消息，表明检测到更改。
3. 在项目更改期间，发生下载的情景如下：
 - 所有进行的读取和写入操作都会停止并失败。
 - 对控制器每 2 秒轮询一次以监视项目更改完成情况。
 - 再未检测到项目更改。
 - 向“事件日志”发布一条消息，表明再未检测到更改。
4. 在项目更改期间，在线和离线编辑的情景如下：
 - 响应数据被忽略。
 - 所有进行的读取和写入操作被重试。
5. 读取和写入操作恢复使用“符号模式”。
6. 如果启用了“与控制器同步”选项：
 - 在“符号模式”下 30 秒后，驱动程序会从控制器上载 (同步) 项目。
 - 读取和写入操作恢复使用“逻辑模式”并采用新的逻辑地址。
7. 如果禁用了“与控制器同步”选项，则读取和写入操作恢复使用“逻辑模式”但采用旧的逻辑地址。

SoftLogix 5800 连接注意事项

为确保正常运行，不得在 SoftLogix PC 的 RSLinx 中安装基于以太网的驱动程序 (如以太网设备、通过网关连接的远程设备等等)。安装一个或多个基于以太网的驱动程序后，请求会返回 CIP error 0x5、Ext. error 0x1 和 CIP error 0x8。

连接到 OPC 服务器所在 PC 上的 SoftLogix Soft PLC

要将 Allen-Bradley ControlLogix Ethernet 驱动程序连接至与服务器运行在同一台 PC 上的 SoftLogix Soft PLC，请按照以下说明进行操作。

1. 确保当前在 PC 上的 RSLinx 中没有运行任何基于以太网的驱动程序。
2. 验证 **EtherNet/IP 消息模块** 是否已安装在 SoftLogix 虚拟机箱中。
3. 在“设备属性”|“常规”组中，找到设备 ID 值。该值不得为“127.0.0.1, 1, <PLC_CPU_slot>”。“设备 ID”应设置为“<specific_IP_address_of_PC>, 1, <PLC_CPU_slot>”。

例如，如果 PC 的 IP 地址为 192.168.3.4 且 SoftLogix CPU 在虚拟机箱的插槽 2 中，则正确的设备 ID 为“192.168.3.4, 1, 2”。

