

SIEMENS

SIMATIC HMI

WinCC V7.5 SP2 WinCC: 组态和通信

系统手册




多用户系统	1
文件服务器	2
WinCC 服务模式	3
冗余系统	4
过程通讯	5
通信通道	6
通讯 - 诊断	7
OPC - 开放式互连	8
REST 接口	9
WinCC/Cloud Connector	10

在线帮助的打印

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

 危险
表示如果不采取相应的小心措施， 将会 导致死亡或者严重的人身伤害。
 警告
表示如果不采取相应的小心措施， 可能 导致死亡或者严重的人身伤害。
 小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。
注意
表示如果不采取相应的小心措施，可能导致财产损失。


当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

 警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

商标

所有带有标记符号®的都是 Siemens AG 的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	多用户系统	15
1.1	WinCC 中的多用户系统	15
1.2	WinCC 中的客户机/服务器系统	16
1.3	典型组态	19
1.4	数量结构与性能	21
1.5	客户机/服务器方案	24
1.6	服务器组态	27
1.6.1	服务器组态	27
1.6.2	在服务器上创建新项目	29
1.6.3	如何在计算机列表中注册客户机	31
1.6.4	如何组态操作员授权	32
1.6.5	如何组态数据包导出	35
1.6.6	在服务器项目中组态客户机	39
1.7	客户机组态	42
1.7.1	客户机组态	42
1.7.2	在客户机上创建新项目	44
1.7.3	组态导入数据包	46
1.7.4	如何组态标准服务器	49
1.7.5	如何组态首选服务器	52
1.7.6	组态客户机的起始画面	54
1.7.7	显示来自不同服务器的画面	56
1.7.8	组态客户机上的画面变化	58
1.7.9	使用来自不同服务器的数据	60
1.7.10	显示来自不同服务器的消息	62
1.7.11	组态多个服务器消息的消息顺序报表	63
1.8	运行系统中的系统特性	65
1.8.1	运行系统中的系统特性	65
1.8.2	启动服务器	67
1.8.3	启动客户机	68
1.8.4	使用具有多个网卡的服务器时的特殊通讯特性	69
1.8.5	关闭服务器	71
1.8.6	关闭客户机	71
1.9	远程组态	73
1.9.1	远程组态	73
1.9.2	加密通信	74
1.9.3	如何访问子网外的计算机	76

1.9.4	从多个客户机访问项目	80
1.9.5	如何打开一个要进行编辑的项目	82
1.9.6	如何编辑服务器项目画面	84
1.9.7	如何激活项目	85
1.9.8	如何取消激活项目	86
1.10	使用客户机/服务器系统中的 OPC 接口	88
2	文件服务器	91
2.1	安装文件服务器	91
3	WinCC 服务模式	93
3.1	WinCC 服务模式：标准项目和服务项目	93
3.2	服务项目的组态	95
3.3	使用服务项目及限制	96
3.4	系统托盘中的 WinCC 状态和控件	98
3.5	功能和先决条件	101
3.5.1	服务项目的操作模式	101
3.5.2	运行服务项目的要求	102
3.6	组态 WinCC 服务模式	104
3.6.1	如何将一个项目定义为服务项目	104
3.6.2	如何组态服务项目的自动启动	105
3.7	运行系统中的服务项目	107
3.7.1	如何激活服务项目	107
3.7.2	如何登录与注销已激活的服务项目	108
3.7.3	如何激活出于维护目的的交互式操作	109
4	冗余系统	111
4.1	冗余	111
4.2	WinCC 冗余	112
4.3	冗余系统的要求	115
4.4	冗余如何工作	116
4.5	组态冗余系统	120
4.5.1	冗余系统安装指南	120
4.5.2	组态相同功能	122
4.5.3	如何组态冗余服务器	123
4.5.4	如何组态用户归档的同步	125
4.5.5	如何为冗余服务器复制项目	127
4.5.6	如何在运行期间复制冗余项目	129
4.6	使用 WinCC Redundancy 的各种情况	131
4.6.1	过程连接出错时切换客户端	131

4.6.2	故障情况	135
4.6.2.1	故障情况	135
4.6.2.2	情况 1: 服务器计算机上的项目未处于运行系统中	136
4.6.2.3	情况 2: 与伙伴服务器的连接出现故障	138
4.6.2.4	情况 3: 与客户机的网络连接出现故障	140
4.6.2.5	情况 4: 过程连接出现故障	140
4.6.2.6	情况 5: 软件出错	141
4.6.3	WinCC 冗余系统变量	142
4.6.4	WinCC 冗余系统消息	146
5	过程通讯	151
5.1	通讯基本信息	151
5.2	组态连接的基本规则	152
5.3	WinCC 过程通讯	153
5.3.1	WinCC 过程通讯	153
5.3.2	WinCC 通讯的原理	153
5.3.3	在运行系统中为连接状态组态变量	155
5.3.4	外部变量	158
5.3.4.1	外部变量	158
5.3.4.2	如何创建新连接	160
5.3.4.3	外部变量的组态方法	161
5.3.4.4	按 WinCC 数据类型排序的格式改编	162
5.3.4.5	按 AS 数据类型排序的格式改编	170
5.3.4.6	二进制写入机制的原理	185
5.3.4.7	如何通过“二进制写入”组态变量	186
5.3.5	通过以太网连接的端口地址	188
5.3.6	SIMATIC S7-PLCSIM Advanced	188
6	通信通道	191
6.1	Allen Bradley - Ethernet IP	191
6.1.1	WinCC 通道“Allen Bradley - Ethernet IP”	191
6.1.2	通道单元分配	191
6.1.3	支持的数据类型	192
6.1.4	组态通道	192
6.1.4.1	组态通道“Allen Bradley - Ethernet IP”	192
6.1.4.2	如何为“Allen Bradley - Ethernet IP”通道组态连接	193
6.1.4.3	实例: 通讯路径	194
6.1.4.4	组态变量	197
6.2	Mitsubishi 以太网	208
6.2.1	WinCC 通道“Mitsubishi 以太网”	208
6.2.2	所支持的数据类型	208
6.2.3	组态通道	209
6.2.3.1	组态“Mitsubishi 以太网”通道	209
6.2.3.2	如何组态“Mitsubishi FX3U 系列”通道单元连接	210

6.2.3.3	如何组态“Mitsubishi Q 系列”通道单元连接	212
6.2.3.4	如何组态“Mitsubishi iQ-R 系列”通道单元连接	213
6.2.3.5	如何组态“Mitsubishi iQ-F 系列”通道单元的连接	215
6.2.3.6	组态变量	216
6.3	Modbus TCP/IP	221
6.3.1	“Modbus TCP/IP”通道	221
6.3.2	支持的数据类型	221
6.3.3	组态通道	222
6.3.3.1	组态“Modbus TCP/IP”通道	222
6.3.3.2	如何组态连接	222
6.3.3.3	组态变量	224
6.4	OPC 通道	230
6.4.1	WinCC OPC 通道	230
6.4.2	OPC 条目管理器	231
6.4.3	所支持的 WinCC 数据类型概述	235
6.4.4	WinCC OPC DA 客户机	236
6.4.4.1	WinCC OPC DA 客户机的功能	236
6.4.4.2	如何通过 OPC 条目管理器访问 WinCC 变量	237
6.4.4.3	在不使用 OPC 条目管理器的情况下访问 WinCC 变量	243
6.4.4.4	在 WinCC OPC DA 客户机上使用结构	245
6.4.4.5	OPC DA 通讯受到干扰时的错误处理	248
6.4.5	WinCC OPC XML 客户机	252
6.4.5.1	WinCC OPC XML 客户机的功能	252
6.4.5.2	如何通过 OPC 条目管理器访问 WinCC 变量	254
6.4.5.3	在不使用 OPC 条目管理器的情况下访问 WinCC 变量	258
6.5	OPC UA WinCC 通道	261
6.5.1	WinCC 通道“OPC UA WinCC 通道”	261
6.5.2	所支持的数据类型概述	262
6.5.3	组态 OPC UA 通道	263
6.5.3.1	常规序列	263
6.5.3.2	在 WinCC 项目中插入 OPC UA WinCC 通道	264
6.5.3.3	创建与 OPC UA 服务器的连接	264
6.5.3.4	通过 NAT 路由器创建与 OPC UA 服务器的连接	270
6.5.3.5	设置通过证书进行认证。	272
6.5.3.6	如何组态到 OPC UA 服务器的连接	279
6.5.3.7	组态 OPC UA 变量	282
6.5.3.8	在 WinCC 中使用 OPC UA 类型	289
6.5.3.9	在 WinCC 中使用 OPC UA 报警	294
6.5.4	OPC UA WinCC 通道中的 OPC UA 数组	304
6.5.4.1	WinCC 中的数组	304
6.5.4.2	如何手动创建和组态 OPC UA 数组	307
6.5.5	错误处理	309
6.5.5.1	OPC 通讯受到干扰时的错误处理	309

6.6	PROFIBUS DP	311
6.6.1	WinCC“PROFIBUS DP”通道	311
6.6.2	WinCC 驱动程序 Profibus DP 的特性	311
6.6.3	集成“Profibus DP”驱动程序	312
6.6.4	组态“PROFIBUS DP”驱动程序	313
6.7	S5 Ethernet Layer 4	315
6.7.1	WinCC 通道“SIMATIC S5 Ethernet Layer 4”	315
6.7.2	变量的数据类型	316
6.7.3	组态通道	316
6.7.3.1	组态通道“SIMATIC S5 Ethernet Layer 4”	316
6.7.3.2	如何组态连接	317
6.7.3.3	组态变量	319
6.7.3.4	系统参数	326
6.7.4	附录	329
6.7.4.1	附录	329
6.7.4.2	内部错误代码和常量	329
6.8	S5 PROFIBUS FDL	337
6.8.1	WinCC 通道“SIMATIC S5 Profibus FDL”	337
6.8.2	受支持的数据类型和数据范围	338
6.8.3	WinCC 通道“SIMATIC S5 Profibus FDL”的功能	339
6.8.4	组态通道	341
6.8.4.1	如何组态通道“SIMATIC S5 Profibus FDL”	341
6.8.4.2	通道单元“FDL (CP5412/A2-1)”	341
6.8.4.3	如何组态连接	342
6.8.4.4	组态变量	344
6.8.4.5	系统参数	347
6.8.5	特殊功能	350
6.8.5.1	“SIMATIC S5 Profibus FDL”通道的特殊功能	350
6.8.5.2	“SIMATIC S5 Profibus FDL”通道的原始数据变量	350
6.8.5.3	组态通讯类型	351
6.8.6	“SIMATIC S5 Profibus FDL”通道的组态实例	355
6.8.6.1	“SIMATIC S5 Profibus FDL”通道的组态实例	355
6.8.6.2	如何组态 AS 中的数据处理块	355
6.8.6.3	如何组态 I/O 域	358
6.9	S5 Programmers Port AS511	360
6.9.1	WinCC 通道“SIMATIC S5 Programmers Port AS511”	360
6.9.2	变量的数据类型	360
6.9.3	组态通道	361
6.9.3.1	组态“SIMATIC S5 Programmers Port AS511”通道	361
6.9.3.2	如何组态连接	361
6.9.3.3	组态变量	363
6.10	S5 Serial 3964R	369
6.10.1	WinCC 通道“SIMATIC S5 Serial 3964R”	369

6.10.2	变量的数据类型.....	369
6.10.3	组态通道.....	370
6.10.3.1	组态“SIMATIC S5 Serial 3964R”通道.....	370
6.10.3.2	如何组态连接.....	370
6.10.3.3	组态变量.....	372
6.11	SIMATIC S7 Protocol Suite.....	377
6.11.1	WinCC 通道“SIMATIC S7 Protocol Suite”.....	377
6.11.2	通道单元选择.....	378
6.11.3	所支持的数据类型概述.....	381
6.11.4	组态通道.....	382
6.11.4.1	“SIMATIC S7 Protocol Suite”通道 - 组态.....	382
6.11.4.2	如何组态“SIMATIC S7 Protocol Suite”通道.....	383
6.11.4.3	如何离线下载 AS 符号.....	384
6.11.4.4	如何组态 AS 结构.....	389
6.11.4.5	如何导出 AS 项目数据.....	391
6.11.4.6	通道单元.....	393
6.11.4.7	组态变量.....	409
6.11.4.8	系统参数.....	415
6.11.5	特殊功能.....	420
6.11.5.1	“SIMATIC S7 Protocol Suite”通道的特殊功能.....	420
6.11.5.2	使用 S7 功能块 AR_SEND 进行数据交换.....	421
6.11.5.3	通道“SIMATIC S7 Protocol Suite”的原始数据变量.....	453
6.11.5.4	软件冗余.....	461
6.12	SIMATIC S7-1200, S7-1500 Channel.....	471
6.12.1	“SIMATIC S7-1200, S7-1500 Channel”通道.....	471
6.12.2	所支持的数据类型概述.....	472
6.12.3	组态通道.....	472
6.12.3.1	组态“SIMATIC S7-1200, S7-1500 Channel”通道.....	472
6.12.3.2	组态原始数据变量.....	475
6.12.3.3	如何组态连接.....	476
6.12.3.4	如何组态没有优先块访问的变量.....	477
6.12.3.5	如何组态具有优先块访问的变量.....	479
6.12.3.6	如何离线下载 AS 符号.....	482
6.12.3.7	如何组态 AS 结构.....	487
6.12.3.8	如何导出 AS 项目数据.....	489
6.12.3.9	冗余系统 S7-1500R/H.....	491
6.12.3.10	组态冗余控制系统.....	493
6.12.3.11	S7-1500R/H 的软件冗余.....	496
6.13	SIMATIC TI Ethernet Layer 4.....	502
6.13.1	WinCC 通道“SIMATIC TI Ethernet Layer 4”.....	502
6.13.2	变量的数据类型.....	503
6.13.3	组态通道.....	503
6.13.3.1	组态通道“SIMATIC TI Ethernet Layer 4”.....	503

6.13.3.2	如何组态连接	503
6.13.3.3	组态变量	505
6.13.3.4	系统参数	510
6.14	SIMATIC TI Serial	514
6.14.1	WinCC 通道“SIMATIC TI Serial”	514
6.14.2	变量的数据类型	514
6.14.3	组态通道	515
6.14.3.1	组态“SIMATIC TI Serial”通道	515
6.14.3.2	如何组态连接	515
6.14.3.3	组态变量	516
6.15	SIMOTION	522
6.15.1	WinCC 通道“SIMOTION”	522
6.15.2	所支持的数据类型概述	522
6.15.3	组态通道	523
6.15.3.1	“SIMOTION”通道的组态	523
6.15.3.2	如何导出 SIMOTION SCOUT 项目	523
6.15.3.3	如何通过 Simotion Mapper 创建 WinCC 项目	524
6.15.3.4	如何通过 Simotion Mapper 更改 WinCC 项目	526
6.15.3.5	如何更改连接参数	527
6.15.3.6	如何更改变量地址	528
6.15.3.7	系统参数组态	529
6.15.4	诊断“SIMOTION”通道	532
6.15.4.1	“SIMOTION”通道的诊断可能性	532
6.15.4.2	日志文件条目的描述	533
6.16	System Info	537
6.16.1	“System Info”通道	537
6.16.2	WinCC System Info 通道	537
6.16.3	所支持的系统信息概述	539
6.16.4	与其它软件组件的不同之处	544
6.16.5	组态通道	544
6.16.5.1	如何组态 System Info 通道	544
6.16.6	评价和显示系统信息的实例	545
6.16.6.1	如何调用和评价系统信息	545
6.16.6.2	如何在“System Info”通道中组态变量	545
6.16.6.3	如何在 I/O 域中显示时间	547
6.16.6.4	如何用棒图显示空闲磁盘容量	548
6.16.6.5	如何在趋势窗口中显示 CPU 负载	549
6.16.6.6	如何组态有关空闲磁盘容量的消息	550
6.16.6.7	如何显示有关可用磁盘容量的消息	552
6.16.6.8	如何在状态显示中显示打印机状态	553
6.16.6.9	如何检查 WinCC 启动参数	555
6.16.6.10	如何插入棒图	555
6.16.6.11	如何插入 I/O 域	555

6.16.6.12	如何启动运行系统	556
6.16.7	特殊功能	556
6.16.7.1	在多用户和客户系统中使用	556
6.16.7.2	监控多台服务器的系统信息的实例	557
7	通讯 - 诊断	563
7.1	通道和变量的诊断	563
7.2	有关错误检测的常规信息	564
7.3	通道诊断	565
7.3.1	通道诊断	565
7.3.2	使用性能变量检查连接	565
7.3.3	如何使用“状态 - 逻辑连接”功能来检查通道	568
7.3.4	使用“通道诊断”进行通道诊断	570
7.3.4.1	通道诊断的原理	570
7.3.4.2	使用 ActiveX 控件进行通道诊断	571
7.3.4.3	如何使用作为 ActiveX 控件的通道诊断来检查通道	571
7.3.4.4	使用“通道诊断”来诊断通道	572
7.3.4.5	如何使用通道诊断检查通道	573
7.3.4.6	如何组态通道的跟踪功能	574
7.3.4.7	如何启动运行系统	575
7.4	“System Info”通道的诊断	577
7.4.1	“System Info”通道 - 诊断选项	577
7.4.2	日志文件条目的描述	577
7.4.3	确定错误变量值的原因	578
7.4.3.1	如何确定错误变量的原因	578
7.4.3.2	如何检查通道和连接	579
7.4.3.3	如何检查变量	581
7.5	诊断通道“SIMATIC S7-1200/S7-1500”	582
7.5.1	通过系统诊断控件进行系统诊断	582
7.5.2	如何组态系统诊断	586
7.6	“SIMATIC S7 Protocol Suite”通道的诊断	588
7.6.1	“SIMATIC S7 Protocol Suite”通道 - 诊断选项	588
7.6.2	日志文件条目的描述	589
7.6.3	确定错误变量值的原因	591
7.6.3.1	如何确定错误变量的原因	591
7.6.3.2	如何检查通信处理器的组态	592
7.6.3.3	检查 SIMATIC NET 中的通讯处理器	594
7.6.3.4	如何检查连接和变量的组态	595
7.6.3.5	如何检查通道和连接	596
7.6.3.6	如何检查变量	598
7.7	“SIMATIC S5 Profibus FDL”通道的诊断	600
7.7.1	“SIMATIC S5 PROFIBUS FDL”通道的诊断选项	600

7.7.2	日志文件条目的描述	601
7.7.3	确定错误变量值的原因	601
7.7.3.1	如何确定错误变量的原因	601
7.7.3.2	如何检查通讯处理器的组态	602
7.7.3.3	检查 SIMATIC NET 中的通讯处理器	604
7.7.3.4	如何检查连接和变量的组态	605
7.7.3.5	如何检查通道和连接	606
7.7.3.6	如何检查变量	609
7.8	“OPC”通道的诊断	610
7.8.1	诊断“OPC”通道的可能性	610
7.8.2	日志文件条目的描述	610
7.8.2.1	日志文件条目的描述	610
7.8.2.2	“INFO”标记条目	612
7.8.2.3	“ERROR”标记条目	613
7.8.3	确定错误变量值的原因	614
7.8.3.1	如何确定无效变量的原因	614
7.8.3.2	如何检查组态数据	615
7.8.3.3	如何检查通道和连接	616
7.8.3.4	如何检查变量	618
7.9	变量质量	620
7.9.1	变量质量	620
7.9.2	变量的质量代码	622
7.9.3	变量状态	628
7.9.4	使用变量状态监视连接状态	629
7.9.5	使用全局动作监视变量状态	631
7.9.6	如何检查内部变量	632
8	OPC - 开放式互连	635
8.1	OPC - 开放式互连	635
8.2	OPC 的功能	636
8.3	OPC 规范及兼容性	637
8.4	在 WinCC 中使用 OPC	638
8.5	如何组态 Windows 以使用 WinCC OPC	640
8.6	WinCC OPC XML DA 服务器	641
8.6.1	WinCC OPC XML DA 服务器的功能	641
8.6.2	安装 WinCC OPC XML DA 服务器	642
8.6.3	使用 IIS 设置安全设置 (WinCC OPC XML DA 服务器)	644
8.6.4	测试安装 (WinCC OPC XML DA 服务器)	644
8.7	WinCC OPC DA 服务器	646
8.7.1	WinCC OPC DA 服务端的功能	646
8.7.2	使用多个 OPC DA 服务器	647

8.7.3	查询 OPC DA 服务器名称	648
8.7.4	OPC DA 连接的实例	649
8.7.4.1	WinCC - WinCC 连接.....	649
8.7.4.2	WinCC - SIMATIC NET FMS OPC 服务器的连接.....	653
8.7.4.3	WinCC - SIMATIC NET S7-OPC 服务器的连接	655
8.7.4.4	WinCC - Microsoft Excel 的连接	660
8.8	WinCC OPC HDA 服务器	665
8.8.1	WinCC OPC HDA 服务器的功能.....	665
8.8.2	WinCC OPC HDA 服务器的数据结构.....	666
8.8.2.1	WinCC OPC HDA 服务器的数据结构.....	666
8.8.2.2	受支持属性的概述	667
8.8.2.3	受支持配件的概述	668
8.8.2.4	受支持函数的概述	669
8.8.2.5	WinCC OPC HDA 服务器的时间格式.....	670
8.8.3	质量代码	672
8.8.4	支持的写访问	673
8.8.5	OPC HDA 连接实例	675
8.8.5.1	OPC HDA 连接实例	675
8.8.5.2	HDA 服务器浏览器.....	677
8.8.5.3	如何使用 HDA 服务器浏览器来组态对 WinCC 归档变量的访问	678
8.8.5.4	读取 WinCC 归档变量的数值	679
8.8.6	WinCC 中 OPC HDA 服务器针对非周期性记录的特殊功能.....	681
8.9	WinCC OPC A&E 服务器	684
8.9.1	WinCC OPC A&E 服务器的功能.....	684
8.9.2	在 OPC A&E 上映射 WinCC 消息系统	685
8.9.2.1	在 OPC A&E 上映射 WinCC 消息系统	685
8.9.2.2	映射 WinCC 消息类别和消息类型	687
8.9.2.3	映射 WinCC 消息优先级	687
8.9.2.4	WinCC 消息系统的属性	688
8.9.2.5	确认方法	690
8.9.3	OPC A&E 质量代码.....	692
8.9.4	OPC A&E 连接实例.....	693
8.9.4.1	OPC A&E 连接实例.....	693
8.9.4.2	如何组态访问 WinCC 消息系统.....	693
8.9.5	带层级访问的 OPC A&E 服务器.....	695
8.9.5.1	OPC A&E 服务器的功能.....	695
8.9.5.2	OPC A&E 与带层级访问的 OPC A&E 之间的差异.....	697
8.9.5.3	在 OPC A&E 上映射 WinCC 消息系统	699
8.9.5.4	OPC A&E 质量代码.....	705
8.9.6	读取归档消息	706
8.9.6.1	访问归档消息	706
8.9.6.2	使用 OPC 访问归档消息的语法.....	707
8.9.6.3	归档消息的读方法	708
8.9.6.4	识别归档消息	710

8.10	WinCC OPC UA 服务器	711
8.10.1	WinCC OPC UA 服务器的工作原理	711
8.10.2	OPC UA 的安全概念	712
8.10.3	组态安全机制	716
8.10.4	支持的 OPC UA 服务和配置文件	720
8.10.5	WinCC OPC UA 服务器的名称区域	722
8.10.6	OPC UA 数据访问	725
8.10.7	OPC UA 历史访问	725
8.10.8	OPC UA 报警和条件	727
8.10.9	WinCC 消息系统的属性	731
8.10.10	WinCC OPC UA 服务器的组态	733
8.10.10.1	WinCC OPC UA 服务器的组态文件	733
8.10.10.2	如何组态 OPC UA 服务器	738
8.11	诊断	741
9	REST 接口	743
9.1	WinCC 中的 REST 接口	743
9.2	如何组态 WinCC REST 服务	746
9.3	方法概述	748
9.4	GET / POST 方法	749
9.4.1	读取一个变量的运行系统值	749
9.4.2	读取所有变量的运行系统值	749
9.4.3	读取一个连接的组态数据	750
9.4.4	读取所有连接的组态数据	751
9.4.5	读取一个变量的组态数据	752
9.4.6	读取所有变量的组态数据	753
9.4.7	读取一个变量组的组态数据	755
9.4.8	读取所有变量组的组态数据	755
9.4.9	读取一个结构类型的组态数据	757
9.4.10	读取所有结构类型的组态数据	757
9.4.11	读取一个结构类型的实例	760
9.4.12	读取多个结构类型的实例	761
9.5	PUT 方法	763
9.5.1	将值写入到一个变量	763
9.5.2	将值写入到多个变量	763
9.6	组态数据过滤器	765
10	WinCC/Cloud Connector	769
10.1	WinCC/Cloud Connector	769
10.2	Cloud Connector 许可	772
10.3	通过 MQTT 将数据传送到云端	773

10.4	通过 REST 将数据传送到云端	777
10.5	WinCC 变量管理中的设置	779
10.6	WinCC Cloud Connector 中的 MQTT 设置	781
10.7	WinCC Cloud Connector 中的 REST 设置	786
10.8	如何通过 MQTT 组态云连接	788
10.9	如何组态 MindSphere 连接	792
10.10	如何通过 REST 组态云连接	795
10.11	云连接诊断	797
	索引	799

多用户系统

1.1 WinCC 中的多用户系统

内容

WinCC 通过组态客户机/服务器系统，可以将系统操作和监控的功能分配到多个客户机和服务器上。对于较大的系统，这样的组态方式既可以降低单台计算机的负担，也可以增加系统的性能。

本章将说明：

- 使用 WinCC 可实现哪些客户机/服务器方案。
- 如何组态客户机/服务器系统中的服务器和客户机。
- 如何组态能提供多个服务器上视图的客户机。
- 在运行系统中客户机/服务器系统是如何运作的。
- 如何从远程客户机组态服务器上的项目。

1.2 WinCC 中的客户机/服务器系统

简介

可使用 WinCC 来组态含有多个客户端和服务器的客户端/服务器系统，从而更有效地操作和监控大型系统。

把对过程进行操作和监控的任务分配到多个服务器上，既减少了单个服务器的使用率，又能使系统的性能得到改善。此外，还可以使用 WinCC 映射技术或拓扑很复杂的结构的系统。

使用客户端/服务器系统，例如：

- 对于大型系统，需要多个用于监视和操作的工作站（客户端）来完成同一个任务。
- 要将操作员控制和监视任务分布到多个操作员站时，诸如用于显示一个系统的全部消息的中央客户端。

客户端可用于：

- 要组态包含一台服务器的多用户系统：
多个客户端访问同一台服务器上的项目，该服务器有过程驱动器连接。
对于多用户系统，不必对客户端进行组态，因为服务器已提供了所有的数据。
- 组态包含多台服务器的分布式系统：
客户端能访问来自不同服务器的数据，这些服务器都有过程驱动器连接。
在分布式系统内，对每个客户端进行单独组态，必要的服务器数据被导入到客户端上，如果数据服务器上的被修改，则客户端上的数据将自动进行更新。
过程数据由服务器提供。
- 远程组态：
从一台客户端组态一个服务器项目。
- 远程监视：
从一台客户端监视一个服务器项目。

组态客户端/服务器系统的条件

- 为了使用 WinCC 组态客户端/服务器系统，每台 WinCC 服务器上都必须安装“WinCC Server”选件。
- 必须在 WinCC 系统中使用的所有 WinCC 站上安装相同的 WinCC 版本。
例如，这包括：
 - WinCC 服务器、冗余服务器、归档服务器和文件服务器
 - 自带项目的 WinCC 客户端和未自带项目的 WinCC 客户端
 - 安装了 WinCC 选件的 PC，例如 WebNavigator 服务器、WebNavigator 客户端、WinCC/Audit、WinCC/Calendar 选件

对于安装了 WinCC 选件的服务器、客户端或 PC，尚未发布支持不同 WinCC 版本的混合组态。

务必在 WinCC 系统中的所有 PC 上安装 WinCC 更新。如果客户端或服务器上安装了不同的更新版本，则无法在 WinCC 项目中进行任何组态更改。

- 客户端/服务器系统中的所有 PC 都必须通过网络（局域网）相互连接。
也可以将通过路由器连接的邻近子网中的 PC 登录为系统中的客户端或服务器。
- 默认情况下，将禁用远程访问。
如果要启用对 PC 的网络访问，请在 Simatic Shell 设置中激活远程通信。

冗余系统

为了在出现故障（例如服务器故障）的情况下仍然能维持系统运转，请组态冗余服务器。有关组态冗余系统的详细信息，可参见 WinCC 文档《冗余系统》。

该文档也包含了在冗余系统中对客户端组态的信息。

冗余服务器对上的客户端

只能在服务器（主服务器或备用服务器）上编辑包含没有本地项目的客户端的 WinCC 项目。

在此时间段内，不能在伙伴服务器上打开 WinCC 项目。WinCC 项目管理器 and 运行系统必须关闭。

仍可将主服务器或备用服务器组态为首选服务器，以在运行系统中分配负载。

运行系统激活时，在客户端上 WinCC 项目管理器应保持关闭。最好使用“自动启动”启动运行系统。为此，可使用 WinCC 的“自动启动组态”工具。

1.2 WinCC 中的客户机/服务器系统

如果在 WinCC 项目管理器仍处于打开状态时切换到其它服务器计算机，操作员输入将被锁定（用户界面“变灰”）。如果打开项目的服务器不是客户端的当前服务器，则无法启动 WinCC 编辑器。

说明

远程访问以打开 WinCC 项目

如果在两个冗余系统上都打开了该项目时，则不允许进行远程组态。

为了组态冗余系统，必须禁止备用计算机打开 WinCC 项目。

为了在运行时组态一个冗余系统，必须按如下步骤操作：

1. 取消激活备用计算机，并关闭项目。
 2. 在运行系统中组态主服务器，远程与本地均可。
 3. 在线组态完成之后，使用运行系统中的项目复制器复制备用计算机上的项目。
 4. 再次打开并激活备用计算机上的项目。
-

参见

使用客户机/服务器系统中的 OPC 接口 (页 88)

远程组态 (页 73)

运行系统中的系统特性 (页 65)

客户机组态 (页 42)

服务器组态 (页 27)

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

典型组态 (页 19)

加密通信 (页 74)

1.3 典型组态

可以根据需要组态不同的客户端/服务器解决方案。这样便可选择使用客户端、Web 客户端和瘦客户端。

另请遵循“数量结构与性能 (页 21)”中有关数量结构和性能の説明。

在分布式系统中使用加密通信

在多用户系统中的计算机之间建立加密通信。

使用“Simatic Shell”为计算机指定 PSK 密钥。这意味着，只有那些在通信前知道指定的共享密钥的计算机才能相互通信。可以为同一网络指定不同环境以及各自的 PSK 密钥。

满足不同要求的客户端/服务器系统的组态

客户端

根据组态的不同，客户端/服务器系统中的客户端可以：

- 在多台客户端上显示来自同一台服务器的视图（多用户系统）
- 在客户端上显示多台服务器的视图（分布式系统）
- 从客户端上组态服务器项目（远程）
- 从客户端上激活和取消激活服务器项目（远程）

为了组态多台客户端，每台服务器上都需要“WinCC Server”选项。

Web 客户端

Web 客户端安装在客户端/服务器系统中，例如在下列情况时：

- 需要通过窄带连接访问系统时
- 只需要临时访问数据时
- 必须远距离（例如通过 Internet）访问数据时

Web 客户端具有下列优点：

- 可使用具有不同操作系统的客户端
- 可以通过多台 Web 客户端同时访问一台服务器
- 可实现大型的数量结构

为了组态 Web 客户端，需要 WinCC 的最小安装和“WinCC WebNavigator”选项。

瘦客户端

瘦客户端基本上具有与 Web 客户端相同的主要特性以及附加的特性：

- 其还可用于稳定可靠的客户端平台

为了组态瘦客户端，需要 WinCC 的最小安装和“WinCC WebNavigator”选项。

参见

客户机/服务器方案 (页 24)

使用来自不同服务器的数据 (页 60)

显示来自不同服务器的画面 (页 56)

如何组态首选服务器 (页 52)

如何组态标准服务器 (页 49)

组态导入数据包 (页 46)

显示来自不同服务器的消息 (页 62)

在客户机上创建新项目 (页 44)

客户机组态 (页 42)

在服务器上创建新项目 (页 29)

如何在计算机列表中注册客户机 (页 31)

如何组态数据包导出 (页 35)

在服务器项目中组态客户机 (页 39)

WinCC 中的客户机/服务器系统 (页 16)

数量结构与性能 (页 21)

加密通信 (页 74)

1.4 数量结构与性能

运行系统中 WinCC 项目的性能

WinCC 项目的性能取决于系统组态、数量结构和组态。其中的每个因素都会影响更改画面所用的时间和归档速度等方面。

有关系统组态的信息，请参见“典型组态 (页 19)”。另请遵循安装说明中的硬件要求。

关于组态的注意事项

以下说明不仅适用于多用户系统，而且还适用于各个类型的 WinCC 项目。

不过，对于多用户系统，应特别注意这些说明，因为相应的数量结构会快速增长，并且组态更改影响巨大。

请注意，组态时，以下因素会影响 WinCC 项目的性能：

- **Web 客户端数**
对于使用的每个附加 Web 客户端，项目中待处理的变量数都会随之增长：
对于每个 Web 客户端，在 Web 服务器中将 WinCC 项目的内部变量作为计算机的本地变量进行管理。附加 Web 客户端会使管理的变量数增多，具体取决于组态。
- **过程窗口中的画面窗口数**
建议最多同时显示 100 个画面窗口。
- **画面窗口中的嵌套数（级数）**
经验证，画面对象的最大嵌套深度可达二十 (20) 级。
- **过程画面中的脚本**
确保周期相同的所有脚本的处理时间不长于组态的周期时间。
- **变量数**
WinCC Runtime 中的服务器负载来自于同时所有客户端或 Web 客户端中注册的所有变量之和。
- **监视器数量（多个 VGA）**
如果使用多 VGA，则可能会减少 WinCC 客户端数量。
在这种情况下，还要注意由过程画面的数量和复杂性引起的潜在性能限制。
建议将显示器数量限制为 60。
示例：如果在所有 WinCC 客户端上都各使用四个监视器，则最多支持 15 个 WinCC 客户端。

要分析 WinCC 项目在读取、写入和归档数据时的性能，请使用“@PRF_...”系统变量。

在服务器上同时启动多个客户端

同时启动与一个服务器连接的多个客户端可能会导致过载。在这种情况下，这些客户端将进入超时状态。

我们建议依次启动各客户端。

客户端和服务器的可能数量

根据所使用客户端的类型和数目，可实现不同的数量结构。可以使用混合系统，即，可以在一个客户端/服务器系统中同时使用多个客户端和 Web 客户端。

如果仅使用带有一个自定义项目的 WinCC 客户端，则在一个 WinCC 网络中最多可有 50 个并行客户端访问一个服务器。在运行系统中，一个 WinCC 客户端最多可访问 18 台服务器。

最多可使用 36 台服务器，其形式为 18 个冗余服务器对。

如果仅使用 Web 客户端，则可归档多达 151 个客户端（1 个客户端和 150 个 Web 客户端）的数量结构。

还应考虑多 VGA 对客户端数目的影响。

混合系统的组态

在组态混合系统时，遵守下列经验规则，以获得最大的数量结构。

为客户端类型定义了下列值：

- Web 客户端/瘦客户端 = 1
- 客户端 = 2
- 具有“远程组态”功能的客户端 = 4

所有客户端数值的总和不应超过以下值：

- 无操作员功能的 WinCC 服务器：每台服务器 160
- 带操作员功能的 WinCC 服务器：每台服务器 16

示例：

组态	含义
3 台具有“远程组态”功能的客户端	$3 \times 4 = 12$
5 台客户端	$5 \times 2 = 10$
138 Web 客户端	$138 \times 1 = 138$
总和	160

说明**未使用 WinCC 服务器混合组态**

将其它 WinCC 服务器作为客户端来访问的 WinCC 服务器的混合组态未经批准。

参见

WinCC 中的客户机/服务器系统 (页 16)

使用客户机/服务器系统中的 OPC 接口 (页 88)

远程组态 (页 73)

运行系统中的系统特性 (页 65)

客户机组态 (页 42)

服务器组态 (页 27)

客户机/服务器方案 (页 24)

典型组态 (页 19)

1.5 客户机/服务器方案

简介

根据应用情况，可以使用 WinCC 来实现不同的客户端/服务器方案：

多用户系统

通常情况下，要为需要将数据分发给多个服务器的小型系统组态多用户系统。

还要组态带有过程驱动器连接的服务器，这样，该服务器就可负责所有中心功能和多个操作员站（客户端）的管理。

单个操作员站可以执行相同或不同的任务。

根据其操作员授权，客户端可用于下列场合：

- 监控系统。
- 监控和操作系统。
- 服务器项目的远程组态，例如，作为服务计算机。

根据其组态，客户端可用于下列场合：

- 举例来说，当过程能够在系统的不同位置进行操作时显示项目的同一视图。
- 显示项目的不同视图，例如，只显示消息。

发布用户授权来定义操作员能够在某个操作控制台上利用的功能。

组态类型：

- 无自带项目的客户端：如果操作员站不需要自己的项目数据，则在服务器项目中组态客户端。
- 具有自带项目的客户端：如果操作员站需要自己的项目数据，例如不同起始画面，则创建客户端项目。

分布式系统

可实现多个服务器的分布式系统通常用于必须处理大量数据的大型系统。在多个服务器中分配任务，其结果是，减少了加载到单个服务器上的负载。这可使得系统性能更高、实现更大型的典型应用。

如果在 WinCC 系统中组态了分布式系统，则可根据过程步骤或功能，通过相应的组态在服务器中分配过程任务：

- 从技术分配的角度来讲，每个服务器将接管系统中技术上有限的区域，例如，某一印刷或烘干单元。
- 就功能上的分配来说，各个服务器将接管某一任务，例如，可视化、归档、发出报警等。

在运行系统中，分布式系统中的每个客户端均可显示多达 18 个不同服务器或冗余服务器对中的数据。分布式系统中的各个客户端将使用基准画面和某些本地数据单独进行组态。用于显示过程数据的服务器数据从服务器传送到客户端，并可在必要时自动进行更新。

文件服务器

可以为客户端-服务器系统提供文件服务器，以便在集中的系统中保存和管理所有项目。例如，这样可以更方便地创建所有项目的定期备份副本。具有过程驱动器连接的服务器可访问文件服务器，并可在文件服务器上组态项目。文件服务器只可用于组态。

必要时，可通过添加更多的硬件组件，使文件服务器满足某些特定的要求。例如，为保证备份安全性而生成镜像磁盘。

中央归档服务器

可将所有连接的 WinCC 服务器的过程值和消息保存到中央归档服务器上（例如，Process Historian）。运行期间，可以照常在 WinCC 在线趋势控件或 WinCC 报警控件的过程画面上显示所保存的过程值和消息。此外，通过定义的接口（如 OLE-DB），可以直接访问已归档的过程值和消息。例如，采取这种方式，可以对整个公司的重要生产数据进行分析。

服务器-服务器通讯

在两台服务器之间的通讯期间，一台服务器访问另一台服务器上的数据。一台服务器可以访问多达 18 个其它服务器或冗余服务器对上的数据。在组态和操作方面，进行数据访问的服务器与客户端相同，除了不能对标准服务器进行组态。

进行访问的每台服务器都需要 WinCC 服务器许可证。在对系统进行组态时，数量结构中必须包含正在访问的服务器。

标准服务器

在分布式系统中，特定服务器的服务器前缀会分配给您的数据，以便 WinCC 控件能够显示消息和过程数据。

1.5 客户机/服务器方案

在分布式系统中，将为客户端组态一个标准服务器，以便可以从标准服务器请求没有指定唯一前缀的数据。如果没有指定任何标准服务器，则将试图从本地访问相应的数据。如果不存在任何本地数据管理（例如消息和归档），则访问将被拒绝，并发出一条出错消息。

首选服务器

如果在多用户系统或分布式系统中使用冗余服务器，请在客户端中组态一个首选服务器。

首选服务器是冗余服务器对中的服务器，它在多用户系统中对客户端而言具有高优先级。为了确保系统的可操作性，可分别为每台客户端选择一个首选服务器。

参见

如何组态首选服务器 (页 52)

如何组态标准服务器 (页 49)

运行系统中的系统特性 (页 65)

客户机组态 (页 42)

服务器组态 (页 27)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.6 服务器组态

1.6.1 服务器组态

引言

WinCC 网络中的服务器可完成下列任务：

- 连接过程
- 采集过程值
- 归档消息和过程值
- 给客户机提供过程数据
- 给客户机提供组态数据

单个服务器所承担的任务可根据技术或功能等方面进行分配：

- 技术方面：每个服务器对过程/系统的指定区进行管理。
- 功能方面：每个服务器在运行时完成整个过程中的某个指定任务，例如报警记录或归档。

组态步骤

组态多用户系统

对于多用户系统，可组态多个客户机，用于显示运行系统中一个服务器的视图。客户机专门接收一个服务器的数据，没有任何单独的组态。

组态多用户系统中的服务器，可进行如下操作：

1. 在服务器上创建类型为“多用户项目”的新项目。
2. 在服务器上组态必需的项目数据（画面、归档、变量……）。
3. 包括要组态的客户机或在服务器的计算机列表中要进行远程监视的客户机。
4. 给客户机分配操作权限，以启用远程组态。
5. 激活服务器上的数据包自动导入功能。
6. 在服务器项目中组态客户机属性（起始画面、锁定组合键……）。

组态分布式系统

在分布式系统中，可用多个服务器上的视图来组态客户机。客户机具有自己的包含本地数据的项目。服务器所更新的数据将通过数据包导出特性传送到客户机。

1.6 服务器组态

组态分布式系统中的服务器的步骤如下：

1. 在每个服务器上创建类型为“多用户项目”的新项目。
2. 在每个服务器上组态必需的项目数据（画面、归档、变量……）。根据分类的不同（技术/功能方面），也可能涉及到特定的项目数据，例如，只有归档。
3. 应具有远程组态能力的客户机必须在服务器上的计算机列表中注册。
4. 给客户机分配操作权限，以启用远程组态。
5. 组态数据包导出（手动或自动）。
6. 组态客户机上的客户机项目。
7. 使客户机可利用服务器数据（数据包）。

说明

请始终在组态相关的客户机之前先组态客户机/服务器系统的服务器。

参见

如何组态首选服务器 (页 52)

在服务器项目中组态客户机 (页 39)

如何组态数据包导出 (页 35)

如何组态操作员授权 (页 32)

如何在计算机列表中注册客户机 (页 31)

在服务器上创建新项目 (页 29)

客户机组态 (页 42)

客户机/服务器方案 (页 24)

WinCC 中的客户机/服务器系统 (页 16)

1.6.2 在服务器上创建新项目

简介

当在 WinCC 中创建新项目时，选择下列项目类型：

- 单用户项目：
项目只有单独的操作员站，它执行运行系统中的所有任务（过程驱动器连接、操作、监控、归档等）。
与客户端/服务器系统无关。
- 多用户项目：
用于多用户系统或组态了多个客户端和/或服务器的分布式系统的服务器项目。
- 客户端项目：
分布式系统内一个客户端的项目，可显示多个服务器上的视图。

说明

更改项目类型

通过修改项目类型，也可将现有的项目转换为服务器项目（“计算机属性”(Computer Properties) 对话框 >“常规”(General) 选项卡）。

设置端口

为在服务器和客户端之间进行通信，操作系统会在 1024 到 65535 的范围内动态选择一个端口。

还可以在 Simatic Shell 的通信设置中指定特定端口。

激活“加密通信”(Encrypted communication) 并输入所需端口。

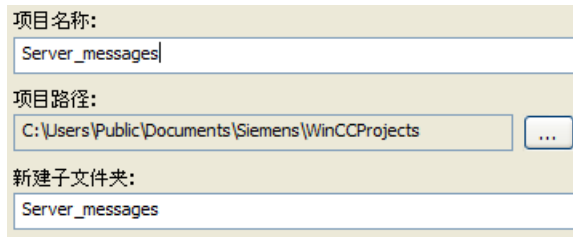
操作步骤

下列步骤描述了如何创建多用户系统或分布式系统的服务器项目：

1. 在 WinCC 项目管理器中打开服务器，选择菜单命令“文件 > 新建”(File > New)。“WinCC 项目管理器”对话框打开：



2. 选择“多用户项目”并单击“确定”。将出现“创建新项目”对话框。
3. 如果目录名称与项目名称不同，输入项目名称和子目录名称。默认将以下文件夹作为项目路径：
 - “Public documents\Siemens\WinCCProjects”



4. 单击“创建”(Create) 按钮。项目在 WinCC 项目管理器中创建并打开。当前项目将自动作为服务器项目。

参见

- 在服务器项目中组态客户机 (页 39)
- 如何组态数据包导出 (页 35)
- 如何组态操作员授权 (页 32)
- 如何在计算机列表中注册客户机 (页 31)
- 服务器组态 (页 27)
- 客户机组态 (页 42)
- 客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.6.3 如何在计算机列表中注册客户机

引言

如果希望客户机以远程方式或在运行系统中访问一个服务器，则该客户机必须注册到服务器的计算机列表中。

先决条件

已经创建了一个多用户项目（多工作站系统或分布式系统）。于是组态计算机将自动作为客户机/服务器系统中的服务器。

步骤

1. 在 WinCC 项目管理器中，选择“计算机”，然后选择弹出式菜单条目“新建计算机...”。将显示“计算机属性”对话框。
2. 输入具有访问当前服务器权限的客户机的计算机名。
3. 单击“确定”，以便将计算机注册到项目的计算机列表中。
4. 对客户机/服务器系统中，所有具有访问当前服务器权限的计算机重复上述步骤。

说明

要重命名客户机，请从计算机列表中删除所选的客户机。将使用修改后的名称的新客户机作为新计算机添加到计算机列表中。

参见

服务器组态 (页 27)

在服务器项目中组态客户机 (页 39)

如何组态数据包导出 (页 35)

如何组态操作员授权 (页 32)

在服务器上创建新项目 (页 29)

客户机组态 (页 42)

1.6 服务器组态

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.6.4 如何组态操作员授权

WinCC 中的操作员授权

为了使客户端可以远程或在运行时打开并处理服务器项目，必须在服务器项目中组态适当的客户端操作员授权。

为此，服务器上提供以下操作员授权：

- “远程组态”：
可从远程工作站打开一个服务器项目，并对其进行完全访问。
- “远程激活”：
客户端可在运行系统中放置服务器项目。
- “Web 访问 - 仅监视”：
Web 客户端被授权监视工厂。
这种操作员授权与其它客户端的组态无关。

客户端上的组态

如果客户端具有组态服务器项目的授权，则也可从客户端更改服务器项目中的操作员授权。

更改操作员授权时，不会通知网络中的计算机。

新客户端登录到服务器后，更改将生效。

行为

只要客户端打开、激活或取消激活相应服务器中的项目，客户端就会请求操作员授权。

如果服务器上没有相应的操作员授权，则不能对项目进行处理。

客户端关闭了服务器项目后再次打开时，需要重新登录。

说明

操作员授权已链接到用户

所组态的操作员授权只与用户相关，而与计算机无关。

因此，所分配的操作员授权对于以相同用户名登录的所有操作员站都有效。

操作系统中的操作员授权

为了使客户端能够访问服务器项目，服务器上相应的项目文件夹必须能够通过网络访问到：

1. 请在“项目属性”(Project properties)对话框中禁用以下选项：
“项目目录仅供写保护访问共享。”
2. 在操作系统中为应该具有项目访问权限的用户设置具有所有所需权限的授权。

说明

Windows 操作员授权

关于网络安全性，可为启用的项目目录设置不同的 Windows 操作员授权。

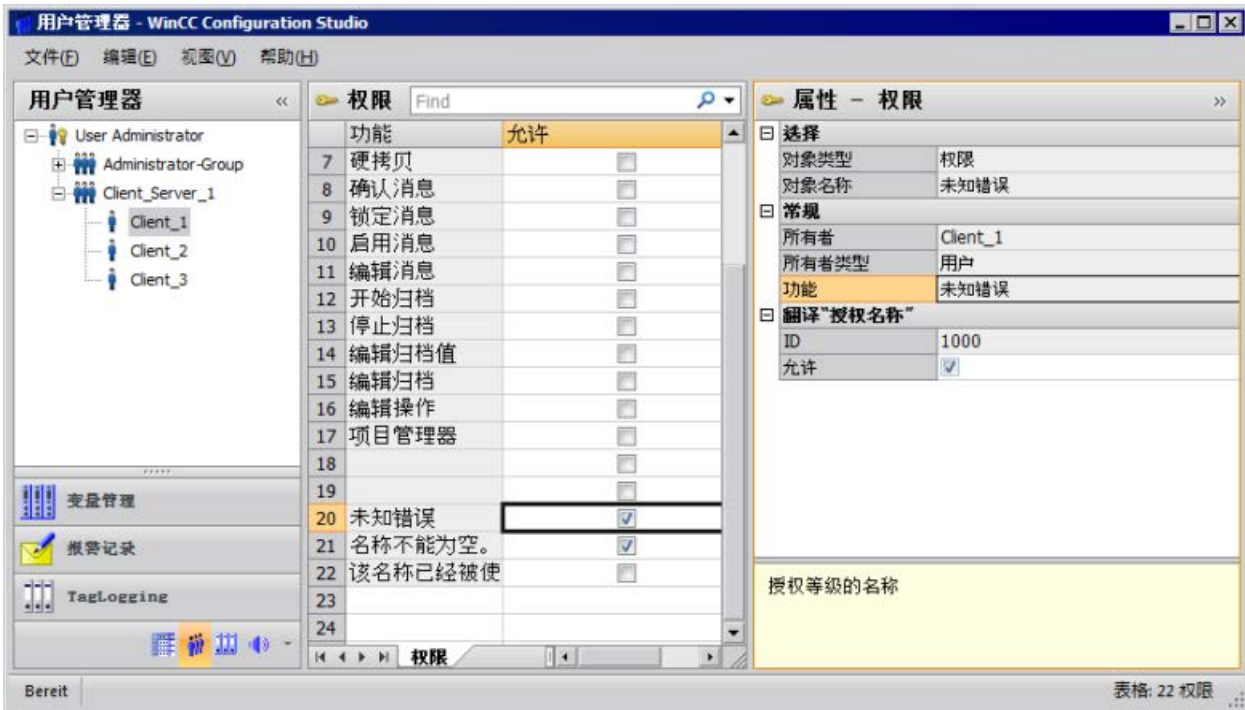
有关操作员授权分配的详细信息，请参见 Windows 文档。

步骤

1. 打开 WinCC 项目管理器中的用户管理器。
2. 在导航区域中选择用户。

1.6 服务器组态

3. 要授予用户对服务器项目的完全访问权限，请启用以下授权：
 - “远程激活”
 - “远程组态”



4. 关闭用户管理器。

参见

- 在服务器项目中组态客户机 (页 39)
- 如何组态数据包导出 (页 35)
- 如何在计算机列表中注册客户机 (页 31)
- 在服务器上创建新项目 (页 29)
- 服务器组态 (页 27)
- 客户机组态 (页 42)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)

1.6.5 如何组态数据包导出

原理

数据包为包含当前所有组态数据（变量、消息、归档等）的数据信息包，这些数据供分布式系统或多用户系统中所有已连接的客户机使用。数据包可从服务器中导出，并可导入到客户机。

首次进行的服务器上的导出和客户机上的导入都是手工完成的。服务器和客户端中数据包的所有进一步更新都可以自动执行。用户可以调整的参数包括数据包何时更新以及启动该更新的原因。例如，在手工调试期间将数据包传送给客户机，以便将组态数据第一次分发给客户机。为了保持客户机上的数据始终是最新的，可组态成在每次修改服务器数据时都对数据包进行自动更新。

说明

如果包含已经创建的数据包的项目被复制到另一台服务器上，则根据新的计算机在 WinCC 计算机属性中修改所复制项目的计算机名称。如果在所复制的项目中重新生成数据包，则必须要在“数据包属性”对话框中更新计算机名称。

选择了服务器-服务器通讯时，服务器将访问另一台服务器的数据。此时访问服务器如同客户端那样，从被访问的服务器上导入数据包。因此，在下面的描述中，有关客户机的细节也同样适用于访问服务器。

在正常操作期间，组态数据既可以手工更新，也可以自动更新：

手工创建数据包

如果有要求，可在服务器上手工创建新的数据包。客户机可以使用这些数据包来进行导入。

自动数据包更新

使用功能“隐含更新”，可自动进行服务器上的数据包导出以及客户机上的数据包导入。

显示在服务器“组态隐含数据包更新”对话框中的选项既影响该服务器上的数据包导出，也影响从其它服务器上的数据包导入。请参见下表中“导入”和“导出”列中的相关内容。

在对话框中，存在 WinCC 的下列各种可能性

WinCC CS 设置	导入	导出	含义
打开项目时更新服务器数据	X		只要项目 打开 ，客户机就进行导入
在收到通知时自动进行更新	X		当满足下列条件时，客户端始终在接收到通知时进行导入： <ul style="list-style-type: none"> • 服务器设置“导出之后通知”已经被激活。 • 项目未激活。

1.6 服务器组态

WinCC CS 设置	导入	导出	含义
监视组态数据的变化。 <ul style="list-style-type: none"> 项目打开时生成服务器数据。 项目关闭时生成服务器数据。 发生变化时立即生成服务器数据 		X	服务器导出数据包 <ul style="list-style-type: none"> 项目打开时 项目关闭时 每当项目数据发生变化时
导出之后通知		X	数据包导出之后，服务器发出通知 如果要使客户机设置“通知时自动更新”生效，该设置应该被激活。
自动导入		X	服务器重新导入其自身导出的数据包 <ul style="list-style-type: none"> 使用符号计算机名称（例如变量）进行组态时，可以不依赖于特定的服务器。 对没有其自身项目的客户机，提供特定服务器的视图。

WinCC RT 的设置	导入	导出	含义
打开项目时更新服务器数据	X		只要项目处于 激活 状态客户机就进行导入
在收到通知时自动进行更新	X		当满足下列条件时，客户端始终在接收到通知时进行导入： <ul style="list-style-type: none"> 客户机设置“通知时自动更新”被激活。 项目被激活。

说明

如果必须经常改变项目数据，例如调试期间或使用组态工具期间，则切勿使用自动数据包导出。

为了组态数据包导出，请使用 WinCC 项目管理器中的服务器数据编辑器。

要求

服务器项目必须打开。

步骤

手工数据包导出

1. 在 WinCC 项目管理器中转到“服务器数据”(Server data)，然后在快捷菜单中选择“创建”(Create)。
2. 在“数据包属性”对话框中，指定服务器的符号和物理名称。该信息可识别客户机上数据包的来源。
组态期间及早定义服务器的物理和符号计算机名。如果符号计算机名称改变，则必须在所有组态数据中都对其进行调整。
符号计算机名称通常由项目名称和物理计算机名称组合而成。
3. 单击“确定”(OK)。创建服务器数据。此过程可能会耗费一些时间，具体取决于组态的大小。

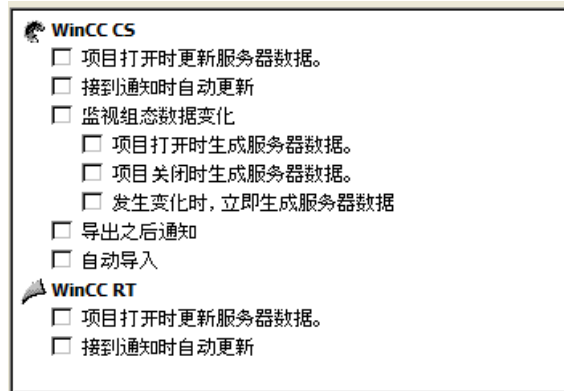
结果

具有服务器数据的数据包位于 WinCC 项目管理器中“服务器数据”下的列表中。数据包将保存在文件系统的 <项目名称>\<计算机\数据包>*.pck 项目目录中。

接下来客户机就可以导入数据包。

自动数据包导出

1. 在 WinCC 项目管理器中转到“服务器数据”(Server data)，然后在快捷菜单中选择“隐含更新”(Implicit update)。



2. 选择需要的选项。可以进行多项选择。
3. 单击“确定”，确认所做的选择。

结果

一旦选择，就将生成来源于服务器的具有该服务器数据的数据包，或者在关闭项目时，更新已从其它服务器导入的数据包。

说明

在使用 SIMATIC Manager 创建的 WinCC 项目中，“服务器数据”(Server data) 快捷菜单中不包含选项“创建...(Create...)”和“隐含更新...(Implicit Update...)”：这同样适用于在 WinCC 中创建并随后使用“导入 WinCC 对象”功能导入到 SIMATIC 管理器中的 WinCC 项目。这种类型的项目也称为 TIA 项目。


如果使用 WinCC 项目管理器复制 TIA 项目，然后使用 WinCC 项目管理器编辑副本，则“服务器数据”(Server data) 快捷菜单中包含菜单条目“创建...(Create...)”和“隐含更新...(Implicit Update...)”。


显示生成的数据包


生成数据包后，它们将在 WinCC 项目管理器数据窗口中作如下显示：

键盘，右边：所加载的数据包

键盘，左边：从服务器导出的数据包

：所加载的数据包，不带标准服务器

：所加载的数据包，带有标准服务器

：服务器导出数据包（未重新导入）

：本地创建的数据包，已重新导入到自己的项目中。

参见

客户机组态 (页 42)

在服务器项目中组态客户机 (页 39)

如何组态操作员授权 (页 32)

如何在计算机列表中注册客户机 (页 31)

在服务器上创建新项目 (页 29)

服务器组态 (页 27)

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.6.6 在服务器项目中组态客户机

原理

如果组态一个多用户系统，在该系统中，多个客户端显示的正好是一个服务器的视图，则不要为客户端创建单独的项目。

可以在服务器项目中配置无自带项目的客户端的行为。

服务器客户端组态

只能在没有自带项目的客户端查看组态了该客户端的服务器。

不允许通过服务器-服务器通讯将该服务器连接至另一个服务器或中央归档服务器。

无自带项目的客户端上的内部变量

以下特性适用于无自带项目的客户端上的内部变量

- “本地计算机”(Computer-local) 设置是相关的。
使用此设置，可以指定在项目范围内或是在本地计算机更新变量更改。
在 WinCC 服务器上，内部变量始终在项目范围内更新。在自带项目的客户端上，内部变量始终在计算机本地更新。
- 如果启用了“本地计算机”(Computer-local) 设置，则“运行系统保持”(Runtime persistence) 设置无效。

WinCC 项目管理器变灰时，应用项目更改

WinCC 项目管理器变灰时，不会应用客户端上的项目更改。

初始状态

- 在自身没有项目的客户机上，WinCC 编辑器在运行系统中处于打开状态。
- WinCC Runtime 在服务器上被禁用。

行为

不会应用在编辑器中所做的更改，例如，项目函数中的脚本更改。

解决方案

在自身没有项目的客户端上，只要 WinCC 项目管理器变灰，则不能进行组态。

要求

- 显示服务器数据的客户端已经注册到服务器的计算机列表中。客户端只尝试访问一个特定的服务器。
- 该服务器不得从其它服务器导入数据包。
- 服务器上的项目已经打开。

步骤

1. 在服务器上打开 WinCC 项目管理器中的计算机列表。
2. 选择要组态的客户端，然后从弹出式菜单中选择“属性”条目：



客户端的“计算机属性”对话框打开。

3. 在“启动”选项卡上，选中要在客户端的运行系统中激活的那些编辑器，例如，如果使用脚本，则激活全局脚本运行系统。
4. 使用“参数”选项卡选择客户端上启动运行系统时所采用的语言。例如，可以组态两台以不同语言显示相同数据的客户端。
5. 在“图形运行系统”选项卡上指定客户端的启动画面。每个客户端的启动画面均可单独进行选择。如有必要，此处可定义窗口属性。
6. 使用“确定”按钮确认设置。
7. 使用同样的方式，组态项目中其它客户端的属性。
8. 在服务器项目中的“服务器数据”编辑器的弹出式菜单中打开“隐含更新”条目。激活“自动导入”设置。
9. 在服务器项目中生成数据包。

参见

如何组态数据包导出 (页 35)

如何组态操作员授权 (页 32)

如何在计算机列表中注册客户机 (页 31)

在服务器上创建新项目 (页 29)

服务器组态 (页 27)

客户机组态 (页 42)

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.7 客户机组态

1.7.1 客户机组态

简介

只有当组态了分布式系统且该系统中客户端将显示多个服务器视图时，才有必要进行客户端组态。如果组态的是多用户系统，且客户端在其中只显示一个服务器上的数据时，则不需要任何客户端组态。客户端将从服务器项目中接收全部数据及其运行环境。

如果组态了包含有多个服务器的客户端/服务器系统，且客户端显示多个服务器（分布式系统）上的不同视图时，则要为每个客户端组态单独的客户端项目。在运行系统中，每个客户端可显示多达 18 个不同服务器或冗余服务器对的视图，例如，显示服务器 1 和服务器 2 的消息，显示并写入服务器 3 的过程值，显示服务器 4 的画面等。

分布式系统中的客户端可根据服务器上各自的操作授权来完成下列操作：

- 监控过程。
- 监控和操作过程。
- 服务器上项目的远程组态。
- 服务器上项目的远程激活和取消激活。

说明

为了显示不同服务器上的数据，服务器前缀（例如服务器名称）在分布式系统内必须是唯一的。

每个客户端都有其自己的组态，并在客户端数据库中本地存储了许多面向管理客户端的数据，例如：

- 局部变量
- 用户管理器数据
- 文本库的数据
- 项目属性
- 用户周期

说明

服务器组态的所有外部数据也必须能为客户端所用，以便可将其正确显示在客户端项目中。例如，外部的数据可以是来自于 WinCC 以外的 ActiveX 控件，以及以 OLE 对象的方式嵌入 WinCC 的外部图形。

组态步骤

1. 组态服务器项目。
 2. 创建和导出服务器数据包。
 3. 组态客户端上的数据包导入。
 4. 组态客户端上的客户端项目。
-

说明

如果取消激活服务器上的运行系统，也必须完成客户端上的运行系统以继续组态。

参见

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

使用来自不同服务器的数据 (页 60)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)

组态客户机的起始画面 (页 54)

如何组态首选服务器 (页 52)

如何组态标准服务器 (页 49)

组态导入数据包 (页 46)

在客户机上创建新项目 (页 44)

服务器组态 (页 27)

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.7.2 在客户机上创建新项目

简介

在 WinCC 中创建新的客户端项目时，请选择下列项目类型：

- 单用户项目：
项目只有单独的操作员站，它执行运行系统中的所有任务（过程驱动器连接、操作、监控、归档等）。
与客户端/服务器系统无关。
- 多用户项目：
用于多用户系统或组态了多个客户端和/或服务器的分布式系统的服务器项目。
- 客户端项目：
分布式系统内一个客户端的项目，可显示多个服务器上的视图。

说明

无自带项目的客户端

如果组态了一个多用户系统，在该系统中，多个客户端显示的正好是一个服务器的视图，则不要为客户端创建本地项目，而是在服务器项目中组态客户端行为。

更改项目类型

通过修改“计算机属性”(Computer Properties)对话框中的项目类型，也可以将现有的项目转换为客户端项目。

设置端口

为在服务器和客户端之间进行通信，操作系统会在 1024 到 65535 的范围内动态选择一个端口。

还可以在 Simatic Shell 的通信设置中指定特定端口。

激活“加密通信”(Encrypted communication) 并输入所需端口。

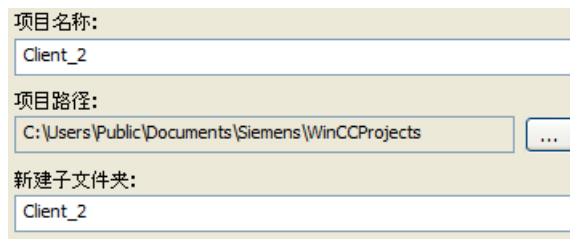
操作步骤

1. 在 WinCC 项目管理器中打开客户端，选择菜单命令“文件 > 新建”(File > New)。“WinCC 项目管理器”对话框打开：



2. 选择“客户端项目”并单击“确定”。将出现“创建新项目”对话框。
3. 如果目录名称与项目名称不同，输入项目名称和子目录名称。默认将以下文件夹作为项目路径：

– “Public documents\Siemens\WinCCProjects”



4. 单击“创建”(Create) 按钮。项目在 WinCC 项目管理器中创建并打开。

参见

- 客户机/服务器方案 (页 24)
- 在服务器项目中组态客户机 (页 39)
- 组态多个服务器消息的消息顺序报表 (页 63)
- 显示来自不同服务器的消息 (页 62)
- 使用来自不同服务器的数据 (页 60)
- 组态客户机上的画面变化 (页 58)
- 显示来自不同服务器的画面 (页 56)
- 组态客户机的起始画面 (页 54)
- 组态导入数据包 (页 46)

1.7 客户机组态

- 如何组态首选服务器 (页 52)
- 如何组态标准服务器 (页 49)
- 客户机组态 (页 42)
- 服务器组态 (页 27)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)
- 典型组态 (页 19)

1.7.3 组态导入数据包

简介

要让分布式系统中的某一客户端显示不同服务器的过程数据，需要相应数据的信息。在分布式系统的服务器上创建含有组态数据的数据包以实现此目的，数据包可供客户端使用。客户端需要服务器的数据包，以便显示这些服务器的数据。

概述

首次进行的服务器上的数据包导出和客户机上的导入都是手工完成的。服务器和客户端中数据包的所有进一步更新都可以自动执行。可以设置更新的执行时间和触发方式。

说明

选择服务器-服务器通信时，服务器可访问另一台服务器的数据。此时访问服务器如同客户端那样从被访问的服务器上导入数据包。因此，在下面的描述中，有关客户机的细节也同样适用于访问服务器。

服务器可以重新导入其自身的数据包，以便组态变量，例如用符号计算机组态独立于特定服务器的变量。

要完成数据包导入，使用 WinCC 项目管理器中的“服务器数据”编辑器。有三种方法可以导入数据包：

手动装载

服务器上生成的数据包被装载到客户机上。使用“加载”(Load) 命令手动触发导入过程。数据包的首次导入必须手动完成。

手动更新

已经由服务器装载到客户机上的数据包使用“更新”命令进行更新。

自动更新

在客户机上可组态隐含数据包更新，以便当指定条件满足时，新建的数据包在客户机上自动进行更新。然而，数据包的首次导入必须手工完成。

设置	含义
对于 WinCC CS <ul style="list-style-type: none"> • 打开项目时更新服务器数据 • 在收到通知时自动进行更新 	<ul style="list-style-type: none"> • 只要项目打开，客户机就进行导入 • 服务器在数据包导出之后发出通知，客户机在收到通知时随时进行导入。 只有在用于数据包导出的服务器上激活了设置“导出之后通知”时，该设置才有效。
对于 WinCC RT <ul style="list-style-type: none"> • 打开项目时更新服务器数据 • 在收到通知时自动进行更新 	<ul style="list-style-type: none"> • 只要项目处于激活状态客户机就进行导入 • 服务器在数据包导出之后发出通知，客户机在收到通知时随时进行导入。 只有在用于数据包导出的服务器上激活了设置“导出之后通知”时，该设置才有效。

要求

- 已在服务器上创建数据包。
- 已打开客户端项目。

步骤

手动装载

1. 打开客户端上的客户端项目。
2. 在 WinCC 项目管理器中选择“服务器数据”(Server data)，并在快捷菜单中选择“加载”(Load)。显示“打开文件”对话框。
3. 选择要加载的数据包，然后单击“确定”(OK)。
默认情况下，数据包以名称“<项目名称_计算机名称>*.pck”存储在目录“...\\<服务器项目名称>\<计算机名称>\Packages\”中。然而，也可以访问存储在任意数据介质中的数据包。
4. 单击“打开”(Open)。数据被装载。如果相应的服务器不可用，在请求新数据包时显示合适的错误条目。

手动更新

1. 打开客户端上的客户端项目。
2. 在 WinCC 项目管理器中选择“服务器数据”(Server data)，并在快捷菜单中选择“更新”(Update) 命令。
3. 数据被更新。在服务器-服务器通讯情况中，如果没有从其它服务器装载任何数据包，一条错误消息显示在服务器上。

自动更新

1. 打开客户端上的客户端项目。
2. 在 WinCC 项目管理器中选择“服务器数据”(Server data)，并在快捷菜单上选择“隐含更新”(Implicit Update)：显示“组态隐含数据包更新”对话框。



3. 选择需要的选项。可以进行多项选择。
4. 单击“确定”(OK) 确认选择。服务器数据将在客户机上自动进行更新，例如在通过网络打开项目或接受通知时。如果相应的服务器不可用，则没有任何错误消息显示在客户机上。

说明


如果在已经激活客户机上项目的情况下，添加新的数据包或删除数据包，就会产生表达上的困难。这种情况可以通过取消激活客户端，然后再次激活客户端来补救。


显示所装载的数据包

数据包装载后，它们将在 WinCC 项目管理器数据窗口中作如下显示：

键盘，右边：所加载的数据包

键盘，左边：已导出，但尚未装载的数据包

 :所装载的数据包，没有标准服务器

 :所装载的数据包，具有标准服务器

参见

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

使用来自不同服务器的数据 (页 60)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)
组态客户机的起始画面 (页 54)
如何组态首选服务器 (页 52)
如何组态标准服务器 (页 49)
在客户机上创建新项目 (页 44)
客户机组态 (页 42)
服务器组态 (页 27)
客户机/服务器方案 (页 24)
数量结构与性能 (页 21)
WinCC 中的客户机/服务器系统 (页 16)
典型组态 (页 19)

1.7.4 如何组态标准服务器

简介

为分布式系统中的客户端组态一个标准服务器，这样如果未指定唯一的服务器前缀（例如，对于变量），将从中请求数据。

如果尚未为组件组态任何标准服务器，则客户端将尝试访问本地的客户端数据（例如内部变量）。如果客户端上未管理任何本地数据（例如消息和归档），则访问将被拒绝，并发出一条出错消息。

要求

只有在导入相应的数据包之后，才能在客户端上选择标准服务器。

步骤

1. 在客户端上的 WinCC 项目管理器中选择“服务器数据”条目。
2. 从弹出式菜单中选择“标准服务器...”条目。
将显示“组态标准服务器”对话框。

1.7 客户机组态

- 单击所需组件条目的计算机符号名称处。从下拉列表框中选择一个服务器。列表包含了客户端上所装载的所有数据包的计算符号名称。



- 对话框中所列出的组件取决于 WinCC 程序安装。如果已经安装选件，则显示组件的同时，还将列出组件选件（例如，SSM - 分屏管理器）。
- 单击“确定”，确认所做的选择。

何时必须为组件选择一个标准服务器？

报警

如果要在客户端上生成操作消息，则必须预先指定用于报警的标准服务器。客户端本身不能组态任何报警记录，而必须在服务器上发出消息。

客户端可以从默认服务器集中检索用户自定义的消息选择。

归档、画面、文本库、用户归档、变量

在客户端上组态了标准服务器后，客户端会在定义的标准服务器上搜索来自那些尚未为其生成任何有效服务器前缀的组件中的数据。如果客户端上尚未组态任何标准服务器，则找不到这些数据相应的服务器，因为不存在任何服务器前缀。

为归档、画面、文本库、用户归档和变量设置默认服务器仅对特殊应用有意义。如果 SIMATIC 文档或客户支持没有明确提示设置一个指定的服务器，则在“组态标准服务器”中的“服务器数据”编辑器中，保留“无标准服务器”的设置。

说明

如果在 WinCC 客户端上为变量输入标准服务器，则在运行系统的工具提示中将不会显示变量管理的任何状态信息。

使用基本过程控制时选择标准服务器

报警

对于报警，必须要指定标准服务器。

变量

决不要为变量指定标准服务器。

SSM (分屏管理器)

一定要为 SSM 组件指定标准服务器。

当 WinCC 客户端上组态了一组趋势时，这组趋势将保存在标准服务器及其冗余伙伴服务器上。其它 WinCC 客户端也可将该服务器指定为 SSM 组件的标准服务器。这样，所编译的趋势组也可为这些 WinCC 客户端所用。如果在 WinCC 客户端上没有为 SSM 组件组态任何标准服务器，则所编译的趋势将被保存在本地计算机上。其它 WinCC 客户端在 WinCC 在线趋势控件中将不能显示这些趋势组。总的来说，在服务器项目中显示这些趋势是不可能的。

如果在 WinCC 客户端上已经组态了画面构成，则只有在为 SSM 组件指定了标准服务器时，才能将它们保存在该服务器上。如果没有指定标准服务器，那么将本地保存所组态的 WinCC 客户端画面构成，而其它客户端则不能对该画面构成进行访问。总的来说，在服务器项目中显示这些画面构成是不可能的。

如果在服务器上组态了冗余，则这些趋势组以及画面构成的数据都会与其冗余伙伴服务器上的数据同步。发生冗余切换后，WinCC 客户端仍然能够请求所有已编译的趋势图的数据和所有画面构成的数据。

参见

组态导入数据包 (页 46)

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

使用来自不同服务器的数据 (页 60)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)

组态客户机的起始画面 (页 54)

如何组态首选服务器 (页 52)

在客户机上创建新项目 (页 44)

客户机组态 (页 42)

服务器组态 (页 27)

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.7.5 如何组态首选服务器

简介

如果使用了冗余服务器，则可以在分布式系统或多用户系统的客户机上组态首选服务器。

首选服务器是冗余服务器组中的服务器，其优先级高于分布式系统中的客户机。只要首选服务器可用，客户机将接收来自首选服务器的数据。

可单独为每个客户机定义首选服务器，以便在冗余服务器中分配客户机，以确保持久的可操作性。如果在连接到组态的服务器期间出现网络中断，客户端将切换到冗余伙伴服务器。当服务器再次可用时，客户端将切换回到首选服务器。

通过在冗余服务器中分配客户机，可对负载进行分配，并改进整个系统的性能。

说明

在主题“冗余系统”下对 WinCC 中冗余系统的组态进行了描述。

步骤

分别为分布式系统和多用户系统的客户机组态首选服务器：

为分布式系统的客户机组态首选服务器

1. 在客户机上的 WinCC 项目管理器中选择“服务器数据”条目。
2. 从快捷菜单中选择“组态”。
随即出现“组态服务器数据”(Configure server data) 对话框。
3. 列表包含了客户机上可从中提取数据包的所有服务器的符号和物理计算机名称。如果冗余服务器可为服务器所用，则将指定物理计算机名称。从冗余服务器组中选择一个服务器作为首选服务器。
分布式系统中的冗余服务器对只有一个通用的符号名称，服务器按该名称进行寻址。

符号的	物理的	冗余的	首选服务器
Project Redundancy Server DPC 40	DPC_4005		无首选服务器

4. 单击“确定”，完成输入。

为多用户系统中的客户机组态首选服务器

客户机必须输入到服务器的计算机列表中。

1. 在服务器上的 WinCC 项目管理器中选择“服务器数据”条目。
2. 从快捷菜单中选择“客户机特定设置”。
出现“客户机特定设置”对话框。
3. 将出现输入到服务器计算机列表中的所有客户机的列表。选择所需要的客户机，并从“首选服务器”列中选择两个冗余服务器中的一个作为首选服务器。



4. 单击“确定”，完成输入。

客户机的运行特性

只要首选服务器可用，客户机就会与作为首选服务器的指定冗余服务器保持连接。

如果首选服务器出现故障，则客户机将切换到冗余伙伴服务器。当出现故障的首选服务器再次变为可用时，客户机又将切换回来。

参见

组态客户机的起始画面 (页 54)

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

1.7 客户机组态

- 使用来自不同服务器的数据 (页 60)
- 组态客户机上的画面变化 (页 58)
- 显示来自不同服务器的画面 (页 56)
- 如何组态标准服务器 (页 49)
- 组态导入数据包 (页 46)
- 在客户机上创建新项目 (页 44)
- 客户机组态 (页 42)
- 服务器组态 (页 27)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)
- 典型组态 (页 19)

1.7.6 组态客户机的起始画面

引言

分布式系统中的任何画面均可用作客户机的起始画面。可以是服务器中的画面、客户机上的本地画面或任何其它画面。

以下描述对如何使用服务器上的画面作为起始画面（起始屏幕）进行了解释。

先决条件

在客户机上导入服务器数据包，其画面将作为起始画面。

步骤

1. 打开客户机上的客户机项目。
2. 在 WinCC 项目管理器中选择计算机，然后从弹出式菜单中选择“属性”选项。
3. 激活“图形运行系统”标签。

4. 输入服务器计算机的名称作为起始屏幕的名称，然后输入将使用的画面，例如：



5. 还可以使用“搜索”按钮搜索画面。选择对话框将显示所有客户机上载入的服务器数据包的画面。
6. 单击“确定”，完成输入。

参见

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

使用来自不同服务器的数据 (页 60)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)

如何组态首选服务器 (页 52)

如何组态标准服务器 (页 49)

组态导入数据包 (页 46)

在客户机上创建新项目 (页 44)

客户机组态 (页 42)

服务器组态 (页 27)

客户机/服务器方案 (页 24)

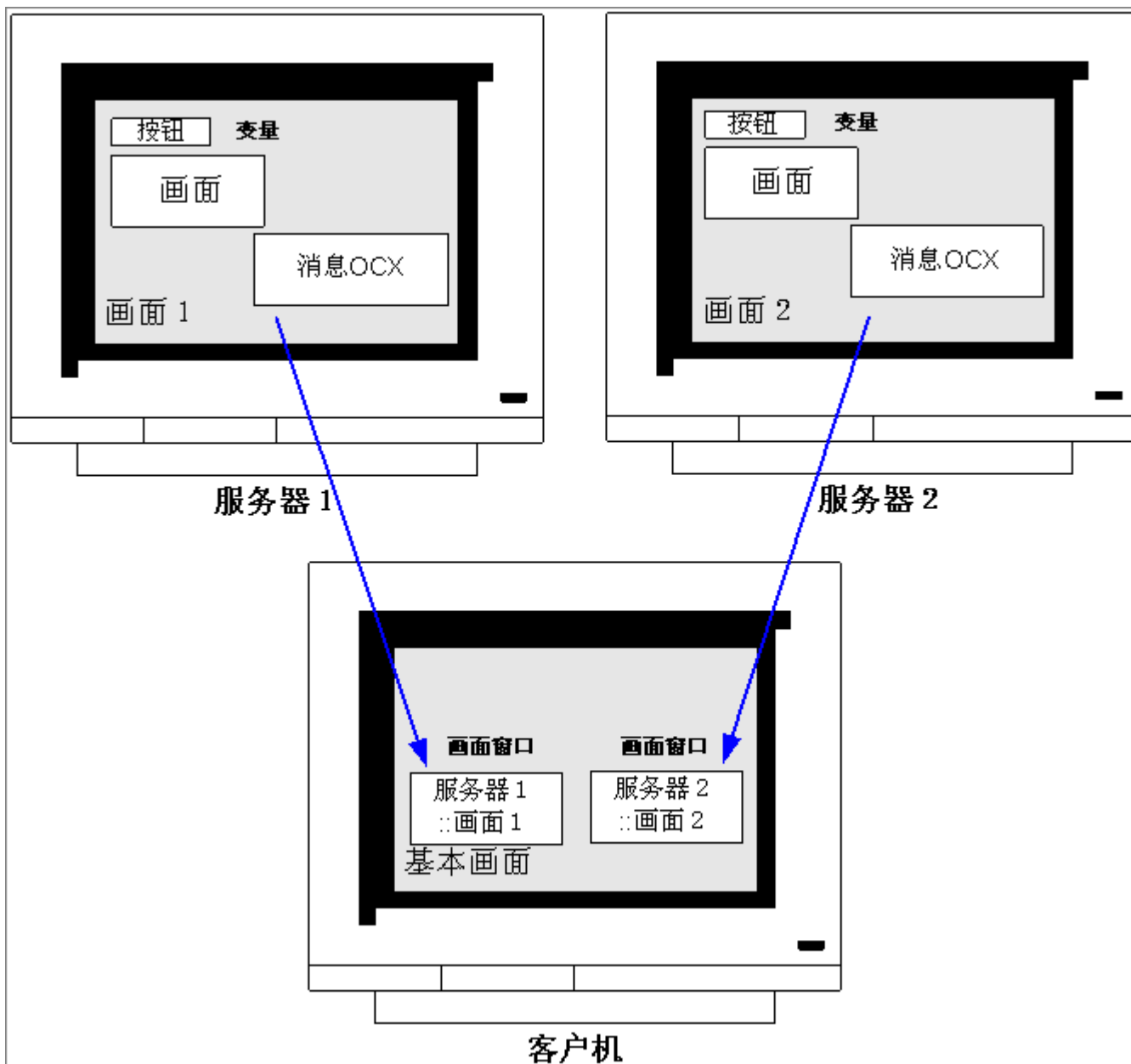
数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

1.7.7 显示来自不同服务器的画面

原理

来自不同服务器的画面可以显示在客户机上所组态的基本画面里面的画面窗口中：



各个画面窗口均可访问来自服务器的数据。为了将服务器画面集成为客户机画面中的画面窗口，必须在画面文件名前面加服务器前缀。

说明

服务器前缀在分布式系统内必须是唯一的。

通过脚本（C 或 VBS）以及通过直接连接，可将服务器画面插入到画面窗口中。

服务器上的画面必须与画面客户机窗口大小相匹配。

先决条件

相应服务器上的数据包必须导入到客户机上。

步骤

1. 在客户机中打开要插入画面窗口的画面。
2. 从图形编辑器的标准选项板中的智能对象组里选择“画面窗口”并将其插入到画面中。
3. 双击画面窗口，打开属性对话框。
4. 从“其它”组中，双击“属性”标签并选择“画面名称”属性，以便搜索画面。

或：

在“画面名称”属性中，双击“静态”列以便以“<服务器前缀>::<画面名称>”的形式直接输入画面名称。

5. 关闭属性对话框。

说明

如果没有在“画面名称”属性中自动指定服务器前缀，则通过“服务器前缀”属性也可输入服务器前缀。双击“服务器前缀”属性后，会出现一个列表，其中包括了所有已将数据包装载到客户机的服务器。

参见

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

使用来自不同服务器的数据 (页 60)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)

1.7 客户机组态

- 如何组态首选服务器 (页 52)
- 如何组态标准服务器 (页 49)
- 组态导入数据包 (页 46)
- 在客户机上创建新项目 (页 44)
- 客户机组态 (页 42)
- 服务器组态 (页 27)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)
- 典型组态 (页 19)

1.7.8 组态客户机上的画面变化

引言

通过给目标画面加上服务器前缀，可以使用分布式系统中的客户机来组态服务器画面上的画面变化。在使用 WinCC 的组态中，对于组态一个“正常”的画面变化还是改变一个基准画面不存在任何差别。

步骤

下列步骤描述了如何组态一个按钮来启动服务器上的画面变化的实例。

1. 在图形编辑器中打开客户机项目的画面。
2. 从 Windows 对象组中将按钮插入到画面中。
显示组态对话框。

3. 在“改变画面”下输入服务器前缀和画面名称，形式为“<服务器前缀>::<画面名称>”，例如：



4. 单击“确定”关闭对话框。

可选步骤

在按钮的“属性”对话框中也可组态画面变化：

- 使用“事件”标签来组态直接连接，例如通过鼠标单击。
- 输入带有服务器前缀的画面名称作为直接连接的常数。

参见

服务器组态 (页 27)

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

使用来自不同服务器的数据 (页 60)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)

如何组态首选服务器 (页 52)

如何组态标准服务器 (页 49)

组态导入数据包 (页 46)

在客户机上创建新项目 (页 44)

客户机组态 (页 42)

1.7 客户机组态

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

1.7.9 使用来自不同服务器的数据

原理

分布式系统中的客户机基准画面以及包含在其中的所有对象均可直接在客户机上进行组态。从每个基本画面中都可对多个服务器中的数据进行访问，例如：

- 来自对系统的部分 A 进行监控的服务器 1 中的过程值输出域、来自对系统另一部分进行监控的服务器 2 中的过程值输出域。
- 以比较的形式来显示不同系统块/服务器中数据的趋势图。
- 显示多个不同服务器消息的消息窗口。



客户机上所设计的基准画面可复制到其他客户机上。为此，在目标客户机上必须有可在基准画面中寻址的服务器数据包。

说明

服务器上所组态的、通过数据包传送给客户机的所有变量，在客户机上的变量选择对话框中均可以使用。

只有当客户机上有 C 动作与函数或 VBS 动作以及全局脚本的过程时，才可以在客户机上运行它们。全局 C 和 VBS 脚本都不是数据包的组成部分。

步骤

下列步骤描述了如何将来自不同服务器的过程数据显示在客户机上的趋势图中的实例。

1. 打开客户机上的客户机项目。
2. 使用图形编辑器组态要用作基本画面的画面。
3. 从对象选项板的“控件”标签中将 WinCC 在线趋势控件插入基本画面中。“WinCC 在线趋势控件属性”对话框打开。
4. 要对当前过程进行监控时，选择“在线变量”作为数据源。
5. 激活“曲线”标签。
6. 通过选择“选择归档/变量”，然后按下“选择”按钮，选择要显示其过程值的第一个趋势的变量。
7. 输入下列形式的变量名称：“<服务器前缀 1>::<变量名称>”。单击“确定”进行确认。
8. 单击“趋势”标签中的“+”按钮，以便添加第二个趋势。
9. 连接第二个趋势，变量来自第二个服务器，其形式为：“<服务器前缀 2>::<变量名称>”。
10. 使用“确定”确认所作的输入。

结果

在运行系统中，两个趋势都显示在客户机上的趋势图中：趋势 1 将显示服务器 1 的数据，趋势 2 将显示服务器 2 的数据。

参见

组态导入数据包 (页 46)

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)

1.7 客户机组态

- 如何组态首选服务器 (页 52)
- 如何组态标准服务器 (页 49)
- 在客户机上创建新项目 (页 44)
- 客户机组态 (页 42)
- 服务器组态 (页 27)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)
- 典型组态 (页 19)

1.7.10 显示来自不同服务器的消息

基本步骤

来自多个服务器的消息在分布式系统中的客户端上显示如下：

- 为应该显示其消息的每台服务器组态一个消息显示
- 在消息显示中指定多个消息服务器作为显示源

说明

如果一个报警控件被集成在客户端的基本画面里，则执行“报警回路”功能时，相关服务器画面将作为基本画面显示在客户端上。这时不能再返回到原有基本画面。

如果报警控件集成在客户端的画面窗口中，则相关的服务器画面在执行“报警回路”功能时将显示在“报警回路”画面窗口中。单击相关的按钮可返回到基本客户端画面。

步骤

1. 打开客户端上的客户端项目。
2. 使用图形编辑器组态要用作基本画面的画面。
3. 在“对象选项板”的“报警控件”标签中将 WinCC 在线趋势控件插入基本画面中。“WinCC 报警控件属性”对话框打开。
4. 当要显示该报警控件中所有已连接服务器的消息时，选择“服务器选择”，并激活“所有服务器”复选框。
5. 如果只要显示指定服务器中的消息，取消激活“所有服务器”复选框，并单击“选择”按钮，从网络中选择一个 WinCC 服务器。
6. 单击“确定”关闭对话框。

说明

在多用户系统中，必须确保客户机选择对话框中所显示的内容在所有服务器上的命名完全相同。

参见

组态多个服务器消息的消息顺序报表 (页 63)

显示来自不同服务器的消息 (页 62)

组态客户机上的画面变化 (页 58)

显示来自不同服务器的画面 (页 56)

如何组态首选服务器 (页 52)

如何组态标准服务器 (页 49)

组态导入数据包 (页 46)

在客户机上创建新项目 (页 44)

客户机组态 (页 42)

服务器组态 (页 27)

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

1.7.11 组态多个服务器消息的消息顺序报表**原理**

如果来自于多个不同服务器的消息将显示在客户机上的基准画面中，则也可以显示相应的消息顺序报表。所有服务器的消息都将按照正确的次序进行收集和输出。

WinCC 为消息顺序报表提供了预组态的布局和打印作业。

步骤

1. 打开 WinCC 项目管理器中报表编辑器下的布局“@CCAlgRtSequence.RPI”。显示行布局编辑器。
2. 单击“选择”按钮。将出现“协议表格选择”对话框。
3. 使用“添加服务器”按钮将应对其消息按消息顺序报表拟定协议的服务器插入到“所选服务器”的列表中。只有那些已在客户机上导入其数据包的服务器才会显示。
4. 使用方向键把将要拟定协议的消息块传送给“报表的列序列”列表。
5. 单击“确定”确认所做的输入。
6. 在 WinCC 项目管理器中打开打印作业“@Report Alarm Logging RT Message sequence”。
7. 已用单独的名称存储了布局，从“布局”列表中选择布局。激活“行式打印机的行布局”复选框。
8. 在“打印机设置”标签中，激活“打印机”复选框。
9. 从所连接打印机的列表中选择打印机，报表将通过其打印输出。
10. 使用“确定”确认所作的输入。
11. 在 WinCC 项目管理器中选择客户机计算机，然后从弹出式菜单中选择“属性”命令。显示“计算机属性”对话框。
12. 激活“启动”标签上的“消息顺序报表”。
13. 使用“确定”确认所作的输入。

参见

- 在客户机上创建新项目 (页 44)
- 组态多个服务器消息的消息顺序报表 (页 63)
- 显示来自不同服务器的消息 (页 62)
- 组态客户机上的画面变化 (页 58)
- 显示来自不同服务器的画面 (页 56)
- 如何组态首选服务器 (页 52)
- 如何组态标准服务器 (页 49)
- 组态导入数据包 (页 46)
- 客户机组态 (页 42)
- 服务器组态 (页 27)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)

1.8 运行系统中的系统特性

1.8.1 运行系统中的系统特性

简介

可使用 WinCC 中的客户端/服务器系统向多个服务器分配系统组态，以减少单个服务器所承担的负载。服务器上所组态的数据可显示在客户端中，且客户端在运行系统中可显示多达 18 个不同服务器或冗余服务器对的数据。

运行系统中编辑器的特性

归档

如果归档系统在操作工作stations上已激活，则变量记录运行系统在服务器上作为归档服务器运行，在客户端上作为归档客户端运行。只有归档服务器才能访问数据库，并对过程数据进行编译和归档。客户端将接收来自归档服务器的归档数据。

归档数据可以以表或图形的形式在每台有变量记录运行系统的客户端上显示。要显示的数据始终来自归档服务器。客户端上的所有操作都将传送给服务器，且处理结果又将传回客户端。

图形

当客户端在运行系统中调用画面时，图形运行系统最初将搜索本地存储的画面。如果本地没有发现任何具有相应名称的画面，则将在服务器的项目文件夹中进行搜索。如果未搜索到任何画面，则会显示相应的消息。

如果画面请求需要与另一个编辑器（报警记录、全局脚本）进行交换，则交换可在本地进行。在运行系统中的多个操作员站可打开和处理同一个画面。

说明

当本地将各自画面复制到客户端时，客户端上的画面构造将更快。在客户端“计算机属性”对话框中的“运行系统”标签输入相关的目录。也可以指定是始终使用该目录，还是优先使用该目录。

如果在服务器项目中修改了画面，则必须通过将所修改的画面手动复制到本地客户端目录来更新数据。

消息

如果消息显示在客户端上，则客户端将从服务器接收所显示的数据。消息服务器会从数据库接收组态的数据。

1.8 运行系统中的系统特性

每台客户端上均可显示归档数据和消息列表。要显示的数据始终来自消息服务器。接收到新消息时，会在消息服务器中对消息进行归档。

当操作员站确认消息时，会将确认转发到消息服务器。服务器在归档中更改输入状态，并向所有参与客户端发布通知。同样的过程也适用于消息的锁定。

如果运行系统中消息服务器不可用，相应的消息将代替消息出现在消息窗口中。当服务器再次可用时，消息将再次显示在消息窗口中。

报表

WinCC 中的协议系统将不会从实际意义上检测运行系统。协议和打印作业可随时进行组态和执行。只有显示归档或过程数据的打印作业才与运行系统有关。

启动例行程序期间，每台客户端上均将自动启动协议系统。服务器将作为协议服务器运行，而客户端作为协议客户端运行。在例行程序启动期间，客户端将登录到服务器上，并接收关于可用打印作业及其状态的当前信息。如果启动客户端上的打印作业，则将从服务器数据库中获得相关的数据。打印作业将从本地启动。协议服务器接收到与客户端打印作业状态有关的当前数据，并将信息传送给其它客户端。

脚本

如果操作员站从本地激活项目，则服务器的项目功能和标准功能均将从本地装载。

用户管理器

用户管理器的运行系统组件将检查操作权限。当激活 WinCC 时，每台计算机都将自动启动用户管理器运行系统组件。如果登录改变，则当前操作权限的列表将从本地数据库中载入。

文本库

如果激活了服务器项目，文本库运行系统在服务器上作为文本服务器运行，在客户端上作为文本客户端运行。数据始终从服务器数据库中读出。

出现系统错误时的特性

如果服务器不可用，则客户端将轮询服务器，直到服务器再次启动。如果出现系统故障，则不能显示服务器上的数据，例如，所有可操作图形对象均将切换到非活动状态。

说明

如果在客户端上运行 WinCC 时出现问题，可以重新启动客户端以重新连接到服务器而又不影响服务器。

参见

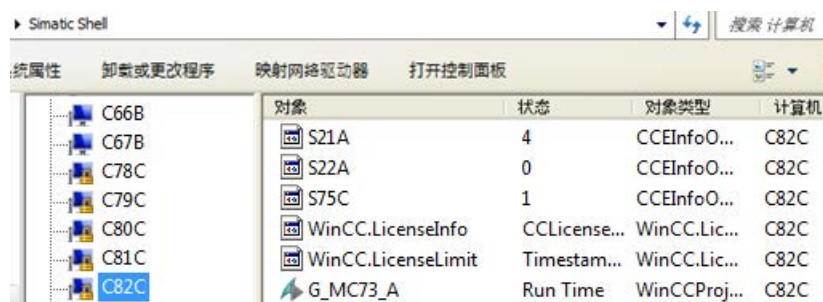
- 启动服务器 (页 67)
- 使用具有多个网卡的服务器时的特殊通讯特性 (页 69)
- 关闭客户机 (页 71)
- 关闭服务器 (页 71)
- 启动客户机 (页 68)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)

1.8.2 启动服务器

原理

客户端/服务器系统中的服务器可独立于客户端进行启动。一旦启动了服务器，客户端就可利用其进行服务，并可检索网络中所有参与者的信息。

可以在“Simatic Shell”对话框的数据窗口中查看所有服务器的当前状态。可通过 Windows 资源管理器打开“Simatic Shell”。



The screenshot shows the 'Simatic Shell' window with a tree view on the left and a table on the right. The tree view lists servers C66B, C67B, C78C, C79C, C80C, C81C, and C82C. The table displays the following data:

对象	状态	对象类型	计算机
S21A	4	CCEInfoO...	C82C
S22A	0	CCEInfoO...	C82C
S75C	1	CCEInfoO...	C82C
WinCC.LicenseInfo	CCLicense...	WinCC.Lic...	C82C
WinCC.LicenseLimit	Timestam...	WinCC.Lic...	C82C
G_MC73_A	Run Time	WinCCProj...	C82C

如果服务器在正常操作期间出现故障，则客户端上的数据将不能再进行更新，且提供出现故障的服务器的信息。

说明

如果客户端/服务器系统中使用了文件服务器，则只有当文件服务器和 WinCC 服务器二者都已经启动时，系统才能够进行操作。

1.8 运行系统中的系统特性

远程激活

也可从另一台远程计算机（客户端或服务器）上启动服务器。其步骤参见“激活项目”。

参见

使用具有多个网卡的服务器时的特殊通讯特性 (页 69)

如何激活项目 (页 85)

关闭客户机 (页 71)

关闭服务器 (页 71)

启动客户机 (页 68)

运行系统中的系统特性 (页 65)

客户机/服务器方案 (页 24)

数量结构与性能 (页 21)

WinCC 中的客户机/服务器系统 (页 16)

1.8.3 启动客户机

原理

客户端/服务器系统中的客户端独立于服务器启动。

当客户端/服务器系统中的客户端启动时，它将通过网络中已知的 WinCC 服务器来接收与下列有关的所有当前信息，例如：

- 项目名称
- 服务器名称和 IP 地址
- 服务器的项目状态（组态或运行系统）

用户可以在“Simatic Shell”对话框的列表中查看相应信息。服务器状态改变时，“Simatic Shell”也会更新。

服务器不可用。

如果服务器不可用，则发出一条相应的出错信息。此外，例如，从服务器接收其数据的图形对象，将切换为不激活状态。

脚本可用于组态客户端连接故障的显示。

参见

- 使用具有多个网卡的服务器时的特殊通讯特性 (页 69)
- 如何激活项目 (页 85)
- 关闭客户机 (页 71)
- 关闭服务器 (页 71)
- 启动服务器 (页 67)
- 运行系统中的系统特性 (页 65)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)

1.8.4 使用具有多个网卡的服务器时的特殊通讯特性

简介

如果在 WinCC 服务器上为过程连接安装了多个网卡或 SIMATIC NET SOFTNET 驱动程序，并使用激活的 TCP/IP 协议运行，则可能影响服务器和 WinCC 客户端之间的通信。

一个可能的原因是服务器中的每个网卡或 SOFTNET 驱动程序都有其自己的 IP 地址。因此，在某些情况下，当服务器注册到网络中时，Windows 将可能通过一个错误的 IP 地址试图建立一个连接，例如，通过 SOFTNET 驱动程序的 IP 地址。如果建立连接的尝试失败，Windows 将把通过该 IP 地址的通信标记为有故障，但不会尝试通过计算机上另一个可用的 IP 地址来建立连接。

在这种情况下，必须由网络管理员进行适当的修改。

检查网卡的顺序

如果在计算机上安装了多个网卡，则用于终端连接的网卡必须放在第一个位置。

在 Windows 的“控制面板”(Control Panel) 中的“网络连接”(Network connections) 下检查网卡顺序。

在“高级”(Advanced) 菜单中，选择菜单命令“高级设置”(Advanced settings)。在“高级”(Advanced) 对话框中，在“网卡和连接”(Network cards and connections) 选项卡的“连接”(Connections) 部分中显示网卡顺序。

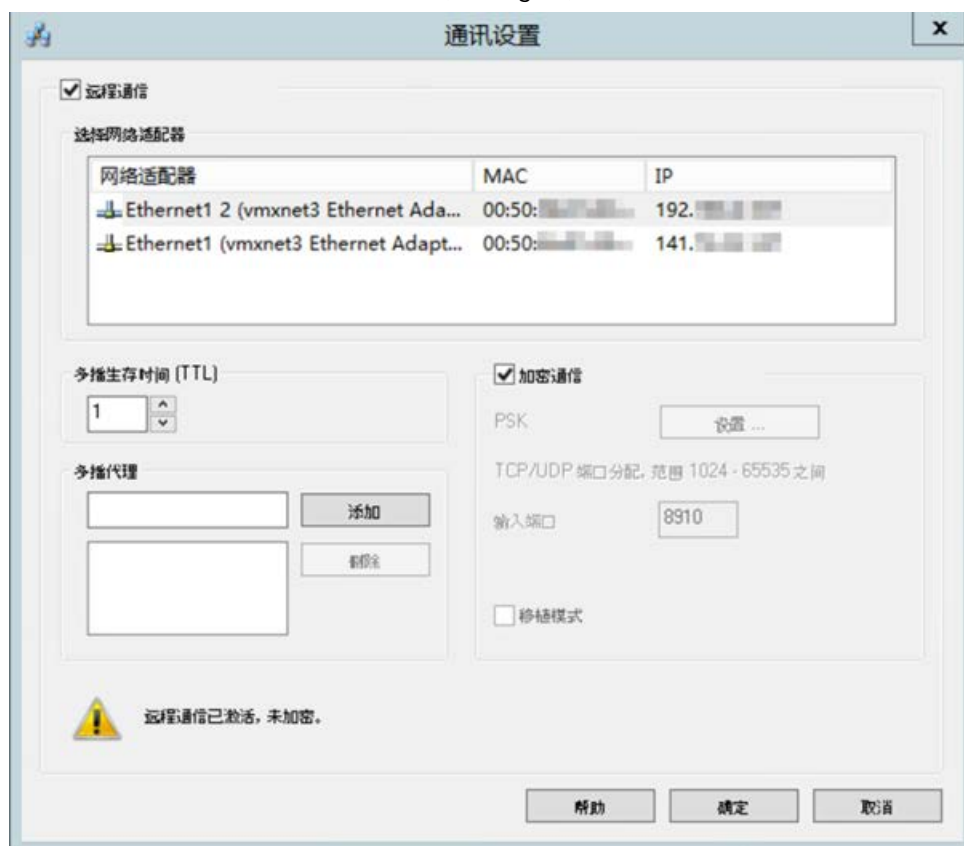
诊断

使用“Simatic Shell”目录，可以检查网卡的组态。

如果确定计算机指示一个错误的地址，即不能访问的网络区，那么选择其它的网络适配器。

步骤

1. 在 Windows 资源管理器的浏览窗口中，单击“Simatic Shell”目录。
2. 在目录的快捷菜单中，选择“设置...(Settings...)”对话框。



3. 如果希望改变网络接口，请单击“网络适配器”区域中期望的网卡。

在服务器上的 SOFTNET 驱动程序的组态中也要进行检查，以判断过程连接所不需要的 Windows 实用程序是否可以取消激活。

如果在检查这几处之后，仍然无法建立连接，请联系客户支持部门。

参见

如何访问子网外的计算机 (页 76)

远程组态 (页 73)

- 启动客户机 (页 68)
- 启动服务器 (页 67)
- 运行系统中的系统特性 (页 65)
- WinCC 中的客户机/服务器系统 (页 16)

1.8.5 关闭服务器

原理

如果关闭了客户端/服务器系统中的服务器，则该服务器不再可以为所连接的客户端提供过程数据。它会同时从系统注销，并在“Simatic Shell”中标记为取消激活。

远程取消激活

也可从另一台远程计算机（客户端或服务器）上关闭服务器。其过程参见“取消激活项目”。

参见

- 启动客户机 (页 68)
- 如何取消激活项目 (页 86)
- 关闭客户机 (页 71)
- 启动服务器 (页 67)
- 运行系统中的系统特性 (页 65)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)

1.8.6 关闭客户机

原理

如果客户机/服务器中的客户机关闭，则其同时从系统登出。

1.8 运行系统中的系统特性

参见

- 如何取消激活项目 (页 86)
- 关闭服务器 (页 71)
- 启动客户机 (页 68)
- 启动服务器 (页 67)
- 运行系统中的系统特性 (页 65)
- 客户机/服务器方案 (页 24)
- 数量结构与性能 (页 21)
- WinCC 中的客户机/服务器系统 (页 16)

1.9 远程组态

1.9.1 远程组态

具有相应操作员授权的客户端可以远程操作服务器项目，例如：

- 服务器项目的远程组态
- 激活服务器项目
- 取消激活服务器项目

有关远程访问和 RDP 的其他信息，请参见 WinCC 信息系统中的版本说明“关于 WinCC 的注意事项 > 远程访问和远程桌面协议 (RDP)”。

Simatic Shell 的功能

对于组态远程访问，“Simatic Shell”对话框可用。

在“Simatic Shell”对话框中，可以通过网络查看已启用的服务器以及具有 WinCC 项目的计算机。

这些项目包括了可在演示许可证下运行的所有项目。

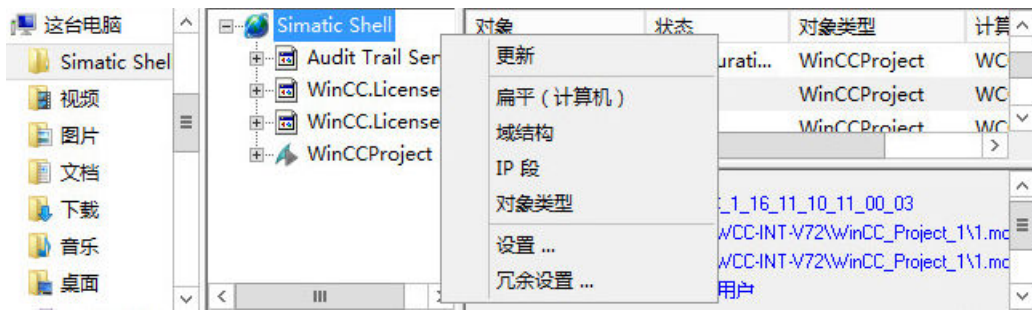
“Simatic Shell”对话框

通过 Windows 资源管理器打开“Simatic Shell”对话框。

PC 和 WinCC 项目可以根据以下标准结构化显示：

扁平式（计算机）	所有条目依次显示在下方。
域结构	服务器及其 WinCC 项目按域组合。
IP 段	服务器及其 WinCC 项目按 IP 地址组合。
对象类型	此视图按服务器类型组合。

可通过双击列标题来分类条目。



为多个路由器组态 IGMP

终端总线上使用“Internet 组管理协议 (IGMP)”网络协议。

使用多台计算机时，只有一台路由器可作为“查询器”处于激活状态。注意下列设置：

设置	组态
IGMP 窥探	“开”
IGMP 查询器	只能通过设置“开”(ON) 激活一个站。 所有其他站均选择设置“关闭”(OFF)。 如果将多个站组态为查询器，则仅激活具有最低交换机 IP 地址的站。
窥探交换机 IP	必须为每个站组态单独、唯一的 IGMP 交换机 IP 地址。

参见

- 如何取消激活项目 (页 86)
- 如何激活项目 (页 85)
- 如何编辑服务器项目画面 (页 84)
- 如何打开一个要进行编辑的项目 (页 82)
- 从多个客户机访问项目 (页 80)
- 如何访问子网外的计算机 (页 76)
- WinCC 中的客户机/服务器系统 (页 16)
- 加密通信 (页 74)

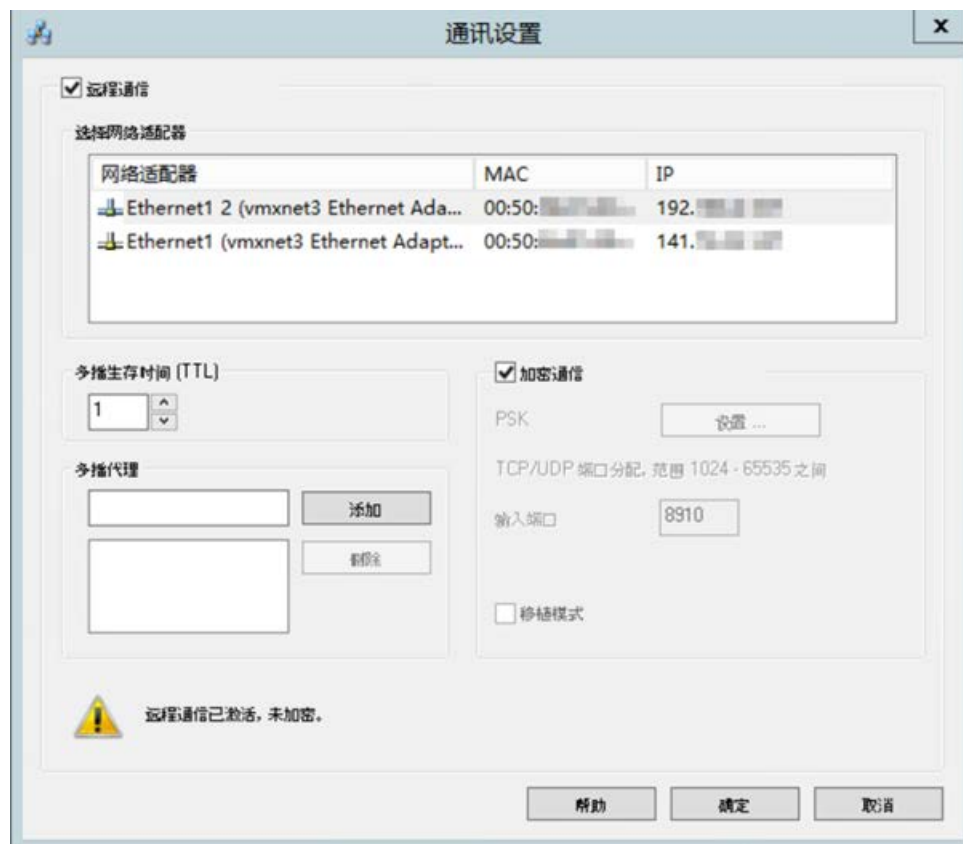
1.9.2 加密通信

访问计算机时，始终确保已建立了计算机的加密通信。

例如，仅为移植临时使用未加密通信选项。

加密通信

在 Windows 资源管理器中的“SIMATIC Shell”快捷菜单中打开通信设置。



如果使用加密通信，则只能与指定了相同 PSK 密钥的计算机建立连接。只能与这些计算机进行通信。不能连接到未加密的计算机。

可以为同一网络指定不同环境以及各自的 PSK 密钥。

根据加密通信的组态，只有相关计算机显示在 Simatic Shell 中。

有关组态的相关信息，请参见“如何访问子网外的计算机 (页 76)”。

身份验证：自签名证书

自签名证书不支持在 WinCC 站之间的通信中使用。

如果服务器上只能找到自签名证书，则会使用组态的 PSK 密钥进行通信。

WinCC 的 Microsoft SQL Server

有关加密通信的详细信息，请参见以下 Microsoft 文章：

- 激活与数据库引擎的加密连接 (<https://docs.microsoft.com/zh-cn/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-2017>)

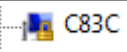


移植模式

运行期间，也可以使用移植模式进行升级。该模式允许网络中同时存在加密和未加密的连接。

在移植模式下，将显示网络中所有已建立加密和未加密连接的计算机。

移植模式只能作为整个工厂过渡到加密通信的一个临时解决方案。

计算机符号

	计算机仅允许加密连接。
	计算机允许加密和未加密的连接。 (移植模式)
	计算机允许未加密的连接。 (移植模式或未加密通信视图)

参见

如何访问子网外的计算机 (页 76)

远程组态 (页 73)

WinCC 中的客户机/服务器系统 (页 16)

典型组态 (页 19)

激活与数据库引擎的加密连接 (<https://docs.microsoft.com/zh-cn/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-2017>)

1.9.3 如何访问子网外的计算机

例如，可通过“Simatic Shell”将网络中位于路由器下游的计算机集成到系统中。

“Simatic Shell”是 WinCC 的一部分，用于集中维护和诊断集成在客户端服务器系统中的所有计算机。

原理

通过“Simatic Shell”中的设置，可以将子网内的计算机指定为“代理”，将来自其它计算机的信息分发到子网内的计算机。

如果启用加密通信，则只有那些在通信前知道指定的共享密钥的计算机才能相互通信。

登录后，系统中的所有参与计算机均可进行通信，甚至可超出网络限制。添加到现有组中的每台计算机均会得知所有计算机的当前状态。

当计算机的状态发生变化时，将给所有参与者发出一条消息，例如：

- 如果计算机激活了一个项目。
- 如果计算机关机。
- 如果计算机启动，并进入了组。

防火墙设置

要允许来自不同网络的 WinCC 计算机相互通信，需要对本地 Windows 防火墙的设置进行如下调整。

对于所有 WinCC 特定的防火墙规则，都需要将范围扩展到其它网络的计算机的 IP 地址或其它网络的完整 IP 范围。

步骤

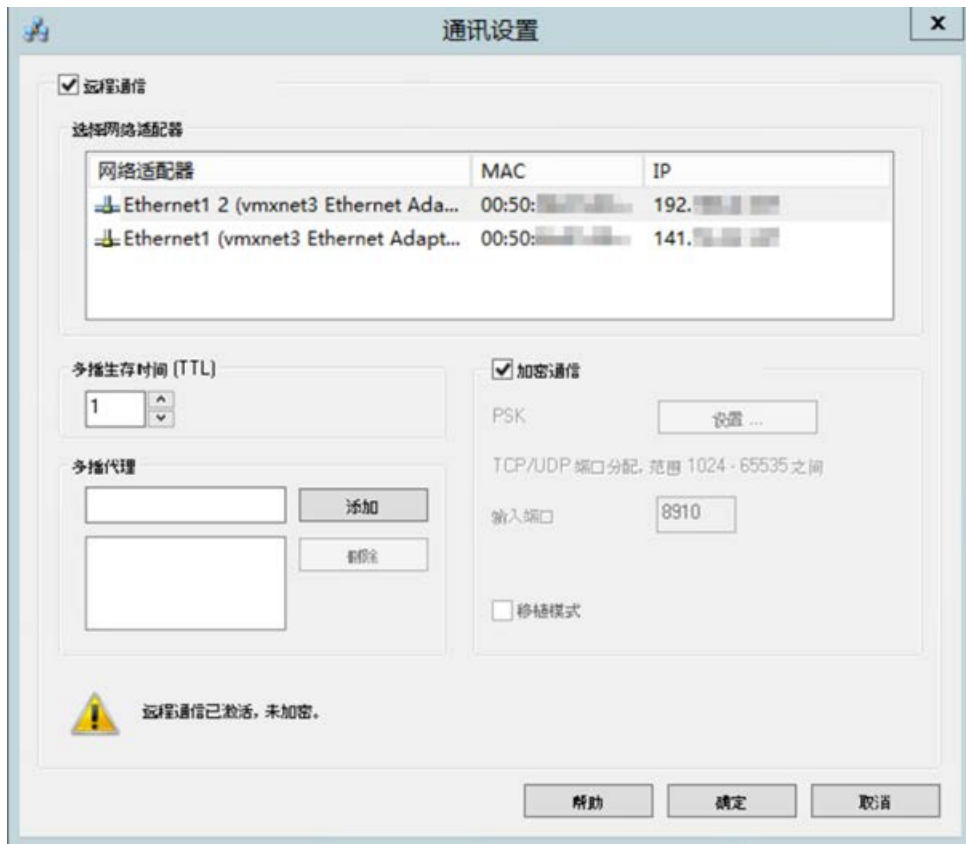
1. 在 Windows 控制面板中打开类别“系统和安全 > Windows 防火墙”(System and Security > Windows Firewall)。
2. 单击“高级设置”(Advanced settings)。将打开“高级安全 Windows 防火墙”(Windows Firewall with Advanced Security) 对话框。
3. 在“进站规则”(Inbound Rules) 下，依次选择所有受影响的防火墙规则，例如 CCAgent、OPC UA Discovery、WinCC ProjectManager。每个规则在“组”列中都有一个以“SIMATIC”开头的组名，例如“SIMATIC Communication Services”、“SIMATIC WinCC OPC”。
4. 通过打开规则快捷菜单打开“属性”(Properties) 对话框。
5. 在“范围”(Scope) 选项卡中，将通信伙伴的 IP 地址或 IP 范围添加到“远程 IP 地址”(Remote IP address) 中。

要求

- “远程通信”(Remote Communication) 选项已启用，网络适配器已组态。

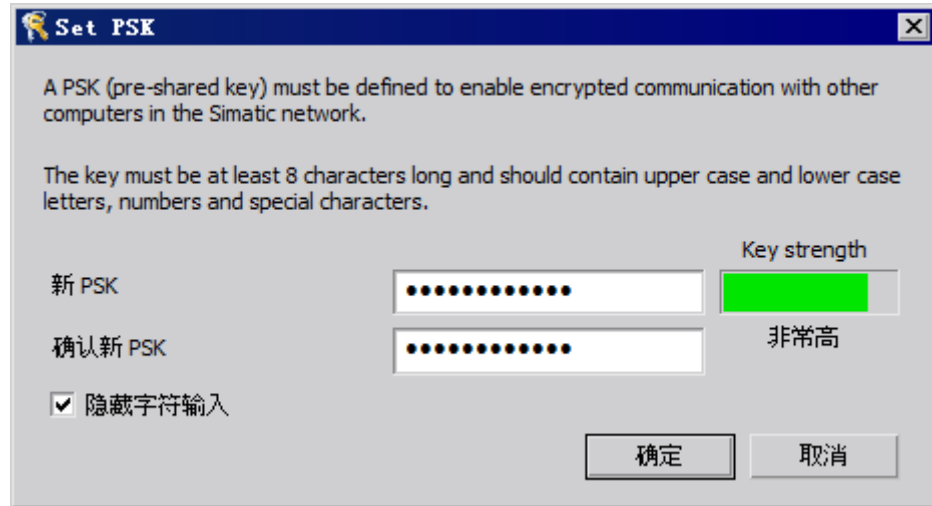
步骤

1. 在作为客户端访问 WinCC 计算机的计算机上打开 Windows 资源管理器。
2. 选择条目“SIMATIC Shell”。
随即打开“Simatic Shell”窗口。
3. 从“SIMATIC Shell”快捷菜单中选择“设置”(Settings) 命令。
将打开“通信设置”(Communication Settings) 对话框。



4. 检查“多点传送寿命 (TTL)”域中的设置。
该数值指定了在各子网之间移动的最大次数 (IP 参数 TTL)。
5. 在“多点传送代理”(Multicast Proxy) 输入字段中, 输入作为子网“(Agent) 代理”的计算机 IP 地址。
它可以是子网中的任何计算机 (客户端或服务器)。
单击“添加”(Add) 按钮将该计算机添加到代理列表中。

- 选择“加密通信”(Encrypted communication) 选项，建立加密通信。
单击“指定”(Specify) 按钮输入 PSK 密钥。



- 输入具有高密钥强度的字符作为密钥。
密钥必须至少为 8 个字符，并且包括大/小写字母、数字和符号。
单击“确定”(OK) 确认设置。
- 如果不想使用分配了默认设置的可用端口，可指定入站端口的分配。
- 要允许加密和未加密的连接同时存在，请选择“移植模式”(Migration mode) 选项。
仅暂时使用此选项，例如，在运行期间进行升级时。
- 使用“确定”按钮确认设置。

参见

- 从多个客户机访问项目 (页 80)
- 如何取消激活项目 (页 86)
- 如何激活项目 (页 85)
- 如何编辑服务器项目画面 (页 84)
- 如何打开一个要进行编辑的项目 (页 82)
- 远程组态 (页 73)
- WinCC 中的客户机/服务器系统 (页 16)
- 加密通信 (页 74)

1.9.4 从多个客户机访问项目

组态选项

在远程操作过程中，根据数据类型，一个或多个客户机可以访问服务器项目。存储在服务器数据库的数据（报警记录、变量记录、变量、用户管理器、文本库）和以文件为基础的数据（画面和图形、报表、脚本）之间存在差别。

说明

多个客户机可同时处理来自服务器数据库的数据。此时，请注意，当多个客户机访问同一数据时，最后一个客户机对数据的修改将被保存。对于来自服务器数据库的数据，各编辑器中的所有数据均被存储，即使只修改了少量数值。

对于文件中存储的数据，当文件打开时，将阻止进一步访问数据。

归档（变量记录）

归档将存储在服务器数据库中。在运行系统中可对变量记录中的数据进行修改。服务器将修改后的数据通知给所有参与的客户机。

画面

画面将以文件的形式存储在服务器或文件服务器上。当客户机访问服务器上的画面时，将阻止其它客户机访问画面。通过不同的客户机可打开不同的项目画面。

画面在运行系统中可进行修改，并且，存储之后，在下次选择画面时是可用的。为了进行编辑，画面也可本地存储，但是必须通过手工执行使其与服务器上的各自画面相匹配。

消息

消息将存储在服务器数据库中。在运行系统中可对报警记录系统中的数据进行修改。服务器将修改后的数据通知给所有参与的客户机。

报表

协议将集中存储在服务器的项目文件夹中。协议数据可分为布局（文件）和打印作业（以项目数据为基础的条目）。只有一个客户端可以组态到各服务器的协议系统。

将要编辑的协议可本地存储，但是必须通过手工执行使其与服务器上的各自协议相匹配。没有必要激活对运行系统中协议系统的修改，因为协议可脱离运行系统单独执行。

脚本

脚本将集中存储在服务器的项目文件夹中。项目专用的脚本可单独在本地计算机上进行定义。脚本将存储在文件中。图形编辑器动作将存储在画面中。编辑期间，将阻止其它客户机访问文件（脚本或画面）。如果不存在与服务器的任何连接，则脚本可本地进行修改，但

必须通过手工使其与服务器上的脚本相匹配。脚本可在运行系统中进行修改。服务器将修改后的脚本通知给所有参与的计算机。

说明

如果不能访问服务器的客户机组态了一个脚本，则脚本将本地存储。如果要使脚本在服务器上可用，则必须手工将脚本复制到相应的服务器文件夹中。

文本库中的文本

文本库中的文本将存储在服务器数据库中。文本对象将单独存储。文本可在运行系统中进行修改。服务器将修改后的脚本通知给所有参与的计算机。在本地计算机上定义的组态语言中将发生更新。

说明

某些 WinCC 编辑器，例如报警记录和用户管理器，可在组态期间对文本库中的同一数据库表进行访问。因此，只能在一台操作员站上同时运行这些编辑器。

变量

变量将存储在服务器数据库中。

说明

如果为了修改变量而取消激活客户机项目，则只有在重新启动所有计算机之后，修改才能生效，且这些计算机上的项目在发生变化时是处于激活状态的。

用户管理器

用户管理器的操作权限将存储在服务器数据库中。用户管理器数据可在运行系统中进行修改。所作的修改不通知参与的计算机。新的数据在客户机再次登录时生效。

参见

如何激活项目 (页 85)

如何取消激活项目 (页 86)

如何编辑服务器项目画面 (页 84)

如何打开一个要进行编辑的项目 (页 82)

如何访问子网外的计算机 (页 76)

远程组态 (页 73)

WinCC 中的客户机/服务器系统 (页 16)

1.9.5 如何打开一个要进行编辑的项目

原理

在项目的组态或运行操作期间，可通过客户端对服务器上的该项目进行编辑。对运行系统中的数据的更新将取决于所组态的数据。

在 Windows 资源管理器中的“Simatic Shell”对话框列出了网络中可用于组态的所有服务器项目的列表。可以获取选定服务器的附加信息，例如，当前激活的模式（组态/运行）。

多个客户端可同时打开和编辑同一项目。

说明

在运行系统中激活服务器项目

如果已通过客户端打开一个要处理的服务器项目，并在 WinCC 中执行了“激活运行系统”(Activate Runtime) 命令，那么必须遵守下列各项：

如果从多用户系统中的一台客户端激活运行系统，即使打开了服务器项目，也只能激活客户端项目。

如果要激活服务器项目，请使用“Simatic Shell”对话框中的“远程激活”(Activate Remote) 命令。

也适用于“取消激活运行系统”(Deactivate Runtime) 命令。

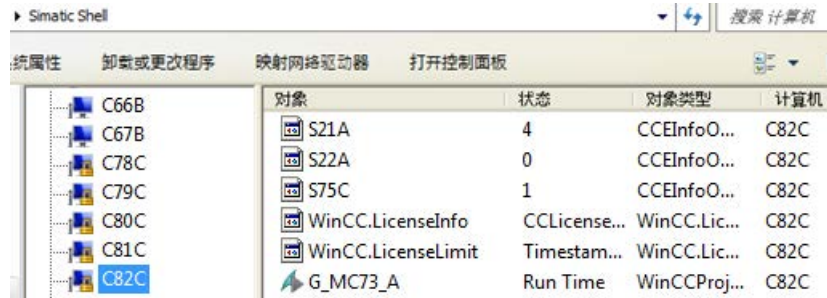
要求

为了在客户端上打开一个服务器项目供远程编辑，必须满足下列条件：

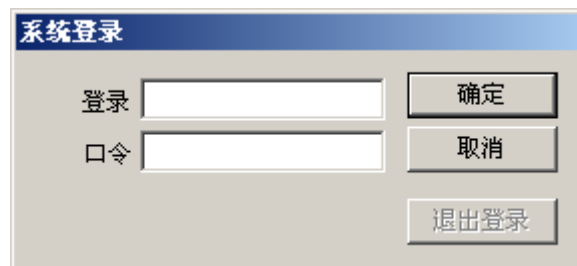
- 在客户端上注册的用户具有服务器 WinCC 项目的“远程组态”操作员授权。
- 客户端必须已经被输入到服务器的计算机列表中。
- 该项目对网络的访问已经激活

步骤

1. 在客户端的 Windows 资源管理器中，选择“Simatic Shell”条目。随即出现“Simatic Shell”窗口。导航窗口中显示所有当前网络中可用的服务器和项目。此外，可以在单独的窗口中打开“Simatic Shell”对话框。在 Windows 资源管理器中，在“Simatic Shell”的弹出式菜单中选择“打开”命令。



2. 选择一台计算机以便显示该计算机特定的项目。
3. 从项目列表中选择要打开的项目，并在快捷菜单中选择“打开”(Open) 命令。将出现登录对话框。



4. 输入当前计算机的用户名和口令。密码区分大小写。
在“WinCC 项目管理器 - 服务器不可用”(WinCC Explorer - Server not available) 对话框中，单击“本地启动服务器”(Start server locally) 按钮。
该项目在客户端上打开，供组态。

参见

- 如何取消激活项目 (页 86)
- 如何激活项目 (页 85)
- 如何编辑服务器项目画面 (页 84)
- 从多个客户机访问项目 (页 80)
- 如何访问子网外的计算机 (页 76)
- 远程组态 (页 73)
- WinCC 中的客户机/服务器系统 (页 16)

1.9.6 如何编辑服务器项目画面

原理

服务器上的画面可在远程客户端上打开、编辑和保存。如果在进行编辑时激活运行系统，则所作修改要在下次画面激活时才有效。

因为每个画面都存储在单个文件中，所以每个画面每次只有一个客户端可进行访问。将阻止其它计算机对该画面的访问。

要求

- 服务器上的项目文件夹必须能够通过网络访问到。
- 在客户端上注册的用户具有服务器 WinCC 项目的“远程组态”操作员授权。

步骤

1. 在客户端的 Windows 资源管理器中，选择“Simatic Shell”条目。随即出现“Simatic Shell”窗口。导航窗口中显示所有当前网络中可用的服务器和项目。
2. 从项目列表中选择要打开的项目，并在快捷菜单中选择“打开”(Open) 命令。将出现登录对话框。输入当前计算机的用户名和口令。
密码区分大小写。
在“WinCC 项目管理器 - 服务器不可用”对话框中，单击“本地启动服务器”按钮。该项目在客户端上打开，供组态。
3. 在客户端的图形编辑器中打开期望的画面。
4. 编辑画面，并将其再次保存到服务器上的项目文件夹中。

参见

如何取消激活项目 (页 86)

如何激活项目 (页 85)

如何打开一个要进行编辑的项目 (页 82)

从多个客户机访问项目 (页 80)

如何访问子网外的计算机 (页 76)

远程组态 (页 73)

WinCC 中的客户机/服务器系统 (页 16)

1.9.7 如何激活项目

原理

客户端/服务器系统不仅提供了远程项目组态的选项，而且也可从远程将其激活和取消激活。

如果通过“Simatic Shell”对话框从客户端上激活服务器项目，也只能激活该服务器项目。然而，若已经打开一个要处理的服务器项目，并通过工具栏中的“启动运行系统”按钮在 WinCC 中激活该项目，即使打开了服务器项目，也只能激活客户端项目。

说明

只有项目位于本地计算机时，才能激活运行系统。

要求

为了在客户端上打开用于远程激活的服务器项目，必须满足下列条件：

- 在客户端上注册的用户具有服务器 WinCC 项目的“远激活”操作员授权。
- 客户端必须已经被输入到服务器的计算机列表中。
- 该项目已启用网络访问功能。

步骤

1. 在客户端的 Windows 资源管理器中，选择“Simatic Shell”条目。随即显示“Simatic Shell”窗口。
将显示网络中所有可用的服务器和项目及其当前状态。
2. 选择将要激活的项目。
3. 从快捷菜单中选择“远程激活”(Activate remote) 命令。
将出现登录对话框。
4. 输入当前计算机的用户名和口令。项目在服务器上被激活。

说明

密码区分大小写。

如何从工程站远程激活 OS 计算机

下列要求通常适用于 OS 项目和远程激活：

- 在 OS 项目的 WinCC 项目中输入 ES 计算机名称作为 OS 服务器。
- 如果要使用另一台计算机远程激活 OS 项目，则此计算机必须作为客户端计算机输入。

1.9 远程组态

因为在 WinCC 项目中服务器和客户端的计算机名称不能相同，所以必须注意下列步骤：

1. 将在计算机列表中“服务器”(Server)下输入的 ES 计算机名称在 WinCC 项目管理器中更改为一个虚构名称。
2. 关闭项目。
3. 打开项目。
4. 在 WinCC 项目管理器的计算机列表中添加新客户端。
5. 在计算机列表中的“客户端”(Client)下输入 ES 计算机名称。
6. 使用 SIMATIC Manager 加载目标系统。
7. 现在可以从工程站远程激活 OS 计算机上的运行系统。

参见

如何取消激活项目 (页 86)

如何编辑服务器项目画面 (页 84)

如何打开一个要进行编辑的项目 (页 82)

从多个客户机访问项目 (页 80)

如何访问子网外的计算机 (页 76)

远程组态 (页 73)

WinCC 中的客户机/服务器系统 (页 16)

1.9.8 如何取消激活项目

原理

客户端/服务器系统不仅提供了远程项目组态的选项，而且也可从远程将其激活和取消激活。

如果通过“Simatic Shell”对话框从客户端上取消激活一个服务器项目，也只能取消激活该服务器项目。然而，若已经打开一个要处理的服务器项目，并通过工具栏中的“停止运行系统”按钮在 WinCC 中取消激活该项目，即使打开了服务器项目，也只能取消激活该客户端项目。

要求

为了在客户端上打开用于远程取消激活的服务器项目，必须满足下列条件：

- 在客户端上注册的用户具有服务器 WinCC 项目的“远激激活”操作员授权。
- 客户端必须已经被输入到服务器的计算机列表中。
- 该项目对网络的访问已经激活

步骤

1. 在客户端的 Windows 资源管理器中，选择“Simatic Shell”条目。随即显示“Simatic Shell”窗口。
可在“Simatic Shell”对话框中查看通过网络获得的客户端/服务器系统中启用的服务器和项目。
2. 选择将要激活的项目。
3. 从弹出式菜单中选择“远程取消激活”命令。将出现登录对话框。
4. 输入当前计算机的用户名和口令。项目在服务器上被取消激活。

说明

密码区分大小写。

参见

如何激活项目 (页 85)

如何编辑服务器项目画面 (页 84)

如何打开一个要进行编辑的项目 (页 82)

从多个客户机访问项目 (页 80)

如何访问子网外的计算机 (页 76)

远程组态 (页 73)

WinCC 中的客户机/服务器系统 (页 16)

1.10 使用客户机/服务器系统中的 OPC 接口

原理

OPC（用于过程控制的 OLE）是一种用于自动化工业领域内组件的全球性通信标准。

基于 Windows 技术而开发的 OPC 提供了一种开放式接口，通过它可在 PLC、操作和监控系统以及来自不同厂商的办公应用程序之间进行标准化的数据交换，而不会有任何问题。

说明

自动化工业领域的主要公司均在“OPC 基金会”内进行合作。

有关 OPC 基金会的更多信息，请访问如下 Internet 地址：[“http://www.opcfoundation.org”](http://www.opcfoundation.org)

在 WinCC 中使用 OPC

在分布式系统内使用 OPC 后，每台 WinCC 服务器均可监控整个系统。然而，WinCC 服务器只承担某个指定范围的任务，例如消息编辑或归档。

WinCC OPC 服务器允许 OPC 通过软件接口访问 WinCC 运行系统的数据。WinCC OPC 服务器支持符合相应 OPC 规范的全部功能。

至于 OPC 客户端，任何基于相应 OPC 规范的软件都可运行。这样，比方说，就可使用 OPC 客户端来分析各种源。通过创建自开发的 OPC 客户端，就可满足最高标准的要求。

为了以 WinCC OPC 服务器模式运行，必须在将要用作 WinCC OPC 服务器的那台计算机上安装“数据连接性包”许可证。OPC DA 服务器不需要“数据连接性包”许可证。

随同 WinCC 安装，已经把 OPC 接口安装在客户端和服务器上。

WinCC 的 OPC 服务器支持下列规范：

- OPC Data Access 2.05a und 3.00
- OPC XML Data Access 1.01
- OPC Historical Data Access 1.20
- OPC Alarm & Events 1.10
- OPC UA 1.03

有关在 WinCC 中使用 OPC 接口的详细信息，请参见 WinCC 信息系统中的“通信” (Communication)。

参见

WinCC 中的客户机/服务器系统 (页 16)

OPC 的功能 (页 636)

1.10 使用客户机服务器系统中的 OPC 接口

文件服务器

2.1 安装文件服务器

简介

WinCC 文件服务器是具有最小 WinCC 组件组态的服务器。

可在文件服务器上保存项目并对其进行集中管理。

这样可以更方便地定期创建所有项目的备份副本。

说明

仅用于组态

单独使用文件服务器进行组态。

要求

“安装注意事项”中介绍的要求适用于安装 WinCC Fileserver V7。

还需要满足以下条件：

- 计算机必须在网络 (LAN) 中可用。
- 如果要使用文件服务器，则需要管理员权限。

说明

未安装基本 WinCC 的 PC

不能在一台计算机上同时安装 WinCC V7 和 WinCC Fileserver V7。

安装

要将计算机设置为文件服务器，需在计算机上运行文件服务器安装程序。

1. 启动 WinCC 安装 DVD。
2. 选择“自定义安装”(Custom installation) 安装类型。
3. 在“程序”(Programs) 对话框的“WinCC”组中选择“WinCC Fileserver”条目。

WinCC 的最小安装版本将安装到计算机上。

2.1 安装文件服务器

组态

项目存储在文件服务器上。

允许访问

如果要允许所有的项目成员访问项目，则必须共享文件服务器上的相应驱动器或文件夹。

要共享文件夹或驱动器，必须具有 Windows 管理员权限。

在组态计算机上为共享文件夹或驱动器分配唯一的驱动器盘符。

随后，项目成员便可像打开本地项目一样去打开文件服务器上的项目。

WinCC 服务模式

3.1 WinCC 服务模式：标准项目和服务项目

WinCC 服务模式作为服务提供操作 WinCC 运行系统的选项。

没有交互式用户登录到计算机时，还可以将 WinCC Runtime 作为服务激活。

该部分说明了：

- WinCC 服务模式可用于哪些组态
- 如何将一个项目组态为服务项目
- 如何激活服务项目

概述

可将 WinCC 项目组态为标准项目或服务项目。

要在 WinCC 服务模式中操作 WinCC 项目，必须将其组态为服务项目。

标准项目

为了运行 WinCC Runtime，用户必须登录到计算机。

可进行交互式用户输入。

服务项目

没有交互式用户登录到计算机时，也可以在计算机上运行 WinCC Runtime。

3.1 WinCC 服务模式：标准项目和服务项目

WinCC Runtime 也可由已登录用户进行操作。在这种情况下，可进行交互式用户输入。

说明

当系统正在被访问时，WinCC 不可执行

不允许在控制面板和 Windows 任务管理器中更改 WinCC 的过程和服务。

以下更改将受到影响：

- 对属性的更改
- 手动访问：
 - 启动
 - 退出
 - 停止
 - 恢复
 - 重新启动
- 优先级更改

在各个进程和服务之间存在依存关系。

请勿进行任何更改。

参见

使用服务项目及限制 (页 96)

服务项目的操作模式 (页 101)

如何激活服务项目 (页 107)

3.2 服务项目的组态

概述

在下列组态中，WinCC Runtime 可在服务器上作为服务项目运行：

- 装有 Windows Server 操作系统的 WinCC 服务器
具有自带项目的 WinCC 客户端
无自带项目的 WinCC 客户端
- WinCC WebNavigator 服务器或专用 Web 服务器
WinCC Web 客户端
- DataMonitor 服务器或专用 DataMonitor 服务器
DataMonitor 客户端

3.3 使用服务项目及限制

使用

在服务器上，WinCC 服务模式中的项目将起到服务项目的作用。

WinCC Runtime 作为服务启动。

可自动或手动启动服务项目。

无登录用户时的操作

服务项目可以在没有交互式用户登录到计算机的情况下运行。

如果没有交互式用户登录，则不能进行交互式操作。

有登录用户时的操作

在服务项目中通常不需要交互式操作。

但交互式用户可以出于维护等目的而登录。在这种情况下，用户可激活服务项目的交互式操作。

自动启动

使用自动启动时，开启服务器并激活已设置项目后，WinCC Runtime 将自动启动。

自动启动可以在没有交互式用户登录的情况下执行。

手动启动

使用手动启动时，用户必须登录到服务器，然后再激活项目。

用户注销服务器登录后，WinCC Runtime 将继续保持激活状态。

用户登录与注销

当服务项目激活时，交互式用户可随时登录和注销服务器。

限制

服务项目受到以下限制：

脚本

因为交互式用户通常不登录到服务项目，所以，如 C 脚本和 VB 脚本在下列情况下会出现问题：

- 脚本要求交互（如输入）的情况。
- 脚本打开消息框。

在服务模式下，没有可用于 C 脚本的公共数据区域。

因此，不能在“全局脚本”和“图形编辑器”之间交换全局 C 变量。

附加程序或任务

对于服务项目，不能将附加程序和任务添加到启动列表。

未发布的组件

未对服务项目发布通过互连站的 OPC 访问。

服务项目的诊断信息

通常，用户不会登录到具有已激活服务项目的服务器。

WinCC 无法在服务器上显示诊断信息。因此，WinCC 会将诊断信息转发给客户端。

更多相关信息，请参见《WinCC 信息系统》中的“使用 WinCC > 使用项目 > 附录 > WinCC 诊断窗口和许可证信息”。

说明

编辑或移植服务项目

要编辑或移植服务项目，需要在计算机上将 ServiceMode 用户设置为管理员。

如果 ServiceMode 用户不可用，必须相应地将登录的 Windows 用户设置为管理员，才能编辑或移植项目。

参见

如何组态服务项目的自动启动 (页 105)

WinCC 服务模式：标准项目和服务项目 (页 93)

3.4 系统托盘中的 WinCC 状态和控件

简介

WinCC 在任务栏通知区（即所谓的托盘区）中显示“SIMATIC WinCC”符号。

此符号提供了有关项目状态的信息。

WinCC 项目可以通过该符号的快捷菜单激活和禁用。

项目状态

下表显示了与各种项目状态一起出现的“SIMATIC WinCC”符号：

图标	状态
	<ul style="list-style-type: none"> WinCC 未激活。 未打开项目。
	WinCC 更改状态： <ul style="list-style-type: none"> WinCC 打开项目。 WinCC 激活项目。 WinCC 禁用项目。 WinCC 关闭项目。
	项目打开。
	项目被激活。
	项目激活，服务器为“故障”状态。

通过弹出式菜单出现的控制选项

“SIMATIC WinCC”符号的快捷菜单提供了以下功能：

- 启用运行系统
- 结束图形运行系统
- 禁用运行系统
- 关闭项目
- 运行系统启动选项（自动启动组态）
- 打开诊断窗口
- WinCC 许可证分析

SIMATIC WinCC® 窗口

要打开“SIMATIC WinCC”窗口，需单击“SIMATIC WinCC”符号。

示例：激活了运行系统的窗口



该窗口显示下列信息：



- 项目名称
- 项目类型
- 项目状态
- 计算机列表
本地计算机以蓝色显示。

计算机列表



该计算机列表包含了网络中的所有计算机。

如果项目为激活状态，将显示所有现有计算机的连接状态。

下表显示了连接状态的图标及其含义：

图标	状态
	<ul style="list-style-type: none"> • 无连接 • 连接已断开
	<ul style="list-style-type: none"> • 本地计算机 • 冗余伙伴服务器

3.4 系统托盘中的 WinCC 状态和控件

图标	状态
	连接 <ul style="list-style-type: none"> • 与备用服务器 • 与主服务器，但备用服务器为首选服务器
	连接 <ul style="list-style-type: none"> • 与主服务器 • 与作为首选服务器的备用服务器

此视图仅提供有关网络中的 PC 的状态信息。

要查询控制器的连接状态，请使用 WinCC 项目管理器中的“驱动程序连接的状态”(Status of driver connections) 功能或使用系统变量“@<Connection name>@ConnectionStateEx”。

参见

如何激活服务项目 (页 107)

3.5 功能和先决条件

3.5.1 服务项目的操作模式

简介

本章介绍 WinCC 服务项目的操作模式。

标准项目

标准项目启动方式如下：

- 用户登录到系统。
- 用户启动 WinCC 运行系统或 WinCC 运行系统自动启动。

WinCC 运行系统将保持激活状态，直到出现下列情况之一：

- 用户退出 WinCC 运行系统。
- 用户从系统注销。
在这种情况下，系统将终止 WinCC 运行系统。

服务项目

对于服务项目，WinCC 运行系统将作为服务启动。根据设置，这些服务将在以下时间启动：

- 在操作系统启动后自动启动。
- 在用户登录并启动 WinCC 运行系统后。

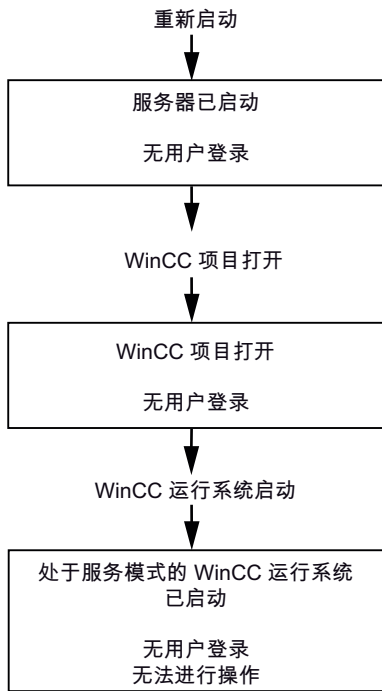
即使用户再次注销，WinCC 也会保持活动状态。

WinCC 运行系统数据仍可访问。

已登录用户可根据需要激活运行系统操作。

下图显示了启动服务器与通过服务项目自动启动运行系统之间的状态。

3.5 功能和先决条件



参见

WinCC 服务模式：标准项目和服务项目 (页 93)

如何激活服务项目 (页 107)

3.5.2 运行服务项目的要求

在服务项目中通常不需要交互式操作。

脚本

因为交互式用户通常不登录到服务项目，所以，如 C 脚本和 VB 脚本在下列情况下会出现问题：

- 脚本要求交互（如输入）的情况。
- 脚本打开消息框。

分布式 WinCC 方案中的服务项目

必须为服务项目设置专门的 Windows 用户。

为服务项目组态的 Windows 用户必须从属于“SIMATIC HMI”用户组。

对于可进行服务器间通信的 WinCC 多用户系统或分布式系统，可使用本地 Windows 用户或 Windows 域用户。

本地 Windows 用户

- 该用户必须是网络中所有计算机上的本地“SIMATIC HMI”用户组的成员。
- 该用户的密码在所有计算机上必须都相同。

Windows 域用户

- 该用户是所有计算机上的本地“SIMATIC HMI”用户组的成员。
- 或者，该用户是本地“SIMATIC HMI”用户组的成员组的成员。

密码

实现 WinCC 服务项目不中断运行操作的要求：

- 不能更改已组态用户的密码。
- 已组态用户的密码不能过期。

为确保这一点，请在设置用户时激活下列选项：

- “用户不能更改密码”
- “密码终身有效”

参见

如何组态服务项目的自动启动 (页 105)

3.6 组态 WinCC 服务模式

3.6.1 如何将一个项目定义为服务项目

简介

在项目属性中指定该项目是作为标准项目运行还是作为服务项目运行。



步骤 - 定义服务项目

1. 在 WinCC 项目管理器的导航窗口中选择项目名称。
2. 在项目快捷菜单中选择“属性”(Properties) 条目。
将打开“项目特性”(Project Properties) 对话框。
3. 切换到“运行模式”选项卡。
4. 启用“服务”选项。
WinCC 将显示消息，提示为转换项目需要重新加载项目。
5. 在“用户”域中输入用户，WinCC 服务项目将以用户名来运行。
有关该用户所需属性的详细信息，请参见：“运行服务项目的要求 (页 102)”。
6. 在“密码”域中输入相应的密码。
7. 确认“密码”域中的密码。
8. 单击“确定”按钮确认输入。
9. 重新加载项目。

将服务项目转换为标准项目

如果要将服务项目转换为标准项目，请选中“运行模式”选项卡上的“标准”选项。

将标准项目转换为服务项目

如果要将标准项目转换为服务项目，请选中“操作模式”选项卡上的“服务”选项。

说明

通常无法将标准项目转换为服务项目

服务项目受到一些限制。在执行转换之前请注意这些限制。

更多相关信息，请参见“使用服务项目及限制 (页 96)”。

参见

如何组态服务项目的自动启动 (页 105)

3.6.2 如何组态服务项目的自动启动

设置自动启动

计算机启动时，WinCC 会激活所选 WinCC 项目。

在“自动启动组态”(AutoStart configuration)对话框中，选择所需服务项目作为自动启动项目。

说明


服务重启时激活项目

“SIMATIC WinCC CCProjectMgr”服务重新启动时，也会重新激活自动启动项目。

更改工作模式：再次设置自动启动

将服务项目转换为标准项目后必须重新组态“自动启动”，反之亦然。

步骤 - 设置自动启动

1. 在“Siemens Automation”Windows 程序组中选择“自动启动”(AutoStart) 条目。
将打开“自动启动组态”(AutoStart Configuration) 对话框。
显示本地计算机的设置。
2. 选择所需计算机。
存在以下选项：
 - 输入计算机名称。
 - 通过“...”选择网络路径中的计算机。
 - 要为本地计算机设置自动启动，请单击“本地计算机”(Local Computer)。
要显示已选计算机的当前组态信息，请单击“读取组态”(Read configuration)。
3. 单击“项目”(Project) 字段中的  按钮，选择服务项目。
项目文件及其完整路径将输入框中。
项目类型将显示在路径下。
4. 组态自动启动行为的设置。
5. 激活“自动启动激活”(Autostart active) 选项。
如果未激活该选项，则已组态计算机无法执行自动启动操作。
6. 单击“应用”(Apply) 确认设置，单击“确定”(OK) 关闭设置。

结果

下一次启动计算机时，WinCC 将自动启动，并打开所选项目。

参见


如何将一个项目定义为服务项目 (页 104)

使用服务项目及限制 (页 96)

3.7 运行系统中的服务项目

3.7.1 如何激活服务项目

简介

WinCC 在任务栏通知区（即所谓的系统托盘区）中创建“SIMATIC WinCC”图标：

可使用该图标的快捷菜单执行下列功能，包括：

- 启用运行系统。
- 结束图形运行系统。
- 禁用运行系统。

有关软件的更新和升级信息，敬请访问“系统托盘中的 WinCC 状态和控件 (页 98)”。

要求

必须将项目保存为服务项目。

更多相关要求，请参见“运行服务项目的要求 (页 102)”。

WinCC 服务模式下的自动启动

如果使用“自动启动组态”工具为项目正确组态了自动启动，则会执行以下操作：

- 启动服务器后，立即自动激活项目。

不需要用户输入。

结果

项目已激活。


没有用户登录到服务器。

WinCC 服务模式下的自动启动

下列步骤假设没有为项目组态自动启动。


1. 启动服务器。
2. 登录到服务器。

3.7 运行系统中的服务项目

3. 打开项目。
4. 在系统托盘  图标的快捷菜单中选择“激活运行系统”(Activate Runtime) 命令。或者，用 WinCC 项目管理器激活项目。

结果

项目被激活。

WinCC 显示图标 .

为确保当用户从服务器注销时 WinCC Runtime 仍保持激活状态，仅退出 WinCC 项目管理器。为此，在对话框“退出 WinCC 项目管理器”中选择“退出 WinCC 项目管理器”条目。

这样 WinCC Runtime 将保持激活状态。

参见

运行服务项目的要求 (页 102)

WinCC 服务模式：标准项目和服务项目 (页 93)

服务项目的操作模式 (页 101)

系统托盘中的 WinCC 状态和控件 (页 98)

3.7.2 如何登录与注销已激活的服务项目

简介

当 WinCC 项目处于运行系统中时，为了在服务器上执行必要的工作，您可以登录到服务然后再注销。

说明

如果安装了需要重新启动的更新，则会结束 WinCC 运行系统。

要求

服务项目已激活。没有用户登录到服务器。

步骤

1. 登录到服务器。
2. 执行所需的操作。
3. 注销服务器。

结果

登录到服务器然后注销。 WinCC 运行系统不受到影响。

3.7.3 如何激活出于维护目的的交互式操作

简介


当 WinCC 服务项目处于运行系统中时，可激活交互式操作。

要求


服务项目处于激活状态。 交互式操作未激活。

作为“SIMATIC HMI”组成员的用户登录。

步骤 - 激活交互式操作

1. 登录到服务器。
2. 从托盘区域  图标的弹出式菜单中选择“启动图形运行系统”命令。
WinCC 将开启交互式操作。 用户可操作 WinCC 项目。

步骤 - 结束交互式操作

1. 从托盘区域  图标的弹出式菜单中选择“结束图形运行系统”命令。
WinCC 将结束图形运行系统。
2. 必要时请注销。

冗余系统

4.1 冗余

内容

WinCC 选件“WinCC/Redundancy”用于组态冗余系统。通过并行运行两个互连的服务器并在发生故障时自动切换服务器，可增强 WinCC 和系统的可用性。

概述

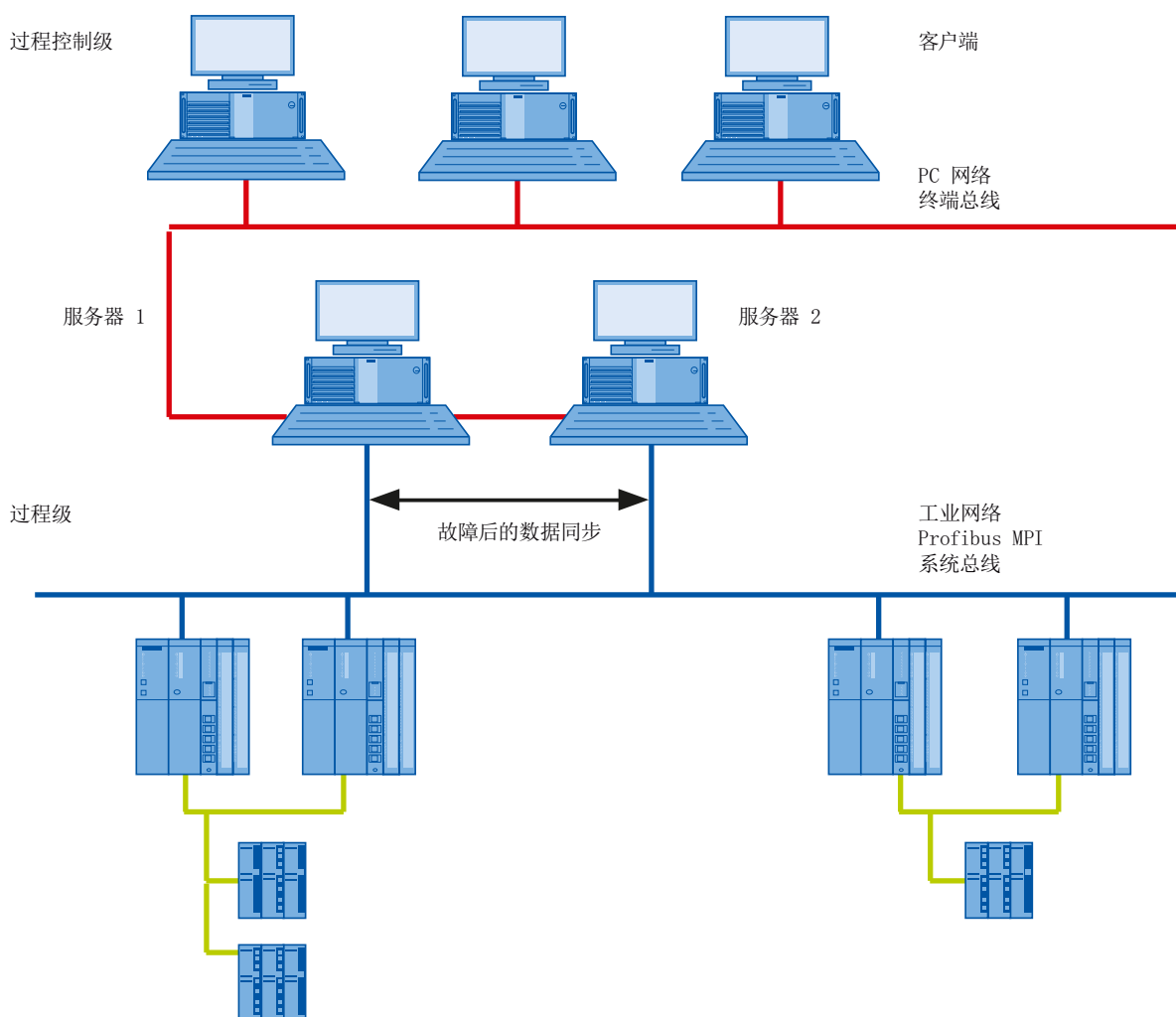
本文档介绍了以下内容：

- 冗余系统的要求。
- 如何在 WinCC 中创建冗余系统。
- 如何组态冗余服务器。
- 如何组态冗余归档的同步。

4.2 WinCC 冗余

简介

冗余 WinCC 项目包括两个 WinCC 服务器，并将这两个服务器组态为执行相同的功能且并行运行：一个主服务器和一个备用服务器。这两台服务器相互连接，并都与自动化系统和客户端相连。



WinCC Redundancy 功能概览

WinCC Redundancy 具有下列功能：

- 当服务器或过程连接出现故障时，自动切换客户端。
- 故障服务器恢复后或过程连接故障消除后，自动同步消息归档、过程值归档和用户归档。
- 在线同步内部消息。
- 在线同步支持变量同步的内部变量。
- 在线同步用户归档。
- 用于将项目复制到冗余服务器的“项目复制器”。
- 监视 WinCC 应用程序的“应用程序正常检查”功能。
- 用于监视本地系统硬件和软件的“SelfDiagnosis”功能。

“应用程序正常检查”功能

“应用程序正常检查”功能会自动监视所有重要的 WinCC 应用程序。

设备状态监视在检测到软件错误后将执行下列操作：

- 在“@RedundantServerState”系统变量中将服务器状态设为“故障”。
- 授权已连接的客户端切换到冗余服务器。
- 通过过程控制消息通知用户软件出错。如果故障由报警服务器引发，则无法触发过程控制消息。

说明

如果“应用程序正常检查”功能检测到软件出错，且已启动客户端切换，则必须重新启动相关的服务器。只有如此，才能将客户端重新连接到服务器。对归档进行可追溯性同步时，仅追溯到检测到软件错误的时刻。

“SelfDiagnosis”功能

“SelfDiagnosis”功能由以下任务构成，用于确保冗余系统的可用性和稳定性：

- 监视和报告本地硬件和软件问题
- 监视本地系统性能
- 监视数据量的状态
- 服务器故障切换（如有必要）

4.2 WinCC 冗余

出现故障时执行以下任务：

- 重启应用程序
- 如有必要，将服务器设置为“故障”状态，服务器将发生切换。
- 将生成日志条目。
- 将触发系统报警。

4.3 冗余系统的要求

概述

对于 WinCC 冗余，必须满足下列先决条件：

- 对于带有多用户操作的冗余 WinCC 服务器，只能使用装有服务器操作系统的计算机。
- 两台服务器上都必须安装 WinCC Redundancy 选件。冗余服务器上必须安装 WinCC Redundancy 许可证。
- 必须为两台冗余服务器组态完全相同的功能。
- 除这两个冗余服务器外，不得将任何其它 PC 组态为冗余服务器。
- 这两个服务器的时间必须同步。建议整个系统都采用时间同步。可使用 WinCC 中的“时间同步”选项组态时间同步。
- 来自自动化系统和客户端的消息和确认必须始终在帧中包含时间戳（按时间顺序发送消息）。这样可以避免条目重复。例如，使用自动化系统中的报警块。
- 必须将来自底层自动化系统的过程值、消息和主动消息块同时发送至两个服务器。
- 冗余服务器之间必须存在以下附加连接之一：
 - 网络适配器
 - 串行连接

此附加连接可确保准确地定义“主机”或“备用机”状态。

在 WinCC 项目管理器中使用“冗余”编辑器通过网卡组态附加连接。使用带有相应 IP 地址的 TCP/IP 协议。IP 地址不能与终端总线存在于相同的子网中。

说明

WinCC 调试过程中激活 WinCC Redundancy 后的运行系统性能

调试过程中，经常会激活和禁用服务器计算机上的 WinCC 运行系统。这种在激活 WinCC Redundancy 时的重复启动每次都会导致归档同步，从而可能导致 WinCC 运行系统的性能明显下降。因此，我们建议您在调试过程中禁用 WinCC Redundancy。

不间断电源

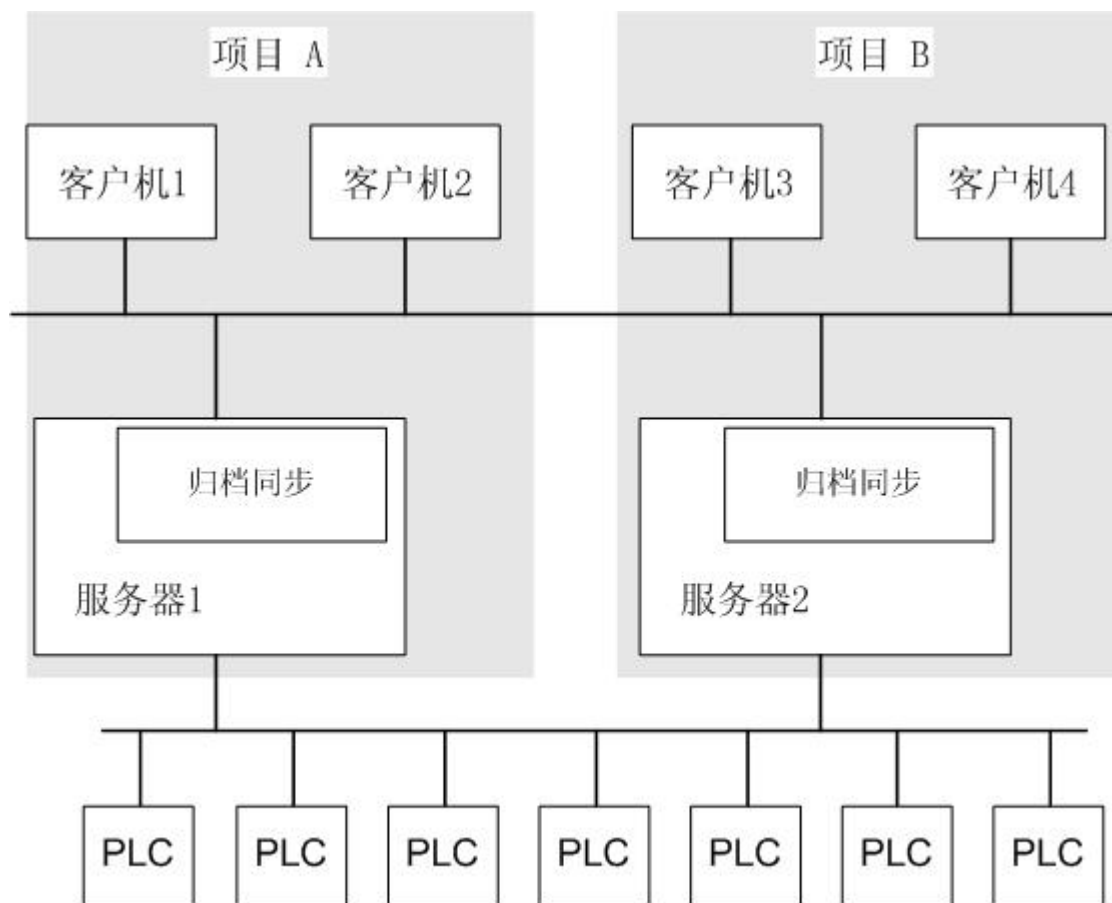
为了在电源出现故障时安全退出 WinCC，建议使用不间断电源 (UPS)。

4.4 冗余如何工作

简介

两台服务器具有相同的权限，并可互相独立地进行工作。两者对用户都可用。如果其中一台服务器出现故障，将始终有一台相同的冗余服务器可用。

下图显示了并行冗余服务器的归档和归档同步。



冗余服务器的标识

两个服务器中的一个已组态为默认主机。

在该服务器的运行系统中，系统变量“@RM_MASTER”设为“1”。如果变量的状态发生变化（例如，由于计算机发生故障），客户端将切换至“备用”计算机。该计算机现在为主服务器计算机。