索引

0

0000-类属模块 21

0x0001 扩展错误代码 96

0x001F 扩展错误代码 97

0x00FF 扩展错误代码 97

0x01 95

0x02 95

0x03 95

0x04 95

0x05 95

0x06 95

0x07 95

0x08 95

0x09 95

0x0A 95

0x0B 95

0x0C 95

0x0D 95

0x0E 95

0x0F 95

0x10 95

0x11 95

0x12 95

0x13 95

0x14 95

0x15 95

0x1A 95

0x1B 95

0x1C 95

0x1D 95

0x1E 96

0x1F 96

0x22 96

0x25 96

0x26 96

0x27 96

1

1761-NET-ENI 102

A

ASCII 文件 80

B

BCD 38

BCD 文件 81

C

CIP 错误代码; 错误代码 95

ControlNet 网关 104

ControlNet 网关设备 ID 104

D

DataHighwayPlus (TM) 网关设置 103

DH+ 的 SLC 500 模块化 I/O 寻址 43

DH+ 网关设备 ID 103, 105

E

ENI DF1/DH+/ControlNet 网关通信参数 19

ENI 的 CompactLogix 5300 寻址 39

ENI 的 ControlLogix 5500 寻址 39

ENI 的 FlexLogix 5400 寻址 40

ENI 的 MicroLogix 寻址 40

ENI 的 SLC 500 固定 I/O 寻址 42

ENI 的 SLC 500 模块化 I/O 寻址 43

ENI 设备 ID 103

EtherNet/IP 网关的 MicroLogix 寻址 40

EtherNet/IP 网关的 SLC 500 固定 I/O 寻址 42

EtherNet/IP 网关的 SLC 500 模块化 I/O 寻址 43

EtherNet/IP 网关设置 105

I

I/O 模块状况文件 (IOS) 90

ID 12

L

LBCD 38

LEN 处的终止字符串数据 17

Logix 高级寻址 52

Logix 设备 ID; ControlLogix 5500 以太网 101

Logix 数组数据的排序 51

Logix 寻址 39

Logix 寻址; ControlLogix 5000 寻址; Logix 基于标记的寻址 45

M

MicroLogix 1100 设备 ID 107

MicroLogix 1100 设置 107

MicroLogix PID 文件 82

MicroLogix 消息文件 84

MicroLogix 寻址 40

P

PID 文件 83

PLC-5 系列寻址 44

S

SLC 500 插槽配置 20

SLC 500 固定 I/O 寻址 42

SLC 500 模块化 I/O 选择指南 22

SLC 500 模块化 I/O 寻址 43

SoftLogix 5800 寻址 40

SoftLogix Soft PLC 连接注意事项 113

SoftLogix 数据库设置; ControlLogix 数据库设置; Logix 数据库设置 17

SoftLogix 通信参数; Logix 通信参数; ControlLogix 通信参数 16

SoftLogix 选项; ControlLogix 选项; Logix 选项 16

T

TCP/IP 端口 16, 19

▼

帮助内容 5

枳

标记层次 90
标记导入文件 18
标记生成 14
标识 12

尪

不扫描, 仅按需求轮询 13

隹

布尔型 37

摭

操作模式 12

捲

插槽 21

败

超时前的尝试次数 14

譚

程序标记 47

串行网关的 CompactLogix 5300 寻址 39
串行网关的 ControlLogix 5500 寻址 39
串行网关的 FlexLogix 5400 寻址 40
串行网关的 SoftLogix 5800 寻址 40
串行网关设备 ID 106
串行网关设置 106

冻

创建 16

罌

从导入文件创建 18

从设备创建 18

钹

错误代码 95

嗽

地址说明 38

穿

端口 ID 107

瞍

短整型 37

宙

对于重复标记 15

𠂇

二进制文件 76

雾

非活动情况监视器 16

审

封装错误代码 95

泊

浮点型 38

浮点型;浮点型文件 79

燭

父组 15

闌

附录 99

襪

覆盖 15

栖

概述 6

駉

高级寻址\: BOOL 52

高级寻址\: DINT 57

高级寻址\: INT 55

高级寻址\: LINT 59

高级寻址\: LREAL 67

高级寻址\: REAL 60

高级寻址\: SINT 53

高级寻址\: UDINT 65

高级寻址\: UINT 63

高级寻址\: ULINT 66

高级寻址\: USINT 62

高速计数器文件 (HSC) 87

别

功能文件 87

擎

故障时降级 14

譚

计时器文件 76

计数器文件 77

桎

检测“控制器项目”中所做的更改 112

闭

降级超时 14

降级期间 14

降级时放弃请求 14

繳

结构标记寻址 47

结构标记寻址; 标记范围 47

捺

控制器到服务器的名称转换 92

控制文件 78

困

块传输文件 86

揭

来自缓存的初始更新 13

祛

离线编辑后同步 17

轻

连接超时 14

连接大小 16

连接路径规范 107

钞

链接地址 107

赳

路由示例 108

槁

模块 21

模拟 13

齰

默认数据类型条件 38

價

内部标记 48

凭

前导下划线 92

廚

强加数组限制 19

講

请求超时 14
请求大小 19
请求间延迟 14

駢

驱动程序 12

儉

全局标记 47

敦

日期 38

儷

冗余 22

戔

扫描模式 13

笏

筛选 19

淨

删除 15

譚

设备启动时 15
设备属性 - 标记生成 14
设置 6

璿

生成 15

嬾

实时时钟文件 (RTC) 88

丫

事件日志消息 98

輳

输出文件 69

输出字 21

输入文件 72

输入字 21

对

属性更改时 15

穰

数据库导入方法 18

数据类型说明 37

数据收集 12

数组标记 91

数组标记;地址格式 46

数组计数限制 19

数组块大小 16

印

双精度 38

双字型 38

辺

通道分配 12

通信超时 13-14

通信路由 107

通信协议 8

繼

统计信息 27

统计信息类型 27

擲

文件列表 68

旄

显示说明 18

闰

限制名称长度 18

頊

项目选项 17

洨

消息文件 85

勳

协议 17

协议模式 17

侁

信道 0 通信状况文件 89

信道 1 通信状况文件 89

托

型号 12

忒

性能调优示例 27

性能统计信息 17

性能统计信息和调优 27

性能优化 24

辮

选择协议模式 111

宛

寻址结构数据类型 50

寻址字符串数据类型 50

丿

以太网的 CompactLogix 5300 寻址 39

以太网的 ControlLogix 5500 寻址 39

以太网的 FlexLogix 5400 寻址 40

璩

用于 ControlNet 的 PLC-5 系列寻址 44

用于 EtherNet/IP 网关的 PLC-5 系列寻址 44

躄

优化通信 24

优化应用程序 26

頤

预定义术语标记 48

匿

原子型数据类型寻址 48

僊

允许功能文件块写入 19

允许子组 16

嚙

在线编辑后同步 17

锷

长整型 38

长整型文件 81

攔

整型文件 78

撤

支持的设备 6

牖

状况文件 75

膊

自动标记数据库生成;长控制器程序 & 标记名称 90

自动降级 14

自动生成标记数据库前准备 93

媾

字 37

字符 37

字符串 38

字符串文件 80

字节 37

逕

遵循标签指定的扫描速率 13