服务器将在运行系统中互相监视，以便及早检测到出现故障的伙伴服务器。

使用通过网络适配器实现的附加连接或服务器间的串行连接来监视状态。该连接可改善冗余伙伴服务器之间的通信，从而提高冗余的可用性。附加连接不能用于同步归档。

正常运行期间的 WinCC 归档

运行系统中两台服务器通常完全同步。每台服务器均有其自己的过程驱动程序连接，并有其自己的数据归档。

自动化系统将过程数据和消息发送到两台冗余服务器，再由这两台服务器进行相应的处理。

用户归档、内部消息和内部变量可在线连续同步。

两个服务器利用 TCP/IP 协议通过 LAN 进行通信来同步归档。

服务器故障

如果其中一台服务器出现故障，客户端将自动从故障服务器切换到冗余伙伴服务器，这样可确保所有的客户端都始终可用于对过程进行监视和操作。

当出现故障时，处于活动状态的服务器将继续对 WinCC 项目的所有消息和过程数据进行归档。

在故障服务器恢复在线状态后，所有消息归档、过程值归档和用户归档的内容都将自动复制到已恢复的服务器。这将填充故障服务器的归档数据间隙。

说明

冗余故障：至少 69 秒。

由于技术原因，停机必须至少持续 69 秒，两个系统才能自动同步。

触发客户机切换的因素

在服务器出现故障期间，系统将自动执行从默认（主）服务器到冗余服务器的客户机切换。

下列因素会引起服务器的切换：

- 与服务器的网络连接出现故障
- 服务器故障
- 过程连接故障
- “应用程序正常检查”功能已检测到出现故障的 WinCC 应用程序，并触发切换。
- 项目被禁用。

如果激活了系统连接中发生故障时的客户端切换冗余选项，将周期性地确定与“主”服务器和冗余伙伴服务器的故障逻辑连接的数目。

4.4 冗余如何工作

如果主服务器上的故障逻辑连接比冗余伙伴服务器上的多，则已登录到主服务器上的客户端将切换到冗余伙伴服务器。

消除过程连接的错误后，客户端将重新切换到其最初连接的首选服务器。

只有两台冗余服务器都在运行时，才启动过程连接的监视。

说明

服务器上的软件错误

如果服务器上的软件出现错误，则所连接的客户端可能不能切换到冗余伙伴服务器，而系统被锁定。

服务器恢复后触发归档同步的因素

在下列错误纠正后将启动服务器之间的归档同步：

- 过程连接出错。但是，可以禁用过程连接监视。
- 与伙伴服务器的网络连接出现故障。
- 服务器故障。
- 项目未激活。

服务器恢复后同步

故障服务器恢复后，WinCC Redundancy 会将丢失的数据传送给故障服务器。

这适用于消息归档、过程值归档、用户归档和内部变量。

对于消息归档和过程值归档，会传送在停机期间处于激活状态的所有分段。传送后，两个等效服务器将再次可用。

归档同步将作为后台功能来实现，并与 WinCC 的过程管理和归档并行运行。因此，可始终保证对系统的操作和观察。

比较内部变量

内部变量必须具有属性“变量同步”。

一旦在其中一台冗余服务器上修改了一个内部变量，就会在伙伴计算机上比较该变量。

内部变量还包括名称以字符“@”开头的系统变量，如“@RM_Master”。不能组态系统变量的在线同步。

过程连接出错后的同步

如果已激活过程连接监视，则在排除服务器与自动化系统之间的故障后，自动启动所有归档的同步。

如果过程连接监视已经激活，则相应的服务器将对组态的所有连接进行设备状态监视。当寻址的自动化系统未能成功地将确认信息发送回服务器时，服务器会检测到服务器与自动化系统之间的过程连接出现故障。

如果一个或多个自动化系统出现网络故障，将对属于该项目的所有自动化系统归档执行同步。未出现故障的自动化系统归档也要进行同步。如果禁用此选项，可降低服务器的运行系统负载。

因为禁用网络连接监视后未检测到自动化系统的网络中存在错误，所以不会进行归档同步。

在线同步

服务器到服务器的直接同步：

- 支持以下内容的报警记录：
 - 内部消息变量
 - 不带变量连接的消息
 - 系统运行消息
 - “批生产”消息
- 受到用户归档的支持
- 受到带变量同步的内部变量的支持

比较已阻塞消息

发生故障的服务器恢复后，将通过自动化系统的常规查询搜索并同步当前被阻塞的消息。

如果消息仅在一台服务器上被动阻塞，则同步该阻塞信息。

4.5 组态冗余系统

4.5.1 冗余系统安装指南

简介

此处，您可大致了解如何设置冗余 WinCC 系统。有关客户端服务器系统结构的常规信息，请参见 WinCC 信息系统中标题为“分布式系统”的部分。

在 Windows 中输入服务器

两台冗余服务器必须能够在网络中相互识别。此外，冗余服务器上的用户/密码必须完全相同。必须为用户设置管理员或用户权限。用户必须是“SIMATIC HMI”用户组的成员。

组态服务器上的项目

WinCC 冗余的组态期间将确定以下内容：

- 标准主站。
- 伙伴服务器。
- 客户端的切换行为。
- 归档同步的类型。

复制项目前，使用 WinCC 项目管理器中的“服务器数据”编辑器创建服务器数据包。最好在标准服务器上创建服务器软件包。

说明

只组态确实需要的用户归档的同步。要同步的用户归档数越大，同步过程需要的时间越长，系统负载就越大。

复制 WinCC 项目

要使冗余伙伴服务器上具有功能相同的 WinCC 项目，需使用“项目复制器”从默认服务器中复制项目。主服务器和备用服务器将具有相同的项目设置。

说明

在复制之前，请确保计算机上有足够的内存可供复制项目。如果正在复制现有项目，则不能打开该项目。

组态备用服务器

要监视冗余的状态，仍然需要通过“冗余”编辑器在备用服务器上设置与主服务器的附加连接。

组态客户端

要在客户端上使用 WinCC Redundancy，请按以下步骤在“服务器数据”编辑器中进行操作：

- 创建默认服务器的数据包
- 设置首选服务器，并激活数据包的自动更新。

激活冗余服务器

1. 首先激活已组态的主站服务器。
2. 下一步，启动相连的客户端。
3. 客户端激活后，立即激活第二台服务器及其相连的客户端。

现在可执行首次同步。该同步所产生的停止时间包括激活第一个服务器和第二个服务器之间的时间间隔。

说明

请注意，在启动冗余服务器时，必须在激活冗余伙伴之前完成第一个服务器的启动。在服务器初始启动时，不必激活客户端。

一旦完全禁用一个冗余服务器对，在重新激活时必须遵循指定的顺序。第一个激活的服务器为最后一个禁用的服务器。当该服务器完全启动后，才可激活其冗余伙伴。

禁用冗余服务器

请注意，在禁用冗余服务器之前，第二台服务器必须能够正常、无错地工作。

4.5 组态冗余系统

禁用前，必须按照相应过程控制消息的指示完成归档同步。

说明

如果在第一个服务器的存档同步完成之前禁用第二个服务器，则可能会出现数据丢失。对于调试过程中频繁激活/禁用服务器的情况，该操作尤为重要。

参见

如何组态冗余服务器 (页 123)

如何组态用户归档的同步 (页 125)

4.5.2 组态相同功能

过程数据归档和消息归档

必须通过在功能上完全相同的方式为冗余服务器组态变量记录和报警记录。需要完全相同的归档，可借此以附加测量点或归档的形式添加。同步时不会同步扩展内容。必须自己调整伙伴服务器上的扩展内容。

WinCC 对以下基于硬盘的归档进行同步：

- 过程值归档
- 压缩归档
- 消息归档

不执行主存储器归档的同步。

用户归档

用户归档要求两台服务器上的结构相同：

对于将要进行同步的用户归档的组态，其属性以及域和记录结构必须相同。

说明

无法通过“加载在线更改”同步已更改的组态数据

用户管理（用户管理器）

用户管理中的更改不会自动同步。这也适用于运行系统中通过 WinCC UserAdminControl 的组态。

如果要组态对用户管理的更改，可选择以下几种操作：

- 在工程师站上组态更改。将更改传送到冗余服务器。
- 在两台冗余服务器上对更改进行相同的组态。

参见

WinCC 冗余 (页 112)

4.5.3 如何组态冗余服务器

简介

使用 WinCC 项目管理器中的“冗余”编辑器来组态冗余服务器和归档同步。

要求

- 必须为两台冗余服务器组态完全相同的功能。

步骤

1. 在 WinCC 项目管理器中打开“冗余”(Redundancy) 编辑器。
转到“常规”(General) 选项卡。
“服务器”(Server) 字段中包含组态了 WinCC Redundancy 的计算机的名称。

The screenshot shows the WinCC Redundancy configuration dialog box with the following settings:

- Tab: 常规 (General)
- Server: WCC-DEV
- 缺省主机 (Default Master):
- 冗余伙伴服务器 (Redundant Partner Server): SV_REDUND_07
- 本地计算机设置 (可选) (Local Computer Settings (Optional)):
 - 与冗余伙伴的连接通过网络适配器 (Connection via Network Adapter): 无 (None)
 - 冗余伙伴的网络地址 (Network Address of Redundant Partner):
 - 动态 (Dynamic):
 - 静态 (Static):
 - 通过串行接口连接到冗余伙伴 (Connect to Redundant Partner via Serial Port): COM1
- 可选设置 (Optional Settings):
 - 伙伴服务器回到在线状态后变量记录同步 (Synchronize variable records after partner server returns to online state)
 - 伙伴服务器回到在线状态后报警记录同步 (Synchronize alarm records after partner server returns to online state)
 - 报警记录在线同步 (Synchronize alarm records online)
 - 过程连接中断后同步 (变量记录 + 报警记录) (Synchronize after process connection interruption (variable records + alarm records))
 - 过程连接出错时切换 WinCC 客户端 (Switch WinCC client when process connection fails)
- 为所有指定选项和用户归档启用同步 (Enable synchronization for all specified options and user archives):
 - 激活冗余 (Activate Redundancy)

2. 如果要组态并使用 WinCC Redundancy，应选中“激活冗余”(Activate Redundancy) 复选框。
3. 激活“默认主机”(Default Master) 选项，指定当两个服务器同时启动时通常会激活上面输入的服务器作为主机。
如果不激活该选项，该服务器将成为备用服务器。

注意**只有一台冗余服务器可以用作“默认主机”**

确保只对两个冗余服务器中的一个激活“默认主机”(Default Master) 选项。

否则，客户端进行冗余切换期间可能会出现问題。

4. 输入伙伴服务器的计算机名称或单击“搜索”(Search)。
5. 为状态监视指定是否通过网络适配器与冗余伙伴相连。
如有需要，选择“静态”(Static) 选项以输入固定网络地址和冗余伙伴的端口。
在网络适配器连接和串行连接中，首选网络适配器连接。
如果要使用串行连接，请选择接口。
也可以在“SIMATIC Shell”中组态这些设置：在 Windows 资源管理器中，选择“Simatic Shell”文件夹快捷菜单中的“冗余设置”(Redundancy settings) 条目。

6. 通过选择相应的可选设置，指定冗余恢复或中断进程连接后要执行的同步：
 - 伙伴服务器回到在线状态后变量记录的同步
 - 伙伴服务器回到在线状态后报警记录的同步
 - 在线同步报警记录：将操作员消息和不带变量连接的消息或带内部消息变量的消息进行同步。
 - 过程链接中断后同步（变量记录 + 报警记录）：在排除服务器与自动化系统之间的中断后，过程连接监视会启动自动归档同步。
 - 过程连接出错时切换 WinCC 客户端：连接到服务器的客户端切换到冗余伙伴服务器。该方案在“过程连接出错时切换客户端 (页 131)”下进行了描述。
7. 单击“确定”(OK) 保存设置。
8. 如有必要，请重新启动运行系统以应用这些设置。
如果不重新启动运行系统，只有以下选项的更改能立即生效：
 - 伙伴服务器回到在线状态后变量记录的同步
 - 伙伴服务器回到在线状态后报警记录的同步
 - 在线同步报警记录：
 - 过程链接中断后同步（变量记录 + 报警记录）其它选项的更改仅在运行系统重新启动之后生效。

参见

过程连接出错时切换客户端 (页 131)

如何组态用户归档的同步 (页 125)

WinCC 冗余 (页 112)

WinCC 冗余系统消息 (页 146)

冗余系统安装指南 (页 120)

故障情况 (页 135)

4.5.4 如何组态用户归档的同步

简介

可以通过操作、独立程序或自动化系统处理用户归档。对于冗余系统，组态用户归档的自动同步。

要求

两台冗余服务器上用户归档的组态必须相同。使用项目复制器来实现此目的。

步骤

1. 在 WinCC 项目管理器中打开“冗余”(Redundancy) 编辑器。打开“用户归档”(User Archive) 选项卡。



2. 在“用户归档”(User Archive) 列，按行显示所有组态的用户归档。双击“同步”(Synchronization) 列，激活或禁用各用户归档的同步。两个伙伴服务器上的设置必须完全相同。
3. 位于“全部用户归档的同步”(Synchronization of all User Archives) 域中的两个按钮用于激活或禁用所有显示的用户归档的同步。
4. 调用“冗余”编辑器后，“用户归档”编辑器中用户归档的组态是否已经更改？如果已更改，单击“更新”(Update) 按钮应用用户归档的当前组态。
5. 单击“确定”(OK) 保存设置。

说明

无法通过“加载在线更改”同步已更改的组态数据

说明

用户归档的归档同步更改后重新启动运行系统

只有重新启动运行系统后用户归档的归档同步更改才能生效。

并行编辑用户归档

在将记录并行添加到冗余用户归档时，请注意：

- 如果在服务器恢复后进行同步，记录只能添加到先前发生故障的服务器中。否则，将在脚本或用户归档控件中得到一条错误消息。
- 即使在在线同步期间，同步冗余归档中的记录仍将花费一段时间。

说明

如果两台服务器计算机均出现故障，或者两台计算机均关闭，则必须先启动最后使用的服务器计算机。否则，尚未保存的更改将丢失。

参见

如何组态冗余服务器 (页 123)

WinCC 冗余 (页 112)

冗余系统安装指南 (页 120)

WinCC 冗余系统消息 (页 146)

故障情况 (页 135)

4.5.5 如何为冗余服务器复制项目

简介

设置两台冗余服务器时，其硬件和软件必须具有相同的功能。

完成 WinCC 组态和 WinCC 项目的各项更改后，使用 WinCC 项目复制器生成冗余伙伴项目。

项目复制器可执行以下操作：

- 将图片、脚本和归档等所有相关的项目数据复制到冗余伙伴中。
- 如果已经对计算机进行组态以使用 WinCC Redundancy，则要在目标计算机上组态所有所需设置。

然后，必须手动更改计算机特定的设置。

说明

要将项目传送到冗余服务器，不能使用 Windows 项目管理器。

可在 SIMATIC Manager 中使用“加载在线更改”功能保存较小的更改，然后将其传送到运行系统中的服务器。

原理

选择希望在项目复制器中复制的项目。

4.5 组态冗余系统

指定要在其中复制项目的目标计算机和文件夹。项目文件夹在此目标文件夹中创建。
不能复制本地计算机上的项目。始终复制网络中具有访问权限的其它计算机上的项目。
根据项目状态，可将组态数据和运行系统数据复制到选定的文件夹中：

项目状态	组态数据	运行系统数据
关闭的项目	+	+
打开并已取消激活的项目	+	-
运行系统中的项目	+	-

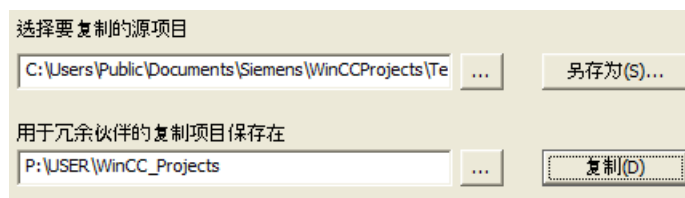
只能复制整个项目和整个文件夹结构。不能从复制操作排除任何数据或文件夹。



要求

- 两台计算机上均已安装 WinCC Redundancy 选项。
- 已在目标计算机上创建用于复制的目标文件夹，而且可以访问该文件夹。
- 具有对目标文件夹进行访问的权限。
- 目标计算机具有足够的可用硬盘空间。
- 必须在目标计算机上安装正确的 WinCC 版本。必须启动计算机。
- 在目标计算机上已取消激活运行系统。
- 目标计算机上的项目已关闭。

步骤

1. 在“Siemens Automation”的 Windows 程序组中选择“项目复制器”(Project Duplicator) 条目。打开 WinCC 项目复制器。



2. 在“选择要复制的源项目”框中输入要复制的项目。
直接输入路径和 <项目>.MCP 项目文件，或通过单击  按钮进行搜索。
3. 在“为冗余计算机保存复制项目在”框中，输入要存储所复制项目的路径。
直接输入文件夹路径和 <项目>.MCP 项目文件，或通过单击  按钮进行搜索。

4. 单击“复制”按钮。
“复制”窗口将会打开。复制时，项目复制器显示文件和文件夹以及一个进度条。使用“取消”按钮停止复制。
复制后，“项目复制器说明”窗口将打开。
WinCC 指示仍需检查的设置。

说明

如果在源计算机上复制一个打开的 WinCC 项目，将不显示进程条。

5. 使用“关闭”按钮关闭项目复制器。
6. 检查复制项目中的设置，并在必要时对其进行更改。
7. 检查下列各项：
 - 计算机名称。
 - 冗余编辑器的设置。
 - 必要时，检查编辑器的设置。

复制具有基于项目的访问保护的项目

必须安装 SIMATIC STEP 7，以便将具有基于项目的访问保护的 WinCC 项目传送到冗余服务器。

在“WinCC 项目复制器”对话框中单击“复制”按钮时，必须输入 STEP 7 项目的密码。

如果没有安装 SIMATIC STEP 7 或者输入的密码错误，项目复制器将中止，并显示一条错误消息。

4.5.6 如何在运行期间复制冗余项目

简介

如果编辑冗余项目，也可在操作期间对冗余服务器上的项目进行更新。

可以使用“加载在线更改”功能保存较小的更改，然后将这些更改传送给运行系统中的服务器。也可参见主题为“加载在线更改”的有关文档。

使用项目复制器进行复制

某些组态不能通过“加载在线更改”功能来保存。此时，必须使用项目复制器将项目复制到冗余服务器。

说明

无冗余

对于在正常操作时进行的更改，必须禁用其中一台伙伴服务器。在此期间，冗余不可用。

要求

- 已创建目标文件夹。
- 具有对目标文件夹进行访问的权限。
- 将在其上存储所复制项目的冗余服务器具有足够的空闲硬盘空间。

步骤

本节举例介绍了如何在包含两台服务器 Server1 和 Server2 的冗余系统中使用该功能。

1. 在冗余 Server1 上退出运行系统，并关闭项目。
2. 在 Server2 上于运行期间进行组态更改，并保存这些更改。
3. 在 Server2 上启动项目复制器。
4. 使用“复制”(Duplicate)按钮，将 Server1 上的项目复制到在第 1 步中禁用的项目的目标文件夹，并将原项目覆盖。
5. 打开 Server1 上的项目。
6. 检查这些设置。
7. 启动运行系统，等待进行冗余同步。

4.6 使用 WinCC Redundancy 的各种情况

4.6.1 过程连接出错时切换客户端

概述

冗余系统由两台功能完全相同的服务器组成。一台服务器是主服务器，而另一台是冗余伙伴服务器。

在未出现故障的运行状态下，服务器状态如下：

- 主服务器状态为“主机”。
- 冗余伙伴服务器状态为“备用机”。

客户端将连接到相应的首选服务器，如果未指定首选服务器，将连接到主服务器。

一旦两台服务器都在运行，将激活过程连接监视。WinCC 冗余周期性地确定主服务器和冗余伙伴服务器的故障逻辑连接的数目。

如果“主”服务器的故障逻辑连接比冗余伙伴服务器的多，则在系统变量“@RedundantServerState”中将服务器的状态设置为“故障”。客户机将切换到冗余伙伴服务器，该服务器现在为“主机”状态。

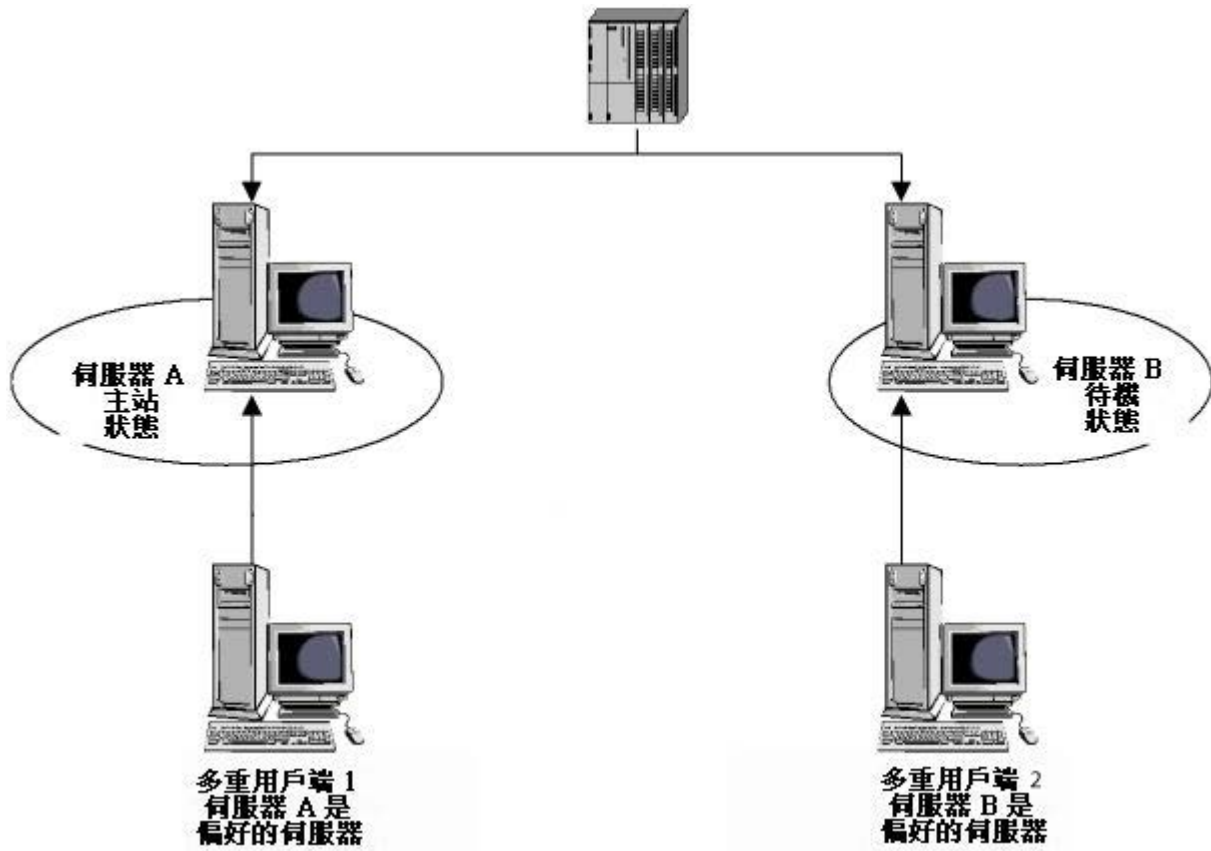
正常运行状态

系统由以下计算机组成：

- 冗余服务器 A
- 冗余服务器 B

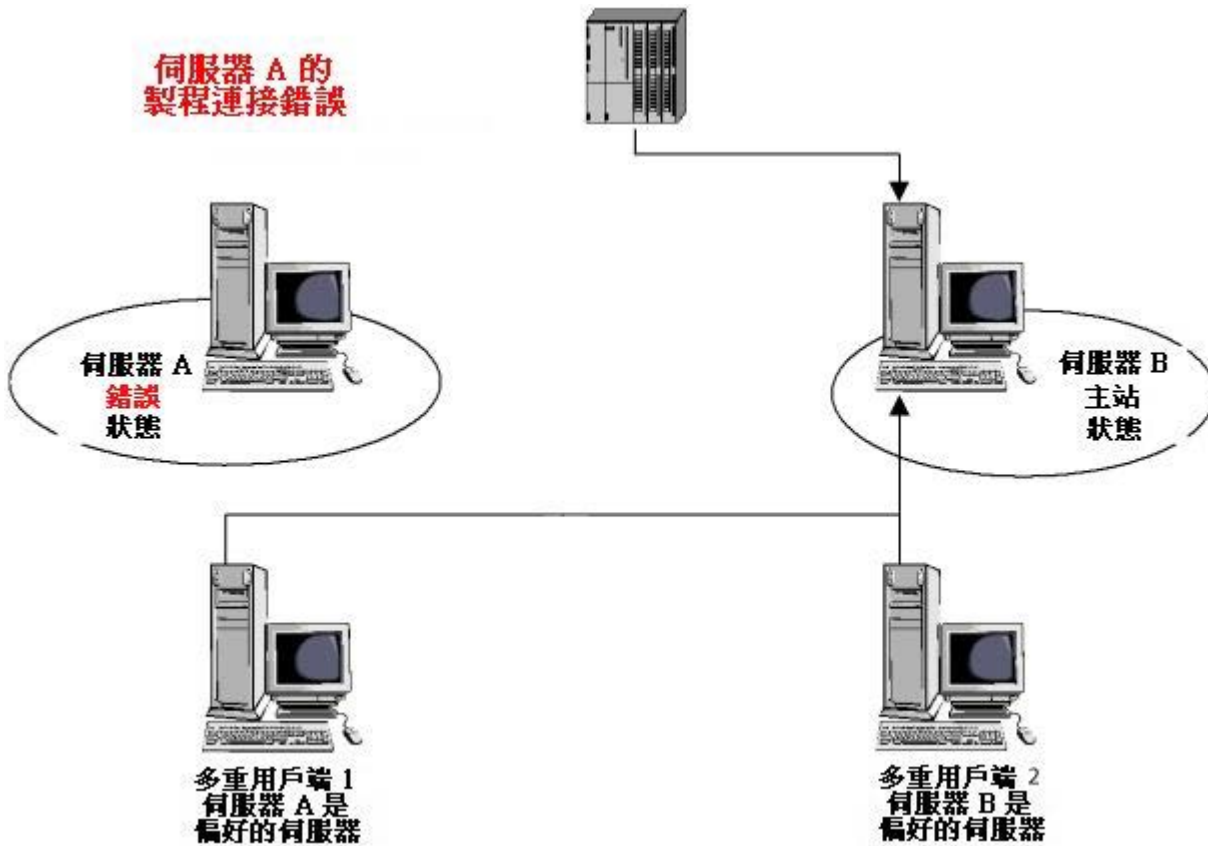
4.6 使用 WinCC Redundancy 的各种情况

- 首选服务器为 A 的客户机 1
- 首选服务器为 B 的客户机 2



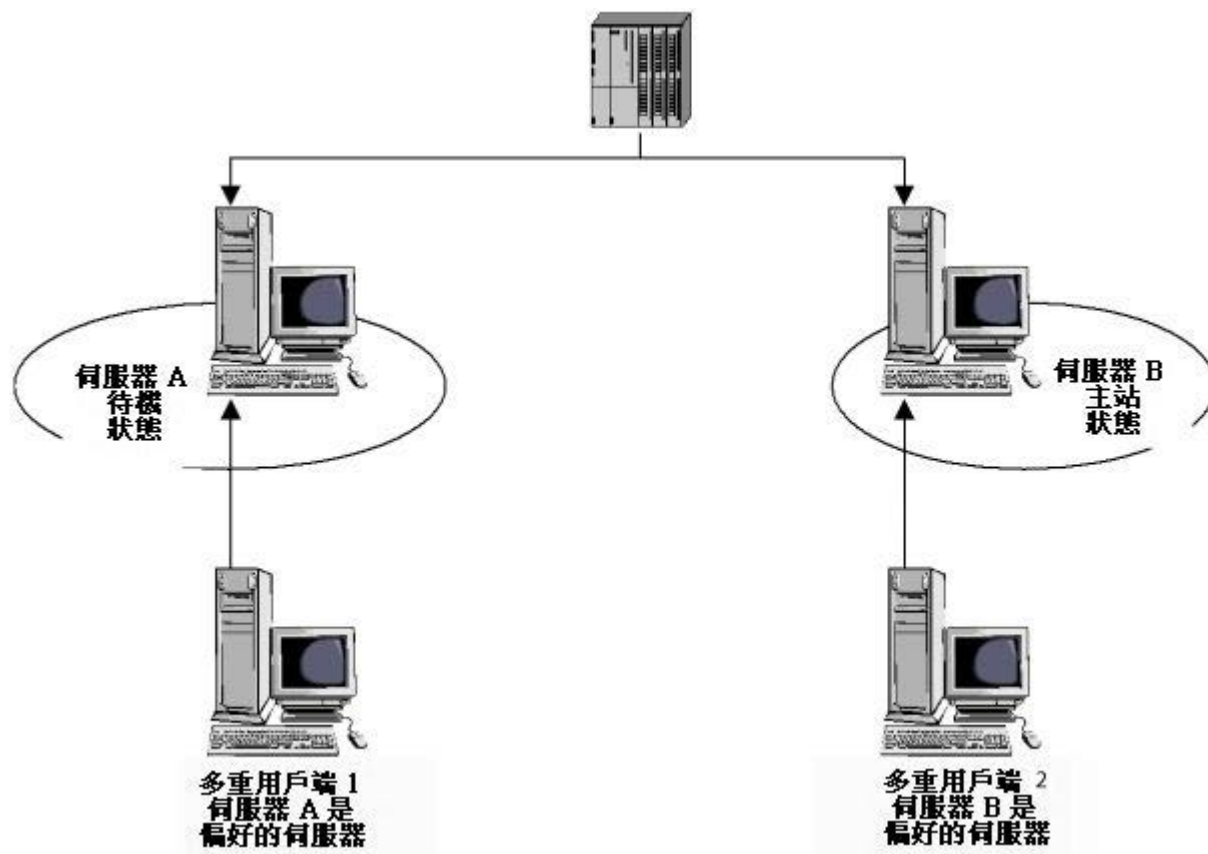
服务器 A 上的过程连接错误

服务器 A 上存在过程连接错误。该错误没有出现在服务器 B 上。服务器 A 上的故障逻辑连接的数目比服务器 B 上的大。因此，服务器 A 将接收“故障”状态。结果，客户端 1 切换到冗余服务器 B。



过程连接错误清除

当服务器 A 上的过程连接错误已经清除时，服务器 A 随后将处于“备用机”状态。由于输入时服务器 A 作为首选服务器，客户端 1 将切换至服务器 A。客户端 2 仍与其首选服务器 B 相连。



说明

不监视 OPC 接口。因此，当 OPC 接口出错时将不会有客户机切换。

参见

故障情况 (页 135)

冗余系统安装指南 (页 120)

WinCC 冗余 (页 112)

如何组态冗余服务器 (页 123)

如何组态用户归档的同步 (页 125)

4.6.2 故障情况

4.6.2.1 故障情况

简介

我们使用实际发生的故障来说明 WinCC Redundancy 是如何工作的。

1. 情况 1: 服务器计算机上的项目未处于运行系统中 (页 136)
2. 情况 2: 与伙伴服务器的连接出现故障 (页 138)
3. 情况 3: 与客户机的网络连接出现故障 (页 140)
4. 情况 4: 过程连接出现故障 (页 140)
5. 情况 5: 软件出错 (页 141)

WinCC 冗余将自己识别当前错误, 或通过下列动作对错误消息作出反应:

- 保存事件的时间。
- 归档同步。
- 更改主机和备用机标识符。
- 切换客户机。
- 触发消息。

服务器 PC 的启动

当服务器 PC 启动时, 冗余组件将确定伙伴服务器是否已经激活。

- 如果伙伴服务器已激活, 则将在该服务器计算机中设置“备用机”状态。
- 如果启动期间没有激活伙伴服务器, 则将在该服务器计算机中设置“主机”状态。

WinCC 冗余系统变量

服务器计算机的状态保存在系统变量"@RM_MASTER"中。

服务器计算机的状态	"@RM_MASTER"状态
主机	1
备用机	0

"@RM_MASTER_NAME"变量包含具有“主机”状态的服务器系统的名称，例如“Server1”。

"@RedundantServerState"变量显示各冗余服务器的冗余状态，如“备用”。

冗余只设置上述变量。两台服务器始终完全相同。

脚本或其它应用程序可以判断这些变量。只有"@RM_MASTER"变量可以更改。

有关系统变量的概述，请参见 WinCC 冗余系统变量 (页 142)。

交换状态信息

冗余的状态可通过单独的连接控制。可按以下方法建立连接：

- 使用网络适配器
- 使用串行接口

在网络适配器连接和串行连接中，首选网络适配器连接。

说明

注意，归档同步通过终端总线实现。不能通过状态连接执行归档同步。

4.6.2.2 情况 1：服务器计算机上的项目未处于运行系统中

简介

这种情况将讨论禁用 Server2 上的项目时 WinCC Redundancy 的行为。

将触发下列动作：

- Server1 存储 Server2 停机的日期和时间。
- Server1 通过系统消息报告 Server2 的故障。

- 如果 Server1 是备用服务器，则 Server1 将接管主服务器的角色。“@RM_MASTER”变量置位，同时“@RM_MASTER_NAME”和“@RedundantServerState”变量将会更改。
- 连接到 Server2 的客户机将切换到 Server1。

Server2 恢复在线状态

停机时间意味着在 Server2 的归档中存在空白。该空白可通过以下措施来填充：

- Server1 存储 Server2 恢复的日期和时间。
- Server1 通过系统消息报告 Server2 的恢复。
- 在 Server2 上执行 Server1 中下列归档的冗余同步。
 - 消息归档
 - 过程数据归档
 - 用户归档
- 对于 Server1，“@RM_MASTER”保持置位，对于 Server2，“@RM_MASTER”复位。两个服务器上的“@RM_MASTER_NAME”和“@RedundantServerState”仍保持不变。
- 将 Server2 组态为其首选服务器的客户机将切换回 Server2。

与在线同步相比，服务器发生故障后归档同步所需的时间会比较长。同步的持续时间取决于将要同步的数据记录数量以及计算机和网络的负载情况。

服务器交替出现故障

如果故障在两台服务器之间交替出现，则将依次对其进行同步。同步后，两个归档中的所有数据都可用。

Server1:



Server2:



如果已组态同步，则始终执行同步。

4.6 使用 WinCC Redundancy 的各种情况

故障 A

Server1 将所有值传送到 Server2。

故障 B

Server2 将所有值传送到 Server1。

故障 C

Server1 将所有值传送到 Server2。

这些过程全部在后台自动运行，与底层自动化系统中同时进行的过程值归档和消息归档无关。

4.6.2.3 情况 2：与伙伴服务器的连接出现故障

简介

这种情况将讨论与伙伴服务器的连接发生故障时冗余的行为。

在此事件出现之前，两个服务器都在运行系统中无故障运行。

然后，由于拔出了 Server1 的网络连接等原因，出现上述连接故障。

初始状况 1

发生连接故障时，Server1 为主服务器，Server2 为备用服务器。

发生连接故障

连接发生故障时将触发下列反应：

- Server2 成为主服务器，并保存发生故障的日期和时间。
- Server2 显示系统消息，指出伙伴服务器出现故障，Server2 现已成为主服务器。
- 两台服务器上的“@RM_MASTER”、“@RM_MASTER_NAME”和“@RedundantServerState”变量将相应地进行调整。

恢复连接

连接故障期间，不同步报警记录 and 用户归档消息。

采取以下措施：

- 主服务器 Server2 存储恢复时间。
- Server2 通过系统消息显示伙伴服务器恢复。
- 从主服务器到备用服务器进行冗余同步。

- 通过在线同步报警记录，从 Server1 向 Server2 报告以下内容，并在 Server1 上显示为系统消息：
 - 冗余运行时出现错误。
 - Server1 已切换到“备用”状态。
 - Server1 恢复。
- 两台服务器上的“@RM_MASTER”、“@RM_MASTER_NAME”和“@RedundantServerState”变量仍然保持不变。

初始状况 2

发生连接故障时，Server1 为备用服务器，Server2 为主服务器。

发生连接故障

连接发生故障时将触发下列反应：

- Server2 仍为主服务器，并保存发生故障的日期和时间。
- Server2 通过系统消息指示伙伴服务器故障。
- Server1 切换为内部状态“故障”(Fault)。将 Server1 用作首选服务器的客户端切换至 Server2。
- 两台服务器上的“@RM_MASTER”、“@RM_MASTER_NAME”和“@RedundantServerState”变量仍然保持不变。

恢复连接

连接故障期间，不同步报警记录 and 用户归档消息。

采取以下措施：

- Server2 存储恢复时间。
- Server2 通过系统消息显示伙伴服务器恢复。
- 从主服务器到备用服务器进行冗余同步。
- 通过在线同步报警记录，从 Server1 向 Server2 报告以下内容，并在 Server1 上显示为系统消息：
 - 冗余运行时出现错误。
 - Server1 恢复。
- 两台服务器上的“@RM_MASTER”、“@RM_MASTER_NAME”和“@RedundantServerState”变量仍然保持不变。

4.6.2.4 情况 3：与客户机的网络连接出现故障

简介

在本情况中，Server2 和从属于 Server2 的客户机“CL5”之间的网络连接存在故障。Server1 为主服务器。

系统将触发以下动作：

- 客户机“CL5”自动从有故障的 Server2 切换到正在运行的 Server1。

客户机网络故障清除

在清除网络故障后，将触发下列反应：

- 如果在故障发生前 Server1 已经是主服务器，则两台服务器上的“@RM_MASTER”、“@RM_MASTER_NAME”和“@RedundantServerState”变量仍然保持不变。
- 客户机“CL5”重新切换到首选服务器 Server2。

参见

WinCC 冗余系统变量 (页 142)

情况 2：与伙伴服务器的连接出现故障 (页 138)

情况 1：服务器计算机上的项目未处于运行系统中 (页 136)

故障情况 (页 135)

情况 4：过程连接出现故障 (页 140)

4.6.2.5 情况 4：过程连接出现故障

简介

在这种情况下，由于与自动化系统的网络连接中断，从而导致 Server2 的过程连接出现故障。

与自动化系统的连接发生故障

如果与服务器的连接发生故障，则只有在 WinCC Redundancy 中才能识别出与自动化系统的连接故障。

自动化系统与两台服务器的连接中断对于冗余并不是故障，例如自动化系统的故障。

出错时的反应

如果 WinCC 识别出故障，将触发下列动作：

- 在 Server2 上报告过程连接的故障。
- Server1 接收到有关伙伴 Server2 发生故障的消息。
- Server1 保存 Server2 发生故障的日期和时间。
- 如果已在“冗余”编辑器中组态“客户端随过程连接中的扰动发生变化”选项，则与该服务器相连的客户端将切换至伙伴服务器。
- 对于 Server1，“@RM_MASTER”变量将被设置为“主机”，对于 Server2，则设置为“备用机”。“@RM_MASTER_NAME”和“RedundantServerState”变量将作相应调整。在 Server2 中，将“@RedundantServerState”变量设置为“故障”。

Server2 上过程连接错误清除

如果已经激活过程连接监视，可通过下列措施填补 Server2 的归档空白：

- Server1 存储 Server2 的恢复时间。
- 如果在 Server1 上没有发现过程连接故障，则执行从 Server1 到 Server2 的冗余同步。将同步所有自动化系统的数据，包括未发生故障的自动化系统的数据。
- 对于 Server2，“@RedundantServerState”变量将从“故障”更改为“备用”。
- 通过一则系统消息宣告 Server2 上的过程连接错误已纠正。

4.6.2.6 情况 5：软件出错

简介

在本情况中，在 Server2 上正被监视的软件中产生了错误。当错误发生时，Server2 为“主机”状态，Server1 为“备用机”状态。有多台客户机连接到两台服务器。

如果“应用程序正常检查”功能检测到 WinCC 软件中出现了错误，则将启动下列动作：

- “应用程序正常检查”将故障报告给 WinCC Redundancy。Server2 的状态将在“@RedundantServerState”变量中设置为“故障”。“@RM_MASTER”变量将设置为“备用”。
- 对于 Server1，“@RM_Master”设为“主机”。“@RM_MASTER_NAME”和“RedundantServerState”将作相应调整。

4.6 使用 WinCC Redundancy 的各种情况

- 连接到 Server2 的客户机将切换到 Server1。
- 如果不是报警服务器本身发生错误，过程控制消息将通知各用户软件出错。

Server2 上软件故障结束时的措施

禁用受影响的 Server2 项目。重新启动 Server2。当激活 Server2 上的项目时，将自动对归档进行同步。

- 对于 Server2，“@RedundantServerState”设为“备用”。Server1 仍为“主机”。
- Server1 存储 Server2 恢复的日期和时间。
- 现在可以再次连接该服务器。归档同步只能追溯到检测到 Server2 发生软件故障的时刻。

参见

WinCC 冗余系统变量 (页 142)

情况 3: 与客户机的网络连接出现故障 (页 140)

情况 2: 与伙伴服务器的连接出现故障 (页 138)

情况 1: 服务器计算机上的项目未处于运行系统中 (页 136)

故障情况 (页 135)

情况 4: 过程连接出现故障 (页 140)

4.6.3 WinCC 冗余系统变量

WinCC 冗余系统变量

会在内部变量组“Performance”和“Redundancy”中创建系统变量，以分析冗余状态。

冗余变量

如果在 WinCC 项目管理器中打开“冗余”(Redundancy) 编辑器，然后再通过“确定”(OK) 将其关闭，则系统变量将由 WinCC Redundancy 创建。

“@RM_MASTER”和“@RM_MASTER_NAME”系统变量用于两个冗余服务器的主机/备用站控制以及客户端切换。

可以通过其它应用程序或脚本读取系统变量。

只能更改“@RM_MASTER”变量。

性能变量

性能变量 @PRF_REDUNDANCY_... 表示冗余系统的状态。

在创建 WinCC 项目时，系统变量将在“Performance”变量组中创建。

更多信息：

- “使用 WinCC > 使用项目 > 对运行系统进行设置 > 使用性能变量进行系统诊断”

冗余变量概览

系统变量	含义
@LocalMachineName	包含本地计算机名称。
@RedundantServerState	服务器的冗余状态： 0：未定义状态或初始值 1：服务器为主服务器（主站） 2：服务器为备用服务器 3：服务器处于“FAULT”状态 4：服务器独立或无冗余操作
@RM_MASTER	变量值 = 1：标识主服务器。 如果服务器成为备用服务器，则“@RM_MASTER”复位为“0”。 可通过脚本或其他方式更改变量值。
@RM_MASTER_NAME	主服务器的名称
@RM_SERVER_NAME	与客户端相连的服务器的名称
@RM_UA_ONL_“归档名称”	用于诊断： • 1：用户归档已更改。 • 0：用户归档的在线同步已完成。 为各个用户归档插入带有相应归档名称的独立变量。
@RM_Offline_UA_Name	用于诊断。 变量中包含刚完成同步的用户归档的名称。

性能变量概述

性能变量 @PRF_REDUNDANCY_... 表示冗余服务器的状态，这些状态同样会在“RedundancyControl”诊断工具中进行评估。

系统变量	描述
@PRF_REDUNDANCY_IS_SYNCHRONIZED	<p>同步状态：</p> <ul style="list-style-type: none"> 0：冗余应用程序未同步。 1：所有应用程序的冗余同步已完成。 <p>状态可能会受到 WinCC 和出于冗余目的而登录的其它应用程序（例如，SIMATIC BATCH）影响。</p>
@PRF_REDUNDANCY_VALIDATION	<p>服务器的评估分数。验证值决定了哪台服务器将成为主服务器。验证值取决于连接、运行系统状态等因素。</p> <p>正确组态冗余后，验证值在两个冗余服务器上相同。如果验证值不同，则数值较大的服务器将成为主服务器。</p> <p>典型值：</p> <ul style="list-style-type: none"> 37：服务器状态良好。 <ul style="list-style-type: none"> 运行系统已激活。 通过串行接口进行冗余连接 35：服务器状态良好。 <ul style="list-style-type: none"> 运行系统已激活。 通过 LAN 进行冗余连接 < 35：服务器具有内部“FAULT”状态。 <p>检查连接状态或服务器的状态。在出现服务器应用程序不再响应之类的关键运行状态时，“FAULT”状态将置位。</p> <p>如果服务器进入“FAULT”状态，伙伴服务器将成为主服务器。</p> <p>计算示例：</p> <ul style="list-style-type: none"> 如果已在服务器上禁用运行系统，验证值减小 4 分。 如果无法访问终端总线，验证值减小 20 分。
@PRF_REDUNDANCY_PARTNER_VALIDATION	<p>冗余伙伴服务器的评估分数</p> <p>正确组态冗余后，该值在两个冗余服务器上相同。</p>

系统变量	描述
@PRF_REDUNDANCY_AS_COUNT	<p>服务器上的 AS 连接数</p> <p>正确组态冗余后，该值在两个冗余服务器上相同。</p> <p>以下条件会导致冗余切换：</p> <ul style="list-style-type: none"> • 冗余服务器上的验证值相同。 • AS 连接的数量不同。 <p>在此情况下，AS 连接数量较多的服务器将成为主服务器。</p>
@PRF_REDUNDANCY_PARTNER_AS_COUNT	<p>冗余伙伴服务器上的 AS 连接数</p> <p>正确组态冗余后，该值在两个冗余服务器上相同。</p>
@PRF_REDUNDANCY_CURRENT_STATE	<p>服务器的冗余状态：</p> <ul style="list-style-type: none"> • 0：未定义的状态 • 1：服务器为主服务器 • 2：服务器为备用服务器 • 3：服务器处于“FAULT”状态 • 4：服务器独立或无冗余操作
@PRF_REDUNDANCY_PARTNER_CURRENT_STATE	<p>冗余伙伴服务器的冗余状态</p>
@PRF_REDUNDANCY_FAULT_POSTPONED	<p>变量值 = 1：服务器具有“FAULT_POSTPONED”状态。</p> <p>本地服务器的内部状态为“FAULT”，但伙伴服务器无法切换到“主站”状态。无法进行冗余故障切换。原因可能是正在运行冗余同步等。</p> <p>满足冗余切换的条件后，服务器会立即切换为“FAULT”状态。</p> <p>“@PRF_REDUNDANCY_CURRENT_STATE”变量的值为“3”。</p>
@PRF_REDUNDANCY_PARTNER_FAULT_POSTPONED	<p>变量值 = 1：冗余伙伴服务器具有“FAULT_POSTPONED”状态。</p>
@PRF_REDUNDANCY_SWITCHOVER_COUNT	<p>自从运行系统激活以来或者自上一次通过“@PRF_REDUNDANCY_SWITCHOVER_COUNT_RESET”复位以来的冗余切换次数。</p>

4.6 使用 WinCC Redundancy 的各种情况

系统变量	描述
@PRF_REDUNDANCY_SWITCHOVER_COUNT_PERIOD	指定周期内的冗余切换次数 默认设置： <ul style="list-style-type: none"> • 时间段：1 个日历日 • 该值会在每天的 0:00 AM 复位。
@PRF_REDUNDANCY_SWITCHOVER_COUNT_RESET	复位变量将复位以下性能变量的值： <ul style="list-style-type: none"> • @PRF_REDUNDANCY_SWITCHOVER_COUNT

“RedundancyControl”诊断工具

要进行全面的冗余诊断，可使用“RedundancyControl”诊断工具。

有关更多信息，请参见应用示例“WinCC V7.x 和 WinCC Professional 中的冗余”：

- <https://support.industry.siemens.com/cs/ww/en/view/109772627> (<https://support.industry.siemens.com/cs/cn/zh/view/109772627>)

参见

WinCC 冗余系统消息 (页 146)

<https://support.industry.siemens.com/cs/ww/en/view/109772627> (<https://support.industry.siemens.com/cs/cn/zh/view/109772627>)

4.6.4 WinCC 冗余系统消息

概述

WinCC Redundancy 提供了一系列系统消息。必须在报警记录编辑器的“选项”(Options) 菜单中选择命令“WinCC 系统消息...”(WinCC System Messages...) 才能使用系统消息。

WinCC Redundancy 可以输出下列系统消息：

消息号	WinCC 消息文本
1012200	REDRT: 伙伴站发生故障 已在伙伴服务器终止 WinCC。
1012201	REDRT: 重启伙伴站 已在伙伴服务器上重启 WinCC

消息号	WinCC 消息文本
1012202	REDRT: 项目功能不一致
1012203	REDRT: 归档同步失败
1012204	REDRT: 冗余中的内部错误
1012205	REDRT: 与伙伴的通信出错 与伙伴服务器的连接发生故障
1012206	REDRT: 已重新建立与伙伴的通信 与伙伴服务器的连接已恢复
1012207	REDRT: 伙伴服务器 - 未激活 WinCC 启动时, 系统检测到 WinCC 未启动。
1012208	REDRT: 归档同步开始 该消息在归档同步开始时发出
1012209	REDRT: 归档同步完成 该消息在归档同步结束时发出
1012216	REDRT: 同步中断 因其它故障导致同步中断
1012217	REDRT: 未激活伙伴服务器项目 启动时, 系统检测到伙伴服务器上的 WinCC 未运行或不在运行系统中
1012218	SWITCH: 已自动切换客户端 客户端已自动切换到伙伴服务器
1012219	SWITCH: 已手动切换客户端 已将客户端手动切换到伙伴服务器
1012220	REDRT: 尚未准备好同步所有用户归档 尚未准备好同步所有本地组态的用户归档, 因为至少一个归档在伙伴服务器上的归档结构不同, 或者在伙伴服务器上尚未激活同步。
1012221	REDRT: 准备好同步所有用户归档 准备好对本机组态的用户归档同步, 而且归档结构与伙伴服务器的相对应。
1012226	REDRT: 已激活伙伴服务器项目 启动时, 系统检测到在伙伴服务器上已激活 WinCC。
1012227	REDRT: 错误 - 伙伴计算机不是服务器 启动时, 系统检测到已组态的伙伴计算机不是服务器。
1012240	REDRT: <应用程序名称>中的<错误描述>错误触发切换。 由于所指定应用程序出现错误, 应用程序正常检查将触发切换。

4.6 使用 WinCC Redundancy 的各种情况

消息号	WinCC 消息文本
1012241	REDRT: 切换到<状态描述>状态 消息指出状态已改变。
1012244	REDRT: 报警记录在线同步期间发生超载 要同步的消息过多。
1012245	REDRT: 丢失串行连接
1012246	REDRT: 重新建立串行连接
1012247	REDRT: <生成消息的计算机名称> OS 服务器 (主) <计算机名称> OS 服务器 (备用) <计算机名称> 冗余错误 根据故障情况不同, 由主服务器和备用服务器或其中一个服务器发送冗余错误。冗余可能遭到损坏。
1012248	REDRT: 已重新建立 OS 服务器 (备用) 冗余
1012349	REDRT: 通过网卡 (MAC) 地址的连接丢失 通过冗余 LAN 与伙伴服务器的连接发生中断或丢失。
1012350	REDRT: 已重新建立通过网卡 (MAC) 地址的连接 通过冗余 LAN 与伙伴服务器的连接已重新建立。
1012351	REDRT: RedundancyControl: 检测到系统阻塞。切换到故障状态。
1012352	REDRT: RedundancyControl: 检测到系统阻塞。尽快重启计算机。
1012354	RedundancyControl: 故障状态发生改变, 但服务器隔离未激活
1012355	RedundancyControl: 故障状态发生改变, 但服务器隔离被 @1@s@ 锁定。原因: @2@s@
1012356	RedundancyControl: 故障状态发生改变 => 服务器被隔离
1012357	RedundancyControl: 故障状态发生改变, 但自动重启未激活
1012358	RedundancyControl: 故障状态发生改变, 但自动重启因网络适配器断开和 DHCP 启用而锁定
1012359	RedundancyControl: 计算机重启被 @1@s@ 锁定。原因: @2@s@
1012360	RedundancyControl: 计算机重启因距上次重启的时间小于 @1@s@ 而取消
1012361	RedundancyControl: 计算机重启因不允许在 @1@s@ 重启后的 @2@s@ 秒内再次重启而取消
1012362	RedundancyControl: 将在 @1@s@ 秒后重启计算机
1012700	自诊断: 节点 @10@s@ 的值 @7@s@ 无效。
1012701	自诊断: 节点 @10@s@ 的值 @7@s@ 超出错误上限。
1012702	自诊断: 节点 @10@s@ 的值 @7@s@ 超出错误下限。

消息号	WinCC 消息文本
1012703	自诊断: @100%s@: 站 @10%s@ 的值 @7%s@ 超出警告上限。
1012704	自诊断: @100%s@: 站 @10%s@ 的值 @7%s@ 超出错误下限。
1012705	自诊断: @100%s@: 节点 @10%s@ 的值 @7%s@ 不再超出错误限值。
1012706	自诊断: @100%s@: 节点 @10%s@ 的值 @7%s@ 正常。
1012707	自诊断: @100%s@: 节点 @10%s@ 导致 @2%s@。

参见

WinCC 冗余系统变量 (页 142)

过程通讯

5.1 通讯基本信息

简介

通信是指在两个通信伙伴之间进行数据交换。

通信

通信伙伴可以是用于与其它网络组件进行通信并交换数据的任何网络组件。在 WinCC 中，通信伙伴可以是自动化系统 (AS) 中的中央模块和通信模块，也可以是 PC 中的通信处理器。

通信伙伴间传送的数据可以用于不同用途。在 WinCC 中，通信有如下用途：

- 控制过程
- 调用过程数据
- 指示过程中的异常状态
- 归档过程数据

5.2 组态连接的基本规则

采集周期和更新时间

组态软件中定义的变量采集周期是决定可实现的更新时间的主要因素。

更新时间是采集周期、传输时间和处理时间之和。

要获得最佳的更新时间，在组态期间请记住以下几点：

- 优化数据区域的最大和最小尺寸。
- 将同属的数据区定义为一个整体。如果建立一个大数据区而非多个小数据区，则可改善更新时间。
- 采集周期过小会降低性能。应根据过程值的变化速率设置采集周期。以烤炉的温度变化为例，它的变化速度要比电气传动装置慢得多。
- 应将报警或画面的变量不留间隙地放在一个数据区中。
- 控制器中的变化至少要持续一个采集周期才能被可靠地检测到。
- 要实现无错误传输，应将传输率设置为尽可能高的值。

画面

画面的刷新频率取决于要显示的数据类型和数据量。

要缩短更新时间，务必为需要快速更新的对象组态更短的采集时间。

曲线

使用位触发的曲线时，如果在“曲线传送区域”中设置了组位，则在 WinCC 站中会更新在该区域中设置了组位的所有曲线。这些位会在下一个周期复位。

只有所有位在 WinCC 站中都复位完毕后，才能在 PLC 程序中再次设置组位。

5.3 WinCC 过程通讯

5.3.1 WinCC 过程通讯

简介

可以从 WinCC 访问自动化系统中的过程变量（外部变量）。但是，在 WinCC 中组态过程连接之前，应使用检查单检查是否已满足以下先决条件：

- 自动化系统必须配备 WinCC 中的通信驱动程序所支持的通信接口。
- 必须在自动化系统中组态此接口，使控制程序能够通过通信调用访问该接口。通信硬件的组态参数必须已知。
- WinCC 应访问的变量的地址必须已知。请注意，这些地址取决于所使用的自动化系统。
- WinCC 系统中必须已安装相应通信硬件（通信处理器、标准 I/O 端口 COMx、...）。为了安装通信硬件，还必须先安装随附的操作系统驱动程序（硬件驱动程序）。通信处理器的硬件和软件设置必须已知。
- 视 WinCC 系统所使用的通信处理器而定，可能还需要进行更多设置。例如，使用工业以太网或 PROFIBUS 时，必须创建本地数据库。这些连接参数也必须已知。

为了在运行期间进行操作，WinCC 和 AS 之间还必须存在物理连接，以便访问外部变量。

S7DOS 组态

如果您使用的是 S7DOS，则需要“S7DOS V9”及更高版本的 IPv4 协议。

因此，请在以太网属性中为网络适配器或 SIMATIC 以太网 CP 激活 IPv4 协议。

通过这种方式，可确保 S7DOS 的模块检测适用于 TCP、RFC1006 和 ISO 协议。

5.3.2 WinCC 通讯的原理

简介

WinCC 使用称为变量管理的功能集中管理其变量。可在 WinCC 运行系统中采集和管理在项目中创建的以及在项目数据库中存储的所有数据和变量。

所有应用程序，如图形运行系统、报警记录运行系统或变量记录运行系统（全局脚本），都必须以 WinCC 变量的形式从变量管理中请求数据。

WinCC 和自动化系统 (AS) 之间的通信

在使用 WinCC 进行的工业通信中，通信意味着使用变量和过程值交换信息。

为了采集过程值，WinCC 通信驱动程序向 AS 发送请求帧。而 AS 则在相应的响应帧中将所请求的过程值发送回 WinCC。



WinCC 和 AS 之间必须始终存在物理连接。

该连接的属性（如，传送介质和通信网络）定义了通信的状态，组态 WinCC 中的通信时需要这些属性。

通信驱动程序

通信驱动程序是用于在 AS 和 WinCC 的变量管理之间建立连接的软件组件，这样可以提供 WinCC 变量和过程值。在 WinCC 中，有大量通信驱动程序可使用各个总线系统连接不同 AS。每个通信驱动程序只能在 WinCC 项目中集成一次。

在 WinCC 中，通信驱动程序也被称为“通道”，并具有文件扩展名“*.chn”。计算机中安装的所有通信驱动程序都位于 WinCC 安装目录的子目录“\bin”中。

一个通信驱动程序针对不同通信网络会有不同的通道单元。

通道单元

每个通道单元形成一个基础硬件驱动程序的接口，因此在 PC 中仅对应一个通信处理器。因此，每个使用的通道单元必须分配到相关的通信处理器。

对于某些通道单元，会在系统参数中进行额外的组态。

对于使用 OSI 模型传输层（第 4 层）的通道单元，还将定义传输参数。

连接（逻辑）

对 WinCC 和 AS 进行了正确的物理连接后，WinCC 中需要通信驱动程序和相关的通道单元来创建和组态与 AS 的（逻辑）连接。

在运行系统中将通过此连接进行数据交换。可使用系统变量来建立或终止此连接，以及查询连接状态。

在 WinCC 中，连接是指两个通信伙伴的已组态逻辑分配，用于执行某种通信服务。

每个连接都有两个包含必要信息的端点，这些信息包括用来对通信伙伴寻址的必要信息以及用来建立该连接的其它属性。

在通道单元下使用特定的连接参数对连接进行组态。一个通道单元下也可以创建多个连接，这取决于通信驱动程序。

参见

在运行系统中为连接状态组态变量 (页 155)

5.3.3 在运行系统中为连接状态组态变量

激活 WinCC Runtime 时，将与已组态控制器建立连接。

要在运行系统中选择性地禁用或激活单个连接，请使用“ConnectionStates”变量组的系统变量。

另一个系统变量支持当前连接状态的查询。

所支持的通道

连接状态的系统变量可用于 WinCC v7.5 及以上版本支持的通信驱动程序。

“系统信息”通道是一个例外，仅用于评估系统信息。

诊断：连接性能

使用 WinCC 性能变量来评估连接的时间特性。更多信息：

- “使用性能变量检查连接 (页 565)”

变量组“ConnectionStates”

为了指定或确定通道的连接状态，为每个连接创建以下系统变量：

- @<Connectionname>@ForceConnectionStateEx
使用变量在运行系统中建立或终止连接
- @<Connectionname>@ConnectionStateEx
使用变量确定运行系统中的连接状态

变量具有变量类型“无符号 32 位数”(DWORD)。

如果更改了连接名称，则两个系统变量也将被重命名。

变量数值

变量	使用	值	解释
@<...>@ForceConnectionStateEx	确定连接状态	1	建立连接 起始值 = 1： 激活运行系统时，建立连接。
		0	连接终止 起始值 = 0： 激活运行系统时，连接保持禁用。 连接变量未归档。
@<...>@ConnectionStateEx	确定当前的连接状态	1	连接准备就绪。
		0	连接被中断或终止。

要求

- 所需连接在变量管理中创建。

操作步骤

1. 在变量管理的导航区域中选择所需连接。
2. 在连接的快捷菜单中，选择条目“为激活/禁用创建变量”(Create tags for activation/deactivation)。
在“内部变量”(Internal tags)下的导航树中创建一个新变量组“ConnectionStates”。
该组包含两个已创建的变量。
3. 为控制系统上的每个变量组态一个单独的地址。
为此，使用一个未使用或虚构的地址。该地址仅用于变量传送。

使用变量

查询连接状态

要确定连接状态，请读取变量“@<...>@ConnectionStateEx”的值。

终止连接

要禁用连接，请在变量“@<...>@ConnectionStateEx”中设置数值“0”。

停止关联过程变量归档。

建立连接

要重新激活已中断的连接，请在变量“@<...>@ConnectionStateEx”中设置数值“1”。

会再次归档相应通信通道的过程变量。

显示连接状态

例如，可在包含 SVG 对象的过程画面中显示连接状态。

自 V2.0 起，SVG 库“IndustryGraphicLibrary”在文件夹“SIMATIC > SystemDiagnostic”中包含以下对象：

- SysDiag_DiagnosticsIndicator
- SysDiag_SignalLamp
- SysDiag_SignalTower

参见

如何使用“状态 - 逻辑连接”功能来检查通道 (页 568)

使用变量状态监视连接状态 (页 629)

使用性能变量检查连接 (页 565)

5.3.4 外部变量

5.3.4.1 外部变量

简介

为获得某些 AS 数据的访问权限，WinCC 中需要变量。这些取决于至 AS 的连接，变量被称为外部变量。而不包含过程连接的变量则被称为内部变量。

数据类型和类型转换

组态外部变量时，必须指定变量名以及数据类型，对于某些数据类型，还必须指定类型转换：

在 WinCC 中，数据类型决定了数据格式。类型转换用于指定从 AS 数据格式到 WinCC 数据格式的转换。类型转换适用于两个传输方向：

- 在 AS 中：例如，针对某些功能（如定时器值/BCD 显示）或因要寻址的信息（如在数据块或 I/O 区中的字节地址和字地址）。
- 在 WinCC 中：例如，针对模拟值处理或计算。

实际上，通常会预设 AS 数据格式。而对于 WinCC 格式的选择，存在以下几种情况：

- WinCC 数据格式可与 AS 格式匹配。为此，可选择以下类型转换：在两侧使用相同的格式并且会考虑到符号（取决于 WinCC 数据类型），例如，“WordToSignedWord”。如果这通过所选数据类型无法实现，则必须在 WinCC 中进行更改。
- WinCC 格式基于 WinCC 中处理的值。

选择数据类型和根据需要选择类型转换时必须遵循以下内容：

- 符号：是否应将其考虑在转换范围内？操作期间是否还会出现负变量值？（例如，存在百分比形式的闭环控制误差时）
- 值范围：操作期间的变量值是否在两种格式的值范围内，或者，WinCC 或 AS 中是否可能出现溢出？如果出现溢出，将不能在另一侧显示值，否则会导致后续处理出现问题。
- 不同的类型转换具有相同的值范围：一种数据类型的多种类型转换可能具有相同的值范围，例如，“ByteToUnsignedDword”和“ByteToUnsignedWord”的值范围均为 [0...127]。在这种情况下，应始终检查 AS 数据所采用的格式，以及检查此格式是否因范围过大（例如，DWord，而非 Word）而造成不必要的资源浪费。

如果所选的类型转换未涵盖 AS 所需的值范围，则必须在 WinCC 中更改数据类型。

说明

如果过程变量组态错误（例如，因寻址错误而引起），则会中断与自动化系统的通信。

WinCC 数据类型和类型转换

下表显示了支持类型转换的 WinCC 数据类型。

数据类型	类型转换
二进制变量	否
无符号 8 位值	是
有符号 8 位值	是
无符号 16 位数	是
有符号 16 位数	是
无符号 32 位数	是
有符号 32 位数	是
浮点数 32 位 IEEE 754	是
64 位 IEEE 754 浮点数	是
文本变量，8 位字符集	否
文本变量，16 位字符集	否
原始数据类型	否

说明

对于类型转换，确保 AS 所发送的数据可由 WinCC 在所选类型转换过程中进行解释。如果数据无法由 WinCC 进行解释，则会在“..\Siemens\WinCC\Diagnose”目录下的“WinCC_sys_0x.log”文件中输入一条错误。

数值变量类型的线性标定

可以对数值数据类型执行线性标定。过程中的变量值范围可采用线性方式映射到已定义的 WinCC 变量值范围。

5.3 WinCC 过程通讯

例如，过程可能需要以 bar 为单位的压力设定值规范，而该值将在 WinCC 中被用户以 mbar 的形式进行输入。通过线性标定，过程中的值范围 [0...1] 可以转换为 WinCC 变量的值范围 [0...1000]。

文本变量的长度信息

“文本变量 8 位字符集”和“文本变量 16 位字符集”数据类型的变量需要长度信息。对于稍后要容纳 10 个字符的文本变量，如果数据类型为“文本变量 8 位字符集”，则其长度值必须为 10；如果数据类型为“文本变量 16 位字符集”，则其长度值必须为 20。

在自动化系统中寻址

在 AS 中，WinCC 变量被分配到某个数据区。必须在 AS 中以某种方式对它们进行寻址。寻址方法取决于通信伙伴的类型。

变量名称的前缀和后缀

从 AS 下载变量后，可为用于连接实例的变量名称定义前缀和后缀。通过变量选择对话框导入过程变量后，会自动为所有连接变量添加前缀或后缀。更改前缀或后缀不会影响已导入的变量。

5.3.4.2 如何创建新连接

简介

只能基于到 AS 的连接创建外部变量。如果不存在所需连接，则必须先进行创建。

要求

- 已安装了所需的通信处理器和相应的硬件驱动程序。
- 并已安装了要求的通信驱动程序，如，“SIMATIC S7 Protocol Suite”。

步骤

1. 在 Configuration Studio 的导航栏中选择“变量管理”(Tag Management)。
2. 在导航区域中选择所需的通道单元，例如“PROFIBUS”。
3. 在通道单元的弹出式菜单中选择“新建连接...”条目。

4. 在数据区域中为连接指定唯一名称。
5. 在“属性”(Properties)窗口中为该连接定义所需参数。更多相关信息，可参见相关通道的帮助/文档。

5.3.4.3 外部变量的组态方法

简介

对于大多数数据类型，变量的创建步骤是相似的。

但是，某些数据类型则需要特殊设置（步骤 5 - 7）。

要求

- 已安装了所需的通信处理器和硬件驱动程序。
- 已安装了要求的通信驱动程序，如，“SIMATIC S7 Protocol Suite”。
- 已根据通道单元创建连接（例如“PROFIBUS”）。

步骤

1. 在导航区域的树形视图中，选择要为其创建变量的连接。
2. 在“名称”(Name)列最上方的空单元格中输入在 WinCC 项目中唯一的变量名，例如：WinCCTag_01”。
3. 在“数据类型”(Data type)字段中定义变量的数据类型，如“浮点数 64 位 IEEE 754”。
4. 在 AS 的“AS 长度”(AS Length)字段中指定变量的地址区域。
使用不支持对二进制或 8 位变量的位/字节访问的通道时，将首先显示“位/字节变量”(Bit/Byte Tag)对话框，然后再显示“变量属性”(Tag properties)对话框。
有关详细信息，请参阅“二进制写入机制的原理”(Principle of the BinWrite-Mechanism)。
单击“确定”(OK)按钮关闭“位/字节变量”(Bit/Byte Tag)对话框或“变量属性”(Tag properties)对话框。
5. 使用数字变量时，WinCC 会在“格式改编”(Format adaptation)域中提供一个建议的格式改编。需要时，可选择另一种格式改编。显示依次为“X to Y”，其中 X = WinCC 格式，Y = AS 格式，如“DoubleToDouble”。
6. 激活“线性标定”(Linear scaling)复选框对数字变量进行线性转换。输入“过程值范围”（在 AS 中）和“变量值范围”（在 WinCC 中）的上限和下限。
7. 将激活文本变量的“长度”域。在此输入文本变量的长度（以字节为单位）。
8. 使用“确定”按钮关闭所有对话框。

5.3.4.4 按 WinCC 数据类型排序的格式改编

简介

在组态外部变量时，必须将所有数字数据类型的数据改编为另一种格式。

在 WinCC 端，数据类型决定数据格式。格式调整还定义从 WinCC 格式到 AS 格式的转换。该定义对两个传输方向均适用。

从以下选择框中选择所需要的 WinCC 数据类型。下表列出相应可能的格式改编和取值范围。

WinCC 数据类型

表格 5-1 有符号 8 位值

格式改编“有符号 8 位值”	取值范围
CharToUnsignedByte	0...127
CharToUnsignedWord	0...127
CharToUnsignedDword	0...127
CharToSignedByte	-128...+127 (无转换)
CharToSignedWord	-128...+127
CharToSignedDword	-128...+127
CharToMSBByte	-127...+127
CharToMSBWord	-128...+127
CharToMSBDword	-128...+127
CharToBCDByte	0...99
CharToBCDWord	0...127
CharToBCDDword	0...127
CharToSignedBCDByte	-9...+9
CharToSignedBCDWord	-128...+127
CharToSignedBCDDword	-128...+127
CharToExtSignedBCDByte	-79...+79
CharToExtSignedBCDWord	-128...+127
CharToExtSignedBCDDword	-128...+127
CharToAikenByte	0...99

格式改编“有符号 8 位值”	取值范围
CharToAikenWord	0...127
CharToAikenDword	0...127
CharToSignedAikenByte	-9...+9
CharToSignedAikenWord	-128...+127
CharToSignedAikenDword	-128...+127
CharToExcessByte	0...99
CharToExcessWord	0...127
CharToExcessDword	0...127
CharToSignedExcessByte	-9...+9
CharToSignedExcessWord	-128...+127
CharToSignedExcessDword	-128...+127

表格 5-2 无符号 8 位值

格式改编“无符号 8 位值”	取值范围
ByteToUnsignedByte	0...255 (无转换)
ByteToUnsignedWord	0...255
ByteToUnsignedDword	0...255
ByteToSignedByte	0...127
ByteToSignedWord	0...255
ByteToSignedDword	0...255
ByteToBCDByte	0...99
ByteToBCDWord	0...255
ByteToBCDDword	0...255
ByteToAikenByte	0...99
ByteToAikenWord	0...255
ByteToAikenDword	0...255
ByteToExcessByte	0...99
ByteToExcessWord	0...255
ByteToExcessDword	0...255

5.3 WinCC 过程通讯

表格 5-3 有符号 16 位数

格式改编“有符号 16 位数”	取值范围
ShortToUnsignedByte	0...255
ShortToUnsignedWord	0...32767
ShortToUnsignedDword	0...32767
ShortToSignedByte	-128...+127
ShortToSignedWord	-32768...+32767 (无转换)
ShortToSignedDword	-32768...+32767
ShortToMSBByte	-127...+127
ShortToMSBWord	-32767...+32767
ShortToMSBDword	-32768...+32767
ShortToBCDByte	0...99
ShortToBCDWord	0...9999
ShortToBCDDword	0...32767
ShortToSignedBCDByte	-9...+9
ShortToSignedBCDWord	-999...+999
ShortToSignedBCDDword	-32768...+32767
ShortToExtSignedBCDByte	-79...+79
ShortToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDDword	-32768...+32767
ShortToAikenByte	0...99
ShortToAikenWord	0...9999
ShortToAikenDword	0...32767
ShortToSignedAikenByte	-9...+9
ShortToSignedAikenWord	-999...+999
ShortToSignedAikenDword	-32768...+32767
ShortToExcessByte	0...99
ShortToExcessWord	0...9999
ShortToExcessDword	0...32767
ShortToSignedExcessByte	-9...+9

格式改编“有符号 16 位数”	取值范围
ShortToSignedExcessWord	-999...+999
ShortToSignedExcessDword	-32768...+32767

表格 5-4 无符号 16 位数

格式改编“无符号 16 位值”	取值范围
WordToUnsignedWord	0...65535 (无转换)
WordToUnsignedByte	0...255
WordToUnsignedDword	0...65535
WordToSignedByte	0...127
WordToSignedWord	0...32767
WordToSignedDword	0...65535
WordToBCDByte	0...99
WordToBCDWord	0...9999
WordToBCDDword	0...65535
WordToAikenByte	0...99
WordToAikenWord	0...9999
WordToAikenDword	0...65535
WordToExcessByte	0...99
WordToExcessWord	0...9999
WordToExcessDword	0...65535
WordToSimaticBCDCounter	0...999
WordToSimaticCounter	0...999

表格 5-5 有符号 32 位数

格式改编“有符号 32 位数”	取值范围
LongToSignedDword	-2147483648...+2147483647 (无转换)
LongToUnsignedByte	0...255
LongToUnsignedWord	0...65535

5.3 WinCC 过程通讯

格式改编“有符号 32 位数”	取值范围
LongToUnsignedDword	0...2147483647
LongToSignedByte	-128...+127
LongToSignedWord	-32768...+32767
LongToSignedQword	-2147483648...+2147483647
LongToMSBByte	-127...+127
LongToMSBWord	-32767...+32767
LongToMSBDword	-2147483647...+2147483647
LongToMSBQword	-2147483648...+2147483647
LongToBCDByte	0...99
LongToBCDWord	0...9999
LongToBCDDword	0...99999999
LongToSignedBCDByte	-9...+9
LongToSignedBCDWord	-999...+999
LongToSignedBCDDword	-9999999...+9999999
LongToSignedBCDQword	-2147483648...+2147483647
LongToExtSignedBCDByte	-79...+79
LongToExtSignedBCDWord	-7999...+7999
LongToExtSignedBCDDword	-79999999...+79999999
LongToExtSignedBCDQword	-79999999...+79999999
LongToAikenByte	0...99
LongToAikenWord	0...9999
LongToAikenDword	0...99999999
LongToSignedAikenByte	-9...+9
LongToSignedAikenWord	-999...+999
LongToSignedAikenDword	-9999999...+9999999
LongToSignedAikenQword	-999999999...+999999999
LongToExcessByte	0...99
LongToExcessWord	0...9999
LongToExcessDword	0...99999999
LongToSignedExcessByte	-9...+9
LongToSignedExcessWord	-999...+999

格式改编“有符号 32 位数”	取值范围
LongToSignedExcessDword	-9999999...+9999999
LongToSignedExcessQword	-999999999...+999999999
LongToSimaticBCDTimer	10...9990000
LongToSimaticTimer	10...9990000
LongToSimaticLTime	00:00:00.000...596.31.23.647

表格 5-6 无符号 32 位值

格式改编“无符号 32 位值”	取值范围
DwordToUnsignedDword	0...4294967295 (无转换)
DwordToUnsignedByte	0...255
DwordToUnsignedWord	0...65535
DwordToUnsignedQword	0...4294967295
DwordToSignedByte	0...127
DwordToSignedWord	0...32767
DwordToSignedDword	0...2147483647
DwordToBCDByte	0...99
DwordToBCDWord	0...9999
DwordToBCDDword	0...99999999
DwordToBCDQword	0...999999999
DwordToAikenByte	0...99
DwordToAikenWord	0...9999
DwordToAikenDword	0...99999999
DwordToAikenQword	0...999999999
DwordToExcessByte	0...99
DwordToExcessWord	0...9999
DwordToExcessDword	0...99999999
DwordToExcessQword	0...999999999
DwordToSimaticBCDTimer	10...9990000
DwordToSimaticTimer	10...9990000
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

5.3 WinCC 过程通讯

表格 5-7 IEEE 754 32 位浮点数

格式改编“IEEE 754 32 位浮点数”	取值范围
FloatToFloat	+/-3.402823e+38 (无转换)
FloatToUnsignedByte	0...255
FloatToUnsignedWord	0...65535
FloatToUnsignedDword	0...4.294967e+09
FloatToSignedByte	-128...+127
FloatToSignedWord	-32768...+32767
FloatToSignedDword	-2.147483e+09...+2.147483e+09
FloatToDouble	+/-3.402823e+38
FloatToMSBByte	-127...+127
FloatToMSBWord	-32767...+32767
FloatToMSBDword	-2.147483e+09...+2.147483e+09
FloatToBCDByte	0...99
FloatToBCDWord	0...9999
FloatToBCDDword	0...9.999999e+07
FloatToSignedBCDByte	-9...+9
FloatToSignedBCDWord	-999...+999
FloatToSignedBCDDword	-9999999...+9999999
FloatToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDDword	-7.999999e+07...+7.999999e+07
FloatToAikenByte	0...99
FloatToAikenWord	0...9999
FloatToAikenDword	0...9,999999e+07
FloatToSignedAikenByte	-9...+9
FloatToSignedAikenWord	-999...+999
FloatToSignedAikenDword	-9999999...+9999999
FloatToExcessByte	0...99
FloatToExcessWord	0...9999
FloatToExcessDword	0...9.999999e+07

格式改编“IEEE 754 32 位浮点数”	取值范围
FloatToSignedExcessByte	-9...+9
FloatToSignedExcessWord	-999...+999
FloatToSignedExcessDword	-9999999...+99999999
FloatToSimaticBCDTimer	10...9990000
FloatToS5Float	+/-1.701411e+38
FloatToSimaticTimer	10...9990000

表格 5-8 IEEE 754 64 位浮点数

格式改编“IEEE 754 64 位浮点数”	取值范围
DoubleToDouble	+/-1.79769313486231e+308 (无转换)
DoubleToUnsignedByte	0...255
DoubleToUnsignedWord	0...65535
DoubleToUnsignedDword	0...4294967295
DoubleToUnsignedQword	0...18446744073709551616
DoubleToSignedByte	-128...+127
DoubleToSignedWord	-32768...+32767
DoubleToSignedDword	-2147483648...+2147483647
DoubleToSignedQword	-9223372036854775808...+9223372036854775808
DoubleToFloat	+/-3.402823e+38
DoubleToMSBByte	-127...+127
DoubleToMSBWord	-32767...+32767
DoubleToMSBDword	-2147483647...+2147483647
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
DoubleToBCDByte	0...99
DoubleToBCDWord	0...9999
DoubleToBCDDword	0...99999999
DoubleToBCDQword	0...9999999999999999
DoubleToSignedBCDByte	-9...+9

5.3 WinCC 过程通讯

格式改编“IEEE 754 64 位浮点数”	取值范围
DoubleToSignedBCDWord	-999...+999
DoubleToSignedBCDDword	-9999999...+9999999
DoubleToSignedBCDQword	-999999999999999...+999999999999999
DoubleToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDWord	-7999...+7999
DoubleToExtSignedBCDDword	-79999999...+79999999
DoubleToExtSignedBCDQword	-799999999999999...+799999999999999
DoubleToAikenByte	0...99
DoubleToAikenWord	0...9999
DoubleToAikenDword	0...99999999
DoubleToAikenQword	0...999999999999999
DoubleToSignedAikenByte	-9...+9
DoubleToSignedAikenWord	-999...+999
DoubleToSignedAikenDword	-9999999...+9999999
DoubleToSignedAikenQword	-999999999999999...+999999999999999
DoubleToExcessByte	0...99
DoubleToExcessWord	0...9999
DoubleToExcessDword	0...99999999
DoubleToExcessQword	0...999999999999999
DoubleToSignedExcessByte	-9...+9
DoubleToSignedExcessWord	-999...+999
DoubleToSignedExcessDword	-9999999...+9999999
DoubleToSignedExcessQword	-999999999999999...+999999999999999
DoubleToSimaticBCDTimer	10...9990000
DoubleToS5Float	+/-1.701411e+38
DoubleToSimaticTimer	10...9990000

5.3.4.5 按 AS 数据类型排序的格式改编

简介

在组态外部变量时，必须将所有数字数据类型的数据改编为另一种格式。

在 WinCC 端，数据类型决定数据格式。格式调整还定义从 WinCC 格式到 AS 格式的转换。该定义对两个传输方向均适用。

从下面的选择框中选择需要的 AS 数据类型。下表将列出相应的可能格式改编及其对应的取值范围。

AS 数据类型

类型转换和值范围：

表格 5-9 AikenByte

类型转换“AikenByte”	取值范围
ByteToAikenByte	0...99
AikenByte	0...99
DoubleToAikenByte	0...99
DwordToAikenByte	0...99
FloatToAikenByte	0...99
LongToAikenByte	0...99
ShortToAikenByte	0...99
WordToAikenByte	0...99

表格 5-10 AikenWord

类型转换“AikenWord”	取值范围
ByteToAikenWord	0...255
CharToAikenWord	0...127
DoubleToAikenWord	0...9999
DwordToAikenWord	0...9999
FloatToAikenWord	0...9999
LongToAikenWord	0...9999
ShortToAikenWord	0...9999
WordToAikenWord	0...9999

5.3 WinCC 过程通讯

表格 5-11 AikenDWord

类型转换“AikenDWord”	取值范围
ByteToAikenDword	0...255
CharToAikenDword	0...127
DoubleToAikenDword	0...99999999
DwordToAikenDword	0...99999999
FloatToAikenDword	0...9,9999999e+07
LongToAikenDword	0...99999999
ShortToAikenDword	0...32767
WordToAikenDword	0...65535

表格 5-12 AikenQWord

类型转换“AikenQWord”	取值范围:
DoubleToAikenQword	0...9999999999999999
DwordToAikenQword	0...999999999

表格 5-13 BCDByte

类型转换“BCDByte”	取值范围
ByteToBCDByte	0...99
CharToBCDByte	0...99
DoubleToBCDByte	0...99
DwordToBCDByte	0...99
FloatToBCDByte	0...99
LongToBCDByte	0...99
ShortToBCDByte	0...99
WordToBCDByte	0...99

表格 5-14 BCDWord

类型转换“BCDWord”	取值范围
ByteToBCDWord	0...255
CharToBCDWord	0...127
DoubleToBCDWord	0...9999
DwordToBCDWord	0...9999
FloatToBCDWord	0...9999
LongToBCDWord	0...9999
ShortToBCDWord	0...9999
WordToBCDWord	0...9999

表格 5-15 BCDDWord

类型转换“BCDDWord”	取值范围
ByteToBCDDword	0...255
CharToBCDDword	0...127
DoubleToBCDDword	0...999999999
DwordToBCDDword	0...999999999
FloatToBCDDword	0...9.999999e+07
LongToBCDDword	0...999999999
ShortToBCDDword	0...32767
WordToBCDDword	0...65535

表格 5-16 BCDQWord

类型转换“BCDQWord”	取值范围:
DoubleToBCDQword	0...9999999999999999
DwordToBCDQword	0...999999999

5.3 WinCC 过程通讯

表格 5-17 Double

类型转换“Double”	取值范围
DoubleToDouble	+1.79769313486231e+308 (无转换)
FloatToDouble	+3.402823e+38

表格 5-18 ExcessByte

类型转换“ExcessByte”	取值范围
ByteToExcessByte	0...99
CharToExcessByte	0...99
DoubleToExcessByte	0...99
DwordToExcessByte	0...99
FloatToExcessByte	0...99
LongToExcessByte	0...99
ShortToExcessByte	0...99
WordToExcessByte	0...99

表格 5-19 ExcessWord

类型转换“ExcessWord”	取值范围
ByteToExcessWord	0...255
CharToExcessWord	0...127
DoubleToExcessWord	0...9999
DwordToExcessWord	0...9999
FloatToExcessWord	0...9999
LongToExcessWord	0...9999
ShortToExcessWord	0...9999
WordToExcessWord	0...9999

表格 5-20 ExcessDWord

类型转换“ExcessDWord”	取值范围
ByteToExcessDword	0...255
CharToExcessDword	0...127
DoubleToExcessDword	0...99999999
DwordToExcessDword	0...99999999
FloatToExcessDword	0...9.9999999e+07
LongToExcessDword	0...99999999
ShortToExcessDword	0...32767
WordToExcessDword	0...65535

表格 5-21 ExcessQWord

类型转换“ExcessQWord”	取值范围:
DoubleToExcessQword	0...9999999999999999
DwordToExcessQword	0...999999999

表格 5-22 ExtSignedBCDByte

类型转换“ExtSignedBCDByte”	取值范围
CharToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDByte	-79...+79
LongToExtSignedBCDByte	-79..+79
ShortToExtSignedBCDByte	-79...+79

表格 5-23 ExtSignedBCDWord

类型转换“ExtSignedBCDWord”	取值范围
CharToExtSignedBCDWord	-128...+127
DoubleToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDWord	-7999...+7999

5.3 WinCC 过程通讯

类型转换“ExtSignedBCDWord”	取值范围
LongToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDWord	-7999...+7999

表格 5-24 ExtSignedBCDDWord

类型转换“ExtSignedBCDDWord”	取值范围
CharToExtSignedBCDDword	-128...+127
DoubleToExtSignedBCDDword	-79999999...+79999999
FloatToExtSignedBCDDword	-7.999999e+07...+7.999999e+07
LongToExtSignedBCDDword	-79999999...+79999999
ShortToExtSignedBCDDword	-32768...+32767

表格 5-25 ExtSignedBCDQWord

类型转换“ExtSignedBCDQWord”	取值范围:
DoubleToExtSignedBCDQword	-799999999999999...+799999999999999
LongToExtSignedBCDQword	-79999999...+79999999

表格 5-26 Float

类型转换“Float”	取值范围
DoubleToFloat	+/-3.402823e+38
FloatToFloat	+/-3.402823e+38 (无转换)

表格 5-27 MSBByte

类型转换“MSBByte”	取值范围
CharToMSBByte	-127...+127
DoubleToMSBByte	-127...+127
FloatToMSBByte	-127...+127
LongToMSBByte	-127...+127
ShortToMSBByte	-127...+127

表格 5-28 MSBWord

类型转换“MSBWord”	取值范围
CharToMSBWord	-128...+127
DoubleToMSBWord	-32767...+32767
FloatToMSBWord	-32767...+32767
LongToMSBWord	-32767...+32767
ShortToMSBWord	-32767...+32767

表格 5-29 MSBDWord

类型转换“MSBDWord”	取值范围
CharToMSBDword	-128...+127
DoubleToMSBDword	-2147483647...+2147483647
FloatToMSBDword	-2.147483e+09...+2.147483e+09
LongToMSBDword	-2147483647...+2147483647
ShortToMSBDword	-32768...+32767

表格 5-30 MSBQWord

类型转换“MSBQWord”	取值范围:
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
LongToMSBQword	-2147483648...+2147483647

表格 5-31 S5Float

类型转换“S5Float”	取值范围
DoubleToS5Float	+/-1.701411e+38
FloatToS5Float	+/-1.701411e+38

5.3 WinCC 过程通讯

表格 5-32 SignedByte

类型转换“SignedByte”	取值范围
ByteToSignedByte	0...127
CharToSignedByte	-128...+127 (无转换)
DoubleToSignedByte	-128...+127
DwordToSignedByte	0...127
FloatToSignedByte	-128...+127
LongToSignedByte	-128...+127
ShortToSignedByte	-128...+127
WordToSignedByte	0...127

表格 5-33 SignedWord

类型转换“SignedWord”	取值范围
ByteToSignedWord	0...255
CharToSignedWord	-128...+127
DoubleToSignedWord	-32768...+32767
DwordToSignedWord	0...32767
FloatToSignedWord	-32768...+32767
LongToSignedWord	-32768...+32767
ShortToSignedWord	-32768...+32767 (无转换)
WordToSignedWord	0...32767

表格 5-34 SignedDWord

类型转换“SignedDWord”	取值范围
ByteToSignedDword	0...255
CharToSignedDword	-128...+127
DoubleToSignedDword	-2147483648...+2147483647
DwordToSignedDword	0...2147483647
FloatToSignedDword	-2.147483e+09...+2.147483e+09

类型转换“SignedDWord”	取值范围
LongToSignedDword	-2147483648...+2147483647 (无转换)
ShortToSignedDword	-32768...+32767
WordToSignedDword	0...65535

表格 5-35 SignedQWord

类型转换“SignedQWord”	取值范围:
DoubleToSignedQword	-9223372036854775808...+9223372036854775808
LongToSignedQword	-2147483648...+2147483647

表格 5-36 SignedAikenByte

类型转换“SignedAikenByte”	取值范围
CharToSignedAikenByte	-9...+9
DoubleToSignedAikenByte	-9...+9
FloatToSignedAikenByte	-9...+9
LongToSignedAikenByte	-9...+9
ShortToSignedAikenByte	-9...+9

表格 5-37 SignedAikenWord

类型转换“SignedAikenWord”	取值范围
CharToSignedAikenWord	-128...+127
DoubleToSignedAikenWord	-999...+999
FloatToSignedAikenWord	-999...+999
LongToSignedAikenWord	-999...+999
ShortToSignedAikenWord	-999...+999

5.3 WinCC 过程通讯

表格 5-38 SignedAikenDWord

类型转换“SignedAikenDWord”	取值范围
CharToSignedAikenDword	-128...+127
DoubleToSignedAikenDword	-9999999...+9999999
FloatToSignedAikenDword	-9999999...+9999999
LongToSignedAikenDword	-9999999...+9999999
ShortToSignedAikenDword	-32768...+32767

表格 5-39 SignedAikenQWord

类型转换“SignedAikenQWord”	取值范围:
DoubleToSignedAikenQword	-999999999999999...+999999999999999
LongToSignedAikenQword	-999999999...+999999999

表格 5-40 SignedBCDByte

类型转换“SignedBCDByte”	取值范围
CharToSignedBCDByte	-9...+9
DoubleToSignedBCDByte	-9...+9
FloatToSignedBCDByte	-9...+9
LongToSignedBCDByte	-9...+9
ShortToSignedBCDByte	-9...+9

表格 5-41 SignedBCDWord

类型转换“SignedBCDWord”	取值范围
CharToSignedBCDWord	-128...+127
DoubleToSignedBCDWord	-999...+999
FloatToSignedBCDWord	-999...+999
LongToSignedBCDWord	-999...+999
ShortToSignedBCDWord	-999...+999

表格 5-42 SignedBCDDWord

类型转换“SignedBCDDWord”	取值范围
CharToSignedBCDDword	-128...+127
DoubleToSignedBCDDword	-9999999...+9999999
FloatToSignedBCDDword	-9999999...+9999999
LongToSignedBCDDword	-9999999...+9999999
ShortToSignedBCDDword	-32768...+32767

表格 5-43 SignedBCDQWord

类型转换“SignedBCDQWord”	取值范围:
DoubleToSignedBCDQword	-9999999999999999...+9999999999999999
LongToSignedBCDQword	-2147483648...+2147483647

表格 5-44 SignedExcessByte

类型转换“SignedExcessByte”	取值范围
CharToSignedExcessByte	-9...+9
DoubleToSignedExcessByte	-9...+9
FloatToSignedExcessByte	-9...+9
LongToSignedExcessByte	-9...+9
ShortToSignedExcessByte	-9...+9

表格 5-45 SignedExcessWord

类型转换“SignedExcessWord”	取值范围
CharToSignedExcessWord	-128...+127
DoubleToSignedExcessWord	-999...+999
FloatToSignedExcessWord	-999...+999
LongToSignedExcessWord	-999...+999
ShortToSignedExcessWord	-999...+999

5.3 WinCC 过程通讯

表格 5-46 SignedExcessDWord

类型转换“SignedExcessDWord”	取值范围
CharToSignedExcessDword	-128...+127
DoubleToSignedExcessDword	-9999999...+9999999
FloatToSignedExcessDword	-9999999...+9999999
LongToSignedExcessDword	-9999999...+9999999
ShortToSignedExcessDword	-32768...+32767

表格 5-47 SignedExcessQWord

类型转换“SignedExcessQWord”	取值范围:
DoubleToSignedExcessQword	-999999999999999...+999999999999999
LongToSignedExcessQword	-999999999...+999999999

表格 5-48 SimaticCounter

类型转换“SimaticCounter”	取值范围
WordToSimaticCounter	0...999

表格 5-49 SimaticBCDCounter

类型转换“SimaticBCDCounter”	取值范围
WordToSimaticBCDCounter	0...999

表格 5-50 SimaticTimer

类型转换“SimaticTimer”	取值范围
DoubleToSimaticTimer	10...9990000
DwordToSimaticTimer	10...9990000
FloatToSimaticTimer	10...9990000
LongToSimaticTimer	10...9990000

表格 5-51 SimaticBCDTimer

类型转换“SimaticBCDTimer”	取值范围
DoubleToSimaticBCDTimer	10...9990000
DwordToSimaticBCDTimer	10...9990000
FloatToSimaticBCDTimer	10...9990000
LongToSimaticBCDTimer	10...9990000

表格 5-52 SimaticLTime

类型转换“SimaticLTime”	取值范围:
LongToSimaticLTime	00:00:00.000...596.31.23.647

表格 5-53 SimaticLTimeOfDay

类型转换“SimaticLTimeOfDay”	取值范围:
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

表格 5-54 UnsignedByte

类型转换“UnsignedByte”	取值范围
ByteToUnsignedByte	0...255 (无转换)
CharToUnsignedByte	0...127
DoubleToUnsignedByte	0...255
DwordToUnsignedByte	0...255
FloatToUnsignedByte	0...255
LongToUnsignedByte	0...255
ShortToUnsignedByte	0...255
WordToUnsignedByte	0...255

5.3 WinCC 过程通讯

表格 5-55 UnsignedWord

类型转换“UnsignedWord”	取值范围
ByteToUnsignedWord	0...255
CharToUnsignedWord	0...127
DoubleToUnsignedWord	0...65535
DwordToUnsignedWord	0...65535
FloatToUnsignedWord	0...65535
LongToUnsignedWord	0...65535
ShortToUnsignedWord	0...32767
WordToUnsignedWord	0...65535 (无转换)

表格 5-56 UnsignedDWord

类型转换“UnsignedDWord”	取值范围
ByteToUnsignedDword	0...255
CharToUnsignedDword	0...127
DoubleToUnsignedDword	0...4294967295
DwordToUnsignedDword	0...4294967295 (无转换)
FloatToUnsignedDword	0...4.294967e+09
LongToUnsignedDword	0...2147483647
ShortToUnsignedDword	0...32767
WordToUnsignedDword	0...65535

表格 5-57 UnsignedQWord

类型转换“UnsignedQWord”	取值范围:
DoubleToUnsignedQword	0...18446744073709551616
DwordToUnsignedQword	0...4294967295

5.3.4.6 二进制写入机制的原理

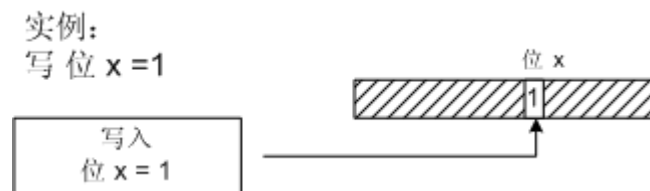
简介

在 WinCC 中，并不是所有的通讯驱动程序和其通道单元都支持直接按位或按字节访问（简称为：位/字节访问）所连接的自动化系统中的地址范围。而是需要借助二进制写入机制来完成。

位/字节访问

使用带位/字节访问的通讯驱动程序的通道单元，可直接读写期望的位或字节。

下图通过直接位/字节访问将值 1 分配给位 x。



二进制写入机制

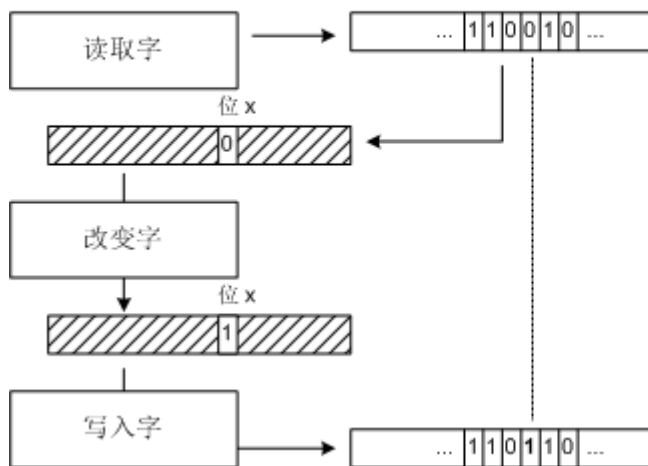
以下通讯驱动程序不支持位/字节访问，而需要对相应的通道单元使用二进制写入机制：

- Modbus Serial
- SIMATIC S5 Ethernet Layer 4
- SIMATIC S5 Programmers Port AS511
- SIMATIC S5 Serial 3964R
- SIMATIC TI Ethernet Layer 4
- SIMATIC TI Serial

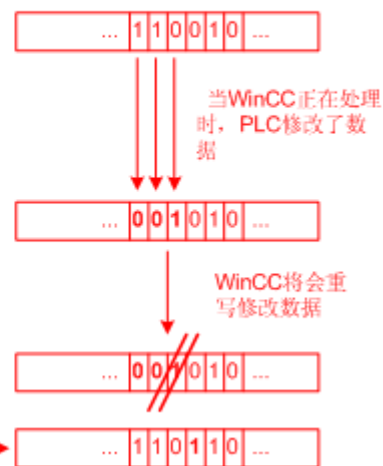
为写入位或字节，通道单元首先使用二进制写入机制读取整个数据字。随后在读取的字中更改要寻址的数据。最后再写回整个 (!) 字，而不是仅写回改变的位或字节。

下图通过二进制访问机制将值 1 分配给位 x。

实例：
写位 x = 1



问题案例



说明

如果在 WinCC 中通过二进制写入机制读取某个数据字的同时该数据字在 AS 中发生更改（请参见图“问题案例”），则当 WinCC 写回该数据字时，AS 中的更改即会丢失。

5.3.4.7 如何通过“二进制写入”组态变量

简介


如果想要为不支持/字节访问的通信驱动程序通道单元组态“二进制变量”，则必须使用对话框激活和组态二进制写入机制，否则该机制不可用。

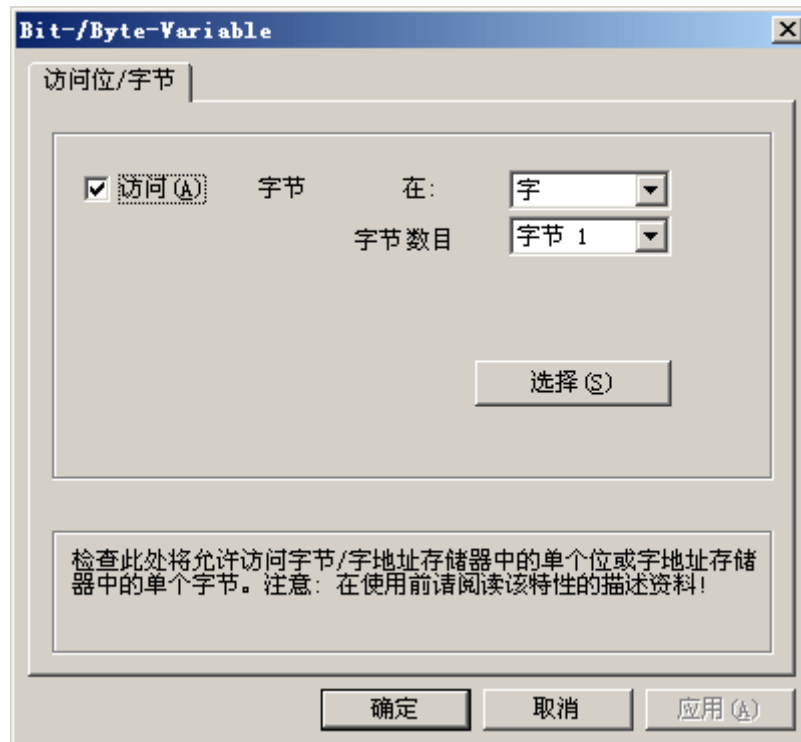
要求

- 已安装了所需的通信处理器和硬件驱动程序。
- 已安装了所期望的通信驱动程序（但它不支持/字节访问），例如，“SIMATIC S5 Ethernet Layer 4”。
- 已根据通道单元创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。


- 单击“名称”(Name) 列最上方的空单元格。
输入变量的名称。
- 选择下列数据类型之一：
 - “二进制变量”
 - “无符号 8 位数”
 - “有符号 8 位数”
- 打开“位/字节变量”(Bit/Byte tag) 对话框。
为此，单击“地址”(Address) 字段，然后单击  按钮。
“位/字节变量”(Bit/Byte tag) 对话框随即打开。使用此按钮设置位/字节访问通道在 AS 中的地址范围。



- 激活“访问一位”(Access to one bit) 复选框或“访问一个字节”(Access to one byte) 复选框并进行常规设置。
该对话框中的显示取决于在第 2 步中选择的 datatype。
- 单击“确定”(OK) 关闭对话框。

5.3.5 通过以太网连接的端口地址

端口地址

 小心
<p>以太网通讯</p> <p>使用基于以太网的通讯时，最终用户应负责其自身数据网络的安全。例如，如果有针对性的攻击导致设备过载，则功能将无法得到保证。</p>

通过以太网连接时，用户会需要端口地址信息。组态防火墙或路由器需要利用该信息。下表显示了 WinCC 应用程序默认使用的端口地址。

	端口地址 TCP/IP	端口地址 UDP
S7 通讯	102	
HTTP (通讯; 传输)	80	
HTTPS (通讯; 传输)	443	
WebServices (SOAP)	80 HTTP 443 HTTPS	
OPC-XML (CE 作为 OPC 主机)	80 HTTP 443 HTTPS	
SendEmail	25	
传输 (通过以太网; CE-Stub; PC 装载程序; PC)	2308 或者 50523	
记录 (通过以太网) CSV 文件	139, 445	137, 138
Modbus Ethernet	502	
Allen-Bradley Ethernet CIP	44818	
Allen-Bradley Ethernet CSP2	2222	

5.3.6 SIMATIC S7-PLCSIM Advanced

WinCC 支持使用 SIMATIC S7-PLCSIM Advanced 仿真软件对虚拟控制器进行仿真。

更多信息，请参见 PLCSIM 产品文档。

- Industry Online Support: SIMATIC S7-PLCSIM Advanced (<https://support.industry.siemens.com/cs/cn/zh/ps/24466/man>)

参见

Industry Online Support: SIMATIC S7-PLCSIM Advanced (<https://support.industry.siemens.com/cs/cn/zh/ps/24466/man>)

通信通道

6.1 Allen Bradley - Ethernet IP

6.1.1 WinCC 通道“Allen Bradley - Ethernet IP”

简介

通道“Allen Bradley - Ethernet IP”用于链接到 Allen-Bradley 自动化系统。其通讯使用 Ethernet IP 协议处理。

根据所用的通讯硬件，系统支持通过下列通道单元的连接：

- Allen Bradley E/IP PLC5
- Allen Bradley E/IP SLC50x
- Allen Bradley E/IP ControlLogix

6.1.2 通道单元分配

简介

必须为通道选择通道单元，以创建从 WinCC 到现有或已设计网络的连接。

通道单元分配

下表显示了通道“Allen Bradley - Ethernet IP”的通道单元如何分配到网络和自动化系统 (AS)。

通道的通道单元	通讯网络	AS
Allen Bradley E/IP PLC5	Ethernet IP	带以太网端口的 PLC-5
Allen Bradley E/IP SLC50x	Ethernet IP	带以太网端口的 SLC 500，如 SLC 5/05
Allen Bradley E/IP ControlLogix	Ethernet IP	ControlLogix 5500

6.1 Allen Bradley - Ethernet IP

6.1.3 支持的数据类型

简介

为逻辑连接定义所需变量。“Allen Bradley - Ethernet IP”通道支持以下数据类型：

- 二进制变量
- 有符号 8 位数
- 无符号 8 位数
- 有符号 16 位数
- 无符号 16 位数
- 有符号 32 位数
- 无符号 32 位数
- 浮点数 32 位 IEEE 754
- 文本变量，8 位字符集（最大长度：82 个字符）
- 文本变量，16 位字符集（最大长度：82 个字符）

6.1.4 组态通道

6.1.4.1 组态通道“Allen Bradley - Ethernet IP”

简介

在 WinCC 中，WinCC 与自动化系统 (AS) 的通讯需要逻辑连接。本节介绍如何组态“Allen Bradley - Ethernet IP”通道。

使用 TCP/IP 协议时，必须定义 AS 的 IP 地址以用于逻辑连接。IP 地址由四个以圆点分隔的数值组成。数值范围必须在 0-255 之内。

说明

超时行为

使用 TCP/IP 协议时，不会立即检测到中断的连接。回查消息可能耗时达一分钟。

可连接的控制器

可对以下 Allen-Bradley PLC 进行连接：

- Allen-Bradley ControlLogix 5500
- Allen-Bradley CompactLogix 5300
- 带以太网端口的 PLC-5
- 带以太网端口的 SLC 500，如 SLC 5/05
- MicroLogix

已发布的通讯类型

以下通讯类型已经过系统测试并发布供“Allen Bradley - Ethernet IP”使用：

- 点对点连接：
- WinCC 站与可选数量控制器之间的多点连接。

在线组态

不支持“Allen Bradley - Ethernet IP”通道的在线组态。

6.1.4.2 如何为“Allen Bradley - Ethernet IP”通道组态连接

简介

“Allen Bradley - Ethernet IP”通道可组态为三个通道单元：

- Allen Bradley E/IP ControlLogix
- Allen Bradley E/IP PLC5
- Allen Bradley E/IP SLC50x

三个通道单元的组态都相同，包括以下任务：

1. 组态连接
2. 组态变量

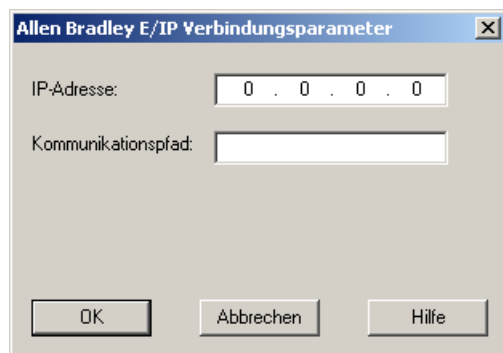
要求

- 通道“Allen Bradley - Ethernet IP”的通信驱动程序已安装并集成到项目中。

6.1 Allen Bradley - Ethernet IP

步骤

1. 在变量管理器中选择所期望的通道单元。
2. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
将创建一个新连接。
3. 选择新连接。
在“属性 - 连接”(Properties - Connection) 区域中的“常规”(General) 下输入连接名称。
4. 在快捷菜单中选择“连接参数”(Connection parameters) 连接。
“Allen Bradley E/IP 连接参数”(Allen Bradley E/IP connection parameters) 对话框随即打开。



5. 输入控制器 Ethernet/IP 模块的 IP 地址。
默认情况下，始终为以太网 IP 设备设置端口 44818。
6. 在“通信路径”(Communication path) 字段中定义从以太网模块到控制器的 CIP 路径。
可组态直接连接和通过路由的连接。
创建以太网模块与 PLC 间的逻辑连接，即使它们位于不同 CIP 网络中。
7. 单击“确定”(OK) 按钮，关闭对话框。

6.1.4.3 实例：通讯路径

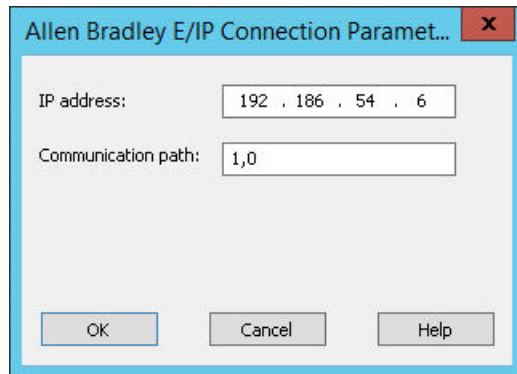
示例 1：直接连接

与位于同一个 Allen-Bradley 机架中的 PLC 连接。

通信路径：

- 1,0

数目	含义
1	代表背板连接。
0	代表 CPU 的插槽编号



示例 2：通过路由连接

与位于其它 Allen-Bradley 机架中的 PLC 连接。

在以太网上连接有两个 Allen-Bradley 机架。

两个相同的通道单元以与两个不同通道单元相同的组态方式进行组态，例如：

- WinCC - Control Logix (1) - Control Logix (2)
- WinCC - Control Logix (1) - SLC50x (1)

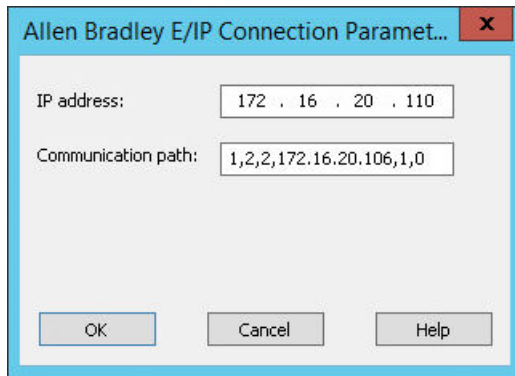
例如，虽然使用不同的协议路由到 SLC50x 或 Control Logix，但地址结构保持不变。

通信路径：

- 1,2,2,172.16.20.106,1,0

数目	含义
1	第一个模块“Control Logix (1)”的背板连接。
2	代表第 2 个以太网模块的 CPU 插槽编号。
2	代表以太网连接。
172.16.20.106	网络中远程 AB 机架的 IP 地址，例如“Control Logix (2)”模块的第 3 个以太网模块
1	第 2 个模块的背板连接，例如“Control Logix (2)”或“SLC50x (1)”。
0	CPU 的插槽编号

6.1 Allen Bradley - Ethernet IP



示例 3：通过路由连接多个模块

与位于其它 Allen-Bradley 机架中的 PLC 连接。

超过 2 个 Allen-Bradley 机架连接到以太网。

应用方案：

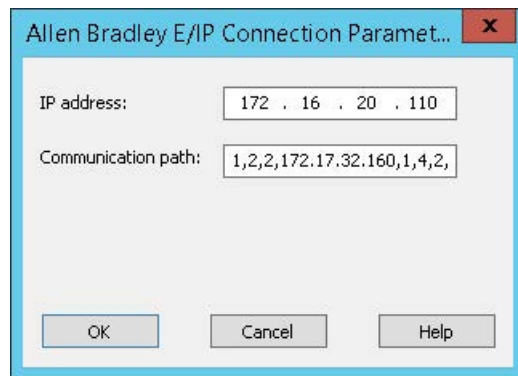
- WinCC - Control Logix (1) - Control Logix (2) - Control Logix (3)
- WinCC - Control Logix (1) - Control Logix (2) - SLC50x (1)

例如，虽然使用不同的协议路由到 SLC50x 或 Control Logix，但地址结构保持不变。

通信路径：

- 1,2,2,172.17.32.160,1,4,2,172.17.32.156,1,0

数目	含义
1	第一个模块“Control Logix (1)”的背板连接。
2	代表第 2 个以太网模块的 CPU 插槽编号。
2	代表以太网连接。
172.17.32.160	网络中远程 AB 机架的 IP 地址，例如“Control Logix (2)”模块的第 1 个以太网模块
1	第 2 个模块“Control Logix (2)”的背板连接。
4	代表第 3 个以太网模块的插槽编号。
2	代表以太网连接。
172.17.32.156	网络中另一个 AB 机架的 IP 地址，例如“Control Logix (3)”或“SLC50x (1)”模块。
1	第 3 个模块的背板连接，例如“Control Logix (3)”或“SLC50x (1)”。
0	CPU 的插槽编号



6.1.4.4 组态变量

组态变量

简介

对于通过通道“Allen Bradley - Ethernet IP”进行的 WinCC 和自动化系统 (AS) 之间的连接，可在 WinCC 中创建不同数据类型的变量。本节将列出允许的数据类型。

变量更新

在画面中同时获取来自 PLC 的变量时，“Allen Bradley - Ethernet IP”通道会尝试优化更新。然而这只能在以下条件下完成：

- 变量在相同的地址范围内。
- 在地址范围内变量尽可能靠近彼此。

如果不遵守这些建议，则会导致在刷新包含大量变量的图片时出现显著的不同。在某些情况下，可能无法维持采集周期。

组态变量时遵守以下规则可使连接获得最佳性能：

- 最多同时更新 2000 个变量。
- 将变量组合在尽可能小的范围内，最好仅在一个地址范围内。

有效数据类型

下面列出的数据类型可用来组态变量。

基本数据类型

数据类型	位地址空间
布尔	-
SInt	0-7
USInt	0-7
Int	0-15
UInt	0-15
DInt	0-31
UDInt	0-31
Real	-
字符串	-

数组

地址	有效数据类型
数组	SInt、USInt、Int、UInt、DInt、UDInt 和 Real

寻址

寻址

WinCC 中通过控制器中的地址唯一地引用变量。地址必须与 PLC 中的变量名称相对应。变量的地址由最大长度为 128 个字符的字符串定义。

使用字符进行寻址

变量寻址的有效字符：

- 字母（a 至 z, A 至 Z）
- 数字（0 到 9）
- 下划线（_）

变量地址由变量名和其它用于在 PLC 中指定变量的字符串组成。

变量名属性：

- 变量名能以下划线字符开头但不能以此结束。
- 出现连续的下划线和空格字符的字符串是无效的。
- 地址不能超过 128 个字符。

说明

保留用作变量寻址的字符不能用在程序/变量名或任何其它地址实例中。

下面列出了保留的字符：

保留的字符	功能
.	元素定界符
:	定义程序变量
,	用于寻址多维数组的定界符
/	保留用于位寻址。
[]	数组元素或数组的寻址

控制器和程序变量

“Allen-Bradley E/IP ControlLogix”允许对 PLC 变量（全局项目变量）和/或程序变量（全局程序变量）进行寻址。程序变量通过控制器中的程序名称和实际变量名称进行声明。控制器变量按名称寻址。

说明

寻址错误

当变量名称和数据类型不一致时会产生寻址错误。

在 WinCC 的地址域中定义的变量名必须与控制器中的变量名对应。WinCC 与控制器中变量的数据类型必须对应。

说明

无法直接寻址模块特定的变量，例如，输入和输出模块上的数据。而应使用控制器中的变量别名代替。

实例：在 WinCC 中无法寻址 Local:3:O.data。

如果在控制器中为 Local:3:O 定义了别名“MyOut”，则使用 WinCC 时可通过 MyOut.Data 进行寻址。

6.1 Allen Bradley - Ethernet IP

寻址语法

地址的表示法

下表定义了各种寻址方式的写入方法。

表格 6-1 访问数组、基本数据类型和结构元素

数据类型	类型	地址
基本数据类型	PLC 变量	变量名称
	程序变量	程序名:变量名
数组	PLC 变量	数组变量
	程序变量	程序名: 数组变量
位	PLC 变量	变量名/位编号
	程序变量	程序名:变量名/位编号
结构元素	PLC 变量	结构变量.结构元素
	程序变量	程序名: 结构变量.结构元素

说明

不允许对 Bool、Real 和 String 类型的数据进行位寻址，这样做将导致寻址故障。

语法描述

语法描述:

(程序名:) 变量名 ([x(, y) (, z)]) {. 变量名 ([x(, y) (, z)])} (/位编号)

- “()”定义表达式的可选的独立实例。
- “{ }”定义可选的具有多个独立实例的表达式。

地址字符串的长度不能超过 128 个字符。

寻址类型

数组元素

通过在变量编辑器中设置索引和相应的表示法寻址 PLC 中的一维、二维和三维数组中的元素。对于可有效进行元素寻址的所有基本类型的数组，数组寻址从元素“0”开始。仅对寻址到的元素执行读/写操作，而不是针对整个数组执行。

位和位变量

允许对所有基本数据类型进行位访问，但不包括 Bool、Real 和 String 类型。还可以对数组/结构元素执行位寻址。在 WinCC 中定义数据类型 Bool，用于寻址基本数据类型中的位和位变量。

将使用“Ix”或“I0x”（x = 位编号）寻址一位位编号。位编号最多由两位数字定义。

说明

通过数据类型 SInt、Int 和 DInt 中的“Bool”数据类型，在更改指定位后，整个变量将再次写入 PLC。同时，对于变量中的其它位是否已经改变，不进行任何检查。因此，PLC 只能对指定变量进行读访问。

结构

用户定义的数据类型由结构创建。这些结构将组合不同数据类型的变量。结构可以包含基本类型、数组或其它结构。在 WinCC 中，只有基本数据类型可作为结构元素进行寻址，而并非整个结构。

结构元素

结构元素通过结构和所需结构元素的名称进行寻址。结构及其元素的名称间用点分隔。除了基本数据类型外，结构元素还可表示数组或其它结构。只能将一维数组用作结构元素。

说明

结构的嵌套深度仅受到地址最大长度 128 个字符的限制。

寻址实例

寻址实例表

下表显示了控制变量的基本寻址实例。还可以通过组合形成其它寻址模式。

类型	类型	地址
常规数据类型	PLC 变量	变量名称
	程序变量	程序:变量名
数组	访问二维数组中的元素	Arraytag[Dim1,Dim2]
	结构数组（一维）中的元素	Arraytag[Dim1].structureelement
	基本类型数组（二维）元素中的位	Arraytag[Dim1,Dim2]/Bit
结构	结构中的数组	Structuretag.arraytag
	子结构数组元素中的位	Structuretag.structure2.arraytag [element]/bit

说明

通过在地址前使用来自 PLC 的程序名并用冒号与地址分隔来寻址程序变量。

实例： Programname:arraytag[Dim1,Dim2]

访问数组元素

类型	地址
PLC 变量	Arraytag[Dim1]
	Arraytag[Dim1,Dim2]
	Arraytag[Dim1,Dim2,Dim3]
程序变量	Programname:arraytag[Dim1]
	Programname:arraytag[Dim1,Dim2]
	Programname:arraytag[Dim1,Dim2,Dim3]

如何组态用于 Allen Bradley E/IP ControlLogix 通道单元的变量


简介

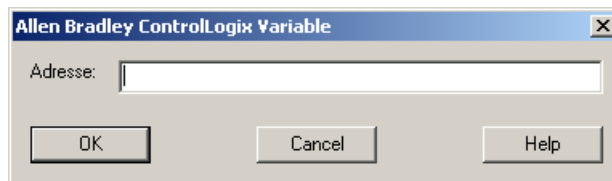
本节介绍如何在自动化系统 (AS) 地址范围内为通道单元“Allen Bradley E/IP ControlLogix”组态变量。

要求

- 通道“Allen Bradley - Ethernet IP”必须已集成在项目中。
- 必须已在“Allen Bradley E/IP ControlLogix”通道单元中创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。输入变量的名称。
4. 在“数据类型”(Data type) 字段中选择所需的数据类型。
5. 打开“Allen Bradley ControlLogix 变量”(Allen Bradley ControlLogix Tag) 对话框。为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在“地址”域中输入变量在 AS 中的地址。

说明

寻址错误

当变量名称和数据类型不一致时会产生寻址错误。

在 WinCC 的地址域中定义的变量名必须与控制器中的变量名对应。WinCC 中变量的数据类型必须与控制器中的数据类型对应。

7. 单击“确定”(OK) 按钮，关闭对话框。

如何组态用于 Allen Bradley E/IP PLC5 或 SLC50x 的按位访问变量

简介


本节说明如何以按位访问方式组态自动化系统 (AS) 中地址区域的变量。

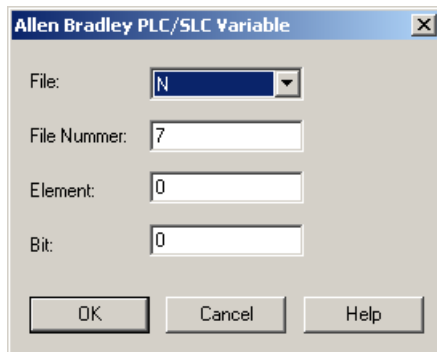
6.1 Allen Bradley - Ethernet IP

要求

- 通道“Allen Bradley - Ethernet IP”必须已集成在项目中。
- 必须已在“Allen Bradley E/IP PLC5x”或“Allen Bradley E/IP SLC50x”通道单元中创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data type) 字段中选择所需的数据类型。
5. 打开“Allen Bradley PLC/SLC 变量”(Allen Bradley PLC/SLC Tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在“文件”域中选择地址范围。条目 N、R、C、T、B、S、I、O、D、A 和 ST 可用。
7. 如与建议的文件号不同，输入“文件号”。
8. 输入“元素”。
9. 根据“文件”域中的设置，定义要寻址的“位”或选择“位（八进制）”值或“子”值。
10. 单击“确定”(OK) 按钮，关闭对话框。

如何组态用于 Allen Bradley E/IP PLC5 或 SLC50x 的按字节访问变量


简介

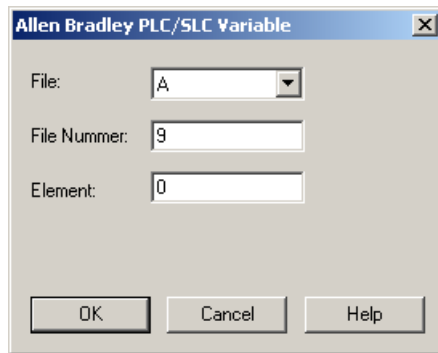
本节说明如何以按字节访问方式组态自动化系统中地址区域的变量。

要求

- 通道“Allen Bradley - Ethernet IP”必须已集成在项目中。
- 必须已在“Allen Bradley E/IP PLC5x”或“Allen Bradley E/IP SLC50x”通道单元中创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data type) 字段中选择所需的数据类型。
5. 打开“Allen Bradley PLC/SLC 变量”(Allen Bradley PLC/SLC Tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在“文件”域中选择 A 或 ST 地址范围。
7. 如与建议的文件号不同，输入“文件号”。
8. 输入“元素”。
9. 单击“确定”(OK) 按钮，关闭对话框。

如何组态用于 Allen Bradley E/IP PLC5 或 SLC50x 的按字访问变量

简介

本节介绍如何以按字访问方式组态自动化系统 (AS) 中地址区域的变量。


要求

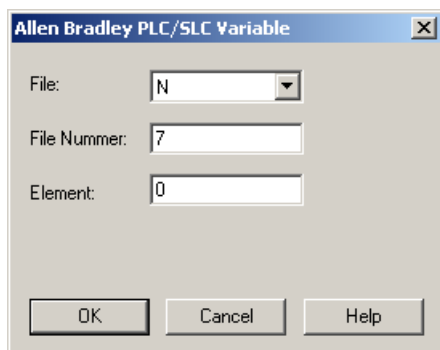
- 通道“Allen Bradley - Ethernet IP”必须已集成在项目中。
- 必须已在“Allen Bradley E/IP PLC5x”或“Allen Bradley E/IP SLC50x”通道单元中创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。

6.1 Allen Bradley - Ethernet IP

- 在“数据类型”(Data type) 字段中选择所需的数据类型。
- 打开“Allen Bradley PLC/SLC 变量”(Allen Bradley PLC/SLC Tag) 对话框。为此，请单击“地址”(Address) 字段，然后单击  按钮。



- 在“文件”域中选择地址范围。条目 N、R、C、T、B、S、I、O、D、A 和 ST 可用。
- 如与建议的文件号不同，输入“文件号”。设置“S”的文件号无法更改。
- 输入“元素”。
- 如果显示“子”域，选择其中一个值。这取决于在“文件”域中进行的设置。
- 单击“确定”(OK) 按钮，关闭对话框。

如何组态用于 Allen Bradley E/IP PLC5 or SLC50x 的文本变量

简介


本节介绍如何以按字访问方式组态自动化系统 (AS) 中地址区域的变量。

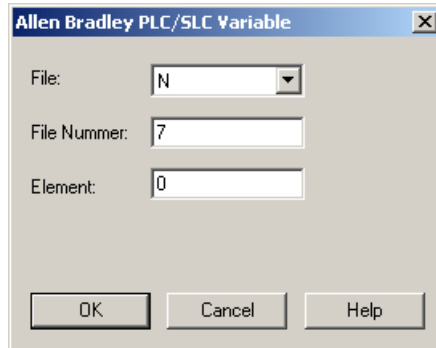
要求

- 通道“Allen Bradley - Ethernet IP”必须已集成在项目中。
- 必须已在“Allen Bradley E/IP PLC5x”或“Allen Bradley E/IP SLC50x”通道单元中创建连接。

步骤

- 选择要为其组态变量的连接。
- 单击表格区下方的“变量”(Tags) 选项卡。
- 单击“名称”(Name) 列最上方的空单元格。输入变量名。
- 在“数据类型”(Data type) 字段中选择所需的数据类型。

5. 打开“Allen Bradley PLC/SLC 变量”(Allen Bradley PLC/SLC Tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在“文件”域中选择 A 或 ST 地址范围。
7. 如与建议的文件号不同，输入“文件号”。设置“S”的文件号无法更改。
8. 输入“元素”。
9. 如果显示“子”域，选择其中一个值。这取决于在“文件”域中进行的设置。
10. 单击“确定”(OK) 按钮，关闭对话框。

6.2 Mitsubishi 以太网

6.2.1 WinCC 通道“Mitsubishi 以太网”

简介

“Mitsubishi 以太网”通道用于 WinCC 站与 Mitsubishi 控制器间的通信。
通过 MELSEC 通信协议（MC 协议）进行通信。

通道单元

WinCC 通道“Mitsubishi 以太网”有以下通道单元：

- Mitsubishi FX3U 系列
- Mitsubishi Q 系列
- Mitsubishi iQ-R 系列
- Mitsubishi iQ-F 系列

6.2.2 所支持的数据类型

简介

定义实现 WinCC 与相连控制器之间的逻辑连接所需的变量。

“Mitsubishi 以太网”通道支持以下数据类型：

- 二进制变量
- 有符号 16 位数
- 无符号 16 位数
- 有符号 32 位数
- 无符号 32 位数
- 浮点数 32 位 IEEE 754
- 浮点数 64 位 IEEE 754
- 文本变量 8 位字符集

- 文本变量 16 位字符集
- 原始数据变量

6.2.3 组态通道

6.2.3.1 组态“Mitsubishi 以太网”通道

简介

在 WinCC 中，WinCC 与自动化系统 (AS) 的通讯需要逻辑连接。

本节说明如何组态“Mitsubishi 以太网”通道。

组态通道

组态“Mitsubishi 以太网”(Mitsubishi Ethernet) 通道时，需要执行以下步骤：

1. 组态连接。
2. 组态变量。

说明

控制器中的连接组态

使用 Mitsubishi 控制器时，还必须组态控制器中的连接。

为此，请使用制造商提供的相应文档。

在线组态

“Mitsubishi 以太网”通道支持在线组态变量和连接。

支持的 Mitsubishi 控制器

可以为以下 Mitsubishi 控制器组态逻辑连接：

- MELSEC FX3U 系列
- MELSEC 系统 Q
- MELSEC 系统 iQ-R
- MELSEC 系统 iQ-F

6.2 Mitsubishi 以太网

对于这两个控制器系列，连接和变量的组态步骤完全相同。仅仅是特定控制器系列可用地址类型的组态不一样。

仅 MELSEC 系统 Q 和 MELSEC 系统 iQ-R 系列的型号支持信息路由。

协议

可以使用 TCP/IP 或 UDP/IP 作为传输协议，建立与 AS 的连接。

组态“Mitsubishi 以太网”通道的逻辑连接：

- 输入 AS 的 IP 地址和 IP 端口号
IP 地址由四个以圆点分隔的数值组成。数值必须在 0 到 255 的范围内。
- 选择 UDP 或 TCP 作为传输协议

说明

超时行为

使用 TCP/IP 协议时，不会立即检测到中断的连接。

反馈时间可能更长，具体时间则取决于操作系统。

参见

如何组态“Mitsubishi FX3U 系列”通道单元连接 (页 210)

如何组态“Mitsubishi Q 系列”通道单元连接 (页 212)

如何组态“Mitsubishi iQ-R 系列”通道单元连接 (页 213)

组态变量 (页 216)

6.2.3.2 如何组态“Mitsubishi FX3U 系列”通道单元连接

简介

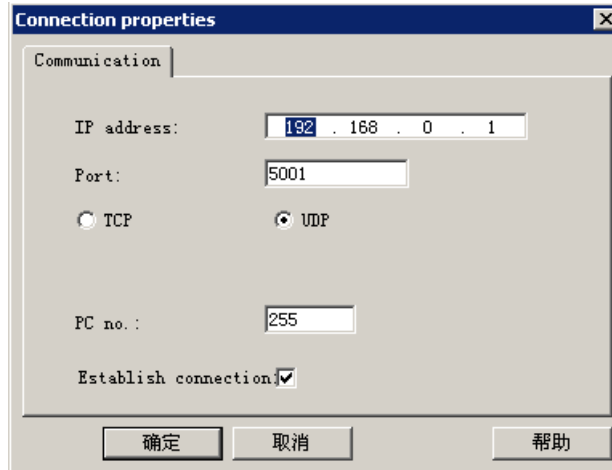
本节介绍如何组态“Mitsubishi FX3U 系列”通道单元的连接。

要求

- 已安装“Mitsubishi 以太网”通道的通信驱动程序，并将其集成到项目中。

步骤

1. 在变量管理的导航区域中，从“Mitsubishi 以太网”(Mitsubishi Ethernet) 通信驱动程序树结构下选择通道单元“Mitsubishi FX3U 系列”(Mitsubishi FX3U Series)。
2. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
3. 输入连接的名称。
4. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。将打开“连接属性”(Connection properties) 对话框。



5. 输入控制器的 IP 地址。
6. 输入想要用于 TCP/IP 连接的端口。
数值的有效范围是 0 到 65535。
7. 选择要使用的“TCP”或“UDP”协议。

说明

建议：UDP

建议使用默认协议“UDP”。使用该协议时，超时后响应可能更佳。
只有在无法使用 UDP 时，才使用 TCP。

8. 输入 PC 编号。
如果不想输入 PC 编号，则必须输入数值 255 或 0。
9. 要建立连接，请选择“建立连接”。
10. 在各种情况下，单击“确定”(OK) 均可关闭对话框。

参见

组态“Mitsubishi 以太网”通道 (页 209)

如何组态变量 (页 218)

6.2 Mitsubishi 以太网

6.2.3.3 如何组态“Mitsubishi Q 系列”通道单元连接

简介

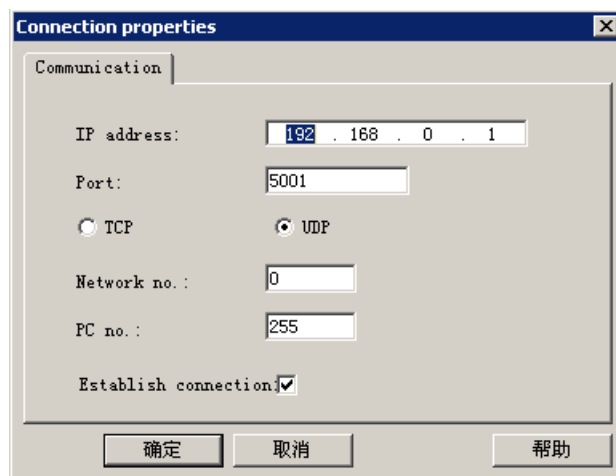
本节介绍如何组态“Mitsubishi Q 系列”通道单元的连接。

要求

- 已安装“Mitsubishi 以太网”通道的通信驱动程序，并将其集成到项目中。

步骤

1. 在变量管理的导航区域中，从“Mitsubishi 以太网”(Mitsubishi Ethernet) 通信驱动程序树结构下选择通道单元“Mitsubishi Q 系列”(Mitsubishi Q series)。
2. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
3. 输入连接的名称。
4. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。将打开“连接属性”(Connection properties) 对话框。



5. 输入控制器的 IP 地址。
6. 输入想要用于 TCP/IP 连接的端口。
数值的有效范围是 0 到 65535。
7. 选择要使用的“TCP”或“UDP”协议。

说明

建议：UDP

建议使用默认协议“UDP”。使用该协议时，超时而响应可能更佳。

只有在无法使用 UDP 时，才使用 TCP。

8. 输入网络编号。
默认设置为 0。
9. 输入 PC 编号。
如果不想输入 PC 编号，则必须输入数值 255 或 0。
10. 要建立连接，请选择“建立连接”。
11. 单击“确定”，关闭各对话框。

参见

组态“Mitsubishi 以太网”通道 (页 209)

如何组态变量 (页 218)

6.2.3.4 如何组态“Mitsubishi iQ-R 系列”通道单元连接

简介

本节介绍如何组态“Mitsubishi iQ-R 系列”通道单元的连接。

要求

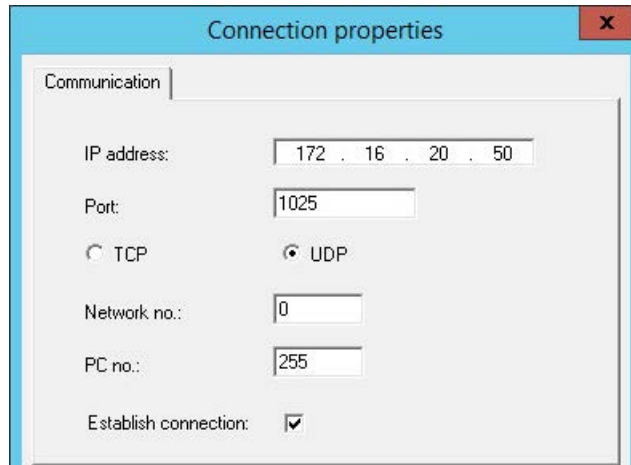
- 已安装“Mitsubishi 以太网”通道的通信驱动程序，并将其集成到项目中。

步骤

1. 在变量管理的导航区域中，从“Mitsubishi 以太网”(Mitsubishi Ethernet) 通信驱动程序树结构下选择通道单元“Mitsubishi iQ-R 系列”(Mitsubishi iQ-R series)。
2. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
3. 输入连接的名称。

6.2 Mitsubishi 以太网

- 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。
将打开“连接属性”(Connection properties) 对话框。



- 输入控制器的 IP 地址。
- 输入想要用于 TCP/IP 连接的端口。
数值的有效范围是 0 到 65535。默认情况下设置端口 1025。
- 选择要使用的“UDP”或“TCP”协议。

说明**建议：UDP**

建议使用默认协议“UDP”。使用该协议时，超时后响应可能更佳。
只有在无法使用 UDP 时，才使用 TCP。

- 输入网络编号。
默认设置为 0。
- 输入 PC 编号。
如果不想输入 PC 编号，则必须输入数值 255 或 0。
- 要建立连接，请选择“建立连接”。
- 单击“确定”，关闭各对话框。

参见

组态“Mitsubishi 以太网”通道 (页 209)

如何组态变量 (页 218)

6.2.3.5 如何组态“Mitsubishi iQ-F 系列”通道单元的连接

简介

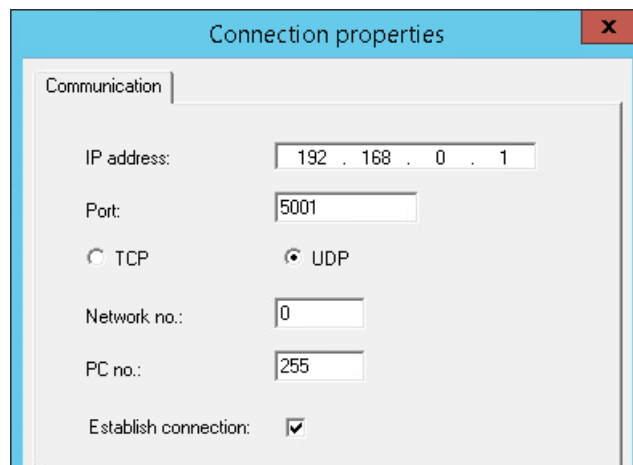
本节介绍如何组态“Mitsubishi iQ-R 系列”通道单元的连接。

要求

- 已安装“Mitsubishi 以太网”通道的通信驱动程序，并将其集成到项目中。

步骤

1. 在变量管理的导航区域中，从“Mitsubishi 以太网”(Mitsubishi Ethernet) 通信驱动程序树结构下选择通道单元“Mitsubishi iQ-F 系列”(Mitsubishi iQ-F Series)。
2. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
3. 输入连接的名称。
4. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。将打开“连接属性”(Connection properties) 对话框。



5. 输入控制器的 IP 地址。
6. 输入想要用于 TCP/IP 连接的端口。
数值的有效范围是 0 到 65535。默认情况下设置端口 1025。
7. 选择要使用的“UDP”或“TCP”协议。

说明

建议：UDP

建议使用默认协议“UDP”。使用该协议时，超时后响应可能更佳。
只有在无法使用 UDP 时，才使用 TCP。

6.2 Mitsubishi 以太网

8. 输入网络编号。
默认设置为 0。
9. 输入 PC 编号。
如果不想输入 PC 编号，则必须输入数值 255 或 0。
10. 要建立连接，请选择“建立连接”。
11. 单击“确定”，关闭各对话框。

参见

组态“Mitsubishi 以太网”通道 (页 209)

如何组态变量 (页 218)

6.2.3.6 组态变量

组态变量

简介

对于通过“Mitsubishi 以太网”通道进行的 WinCC 和自动化系统之间的连接，可在 WinCC 中创建不同数据类型的变量。

以下章节描述如何组态变量。

地址类型、地址范围和数据类型

下表列出了组态变量和结构变量时可使用的地址类型、地址范围和数据类型。

此外，指定了由相应地址类型支持的自动化系统 (AS)：

- FX: MELSEC FX3U 系列
- Q: MELSEC 系统 Q
- iQR: MELSEC 系统 iQ-R
- iQF: MELSEC 系统 iQ-F

说明

WinCC 数据类型依赖于地址类型

如果要使用特定的 WinCC 数据类型，必须考虑所选的地址类型。例如，地址类型“D”不支持 WinCC 数据类型“二进制”。

地址类型	代码	显示	数据类型	所支持的 AS
继电器				
链接继电器	B	十六进制	位	Q、iQR、iQF
输入继电器（直接）	DX	十六进制	位	Q、iQR
输出继电器（直接）	DY	十六进制	位	Q、iQR
锁存继电器	L	十进制	位	Q、iQR、iQF
特殊链接继电器	SB	十六进制	位	Q、iQR、iQF
边沿继电器	V	十进制	位	Q、iQR
输入继电器	X	十六进制（FX：八进制）	位	FX、Q、iQR、iQF
输出继电器	Y	十六进制（FX：八进制）	位	FX、Q、iQR、iQF
标志				
错误标记 (Annunciator)	F	十进制	位	Q、iQR、iQF
标志 (Internal relay)	M	十进制	位	FX、Q、iQR、iQF
步标记 (Step relay)	S	十进制	位	Q、iQF
诊断标志 (Special relay)	SM	十进制	位	Q、iQR、iQF
寄存器				
数据寄存器	D	十进制	字	FX、Q、iQR、iQF
扩展寄存器 (File register)	R	十进制	16 位	FX、Q、iQF
扩展寄存器	R	十进制	字	iQR
刷新数据寄存器	RD	十进制	字	iQR
诊断寄存器 (Special register)	SD	十进制	字	Q、iQR、iQF
特殊链接寄存器	SW	十六进制	字	Q、iQR、iQF
链接寄存器	W	十六进制	字	Q、iQR、iQF
索引寄存器	Z	十进制	字	Q、iQR、iQF
长索引寄存器	LZ	十进制	双字	iQF
文件寄存器（序列号访问）	ZR	十六进制	字	Q、iQR
计数器				
计数器/线圈	CC	十进制	位	Q、iQR、iQF
计数器/当前值	CN	十进制	字	FX、Q、iQR、iQF

6.2 Mitsubishi 以太网

地址类型	代码	显示	数据类型	所支持的 AS
计数器/触点	CS	十进制	位	FX、Q、iQR、iQF
长时计数器（线圈）	LCC	十进制	位	iQR、iQF
长时计数器（当前值）	LCN	十进制	双字	iQR、iQF
长时计数器（触点）	LCS	十进制	位	iQR、iQF
定时器				
长时保持型定时器（当前值）	LSTN	十进制	双字	iQR
长时定时器（当前值）	LTN	十进制	双字	iQR
保持型定时器（线圈）	SC	十进制	位	Q、iQF
保持型定时器（当前值）	SN	十进制	字	Q、iQF
保持型定时器（触点）	SS	十进制	位	Q、iQF
保持型定时器（线圈）	STC	十进制	位	iQR
保持型定时器（当前值）	STN	十进制	字	iQR
保持型定时器（触点）	STS	十进制	位	iQR
定时器（OUT 线圈）	TC	十进制	位	Q、iQR、iQF
定时器（当前值）	TN	十进制	字	FX、Q、iQR、iQF
定时器（触点）	TS	十进制	位	FX、Q、iQR、iQF

参见

如何组态变量 (页 218)

如何组态变量


简介

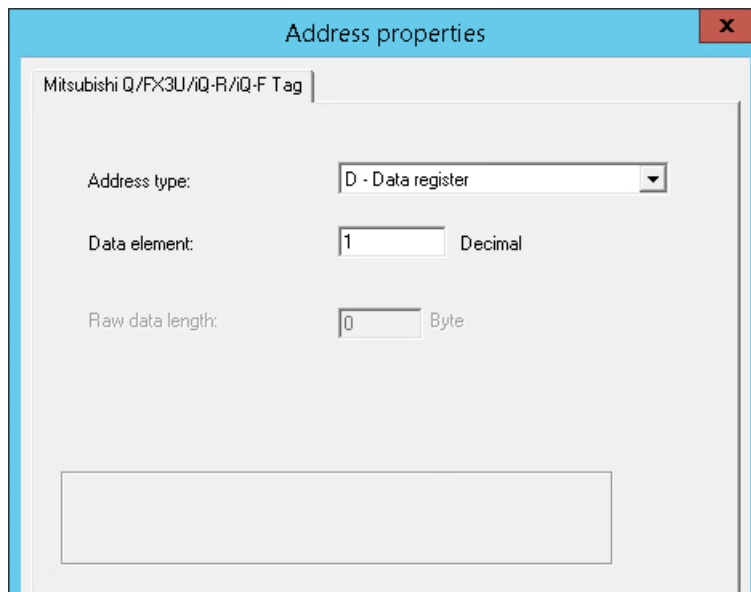
本节介绍如何组态适合自动化系统 (AS) 中的地址范围的变量。

要求

- “Mitsubishi 以太网”通道已集成到项目中。
- 在其中一个通道单元中创建连接：
 - Mitsubishi FX3U 系列
 - Mitsubishi Q 系列
 - Mitsubishi iQ-R 系列
 - Mitsubishi iQ-F 系列

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags)选项卡。
3. 单击“名称”(Name)列最上方的空单元格。
输入变量的名称。
4. 在“数据类型”(Data type)字段中选择所需的数据类型。
5. 打开“地址属性”(Address properties)对话框。
为此，单击“地址”(Address)字段，然后单击  按钮。



6. 设置地址类型。
7. 输入相应的数据元素。
该值取决于控制器的组态。
8. 必要的话，输入原始数据长度。
9. 单击“确定”，关闭“地址属性”对话框。
变量地址将显示在“变量属性”对话框的“地址”字段中。
地址将调整为 AS 的数据格式。

6.2 Mitsubishi 以太网

参见

如何组态“Mitsubishi FX3U 系列”通道单元连接 (页 210)

如何组态“Mitsubishi Q 系列”通道单元连接 (页 212)

如何组态“Mitsubishi iQ-R 系列”通道单元连接 (页 213)

组态变量 (页 216)

6.3 Modbus TCPIP

6.3.1 “Modbus TCP/IP”通道

简介

“Modbus TCPIP”通道用于 WinCC 站和支持 Modbus 的 PLC 之间通过以太网的通讯。其通讯使用 Modbus TCP/IP 协议处理。

通道单元

“Modbus TCPIP”通道附带“Modbus TCP/IP Unit #1”通道单元。

6.3.2 支持的数据类型

简介

为与已连接控制器的逻辑连接定义所需变量。“Modbus TCPIP”通道支持以下数据类型：

- 二进制变量
- 有符号 16 位数
- 无符号 16 位数
- 有符号 32 位数
- 无符号 32 位数
- 浮点数 32 位 IEEE 754
- 文本变量 8 位字符集
- 文本变量 16 位字符集

6.3 Modbus TCPIP

6.3.3 组态通道

6.3.3.1 组态“Modbus TCPIP”通道

简介

在 WinCC 中，WinCC 与自动化系统 (AS) 的通讯需要逻辑连接。本节说明与“Modbus TCP/IP Unit #1”通道单元之间的通讯。所有连接特定的参数均在安装期间定义。

使用 TCP/IP 协议时，必须定义 AS 的 IP 地址以用于逻辑连接。IP 地址由四个以圆点分隔的数值组成。数值范围必须在 0-255 之内。

说明

超时行为

使用 TCP/IP 协议时，不会立即检测到中断的连接。回查消息可能耗时达一分钟。

可用的 Modbus TCPIP 通讯方法

下列通讯类型已经过系统测试并得到认证：

- 点对点通讯
- WinCC 站与可选数量控制器之间的多点连接。

说明

因为 WinCC 站作为 Modbus 主机工作，所以不可能通过桥接器将 WinCC 站集成在 Modbus 网络中。

在线组态

不支持在线组态。

6.3.3.2 如何组态连接

简介

组态通道“Modbus TCPIP”需要以下步骤：

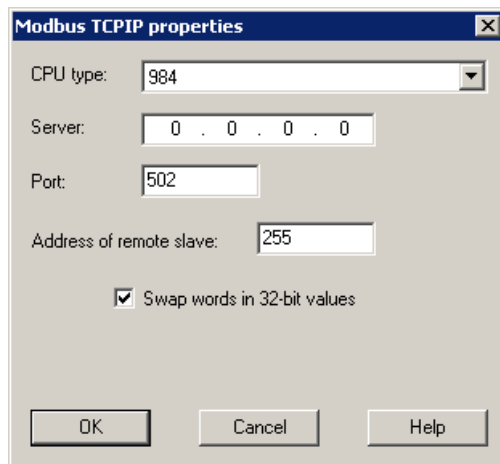
1. 组态连接
2. 组态变量

要求

- 通道“Modbus TCP/IP”的通信驱动程序已安装并集成到项目中。

操作步骤

1. 在变量管理的导航区域中，从“Modbus TCP/IP”通信驱动程序树结构下选择通道单元“Modbus TCP/IP Unit#1”。
2. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
3. 输入连接的名称。
4. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。随即将打开“Modbus TCP/IP 属性”(Modbus TCP/IP properties) 对话框。



5. 在“CPU 类型”下选择已连接的 Modicon 控制器。
以下 CPU 可供选择：
 - 984
将该 CPU 类型用于 CPU 984（CPU 984A、984B 和 984X 除外）
 - Modicon Compact、Modicon Quantum 和 Modicon Momentum
 - Modicon Premium 和 Modicon Micro
6. 在“服务器”域中输入控制器的 IP 地址。
7. 在“端口”域中输入用于 TCP/IP 连接的端口。
Modbus TCP/IP 连接的默认端口为 502。
8. 如果正在使用网桥，请在“远程从站的地址”(Address of remote slave) 字段中输入远程控制器的地址。
如果未使用网桥，则必须输入默认值 255 或 0 作为地址。

说明

因为 WinCC 站作为 Modbus 主机工作，所以不可能通过桥接器将 WinCC 站集成在 Modbus 网络中。

6.3 Modbus TCPIP

9. 要交换 32 位过程值中 16 位寄存器的顺序，请选中“交换 32 位值中的字”(Swap words in 32-bit values)。
10. 单击“确定”(OK) 关闭对话框。

6.3.3.3 组态变量

组态变量

简介

对于通过通道“Modbus TCPIP”进行的 WinCC 和自动化系统 (AS) 之间的连接，可在 WinCC 中创建不同数据类型的变量。

以下章节描述如何组态变量。AS 中的数据范围与 WinCC 变量的数据类型寻址方式不同。

使用 Modbus TCP/IP 协议进行变量更新

在画面中同时获取来自控制器的变量时，Modbus TCP/IP 通道会尝试优化更新。然而这只能在以下条件下完成：

- 变量在相同的地址范围内。
- 在地址范围内变量尽可能靠近彼此。

如果不遵守这些建议，则会导致在刷新包含大量变量的图片时出现显著的不同。在某些情况下，可能无法维持采集周期。

组态变量时遵守以下规则可使连接获得最佳性能：

- 最多同时更新 2000 个变量。
- 将变量组合在尽可能小的范围内，最好仅在一个地址范围内。

控制器中的数据类型和地址范围

下表列出了组态变量和结构化变量时可使用的数据类型和地址范围。

名称	使用 CPU Premium/ Micro 时的区域	使用 CPU 984、 Compact、 Quantum 和 Momentum 时的区 域	数据类型
线圈 (离散量输出)	%M ¹⁾	0x	位
离散量输入	(%I) – Premium/ Micro 未实现	1x	位
输入寄存器	(%IW) – Premium/ Micro 未实现	3x	Bit、 +/- Int、Int
暂停寄存器 (输出)	%MW	4x	Bit ²⁾ 、 +/-Int、Int、 +/-Double、Double、 Float、ASCII
扩展内存 (仅适用于“Quantum/Momentum”CPU)	--	6x	Bit ²⁾ 、 +/-Int、Int、 +/-Double、Double、 Float、ASCII

1) 由于外部控制器的系统特征，无法访问地址区域末尾的最后 x 位。

2) 对于写访问，请注意：

通过“4x”、“6x”和“%MW”区域中的“bit”数据类型，在更改指定位后，整个字将写回 PLC。但期间并不检查字中的其它位是否已改变。因此，PLC 只对指定字进行读访问。

984、Compact、Quantum 和 Momentum 系列控制器所使用的标准位计数方法 (16 LSB - 1 MSB) 只能用于那些在“变量”编辑器中显示数据类型“bit”的 CPU。位位置分配如下：

变量计数	左字节								右字节							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
									0	1	2	3	4	5	6	

6.3 Modbus TCPIP

当在 WinCC 中的其它位置输入位编号时，WinCC 适用的位分配如下 (0 LSB - 15 MSB):

位位置的计数方法	左字节								右字节							
在 WinCC 中，可组态:	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
	5	4	3	2	1	0										

此位计数方法也适用于 Modicon Premium 和 Modicon Micro 控制器

“有符号数”的格式

占位符“+/-”适用于“有符号整数”和“有符号长整数”两种数据类型。

参见

如何组态按位访问的变量 (页 226)

如何组态按字访问的变量 (页 227)

如何组态文本变量 (页 228)

如何组态按位访问的变量

简介


本节说明如何以按位访问方式组态自动化系统 (AS) 中地址区域的变量。

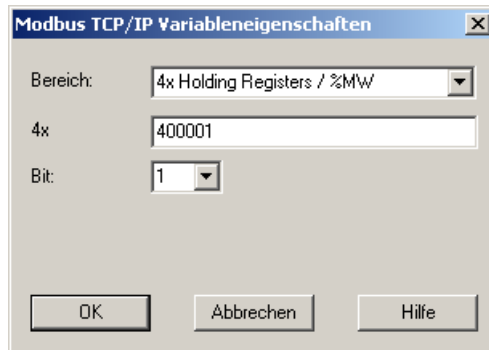
要求

- 通道“Modbus TCPIP”必须集成在项目中。
- 必须已在“Modbus TCP/IP Unit #1”通道单元中创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data Type) 字段中，设置“二进制变量”(Binary tag) 数据类型。

5. 打开“Modbus TCP/IP 变量属性”(Modbus TCP/IP Tag Properties) 对话框。为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在相应的地址字段（例如“4x”）中输入变量的地址。该值取决于控制器的组态。
7. 如必要，在“位”域中输入位的地址。条目是否可用取决于您在“来源区域”域中的选择。
8. 如果已在“区域”域中设置了值“6x 扩展的内存”，则为“文件”选择一个值。
9. 单击“确定”(OK) 关闭对话框。

说明

关闭“Modbus TCP/IP 变量属性”对话框后，将在“变量属性”对话框的“地址”域中显示控制器中变量的内部地址。此地址已按照 AS 数据格式进行调整，因此可能会与输入的地址不同。

参见

如何组态文本变量 (页 228)

如何组态按字访问的变量


简介

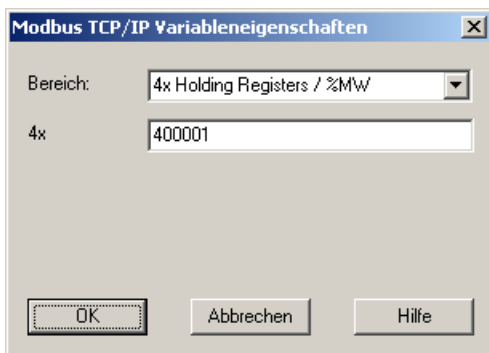
本节介绍如何以按字访问方式组态自动化系统 (AS) 中地址区域的变量。

要求

- 通道“Modbus TCP/IP”必须集成在项目中。
- 必须已在“Modbus TCP/IP Unit #1”通道单元中创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”域中，将数据类型设置为“无符号 16 位数”。
5. 打开“Modbus TCP/IP 变量属性”(Modbus TCP/IP tag properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在相应的地址字段（例如“4x”）中输入变量的地址。该值取决于控制器的组态。
7. 单击“确定”(OK) 按钮，关闭上述两个对话框。

说明

关闭“Modbus TCP/IP 变量属性”对话框后，将在“变量属性”对话框的“地址”域中显示控制器中变量的内部地址。此地址已按照 AS 数据格式进行调整，因此可能会与输入的地址不同。

如何组态文本变量


简介

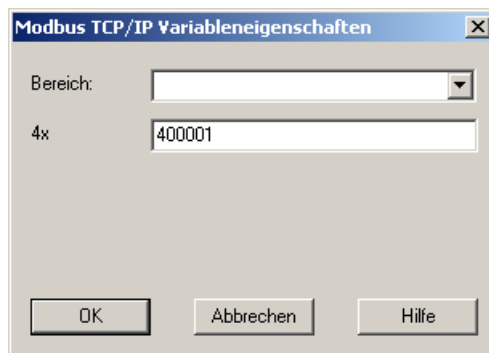
本节介绍如何以按字访问方式组态自动化系统 (AS) 中地址区域的变量。

要求

- 通道“Modbus TCP/IP”必须集成在项目中。
- 必须已在“Modbus TCP/IP Unit #1”通道单元中创建连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”域中，将“文本变量 8 位字符集”设置为数据类型。
5. 打开“Modbus TCP/IP 变量属性”(Modbus TCP/IP tag properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在地址域“4x”中，输入变量的地址。该值取决于控制器的组态。
7. 单击“确定”(OK) 按钮，关闭上述两个对话框。

6.4 OPC 通道

6.4.1 WinCC OPC 通道

简介

WinCC 可以用作 OPC 服务器和 OPC 客户端。OPC 通道是 WinCC 的 OPC 客户端应用程序。OPC 通信驱动程序可用作 OPC DA 客户端、OPC XML 客户端和 OPC UA 客户端。在“OPC UA 通道”下可找到有关 OPC UA 客户端的文档。

下列 OPC 组件会自动安装：

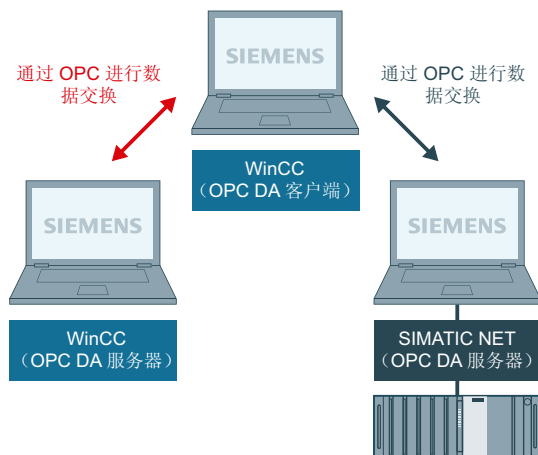
- OPC 通讯驱动程序
- OPC 条目管理器

可能的应用

WinCC 作为 OPC DA 客户端

将 WinCC 用作 OPC DA 客户端时，OPC 通道必须添加到 WinCC 项目中。在 WinCC OPC DA 客户端的 WinCC 项目中创建用于数据交换的连接；将用来处理对 OPC DA 服务器的 WinCC 变量的访问。

为了简化过程，系统使用 OPC 条目管理器。一个 WinCC OPC DA 客户端可以访问多个 OPC DA 服务器。这需要为每个 OPC 服务器创建一个连接。通过这种方式，WinCC OPC DA 客户端可用作中央操作和监控站。



说明

WinCC OPC 通道只能与状态为“OPC_STATUS_RUNNING”的 OPC 服务器建立连接。

说明**“OPC”通道**

连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

6.4.2 OPC 条目管理器

简介

在 WinCC OPC 客户端的 WinCC 项目中组态连接和 WinCC 变量以访问 OPC 服务器的变量。OPC 条目管理器可简化此过程。OPC 条目管理器随 WinCC 自动安装。

说明**“OPC”通道**

连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

要求

为使用 OPC 条目管理器实现组态，必须满足下列要求：

- OPC 服务器是 OPC DA 服务器或 OPC XML 服务器。
- 已在 OPC 服务器上组态了一个变量。
- 如果要将 WinCC 用作 OPC 服务器，必须启用 WinCC OPC 服务器的 WinCC 项目。如果不是这样，OPC 条目管理器不能访问 WinCC OPC 服务器。

6.4 OPC 通道

- 必须可以通过 IP 地址或 HTTP 访问 OPC 服务器的计算机。
- OPC 服务器必须支持浏览器功能。如果不是这样，必须手动组态对 OPC 服务器变量的访问。

说明

如果在 OPC 条目管理器打开时更改 WinCC 项目管理器中的语言，则在单击“浏览服务器”(Browse Server) 按钮时，不会显示任何变量。请在更改语言之前退出 OPC 条目管理器。

OPC 条目管理器的任务

OPC 条目管理器完成下列任务：

- 选择 OPC 服务器
- 创建连接
- 变量选择
- 添加变量

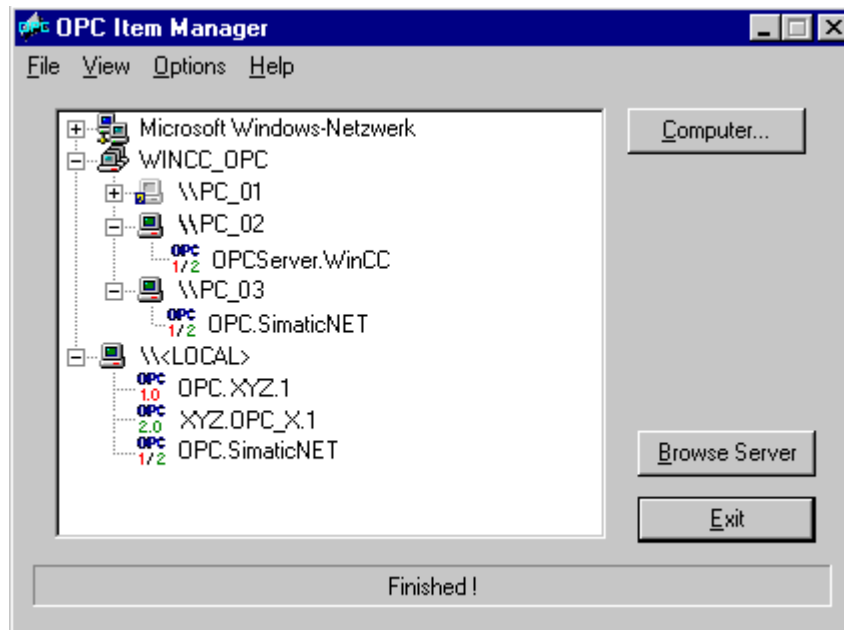
选择 OPC 服务器

OPC DA 服务器

OPC 条目管理器可用于确定网络中 OPC DA 服务器的名称。这些 OPC DA 服务器能够在同一计算机上运行或在已联网的不同计算机上运行。有关详细信息，请参见“WinCC OPC DA 客户端”。

OPC XML 服务器

OPC 条目管理器不能用来查询 OPC XML 服务器的名称。可以使用“OPC Web 服务器”按钮添加 OPC XML 服务器。有关详细信息，请参见“WinCC OPC XML 客户端”。



OPC 条目管理器的图标	描述
	联网计算机尚未检查其是否装有 OPC DA 服务器。
	未在网络中找到计算机或不能访问计算机。
	联网计算机经检查装有 OPC DA 服务器。
	联网计算机包含以 OPC 符号标识的 OPC DA 服务器。数字指示使用了 WinCC OPC DA 客户端的哪种 OPC DA 规范。
\\<LOCAL>	指运行 OPC 条目管理器的计算机。
	OPC XML 服务器的名称。
http://	可以使用“OPC Web 服务器”按钮添加 OPC XML 服务器。

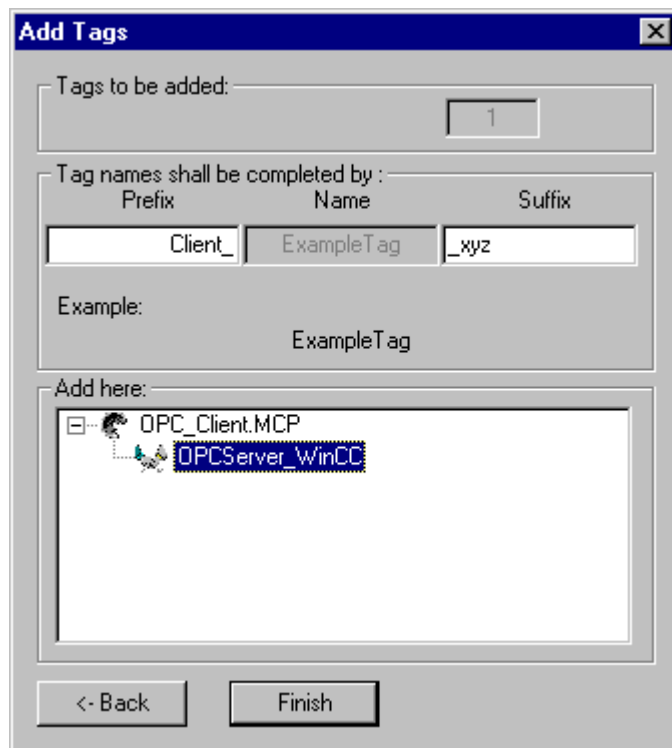
创建连接

创建连接时，OPC 条目管理器会组态所有需要的设置。如果到 OPC 服务器的连接已经创建，则该功能不可用。

变量选择

在 WinCC OPC 客户端要访问的 OPC 服务器上，可以使用变量选择对话框选择一个或多个变量。可以使用过滤标准限制变量选择对话框中的选择。

添加变量



可以在“添加变量”(Add Tags)对话框中设置访问 OPC 服务器变量的 WinCC 变量的名称。

WinCC 变量名由“前缀”、“名称”和“后缀”组成。“名称”域用“ExampleTag”文本预组态。“ExampleTag”代表 WinCC OPC 服务器的 WinCC 变量名称。

为了区分 WinCC OPC 客户端上的 WinCC 变量名和 WinCC OPC 服务器上的 WinCC 变量名，可以分配一个前缀或一个后缀。当组态项目监控时，必须指定一个前缀或一个后缀。

变量名在给定的 WinCC 项目中只能指定一次。

实例

WinCC OPC DA 服务器上的 WinCC 变量名称为“OPC_Server_Tag”。在前缀域中输入值“Client_”，在后缀域中输入“_xyz”。在 WinCC OPC DA 客户端的 WinCC 项目中，将创建 WinCC 变量“Client_OPC_Server_Tag_xyz”。

如果 OPC 服务器上的变量名中含有特殊字符，它们将用下划线“_”代替，因为 OPC 条目管理器并不支持在变量名中出现的所有特殊字符。

单击“完成”(Finish)将 WinCC 变量添加至 WinCC OPC DA 客户端的 WinCC 项目中。OPC 条目管理器自动设置 WinCC 变量的数据类型、名称和地址参数。

参见

如何通过 OPC 条目管理器访问 WinCC 变量 (页 254)

如何通过 OPC 条目管理器访问 WinCC 变量 (页 237)

6.4.3 所支持的 WinCC 数据类型概述

下面的列表显示了 WinCC OPC DA 客户端和 WinCC OPC DA 服务器所支持的数据类型:

- 二进制变量
- 有符号 8 位数
- 无符号 8 位数
- 有符号 16 位数
- 无符号 16 位数
- 有符号 32 位数
- 无符号 32 位数
- 浮点数 32 位 IEEE 754
- 64 位 IEEE 754 浮点数
- 文本变量, 8 位字符集
- 文本变量, 16 位字符集
- 原始数据类型
- 结构类型
- 文本参考
- 日期/时间

说明

结构类型

对于结构类型, 只支持结构元素, 而不支持结构本身。但可在以后对结构进行组态。有关更多的信息, 请参阅主题“在 WinCC OPC DA 客户端上使用结构”。

文本参考

如果使用 OPC 条目管理器创建一个文本变量, 它会被分配 160 个字符的长度。可以任意更改该长度。

6.4 OPC 通道

参见

如何在 WinCC OPC DA 客户机上使用结构 (页 245)

6.4.4 WinCC OPC DA 客户机

6.4.4.1 WinCC OPC DA 客户机的功能

简介

OPC 通道不需要独立的通讯模块。OPC 通道是使用 OPC 软件接口通过 OPC DA 服务器访问过程数据的应用程序。

当使用 WinCC 作为 OPC DA 客户端时，OPC 通道必须添加到 WinCC 项目上。

如果建立了与 WinCC OPC DA 服务器的通讯，将交换 WinCC 变量的值。为此，在 WinCC OPC DA 客户端的 WinCC 项目中建立一个连接，用于处理对 WinCC OPC DA 服务器的访问。

为使 WinCC OPC DA 客户端能访问多个 OPC DA 服务器，必须在 WinCC 项目中为每台 OPC DA 服务器创建连接。有关检测通道和变量故障的更多信息，请参阅“故障检测”。

说明

WinCC OPC 通道只能与状态为“OPC_STATUS_RUNNING”的 OPC 服务器建立连接。

说明

“OPC”通道

连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

连接监控

WinCC OPC-DA 客户端中集成了三种用于连接监视的机制。因而在网络出错或 OPC DA 服务器出现故障的情况下，可以采取最佳的可行性措施。

1. 如果对 DCOM 激活的处理时间超出 5 秒的警告值，变量将被分配值“寻址错误”。如果处理时间段超出 10 秒的取消值，到 OPC DA 服务器的连接将被中断。这显示在 WinCC 项目管理器的“连接状态”对话框中。

OPC DA 规范 3.00 附带“保持有效”特性。如果 OPC DA 服务器支持 OPC DA 规范 3.00，将使用该特性。该特性促使 OPC DA 服务器自动触发周期更新（调用 OnDataChange），即使变量值没有改变。如果此定期更新被禁止，WinCC OPC DA 客户端将终止连接。

这一行为对于支持 OPC DA 规范 2.05a 的 OPC DA 服务器也同样适用。为了检查与 OPC DA 服务器的连接，WinCC OPC DA 客户端每 10 秒周期性地定期进行状态请求。如果此定期更新被禁止，WinCC OPC DA 客户端将终止连接。

通常，当连接失效时，WinCC OPC DA 客户端将终止与 OPC DA 服务器的连接。WinCC OPC DA 客户端每 10 秒自动试图再次重新建立连接。

参见

如何在 WinCC OPC DA 客户机上使用结构 (页 245)

在不使用 OPC 条目管理器的情况下访问 WinCC 变量 (页 243)

通过 OPC 条目管理器组态访问 (页 239)

所支持的 WinCC 数据类型概述 (页 235)

OPC 条目管理器 (页 231)

通道和变量的诊断 (页 563)

OPC 规范及兼容性 (页 637)

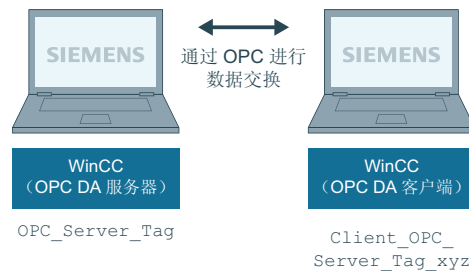
WinCC OPC DA 服务端的功能 (页 646)

6.4.4.2 如何通过 OPC 条目管理器访问 WinCC 变量

如何通过 OPC 条目管理器访问 WinCC 变量

简介

WinCC 之间建立 OPC 连接后，就会通过 WinCC 变量进行数据交换。WinCC OPC DA 客户机会通过 OPC 连接来读取 WinCC OPC DA 服务器上的 WinCC 变量“OPC_Server_Tag”。为了简化过程，系统使用 OPC 条目管理器。



6.4 OPC 通道

要求

- 两台带 WinCC 项目的计算机。
- 这两台计算机必须能够通过其 IP 地址被访问。

组态步骤

在 WinCC OPC DA 客户机的 WinCC 项目中，需要下列组态：

- 创建一个连接。
- 在 WinCC OPCXML 客户端上组态“XMLClient OPC_Var1_xyz”WinCC 变量，此变量用于访问 WinCC OPC DA 服务器的 WinCC 变量。

参见

通过 OPC 条目管理器组态访问 (页 239)

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)

通过 OPC 条目管理器组态访问 (页 255)

WinCC 至 WinCC 的连接实例 (页 649)

在 WinCC OPC DA 客户机上组态 OPC 通道

引言

为使用 OPC 进行数据交换，在 WinCC 项目中必须建立 OPC 通道。

步骤

1. 在 WinCC OPC DA 客户机上的 WinCC 项目浏览器的浏览窗口中，单击“变量管理器”图标。
2. 从“变量管理器”的快捷菜单中选择“添加新的驱动程序”。将打开“添加新的驱动程序”对话框。
3. 选择“OPC.chn”驱动程序并单击“打开”按钮。将创建通道并在变量管理器中显示通讯驱动程序。

参见

通过 OPC 条目管理器组态访问 (页 239)

通过 OPC 条目管理器组态访问

简介

本节说明如何用 OPC 条目管理器对 WinCC OPC DA 服务器的 WinCC 变量“OPC_Server_Tag”的访问进行组态。

要求

- 在 WinCC OPC DA 服务器的 WinCC 项目中，组态一个数据类型为“有符号 16 位数”，名为“OPC_Server_Tag”的内部变量。
- 启用 WinCC OPC DA 服务器的 WinCC 项目。
- 在 WinCC OPC DA 客户端的 WinCC 项目中添加“OPC”通道。

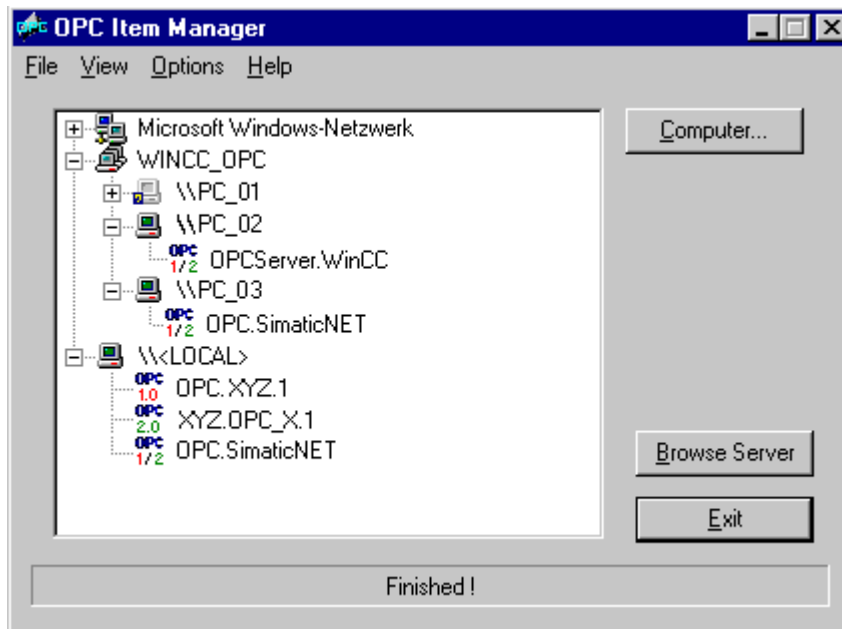
说明

“OPC”通道

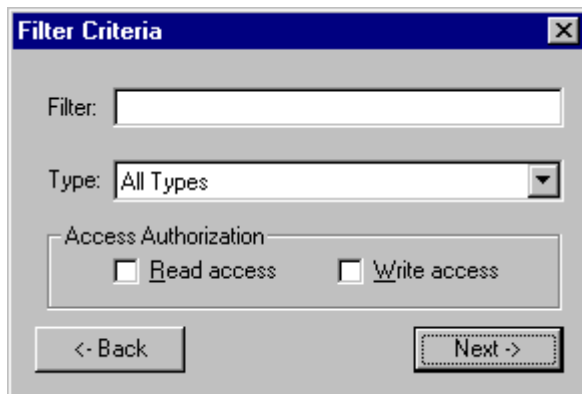
连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

步骤

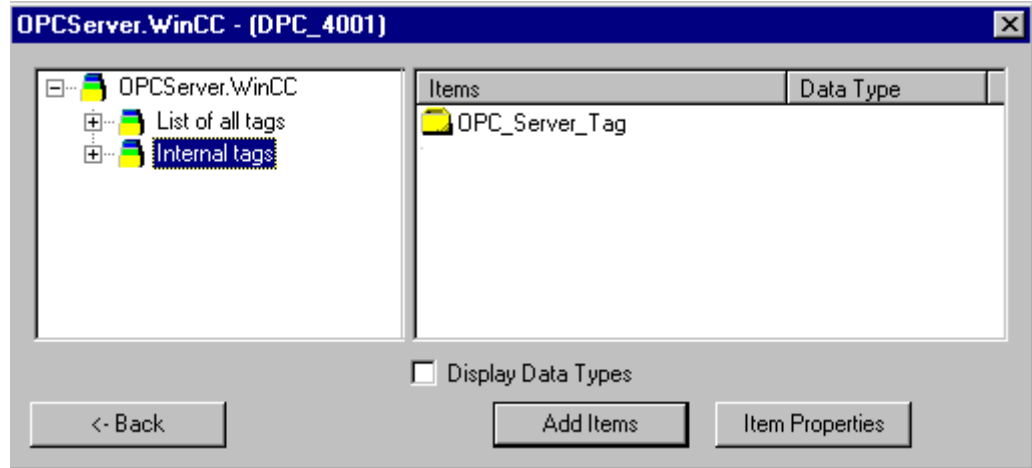
1. 在 WinCC OPC DA 客户端上，从“OPC 组 (OPCHN Unit#1)”通道单元的快捷菜单中选择“系统参数”。将打开“OPC 条目管理器”(OPC Item Manager)。



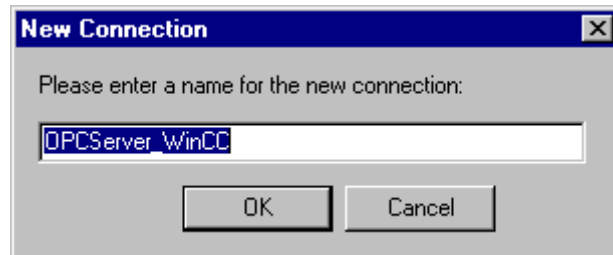
2. 从选择对话框中选择用作 WinCC OPC DA 服务器的计算机的名称。
从显示的列表中选择“OPCServer.WinCC”。
3. 单击“浏览服务器”按钮。
将打开“过滤标准”(Filter criteria) 对话框。



- 单击“下一步”(Next)。
将打开“OPCServer.WinCC ...”对话框。

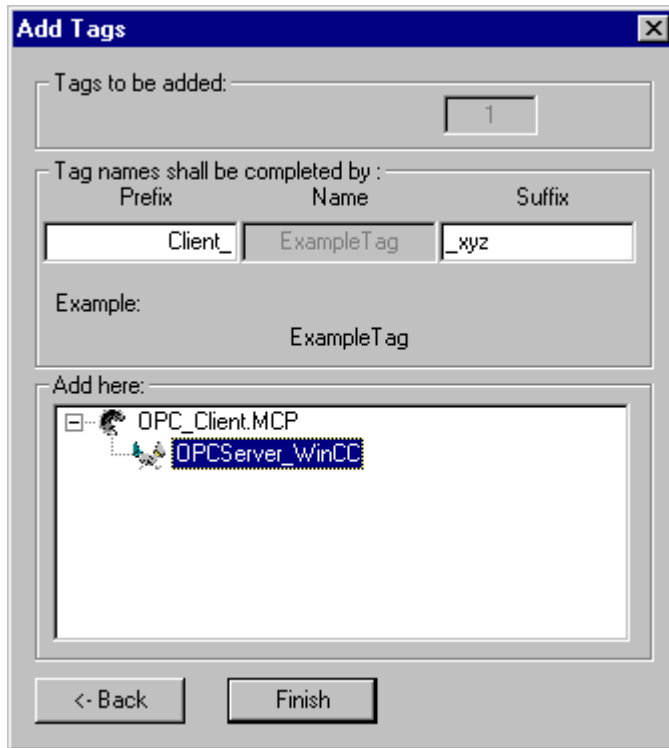


- 选择 WinCC 变量“OPC_Server_Tag”。
单击“添加条目”按钮。
- 如果已创建到 WinCC OPC DA 服务器的连接，继续步骤 6。
如果没有创建连接，将显示一条消息。
单击“是”(Yes)按钮。将打开“新建连接”(New Connection)对话框。



6.4 OPC 通道

7. 输入“OPCServer_WinCC”作为连接的名称。单击“确定”(OK)。将打开“添加变量”(Add Tags)对话框。



8. 在前缀字段中输入“Client_”，在后缀字段中输入“_xyz”。
9. 选择连接“OPCServer_WinCC”。单击“Finish”（完成）。
10. 在“OPCServer.WinCC ...”对话框中，单击“返回”(Back)按钮。单击“退出”关闭 OPC 条目管理器。

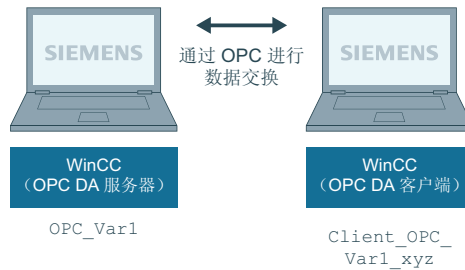
参见

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)

6.4.4.3 在不使用 OPC 条目管理器的情况下访问 WinCC 变量

简介

不支持浏览器功能的 OPC 服务器需要手动组态访问。下面通过一个 WinCC-WinCC OPC 连接的实例来说明 WinCC OPC DA 客户机上 WinCC 变量的组态。



说明

要在不使用 OPC 条目管理器的情况下访问 WinCC 变量，必须手动设置 ItemID。对 WinCC 变量进行寻址时，也可以指定符号计算机名称（服务器前缀）。条目标识号具有下列语法：服务器前缀::WinCC 变量。对本地 WinCC 项目的 WinCC 变量编址时，将忽略服务器前缀。

在 WinCC OPC DA 客户机的 WinCC 项目中，需要下列组态：

1. 选择要访问的“OPC_Var1”WinCC 变量。
2. 创建一个连接。
3. 对访问 WinCC OPC DA 服务器 WinCC 变量的“Client_OPC_Var1_xyz”WinCC 变量进行组态。

要求

- 两台带 WinCC 项目的计算机。
- 这两台计算机必须能够通过其 IP 地址被访问。
- 在 WinCC OPC DA 服务器的 WinCC 项目中，组态名称为“OPC_Var1”且数据类型为“有符号 16 位数”的内部变量。
- 启用 WinCC OPC DA 服务器的 WinCC 项目。
- 将 OPC 通道添加到 WinCC OPC DA 客户机的 WinCC 项目。

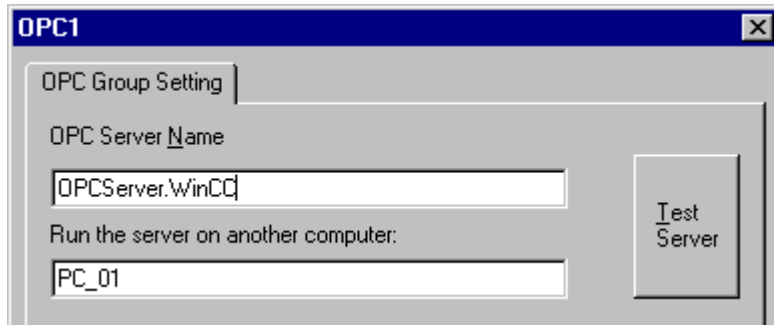
说明

“OPC”通道

连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

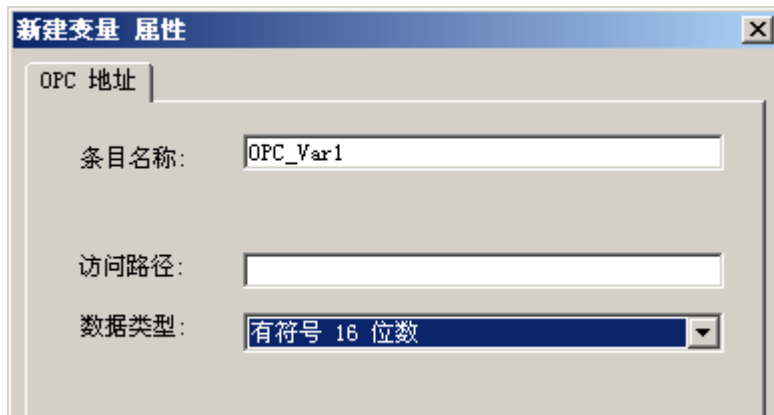
步骤

1. 在 WinCC OPC DA 客户机上，从“OPC 组 (OPCHN Unit#1)”通道单元的快捷菜单中选择“新建连接”。将打开“连接属性”(Connection Properties) 对话框。在相应域中输入连接的名称。
2. 单击“属性”按钮。将显示以连接名称为标题的对话框。



对于与 WinCC V6 的连接，“OPC 服务器名称”域中的条目必须为“OPCServer.WinCC”。

3. 在“在本计算机上启动服务器”域中，输入要用作 OPC DA 服务器的计算机名称。单击“测试服务器”，检查与 WinCC OPC DA 服务器的连接。
4. 从连接的快捷菜单中选择“新建变量”。将打开“变量属性”对话框。
5. 在“变量”域中输入名称“Client_OPC_Var1_xyz”。将数据类型设置为“有符号 16 位数”。
6. 在“变量属性”对话框中，单击“选择”按钮。将打开“地址属性”对话框。



在“条目名称”域中输入 WinCC OPC DA 服务器的 WinCC 变量的名称。保持“访问路径”域中的条目不变。将数据类型设置为“有符号 16 位数”。

7. 单击“确定”关闭所有打开的对话框。

6.4.4.4 在 WinCC OPC DA 客户机上使用结构

如何在 WinCC OPC DA 客户机上使用结构

简介

结构用于组织构成逻辑单元的变量和变量类型。这允许使用单个的逻辑名称对其进行引用。

OPC DA 规范不支持结构。这样就不能使用 OPC 条目管理器建立结构，只能建立结构中的单个变量。如果仍然想要在 WinCC OPC DA 客户端上使用结构，必须随后在 WinCC OPC DA 客户端的 WinCC 项目中组态数据结构，以便为结构提供服务器变量的相关条目名称。

要求

- 两台带 WinCC 项目的计算机。
- 这两台计算机必须能够通过其 IP 地址被访问。

组态步骤

要在 WinCC OPC DA 客户端上使用结构，下列组态步骤是必需的：

- 在 WinCC OPC DA 服务器上组态结构和结构变量
- 使用 WinCC OPC DA 客户端上 WinCC 项目中的结构

参见

如何在 WinCC OPC DA 客户机上组态结构 (页 246)

在 WinCC OPC DA 服务器上组态结构和结构变量 (页 245)

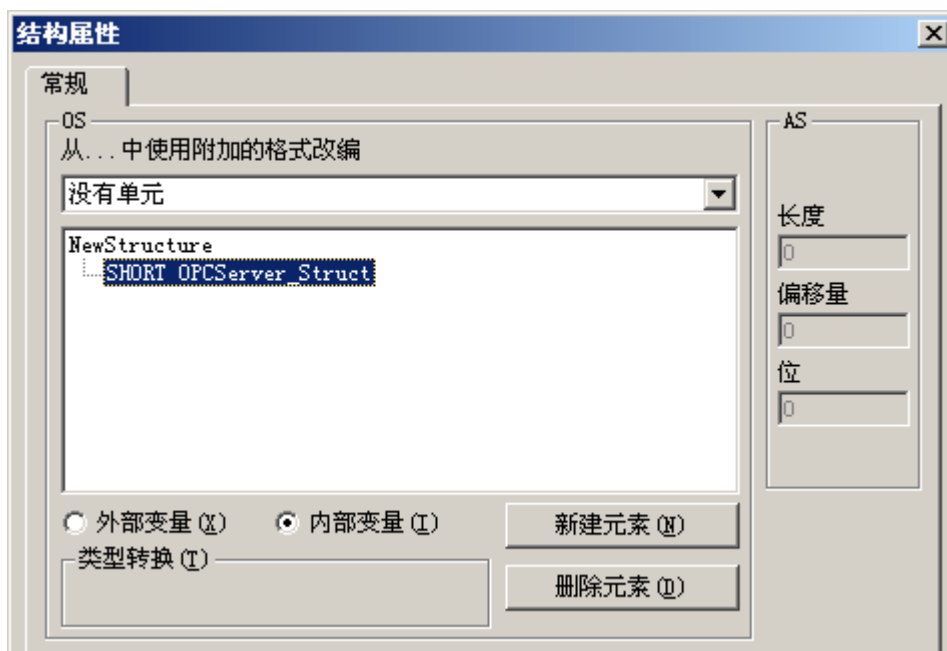
在 WinCC OPC DA 服务器上组态结构和结构变量

引言

在本节中，将在 OPC DA 服务器的 WinCC 项目中创建一个结构和一个结构变量。OPC DA 客户机需要该组态来访问结构变量。

步骤

1. 从 WinCC OPC DA 服务器上的结构类型快捷菜单中选择“新建结构类型”。将显示“结构属性”对话框。
2. 单击“新建元素”，创建数据类型为 SHORT 的内部变量“OPCServer_Struct”。



单击“确定”关闭对话框。

3. 在浏览窗口中，单击变量管理器图标前的加号。从内部变量快捷菜单中选择“新建变量”。创建名称为“Var”具有该结构类型的 WinCC 变量。
4. WinCC 项目管理器的数据框架显示单个变量“Var”和结构变量“Var.OPCServer_Struct”。
5. 激活 WinCC 项目。

参见

如何在 WinCC OPC DA 客户机上组态结构 (页 246)

如何在 WinCC OPC DA 客户机上组态结构

简介

OPC DA 规范不支持结构。这样就不能使用 OPC 条目管理器建立结构。在本节中，我们将 WinCC OPC DA 服务器 WinCC 项目中已有的结构针对 WinCC OPC DA 客户机的 WinCC 项目进行组态。将在 WinCC OPC DA 客户机上组态用于访问 WinCC OPC DA 服务器上现有结构变量的 WinCC 变量。

要求

- 在 WinCC OPC DA 服务器的 WinCC 项目中创建结构和名称为“Var.OPCServer_Struct”的结构变量。
- 启用 WinCC OPC DA 服务器的 WinCC 项目。
- 将 OPC 通道添加到 WinCC OPC DA 客户机的 WinCC 项目。

说明

“OPC”通道

连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

步骤

1. 在 WinCC OPC DA 客户机上，从结构类型快捷菜单中选择“新建结构类型”。将显示“结构属性”对话框。
2. 单击“新建元素”按钮并设置一个外部变量。命名元素，使其名称与在 OPC-DA 服务器的 WinCC 项目中的名称完全相同。单击“确定”关闭“结构属性”对话框。
3. 如果已创建与 OPC DA 服务器的连接，继续步骤 6。
如果没有创建任何连接，从“OPC”通道单元的快捷菜单中选择“新建连接”(New Connection)。将打开“连接属性”(Connection Properties)对话框。在相应域中输入连接的名称。
4. 单击“属性”按钮。将显示以连接名称为标题的对话框。对于与 WinCC V6 的连接，域“OPC 服务器名称”中的条目必须为“OPCServer.WinCC”。
5. 在域“在本计算机上启动服务器”中，输入要用作 WinCC OPC DA 服务器的计算机名称。单击“测试服务器”，检查与 WinCC OPC DA 服务器的连接。单击“确定”关闭对话框。
6. 从连接的快捷菜单中选择“新建变量”。将打开“变量属性”对话框。选择新创建的结构类型作为数据类型。
7. 在“变量属性”对话框中，单击“选择”按钮。将打开“地址属性”(Address properties)对话框。
在“条目名称”(Item Name)字段中，输入“Var.OPCServer_Struct”作为 WinCC OPC DA 服务器的结构变量名称。保持“访问路径”域中的条目不变。
8. 单击“确定”关闭所有打开的对话框。

参见

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)

在 WinCC OPC DA 服务器上组态结构和结构变量 (页 245)

6.4 OPC 通道

6.4.4.5 OPC DA 通讯受到干扰时的错误处理

OPC 通讯受到干扰时的错误处理

引言

通讯测试的过程与 WinCC 的使用方式无关。

WinCC 用作 OPC DA 服务器

在 WinCC OPC DA 客户机上使用通道诊断来确定是否可以建立与 OPC DA 服务器的连接。有关通道故障分析的更多信息，请参阅“故障检测”。

WinCC 用作 OPC DA 客户机

在 WinCC OPC DA 客户机上使用通道诊断来确定是否可以建立与 OPC DA 服务器的连接。有关通道故障分析的更多信息，请参阅“故障检测”。

参见

WinCC 用作 OPC DA 客户机，未建立连接。(页 252)

WinCC 用作 OPC DA 客户机，建立连接。(页 251)

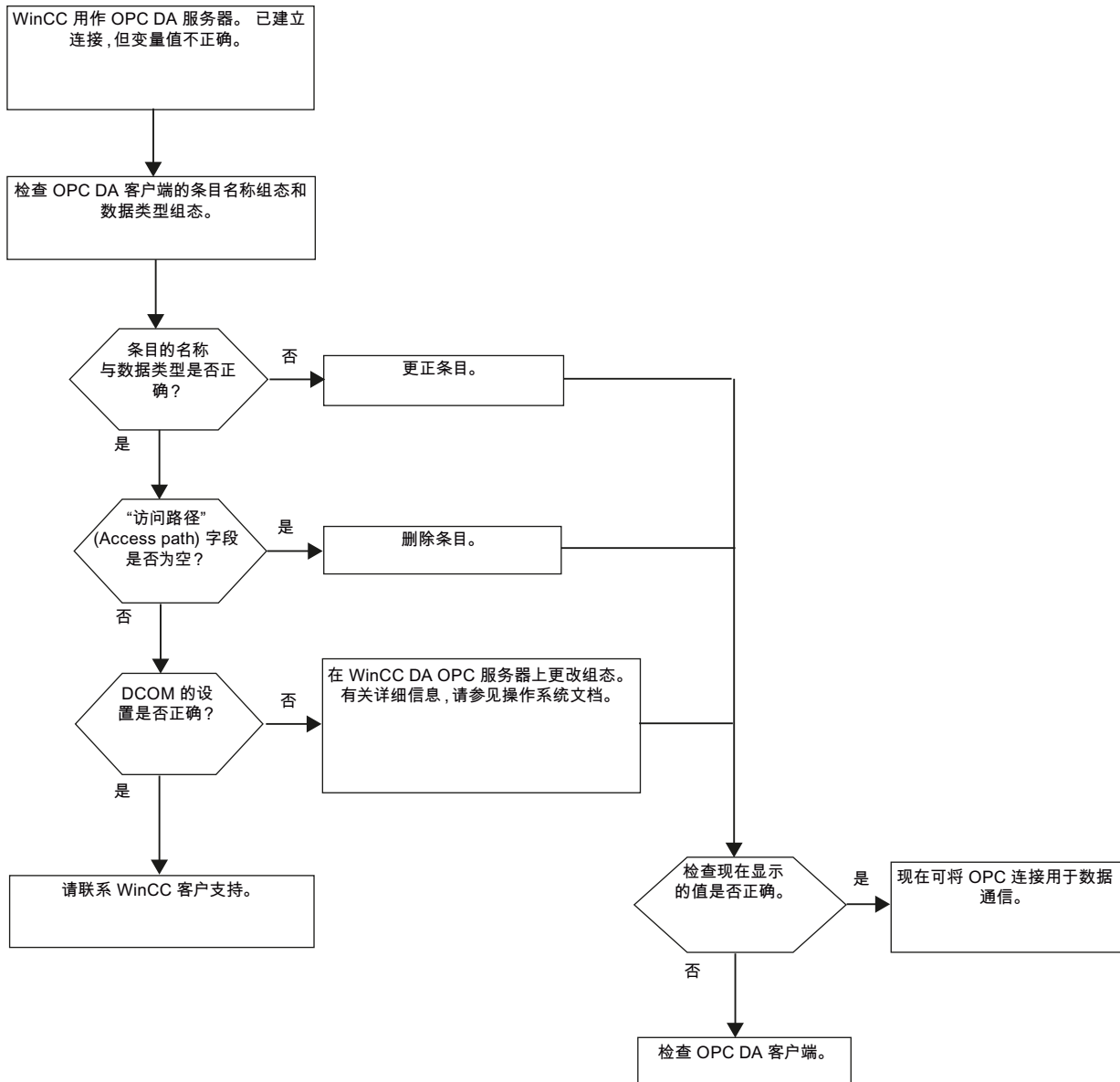
WinCC 用作 OPC DA 服务器，未建立连接。(页 250)

WinCC 用作 OPC DA 服务器，成功建立连接。(页 249)

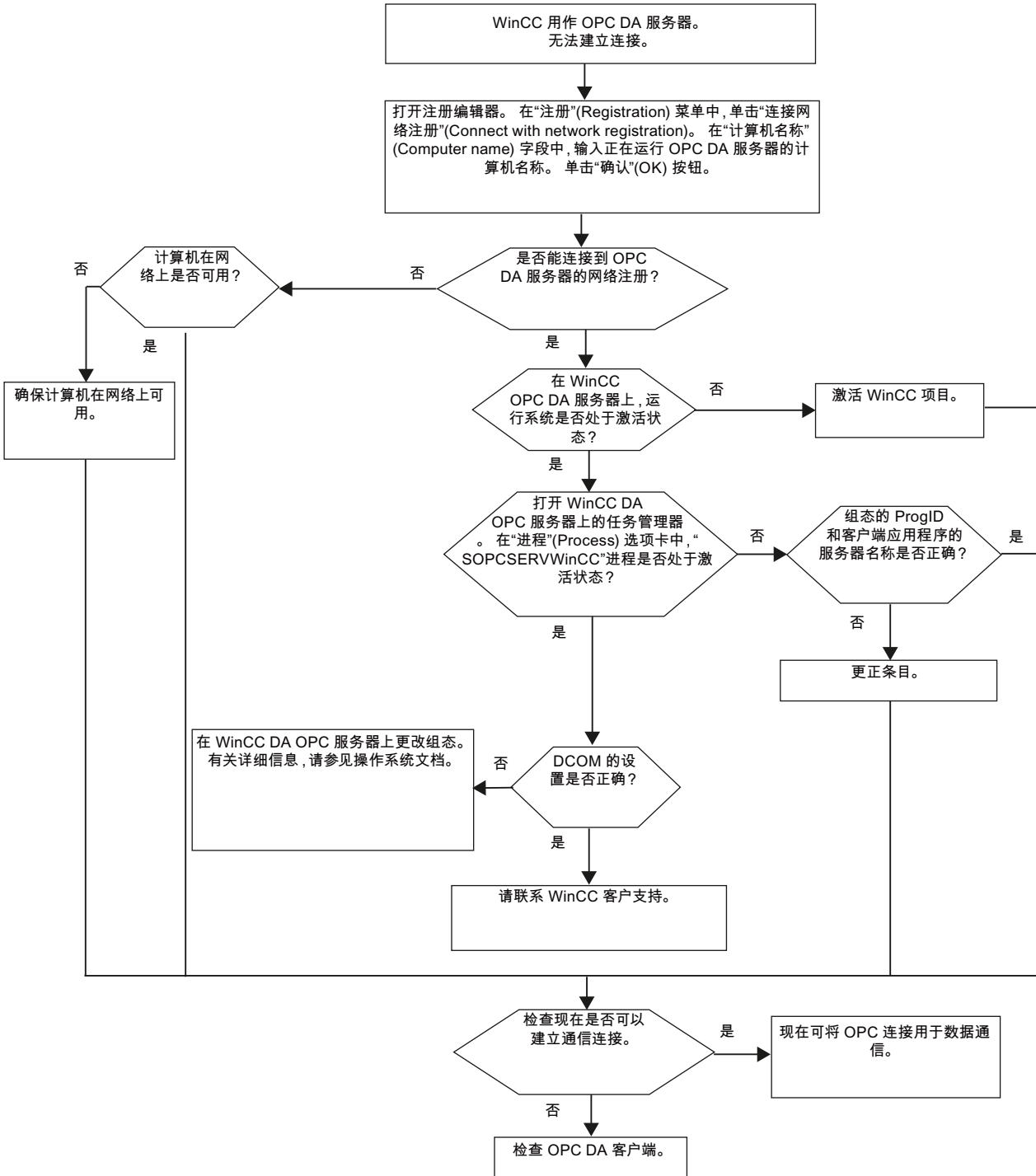
诊断“OPC”通道的可能性(页 610)

WinCC 作为 OPC DA 服务器

WinCC 用作 OPC DA 服务器，成功建立连接。

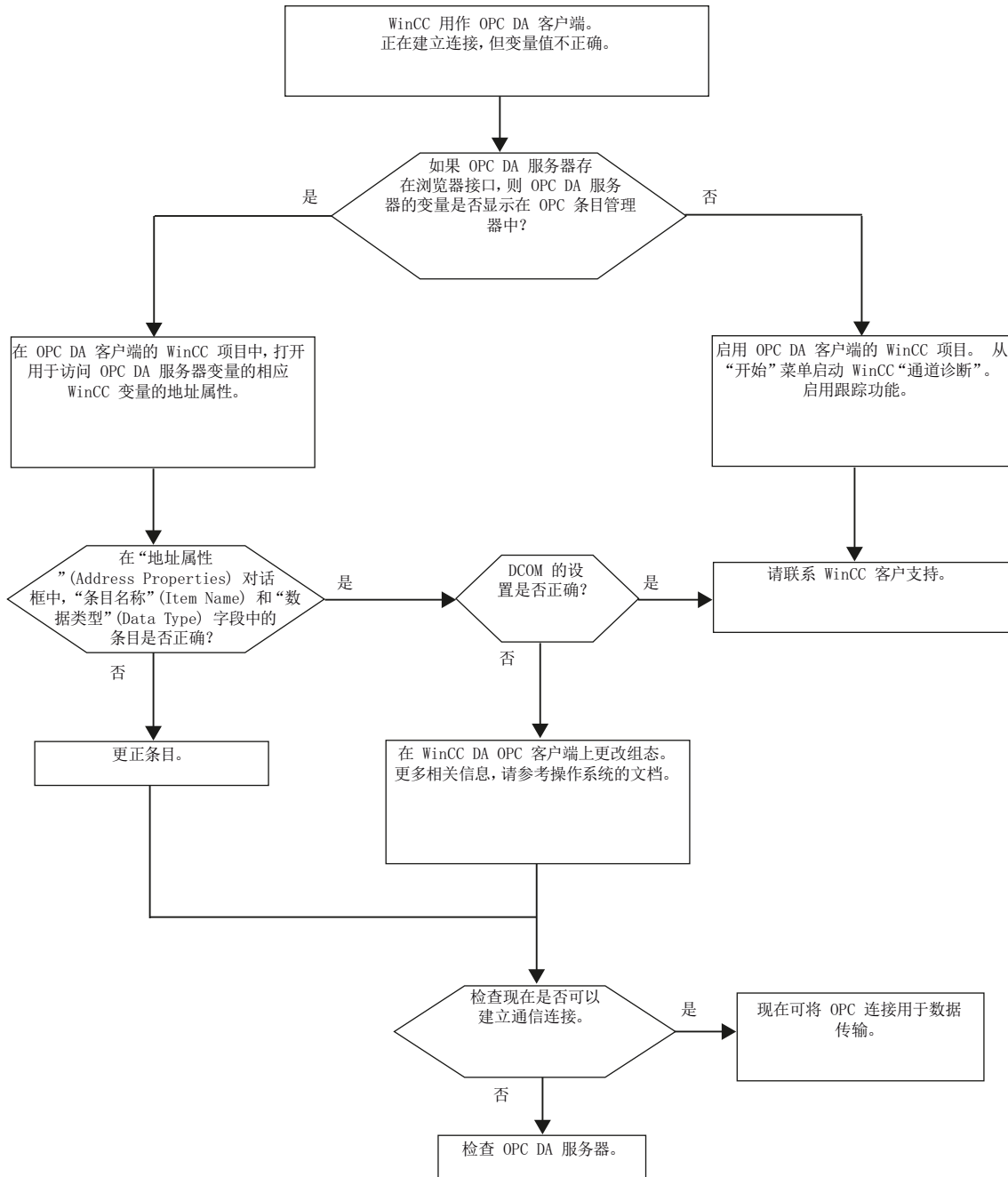


WinCC 用作 OPC DA 服务器，未建立连接。

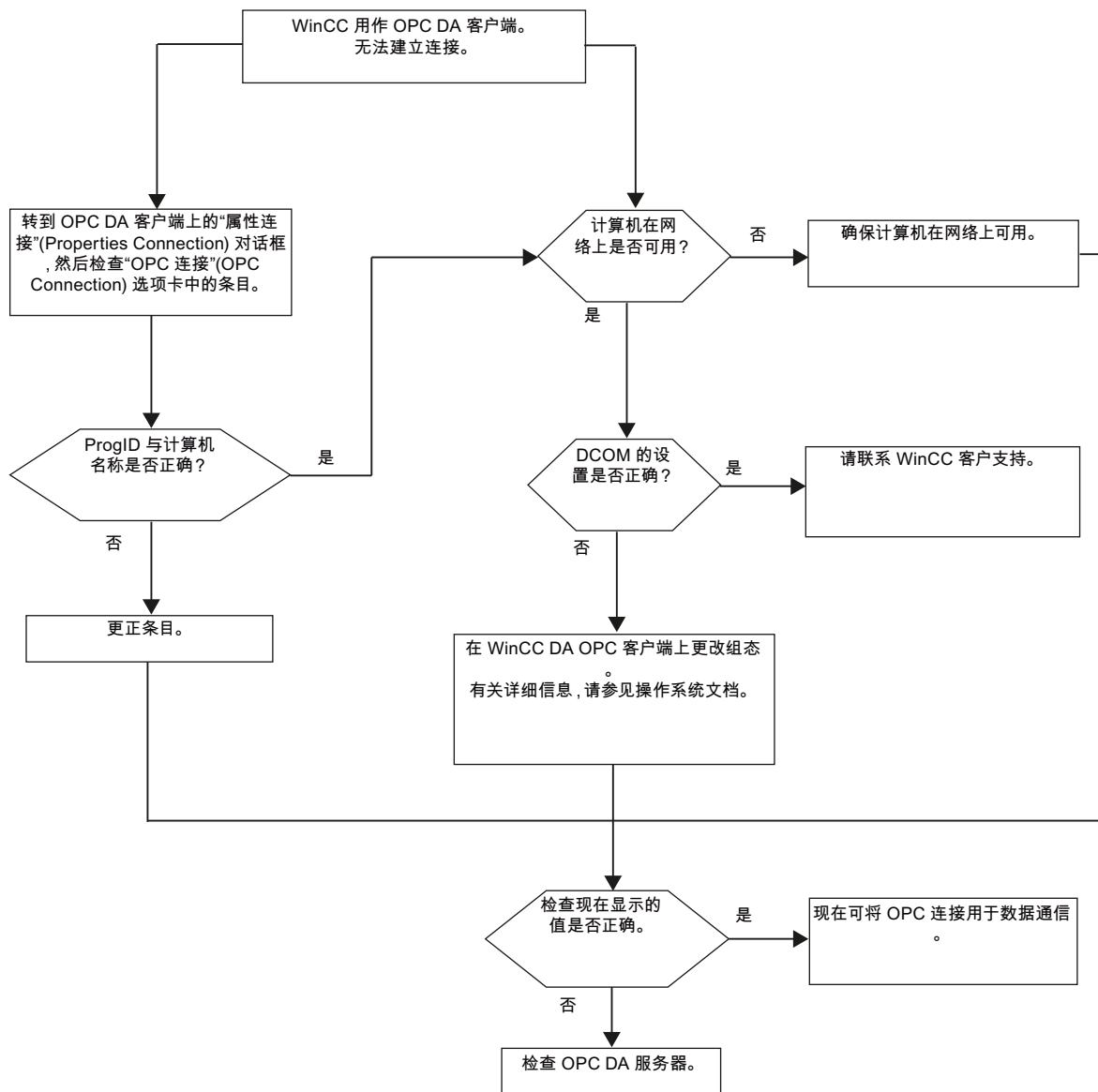


WinCC 作为 OPC DA 客户机

WinCC 用作 OPC DA 客户机，建立连接。



WinCC 用作 OPC DA 客户端，未建立连接。



6.4.5 WinCC OPC XML 客户端

6.4.5.1 WinCC OPC XML 客户端的功能

简介

OPC 通道不需要独立的通讯模块。 OPC 通讯驱动程序可以作为 OPC XML 客户端使用。

为了将 WinCC 用作 WinCC OPC XML 客户端，必须将 OPC 通道添加到 WinCC 项目上。

WinCC OPC XML 客户端以 Web 页面形式为 OPC XML 服务器提供 OPC 过程数据。可以使用 HTTP 通过 Internet/Intranet 对 Web 页面进行访问。WinCC OPC XML 客户端请求数据时，由 Web 服务器自动启动 Web 服务。

为了使 WinCC OPC XML 客户端可以访问多个 OPC XML 服务器，必须与 WinCC 项目中的每个 OPC XML 服务器建立一个连接。

如果建立了与 WinCC OPC XML 服务器的通讯，将交换 WinCC 变量的值。在 WinCC OPC XML 客户端的 WinCC 项目中建立了一个连接，可以通过其处理对 WinCC OPC XML 服务器的访问。如果是 WinCC OPC XML 客户端，连接监视不被激活。

注意**使用多个网络适配器和激活的 TCP/IP 协议进行操作**

要使用多个网络适配器和激活的 TCP/IP 协议进行操作，请阅读“WinCC 发行说明/与运行/网络技术及 UPS 有关的说明”和“组态/分布式系统/运行期间的系统行为/组合使用服务器与若干个网络适配器时的特殊通信功能”这两个部分中的信息。

说明

WinCC OPC 通道只能与状态为“OPC_STATUS_RUNNING”的 OPC 服务器建立连接。

说明**“OPC”通道**

连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

参见

所支持的 WinCC 数据类型概述 (页 235)

OPC 条目管理器 (页 231)

OPC 规范及兼容性 (页 637)

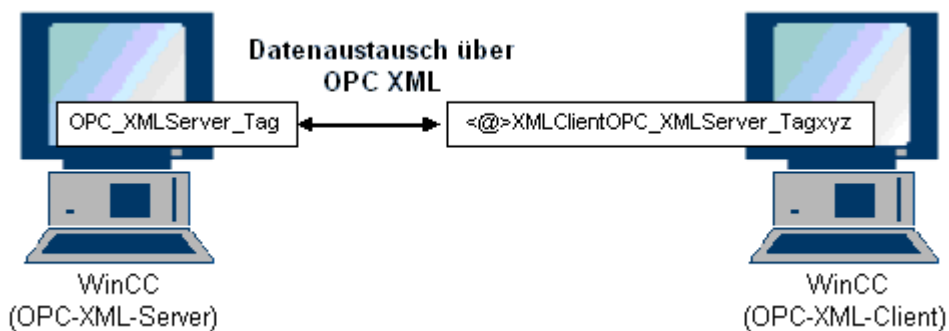
WinCC OPC XML DA 服务器的功能 (页 641)

6.4.5.2 如何通过 OPC 条目管理器访问 WinCC 变量

如何通过 OPC 条目管理器访问 WinCC 变量

引言

WinCC 之间建立 OPC 连接后，数据就会通过 WinCC 变量进行交换。WinCC OPC DA 客户机使用 OPC 连接来读取 WinCC OPC XML 服务器上的 WinCC 变量“OPC_Server_Tag”。为了简化过程，系统使用 OPC 条目管理器。



要求

- 两台带 WinCC 项目的计算机。
- 必须可以使用 HTTP 访问这两台计算机。

组态步骤

在 WinCC OPC XML 客户机的 WinCC 项目中，需要下列组态：

参见

- 在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)
- 通过 OPC 条目管理器组态访问 (页 255)

通过 OPC 条目管理器组态访问

简介

本节说明如何使用 OPC 条目管理器组态对 WinCC OPC XML 服务器上的“OPC_XMLServer_Tag” WinCC 变量的访问。

要求

- 在 WinCC OPC XML 服务器的 WinCC 项目中，组态一个名称为“OPC_XMLServer_Tag”、数据类型为“有符号 16 位数”的内部变量。
- 激活 WinCC OPC XML 服务器的 WinCC 项目。
- 必须将 OPC 通道添加到 WinCC OPC XML 客户机的 WinCC 项目。

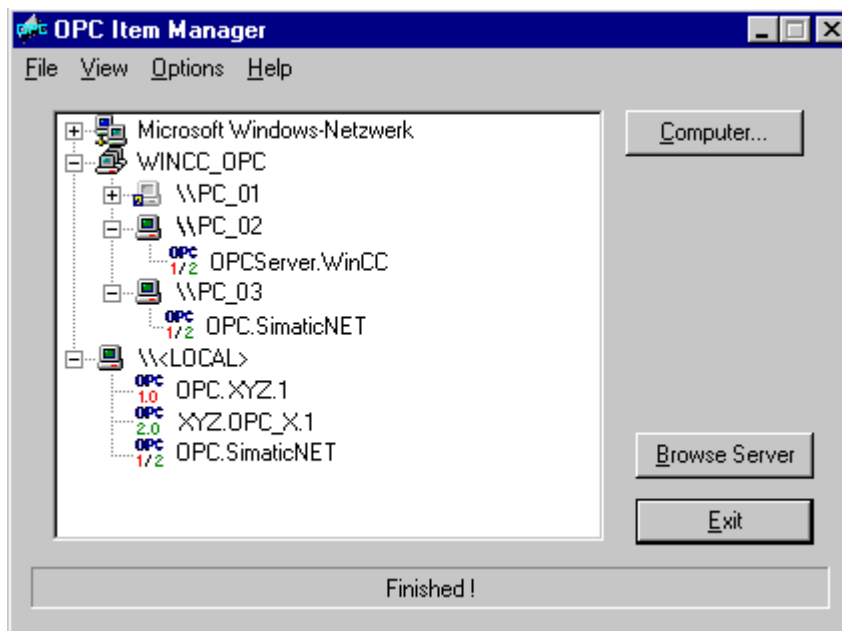
说明

“OPC”通道

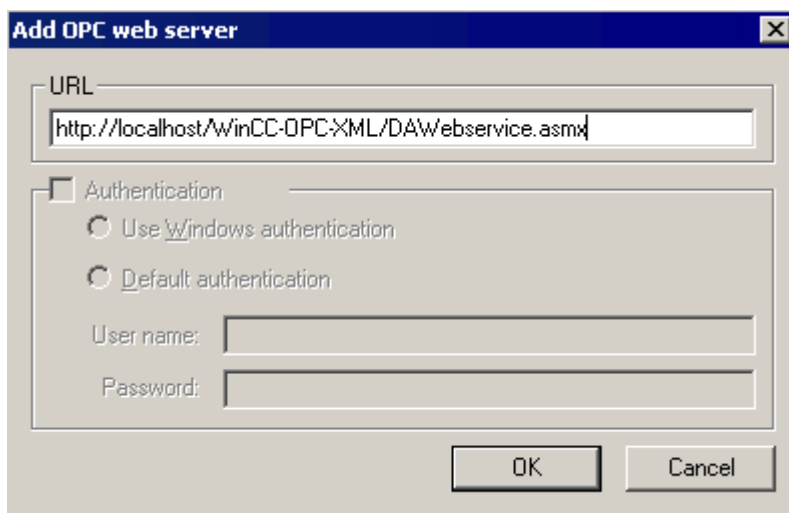
连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

步骤

1. 在 WinCC OPC XML 客户机上，从“OPC 组 (OPCHN Unit#1)”通道单元的快捷菜单中选择“系统参数”。将打开“OPC 条目管理器”(OPC Item Manager)。

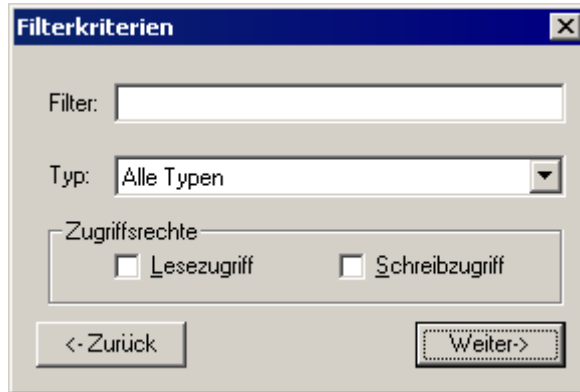


2. 单击“OPC Web 服务器”按钮。将打开“添加 OPC Web 服务器”对话框。以下列形式在“URL”域中输入 WinCC OPC XML 服务器的 URL: `<http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx>`。使用运行 OPC XML Web 服务的计算机的 IP 地址或计算机名称替代 xxx。

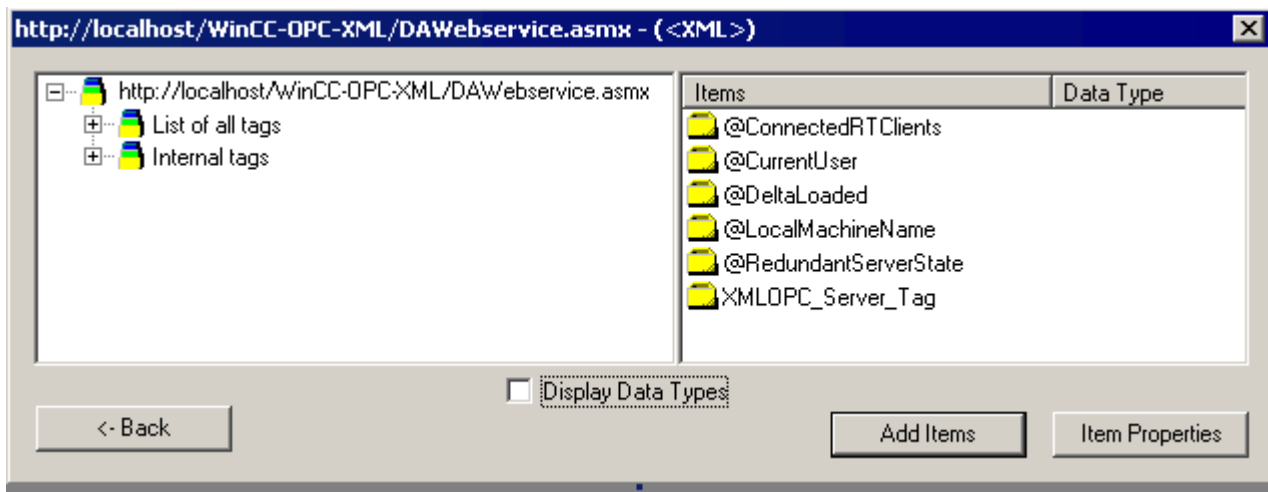


单击“确定”关闭对话框。

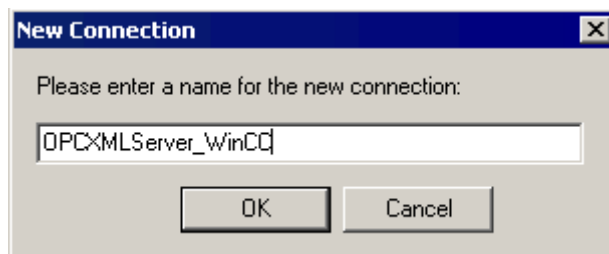
3. 显示一个列表，从其中选择“/WinCC-OPC-XML/DAWebservice.asmx>”。单击“浏览服务器”按钮。将打开“过滤标准”(Filter criteria)对话框。



4. 在“过滤标准”对话框中，单击“下一步->”按钮。将打开“http:// ...”对话框。



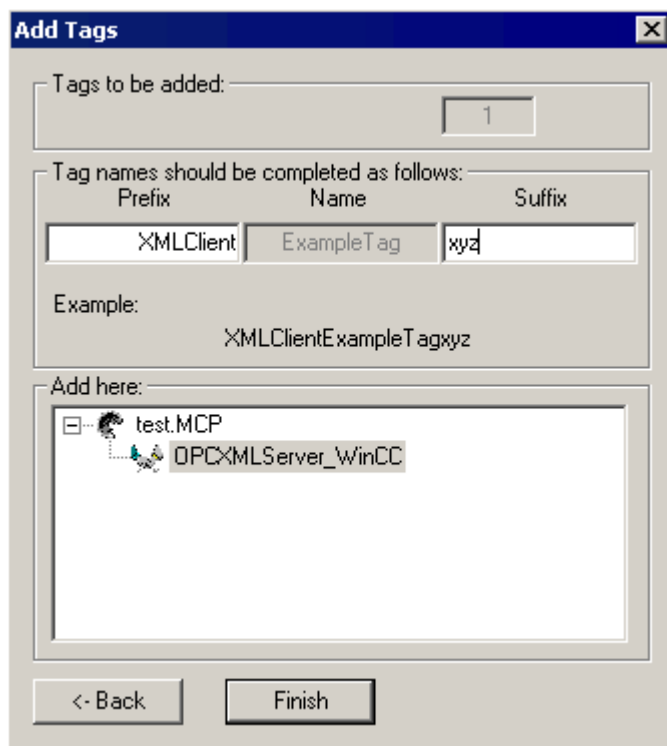
5. 在“http:// ...”对话框中，选择 WinCC 变量“XMLOPC_Server_Tag”。单击“添加条目”按钮。
6. 如果已创建与 WinCC OPC XML 服务器的连接，继续步骤 7。如果尚未组态连接，则会输出相应的消息。
单击“是”(Yes)。将打开“新建连接”(New Connection)对话框。



为连接输入名称“OPCXMLServer_WinCC”。单击“确定”(OK)。

6.4 OPC 通道

7. 将打开“添加变量”(Add Tags)对话框。
在前缀字段中输入“XMLClient_”，在后缀字段中输入“_xyz”。选择“OPCXMLServer_WinCC”连接。单击“Finish”（完成）。



8. 在“http:// ...”对话框中，单击“<-返回”。单击“退出”关闭 OPC 条目管理器。

参见

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)

6.4.5.3 在不使用 OPC 条目管理器的情况下访问 WinCC 变量

引言

不支持浏览器功能的 OPC 服务器需要手动组态访问。下面通过一个 WinCC-WinCC OPC 连接的实例来说明 WinCC OPC XML 客户机上 WinCC 变量的组态。

说明

要在不使用 OPC 条目管理器的情况下访问 WinCC 变量，必须手动设置 ItemID。对 WinCC 变量进行寻址时，也可以指定符号计算机名称（服务器前缀）。条目标识号具有下列语法：
server_prefix::<@>WinCC 变量。如果对本地 WinCC 项目的 WinCC 变量编址，ItemID 具有下列语法：<@>WinCC 变量。

组态步骤

在 WinCC OPC XML 客户机的 WinCC 项目中，需要下列组态：

1. 创建一个连接。
2. 在 WinCC OPC XML 客户机上组态“XMLClient_OPC_Var1_xyz”WinCC 变量，此变量用于访问 WinCC OPC DA 服务器 WinCC 变量。

要求

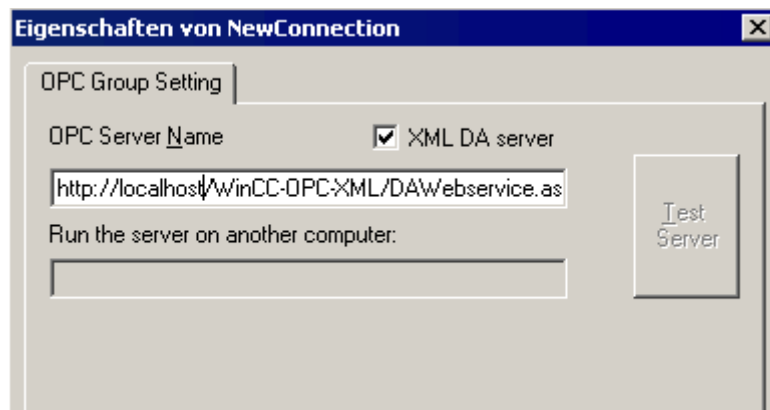
- 两台带 WinCC 项目的计算机。
- 必须可以使用 HTTP 访问这两台计算机。
- 在 WinCC OPC XML 服务器的 WinCC 项目中，组态一个名称为“XMLOPC_Server_Tag”、数据类型为“有符号 16 位数”的内部变量。
- 启用 WinCC OPC XML 服务器的 WinCC 项目。
- 必须将 OPC 通道添加到 WinCC OPC XML 客户机的 WinCC 项目。

说明

当在 OPC 通道中组态外部变量时，“变量属性”对话框的类型转换域中来自 WinCC 的预置值一定不能改变。正在处理的变量的数据类型在“地址属性”对话框的数据类型域中设置。

步骤

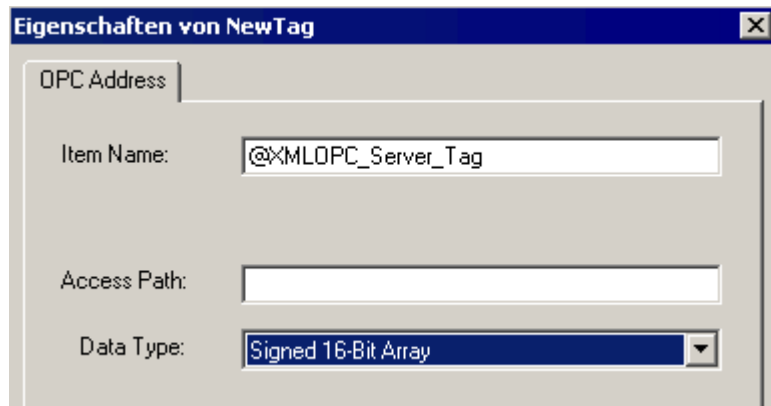
1. 在 WinCC OPC XML 客户机上，从“OPC Groups(OPCHN Unit#1)”通道单元的快捷菜单上选择“新建连接”。将打开“连接属性”对话框。在相应域中输入连接的名称。
2. 单击“属性”按钮。将显示以连接名称为标题的对话框。



选中“XML DA 服务器”复选框。如果是到 WinCC OPC XML 服务器的连接，“OPC 服务器名称”域必须包含 WinCC OPC XML 服务器的 URL。URL 具有下列语法：“http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx”。使用运行 OPC-XML Web 服务的计算机的 IP 地址或计算机名称替代 xxx。

6.4 OPC 通道

3. 从连接的快捷菜单中选择“新建变量”。将打开“变量属性”对话框。
4. 在“变量”域中输入名称“XMLClient_OPC_Var1_xyz”。将数据类型设置为“有符号 16 位数”。
5. 在“变量属性”对话框中，单击“选择”按钮。将显示以变量名为标题的对话框。



在“条目名称”域中输入符号“<@>”和 WinCC OPC XML 服务器的 WinCC 变量的名称。保持“访问路径”域中的条目不变。将数据类型设置为“有符号 16 位数”。

6. 单击“确定”关闭所有打开的对话框。

6.5 OPC UA WinCC 通道

6.5.1 WinCC 通道“OPC UA WinCC 通道”

简介

WinCC 可以用作 OPC UA 服务器和 OPC UA 客户端。通道“OPC UA WinCC Channel”是 WinCC 的 OPC UA 客户端应用程序。

借助 OPC UA（统一架构），WinCC 支持独立于平台的 OPC 跟进技术。有关详细信息，请参见 OPC UA 规范及 WinCC Information System 下的“接口 > OPC - 开放式互连 > WinCC OPC UA 服务器”(Interfaces > OPC - Open Connectivity > WinCC OPC UA Server)。需要了解 OPC UA 的基本知识才能进行组态。

可在 WinCC Information System 下的“通信 > OPC 通道”(Communication > OPC channel) 中找到有关 OPC-DA 客户端及 OPC-XML 客户端的文档。

WinCC OPC UA 客户端的功能

WinCC OPC UA 客户端支持对任何符合“OPC 统一架构”规范的 OPC UA 服务器的数据访问。

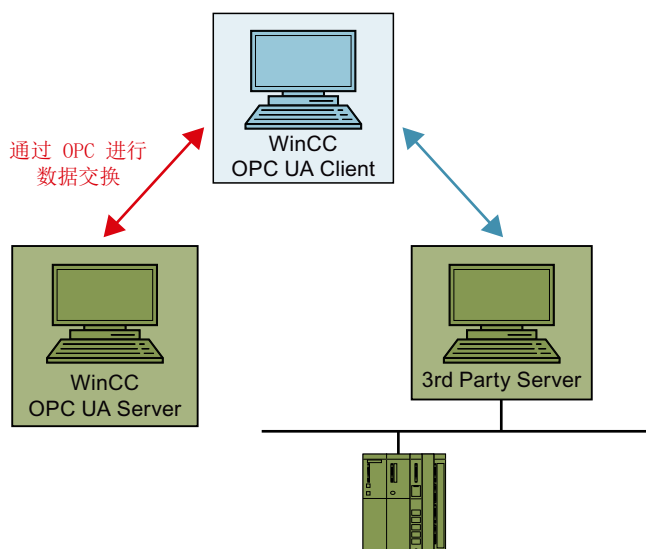
OPC Unified Architecture (OPC UA) 提供了验证和加密等其它机制以保证相关伙伴之间的通信安全。

通过 OPC UA 通信

要将 WinCC 用作 WinCC OPC UA 客户端，请将 OPC UA 通信驱动程序“OPC UA WinCC Channel”插入到 WinCC 项目中。无需单独的通信模块。

一个 WinCC OPC UA 客户端可以访问多个 OPC UA 服务器。为此，可以组态与每个 OPC UA 服务器的连接。可将 WinCC OPC UA 客户端用作中央操作员监控站。

在 WinCC Configuration Studio 的变量管理中执行组态。



6.5.2 所支持的数据类型概述

简介

为了进行数据通信，在 WinCC OPC UA 客户端的 WinCC 项目中组态访问 OPC UA 服务器的变量。

要实现这一目的，在变量管理中导入 OPC UA 节点作为 WinCC 变量。

所支持的数据类型

WinCC OPC UA 客户端和 WinCC OPC UA 服务器支持以下 WinCC 数据类型。

- 二进制变量
- 有符号 8 位数
- 无符号 8 位数
- 有符号 16 位数
- 无符号 16 位数
- 有符号 32 位数
- 无符号 32 位数
- 浮点数 32 位 IEEE 754
- 64 位 IEEE 754 浮点数

- 文本变量，8 位字符集
- 文本变量，16 位字符集
- 原始数据类型
- 结构类型
- 日期/时间

6.5.3 组态 OPC UA 通道

6.5.3.1 常规序列

简介

借助 WinCC 与 OPC UA 服务器间的 OPC UA 链接，WinCC OPC UA 客户端可以通过安全连接访问 OPC UA 服务器的变量。为此将 OPC UA 服务器的数据映射为 WinCC 变量。

在变量管理中执行组态。

要求

- OPC UA 服务器已激活。
- 计算机必须通过 TCP 连接。
- 通信不得被防火墙拦截。
- OPC UA 服务器的端口号必须激活。

组态步骤

- 在 WinCC 项目中插入 OPC UA WinCC 通道 (页 264)
- 创建与 OPC UA 服务器的连接 (页 264)
- 设置通过证书进行认证。(页 272)
- 组态 OPC UA 变量 (页 282)
- 在 WinCC 中使用 OPC UA 报警 (页 294)

6.5 OPC UA WinCC 通道

6.5.3.2 在 WinCC 项目中插入 OPC UA WinCC 通道

要求

- 已创建 WinCC 项目。
- 已打开变量管理。

操作步骤

1. 在导航区域中打开变量管理的快捷菜单。
2. 选择“添加新的驱动程序 > OPC UA WinCC Channel”。

结果

通道 OPC UA WinCC Channel 添加到变量管理中。

在通道下创建了通道单元“OPC UA Connections”。在此通道单元下，可组态一个或多个 OPC UA 服务器连接。

6.5.3.3 创建与 OPC UA 服务器的连接

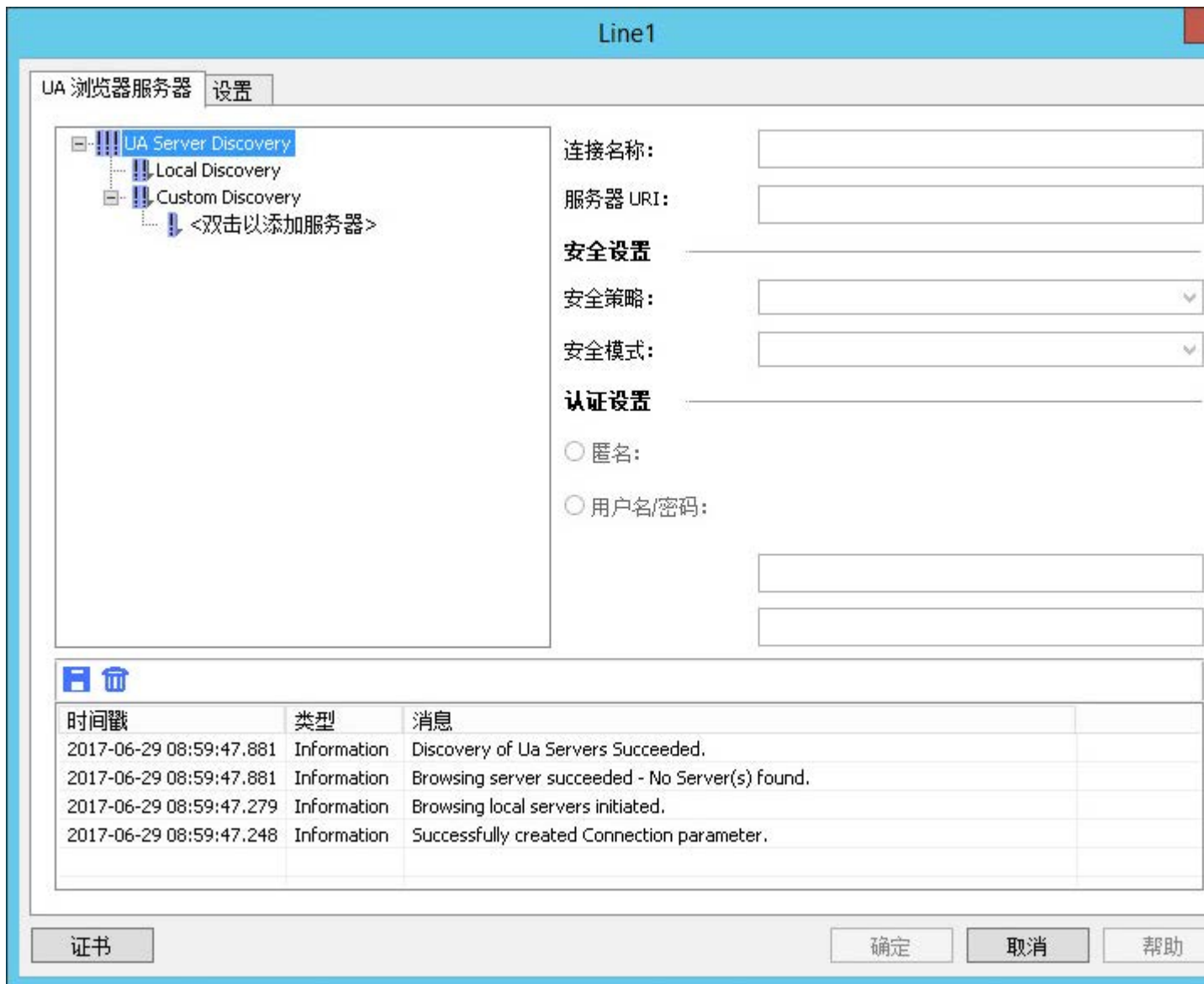
要连接 OPC UA 服务器，需要服务器和安全设置的相关信息。

可以在 WinCC 项目中为每个 OPC UA 服务器创建一个连接。

在变量管理中，使用连接的快捷菜单中的条目打开“连接参数”(Connection Parameters) 对话框。

连接参数

在“UA 服务器浏览器”(UA Server Browser) 选项卡中，为与 OPC UA 服务器的通信组态所需设置。



6.5 OPC UA WinCC 通道

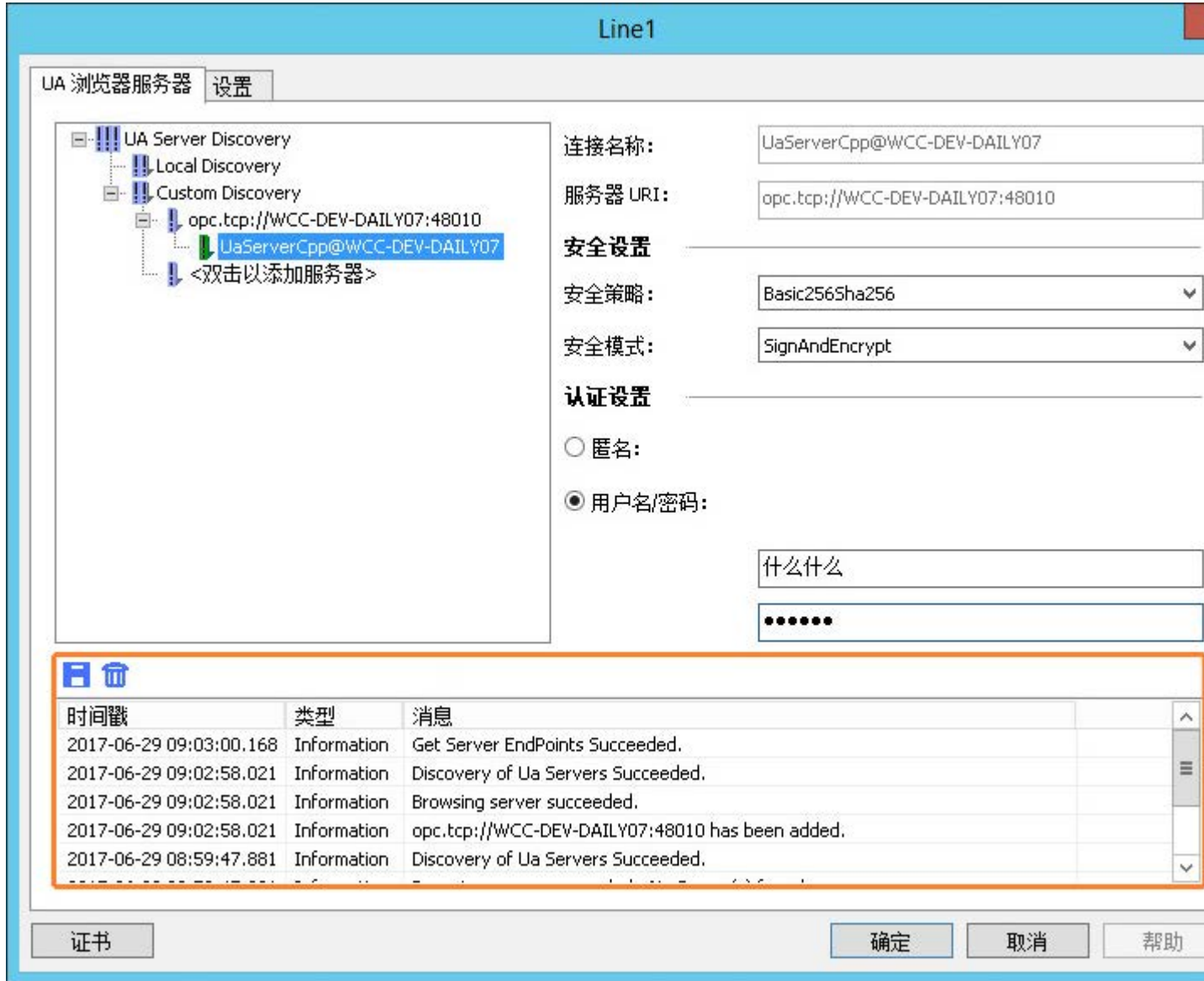
域/设置	内容
UA Server Discovery	UA Discovery Server 提供了可用 OPC UA 服务器的列表。
Local Discovery	Local Discovery 列出了本地计算机中的所有 OPC UA Server，这些服务器已注册到 Local Discovery Server (LDS)。 要求：在本地计算机上已安装 LDS。
Custom Discovery	使用 Custom Discovery，用户可通过其连接名称手动输入 OPC UA Server。 这对于 OPC UA Server 位于远程计算机中的情况尤为必要。 如果 OPC UA Server 未注册到 Discovery Server，则采用以下格式输入 OPC UA Server 的 Discovery 地址： <ul style="list-style-type: none"> • <opc.tcp://Discovery 服务器地址: 端口号> 使用服务器列表快捷菜单的“浏览”(Browse) 命令，刷新服务器显示。
连接名称	OPC UA 服务器的连接名称。 OPC UA 服务器的名称不受此域影响。
服务器 URI	OPC UA 服务器的地址。
安全策略	选择 OPC UA 服务器提供的其中一个安全策略。 <ul style="list-style-type: none"> • 无 • Basic128Rsa15 • Basic256 • Basic256Sha256 • 自动选择最安全的选项
安全模式	选择所需安全模式。 <ul style="list-style-type: none"> • Sign • SignAndEncrypt
验证设置	选择连接是否需要用户 ID 或是否允许匿名访问。 如果已设置用户标识，则输入 WinCC OPC UA 客户端访问 OPC UA 服务器时所需的用户名和密码。 由 OPC UA 服务器检查授权。 对于 WinCC OPC UA 服务器，可以通过运行 OPC UA 服务器的 PC 的 Windows 用户管理组态授权。
证书	打开安装目录中的“PKI”文件夹。

6.5 OPC UA WinCC 通道

域/设置	内容
ConnectTimeout (ms)	建立连接的超时时间。 对于连接而言，ConnectTimeout 会覆盖 CallTimeout，因为服务器可能会在其它 PC 上运行，且可远程访问。
在未指定 IndexRange 的情况下写入数组条目	激活或取消激活用于远程访问的 IndexRange 支持。

控制台

“控制台”(Console) 输出窗口包含状态信息和错误消息（带时间戳和消息类型）。



删除消息

使用以下按钮可删除所有条目：


注意

删除：无法“撤消”

删除的条目无法恢复。

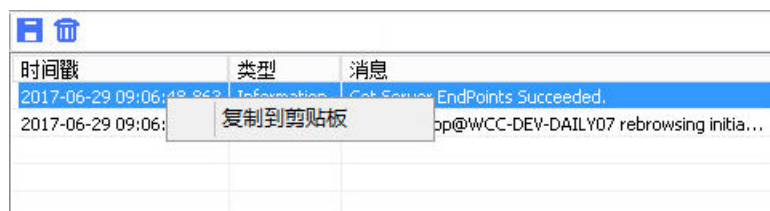
6.5 OPC UA WinCC 通道

保存消息

通过单击以下按钮可将消息存储到日志文件中：

从此处打开的对话框中选择存储位置。

可通过消息的快捷菜单将各个消息复制到剪贴板中。



参见

如何组态到 OPC UA 服务器的连接 (页 279)

通过 NAT 路由器创建与 OPC UA 服务器的连接 (页 270)

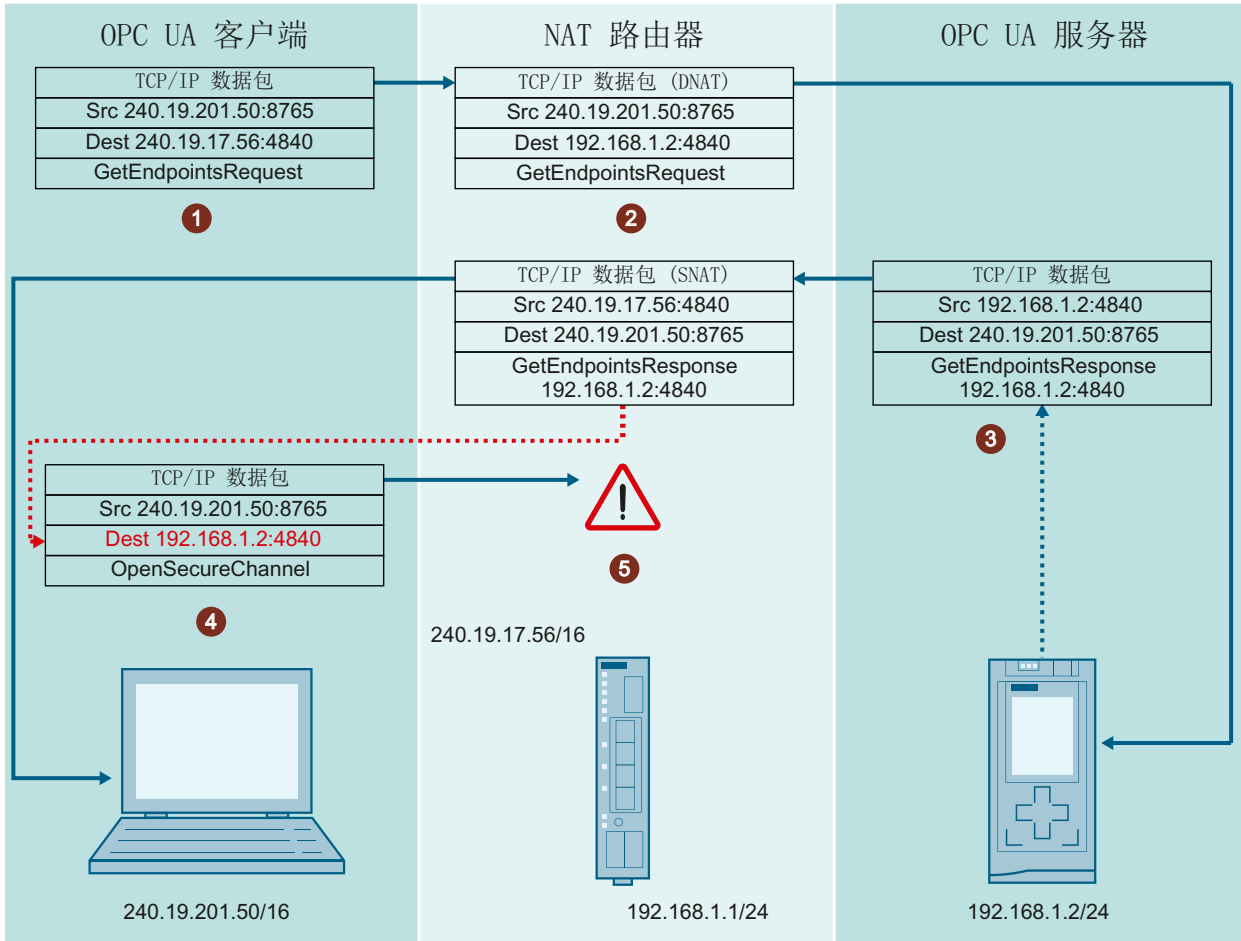
6.5.3.4 通过 NAT 路由器创建与 OPC UA 服务器的连接

通过 NAT 路由器与 OPC UA 服务器建立连接可能会失败，并且将显示错误消息“BadCommunicationError”或“BadNotConnected”。

IPv4 数据包由 NAT 系统中的路由器处理。结果是，数据包的源 IP（“源 NAT”）或目标 IP（“目标 NAT”）替换为路由器中组态的 IP 地址（取决于目标端口）。

错误的连接建立过程

以下示例说明了通过 NAT 路由器建立的错误连接：



1. OPC UA 客户端发送目标 IP 地址 (DNAT) 为 240.19.17.56 的“GetEndpointsRequest”。
2. NAT 路由器将数据包的目标 IP 更改为 192.168.1.2，以便将数据包发送到 OPC UA 服务器。
3. OPC UA 服务器生成一个包含自身 IP 地址 (192.168.1.2) 的“GetEndpointResponse”，并通过 NAT 路由器将电报发送回客户端。
为了使客户端能够分配电报，路由器需要将数据包的源 IP 地址 (SNAT) 调整为 240.19.17.56。
4. 客户端开始与服务器的“GetEndpointResponse”中包含的 IP 地址建立连接 (“OpenSecureChannel”)。
NAT 路由器无法处理服务器的“GetEndpointResponse”中的 IP 地址。
5. 但是，无法直接从客户端访问响应电报的 IP 地址。
因此，无法建立连接。

解决方案：通过 OPC UA 客户端比较 IP 地址

通过 OPC UA 客户端将“GetEndpointRequest”的 IP 地址与 OPC UA 服务器发送的“GetEndpointResponse”的 IP 地址进行比较。

如果 IP 地址不同，则 OPC UA 客户端会将“GetEndpointResponse”的 IP 地址替换为“GetEndpointRequest”的 IP 地址。

这样 OPC UA 客户端便可与经处理的“GetEndpointResponse”中的信息相连接。

证书验证期间的故障排除

如果连接尝试终止并显示错误消息“BadCertificateHostNameInvalid”，可以通过“ValidationOptions”抑制组态客户端和运行系统客户端的组态文件中的错误。

必须在“TrustedCertificateStore”和“IssuerCertificateStore”中设置“SuppressHostNameInvalid”标记。

更多相关信息，请参见“设置通过证书进行认证。(页 272)”。

参见

创建与 OPC UA 服务器的连接 (页 264)

6.5.3.5 设置通过证书进行认证。

服务器证书和客户端证书

组态时区分客户端和服务器证书。仅当客户端和服务器能识别对方的证书时，才能进行安全通信。

将证书链接到各自的计算机。将 WinCC 项目移动、拷贝或复制到其它计算机后，请重复该过程，从而使各计算机都能检查其它计算机的证书。

诊断

使用控制台和 WinCC 通道诊断进行分析。

更多相关信息，请参见：

- “接口 > OPC - 开放式互连 > WinCC OPC UA 服务器 > OPC UA 的安全概念 (页 712)”
- 通信 > 通信诊断 > 诊断通道“OPC” (页 610)

使用证书

创建 WinCC OPC UA Client 的自签名证书以供安装。

只有 OPC UA Server 将 OPC UA Client 证书识别为可信时，此客户端才能连接至该服务器。

创建新连接时，OPC UA Server 会检查客户端证书。对于通过 WinCC 通道“OPC UA WinCC 通道”进行的通信，服务器必须将组态客户端和运行系统客户端的证书识别为可信任。

证书位于 WinCC 安装路径的以下文件夹中：

证书	opc\UAClient\PKI\OPCUA\certs
<ul style="list-style-type: none"> • Siemens OPC UA Tag Importer for WinCC • Siemens OPC UA Client for WinCC Runtime 	
私钥	opc\UAClient\PKI\OPCUA\private

被拒绝的证书

如果服务器未将客户端证书识别为可信，则该连接会被拒绝并以红色标记。

将在控制台中生成一条消息，或在日志文件中输入一条消息，例如：

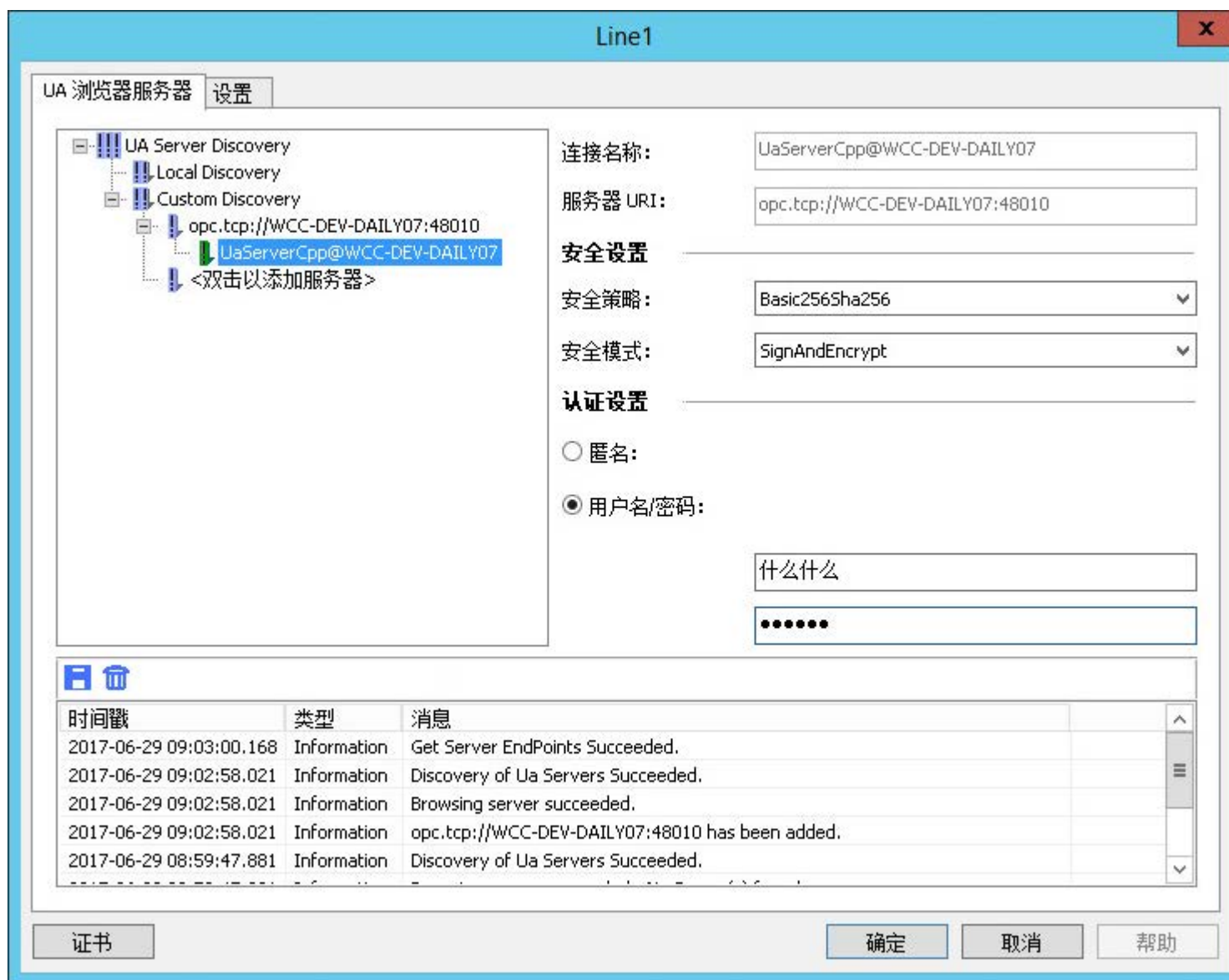
- Discovery of UA Server failed - The Certificate is not trusted.

被拒绝的证书均存储在“PKI...\Rejected\Certs”文件夹中。

信任证书

要指定证书受信，将该证书移至“Trusted\Certs”文件夹中。

为此，单击“UA 服务器浏览器”(UA Server Browser) 选项卡中的“证书”(Certificates) 按钮，打开“PKI”文件夹。



将不可信证书颁发机构的证书移入“Issuer\certs”文件夹，如果可行，还可移入相应证书吊销列表的“Issuer\crl”文件夹。

组态文件

说明	应用程序	组态文件
组态客户端 证书: Siemens OPC UA Tag Importer for WinCC 未找到有效客户端证书时, 会中止尝试建立连接。 存储路径: <ul style="list-style-type: none"> • opclUAClient\UaConfigServer 	CCOpcUaImporter.exe	CCOpcUaImporter.xml
运行系统客户端 证书: Siemens OPC UA Client for WinCC Runtime 在没有有效运行系统证书的情况下, 不会在运行系统中显示当前值。 存储路径: <ul style="list-style-type: none"> • opclUAClient\UaDAS 	CcUaDAS.exe	CcUaDAS.xml

在两个客户端的组态文件中, 可在 <CertificateDescriptor> 部分的 <ClientConfiguration> 中找到证书参数。

示例: 用于控制证书的参数

```

<ClientConfiguration>
  <CertificateDescriptor>
    <OrganizationUnit>DF PL DER HMI</OrganizationUnit>
    <Organization>Siemens AG</Organization>
    <Country>DE</Country>
    <KeyLength>2048</KeyLength>
    <SignatureAlgorithm>SHA256</SignatureAlgorithm>
    <LifetimeInMonths>60</LifetimeInMonths>
  </CertificateDescriptor>

```

说明

参数	含义
OrganizationUnit Organization Country	描述性元素 参数可以更改，且不会影响应用程序的功能。
KeyLength	用于创建证书的私钥的长度 该长度取决于签名算法。 <ul style="list-style-type: none"> • 1024：通过 OPC UA 进行安全通信的最小长度 • 2048：使用 Sha256 时的最小长度¹⁾
SignatureAlgorithm	用于签名证书的签名算法 <ul style="list-style-type: none"> • 可能值：Sha1、Sha224、Sha256、Sha384、Sha512 • 常用值：Sha1、Sha256 • 默认值：私钥长度为 2048 的 Sha256¹⁾
LifetimeInMonths	证书的有效期限按月计 在指定的时间到期后，将无法再使用此证书对客户端进行操作。 <ul style="list-style-type: none"> • 默认值：60

1) 要使用安全策略“Basic256Sha256”建立与 OPC UA 服务器的安全连接，服务器以及 OPC UA 客户端需要具有以下值的证书：

- KeyLength：至少 2048
- SignatureAlgorithm：Sha256

创建新的客户端证书

用户需要具备管理员权限才能在 OPC UA 客户端上创建新证书。

创建新证书时，将重置信任设置。以前识别为可信任的证书不再受信任。

1. 创建备份。
2. 删除相应文件夹中的现有证书和相关的私钥。
3. 在组态文件中，更新证书参数并保存 XML 文件。
4. 使用管理员权限在 Windows 中打开 DOS 窗口“cmd.exe”。

5. 要创建证书，请转到相应 OPC UA 应用程序的安装路径。

6. 输入相应调用：

– CCOpcUaImporter.exe/CreateCertificate

– CcUaDAS.exe/CreateCertificate

在存储路径中创建新证书和私钥。

指定新证书是可信的。

自动创建客户端证书

如果在应用证书存储路径中未发现任何证书并且组态文件中的“ForceCertificateCreation”标记为“true”，则客户端将创建自己的实例证书。

客户端启动后立即创建实例证书。

```
- <ClientConfiguration>  
  + <CertificateDescriptor>  
    <ForceCertificateCreation>true</ForceCertificateCreation>
```

抑制证书检查错误

说明

必须在证书检查中告知操作员已抑制的错误。

证书将经过严格检查。

可以选择抑制客户端的两个应用的某些已知错误。

客户端自己的实例证书

客户端启动时，将检查其自己的实例证书。

如果证书无效或已过期，则无法与服务器建立连接。

可以选择安装有效证书。

也可以在客户端的组态文件中抑制发生的错误。

为此，请修改“ApplicationCertificate”下的“ValidationOptions”。

可信证书和证书颁发机构证书

服务器证书检查可能由于证书无效而失败。可以在客户端的组态文件中抑制发生的错误。

为此，请修改“TrustedCertificateStore”和“IssuerCertificateStore”下的“ValidationOptions”。

说明

“TrustedCertificateStore”和“IssuerCertificateStore”下的“ValidationOptions”标记必须相同。

“ValidationOptions”

标记	说明	注释
SuppressCertificateExpired	忽略由于证书或证书颁发机构的有效到期而发生的错误。	
SuppressHostNameInvalid	忽略应用的主机名或统一资源标识符 (URI) 无匹配项错误。	在服务器端进行验证时跳过该检查。
SuppressRevocationStatusUnknown	忽略由于缺少证书颁发机构的证书吊销列表而发生的错误。	默认抑制该错误。
CheckRevocationStatusOnline	在线检查吊销状态。	不支持该功能。
CheckRevocationStatusOffline	离线检查吊销状态。	跳过该检查。
UseDefaultOptions	使用标准选项：检查有效期和主机名。	

示例：证书的“ValidationOptions”

```

<StoreType>Directory</StoreType>
<StorePath>[ApplicationPath]\..\PKI\OPCUA</StorePath>
<SubjectName>Siemens OPC UA Client for WinCC Runtime</SubjectName>
<Thumbprint/>
<RawData/>
  <!--One or more of: UseDefaultOptions, SuppressCertificateExpired, SuppressHostNameInvalid-->
  <ValidationOptions>UseDefaultOptions</ValidationOptions>
  <OfflineRevocationList/>
  <OnlineRevocationList/>
</ApplicationCertificate>
- <TrustedCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\..\PKI\Trusted</StorePath>
  <!--One or more of: UseDefaultOptions, SuppressCertificateExpired, SuppressHostNameInvalid,
  SuppressRevocationStatusUnknown, CheckRevocationStatusOnline, CheckRevocationStatusOffline -->
  <ValidationOptions>UseDefaultOptions</ValidationOptions>
</TrustedCertificateStore>
+ <TrustedCertificates>
- <IssuerCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\..\PKI\Issuers</StorePath>
  <!--One or more of: UseDefaultOptions, SuppressCertificateExpired, SuppressHostNameInvalid,
  SuppressRevocationStatusUnknown, CheckRevocationStatusOnline, CheckRevocationStatusOffline -->
  <ValidationOptions>UseDefaultOptions</ValidationOptions>
</IssuerCertificateStore>
- <IssuerCertificates>

```

参见

OPC UA 的安全概念 (页 712)

如何组态到 OPC UA 服务器的连接 (页 279)

通过 NAT 路由器创建与 OPC UA 服务器的连接 (页 270)

“OPC”通道的诊断 (页 610)

6.5.3.6 如何组态到 OPC UA 服务器的连接

简介

本节说明如何连接 OPC UA 服务器。

要求

- OPC UA 服务器已激活。
- 通信不得被防火墙拦截。
OPC UA 服务器的端口号必须激活。

6.5 OPC UA WinCC 通道

- 必须可以通过 IP 地址从 WinCC PC 访问 OPC 服务器的 PC。
- OPC UA 服务器信任客户端证书。
- 已在 WinCC OPC UA 客户端的 WinCC 项目中添加了通道“OPC UA WinCC Channel”。

操作步骤

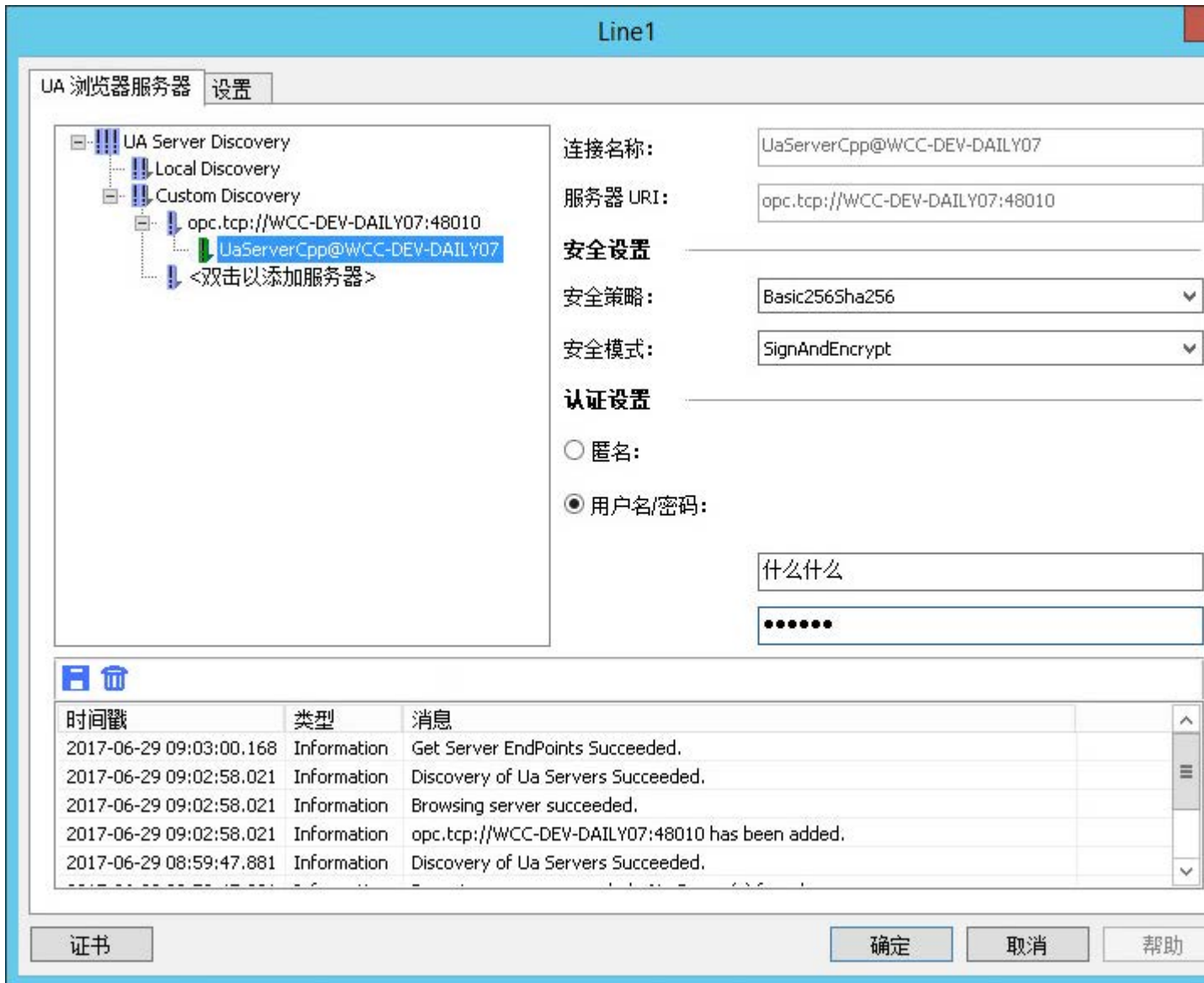
1. 在 WinCC Configuration Studio 中打开 WinCC 变量管理。
2. 使用“OPC UA Connections”的快捷菜单创建新连接。
3. 为连接命名。

说明

仅当运行系统禁用时才能重命名连接。

4. 从所创建连接的快捷菜单中打开“连接参数”(Connection Parameters)对话框。
5. 选择服务器：
 - 要更新本地 OPC UA 服务器的显示，请在“Local Discovery”的快捷菜单中选择“浏览”(Browse)。
 - 要输入 OPC UA 服务器的 URL，请双击“Custom Discovery”下的行。
按照以下格式输入 IP 地址：
- opc.tcp://<OPC-UA-Server_Address:Port_Number>

6. 指定所需设置，并使用“确定”(OK) 按钮确认输入。
已建立与 OPC UA 服务器的连接，该连接在“UA 服务器浏览器”(UA Server Browser) 选项卡中
标记为绿色。



7. 要为连接建立和连接状态创建系统变量，请从连接的快捷菜单中选择“创建启用/禁用变量”(Create Enable/Disable Tags) 条目。
内部变量组“ConnectionStates”中将创建下列变量：
- @<Connectionname>@ForceConnectionStateEx
 - @<Connectionname>@ConnectionStateEx

结果

如果运行系统已激活，则连接在“变量管理器”(Tag Management) 中以绿色复选标记进行标记。

客户端证书

如果 OPC UA 服务器未识别客户端证书，则未建立连接。

连接在“UA 服务器浏览器”(UA Server Browser) 选项卡中标记为红色。连接在变量管理的项目树中标记有红色感叹号。

确保 OPC UA 服务器接受客户端证书。

参见

“符号”(Symbols) 视图 (页 285)

OPC UA 变量 (页 282)

设置通过证书进行认证。(页 272)

6.5.3.7 组态 OPC UA 变量

OPC UA 变量

支持的 OPC UA 节点

建立与 OPC UA 服务器的连接后，将 OPC UA 服务器的对象和 OPC UA 节点加载到“符号”(Symbols) 视图中。

数据类型

支持以下数据类型的变量：

OPC UA 节点	WinCC 变量类型
Binary Tag	二进制变量
Byte	有符号 8 位数或无符号 8 位数
Int16	有符号 16 位数
UInt16	无符号 16 位数
Int32	有符号 32 位数
UInt32	无符号 32 位数

OPC UA 节点	WinCC 变量类型
Float	32 位浮点数 IEEE 754 或 64 位浮点数 IEEE 754
String	文本变量 8 位字符集和文本变量 16 位字符集
ByteString	原始数据变量
DateTime	日期/时间
Enumerations	有符号 32 位数

将 OPC UA 节点作为 WinCC 变量导入

对于可作为 WinCC 变量导入的 OPC UA 节点，可在数据区的“访问”(Access) 列中通过 图标进行访问。对于不受支持的数据类型，会禁用该字段。

要导入与所有 WinCC 数据类型均不匹配的 OPC UA 节点，请手动将匹配的 WinCC 数据类型分配给各个 OPC UA 节点。

OPC UA 节点只能导入一次。

更多相关信息，请参见“导入 OPC UA 节点作为 WinCC 变量 (页 287)”。

WinCC 变量名称

导入 OPC UA 节点时，自动分配 WinCC 变量名称。

加载 OPC UA 节点后，可在相应连接的属性中组态变量名称的设置。

如果 OPC UA 服务器上的变量名称含有特殊字符，则将替换为下划线“_”。

名称生成设置

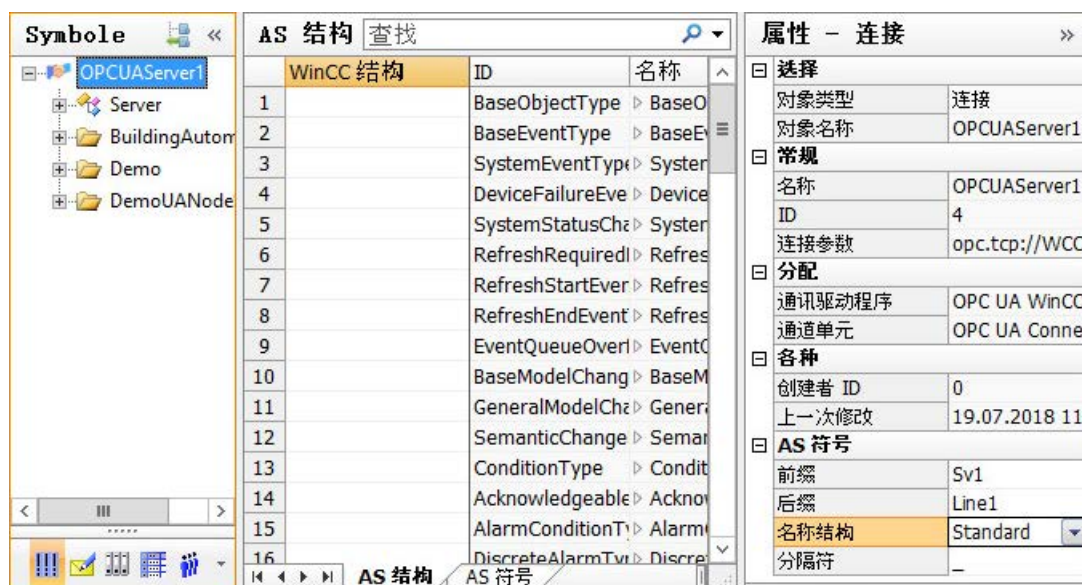
要在“AS 符号”(AS symbols) 属性组中显示设置，请在“符号”(Symbols) 视图中单击连接名称。

可进行下列设置：

- OPC UA 节点路径可用作名称。
- 应用 OPC UA 节点名称，并在适当的情况下，通过前缀或后缀补充该名称。
- 应用 OPC UA 节点路径，并在适当的情况下，通过前缀或后缀补充该路径。

“前缀”(prefix) 或“后缀”(suffix) 选项可向变量名称添加指定字符串。

节点名称的组件通过分隔符连接。默认情况下使用下划线。



示例

OPC UA 服务器的路径“Spectrometer/Channel_0/ChannelStateMachine”中有“CurrentState”变量。

在“前缀”(Prefix)域中输入“Prefix_”并在“后缀”(Suffix)域中输入“_Suffix”。

在 WinCC OPC UA 客户端的 WinCC 项目中创建以下 WinCC 变量：

设置	WinCC 变量名称
不带前缀和后缀的路径名称	Spectrometer/Channel_0/ChannelStateMachine/CurrentState
OPC UA 节点名称	Prefix_CurrentState_Suffix
路径名称	Prefix_Spectrometer/Channel_0/ChannelStateMachine/CurrentState_Suffix

删除 WinCC 变量

要删除 WinCC 变量，必须与 OPC UA 服务器没有活动连接。

要删除 WinCC 变量管理中已导入的 WinCC 变量，请选择 WinCC 变量快捷菜单中的“删除”(Delete) 条目或使用 键。

创建变量组

要在连接下创建变量组，请在连接快捷菜单中选择“新建组”(New group)。

要更改名称，单击变量组名称。

创建 WinCC 变量时，应留意以下操作：

- 在项目导航窗口中选择了连接：
 - 直接在连接下创建 WinCC 变量。
 - “变量”(Tags) 数据区域会显示所有变量，即使变量分配给变量组的情况下也不例外。“组”(Group) 字段可能包含已分配的变量组。可通过下拉列表更改组分配。
- 在项目导航窗口中选择了变量组：
 - 在变量组中创建 WinCC 变量。
 - “变量”(Tags) 数据区只显示已在变量组中创建的变量。

WinCC 项目的移植

低于 V7.4 版本的 WinCC，已在 OPC 通道中创建了 WinCC OPC UA 连接。

移植 WinCC 项目期间，WinCC OPC UA 客户端的连接和变量也将移植到更改的结构中。

如果已导出 WinCC OPC UA 变量，请注意以下顺序：

1. 导入已导出的 WinCC OPC UA 变量。
2. 移植 WinCC 项目。

参见

“符号”(Symbols) 视图 (页 285)

导入 OPC UA 节点作为 WinCC 变量 (页 287)

“符号”(Symbols) 视图

简介

成功进行连接组态后，即可访问 OPC UA 服务器上的 OPC UA 节点。

要创建 WinCC 变量，在“变量管理器”(Tag Management) 中将 OPC UA 节点加载至“符号”(Symbols) 视图中。

符号显示

数据结构的表示

结构树中的数据表示对应于 OPC UA 服务器中的层级。

6.5 OPC UA WinCC 通道

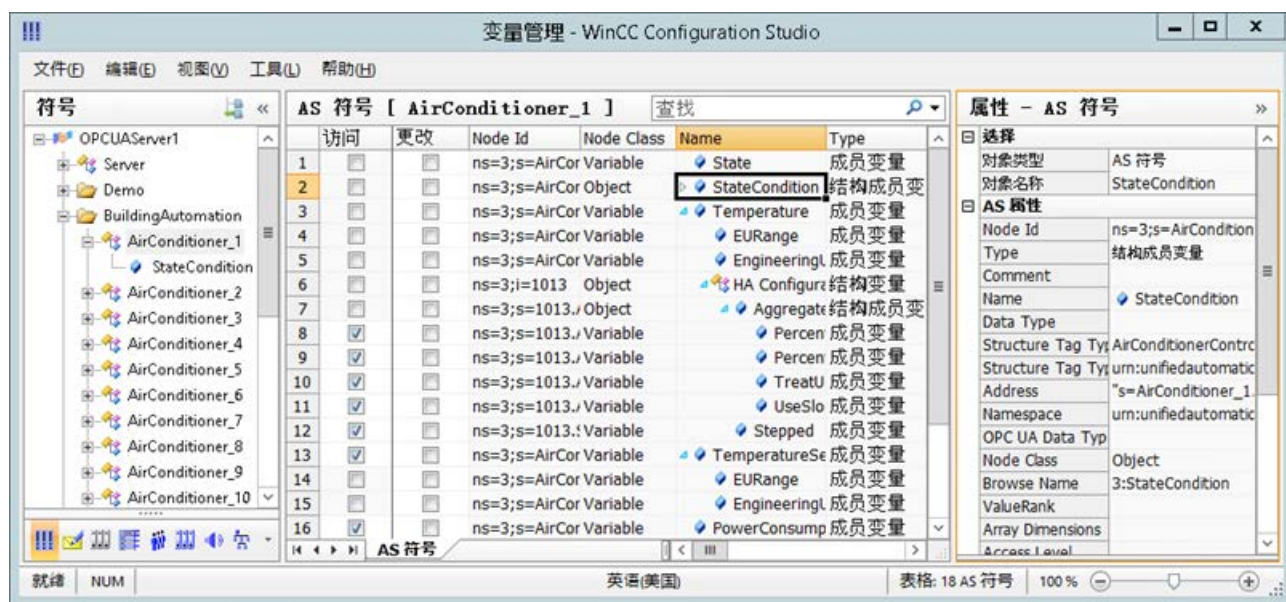
复杂数据结构将显示在数据区中。

要在数据区中显示较低层级的结构，请在“名称”(Name)列中单击要扩展的符号：▶。

链接 WinCC 变量

在“AS 符号”(AS Symbols)和“AS 结构”(AS structures)选项卡中，可指定与 WinCC 变量链接的 OPC UA 节点。

仅可在 OPC UA 服务器上更改 OPC UA 节点的属性。



“AS 结构”(AS structures) 选项卡

如果加载的数据还包含结构，还会额外显示“AS 结构”(AS structures) 选项卡。

在导航区域中选择连接名称后，将显示该选项卡。

还可以在“变量管理器”(Tag Management) 的默认视图中组态结构分配。

单击通信通道“OPC UA WinCC Channel”下的连接可显示“AS 结构”(AS structures) 选项卡。

将视图切换为

在“变量管理器”(Tag Management) 中，可使用以下按钮在默认视图和“符号”(Symbols) 视图之间进行切换：

该按钮仅在加载数据记录后才可用。

WinCC Configuration Studio 关闭后，包含“AS 符号”(AS Symbols)和“AS 结构”(AS structures) 选项卡的“符号”(Symbols) 视图将再次隐藏。

在默认视图下，仅当 OPC UA 节点再次加载后，“AS 结构”(AS structures) 选项卡才会再次显示。

参见

如何使用自动生成的结构类型 (页 291)

导入 OPC UA 节点作为 WinCC 变量 (页 287)

如何组态到 OPC UA 服务器的连接 (页 279)

OPC UA 变量 (页 282)

导入 OPC UA 节点作为 WinCC 变量

简介

本节说明如何将 OPC UA 节点作为 WinCC 变量导入到 WinCC 变量管理中。

在 WinCC Configuration Studio 中的通道单元“OPC UA Connections”下创建了 OPC UA WinCC 通道变量。

数据类型的分配

支持的 OPC UA 节点类型自动分配给相应的 WinCC 变量类型。可以在“OPC UA 变量 (页 282)”中找到分配表。

对于不受支持的 OPC UA 节点，请一开始就禁用“访问”(Access) 选项。这些节点不会导入到变量管理中。

要导入与所有 WinCC 数据类型均不匹配的 OPC UA 节点，请手动分配匹配的 WinCC 数据类型。

要求

- 已建立与 OPC UA 服务器的连接。

操作步骤

1. 在“OPC UA Connections”下的导航区域中选择组态的连接。
2. 从连接的快捷菜单中选择“浏览 OPC 服务器”(Browse OPC server) 条目。
OPC UA 服务器的可用数据已加载，“符号”(Symbols) 视图打开。
加载的数据显示在表格区的“AS 符号”(AS Symbols) 选项卡中。
如果加载的数据还包含结构，还会额外显示“AS 结构”(AS structures) 选项卡。
3. 在“符号”(Symbols) 视图的导航区域中选择连接。

6.5 OPC UA WinCC 通道

4. 在“AS 符号”(AS Symbols) 组的 “属性 - 连接”(Properties - Connection) 部分中选择 WinCC 变量名称的选项：
 - 名称结构
 - 分隔符
 - 前缀和后缀（若适用）
5. 在导航区域中选择所需条目。
包含的相应已加载 OPC UA 节点显示在“AS 符号”(AS Symbols) 选项卡上。
6. 要为所需 AS 符号创建 WinCC 变量，任何情况下都要激活 “访问”(Access) 列。
要在 WinCC 中导入所选对象的所有受支持的 OPC UA 节点，请从 “访问”(Access) 列的快捷菜单中选择 “全选”(Select All)。
7. 要导入不受支持的 OPC UA 节点，请在“Data Type”列中选择匹配的 WinCC 数据类型。
现在可以选择 OPC UA 节点的 “访问”(Access) 字段。
启用 “访问”(Access) 选项后，OPC UA 节点将作为具有所选数据类型的 WinCC 变量导入。

结果

将在 WinCC 变量管理中看到新组态的 WinCC 变量。

但是，只能在 OPC UA 服务器中更改变量属性。

使用 OPC UA 服务器同步 WinCC 变量

从控制器或文件进行加载后，变量管理器会将 AS 符号的属性与链接的 WinCC 变量进行比较。如果符号的属性不匹配，“AS 符号”(AS Symbols) 选项卡的 “已修改”(Modified) 字段会激活。相应的属性字段会以红色突出显示。字段的工具提示包含额外信息。

更新 WinCC 变量

要应用 OPC UA 节点的当前属性，请禁用 “已修改”(Modified) 字段。

或者，禁用 “访问”(Access) 字段后再次激活，以重新创建 WinCC 变量。

参见

如何使用自动生成的结构类型 (页 291)

“符号”(Symbols) 视图 (页 285)

6.5.3.8 在 WinCC 中使用 OPC UA 类型

将 OPC UA 类型导入为 WinCC 结构类型

简介

本节说明如何将 OPC UA 服务器的对象类型或对象导入 WinCC 变量管理器中。

目标是将 OPC UA 对象轻松组态为 WinCC 中的结构变量。

概述：基本步骤

1. 分配对象类型
2. 组态对象

导入的对象创建为 WinCC 结构或结构变量，并映射如下：

OPC UA	WinCC
OPC UA 对象类型	结构类型
Properties/OPC UA 对象类型的变量	结构类型元素
OPC UA 对象	结构变量
Properties/OPC UA 对象的变量	结构变量元素

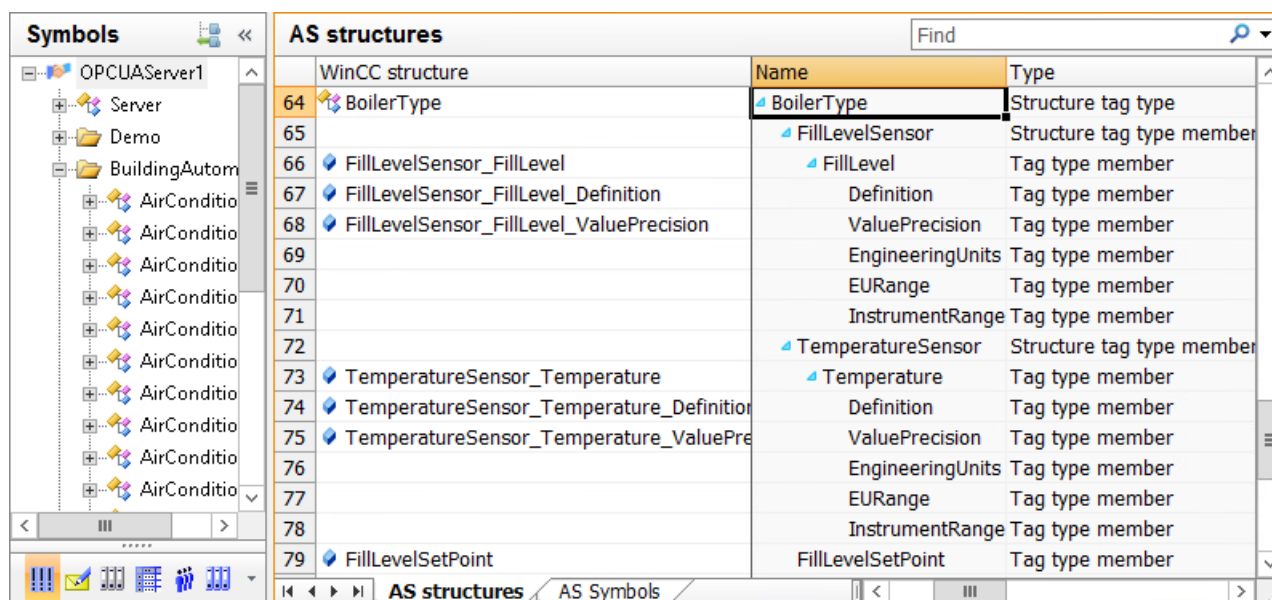
组态 STEP 1：分配对象类型

如果在“符号”(Symbols) 视图的导航区域中选择连接名称，则会显示“AS 结构”(AS structures) 选项卡。

可将 OPC UA 对象类型与“AS 结构”(AS structures) 选项卡中的 WinCC 结构类型链接。

OPC UA 对象类型的属性和变量与结构类型元素链接。

可以自动创建 WinCC 结构类型和结构类型元素，也可以分配已创建的 WinCC 结构类型。



自动分配

正在自动创建 WinCC 结构类型和结构类型元素。

在 WinCC 变量管理器中创建具有 OPC UA 结构名称的结构类型。

为可在 WinCC 中映射的 OPC UA 对象类型的属性和变量创建结构类型元素。

OPC UA 对象类型的层级结构通过结构类型元素的名称进行映射，例如“FillLevelSensor_FillLevel_Definition”。

说明

变量名称的最大长度

请注意，WinCC 变量名称最多包含 128 个字符。

对于结构变量元素，此限制适用于整个表达式：

- 结构变量名称 + 句点 + 结构类型元素名称

手动分配

在 WinCC 变量管理器中创建结构类型和结构类型元素。确保结构类型元素的数据类型和相应属性或变量的 Data Type 始终相同。

将读取的 OPC UA 对象类型与创建的 WinCC 结构类型链接。

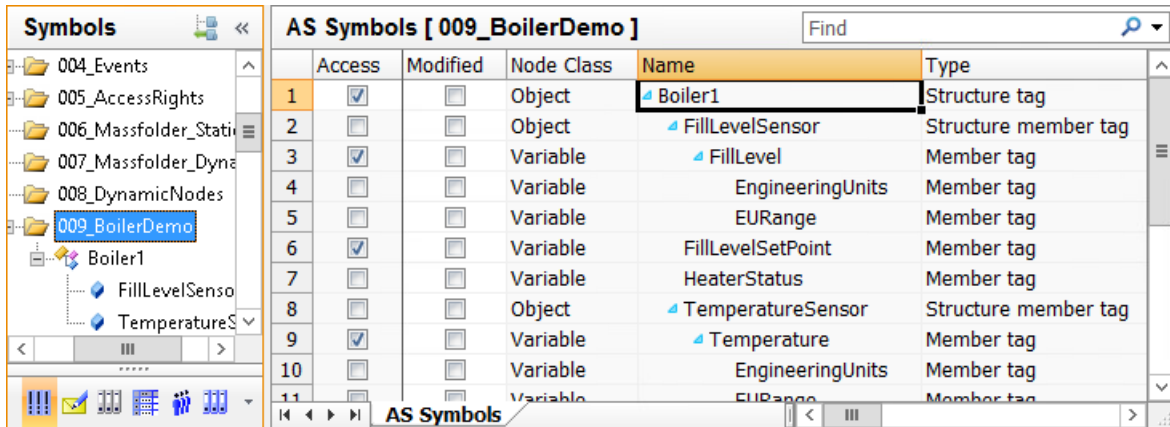
如果结构类型元素和属性或变量具有相同的名称和数据类型，则可对其自动分配。

或者，可将结构类型元素分别分配给属性和变量。

组态 STEP 2: 分配对象

可在“AS 符号”(AS Symbols) 选项卡中将 OPC UA 对象组态为 WinCC 结构变量。

激活 OPC UA 对象的访问权限后，就会自动创建结构变量和结构变量元素。



如何使用自动生成的结构类型

简介

在此方法中，在导入 OPC UA 对象时将自动创建 WinCC 结构类型和结构类型元素。

要求

- 已建立与 OPC UA 服务器的连接。

操作步骤

1. 从 OPC UA 连接的快捷菜单中选择“浏览 OPC 服务器”(Browse OPC server) 条目。OPC UA 服务器的可用数据已加载。
“AS 结构”(AS structures) 选项卡显示在带 OPC UA 对象类型的“符号”(Symbols) 视图中。在导航区域中选择连接名称后，将显示该选项卡。
要显示对象类型下方的元素，请在“名称”(Name) 字段中单击对象类型名称前的箭头。
2. 要选择 OPC UA 对象类型，请单击行号。
可以选择多个行。

6.5 OPC UA WinCC 通道

3. 从行的快捷菜单中选择“创建结构”(Create structure) 条目。
 - 在 WinCC 变量管理器中创建具有 OPC UA 对象类型名称的结构类型。
 - 为可映射的 OPC UA 对象类型的所有属性和变量创建一个结构类型元素。
 - 层级结构通过结构类型元素的名称进行映射。
 请注意，WinCC 变量名称最多包含 128 个字符。
 如有必要，请在创建结构变量之前缩短结构类型元素的名称。

WinCC 结构		名称	类型
1		▷ AccessPermissionObjectType	结构变量类型
2		▷ AcknowledgeableConditionType	结构变量类型
3		▷ AddressSpaceFileType	结构变量类型
4		▷ AggregateConfigurationType	结构变量类型
5		▷ AggregateFunctionType	结构变量类型
6	▷ AirConditionerControllerType	▷ AirConditionerControllerType	结构变量类型
7	▷ Humidity	▷ Humidity	变量类型成员
8	▷ HumiditySetpoint	▷ HumiditySetpoint	变量类型成员
9	▷ PowerConsumption	PowerConsumption	变量类型成员
10		State	变量类型成员
11		▷ StateCondition	结构变量类型成员
12	▷ Temperature	▷ Temperature	变量类型成员
13		EngineeringUnits	变量类型成员
14		EURange	变量类型成员
15		▷ HA Configuration	结构变量类型成员
16		▷ AggregateConfiguratio	结构变量类型成员
17	▷ Temperature_HA_Configuration_Steppe	Stepped	变量类型成员
18	▷ TemperatureSetPoint	▷ TemperatureSetPoint	变量类型成员

4. 要在“变量管理器”(Tag Management) 视图中编辑结构类型和结构类型元素，请单击以下符号：。
 如有需要，请在“结构变量”(Structure tags) 下更改结构类型或结构类型元素的名称。
 如有必要，请删除不需要的所有结构类型元素。
 这些更改应用于 OPC UA 连接的结构变量中。
5. 要再次显示“符号”(Symbols) 视图，请单击 OPC UA 连接和以下符号：。
6. 要显示 OPC UA 对象，请在导航中选择所需的节点。
7. 在“AS 符号”(AS Symbols) 选项卡中，激活 OPC UA 对象的“访问”(Access) 字段。

访问	更改	Node Class	名称	类型
<input type="checkbox"/>	<input type="checkbox"/>	Variable	ControllerConfigurat	变量
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Object	▷ AirConditioner_1	结构变量
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Object	▷ AirConditioner_2	结构变量
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Object	▷ AirConditioner_3	结构变量
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Object	▷ AirConditioner_4	结构变量

在 OPC UA 对象链接的结构类型中创建结构变量。
 OPC UA 对象的属性和变量映射到结构类型元素。

参见

- 导入 OPC UA 节点作为 WinCC 变量 (页 287)
- “符号”(Symbols) 视图 (页 285)

如何使用手动生成的结构类型

简介

在此方法中，使用在 WinCC 变量管理中创建的结构类型和结构类型元素来导入 OPC UA 对象。

要求

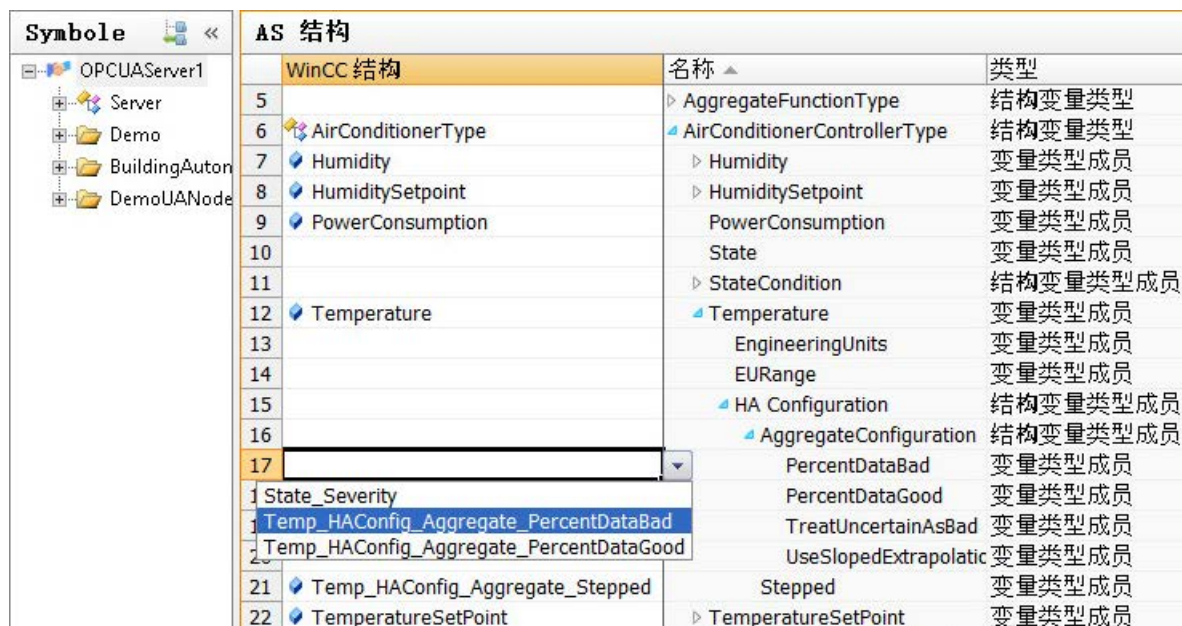
- 已建立与 OPC UA 服务器的连接。
- 在 WinCC 变量管理中创建结构类型。
- 具有以下属性的结构类型元素可在结构类型中组态：
 - 外部：已启用
 - 数据类型：OPC UA 对象类型相应属性或变量的 DataType

操作步骤

1. 从 OPC UA 连接的快捷菜单中选择“浏览 OPC 服务器”(Browse OPC server) 条目。
OPC UA 服务器的可用数据已加载。
“AS 结构”(AS structures) 选项卡显示在带 OPC UA 对象类型的“符号”(Symbols) 视图中。在导航区域中选择连接名称后，将显示该选项卡。
要显示对象类型下方的元素，请在“名称”(Name) 字段中单击对象类型名称前的箭头。
2. 在“WinCC 结构”(WinCC structure) 字段中，选择要分配给 OPC UA 对象类型的已创建结构类型。
与对象类型的属性或变量具有相同名称和数据类型的结构类型元素将自动分配。

6.5 OPC UA WinCC 通道


3. 要为属性或变量分配具有不同名称的结构类型元素，请在“WinCC 结构”(WinCC structure) 字段中选择元素。
列表中包含所有尚未分配且与属性或变量具有相同数据类型的结构类型元素。



4. 要显示 OPC UA 对象，请在导航中选择所需的节点。
5. 在“AS 符号”(AS Symbols) 选项卡中，激活 OPC UA 对象的“访问”(Access) 字段。



在 OPC UA 对象链接的结构类型中创建结构变量。
OPC UA 对象的属性和变量映射到结构类型元素。

6. 要在“变量管理器”(Tag Management) 视图中编辑结构变量，请单击以下符号：。
如有需要，在“结构变量”(Structure tags) 下更改结构类型、结构类型元素或结构变量的名称。
这些更改应用于 OPC UA 连接的结构变量中。

6.5.3.9 在 WinCC 中使用 OPC UA 报警

“受监视的对象”(Monitored objects) 视图

Event Notifier 和报警

成功进行连接组态后，即可访问 OPC UA 服务器上的 Event Notifier。

Event Notifier 可触发报警或事件，用户可将报警作为 WinCC 消息输出。

要为 OPC UA 报警组态 WinCC 消息，在“报警记录”中将 Event Notifier 加载至“受监视的对象”(Monitored objects) 视图中。

注意**不支持本地 WinCC OPC UA 服务器**

未对本地 WinCC OPC UA 服务器启用该功能。

若组态不受支持，则通过 Event Notifier 链接本地 WinCC OPC UA 服务器的 WinCC 消息时会导致“报警记录”连续循环。

WinCC 报警记录中的 OPC Event Notifier

只要在“OPC UA WinCC Channel”通信通道下创建了连接，就会立即在“报警记录”(Alarm Logging) 编辑器中创建“OPC 消息”(OPC messages) 条目。

已创建连接在条目中列出。

可将已连接 OPC UA 服务器的 Event Notifier 加载至“受监视的对象”(Monitored objects) 视图中。

触发 WinCC 消息

要显示 Event Notifier 在 WinCC 项目中触发的报警，将 Event Notifier 链接到 WinCC 消息。

随后，所有触发 Event Notifier 及其层级附属节点的报警都会触发 WinCC 消息。这表示消息数量可能会显著增加。

用户可使用过滤器来确定触发 WinCC 消息的 OPC UA 报警。

这样可以减少触发消息数量，并且仅接受相关事件的报警。

说明

系统性能：避免 Event Notifier “Server”

将 WinCC 消息与更高级别的 Event Notifier“服务器对象”链接时，可能会产生大量消息。即使使用过滤器减少 OPC UA 事件数量，该过程也会对性能产生负面影响。

分配 WinCC 消息

通过消息编号将 Event Notifier 与 WinCC 消息链接起来。

如果已在“报警记录”中创建了消息编号，则会链接此消息。否则，会在“报警记录”中创建具有指定编号的消息。

可将同一条消息链接到多个 Event Notifier。

6.5 OPC UA WinCC 通道

但是，只可对一个 OPC UA 连接使用一条 WinCC 消息。如果创建了多个 OPC UA 连接，则每个连接需使用不同的 WinCC 消息。

分配多条 WinCC 消息

一个 Event Notifier 可链接至多条 WinCC 消息。

当触发报警时，会从上到下对消息和过滤器进行处理。消息编号不会影响 WinCC 消息的触发顺序。

要更改消息顺序，折叠过滤器并在行快捷菜单中选择“上移”(Move up) 或“下移”(Move down)。

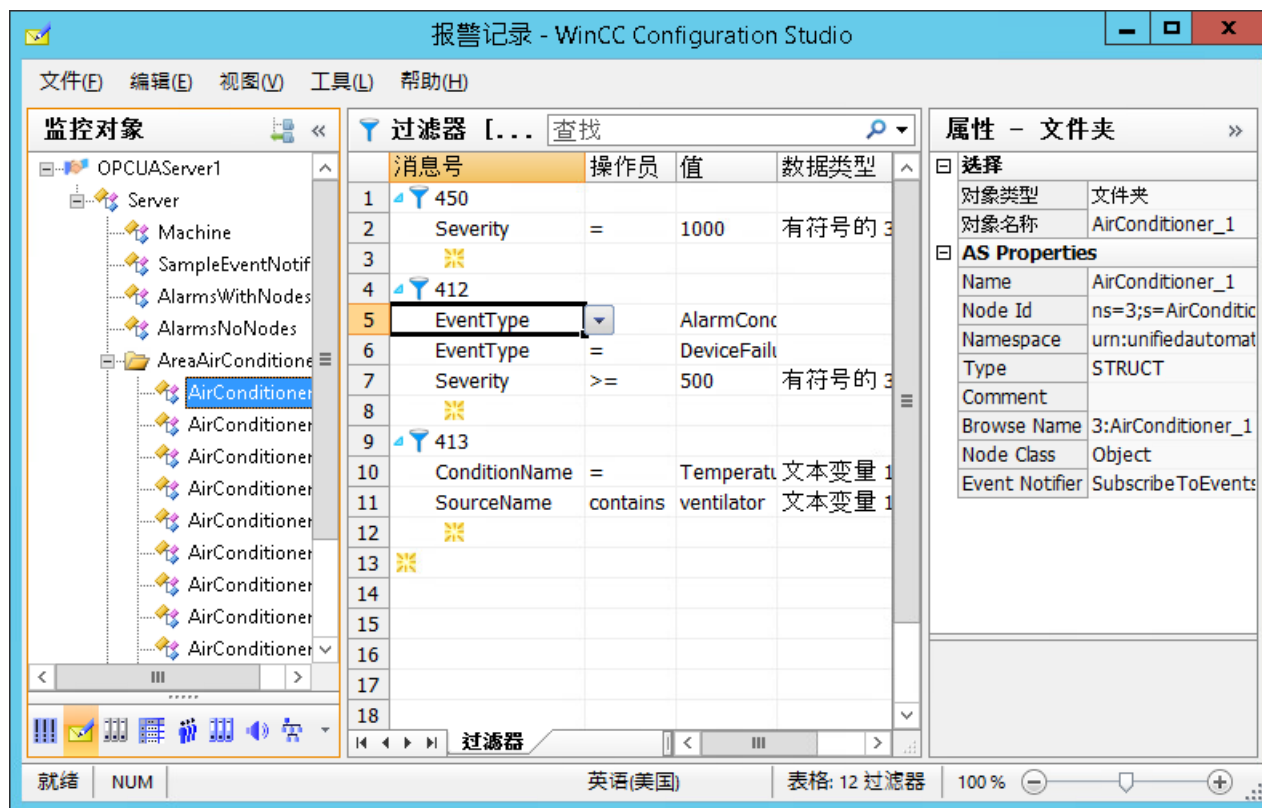
符号显示

结构树中的数据表示对应于 OPC UA 服务器中的层级。


在“过滤器”(Filters) 选项卡中，指定与 Event Notifier 链接的 WinCC 消息。

对于每条消息，可为触发的报警定义一个或多个过滤器。

所选 Event Notifier 的属性显示在“属性 - 文件夹”(Properties - Folders) 区域中。仅可在 OPC UA 服务器上更改属性。



将视图切换为

在“报警记录”(Alarm Logging)中,可使用以下按钮在默认视图和“受监视的对象”(Monitored objects)视图之间进行切换: 

该按钮仅在加载数据记录后才可用。

WinCC Configuration Studio 关闭后,“受监视的对象”(Monitored objects)视图会再次隐藏。

在默认视图“报警记录”(Alarm Logging)中,“受监视的对象”(Monitored objects)选项卡仍然可见。

报警记录:“分配”(Assignments)选项卡

单击“OPC 消息”(OPC messages)可在“报警记录”的默认视图中显示“分配”(Assignments)选项卡。

在此可处组态 OPC UA 报警属性的分配规则。

组态的规则适用于所有 OPC UA 连接。

组态分配规则

如果触发的 OPC UA 报警与 WinCC 消息链接,则其属性将应用于过程值块 1 至 10。

为此,在报警记录的“消息块”(Message blocks)中将过程值块组态为“已使用”(Used)。

过程值块 1 始终包含 OPC UA 报警的消息文本。

对于过程值块 2 至 10,分别从下拉列表框中选择所需属性。

默认规则“Default”无法更改。

报警记录:“受监视的对象”(Monitored objects)选项卡

要在报警记录的默认视图中显示“受监视的对象”(Monitored objects)选项卡,请单击“OPC 消息”(OPC messages)下的连接。

显示与 WinCC 消息链接的 Event Notifier 及其过滤器。

分配分配规则

在该选项卡上,选择 Event Notifier 的分配规则。默认分配“Default”规则。

同一分配规则用于 Event Notifier 及其层级附属节点的所有报警。

在默认视图中编辑过滤器

用户也可以在“受监视的对象”(Monitored objects)选项卡上编辑过滤器。

更改的过滤器用于“受监视的对象”(Monitored objects)视图中的“过滤器”(Filters)选项卡上。

但是，在默认视图中，不会检查过滤条件是否一致及输入是否正确。

测试更改的过滤器，如有必要，在“受监视的对象”(Monitored objects) 视图中修正过滤器。

参见

如何将 Event Notifier 作为 WinCC 消息导入 (页 301)

OPC UA 报警过滤器 (页 298)

OPC UA 报警过滤器

过滤 OPC UA 报警

可为与 Event Notifier 链接的每条 WinCC 消息指定一个或多个过滤器。

一个 Event Notifier 可触发多次报警或事件，通常，WinCC 消息只需其中部分报警或事件。

使用过滤器可将触发的消息限制为事件相关消息。

说明

系统性能：避免 Event Notifier “Server”

将 WinCC 消息与更高级别的 Event Notifier“服务器对象”链接时，可能会产生大量消息。

即使使用过滤器减少 OPC UA 事件数量，该过程也会降低系统性能。

WinCC 消息：唯一分配

确保定义特定的过滤器，以尽可能清楚地对 OPC UA 报警或事件和 WinCC 消息进行分配。

应以此方式对 WinCC 消息进行组态，从而使其对报警或事件的属性进行映射，例如，确认方式和消息来源 (Source)。

组态过滤器

用户可使用过滤器来定义触发 WinCC 消息的 OPC UA 报警或 OPC UA 事件。

可将一条消息链接到多个 Event Notifier，但通过不同的报警对其进行过滤。

可在“受监视的对象”(Monitored objects) 视图中的“过滤器”(Filters) 选项卡上组态过滤器。

要创建过滤器，请单击消息编号前面的箭头。在显示的行中，选择过滤条件、运算符和值。

会自动添加过滤条件对应的数据类型且无法更改。

在线组态

如果在运行系统中更改过滤器，则会立即应用更改。

过滤条件和运算符

通过过滤条件，可确定 Event Notifier 报警触发链接的消息时必须满足的条件。

运算符取决于选择的过滤条件。

过滤条件	运算符		说明
EventType	=		类型的下拉列表 “BaseEventType”值： <ul style="list-style-type: none"> 返回所有 OPC UA 报警或 OPC UA 事件（无过滤）。
ConditionName SourceName	= contains		自由文本输入 必须注意大小写。 运算符“contains”： <ul style="list-style-type: none"> 包含输入的文本。 不使用占位符。
Severity	=	等于	数值输入 值范围： <ul style="list-style-type: none"> 1 到 1000 WinCC 消息中的映射： <ul style="list-style-type: none"> 优先级 0 = Severity 1 优先级 1 到 15 = 在 0 和 1000 之间线性插值 优先级 16 = Severity 1000 示例“between”： <ul style="list-style-type: none"> 100、200 对应于 Severity 100 至 200（分别包括指定的值）
	!=	不等于	
	>	大于	
	<	小于	
	>=	大于或等于	
	<=	小于或等于	
	between	范围跨度	

组合过滤条件

可以组合过滤器的过滤条件或多次使用同一过滤条件：

- 不同的过滤条件通过“AND”链接。
- 相同的过滤条件通过“OR”链接。
- 过滤器无层级。输入的过滤条件的顺序对过滤器应用无影响。

示例：“报警记录”(Alarm Logging) 视图

在报警记录的默认视图中，会为每个连接列出与 WinCC 消息链接的 Event Notifier。

在 Event Notifier 的节点下，将显示消息编号，并在编号下方显示相应的过滤器。

此外，可以使用该视图将多个 Event Notifier 的过滤器互相同步。

在该示例中，可以看到“OPCUAServer1”连接的 Event Notifier：

报警记录		监控对象 [OPCUAServer1]									
消息	消息块	消息组	系统消息	限值监视	AS 消息	OPC 消息	OPCUAServer1	消息号	操作员	值	数据类型
								1			
								2			
								3		1000	有符号的 32 位
								4			
								5			
								6		AlarmConditionType	
								7		DeviceFailureEventType	
								8		500	有符号的 32 位
								9			
								10			
								11	contains	ventilator	文本变量 16 位
								12	=	TemperatureHigh	文本变量 16 位
								13	contains	cool	文本变量 16 位
								14	=	AlarmConditionType	
								15			
								16			
								17			
								18			
								19			
								20			
								21			
								22			
								23		1000	有符号的 32 位
								24			
								25			
								26		AlarmConditionType	
								27		DeviceFailureEventType	
								28			
								29			
								30		AcknowledgeableCondi	
								31	contains	heating	文本变量 16 位
								32	=	torch	文本变量 16 位
								33	=	flame	文本变量 16 位

过滤器示例

- EventType = AlarmConditionType
- EventType = DeviceFailureEventType
- Severity >= 500

该示例对应于以下条件：

- (EventType=AlarmConditionType OR EventType=DeviceFailureEventType) AND Severity>=500

参见

“受监视的对象”(Monitored objects) 视图 (页 294)

如何将 Event Notifier 作为 WinCC 消息导入 (页 301)

如何将 Event Notifier 作为 WinCC 消息导入

简介

本节说明如何将 OPC UA 服务器的 Event Notifier 与 WinCC 消息连接。

在运行系统中，Event Notifier 的 OPC UA 报警可触发 WinCC 报警记录中的消息，并可在 WinCC 报警控件中进行归档与显示。

概述：基本步骤

1. 加载 WinCC 报警记录中的 OPC UA Event Notifier
2. 将 Event Notifier 与 WinCC 消息编号链接
3. 可选：确定 Event Notifier 触发报警的过滤器
4. 可选：定义分配规则
5. 为每个 Event Notifier 指定分配规则
6. 可选：组态 WinCC 消息的属性、显示和归档

说明

系统性能：避免 Event Notifier “Server”

将 WinCC 消息与更高级别的 Event Notifier“服务器对象”链接时，可能会产生大量消息。即使使用过滤器减少 OPC UA 事件数量，该过程也会对性能产生负面影响。

WinCC 消息：唯一赋值

确保定义特定的过滤器，以尽可能清楚地对 OPC UA 报警或事件和 WinCC 消息进行分配。应以此方式对 WinCC 消息进行组态，使其可对报警或事件的属性进行映射，例如，确认原理和消息来源 (Source)。

注意**不支持本地 WinCC OPC UA 服务器**

未对本地 WinCC OPC UA 服务器启用该功能。

若组态不受支持，则通过 Event Notifier 链接本地 WinCC OPC UA 服务器的 WinCC 消息时会导致“报警记录”连续循环。

要求

- 已建立与 OPC UA 服务器的连接。
- 已激活过程值块以供使用。

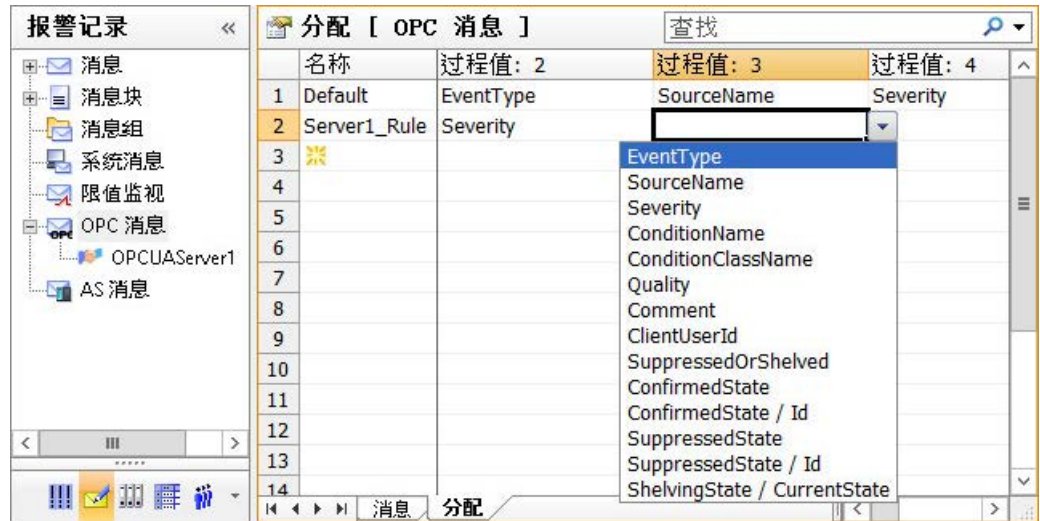
操作步骤

1. 在“报警记录”(Alarm Logging) 编辑器的“OPC 消息”(OPC messages) 下选择组态的连接。
2. 从 OPC UA 连接的快捷菜单中选择“浏览 OPC 服务器”(Browse OPC server) 条目。
OPC UA 服务器的可用数据已加载。“受监视的对象”(Monitored objects) 视图打开。
在导航区域中，Event Notifier 显示在连接名称下。
3. 在导航区域中选择 Event Notifier。
4. 在数据区域中输入一个或多个 WinCC 消息编号。
避免链接最高层级“Server”，否则所有附属 Event Notifier 也会触发链接的消息。大量触发消息可能会降低性能。

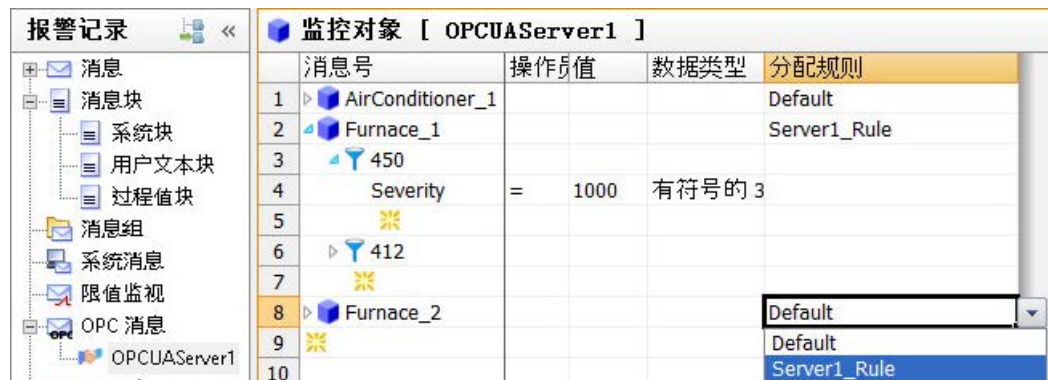
消息号		操作员	值	数据类型
1	450			
2	Severity	=	1000	有符号的 32
3				
4	412			
5	EventType	=	AlarmConditionType	
6	EventType	=	DeviceFailureEventType	
7				
8	413			
9	SourceName	contains	ventilator	文本变量 16
10	ConditionName	=	TemperatureHigh	文本变量 16
11				
12	ConditionName			
13	EventType			
14	Severity			
15	SourceName			

5. 要定义过滤器，请单击消息编号前面的箭头。
在显示的行中，选择过滤条件、运算符和值。
会自动添加过滤条件对应的数据类型且无法更改。
6. 要在“报警记录”(Alarm Logging) 视图中编辑消息，请单击以下符号：
7. 要显示分配规则，请单击“OPC 消息”(OPC messages) 并选择“分配”(Assignments) 选项卡。

8. 在“默认”(Default) 规则的“名称”(Name) 字段中输入新的规则名称。



9. 从过程值字段的下拉列表中选择相应 Event Notifier 的所需属性。
属性与相应的过程值块链接。
10. 单击“OPC 消息”(OPC messages) 下的连接名称。
将显示 OPC UA 连接的消息和过滤器。
11. 为 Event Notifier 选择相应的分配规则。
默认链接“默认”(Default) 规则。



12. 要显示 WinCC 消息，请单击“OPC 消息”(OPC messages)。
“消息”(Messages) 选项卡中会显示消息及其属性。
13. 组态 WinCC 消息的属性，例如消息类别、消息类型和归档。
要在用户文本块中访问过程值块的内容，请使用“@1%”格式。
有关过程值块的更多信息，请参见 WinCC 信息系统中的“使用 WinCC > 设置消息系统 > 组态消息系统 > 使用消息”(Working with WinCC > Setting up a message system > Configuring the message system > Working with messages):
- “如何指定消息的文本”
 - “如何在用户文本块中插入过程值”

参见

“受监视的对象”(Monitored objects) 视图 (页 294)

OPC UA 报警过滤器 (页 298)

6.5.4 OPC UA WinCC 通道中的 OPC UA 数组

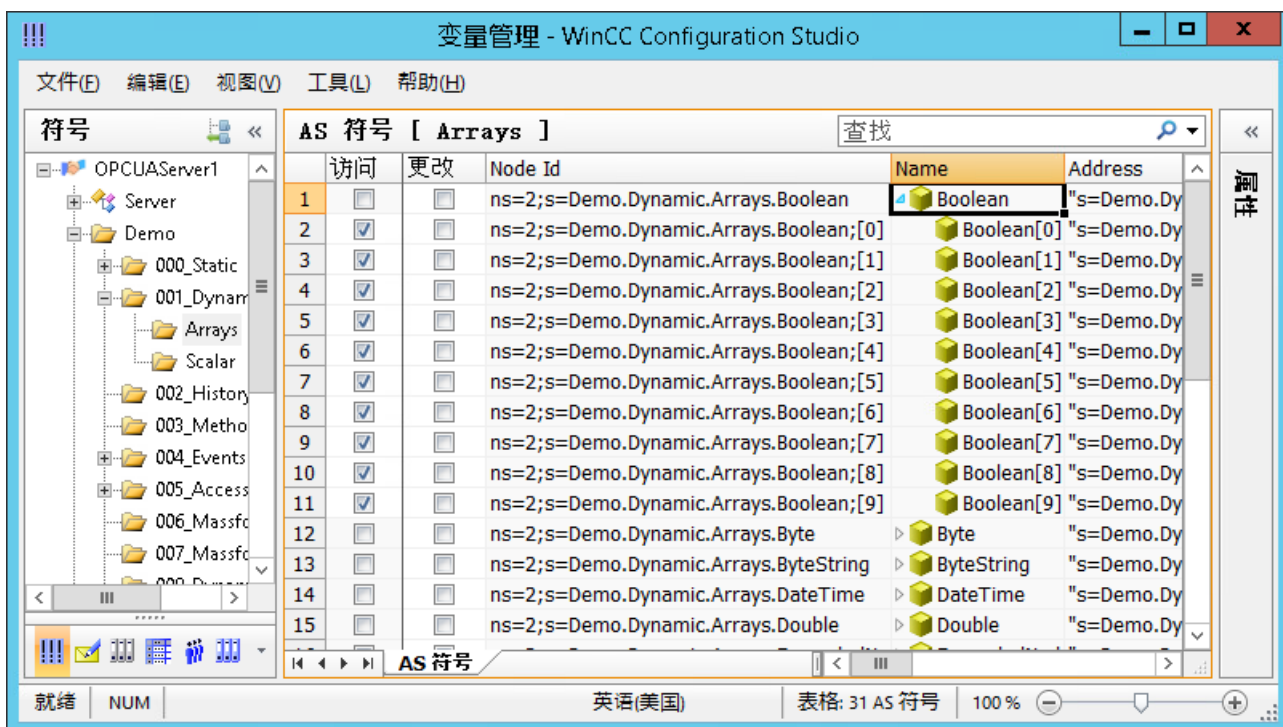
6.5.4.1 WinCC 中的数组

WinCC 支持 OPC UA 数组的组态。在这种情况下，一个 WinCC 变量只能对应一个数组元素。这意味着不可能将一个完整的 OPC UA 数组值只映射到一个 WinCC 变量。

WinCC Configuration Studio 中数组的表示

在 WinCC Configuration Studio 中，数组在“AS 符号”(AS symbols) 工作区中表示为扩展数据条目。数组类型和大小显示在 OPC UA DataType 列中。

通过单击“名称”(Name) 列中的相应箭头符号，可以访问各个数组元素。



为了将数组元素映射为 WinCC 变量，激活“访问”(Access) 列中的复选框。

数据类型

WinCC 支持以下 OPC UA 数组数据类型（任意大小）：

- 布尔型
- SByte
- 字节
- Int16
- UInt16
- Int32
- UInt32
- 单精度浮点数
- 双精度型
- 字符串
- DateTime
- ByteString

说明

如果服务器输出不支持的数组数据类型，则 WinCC Configuration Studio 中不能组态变量。

在未指定 IndexRange 的情况下写入数组条目

在读取和写入 OPC UA 数组时，OPC UA 客户端可以通过在所使用的 OPC-UA 规范中指定的 IndexRange 函数访问整个数组或数组的一部分。

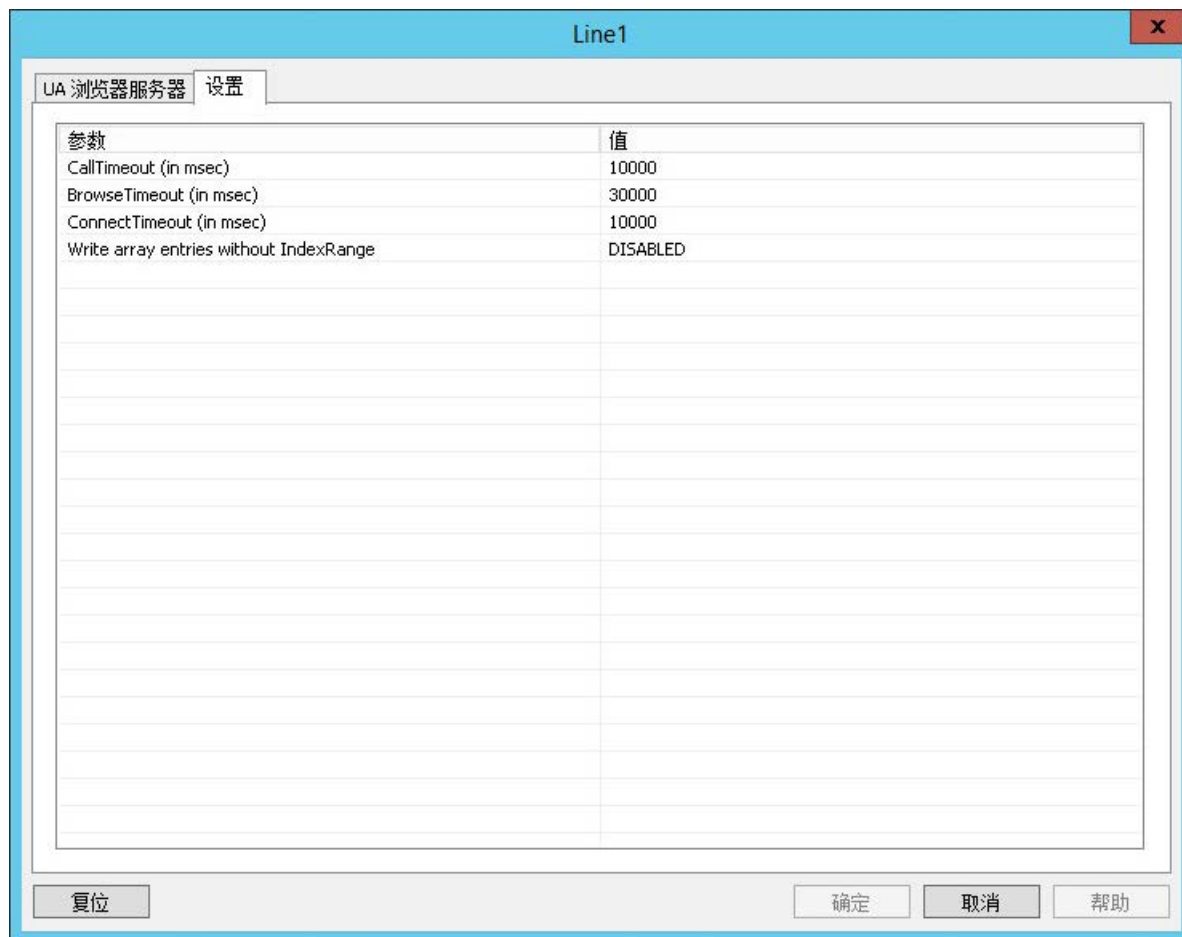
例如，客户端可以通过将“3, 10”指定为索引范围来只读取或写入数组元素 3 到 10。或者通过将“5”指定为 IndexRange 来只读取和写入数组元素 5。

符合 OPC UA 规范的 OPC UA 服务器在读取数组时应支持 IndexRange，而不支持使用 IndexRange 写入数组。

WinCC 将始终尝试使用 IndexRange 读取或写入数组元素。如果无法使用 IndexRange 读取数组元素，则 WinCC 会尝试读取完整的数组并提取在 WinCC 中组态的数组元素。如果无法使用 IndexRange 写入数组元素，则 WinCC 将使用回退策略“在未指定 IndexRange 的情况下写入数组条目”。由于这可能会导致数据不一致，因此在默认情况下它将处于取消激活状态。

6.5 OPC UA WinCC 通道

在“连接参数”(Connection parameters)对话框的“设置”(Settings)选项卡中，“在未指定 IndexRange 的情况下写入数组条目”功能已激活。



注意
数据不一致
激活此功能可能会导致数据不一致！

限制

数据一致性

在运行系统中，不能保证在 WinCC 中组态的 OPC UA 数组值的数据一致性。

在以下情况下，可能会出现数组数据不一致情况：

- 在不同的读取周期中读取多个数组元素时。
- 读取和更新完整数组时（请参见“在未指定 IndexRange 的情况下写入数组条目”部分）。

性能

如果 OPC 服务器不支持 IndexRange，则读取数组中的多个元素时可能会出现性能问题，因为 WinCC 必须读取完整的数组值并提取组态的数组元素。

动态数组

OPC-UA 数组的长度可以是动态的，且可以在运行系统中修改。例如，如果数组值比预期的要短，则 WinCC 变量值可能会质量差。

兼容性

某些 OPC UA 服务器与 OPC UA 规范不完全兼容，因此变量管理可能无法显示数组元素。在这种情况下，必须手动创建数组变量并修改数组地址。

因此，需要服务器特定的知识才能为此类服务器组态数组。

有关手动组态的更多信息，请参见“如何手动创建和组态 OPC UA 数组 (页 307)”部分。

6.5.4.2 如何手动创建和组态 OPC UA 数组

如果 OPC UA 服务器不按照规范显示数组，则 WinCC 可能无法显示 OPC UA 数组的数组元素。在这种情况下，必须在 WinCC 变量管理中手动创建和组态数组元素。为此，需要服务器特定的知识。

要求

- 已建立与 OPC UA 服务器的连接。
- 已打开变量管理。

步骤

1. 创建新变量时，可在“变量”(Tags)工作区中，通过快捷菜单复制和插入现有变量。

说明

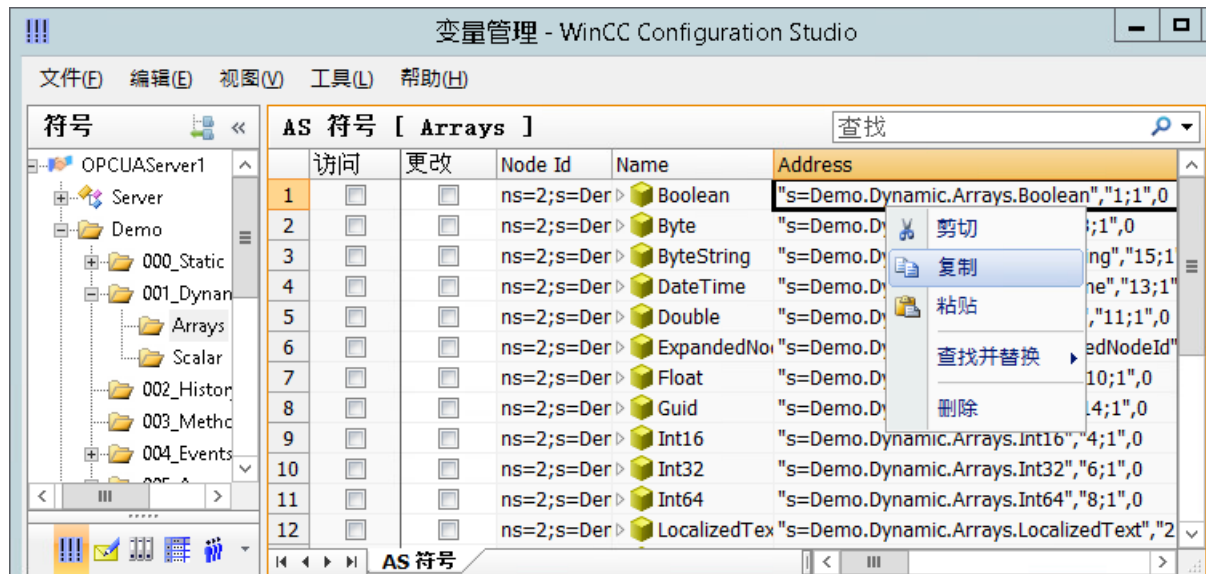
复制和插入变量

为了复制变量及所有属性，必须标记完整的行，而不仅仅是变量名称。为此，单击前面的数字。

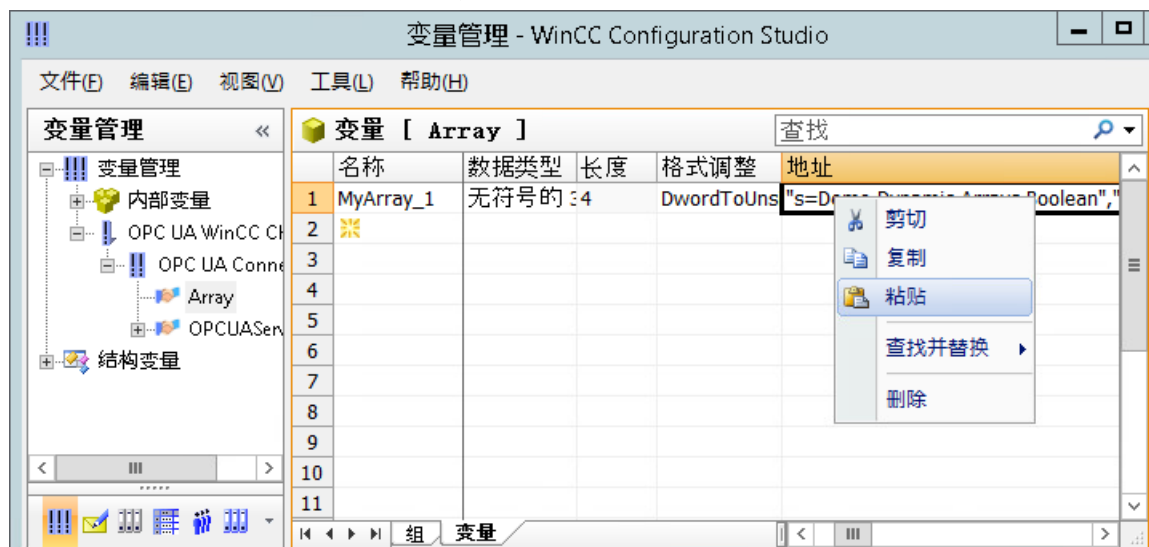
2. 重命名新建变量（如果适用）。


6.5 OPC UA WinCC 通道

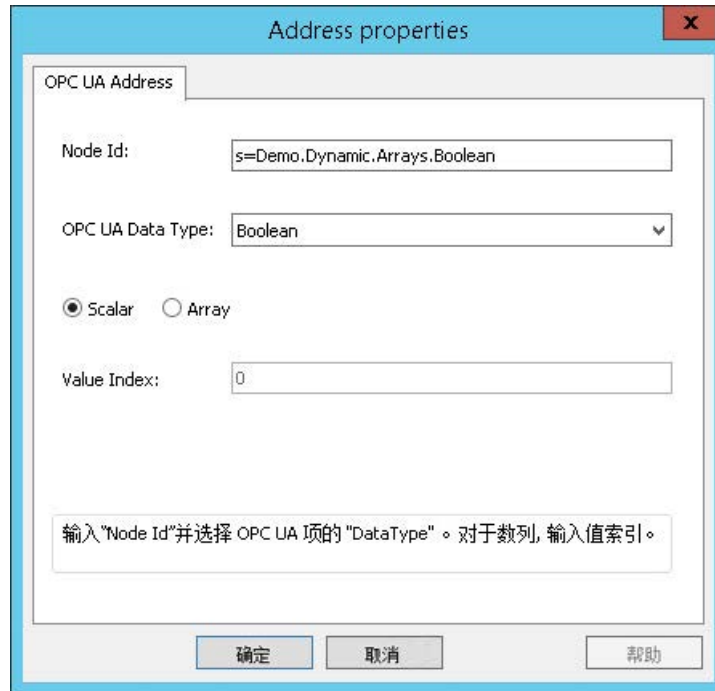
3. 将“AS 符号”(AS symbols) 工作区中相应数组值的地址复制到剪贴板中。



4. 将“AS 符号”(AS symbols) 工作区中相应数组值的地址插入到复制变量的“地址”(Address) 列中。



5. 通过使用  按钮在变量属性中打开“地址属性”(Address properties) 对话框。



6. 执行所需设置，然后单击“确定”(OK) 进行确认。

6.5.5 错误处理

6.5.5.1 OPC 通讯受到干扰时的错误处理

简介

通讯测试的过程与 WinCC 的使用方式无关。

有关通道诊断的更多信息，请参见“通信 > 通信诊断”(Communication > Communication Diagnostics)。

WinCC 用作 OPC UA 服务器

在 WinCC OPC UA 客户端上使用通道诊断来检查是否可以与 WinCC OPC UA 服务器建立连接。

WinCC 用作 OPC UA 客户端

在 WinCC OPC UA 客户端上使用通道诊断来检查是否可以与 WinCC OPC UA 服务器建立连接。

6.6 PROFIBUS DP

6.6.1 WinCC“PROFIBUS DP”通道

内容

“PROFIBUS DP”通道用于 WinCC 站（作为 PROFIBUS DP 主站）和相应外围配件（例如 ET200）之间的通信。

通信使用 PROFIBUS DP 协议。

本章描述了下列主题：

- 如何使用“PROFIBUS DP”通道组态数据传送
- 如何组态连接和变量

PROFIBUS DP 当前版本的更改

与随附的文档相比，当前版本的 PROFIBUS DP 中进行了以下更改：

- PROFIBUS DP 主站用作应用程序 OPC 服务器。

6.6.2 WinCC 驱动程序 Profibus DP 的特性

特性

WinCC 驱动程序 Profibus DP 具有以下特性：

- 带有通信处理器 (= CP) 的 WinCC PC 是 Profibus 上的 DP 主站。
- 所有 DP 标准从站均可寻址。
- 最多可以对四个 CP 卡进行初始化，并可以为每个 CP 模块组态最多 123 个 DP 从站。
较新版本的限制值可能会发生更改，因此应在调试前进行检查。

如果驱动程序允许，可以使用其它协议与其它 Profibus 总线伙伴进行通信。

说明

只有一个 DP 主站

Profibus 总线上不能连接寻址相同从站的其它 DP 主站。

6.6.3 集成“Profibus DP”驱动程序

标准

基于 Profibus 分布式 I/O (DP) 标准:

- DIN 19245-3, 或根据
- pr EN 50170

要求

硬件:

- 要使用 WinCC 驱动程序 Profibus DP, 需要通信处理器 CP 5613 (A3) 或 CP 5612 来连接 Profibus
所有 DP 标准从站均可使用它来寻址。
- 使用的通信处理器的数量取决于 PC 中仍然空闲的中断。

软件:

- 要安装和组态通信处理器, 需要驱动程序和组态软件。
可在 SIMATIC NET CD 上找到这些程序和软件。

操作步骤

1. 可在变量管理的导航区域中选择“Profibus DP”通信驱动程序。
通道单元已创建。
2. 在通道单元快捷菜单中, 选择“系统参数”(System parameters)。
将打开组态对话框。
3. 指定 CP 板号和监视时间。
4. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
5. 输入连接的名称。
6. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。
将打开组态对话框。
7. 选择从站地址, 然后单击“确定”(OK) 进行确认。

6.6.4 组态“PROFIBUS DP”驱动程序

设置系统参数

CP 板号

PC 中 CP 卡的编号（来自组态工具）。

值范围：

- 1 至 4
- 0 = 未安装

看门狗时间

在通信卡上输入 WinCC 的监视时间系数。对于每种情况，监视时间是 0.4 秒的倍数。

该功能仅对可以输出数据的从站有效。

- 输入 0：
监视已禁用。
- 输入 > 0：
如果该时间过后没有进一步的写访问，则输出设置为 0。
这必须通过适当的 WinCC 组态来保证这一点。

设置连接参数

从站地址

要读取或写入的从站地址。

值范围：

- 1 至 127

设置变量地址

要组态变量地址，请单击“属性 - 变量”(Properties - Tag) 窗口中的空白“地址”(Address) 字段。

使用以下符号打开组态对话框：

说明

连接的性能

如果电源和吞吐量受到影响，请注意以下事项：

互连变量的更新时间会影响连接，因为只能访问 Profibus 的整个 DP 设备。

过程变量的属性

域	含义
输入	从站的输入范围
输出	从站的输出范围
长度（位）	以位为单位显示变量大小 值取决于之前选择的数据类型。 例外：原始数据变量
字节偏移量	存储变量内容之前的字节数 值范围：0 至长度 -1
位偏移量	只在“二进制变量”数据类型时激活 在输入二进制变量的上述指定字节中的位数 值范围：0 至 7 如果未超过缓冲区的长度，则可能为更大的值。
更改字节排列	已取消激活：低位字节（默认设置） 已激活：高位字节

原始数据变量的属性

域	含义
输入	从站的输入范围
输出	从站的输出范围
长度（位）	未激活
字节偏移量	存储变量内容之前的字节数 值范围：0 至长度 -1
位偏移量	未激活
长度（字节）	在此字段中输入所需的块长度。 此字段的长度单位为字节。
发送/接收块	经 WinCC 的请求之后，所定义的数据块将进行发送或接收。

6.7 S5 Ethernet Layer 4

6.7.1 WinCC 通道“SIMATIC S5 Ethernet Layer 4”

简介

例如，该通信驱动程序可用于通过 ISO 传输协议或 TCP/IP 协议，来连接自动化系统 SIMATIC S5-115U/H、SIMATIC S5-135U 和 SIMATIC S5-155U/H。

根据所用的通信协议，将实施下列通信伙伴：

通信协议	WinCC 侧	SIMATIC S5 端
ISO 传输协议	CP1612 A2 (3Com 兼容) CP1613 A2 CP1623	CP1430 TF
TCP/IP (符合 RFC1006)	CP1612 A2 (3Com 兼容) CP1613 A2 CP1623	CP1430 TCP

使用此通道时，不需要本地数据库。

通道单元

通信驱动程序具有两个通道单元“CP1413-x”，其中最多可运行两个通道单元 CP1612 A2、CP1613 A2 或 CP1623。通道单元的功能完全相同。只是两个 CP 的逻辑设备名称不同。CP1623 与 CP1613 A2 相同，但通过 PCI Express 运行。

可以使用第三个通道单元“TCP/IP”通过 TCP/IP 协议与 CP1612 A2、CP1613 A2 或 CP1623 建立通信。

逻辑设备名称可在通道单元的系统参数中进行更改。在此，也可以为所用协议设置参数。

存在下列应用能力：

- 通道单元“S5-Transport (CP 1413-1)”，用于 SIMATIC 工业以太网的通信模块 (CP 1612 A2 / 1613 A2 /1623)。
- 通道单元“S5-Transport (CP 1413-2)”，用于 SIMATIC 工业以太网的通信模块 (CP 1612 A2 / 1613 A2 /1623)。
- 通道单元“S5 传输 (TCP/IP)”，用于 SIMATIC 工业以太网的通信模块 (CP 1612 A2 /1613 A2 / 1623)。

6.7 S5 Ethernet Layer 4

6.7.2 变量的数据类型

简介

为逻辑连接定义所需变量。从 WinCC 的角度而言，可以访问以下数据类型：

- 二进制变量
- 无符号 8 位数
- 有符号 8 位数
- 无符号 16 位数
- 有符号 16 位数
- 无符号 32 位数
- 有符号 32 位数
- 浮点数 32 位 IEEE 754
- 文本变量，8 位字符集
- 原始数据类型

6.7.3 组态通道

6.7.3.1 组态通道“SIMATIC S5 Ethernet Layer 4”

简介

组态通道“SIMATIC S5 Ethernet Layer 4”需要以下步骤。

1. 组态连接
2. 组态变量
3. 系统参数组态

6.7.3.2 如何组态连接

简介

连接参数对于使用的所有协议几乎完全相同。下例描述了使用 ISO 传输协议和通道单元“CP1413-x”的通信。

实施 TCP/IP 协议时，输入 AS 的 IP 地址而非以太网地址。IP 地址由四个以圆点分隔的数值组成。数值必须位于 0-255 的范围内。

对于逻辑连接，WinCC 在传输层中建立一个用于读取（“读功能”区域）的连接和一个用于写入（“写功能”区域）的连接。这两个功能的地址参数在对话框中定义。仅当这两个连接都已建立时，才会将逻辑连接指示为“已建立”。

读功能的分配

WinCC 端	SIMATIC S5 端
FETCH-Active (请求“READ-Active”)	READ-Passive (请求“READ-Passive”)
FETCH-Passive (请求“WRITE-Passive”)	WRITE-Active (请求“WRITE-Active”)

说明

如果 AS 的数据为发送主动，即在连接参数中读功能被设置为“读取被动”，则无法将二进制或字节变量写入 AS 的数据区域。

如果已将至少一个报文从 AS 发送到 WinCC，则只能为“读取被动”连接分配“正常”状态。

写功能的分配

WinCC 端	SIMATIC S5 端
请求“WRITE Active”	请求“WRITE Passive”

步骤

1. 从连接的快捷菜单中选择“连接参数”(Connection parameters) 条目。
“连接属性”(Connection properties) 对话框随即打开。



2. 在“以太网地址 AG”域中，输入工业以太网总线上 SIMATIC S5 的站地址。实施 TCP/IP 协议时，在“IP 地址 AG”域中输入 IP 地址。
3. 在 WinCC 系统中为读功能定义参数。这些参数独立于 SIMATIC S5 中所使用的请求。
4. 然后，在“自身 TSAP”分配域中输入组态 CP 1430 TF 时在“远程参数”中被组态为“TSAP”的值。
5. 现在，在“远程 TSAP”分配域中输入组态 CP 1430 TF 时在“本地参数”中被组态为“TSAP”的值。
6. 为写功能相应地定义“自身 TSAP”和“远程 TSAP”参数。

说明

在“TSAP”条目中，不得使用任何空格。

6.7.3.3 组态变量

组态变量

简介

对于 WinCC 与 AS 之间通过通道“SIMATIC S5 Ethernet Layer 4”进行的连接，可在 WinCC 中创建不同数据类型的变量。下面将说明如何组态这些数据类型的变量。

- 变量的地址
- 组态按位访问的变量
- 组态按字节访问的变量
- 组态按字访问的变量
- 组态原始数据变量

变量的地址

简介

变量地址根据 SIMATIC S5 的地址结构进行输入。

视变量类型而定，对 AS 中存储器区域的访问有按位、按字节或按字这几种类型。为此，每次写请求时均从 AS 读取已赋址的存储器区域，并修改相应的位和/或字节。此后，数据写回到 AS 的存储器。

说明

AS 在读取数据区中执行的更改会在写回到该数据区时被覆盖。


如果 AS 的数据为发送主动，即在连接参数中读功能被设置为“读取被动”，则无法将二进制或字节变量写入 AS 的数据区。

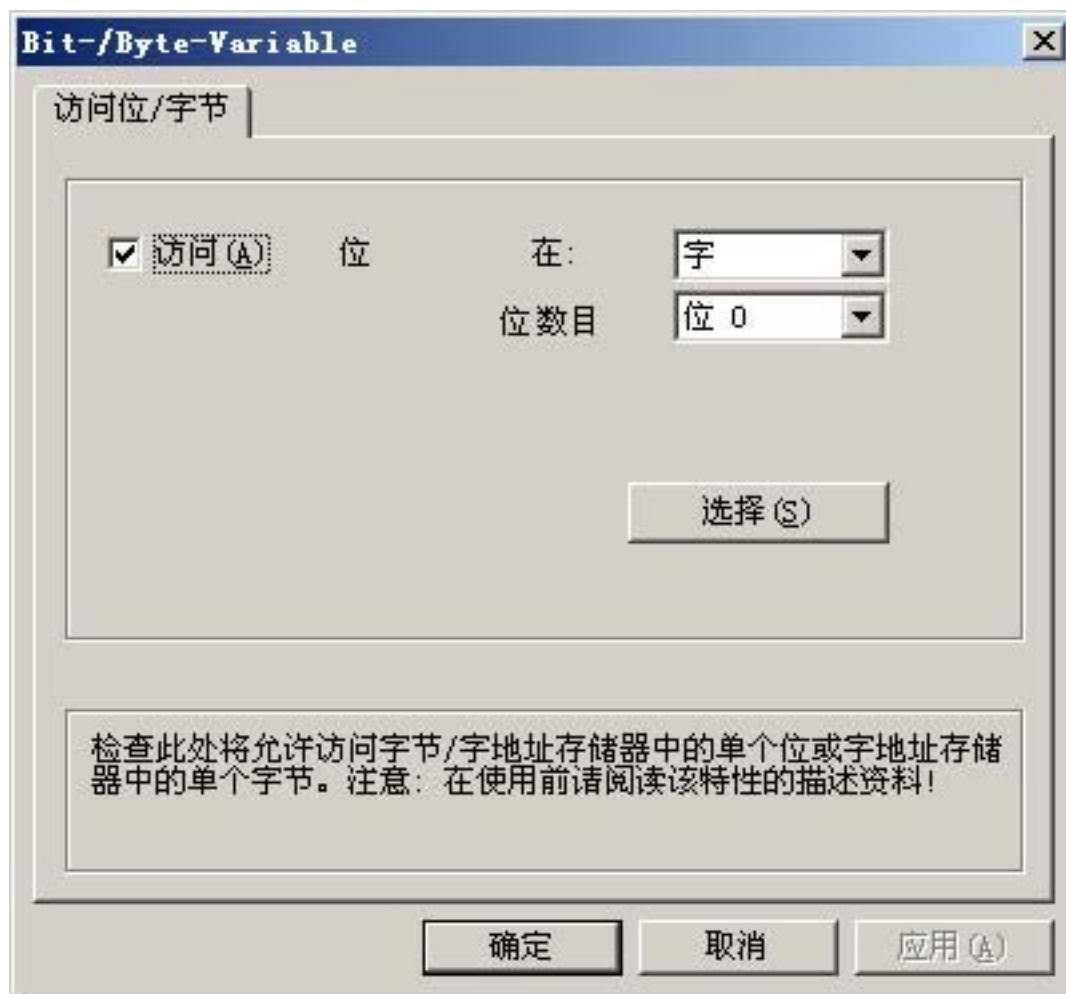
组态变量的地址而不依赖于变量类型：

- 对于“二进制”或“8 位数”类型的变量，首先打开“位/字节变量”对话框，在此对话框中定义按位或按字节访问 AS 的存储器区域。
然后，在“地址属性”对话框中定义变量在 AS 存储器中的地址。
- 对于按字访问的变量，在“地址属性”对话框中定义变量在 AS 存储器中的地址。
由于对 AS 存储器的访问是按字进行的，所以不会打开“位/字节变量”对话框。

如何组态按位访问的变量

步骤

1. 选择连接，然后在快捷菜单中打开“位/字节变量”(Bit/Byte tag) 对话框。为此，请单击“地址”(Address) 字段，然后单击  按钮。
2. 单击“选择”(Select) 按钮。“位/字节变量”(Bit/Byte tag) 对话框随即打开。



3. 使用复选框定义是否启用对存储区中某些位的读写访问。
4. 在选择域中选择 AS 存储器的寻址方法，例如“字”或“字节”。
5. 在选择域中选择要更改的位号。
6. 使用“选择”按钮打开“地址属性”对话框，以定义 AS 中的变量地址。

说明


在 S5 中，可以按字节寻址标志位、输入和输出，按字寻址数据块（DB、DX）。

激活“访问位”(Access a bit) 复选框会影响“地址属性”(Address properties) 对话框中字段的显示。

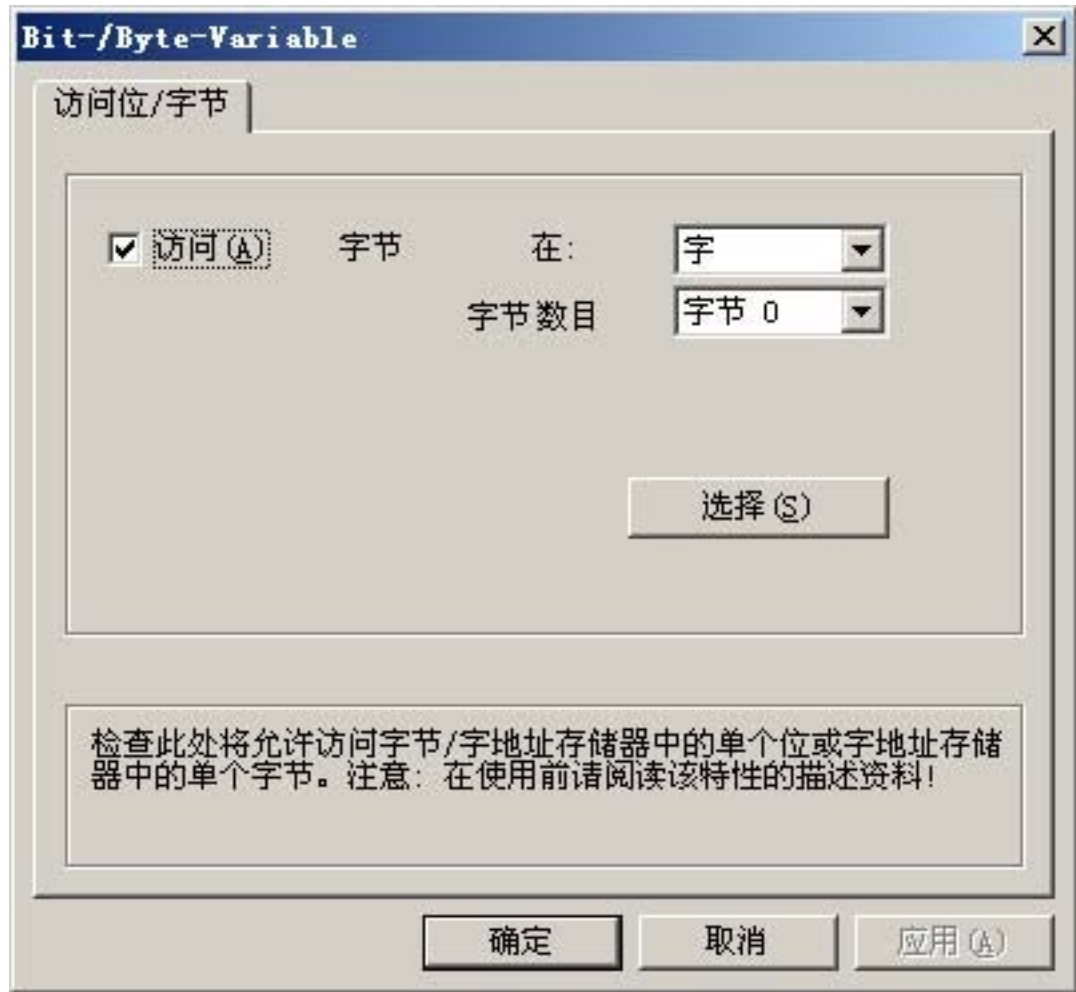
对于按字访问的变量，不会打开所述的“位/字节变量”(Bit/Byte tag) 对话框，因为变量地址是按字存储，从而导致对 PLC 存储器的访问也是按字进行。

如何组态按字节访问的变量

步骤

1. 选择变量，然后在“数据类型”(Data type) 字段中选择数据类型“无符号 8 位数”(Unsigned 8-bit value) 或“有符号 8 位数”(Signed 8-bit value)。
2. 选择连接，然后在快捷菜单中打开“位/字节变量”(Bit/Byte tag) 对话框。为此，请单击“地址”(Address) 字段，然后单击  按钮。

- 单击“选择”(Select)按钮。“位/字节变量”(Bit/Byte tag)对话框随即打开。



- 使用复选框定义是否启用对存储区中某些字节的读写访问。
- 在选择域中，只有“字”显示为 AS 存储器的寻址类型。
- 在选择域中，选择要更改的字节号。
- 使用“选择”按钮打开“地址属性”对话框，以定义 AS 中的变量地址。

说明

在 S5 中，可以按字节寻址标志位、输入和输出，按字寻址数据块 (DB、DX)。

选择“访问字节”(Access a byte) 复选框会影响“地址属性”(Address properties) 对话框中字段的显示。

对于按字访问的变量，不会打开所述的“位/字节变量”(Bit/Byte tag) 对话框，因为变量地址是按字存储，从而导致对 PLC 存储器的访问也是按字进行。


如何组态按字访问的变量

简介

使用此处介绍的对话框定义 AS 中的变量地址。

- 对于“二进制”或“8 位数”类型的变量，首先打开“位/字节变量”对话框，在此对话框中定义按位或按字节访问 AS 的存储器区域。
- 对于按字访问的变量，不会打开“位/字节变量”对话框，因为变量地址是按字存储，从而导致对 AS 存储器的访问也是按字进行。

步骤

1. 选择变量，然后在“数据类型”(Data type) 字段中选择变量所需的数据类型（例如：有符号 16 位数）。
2. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



6.7 S5 Ethernet Layer 4

3. 在“地址”选项卡的“数据区”域中，选择变量是位于数据块、标记区域、输入区域还是输出区域。
4. 如果变量位于数据块中，则还会显示“DB 号”域。在此，输入数据块的编号。
5. 在“寻址”域中输入寻址类型。通常，可以使用默认定义。
6. 在相应域（例如“DW”）中，输入地址。

说明

对于“二进制”或“8 位数”类型的变量，此对话框中显示的域取决于在“位/字节变量”对话框中对“访问位/字节”做出的选择。

如果要写入按字访问数据区的变量，则起始地址必须在左字节中，而变量长度必须为偶数。

如何组态原始数据变量


简介

以下描述了如何定义原始数据变量的地址。

说明

如果要写入按字访问数据区的变量，则起始地址必须在左字节中，而变量长度必须为偶数。

步骤

1. 选择变量，然后在“数据类型”(Data type) 字段中选择条目“原始数据类型”(Raw data type)。
2. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



3. 在“数据区”域中，选择变量是位于数据块、标记、输入区域还是输出区域。
4. 如果变量位于数据块中，则还会显示“DB 号”域。在此，输入数据块的编号。
5. 在“寻址”域中输入寻址类型。通常，可以使用默认定义。
6. 在下面的域中输入数据地址。域标签取决于“寻址”域中的条目，例如，对于“字”寻址则为“DBW”。
7. 在“长度”域中输入原始数据块的长度（以字节为单位）。
8. 在“原始数据类型”区域中，定义与原始数据变量相关的类型。

6.7.3.4 系统参数

通道单元的系统参数

简介

如果需要与 WinCC 标准设置不同的组态，则可使用通道单元的“系统参数”对话框进行所需的全部更改。

系统参数对于使用的所有协议几乎完全相同。实施 TCP/IP 协议时，只有安装期间所给定的设备名称不同。

以下各项均可更改：

- 设备名称
- 传输参数

说明

系统参数适用于 AS 中的所有 CP。

设备名称

WinCC 和自动化系统之间的通信通过逻辑设备名称进行。这些名称是在安装通信模块时分配的，并且是单元特定的。设备名称代表逻辑设备名称。实施 ISO 传输协议时，将名称“/CP_H1_1:/SCP”指定为逻辑设备名称的默认定义；实施 TCP/IP 协议时，将名称“/TCP_IP:/SCP”指定为逻辑设备名称的默认定义。

说明

使用 TCP/IP 协议时，必须检查 WinCC 中的设备名称是否与“设置 PG/PC 接口”对话框中的“应用程序访问点”相匹配。还必须在“设置 PG/PC 接口”中更改设备名称。

传输参数

在传输参数中进行通道单元的特定设置，例如 PDU 大小、尝试建立次数等。

如何更改设备名称

简介

使用系统参数（如逻辑设备名称或传输参数）设置通道单元的参数。

系统参数对于使用的所有协议几乎完全相同。

下例描述了使用 ISO 传输协议和通道单元“CP1413-x”的通讯。

实施 TCP/IP 协议时，只有安装期间所给定的设备名称不同。

步骤

1. 选择通道单元，然后通过上下文菜单打开“系统参数”对话框。
2. 选择“设备名称”选项卡。



3. 现在，可使用鼠标来选择以粗体显示的设备名称，并在该设备名称的名称域中单击鼠标对其进行更改。

说明

设备名称在安装硬件驱动程序时定义。只有在安装硬件驱动程序时定义了其它名称（不推荐），才必须也更改此处的设备名称。

如何更改传输参数

操作步骤

1. 选择通道单元，然后通过快捷菜单打开“系统参数”(System parameters)对话框。
2. 选择“传输参数”(Transport Parameters)选项卡。



3. 将“PDU 大小”的值设置为在通讯模块 CP 1430 上组态的值。
4. 在“建立尝试”(Setup Attempts) 字段中，定义尝试建立连接的频率。

5. 在“重复发送持续时间”(Duration of Send Repetitions) 区域中选择“无限”(Infinite)。
6. 例如，在“确认时间”(Acknowledgment Time) 字段中输入值 30，这样，如果通信伙伴在此时间内未作出响应，则会在 30 秒内通知您变量状态（例如，AS 处于“停止”状态）。

6.7.4 附录

6.7.4.1 附录

简介

附录中提供有关通道“SIMATIC S5 Ethernet Layer 4”的附加信息。

6.7.4.2 内部错误代码和常量

内部错误代码和常量

简介

下表包含了最重要的错误代码和常量。这些信息仅供“内部人员”使用。因此，我们就不详细介绍这些代码和常量的含义了。

- 连接受干扰时的错误代码
- iNA960 消息
- SCI 消息

连接受干扰时的错误代码

简介

本部分列出了最重要的错误代码。如果出现该表中未列出的错误代码所对应的错误，请拨打 WinCC 热线。

Fehler_0002-INVALID_RQ

请求块出现故障。

Fehler_0004-NO_RESOURCES

CP 中无可用资源。

Fehler_0006-UNKNOWN_REFERENCE

所定义的 OPEN 参考不正确。

Fehler_0008-BUFFER_TOO_SHORT

用户缓冲区过短。

Fehler_0010-BUFFER_TOO_LONG

用户缓冲区过长。

Fehler_0012-ILLEGAL_REQ

所定义的“negot_options”不正确。

Fehler_0014-REM_ABORT

连接被远程站中止。

Fehler_0016-LOC_TIMEOUT

超时。

Fehler_0018-UNKNOWN_CONN_CLASS

连接类别未知。

Fehler_0020-DUP_REQ

连接已建立。

Fehler_0022-CONN_REJECT

连接请求被远程拒绝。

Fehler_0024-NEGOT_FAILED

“negot-option”出现故障，连接中止。

Fehler_0026-ILLEGAL_ADDRESS

传输地址有误。

Fehler_0028-NETWORK_ERROR

总线或 CP 受到干扰。

Fehler_0030-PROTOCOL_ERR

协议错误。

Fehler_0032-ILLEGAL_RB_LENGTH

请求块长度不正确。

Fehler_0784-E_NO_HW

未找到通讯硬件。

- 通讯模块存在缺陷。
- 通讯模块安装得不正确。
- 所定义的端口地址错误。

Fehler_0786-E_CNF

驱动程序组态不正确，或者注册表中的参数无效。

Fehler_0787-E_BAUDRATE

波特率不正确或所定义的中断向量不正确。

Fehler_0788-E_HSA

所定义的 HSA（最高站地址）不正确。

Fehler_0789-E_TS

所定义的本地参与者号 (TS_ADR) 已分配。

Fehler_0791-E_INT_NOT_PROV

所定义的中断向量 (IRQ) 在通讯模块上不可用。

Fehler_0792-E_INT_BUSY

所定义的中断向量 (IRQ) 在通讯模块上已被占用。

Fehler_0800-E_NO_FILE

无法装载所选的通讯驱动程序；文件未找到。

- 通讯驱动程序安装得不正确。

Fehler_0897-E_LOGDEV

注册表中未定义逻辑设备。

- 通讯驱动程序安装得不正确。
- 注册表项已损坏或删除。
- 使用“设置 PG/PC 接口”程序检查逻辑设备名称的设置。
- 检查“系统参数 - 设备”掩码中逻辑设备名称的设置。

Fehler_0898-E_L2DRIVER

注册表中缺少条目“L2DRIVER”。

- 模块错误或模块安装得不正确。

Fehler_0900-E_L4DRIVER

注册表中缺少条目“L4DRIVER”。

- 模块错误或模块安装得不正确。

Fehler_30000-EC_WATCHDOG

监视狗错误。

Fehler_30001-EC_PDUERROR

PDU 不是所需的。

Fehler_30005-EC_ONLERROR

装载 S7-Online-DLL 时出现故障。

iNA960 消息**常规 iNA960 消息**

OK_RESP	1	0x01	请求已正确执行
OK_EOM_RESP	3	0x03	数据块已正确接收
OK_DECIDE_REQ_RESP	5	0x05	请求已正确执行
OK_CLOSED_RESP	7	0x07	连接已由本地用户中止

iNA960 错误消息

INVALID_REQ	2	0x02	请求块出现故障
NO_RESOURCES	4	0x04	CP 中无可用资源
UNKNOWN_REFERENCE	6	0x06	所定义的 OPEN 参考不正确
BUFFER_TOO_SHORT	8	0x08	用户缓冲区过短
BUFFER_TOO_LONG	10	0x0A	用户缓冲区过长
ILLEGAL_REQ	12	0x0C	所定义的“negot_options”不正确
REM_ABORT	14	0x0E	连接被远程站中止
LOC_TIMEOUT	16	0x10	超时
UNKNOWN_CONN_CLASSES	18	0x12	连接类别未知
DUP_REQ	20	0x14	连接已建立
CONN_REJECT	22	0x16	连接请求被远程拒绝
NEGOT_FAILED	24	0x18	negot-option 出现故障，连接中止
ILLEGAL_ADDRESS	26	0x1A	传输地址有误

NETWORK_ERROR	28	0x1C	总线或 CP 受到干扰
PROTOCOL_ERR	30	0x1E	协议错误
ILLEGAL_RB_LENGTH	32	0x20	请求块长度不正确

SCI 消息

请参见《SINEC 通讯接口 SCI》手册中的说明 (A/5-15)。

SCI 消息

SCP_OK	0	0x00	无错
SCP_INCONS	201	0xC9	较小的设备编号不是 00
SCP_RESOURCE	202	0xCA	DPRAM 请求无效
SCP_CONFIG	203	0xCB	组态错误 (NUM_PROCS)
SCP_NOCONFIG	204	0xCC	SCP 驱动程序未组态
SCP_PARAM	206	0xCE	模式不正确
SCP_DEVOPEN	207	0xCF	已执行打开操作
SCP_BOARD	208	0xD0	未插入/不识别电路板
SCP_SOFTWARE	209	0xD1	IRQ 错误或软件未找到
SCP_MEM	210	0xD2	DPRAM 内存小
SCP_MODE	211	0xD3	下载过程尚未结束
SCP_LOADER	212	0xD4	装载程序无响应
SCP_SIGNAL	213	0xD5	过程被异步启动
SCP_NOMESS	215	0xD7	该过程中没有消息到达
SCP_USERMEM	216	0xD8	length_of_buffer 过小
SCP_WINDOW	217	0xD9	SEND 的调用次数过多
SCP_TIMEOUT	219	0xDB	SCP 超时
SCP_ATTACH	220	0xDC	未执行重置/通道仍处于活动状态
SCP_ILLEGAL_REQUEST	221	0xDD	非法请求
SCP_ERECOVERF	223	0xDF	未通过 scp_receive 获取缓冲区
SCP_ECLOSED	224	0xE0	已为连接分配了所有缓冲区
EUSERMAX	225	0xE1	
SCP_EINTR	226	0xE2	

SCP_BOARD_OPEN	231	0xE7	
SCP_NO_WIN_SERV	233	0xE9	
EPROTECT	234	0xEA	未找到许可证

SCI 消息

SCP_DB_FILE_DOES_NOT_EXIST	240	0xF0	
SCP_DB_FILE_CLOSE_NOT_OK	241	0xF1	
SCP_SEND_NOT_SUCCESSFUL	242	0xF2	
SCP_RECEIVE_NOT_SUCCESSFUL	243	0xF3	
SCP_NO_DEVICE_AVAILABLE	244	0xF4	
SCP_ILLEGAL_SUBSYSTEM	245	0xF5	
SCP_ILLEGAL_OPCODE	246	0xF6	
SCP_BUFFER_TOO_SHORT	247	0xF7	
SCP_BUFFER_1_TOO_SHORT	248	0xF8	
SCP_ILLEGAL_PROTOCOL_SEQUENCE	249	0xF9	
SCP_ILLEGAL_PDU_ARRIVED	250	0xFA	
SCP_REQUEST_ERROR	251	0xFB	
SCP_NO_LICENSE	252	0xFC	

SCP 接口的其它在线 DLL 消息

E_TIMER_INIT	768	0x0300	WIN Set-timer 请求不成功
E_INIT_COM	769	0x0301	
E_NO_HW	784	0x0310	未找到 MPI 模块
E_HW_DEFECT	785	0x0311	硬件出现问题
E_CNF	786	0x0312	组态参数不正确
E_BAUDRATE	787	0x0313	波特率/中断向量不正确
E_HSA	788	0x0314	组态的 HSA 不正确
E_TS	789	0x0315	所组态的地址已分配
E_OCC	790	0x0316	HW_Device 已分配
E_INT_NOT_PROV	791	0x0317	中断不可用

6.7 S5 Ethernet Layer 4

E_INT_BUSY	792	0x0318	中断已占用
E_SAP	793	0x0319	SAP 禁用: SAP 未被占用
E_UNPLUGGED	794	0x031a	未找到远程站
E_SYNI	795	0x031b	发生 Syni 错误
E_AMPRO	796	0x031c	AMPRO 2 报告了系统错误
E_BUFFSIZE	797	0x031d	未创建此大小的缓冲区
E_NO_FILE	800	0x0320	未找到 DLL/VxD 文件或注册表项已遭到破坏
E_NO_ENTRY	801	0x0321	DLL 中不存在地址
E_VERSION	816	0x0330	SMC 驱动程序和 SMC 固件之间发生版本冲突
E_COMCNF	817	0x0331	COM 端口组态出现问题
E_NO_SMC	818	0x0332	SMC 不再响应
E_COMMBADID	819	0x0333	COM 端口未组态
E_COMMOPEN	820	0x0334	COM 端口不可用
E_SMCBUSY	821	0x0335	串行驱动程序目前正在与另一个组态一起使用
E_SMCMODEM	822	0x0336	PC/MPI 电缆不存在连接
E_SMCNOLEG	823	0x0337	PC/MPI 电缆拒绝请求, 缺少必需的授权
E_ONLINE	896	0x0380	IOCTL 接口出现内部错误
E_LOGDEV	897	0x0381	注册表中没有逻辑设备
E_L2DRIVER	898	0x0382	注册表中缺少 L2DRIVER 条目
E_L4DRIVER	900	0x0384	注册表中缺少 L4DRIVER 条目
E_SYSERROR	1023	0x03FF	系统错误

通道特定的错误代码

EC_WATCHDOG	30000	0x7530	监视狗错误
EC_PDUERROR	30001	0x7531	PDU 不是所需的
EC_ONLERROR	30005	0x7535	装载 S7-Online-DLL 时出现故障

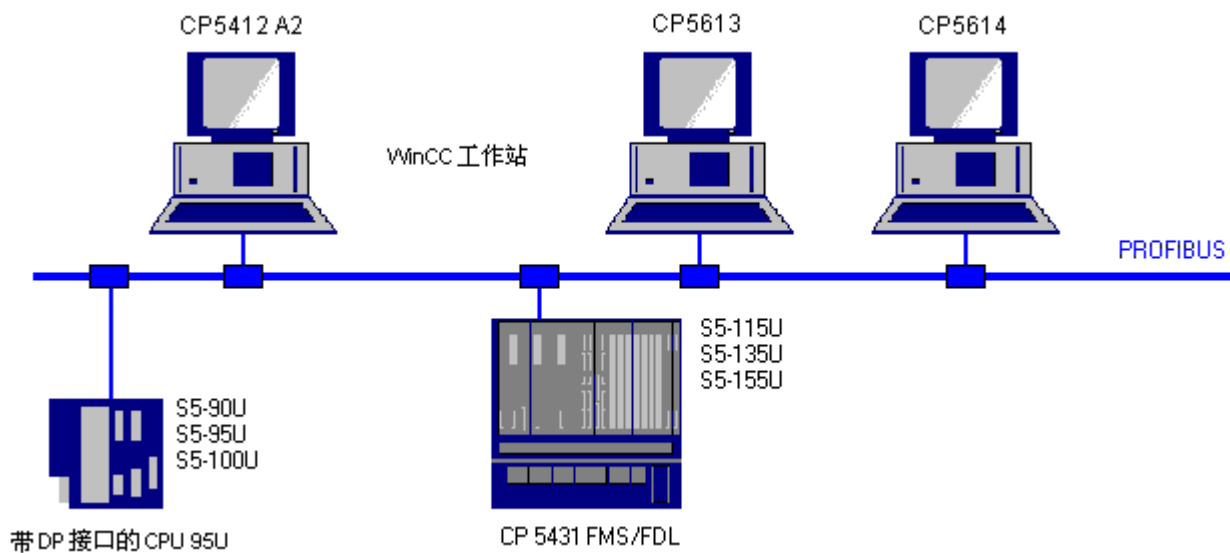
6.8 S5 PROFIBUS FDL

6.8.1 WinCC 通道“SIMATIC S5 Profibus FDL”

简介

通道“SIMATIC S5 Profibus FDL”用于在 WinCC 站与 SIMATIC S5 自动化系统之间进行通信。在这种情况下，采用网络类型 PROFIBUS（过程现场总线）和 FDL（现场数据链接层）协议。

PROFIBUS 是适用于中小型数据容量的网络。最多 127 个可连接节点可满足广泛的自动化任务。



变量是使用通过请求和响应帧完成的 FDL 协议，通过 PROFIBUS 进行读/写操作。请求帧将被从 WinCC 发送至自动化设备。AS 以响应帧进行回答。

FDL 连接通过本地和远程连接端点（服务访问点）指定。

本部分将说明

- 如何使用“SIMATIC S5 Profibus FDL”通道组态数据传输。
- 如何创建实例项目

通道单元 FDL (CP5412/A2-1)

无论使用哪种通信处理器，都可通过“FDL (CP5412/A2-1)”通道单元连接到 SIMATIC S5。

此通道单元最多可支持 24 个连接。为了使通道起作用，必须创建通道单元和连接。

服务访问点

SAP 是 PROFIBUS 节点中的本地数据接口。SAP 必须在 WinCC 和 AS 中进行组态。使用“服务访问点”定义唯一标识。WinCC 和 AS 之间的通信需要此唯一标识。

主动连接

主动连接也称为“读取”连接。主动伙伴通过此连接读取通信伙伴的数据。从其读取数据的通信伙伴被称为被动伙伴。

被动连接

被动连接存在的条件是，主动 AS 以异步而没有请求帧的方式向被动 WinCC 伙伴发送数据。

6.8.2 受支持的数据类型和数据范围

简介

通过 PROFIBUS FDL 进行的 SIMATIC S5 通讯仅支持某些数据类型和数据范围。

所支持的数据类型

WinCC 数据类型	SIMATIC S5 数据类型
二进制变量	BIT
有符号 8 位数	SIMATIC S5 中不存在
无符号 8 位数	BYTE
有符号 16 位数	WORD
无符号 16 位数	WORD
有符号 32 位数	DWORD
无符号 32 位数	DWORD
浮点数 32 位 IEEE 754	DWORD
浮点数 64 位 IEEE 754	SIMATIC S5 中不存在
文本变量，8 位字符集	ARRAY OF BYTE

WinCC 数据类型	SIMATIC S5 数据类型
文本变量, 16 位字符集	SIMATIC S5 中不存在
原始数据类型	ARRAY OF BYTE

访问 SIMATIC S5 变量

访问 SIMATIC S5 变量通过按字访问数据块 DB 或扩展数据块 DX 来完成。这允许读写访问。

访问 BIT 数据类型的 SIMATIC S5 变量

BIT 数据类型的 SIMATIC S5 变量只允许读访问。此限制适用于主动或被动连接。

访问 BYTE 数据类型的 SIMATIC S5 变量

BYTE 数据类型的 SIMATIC S5 变量只允许读访问。

要组态字节变量，必须选择 16 位数据字的“左字节”或“右字节”以用于寻址。

访问 ARRAY OF BYTE 数据类型的 SIMATIC S5 变量

ARRAY OF BYTE 数据类型的 SIMATIC S5 变量只允许读访问。

6.8.3 WinCC 通道“SIMATIC S5 Profibus FDL”的功能

简介

以下列出了在 WinCC 中通过 PROFIBUS 的通信驱动程序的通信功能。同时还列出了所有受支持的数据类型以及相应的类型转换功能。

说明

写入变量时的特性

在 WinCC 中组态时，确保在将多个变量写入自动化系统的数据区时单独传送各个变量。

使用“SetTagMultiWait”函数写入多个变量（例如在脚本中）时，此行为特别重要。由于此函数只有在将发送到该函数的所有变量传输结束时才执行，因而传输大量变量就会出现明显的等待时间。

必须通过检查来确定在传输大量变量时是否需要使用“Wait”函数。在这种情况下，使用原始数据变量可能也是一个好方法，当 AS 数据区中的数据连续存储时尤其如此。

类型转换

如果需要某个值范围或者需要某种转换，例如，十进制转换为BCD码（“无符号8位数”转换为“ByteToBCDWord”），则需要进行类型转换。默认情况下不执行类型转换。

下表列出了受支持的WinCC数据类型以及相应的类型转换能力。

WinCC 数据类型	类型转换
二进制变量	否
有符号8位数	不适用于S5
无符号8位数	是
有符号16位数	是
无符号16位数	是
有符号32位数	是
无符号32位数	是
浮点数32位IEEE 754	是
64位IEEE 754浮点数	不适用于S5
文本变量，8位字符集	否
文本变量，16位字符集	不适用于S5
原始数据类型	否

WinCC 端

通信驱动程序 SIMATIC S5 Profibus FDL 支持使用下列通信处理器进行通信：

通信处理器	总线类型
CP 5613 A3	PCI
CP 5612	PCI

AS 端

通常，可以通过两种不同方式将可编程逻辑控制器连接到 PROFIBUS 网络。

可以通过中央模块上的集成接口进行连接，也可以使用特殊通信模块进行连接。

系统	模块
S5-90U、S5-95U、S5-100U	CPU95U
S5-115U、S5-135U、S5-155U	CP5431 FMS/DP

说明

由于 WinCC 只能作为被动伙伴，因而使用 L2-SS 连接到 S5-95U 时，不能进行“读取”连接。

6.8.4 组态通道

6.8.4.1 如何组态通道“SIMATIC S5 Profibus FDL”

简介

本章及后文内容都介绍了组态通道“SIMATIC S5 Profibus FDL”的步骤。

本部分介绍如何组态通道“SIMATIC S5 Profibus FDL”。

步骤

1. 在变量管理的导航区域中，从节点“变量管理”(Tag Management)的快捷菜单中选择条目“添加新的驱动程序”(Add new driver)。
2. 选择驱动程序“SIMATIC S5 Profibus FDL”。随即将创建该通道，并在变量管理器中显示该通信驱动程序。

6.8.4.2 通道单元“FDL (CP5412/A2-1)”

简介

“SIMATIC S5 Profibus FDL”通信驱动程序仅包含“FDL (CP5412/A2-1)”通道单元。

WinCC 和 SIMATIC S5 可编程逻辑控制器之间的通信可通过“FDL (CP5412/A2-1)”通道单元进行。

在通道单元中最多可创建 24 个连接。必须为每个已组态的连接设置特殊的连接参数。必须通过变量参数定义各个组态的变量。

说明

“FDL (CP5412/A2-1)”通道单元的名称受通信驱动程序“SIMATIC S5 Profibus FDL.CHN”的约束，而与所使用的通信处理器无关。

例如，CP5613 A3 可用作通信处理器。

变量参数

必须为每个已组态的变量指定下列变量参数：

- 数据区（如 DB）
- 数据块号
- 寻址（如“左字节”）
- 起始地址（如 DL 0，“左字节”已被选择用于寻址时）

连接参数

必须为每个已组态的连接指定下列连接参数：

- AS 的站地址
- 优先级
- 必须为读写功能指定自身和外部 SAP（服务访问点）

无论连接是主动连接还是被动连接，都必须为读取功能组态此连接。在主动读取连接的情况下，这些值是由 WinCC 站请求。在被动连接情况下，则由 AS 启动向 WinCC 站进行的数值传送。

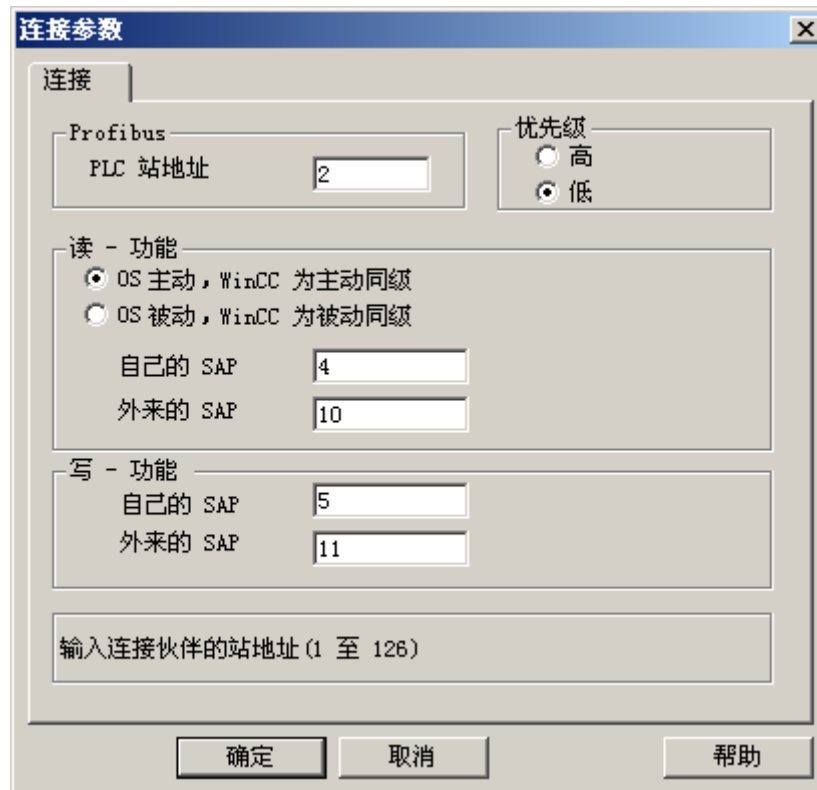
6.8.4.3 如何组态连接

要求

- 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。

步骤

1. 选择通道单元“FDL (CP5412/A2-1)”。
2. 在通道单元的快捷菜单中选择“新建连接”(New Connection) 条目。
3. 输入连接的名称。
4. 从连接的快捷菜单中选择“连接参数”(Connection parameters) 条目。“连接属性”(Connection properties) 对话框随即打开。



5. 在“PLC 站地址”域中，输入 AS 的唯一地址。
6. 对于 FDL 连接，“优先级”选项按钮必须始终设置为“低”。
7. 使用“OS 主动，WinCC 为主动伙伴”或“OS 被动，WinCC 为被动伙伴”域定义 WinCC 站的功能。激活所需的选项按钮。
8. 在“自身的 SAP”和“外来的 SAP”域中，输入为读写访问组态的 SAP 地址。SAP 值范围介于 2 和 54 之间。
9. 单击“确定”(OK) 关闭所有打开的对话框。
10. 从连接的快捷菜单中选择“新建变量”(New Tag) 选项。将打开“变量属性”(Tag Properties) 对话框。组态该变量。
11. 单击“确定”(OK) 关闭所有打开的对话框。

6.8 S5 PROFIBUS FDL

6.8.4.4 组态变量

组态变量

简介

对于 WinCC 与 AS 之间通过通道“SIMATIC S5 Profibus FDL”进行的连接，可在 WinCC 中定义二进制、字节和字这些数据类型。下面将说明如何组态这些数据类型的变量。

如何组态按位访问的变量

简介

本部分将介绍如何为 AS 中的地址区域组态一个按位访问的变量。

说明


按位访问变量为只读访问。

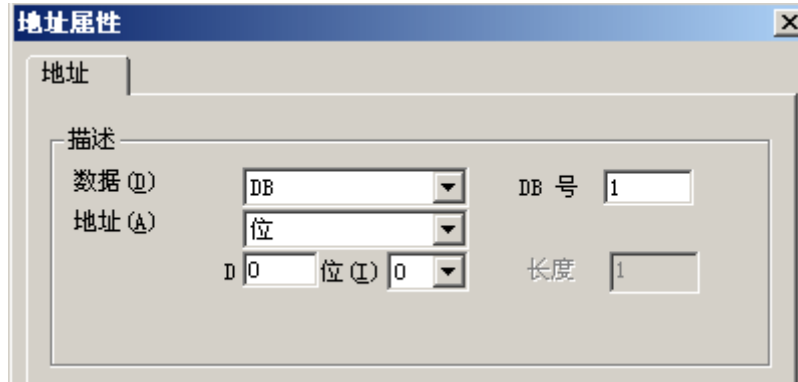
要求

1. 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。
2. 必须在通道单元“FDL (CP5412/A2-1)”中定义连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data Type) 字段中，设置“二进制变量”(Binary tag) 数据类型。

- 打开“地址属性”(Address properties)对话框。
为此，请单击“地址”(Address)字段，然后单击  按钮。



- 在“D”域中输入字节地址，在“位”域中输入位地址。左侧域标签取决于“数据区”域中的条目，例如，如果数据区为“DB”且变量的数据类型为二进制变量，则左侧域标签为“D”。
- 单击“确定”(OK)关闭所有打开的对话框。

说明

无法更改“寻址”域中的“位”条目，因为这是由 WinCC 变量的“二进制变量”数据类型定义的。

如何组态按字节访问的变量

简介

本节说明如何组态一个用于按字节访问 AS 中的地址区域的变量。


说明

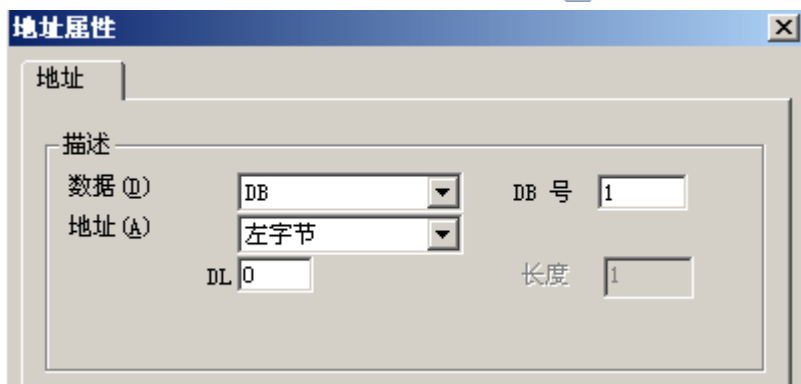
按字节访问变量只是读访问。

要求

- 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。
- 必须在通道单元“FDL (CP5412/A2-1)”中定义连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data Type) 字段中，将数据类型设置为“无符号 8 位数”(Unsigned 8-bit value)。
5. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



6. 在“寻址”域中，可以在“左字节”和“右字节”之间进行选择。
7. 在“DL”域中，输入字节地址。域标签取决于“寻址”域中的条目，例如，对于“左字节”地址则为“DL”。
8. 单击“确定”(OK) 关闭所有打开的对话框。

如何组态按字访问的变量

简介

本节说明如何组态一个用于按字访问 AS 中的地址区域的变量。


说明

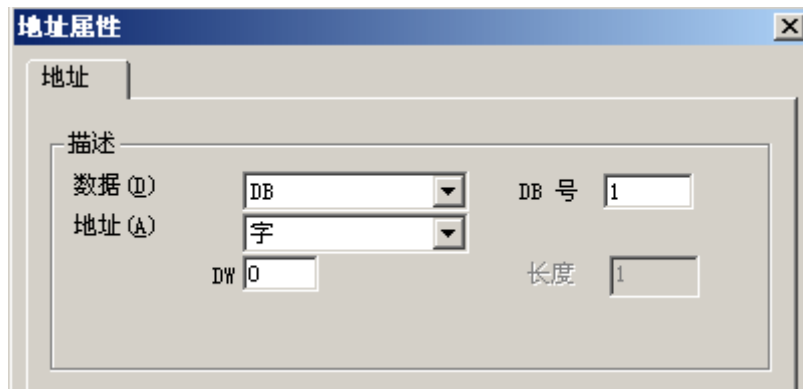
按字访问变量为读访问和/或写访问。

要求

1. 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。
2. 必须在通道单元“FDL (CP5412/A2-1)”中定义连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data Type) 字段中, 将数据类型设置为“无符号 16 位数”(Unsigned 16-bit value)。
5. 打开“地址属性”(Address properties) 对话框。
为此, 请单击“地址”(Address) 字段, 然后单击  按钮。



6. 在下面的域中输入字地址。域标签取决于“寻址”域中的条目, 例如, 对于“字”地址则为“DW”。
7. 单击“确定”(OK) 关闭所有打开的对话框。

说明

“寻址”(Addressing) 字段中的条目“字”(Word) 无法更改, 因为这已经由 WinCC 变量的数据类型“无符号 16 位数”(Unsigned 16-bit value) 定义。

6.8.4.5 系统参数

通道单元的系统参数

简介

如果需要与 WinCC 标准设置不一样的组态, 则可使用通道单元的“系统参数”对话框进行所需的全部更改。

可以更改以下各项：

- 设备名称
- 读/写监视时间

设备名称

WinCC 和自动化系统之间的通讯通过逻辑设备名称执行。这些名称是在安装通讯模块时分配的，并且是单元特定的条目。设备名称代表逻辑设备名称。默认情况下，此域定义为条目“/CP_L2_1:/SCP”。

读/写监视时间

读/写监视时间是 AS 写/读响应的最长等待时间（单位为秒）。如果 AS 在定义的时间内未做出任何响应，则表示连接断开了。默认情况下，为此域分配的等待时间值为 30 秒

说明

系统参数适用于 AS 中的所有 CP。

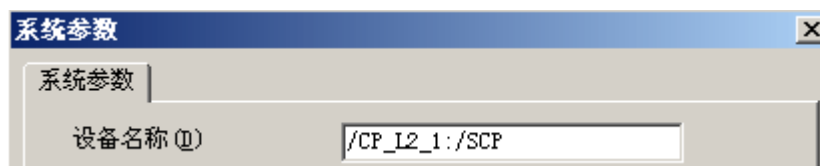
如何更改设备名称

要求

- 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。

步骤

1. 在通道单元快捷菜单中，选择“系统参数”。将打开“System Parameters”（系统参数）对话框。



2. 在“设备名称”域中，输入访问点的名称。此名称必须与在 Windows 下通过“开始” → “设置” → “控制面板” → “设置 PG/PC 接口”所设置的一样。
3. 单击“确定”按钮，关闭对话框。

说明

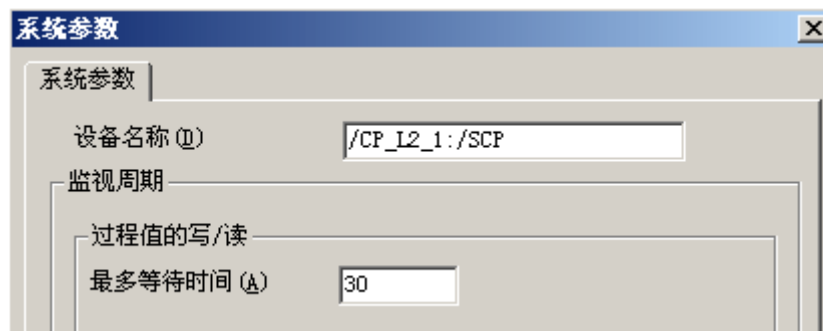
这些更改只有在 WinCC 重启后才有效。

如何更改过程值的写/读监视时间**要求**

- 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。

步骤

1. 在通道单元快捷菜单中，选择“系统参数”。将打开“System Parameters”（系统参数）对话框。



2. 在“最多等待时间”域中，输入所需的值（单位为秒）。可以定义一个介于 1 和 3600 秒之间的值。为此域分配的默认值为 30 秒。
3. 单击“确定”按钮，关闭对话框。

说明

这些更改只有在 WinCC 重启后才有效。

6.8.5 特殊功能

6.8.5.1 “SIMATIC S5 Profibus FDL”通道的特殊功能

简介

“SIMATIC S5 Profibus FDL”通道具有一些特殊功能，本章将介绍这些特殊功能。

6.8.5.2 “SIMATIC S5 Profibus FDL”通道的原始数据变量

“SIMATIC S5 Profibus FDL”通道的原始数据变量

简介

“原始数据类型”的变量是一种数据报文。

原始数据变量是从 AS 传送用户数据块或者将用户数据块传送到 AS 所需的 SIMATIC S5 Profibus FDL 使用的原始数据变量最长可为 220 个字节。

原始数据变量作为字节数组

作为字节数组的原始数据变量的处理方式类似于通常的过程变量，都是通过数据块的地址和长度进行寻址（例如：DB 100，DW 20，长度为 40 个字节）。

使用脚本写入原始数据变量


如果使用 VB 脚本将长度长于 WinCC 中组态的变量长度的原始数据写入 S5，写入过程将中止。而应通过 C 脚本使用“SetTagRaw”函数写入原始数据变量。对于此函数，需指定要写入的变量的长度。

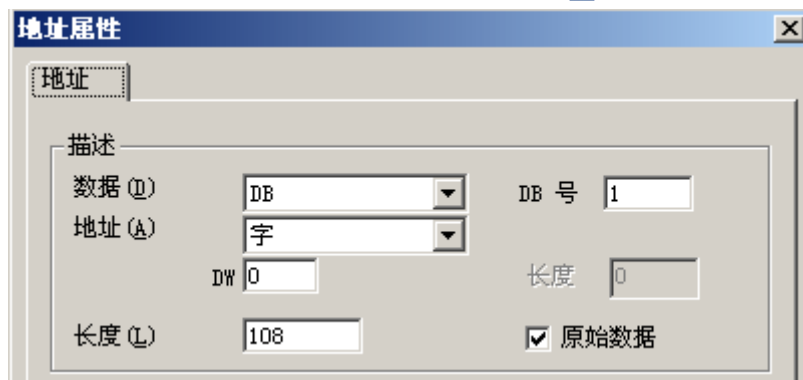
如何组态原始数据变量

要求

1. 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。
2. 必须在通道单元“FDL (CP5412/A2-1)”中定义连接。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data type) 字段中, 选择“原始数据类型”(Raw Data Type)。
5. 单击“选择”按钮以打开“地址属性”对话框。
打开“地址属性”(Address properties) 对话框。
为此, 请单击“地址”(Address) 字段, 然后单击  按钮。



6. 选中“原始数据”(Raw Data) 复选框。
7. 在“长度”域中, 输入原始数据块的长度 (以字节为单位)。
8. 在“数据区”中, 设置数据所在的 PLC 数据区。如果选择“DB”作为数据区, 则在已启用的“DB 号”(DB No.) 字段中输入数据块的编号。
9. 在“寻址”域中设置寻址类型。条目“左字节”、“右字节”、“字”和“双字”都可用于 WinCC 变量的数据类型“原始数据类型”。
10. 在下面的域中, 输入起始地址的值。左侧域上的标签取决于“数据区”和“寻址”域中的条目, 例如, 数据区为“DB”、寻址类型为“字”时, 显示“DW”条目。
11. 单击“确定”(OK) 关闭所有打开的对话框。

6.8.5.3 组态通讯类型

组态通讯类型

简介

可以组态 FDL 连接, 以便 WinCC 作为主动或被动伙伴运行。

如果将 WinCC 组态为主动伙伴, 则由 WinCC 站请求数值。

如果将 WinCC 组态为被动伙伴，则由 AS 启动向 WinCC 站进行的数值传送。

如何组态主动数据传送

简介

本节说明如何组态一个向 AS 中的地址区域进行的主动数据传送。

说明

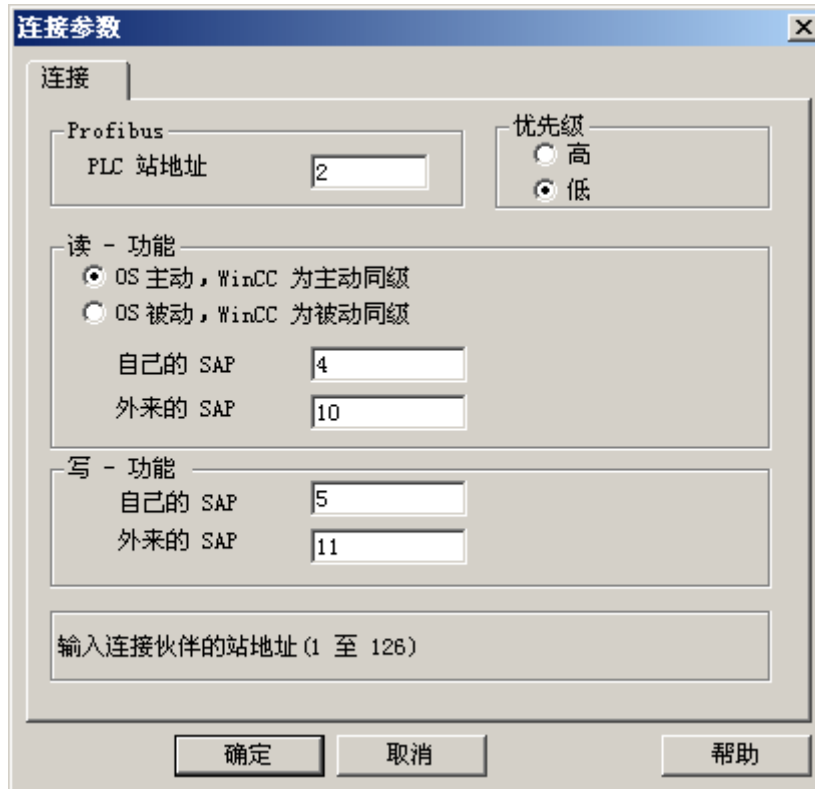
如果组态了多个连接，请注意：一个 SAP 只能被分配一次。

要求

1. 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。
2. 必须在通道单元“FDL (CP5412/A2-1)”中定义连接。
3. 必须知道由 AS 定义的 SAP 地址。

步骤

1. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。



2. 在“连接”选项卡上的“PLC 站地址”域中，输入 AS 的站地址。
3. 在“读 - 功能”区域中，必须进行下列设置：
4. 激活选项“OS 主动，WinCC 为主动伙伴”。
5. 在“自己的 SAP”域中，输入 WinCC 站的 SAP-ID。
6. 在“外来的 SAP”域中，输入 AS 的 SAP-ID。
7. 在“写 - 功能”区域中，必须进行下列设置：
8. 在“自己的 SAP”域中，输入 WinCC 站的 SAP-ID。
9. 在“外来的 SAP”域中，输入 AS 的 SAP-ID。
10. 单击“确定”(OK) 关闭所有打开的对话框。

如何组态被动数据传送

简介

本节说明如何组态一个向 AS 中的地址区域进行的被动数据传送。

说明

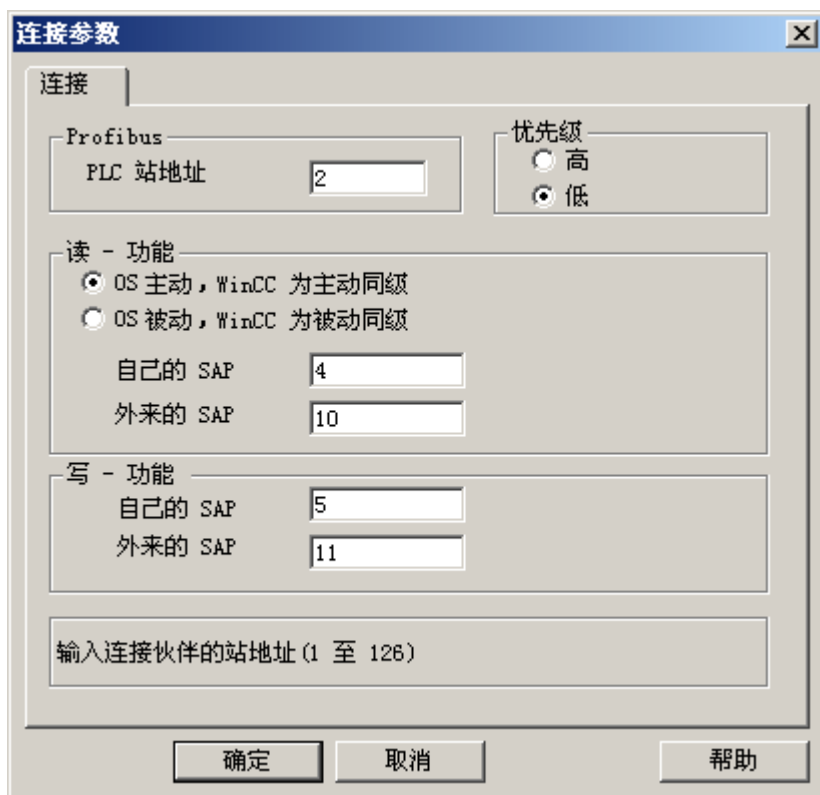
如果组态了多个连接，请注意：一个 SAP 只能被分配一次。

要求

1. 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。
2. 必须在通道单元“FDL (CP5412/A2-1)”中定义连接。
3. 必须知道由 AS 定义的 SAP 地址。

步骤

1. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。



2. 在“连接”选项卡上的“PLC 站地址”域中，输入 AS 的站地址。
3. 在“读 - 功能”区域中，必须进行下列设置：

4. 激活选项“OS 被动，WinCC 为被动伙伴”。
5. 在“自己的 SAP”域中，输入 WinCC 站的 SAP-ID。
6. 在“外来的 SAP”域中，输入 AS 的 SAP-ID。
7. 在“写 - 功能”区域中，必须进行下列设置：
8. 在“自己的 SAP”域中，输入 WinCC 站的 SAP-ID。
9. 在“外来的 SAP”域中，输入 AS 的 SAP-ID。
10. 单击“确定”(OK) 关闭所有打开的对话框。

6.8.6 “SIMATIC S5 Profibus FDL”通道的组态实例

6.8.6.1 “SIMATIC S5 Profibus FDL”通道的组态实例

简介

在本实例中，将在图形编辑器中组态 I/O 域，并将所需的值分配给 AS 中的数据处理块。

6.8.6.2 如何组态 AS 中的数据处理块

简介

在本部分，将组态 AS 中的标准功能块 OB 21 (L2ANLAUF) 和 OB 1 (L2SNDRVC)。

默认情况下，通过 PROFIBUS FDL 进行的 SIMATIC S5 连接的数据通信流量使用以下块处理。

在实例中，采用以下 SAP 编号：

	WinCC	可编程控制器
读功能的 SAP 编号	12	6
写功能的 SAP 编号	11	4

块

功能	块
用于启动的 OB 20、21、22	FB-L2ANLAUF (FB 9)
用于循环操作的 OB 1	FB-L2SNDRCV (FB 10)
作为两个 FB 的内部工作数据块	DB-L2DBVC3 (DB 10) DB-L2DBVC4 (DB 11) DB-L2DBVC5 (DB 12)

启动块

在启动块中，指定通信参数，注册工作 DB，并同步通信处理器。

例如，通过调用功能块 FB9 L2ANLAUF 来执行这些工步。

循环块

帧通信流量在循环 FB 中处理。

接收到的帧被输入到目标数据块中。如果在此期间出现错误，就会拒绝该帧并发出错误消息。

用户指定要发送的帧，与标准数据处理块类似。

反馈会在完成传输后发生。

要求

- 数据处理块 SYNCHRON、CONTROL、SEND 和 RECEIVE 必须在 AS 中可用。

步骤

1. 在 STEP5 软件中，使用菜单项“编辑器” → “STEP5 块” → “在程序文件中”，创建启动块（OB 20、21、22）。
示例中的程序块名称为“L2ANLAUF”。
2. 必须预先分配下列参数：
 - CP（例如 CP5431）的接口号 (SSNR)
 - WinCC 计算机上的通信处理器 CP 5613 A3 的 PROFIBUS 地址 (RADR)。该编号在网络中必须唯一。
 - 指定了 WinCC 站的 SAP 的所用请求类型的连接参数，例如，用于写访问的参数 RVC4 和用于读访问的 RVC5。在 WinCC 中创建连接时指定这些 SAP。
 - 组态通信处理器的 FDL 连接期间，设置的请求编号（ANR4 和 ANR5）
 - 工作数据块的编号，即 DBX4（用于写访问）和 DBX5（用于读访问）

```

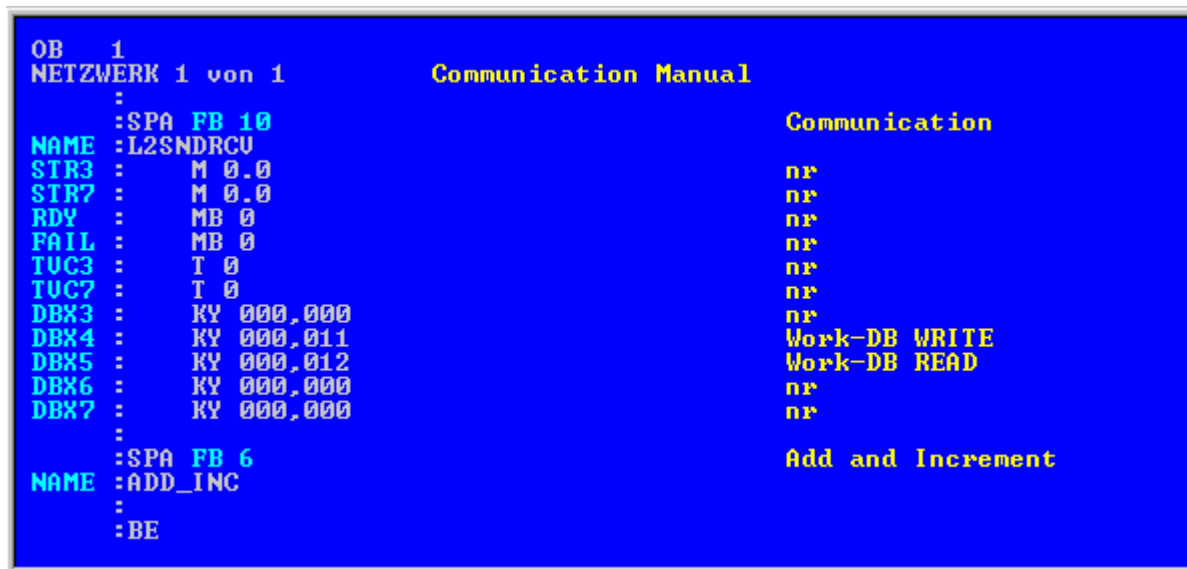
OB 21
NETZWERK 1 von 1
:
:SPA FB 9
NAME :L2ANLAUF
SSNR : KF +0          SSNR of CP 5431
TIM3 : KT 000.0      nr
TIM7 : KT 000.0      nr
RADR : KF +8         PROFIBUS-Address WinCC Station
RUC3 : KF +0         nr
RUC4 : KF +4         SAP WRITE
RUC5 : KF +6         SAP READ
RUC6 : KF +0         nr
RUC7 : KF +0         nr
ANR3 : KF +0         nr
ANR4 : KF +134       ANR WRITE
ANR5 : KF +135       ANR READ
ANR6 : KF +0         nr
ANR7 : KF +0         nr
DBX3 : KY 000,000    nr
DBX4 : KY 000,011    Work-DB WRITE
DBX5 : KY 000,012    Work-DB READ
DBX6 : KY 000,000    nr
DBX7 : KY 000,000    nr
S/R3 : KF +0         nr
:
:BE

```

3. 在 STEP5 软件中，使用菜单项“编辑器” → “STEP5 块” → “在程序文件中”，创建 OB 1（循环操作）。
示例中的程序块名称为“L2SNDRCV”。

6.8 S5 PROFIBUS FDL

- 例如，使用通信处理器 CP5431 和功能块 FB10 L2SNDRVCV 执行与 WinCC 间的通信。当 WinCC 发送和请求数据时，只需指定两个关联 IN/OUT 参数即可。它们就是参数 DBX4（用于写访问）和 DBX5（用于读访问），这些参数指定所用请求类型的两个工作数据块的编号。在 WinCC 中创建连接时指定这些 SAP。



- 将 STEP 5 程序下载到可编程逻辑控制器。这通过在 STEP5 软件中使用菜单项“对象” → “块” → “传送” → “PLC 文件”实现。在“选择”(Selection)域中选择“所有块”(All blocks)选项，将先前创建的所有块下载到自动化系统。

6.8.6.3 如何组态 I/O 域


简介

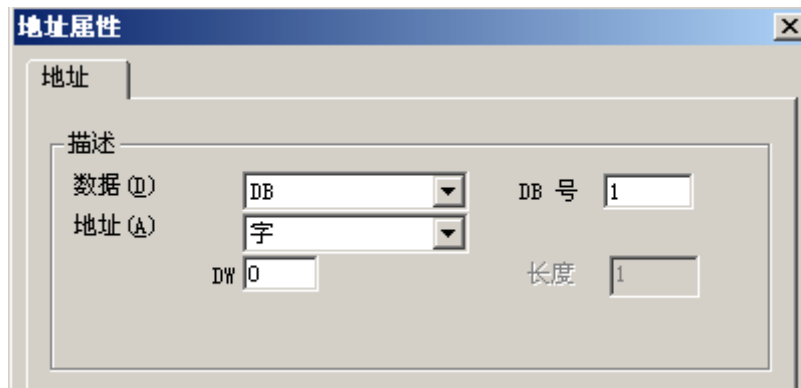
在本部分，将组态 I/O 域。

要求

- 通道“SIMATIC S5 Profibus FDL”必须集成在项目中。

步骤

1. 从通道单元“FDL (CP5412/A2-1)”的快捷菜单中选择“新建连接”(New Connection) 选项，并建立名为“TestFDL”的连接。
2. 从连接的快捷菜单中选择“连接参数”(Connection parameters) 条目。
“连接属性”(Connection properties) 对话框随即打开。
在“PLC 站地址”(PLC Station Address) 字段中输入 AS 的站地址。
可以用此种方式组态 FDL 连接，使 WinCC 成为主动或被动伙伴。如果将 WinCC 组态为主动伙伴，则由 WinCC 站请求数值。如果将 WinCC 组态为被动伙伴，则由 AS 启动向 WinCC 站传送数值。
单击“确定”(OK)，关闭所有打开的对话框
3. 单击表格区下方的“变量”(Tags) 选项卡。
4. 单击“名称”(Name) 列最上方的空单元格。
输入“FDLWord1_Test”作为名称。
5. 在“数据类型”(Data Type) 字段中，将数据类型设置为“无符号 16 位数”(Unsigned 16-bit value)。
6. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



7. 在“数据区”中，设置数据所在的 PLC 数据区。如果选择“DB”作为数据区，则在已启用的“DB 号”(DB No.) 字段中输入数据块的编号。在“DW”域中，输入字节地址。
8. 单击“确定”(OK) 关闭所有打开的对话框。
9. 必须在图形编辑器中将智能对象“I/O 域”集成到画面中。
10. 将“I/O 域”与一个组态变量连接。
11. 通过单击工具栏上的“激活运行系统”按钮，或通过选择“文件”菜单中的“激活运行系统”来激活项目。组态变量的所有更改均显示在运行系统的“I/O 域”中。

6.9 S5 Programmers Port AS511

6.9.1 WinCC 通道“SIMATIC S5 Programmers Port AS511”

简介

通讯驱动程序“SIMATIC S5 Programmers Port AS511”用于通过 TTY 接口与 SIMATIC S5 自动化系统进行串行连接。

本章描述了

- 如何使用“SIMATIC S5 Programmers Port AS511”通道组态数据传送。
- 如何组态连接和变量。

通道单元

通讯驱动程序具有一个用于控制串行连接的 COM 端口的通道单元。

提供了以下功能：

- 通道单元 S5-AS511，用于通过“西门子特定的”协议进行串行通讯。

6.9.2 变量的数据类型

简介

为逻辑连接定义所需变量。从 WinCC 的角度而言，可以访问以下数据类型：

- 二进制变量
- 无符号 8 位数
- 有符号 8 位数
- 无符号 16 位数
- 有符号 16 位数
- 无符号 32 位数
- 有符号 32 位数
- 浮点数 32 位 IEEE 754

- 文本变量，8 位字符集
- 原始数据类型

6.9.3 组态通道

6.9.3.1 组态“SIMATIC S5 Programmers Port AS511”通道

简介

组态通道“SIMATIC S5 Programmers Port AS511”需要以下步骤。

- 组态连接
- 组态变量

6.9.3.2 如何组态连接

简介

SIMATIC S5 自动化系统可以使用串行连接来进行过程连接。自动化系统中使用 AS 511 通信处理器。

WinCC 中不需要附加通信模块。通信可通过 PG 760 上的 TTY 端口或作为系统标准设备一部分的 COM 端口加以建立。对于后者，还需要一个端口转换器 V.24/V.28 <---> TTY。

这种串行连接所支持的传输率高达 19200 波特。

可以使用以下步骤将 PC 的一个串行端口分配给 AS511-NT 驱动程序。

说明

在 WinCC 与 S5 自动化系统通过“AS511”通道进行通信期间，不能在 AS 中传送、创建或删除数据块。S5 中的内存也不能进行压缩。这种终极限制是由于对 S5 中的内存进行绝对寻址而导致的。如果需要更改，则必须断开与 WinCC 的连接。

步骤

1. 在通道单元“S5-AS511”下选择连接。
2. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。



3. 在“端口”域中，选择要用于串行连接的端口。

6.9.3.3 组态变量

组态变量

简介

对于 WinCC 与 AS 之间通过通道“SIMATIC S5 Programmers Port AS511”进行的连接，可在 WinCC 中创建不同数据类型的变量。以下部分将对此进行说明。

说明

在 WinCC 中，不检查变量地址的似真性。如果所使用的地址在 AS 中不可用，则会设置“寻址出错”状态。

在 DB 和 DX 数据块中，只能对地址 255 进行读写访问。

不能写入时间。

如何组态变量的地址


简介

变量地址根据 SIMATIC S5 的地址结构输入。

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡并选择变量。
3. 在“常规”(General) 选项卡中选择所需的数据类型（例如，有符号 8 位数）。

6.9 S5 Programmiers Port AS511

- 4. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
选择“SYSVAR”选项卡。



- 5. 单击选择域，以选择该变量应传送“PLC 类型”、当前状态（“PLC 状态”）还是其它数据（“PLC 数据”）。

6. 只有当您选择了“PLC 数据”，您才会需要单击“地址”选项卡来定义变量的 S5 地址。



7. 在“数据区”域中，选择变量是位于数据块、扩展数据块、标记区、输入范围还是输出范围中。

8. 如果变量位于数据块中，则还会显示“DB 号”域。在此，输入数据块的编号。

9. 在“寻址”域中输入寻址类型。通常，可以使用默认定义。

10. 在相应域（例如“DW”）中，输入地址。

PLC 中的内存经常只能按字节或字存取。在使用二进制和“8 位”变量时，除了会打开对话框“地址属性”外，还将打开“位/字节变量”对话框，该对话框也可用于更改 PLC 内存中的

各个位。为此，对于每个写入请求，均从 PLC 读取已赋址的存储区，从而修改相应的位和/或字节。然后，数据被写回到 PLC 的存储器。

说明

PLC 在读取数据区中执行的更改会在写回到该数据区时被覆盖。

视变量类型而定，可以按位或按字节访问自动化系统的内存。

在 WinCC 中，不检查变量地址的似真性。如果所使用的地址在 AS 中不可用，则会设置“寻址出错”状态。


在 DB 和 DX 数据块中，只能对地址 255 进行读写访问。

不能写入时间。

如何组态按位访问的变量

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
4. 在“数据类型”(Data Type) 字段中，设置“二进制变量”(Binary tag) 数据类型。

5. 打开“位/字节变量”(Bit/Byte tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。




6. 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。
7. 在选择域中，选择 PLC 存储器的寻址类型。
8. 在选择域中，选择要更改的位号。

如何组态按字节访问的变量

步骤

1. 选择要为其组态变量的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。

6.9 S5 Programmers Port AS511

- 在“数据类型”(Data Type) 字段中，将数据类型设置为“无符号 8 位数”(Unsigned 8-bit value) 或“有符号 8 位数”(Signed 8-bit value)。
- 打开“位/字节变量”(Bit/Byte tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



- 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。
- 在选择域中，选择 PLC 存储器的寻址类型。
- 在选择域中，选择要更改的字节号。

6.10 S5 Serial 3964R

6.10.1 WinCC 通道“SIMATIC S5 Serial 3964R”

简介

通讯驱动程序“SIMATIC S5 Serial 3964R”用于使 WinCC 站与 SIMATIC S5 自动化系统之间实现串行连接。

本章描述了

- 如何使用“SIMATIC S5 Serial 3964R”通道组态数据传送。
- 如何组态连接和变量。

通道单元

通讯驱动程序具有一个用于控制串行连接的 COM 端口的通道单元。

提供了以下功能：

- 通道单元 S5-RK512 (3964R)，用于通过 3964R 或 3964 协议进行串行通讯。

6.10.2 变量的数据类型

简介

为逻辑连接定义所需变量。从 WinCC 的角度而言，可以访问以下数据类型：

- 二进制变量
- 无符号 8 位数
- 有符号 8 位数
- 无符号 16 位数
- 有符号 16 位数
- 无符号 32 位数
- 有符号 32 位数
- 浮点数 32 位 IEEE 754

6.10 S5 Serial 3964R

- 文本变量，8 位字符集
- 原始数据类型

6.10.3 组态通道

6.10.3.1 组态“SIMATIC S5 Serial 3964R”通道

简介

组态通道“SIMATIC S5 Serial 3964R”需要以下步骤。

6.10.3.2 如何组态连接

简介

SIMATIC S5 自动化系统可以使用串行连接来进行过程连接。在自动化系统中，CPU 模块上使用通信处理器 CP 544 或另一个插入式串行端口（模块插座 SI2）。

WinCC 中不需要附加通信模块。通过系统中提供的默认 COM 端口进行通信。

这种串行连接所支持的传输率高达 19200 波特。

说明

当 SIMATIC S5 主动发送“Pseudowrite”作业类型时，消息长度不得超过 64 个字。

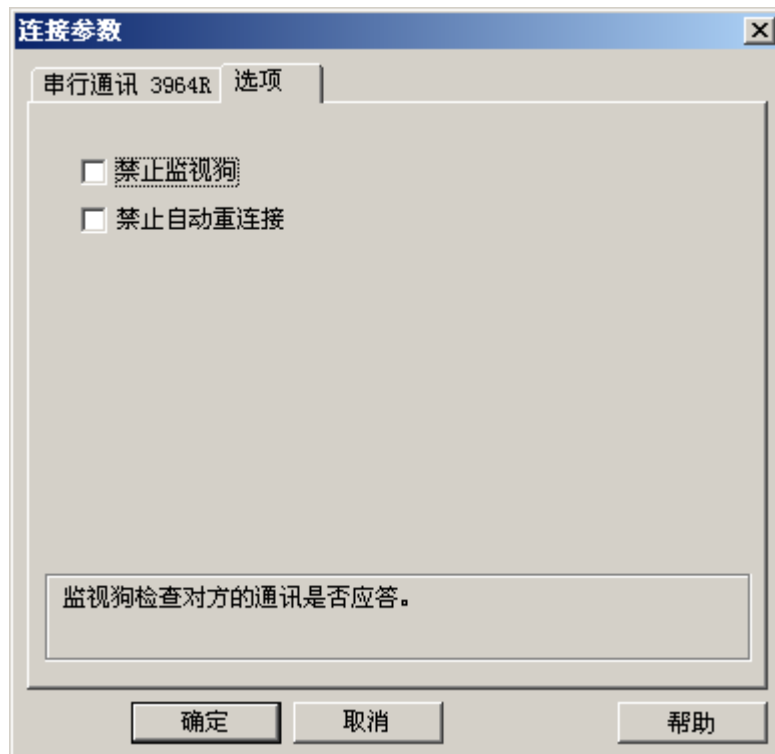
步骤

1. 选择一个连接，然后在快捷菜单中选择“连接参数”(Connection parameters)。
2. 选择“串行通信 3964R”选项卡。



3. 在“端口”域中，选择用于连接的通信端口（COM1 或 COM2）。
4. 在“程序参数”区域的“波特率”域中，将数据传送速度设置为所用的值。在“优先级”域中，设置出现启动冲突时的优先级（WinCC 和自动化系统同时发出的占线请求）。设置的优先级必须与 SIMATIC S5 中设置的优先级不同。
5. 在“程序数据”域中，选择线路协议“3964”或“3964R”。只有在出现意外情况时，才可更改程序数据（例如，确认时间、字符延迟时间等）的默认值。请确保它们与自动化系统中的参数匹配。

6. 现在，选择“选项”选项卡。



7. 在“选项”选项卡上，可以禁用设备状态循环监视，以及禁用自动重新连接。

6.10.3.3 组态变量

组态变量

简介


对于 WinCC 与 AS 之间通过通道“SIMATIC S5 3964R”进行的连接，可在 WinCC 中定义二进制和字节这些数据类型。下面将说明如何组态这些数据类型的变量。

如何组态变量的地址

简介

变量地址根据 SIMATIC S5 的地址结构输入。

步骤

1. 选择变量，然后在“数据类型”(Data Type) 字段中设置变量所需的数据类型（例如：有符号 8 位数值）。
2. 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。打开“地址属性”(Address properties) 对话框。为此，请单击“地址”(Address) 字段，然后单击  按钮。



3. 在“数据区”域中，选择变量是位于数据块、扩展数据块、标记区、输入范围还是输出范围中。
4. 如果变量位于数据块中，则还会显示“DB 号”域。在此，输入数据块的编号。
5. 在“寻址”域中输入寻址类型。通常，可以使用默认定义。
6. 在相应域（例如“DL”）中，输入地址。

说明

只能对输入、输出、定时器和计数器地址区域进行读访问。可以对数据块（DB、DX）进行读写访问。

不要使用值大于 255 的数据字地址。由于系统特性为 RK 512，因此只允许使用 0 到 255 之间的数据字地址。

也可以组态更大的地址，但这会导致此连接的所有组态变量数据损坏。

PLC 中的内存经常只能按字节或字存取。在使用二进制和“8 位”变量时，除了会打开对话框“地址属性”外，还将打开“位/字节变量”对话框，该对话框也可用于更改 PLC 内存中的各个位。为此，对于每个写入请求，均从 PLC 读取已赋址的存储区，从而修改相应的位和/或字节。然后，数据被写回到 PLC 的存储器。

说明

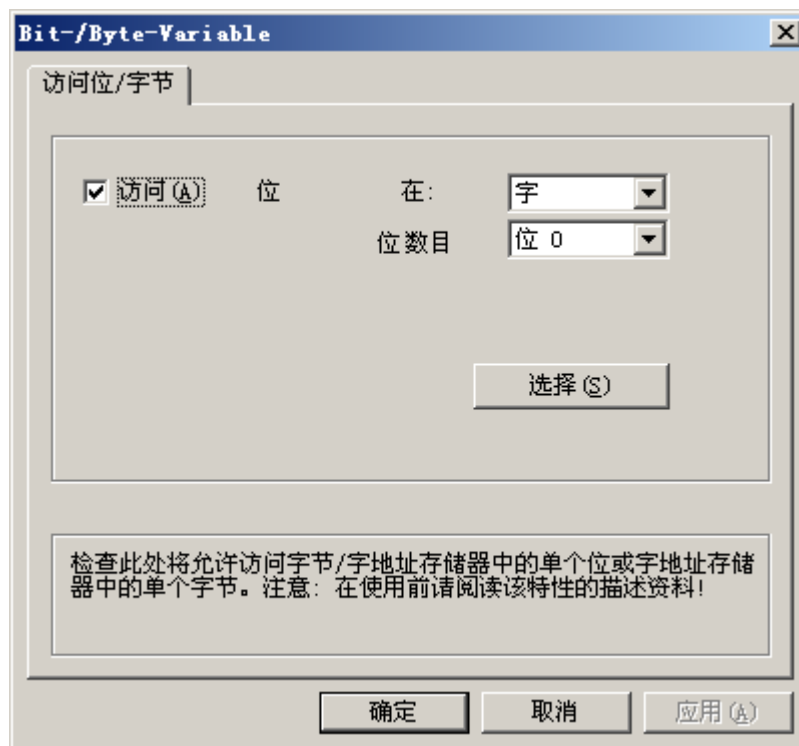
PLC 在读取数据区中执行的更改会在写回到该数据区时被覆盖。

视变量类型而定，可以按位或按字节访问自动化系统的内存。

如何组态按位访问的变量

步骤

1. 选择变量，然后在“数据类型”(Data Type) 字段中设置数据类型“二进制变量”(Binary tag)。
2. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
3. 单击“选择”(Select) 按钮。“位/字节变量”(Bit/Byte tag) 对话框随即打开。
4. 选择“访问位”(Access a bit) 复选框，并定义该位的地址。




5. 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。
6. 在选择域中，选择 PLC 存储器的寻址类型。
7. 在选择域中，选择要更改的位号。

说明

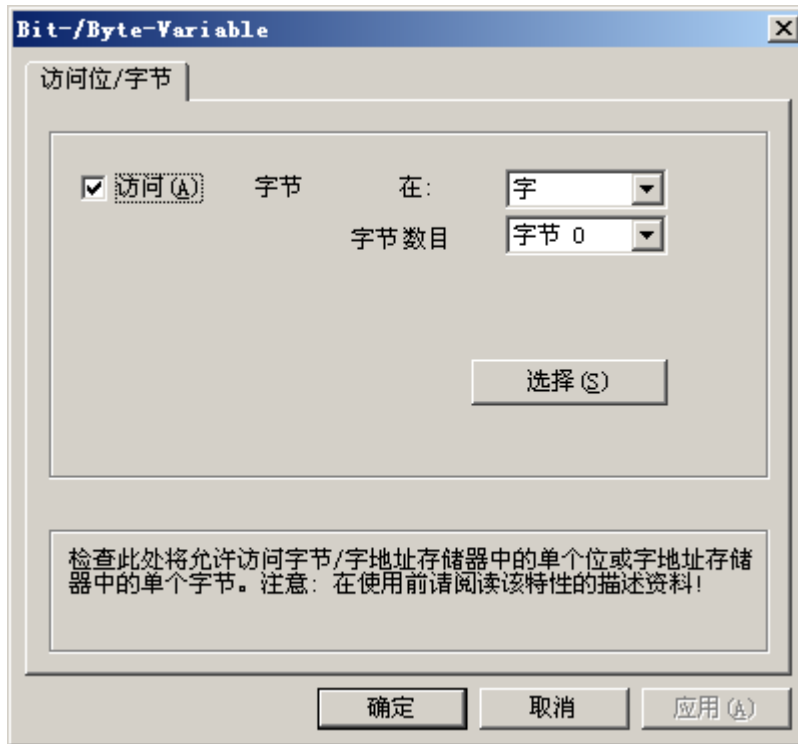
在 S5 中，可以按字节寻址标志位、输入和输出，按字寻址数据块 (DB、DX)。可以对输入、输出、定时器和计数器地址区域进行只读访问。可以对数据块 (DB、DX) 进行读写访问。

如何组态按字节访问的变量

步骤

1. 选择变量，并在“数据类型”(Data Type) 字段中将数据类型设置为“无符号 8 位数”(Unsigned 8-bit value) 或“有符号 8 位数”(Signed 8-bit value)。
2. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
3. 单击“选择”(Select) 按钮。“位/字节变量”(Bit/Byte tag) 对话框随即打开。

4. 选择“访问字节”(Access to a byte) 复选框，并定义该字节的地址。



5. 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。
6. 在选择域中，选择 PLC 存储器的寻址类型。
7. 在选择域中，选择要更改的字节号。

说明

在 S5 中，可以按字节寻址标志位、输入和输出，按字寻址数据块 (DB、DX)。可以对输入、输出、定时器和计数器地址区域进行只读访问。可以对数据块 (DB、DX) 进行读写访问。

6.11 SIMATIC S7 Protocol Suite

6.11.1 WinCC 通道“SIMATIC S7 Protocol Suite”

简介

“SIMATIC S7 Protocol Suite”通道支持 WinCC 站和 SIMATIC S7 自动化系统之间的通信。该协议集支持多种网络协议和类型。

本部分将说明如何：

- 为通道组态各种连接和变量
- 创建示例项目
- 使用通道的特殊功能，例如 AR_SEND 功能、原始数据变量和软件冗余

工作原理

通道“SIMATIC S7 Protocol Suite”用于链接 SIMATIC S7-300 和 SIMATIC S7-400 自动化系统。

根据所用的通信硬件，系统支持通过下列通道单元的连接：

- 工业以太网和工业以太网 (II)：使用通信处理器（例如 CP 1612 A2、CP 1613 A2）通过 SIMATIC NET 工业以太网进行通信。
- MPI：通过编程设备（例如 PG 760/PC RI45）的内部 MPI 接口、MPI 通信处理器或通信模块（例如 CP 5613 A3）进行通信。
- 命名连接：用于通过符号连接与 STEP 7 进行通信。这些符号连接使用 STEP 7 进行组态，为实现高度可靠通信，这些符号连接与 AS S7-400 系列的 H/F 冗余系统一样，都是必须的。
- PROFIBUS 和 PROFIBUS (II)：使用通信处理器（例如 CP 5613 A3）通过 SIMATIC NET PROFIBUS 进行通信。
- Slot-PLC：与 Slot PLC（例如 WinAC Pro）进行通信，这种 PLC 作为 PC 卡安装在 WinCC 计算机上。
- Soft-PLC：与 Software PLC（例如 WinAC Basis）进行通信，这种 PLC 作为应用程序安装在 WinCC 计算机上。
- TCP/IP：使用 TCP/IP 协议与网络进行通信。

有关通道和变量诊断的更多信息，请参阅“通信诊断”。

详细步骤

有关通道组态的更多信息和详细示例，请参见“WinCC V6 通信手册”：

- <http://support.automation.siemens.com/WW/view/en/21320307> (<http://support.automation.siemens.com/WW/view/zh/21320307>)

有关诊断通道和变量的更多信息，请参见“通信诊断(页 563)”。

参见

软件冗余 - 连接专用的内部变量 (页 462)

“SIMATIC S7 Protocol Suite”通道 - 组态 (页 382)

所支持的数据类型概述 (页 381)

通道单元选择 (页 378)

通道和变量的诊断 (页 563)

<http://support.automation.siemens.com/WW/view/en/21320307> (<http://support.automation.siemens.com/WW/view/zh/21320307>)

6.11.2 通道单元选择

简介

要为现有或计划网络建立通信连接，必须进行选择：

- 某一种通道的通道单元
- 用于 WinCC 站的合适的通信处理器
- 用于指定自动化系统的合适的通信模块

本节对可能出现的不同变化进行概述。

有两种不同类型的通信处理器可用于 WinCC：

- 用于所谓的 Hardnet 的通信处理器。它们具有自己的微处理器，可减轻计算机 CPU 上的负荷。可以同时使用两种不同的协议（多协议操作）。
- 用于所谓的 Softnet 的通信处理器。它们没有自己的微处理器。同一时间内只能使用一种协议（单协议操作）。

通道单元的分配

下表显示了“SIMATIC S7 Protocol Suite”通道的通道单元与网络和自动化系统间的分配关系。

通道的通道单元	通信网络	自动化系统
MPI	MPI	S7-300 和 S7-400
PROFIBUS + PROFIBUS (II)	PROFIBUS	S7-300 和 S7-400
工业以太网 + 工业以太网 (II)	工业以太网	S7-300 和 S7-400
TCP/IP	使用 TCP/IP 的 工业以太网	S7-300 和 S7-400
指定连接	工业以太网或 PROFIBUS	S7-400 H/F 系统
Slot PLC	“Soft K-Bus”（内部）	PC 内部
Soft-PLC	“Soft K-Bus”（内部）	PC 内部

MPI

通过 MPI 将 S7-300 和 S7-400 自动化系统进行通信，“MPI”通道单元在“SIMATIC S7 Protocol Suite”通道中可用。

MPI 网络主要对应于具有指定参数以及节点数和传输速率限制的 PROFIBUS 网络。MPI 通信使用与 PROFIBUS 网络相同的通信处理器和模块。而且也使用相同的通信协议。

自动化系统的通信连接

通过 MPI 网络的 S7-300 和 S7-400 自动化系统进行的通信可以通过 AS 内部 MPI 接口或使用合适的通信模块实现。下表显示了建议的组件。

系统	CPU 或通信模块（建议）
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

6.11 SIMATIC S7 Protocol Suite

用于 WinCC 的通信处理器

下表显示了建议用于将 WinCC 站连接到 MPI 网络的通信处理器。每台 WinCC 计算机只能使用一个通信处理器进行 MPI 通信。每个卡也有适合于各个通信协议的驱动程序软件。

通信处理器 (WinCC)	设计/类型
CP 5613 A3	PCI 卡/Hardnet
CP 5612	PCI 卡/Softnet

PROFIBUS

通过 PROFIBUS 与 S7-300 和 S7-400 自动化系统进行通信，“PROFIBUS”和“PROFIBUS II”通道单元在“SIMATIC S7 Protocol Suite”通道中可用。

通道单元支持使用 Hardnet 和 Softnet 模块进行通信。

自动化系统的通信连接

S7-300 和 S7-400 自动化系统通过 PROFIBUS 网络进行的通信可以通过 AS 内部接口或使用通信模块实现。下表显示了建议的组件。

系统	CPU 或通信模块
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

用于 WinCC 的通信处理器

下表显示了建议用于将 WinCC 站连接到 PROFIBUS 的通信处理器。“PROFIBUS”通道单元支持使用 Hardnet 和 Softnet 卡进行通信。WinCC 站可最多使用两个此类模块。每个通信处理器也有适合于各个通信协议的驱动程序软件。

通信处理器 (WinCC)	设计/类型
CP 5613 A3	PCI 卡/Hardnet
CP 5612	PCI 卡/Softnet

工业以太网和 TCP/IP

在 WinCC 中，通过工业以太网进行通信的多个通道单元可在“SIMATIC S7 Protocol Suite”通道中使用。

- “工业以太网”和“工业以太网 (II)”通道单元，使用带 S7 功能的“ISO”协议
- “TCP/IP”通道单元，使用带 S7 功能的“ISO-on-TCP”协议

通道单元支持使用 Hardnet 和 Softnet 模块进行通信。

自动化系统的通信模块

对于通过工业以太网使用“ISO”或“ISO-on-TCP”协议进行的 S7-300 或 S7-400 自动化系统的通信，这些系统配备了合适的通信模块。下表显示了建议的组件。

系统	用于工业以太网的通信模块	用于 TCP/IP 协议的通信模块
S7-300	CP 343-1	CP 343-1 TCP
S7-400	CP 443-1	CP 443-1 TCP CP 443-1 IT

用于 WinCC 的通信处理器

WinCC 站通过工业以太网使用“ISO”或“ISO-on-TCP”协议进行通信，可使用下表中推荐的通信处理器实现。

每个通信处理器也有适合于各个通信协议的驱动程序软件。

通信处理器 (WinCC)	设计/类型
CP 1612 A2	PCI 卡/Softnet
CP 1613 A2	PCI 卡/Hardnet

6.11.3 所支持的数据类型概述

引言

组态变量时，需要根据 AS 中的数据格式来定义数据类型和类型转换。

下表显示了通道支持的数据类型和类型转换的使用。

所支持的数据类型

数据类型	类型转换
二进制变量	否
有符号 8 位数	是
无符号 8 位数	是
有符号 16 位数	是
无符号 16 位数	是
有符号 32 位数	是
无符号 32 位数	是
浮点数 32 位 IEEE 754	是
文本变量, 8 位字符集	否
原始数据类型	否

可以在“通讯”一节中找到有关类型转换的附加信息。

6.11.4 组态通道

6.11.4.1 “SIMATIC S7 Protocol Suite”通道 - 组态

引言

本节说明如何组态“SIMATIC S7 Protocol Suite”通道。

1. 安装通道
2. 通道单元选择
3. 组态连接
4. 变量组态

系统参数组态

有关通道、连接和变量诊断的详细信息，请参见“通讯诊断”。

参见

通道单元的系统参数 (页 415)

组态变量 (页 409)

“SIMATIC S7 Protocol Suite”通道的通道单元 (页 393)

通道和变量的诊断 (页 563)

6.11.4.2 如何组态“SIMATIC S7 Protocol Suite”通道

简介

本节说明如何安装“SIMATIC S7 Protocol Suite”通道。

1. 安装通道
2. 通道单元选择
3. 创建连接
4. 插入一个变量
5. 在 WinCC 自定义安装中组态系统参数


先决条件:

- 通信模块为内置模块。
- 已经安装硬件驱动程序。
- 已创建与 AS 的电缆连接。

步骤

1. 在变量管理的导航区域中,从节点“变量管理”(Tag Management)的快捷菜单中选择条目“添加新的驱动程序”(Add new driver)。
2. 选择驱动程序“SIMATIC S7 Protocol Suite”。
创建通道。
通信驱动程序和相关联的通道单元显示在变量管理中。
3. 选择所需的通道单元,并在快捷菜单中选择条目“新建连接”(New Connection)。
4. 输入连接的名称。
5. 要为建立连接和连接状态创建系统变量,请从连接的快捷菜单中选择“创建变量以激活/禁用”(Create tags for activation/deactivation)条目。
在内部变量组“ConnectionStates”中创建下列变量:
 - @<Connectionname>@ForceConnectionStateEx
 - @<Connectionname>@ConnectionStateEx
6. 单击表格区下方的“变量”(Tags)选项卡。
7. 单击“名称”(Name)列最上方的空单元格。
输入变量的名称。

6.11 SIMATIC S7 Protocol Suite

8. 在“数据类型”(Data Type) 字段中选择所需的数据类型。
也可以在“属性”(Properties) 区域中为变量定义起始值和替换值。
如果需要一个详细说明来组态特定通道单元连接的变量，请关闭对话框并继续在所涉及的通道单元中查看主题“组态变量”。
9. 打开“地址属性”(Address properties) 对话框。
为此，单击“地址”(Address) 字段，然后单击  按钮。
10. 单击“确定”(OK) 按钮，关闭上述两个对话框。
11. 如果 WinCC 系统和通信硬件为非标准安装，则还需要将系统参数设置为非标准值。
关于此主题的其他信息请参见“系统参数”。

参见

如何离线下载 AS 符号 (页 384)

6.11.4.3 如何离线下载 AS 符号

简介

可离线组态以下 S7 通道：

- SIMATIC S7 Protocol Suite
- SIMATIC S7-1200, S7-1500 Channel

为此，可从现有 TIA Portal 项目中导出数据记录，并将导出文件加载至 WinCC 项目。

支持的导出格式

导入时支持以下文件格式：

格式	内容	说明
*.bin	二进制数据	从 WinCC 变量管理器导出： <ul style="list-style-type: none"> • “变量管理器”(Tag Management) 视图 > 连接的快捷菜单： AS 符号 > 保存到文件 (AS Symbols > Save to file) 不受“SIMATIC S7 Protocol Suite”通道支持。
*.sdz	结构化导出	从 WinCC 变量管理器导出： <ul style="list-style-type: none"> • “符号”(Symbols) 视图 > 菜单： 编辑 > 导出 (Edit > Export) 同样会从导航区导出结构信息。
*.zip	TIA Portal 导出文件	通过“SIEMENS SIMATIC SCADA Export”工具从 TIA Portal 导出

用于 TIA Portal 的“SIEMENS SIMATIC SCADA Export”

要从 TIA Portal 项目中导出数据记录，请使用“SIEMENS SIMATIC SCADA Export”工具。

在 TIA Portal 项目中，从 PLC 的快捷菜单中选择“导出到 SIMATIC SCADA”(Export to SIMATIC SCADA) 条目。

可以在工业在线支持中下载适用于各种 TIA Portal 版本的工具：

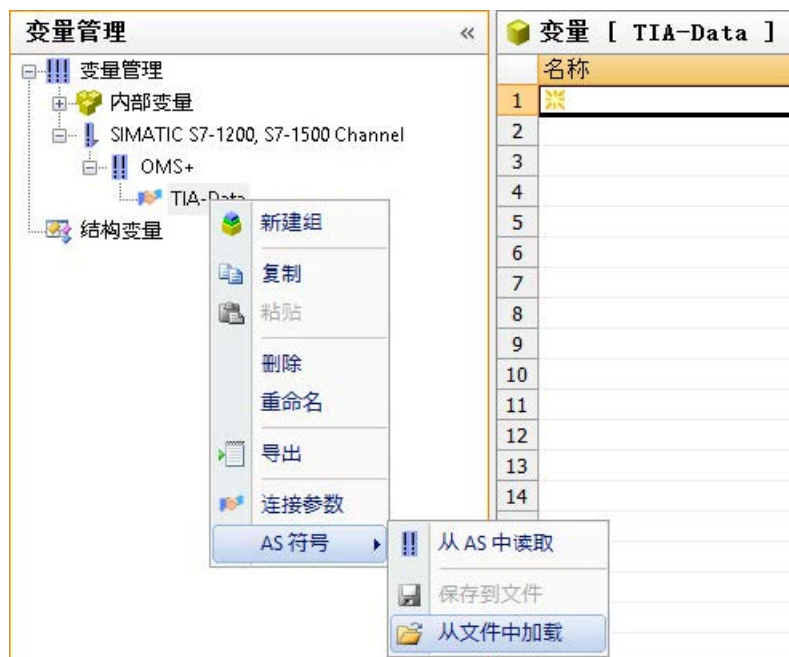
- “SIMATIC SCADA Export”下载 (ID 109748955) (<https://support.industry.siemens.com/cs/ww/zh/view/109748955>)
- “SIMATIC SCADA Export”文档 (ID 101908495) (<https://support.industry.siemens.com/cs/ww/zh/view/101908495>)

要求

- 已在 TIA Portal 中编译 AS。
- 相应的 PLC 组态数据已导出且可用，例如以 .zip 文件形式提供。
- 已在 WinCC 项目中安装了通信处理器和相关硬件驱动程序。
- 已在“SIMATIC S7-1200, S7-1500 Channel”或“SIMATIC S7 Protocol Suite”中创建了连接。
- 已打开“变量管理”编辑器。

操作步骤

1. 在连接快捷菜单中选择“AS 符号 > 从文件加载”(AS Symbols > Load from file)。



2. 选择要加载的所需数据记录。
加载可用的控制器数据。

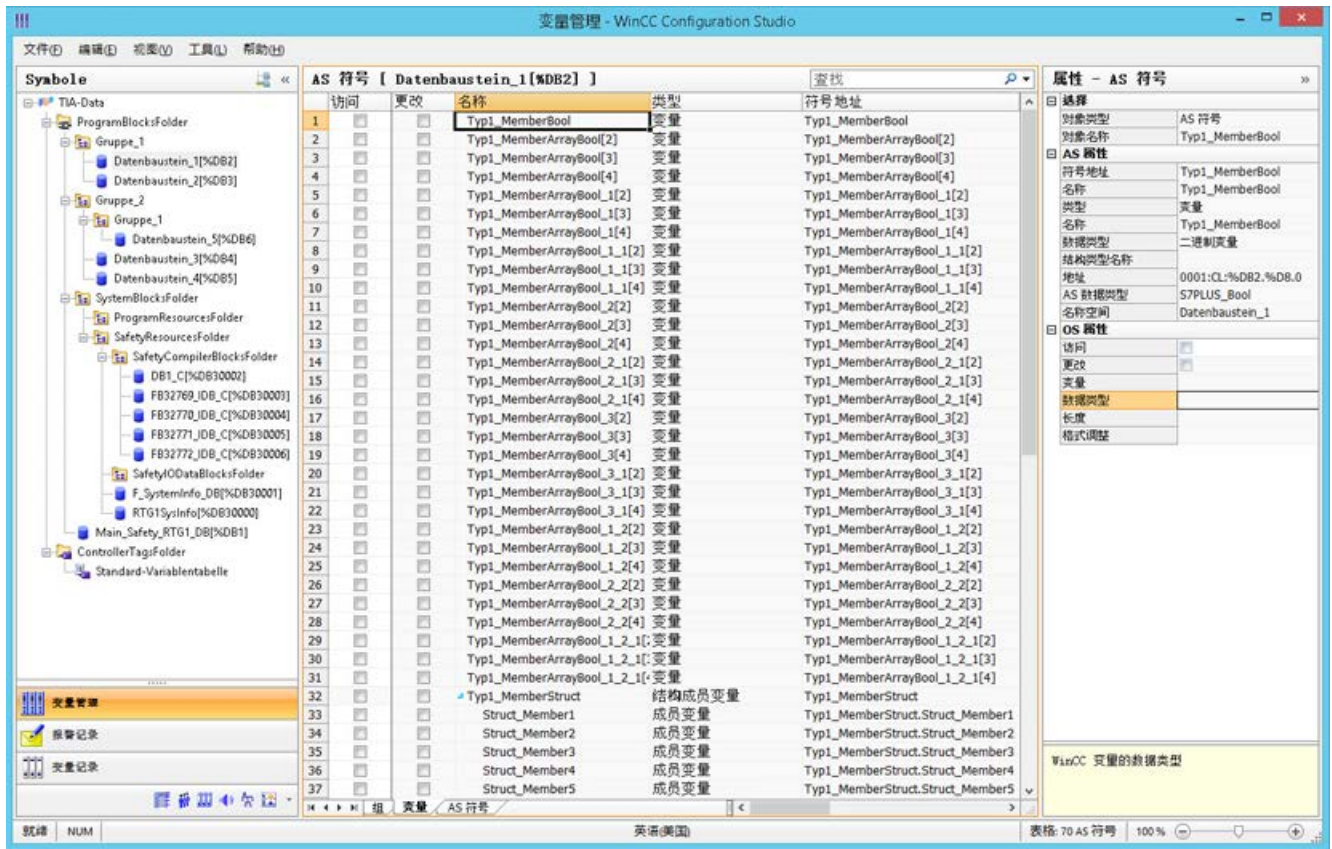
结果

已导入组态，“符号”(Symbols) 视图打开。

加载的数据显示在表格区的“AS 符号”(AS Symbols) 选项卡中，且可用于其它处理操作。

如果加载的数据还包含结构，还会额外显示“AS 结构”(AS structures) 选项卡。

编辑器关闭后，“AS 符号”(AS Symbols) 和“AS 结构”(AS structures) 选项卡会再次隐藏。



符号显示

在“变量管理器”(Tag Management)中，可使用以下按钮在默认视图和“符号”(Symbols)视图之间进行切换：

该按钮仅在加载数据记录后才可用。

导航区域

结构树中的数据表示对应于 TIA Portal 的层级。

表格区域

找到的 WinCC 变量与 AS 变量不匹配时，会自动选中“已修改”(Modified) 列的复选框。因此，可以按照这些内容进行过滤。

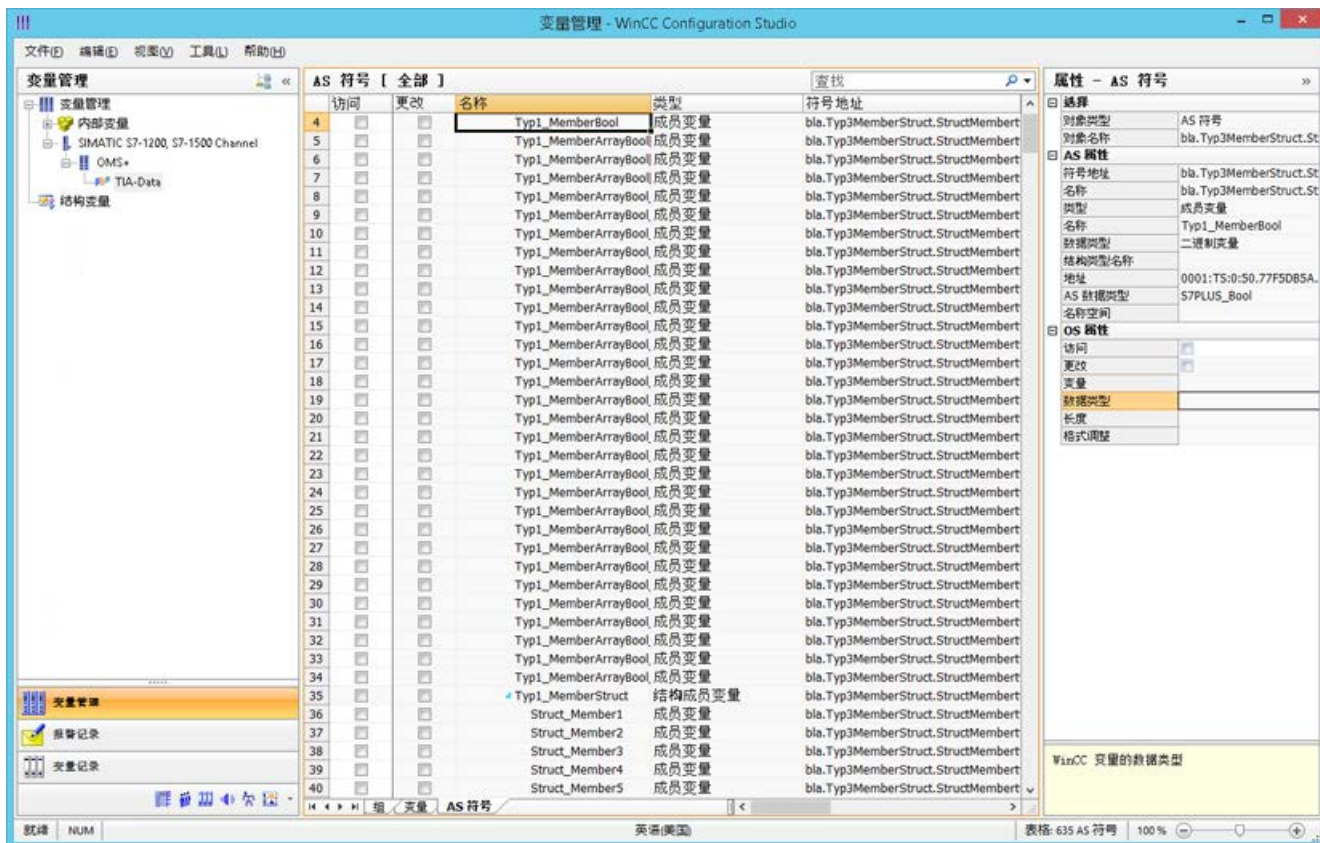
通过选中“访问”(Access) 列的复选框，可从找到的各个 AS 变量创建一个 WinCC 变量。

变量管理中的 AS 符号

还可通过“AS 符号”(AS Symbols) 选项卡访问变量管理中的 AS 符号。

6.11 SIMATIC S7 Protocol Suite

与数据块特定的“符号”(Symbols)视图不同，所有可用的控制器变量均会在此处显示。该视图还显示了之前已组态但现在不再位于 AS 中的变量。



参见

- 如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)
- 如何导出 AS 项目数据 (页 391)
- 如何组态 AS 结构 (页 389)
- “SIMATIC SCADA Export”文档 (ID 101908495) (<https://support.industry.siemens.com/cs/ww/zh/view/101908495>)
- “SIMATIC SCADA Export”下载 (ID 109748955) (<https://support.industry.siemens.com/cs/ww/zh/view/109748955>)

6.11.4.4 如何组态 AS 结构

简介

如果加载 AS 符号，还会导入控制系统的结构。

该过程取决于通信通道：

- SIMATIC S7 Protocol Suite:
 - 从文件加载
- SIMATIC S7-1200, S7-1500 Channel
 - 从文件加载
 - 从 AS 加载

变量管理器中的 AS 结构

AS 结构显示在默认视图以及“符号”(Symbols) 视图的“AS 结构”(AS structures) 选项卡中。

在 WinCC 中可如下使用 AS 结构：

- 为 AS 结构变量创建 WinCC 结构类型。
结构在 WinCC 变量管理器中的“结构变量”(Structure tags) 下创建为结构类型。
还会为每个包含的“变量类型成员”创建结构类型元素。
- 为 AS 结构变量分配 WinCC 结构类型。
然后，为每个“变量类型成员”选择已选结构类型的结构类型元素。

可在变量管理器中更改 WinCC 结构类型的名称和结构类型元素。AS 结构的分配会自动调整。

要求

- 可通过下列方法之一访问 PLC 的组态数据：
 - 在运行系统中建立与 PLC 的连接。
 - 导出的组态数据以压缩文件等形式提供。
- 已在“SIMATIC S7-1200, S7-1500 Channel”或“SIMATIC S7 Protocol Suite”中创建了连接。

操作步骤

1. 通过“从 AS 读取”(Read from AS) 或“从文件加载”(Load from file) 加载 AS 符号。
已加载的消息显示在“符号”(Symbols) 变量管理器视图中。
加载的结构显示在“AS 结构”(AS structures) 选项卡上。
从 AS 加载时，不会传送结构名称。ID 显示为“结构变量类型”的名称。

Symbole		AS 结构	
CPU1516_V2		WinCC 结构	名称
ProgramBlocksFolder		1	SimData1UDT 结构变量类型
01_DBs		2	LEN_UDT1 结构变量类型

2. 单击“AS 结构”(AS structures)。
要显示结构的元素，请单击结构名称前面的箭头。
3. 选择结构的整个行，并在快捷菜单中选择“创建结构”(Create structure) 条目。

AS 结构		
WinCC 结构	名称	类型
1	SimData1UDT	结构变量类型
2	BitData	变量类型成员
3	IntData	变量类型成员
4	RealData	变量类型成员
5	ByteData	变量类型成员
6	LEN_UDT1	结构变量类型
7		变量类型成员
8		结构变量类型
9		结构变量类型
10		
11		
12		
13		
14		
15		

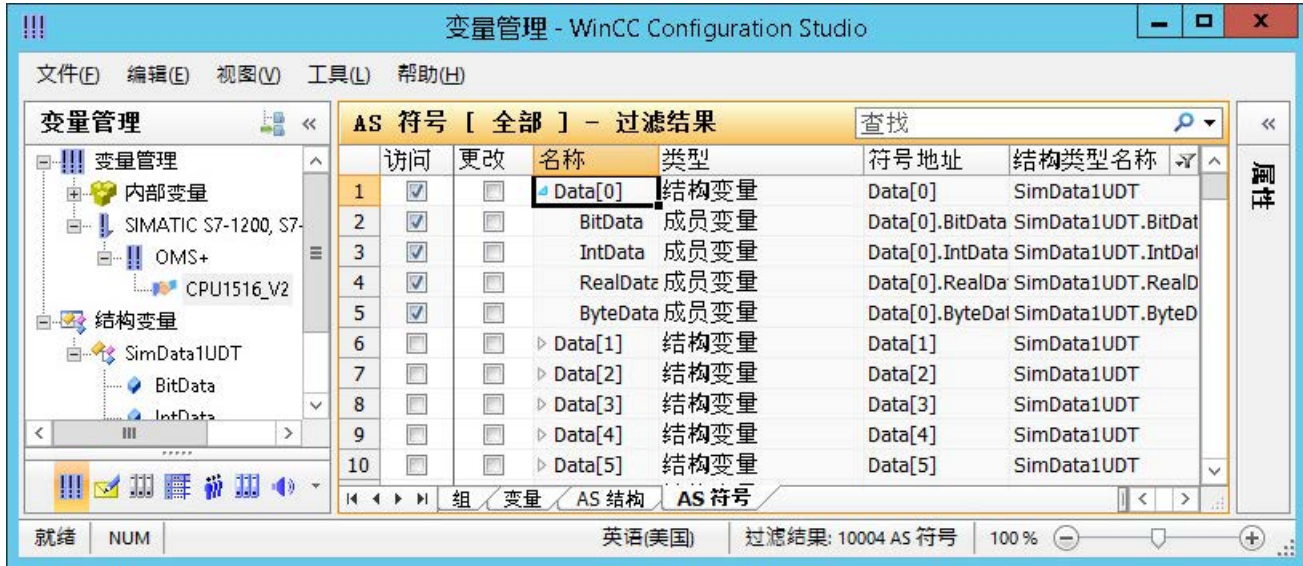
或者选择一个已在 WinCC 变量管理器中创建的结构类型。
然后将结构类型元素分配给“变量类型成员”。

AS 结构		
WinCC 结构	名称	类型
1	SimData1UDT	结构变量类型
2	SimData1UDT	
3	LEN_UDT1	
4	WinCC_StructType1	

会在 WinCC 变量管理器中为 AS 结构的每个“结构变量类型”创建结构类型。
会为每个“变量类型成员”创建结构类型元素。

4. 在“变量管理器”(Tag Management) 视图中选择“AS 符号”(AS Symbols) 选项卡。

5. 如果要只显示结构变量和成员变量，请在“结构类型名称”(Structure type name) 列中筛选出所需 AS 结构。



6. 要访问 WinCC 变量管理器中的 AS 结构变量，请激活“访问”(Access) 字段。
包含的成员变量会自动激活。
AS 结构变量将在 WinCC 变量管理器中创建为结构变量。

结果

通过 WinCC 变量管理器中的结构类型和结构变量，可访问 AS 结构变量。

举例来说，可通过这种方式访问 WinCC 面板类型中的 AS 结构，并在面板实例中表示这些结构。

参见

如何离线下载 AS 符号 (页 384)

如何导出 AS 项目数据 (页 391)

6.11.4.5 如何导出 AS 项目数据

导出 AS 符号

可使用导出文件进行离线组态。

可将 AS 项目数据导出为以下格式：

通信通道	导出的数据	导出文件的格式
SIMATIC S7-1200, S7-1500 Channel	AS 符号和 AS 结构	二进制数据: *.bin 结构化导出: *.sdz
SIMATIC S7 Protocol Suite	AS 符号和 AS 结构	结构化导出: *.sdz

要求

- 已在“SIMATIC S7-1200, S7-1500 Channel”或“SIMATIC S7 Protocol Suite”中创建了连接。
- 已加载 AS 项目数据并在 WinCC 中进行了组态。

步骤：导出二进制数据

1. 在变量管理器中选择连接。
2. 从快捷菜单中选择“AS 符号 > 保存到文件”(AS Symbols > Save to file) 条目。
将打开“导出”(Export) 对话框。
3. 选择存储路径并输入文件名。
单击“导出”(Export) 按钮关闭对话框。
组态数据以二进制数据集的形式导出到 .bin 文件中。

步骤：导出结构化数据

1. 在变量管理器中选择“符号”(Symbols) 视图。
2. 选择“编辑 > 导出”(Edit > Export) 菜单命令。
3. 选择存储路径并输入文件名。
单击“导出”(Export) 按钮关闭对话框。
组态数据导出到 *.sdz 文件中。
结构化导出还包含导航区中的结构信息。

参见

如何离线下载 AS 符号 (页 384)

如何组态 AS 结构 (页 389)

6.11.4.6 通道单元

“SIMATIC S7 Protocol Suite”通道的通道单元

引言

以下章节描述了如何组态通道单元以及相应的连接。在同一通道单元中可能存在多个连接。

参见

“TCP/IP”通道单元 (页 407)

“Soft PLC”通道单元 (页 405)

“Slot PLC”通道单元 (页 403)

通道单元“PROFIBUS (I + II)” (页 401)

“命名连接”通道单元 (页 398)

“MPI”通道单元 (页 396)

“工业以太网”+“工业以太网 (II)”通道单元 (页 393)

“工业以太网 (I+II)”通道单元

“工业以太网”+“工业以太网 (II)”通道单元

工作原理

通道单元“工业以太网”用于通过工业以太网将 WinCC 连接到 S7 自动化系统。可以通过通信模块 (CP) 进行通信, 例如, 通过 CP 343-1 实现自动化系统 S7-300 通信, 通过 CP 443-1 实现自动化系统 S7-400 通信。

在 WinCC 中可以使用不同的通信处理器, 例如 CP 1613 A2。可以通过“工业以太网 (II)”通道单元寻址第二个通信处理器。因为通信是通过“ISO”传输协议进行的, 因此不需在本地数据库中组态逻辑连接。

与这些通道单元有关的功能和组态是相同的。

单元典型术语

通信处理器

通信处理器 (CP) 是一个模块，通过该模块可在 WinCC 计算机与特定网络之间进行通信。

“ISO”传输协议

ISO 传输层是 ISO-OSI 参考模型中的一层，提供与数据传送相关的面向连接的服务。传输层处理数据流控制、阻塞和确认任务。

协议定义与物理线上内容相关的数据流的结构。它定义许多部分，包括操作模式、建立连接的过程、数据备份或传输速度。

工业以太网

工业以太网是工业环境中最有效的一种子网。它适用于工厂和车间一级，有利于大量成员在大范围内的大数据量的交换。

工业以太网是标准化的开放式通信网络，且符合 IEEE 802.3 标准。其主要优点不仅在于其高可靠性、使用范围广，而且速度快、易扩展和开放性。组态过程十分简便。

参见

组态变量 (页 409)

如何组态“工业以太网”通道单元连接 (页 394)

如何组态“工业以太网”通道单元连接

简介

除了通道单元以外，WinCC 也需要逻辑连接来与 PLC 进行通信。所有特定的参数在建立逻辑连接时定义。

在 S7 自动化系统中使用通信模块进行通信，例如在 S7-300 中使用 CP 343-1，或者在 S7-400 中使用 CP 443-1。

在 WinCC 中使用通信处理器，例如 CP 1613 A2。可以通过“工业以太网 II”通道单元寻址/增加第二个通信处理器。

有关通道、连接和变量诊断的详细信息，请参见“通信诊断”。

说明

S7-300/S7-400: CPU 的机架/插槽号

当使用带外部通信模块的 S7-300 或 S7-400 时，必须输入 CPU 的机架/插槽号。
如果输错了机架或插槽号，将不会建立通信链路。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。

步骤

1. 在通道单元“工业以太网”(Industrial Ethernet) 的快捷菜单中选择“新建连接”(New Connection) 条目。
将创建一个新连接。
2. 输入一个连接名称，例如“Test_Ind_Eth”。
3. 从快捷菜单中选择“连接参数”(Connection parameters) 连接。
将打开“连接参数 - 工业以太网”对话框。



4. 在“以太网地址”域中输入总线上自动化系统的站地址。
5. 在“机架号”域中输入要寻址 CPU 的机架号。
6. 必须将指定机架中的 CPU 插槽号输入到相应的“插槽号”域中。
7. 如果想要通过连接使用 BSEND/BRCV 传送数据块，激活复选框“发送/接收原始数据块”。
如果激活了复选框，则可以编辑“连接资源”域。
为连接源输入十六进制数值。
在 PLC 中组态连接时，将由 STEP7 分配该连接资源。
8. 单击“确定”(OK) 按钮，关闭上述两个对话框。

参见

组态变量 (页 409)

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

通道和变量的诊断 (页 563)

“MPI”通道单元

“MPI”通道单元

工作原理

“MPI”通道单元用于通过 MPI 将 WinCC 连接到 SIMATIC S7-300 和 S7-400 自动化系统。

在 WinCC 中，这可以通过下列设备完成

- 编程设备（例如 PG 760/PC RI45）的内部 MPI 接口
- 通信处理器，例如 CP 5613 A3（PCI 卡）。

所谓的 MPI 模块（ISA 卡）也适用 - 有这种模块，但不再可用。它已被通信处理器取代。

在 AS 中，通过 CPU 的 MPI 接口或相应的通信模块进行连接。

单元典型术语

MPI

MPI 表示多点接口，是一种可以有多个节点的通信连接。与通信网络的连接方式如下：

- 在 AS 中，使用 CPU 的 MPI 接口或通信模块，
- 在 WinCC 中，通过内置 MPI 接口，例如编程设备的接口或使用通信处理器（网卡）。

通信处理器

通信处理器 (CP) 是一个模块，通过该模块可在 WinCC 计算机与特定网络之间进行通信。

参见

组态变量 (页 409)

如何组态“MPI”通道单元连接 (页 397)

如何组态“MPI”通道单元连接

简介

除了通道单元以外，WinCC 也需要逻辑连接来与 PLC 进行通信。所有特定的参数在建立逻辑连接时定义。

S7-300 和 S7-400 PLC 使用内部 MPI 接口或通信模块，例如 CP 342-5（SIMATIC S7-300）或 CP 443-5（SIMATIC S7-400）。

如果 PG 760/PC RI45 上安装了 WinCC，则可使用内部 MPI 接口；否则需要内置的 MPI 模块。或者，还可以使用通信模块。

有关通道、连接和变量诊断的详细信息，请参见“通信诊断”。

说明

S7-300/S7-400：CPU 的机架/插槽号

当使用带外部通信处理器的 S7-300 或 S7-400 时，必须输入 CPU 的机架/插槽号。

如果输入的机架号或插槽号不正确，则不会建立通信连接。

使用 S7-300 时，对于通过 CPU 内部 MPI 接口的链接，必须给定机架/插槽号 = 0。

要求

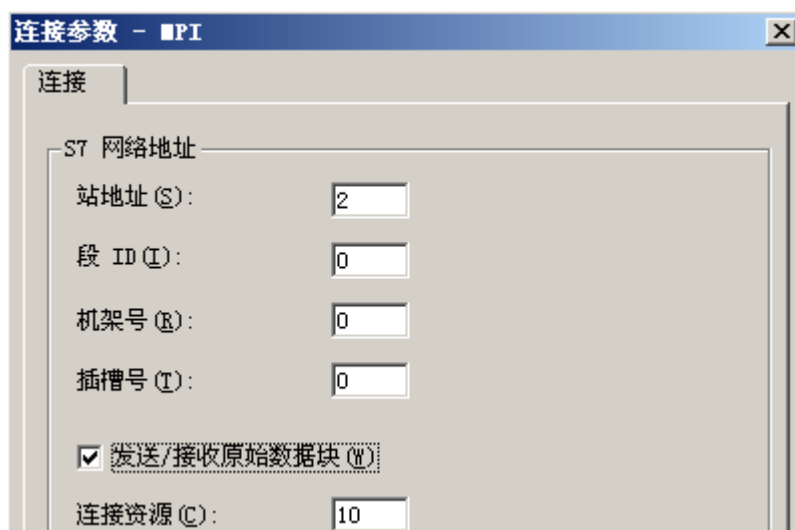
- 必须将“SIMATIC S7 Protocol Suite”（通信）驱动程序集成到项目中。

步骤

1. 在通道单元“MPI”的快捷菜单中选择“新建连接”(New Connection) 条目。
将创建一个新连接。
2. 输入连接名称“Test_MPI”。

6.11 SIMATIC S7 Protocol Suite

3. 从快捷菜单中选择“连接参数”(Connection parameters) 连接。
将打开“连接参数 - MPI”对话框。



4. 在“站地址”域中输入总线上相应域中自动化系统的站地址。
5. 当前不支持“段 ID”域。数值必须保持为“0”。
6. 在“机架号”域中输入要寻址 CPU 的机架号。
7. 输入指定机架上 CPU 的“插槽号”。
8. 如果想要通过连接使用 BSEND/BRCV 传送数据块，激活复选框“发送/接收原始数据块”。
如果选中该复选框，“连接资源”字段也将激活。
为连接源输入十六进制数值。
在 PLC 中组态连接时，将由 STEP7 分配该连接资源。
9. 单击“确定”(OK) 按钮，关闭上述两个对话框。

参见

组态变量 (页 409)

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

“命名连接”通道单元

“命名连接”通道单元

工作原理

该通道单元用于设置使用 STEP 7 组态的符号连接。从而 WinCC 可以通过符号连接名称访问冗余和非冗余连接。这些符号连接与 S7-400 PLC 系列 H/F 冗余系统一起，对于保证可靠性通信是必需的。

符号连接名称在 STEP 7 中使用 NETPRO 应用程序组态。连接名称、连接参数和应用程序名称存储在数据库 (*.XDB) 中。该数据库由相应 WinCC 项目目录中的 PLC/OS 工程工具“Mapper”自动存储；但是，也可以在此目录外复制该数据库，例如在不使用“Mapper”时。

说明

WinCC 系统中的每位通信成员只能有一个 XDB 文件。
因此，不能在多台 WinCC 计算机上复制和使用 XDB 文件。

可使用下列选项在 WinCC 中激活此数据库：

- 如果 XDB 文件位于项目目录外（例如由于未使用 Mapper 工具），则在启动 WinCC 前，需要在“设置 PG/PC 接口”的“STEP 7 组态”标签（控制面板）中输入 XDB 文件的路径和名称。
启动 WinCC 时，如果项目目录中没有任何文件，将从外部目录中读取该 XDB 文件。当多个项目要使用同一个中央保存的数据库时，该过程十分有用。
- 如果使用了 Mapper 工具，将自动把 XDB 文件复制到 WinCC 项目目录中。启动 WinCC 并打开项目时，将从 S7 通道读取数据，并将数据输入 Windows 的注册数据库。

然后，可通过将一个符号连接名称指定给所选的应用程序名称，以在 WinCC 中组态连接。

说明

也可以手动输入应用程序名称和连接名称。
必须检查 STEP 7 中组态的名称拼写是否正确，因为在 CS 模式中不检查名称。
例如，在下列情况下可能需要进行该操作：

- 没有 XDB 文件可用于符号连接名称。在这种情况下，直接将组态传送到“组件组态器”。
 - 该项目将传送到其它计算机上。
-

典型单元术语

通信处理器

通信处理器 (CP) 是支持 PLC 和指定网络之间通信的模块。

参见

组态变量 (页 409)

如何组态“命名连接”通道单元连接 (页 400)

如何组态“命名连接”通道单元连接

简介

除通道单元外，WinCC 还需要逻辑连接，以便通过符号连接与 S7-400 PLC 进行通信。

设置逻辑连接时，“连接名称”域中列出的某个符号连接名称将分配给所选的应用程序名称。

符号连接名称和应用程序名称在 STEP 7 中组态。

有关通道、连接和变量诊断的详细信息，请参见“通信诊断”。

说明

也可以手动输入应用程序名称和连接名称。

必须检查 STEP 7 中组态的名称拼写是否正确，因为在 CS 模式中不检查名称。

例如，在下列情况下可能需要进行该操作：

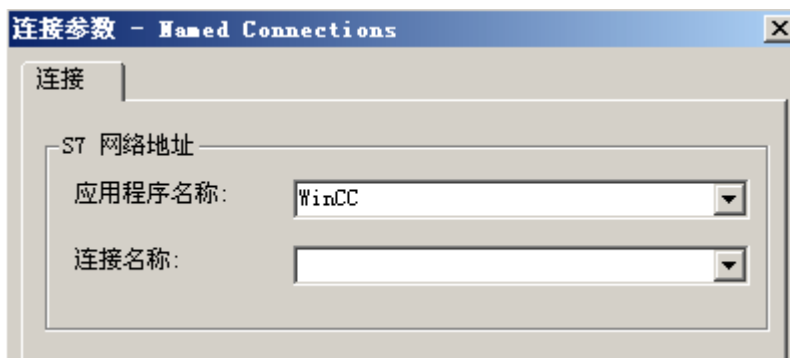
- 没有 XDB 文件可用于符号连接名称。在这种情况下，直接将组态传送到“组件组态器”。
- 该项目将传送到其它计算机上。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。

步骤

1. 在通道单元“已命名连接”(Named Connections) 的快捷菜单中选择“新建连接”(New Connection) 条目。
将创建一个新连接。
2. 输入一个连接名称，例如“Test_NC”。
3. 在连接的快捷菜单中选择“连接参数”(Connection parameters)。
将打开“连接参数 - 已命名连接”(Connection parameters - Named Connections) 对话框。



4. 在“应用程序名称”(Application name) 字段中，输入 STEP 7 中组态的应用程序名称。默认值为 WinCC。

5. 在“连接名称”域中，输入 STEP 7 中组态的符号连接名称。
6. 单击“确定”(OK) 按钮，关闭上述两个对话框。

参见

组态变量 (页 409)

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

“PROFIBUS (I+II)”通道单元

通道单元“PROFIBUS (I + II)”

工作原理

通道单元用于通过 PROFIBUS 网络将 WinCC 连接到 SIMATIC S7-300 和 S7-400 自动化系统。

在 S7 自动化系统中使用通信模块，例如在 S7-300 中使用 CP 342-5，或者在 S7-400 中使用 CP 443-5。

在 WinCC 中使用通信处理器，例如 CP 5613 A3。

可以通过“PROFIBUS II”通道单元连接第二个通信处理器。因此，可增大最大连接数。

单元典型术语

PROFIBUS

PROFIBUS 是一种开放式、与供应商无关的通信系统，适用于车间和现场级别，最多可有 127 个节点。PROFIBUS 基于欧洲标准 EN 50170，卷 2，PROFIBUS。PROFIBUS 使用以较低级别主-从令牌传递作为访问方法。

通信处理器

通信处理器 (CP) 是一个模块，通过该模块可在 WinCC 计算机与特定网络之间进行通信。

参见

组态变量 (页 409)

如何组态“PROFIBUS”通道单元连接 (页 402)

如何组态“PROFIBUS”通道单元连接

简介

除了通道单元，WinCC 还必须具有逻辑连接来与 PLC 进行通信。所有特定的参数在建立逻辑连接时定义。

S7 自动化系统将使用通信模块，例如，在 S7-300 中使用 CP 342-5，或者在 S7-400 中使用 CP 443-5。

在 WinCC 中使用通信处理器，例如 CP 5613 A3。可以通过“PROFIBUS II”通道单元连接第二个通信处理器。

有关通道、连接和变量诊断的详细信息，请参见“通信诊断”。

说明

断开状态中的连接

启动 PROFIBUS 通信时，如果在 WinCC 计算机打开时将通信处理器连接到 PROFIBUS，可能产生 PROFIBUS 错误。

因此，建议计算机连接到 PROFIBUS 之前将其关闭。

否则，（按照 PROFIBUS 标准）总线上可能会产生多个令牌，这些令牌将引起总线错误。

S7-300/S7-400：CPU 的机架/插槽号

当使用带外部通信模块的 S7-300 或 S7-400 时，必须输入 CPU 的机架/插槽号。

如果输错了机架或插槽号，将不会建立通信链路。

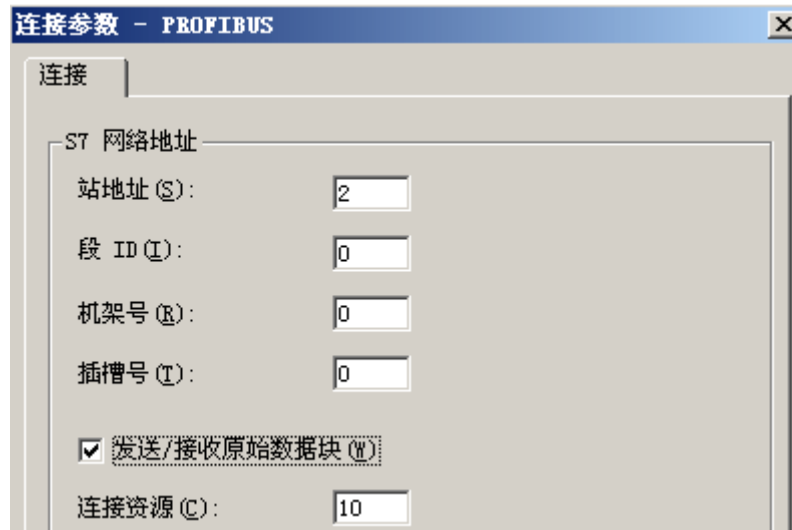
要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。

步骤

1. 在通道单元“PROFIBUS”的快捷菜单中选择“新建连接”(New Connection) 条目。
将创建一个新连接。
2. 输入连接名称“Test_PROFIBUS”。

3. 从快捷菜单中选择“连接参数”(Connection parameters) 连接。
将打开“连接参数 - PROFIBUS”对话框。



4. 在相应“站地址”域输入总线上自动化系统的站地址。
5. 当前不支持“段 ID”域。数值必须保持为“0”。
6. 输入要寻址的 CPU 其中的“机架号”。
7. 输入指定机架上 CPU 的“插槽号”。
8. 如果想要通过连接使用 BSEND/BRCV 传送数据块，激活复选框“发送/接收原始数据块”。
如果选中该复选框，“连接资源”字段也将激活。
为连接源输入十六进制数值。
在 PLC 中组态连接时，将由 STEP7 分配该连接资源。
9. 单击“确定”(OK) 按钮，关闭上述两个对话框。

参见

组态变量 (页 409)

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

“Slot PLC”通道单元

“Slot PLC”通道单元

工作原理

“Slot PLC”通道单元用于实现 WinCC 和安装在 WinCC 计算机中的最多 4 个 Slot PLC (WinAC Pro) 之间的通讯。因为 Slot PLC 具有集成接口，因此 WinCC 和 Slot PLC 之间的连接不需要任何附加的通讯硬件。

参见

组态变量 (页 409)

如何组态“Slot PLC”通道单元连接 (页 404)

如何组态“Slot PLC”通道单元连接

简介

为了与所安装的 SPS 卡进行通信，除通道单元外，WinCC 还需要一个逻辑连接。所有特定的参数在建立逻辑连接时定义。

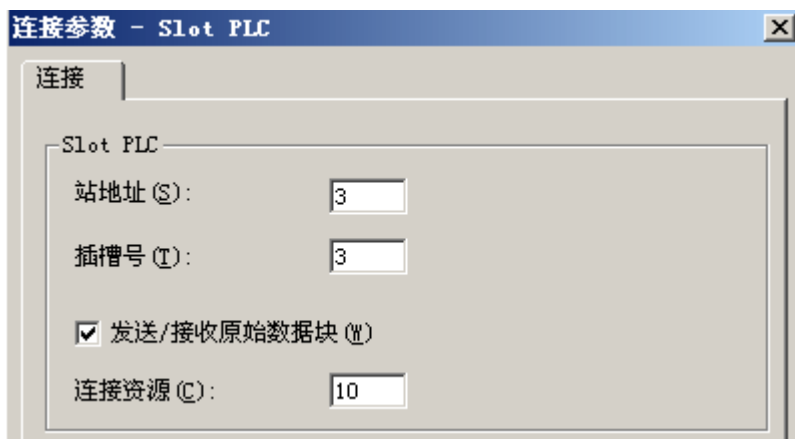
有关通道、连接和变量诊断的详细信息，请参见“通信诊断”。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 如果要组态多个 Slot PLC，则需要使用 Slot PLC V3.4 版本。

步骤

1. 在通道单元“插槽 PLC”(Slot PLC) 的快捷菜单中选择“新建连接”(New connection) 条目。将创建一个新连接。
2. 输入一个连接名称，例如“Test_SPLC”。
3. 从快捷菜单选择“连接参数”(Connection parameters) 命令。将打开“连接参数 - 插槽 PLC”(Connection parameters - Slot PLC) 对话框。



4. 在“站地址”域中，输入 Soft K 总线上 Slot PLC 的站地址。
5. 在“插槽号”域中，输入 Slot PLC 安装在其中的插槽的编号。
6. 如果想要通过连接使用 BSEND/BRCV 传送数据块，激活复选框“发送/接收原始数据块”。

7. 如果选中该复选框，“连接资源”字段也将激活。为连接源输入十六进制数值。在 PLC 内组态连接时，将由 STEP7 分配该连接资源。
8. 单击“确定”(OK) 按钮，关闭上述两个对话框。

说明

所安装的多个 Slot PLC 的“站地址”和“插槽号”连接参数必须完全相同，且“插槽号”从“3”开始编号。

参见

组态变量 (页 409)

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

“Soft PLC”通道单元

“Soft PLC”通道单元

工作原理

“Slot PLC”通道单元用于实现 WinCC 和安装在 WinCC 计算机中的一个 Soft PLC (WinAC Pro) 之间的通讯。对于 WinCC 和 Soft PLC 之间的连接，不需要任何附加的通讯硬件。

参见

组态变量 (页 409)

如何组态“Soft PLC”通道单元上的连接 (页 405)

如何组态“Soft PLC”通道单元上的连接

简介

除了通道单元，WinCC 还必须具有逻辑连接来与 Soft PLC 进行通信。所有特定的参数在建立逻辑连接时定义。

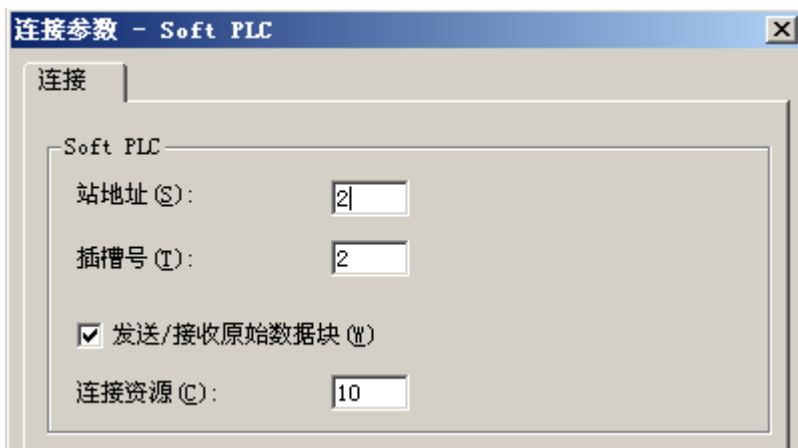
有关通道、连接和变量诊断的详细信息，请参见“通信诊断”。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。

步骤

1. 在通道单元“Soft PLC”的快捷菜单中选择“新建连接”(New connection) 条目。
将创建一个新连接。
2. 输入一个连接名称，例如“Test_SOFTPLC”。
3. 从快捷菜单选择“连接参数”(Connection parameters) 命令。
将打开“连接参数 - 插槽 PLC”(Connection parameters - Slot PLC) 对话框。



4. 在“站地址”域中，输入 Soft K 总线上 Soft PLC 的站地址。
5. 在域“插槽号”中，输入插槽的编号。插槽号在 Soft PLC 的硬件配置期间组态；并且在相同 WinCC 计算机中使用多个 Soft PLC 时需要插槽号。
6. 如果想要通过连接使用 BSEND/BRCV 传送数据块，激活复选框“发送/接收原始数据块”。
7. 如果选中该复选框，“连接资源”字段也将激活。为连接源输入十六进制数值。在 PLC 内组态连接时，将由 STEP7 分配该连接资源。
8. 单击“确定”(OK) 按钮，关闭上述两个对话框。

参见

组态变量 (页 409)

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

“TCP/IP”通道单元

“TCP/IP”通道单元

工作原理

通道单元“TCP/IP”使用“ISO-on-TCP 传输”协议通过工业以太网将 WinCC 连接到自动化系统 SIMATIC S7-300 和 S7-400。

协议对应于标准 TCP/IP 扩展协议 RFC 1006。该扩展是必需的，因为 TCP/IP 使用其中没有任何数据阻塞的通信。

通信如果是自动化系统 S7-300，通信通过通信模块 CP 343-1 TCP 等进行，如果是 S7-400，通过 CP 443-1 TCP 或 CP 443-1 IT 进行。

在 WinCC 中使用通信处理器，例如 CP 1613 A2。

因为通信通过 ISO-on-TCP 传输协议进行，因此不必在本地数据库中组态逻辑连接。

单元典型术语

通信处理器

通信处理器 (CP) 是一个模块，通过该模块可在 WinCC 计算机与特定网络之间进行通信。

ISO 传送协议

ISO 传输层是 ISO-OSI 参考模型中的一层，提供与数据传送相关的面向连接的服务。传输层处理数据流控制、阻塞和确认任务。

协议定义与物理线上内容相关的数据流的结构。它定义许多部分，包括操作模式、建立连接的过程、数据备份或传输速度。

工业以太网

工业以太网是工业环境中最有效的一种子网。它适用于工厂和车间一级，有利于大量成员在大范围内的大数据量的交换。

工业以太网是标准化的开放式通信网络，且符合 IEEE 802.3 标准。其主要优点不仅在于其高可靠性、使用范围广，而且速度快、易扩展和开放性。组态过程十分简便。

参见

组态变量 (页 409)

如何组态“TCP/IP”通道单元连接 (页 408)

如何组态“TCP/IP”通道单元连接

简介

除了通道单元以外，WinCC 也需要逻辑连接来与 PLC 进行通信。所有特定的参数在建立逻辑连接时定义。

如果是自动化系统 S7-300，通信通过通信模块 CP 343-1 TCP 等进行，如果是 S7-400，通过 CP 443-1 TCP 或 CP 443-1 IT 进行。

在 WinCC 中使用通信处理器，例如 CP 1613 A2。

有关通道、连接和变量诊断的详细信息，请参见“通信诊断”。

说明

S7-300/S7-400: CPU 的机架/插槽号

当使用带外部通信模块的 S7-300 或 S7-400 时，必须输入 CPU 的机架/插槽号。如果输错了机架或插槽号，将不会建立通信链路。

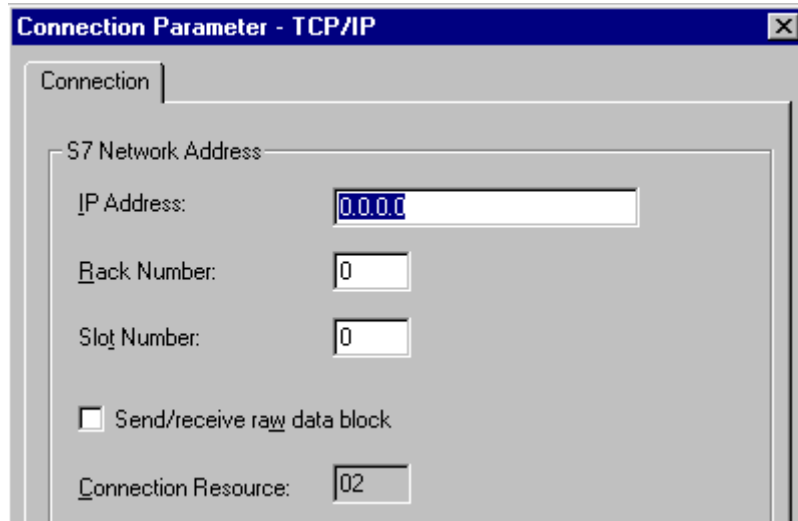
要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。

步骤

1. 在通道单元“TCP/IP”的快捷菜单中选择条目“新建连接”(New Connection)。将创建一个新连接。
2. 输入连接名称“Test_TCP”。

3. 从快捷菜单中选择“连接参数”(Connection parameters) 连接。
将打开“连接参数 - TCP/IP”(Connection parameters - TCP/IP) 对话框。



4. 在“IP 地址”域中输入总线上自动化系统的 Internet 协议地址。
5. 在“机架号”域中输入要寻址 CPU 的机架号。
6. 必须将指定机架中的 CPU 插槽号输入到相应的“插槽号”域中。
7. 如果想要通过连接使用 BSEND/BRCV 传送数据块，激活复选框“发送/接收原始数据块”。
如果选中该复选框，“连接资源”字段也将激活。
为连接源输入十六进制数值。
在 PLC 中组态连接时，将由 STEP7 分配该连接资源。
8. 单击“确定”(OK) 按钮，关闭上述两个对话框。

参见

组态变量 (页 409)

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

6.11.4.7 组态变量

组态变量

引言

以下章节描述如何组态变量。根据对 PLC 中数据区域的访问类型和 WinCC 变量的数据类型，组态有所不同。

有关通道、连接和变量诊断的详细信息，请参见“通讯诊断”。

参见

如何组态文本变量 (页 414)

如何组态按字访问的变量 (页 412)

如何组态按字节访问的变量 (页 411)

如何组态按位访问的变量 (页 410)

如何组态按位访问的变量

简介


本节说明如何以按位访问方式组态 PLC 中地址区域的变量。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须在通道单元（例如“工业以太网”）中创建连接（例如“Test_Ind_Eth”）。

步骤

1. 选择连接“Test_Ind_Eth”。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
在“名称”(Name) 字段中输入“ETH_Var1_bit”作为变量名称。
4. 在“数据类型”(Data Type) 字段中，设置“二进制变量”(Binary tag) 数据类型。

- 打开“地址属性”(Address properties)对话框。
为此,请单击“地址”(Address)字段,然后单击  按钮。
在“数据区”(Data area)字段中,设置数据所在的自动化系统的数据区。如果选择“DB”作为数据区,则在已启用的“DB号”(DB No.)字段中输入数据块的编号。



- “寻址”域中的条目“位”无法改变,因为这已经由 WinCC 变量的数据类型“二进制变量”定义。
- 在其下面的两个域中输入字节地址和位地址。左侧的域标签将取决于“数据区”域中的条目,例如:如果数据区为“DB”且类型域为“二进制变量”,则此域的标签显示为“D”。
- 如果变量具有要在 WinCC 中使用的质量代码,则选中复选框“质量代码”(quality code)。因此,代码还必须存在于 PLC 中。只有在将数据区选作“DB”时,才能启用此复选框。
- 单击“确定”(OK)按钮,关闭上述两个对话框。

参见

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

如何组态按字节访问的变量


简介

本节说明如何以按字节访问方式组态 PLC 中地址区域的变量。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须在通道单元(例如“工业以太网”)中创建连接(例如“Test_Ind_Eth”)。

步骤

1. 选择连接“Test_Ind_Eth”。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
在“名称”(Name) 字段中输入“ETH_Var1_byte”作为变量名称。
4. 在“数据类型”(Data Type) 字段中, 将数据类型设置为“无符号 8 位数”(Unsigned 8-bit value)。
5. 打开“地址属性”(Address properties) 对话框。
为此, 请单击“地址”(Address) 字段, 然后单击  按钮。
在“数据区”(Data area) 字段中, 设置数据所在的自动化系统的数据区。如果选择“DB”作为数据区, 则在已启用的“DB 号”(DB No.) 字段中输入数据块的编号。



6. “寻址” 字段中的条目“位”无法改变, 因为这已经由 WinCC 变量的数据类型“无符号 8 位数”定义。
7. 在下面的字段中输入字节地址。左侧的字段标签将取决于“数据区”(Data Area) 字段中的条目, 例如: 如果数据区为“DB”且类型为“无符号 8 位数”, 则此字段的标签显示为“D”。
8. 如果变量具有要在 WinCC 中使用的质量代码, 则选中复选框“质量代码”(quality code)。因此, 代码还必须存在于 PLC 中。只有在将数据区选作“DB”时, 才能启用此复选框。
9. 单击“确定”(OK) 按钮, 关闭上述两个对话框。

参见

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

如何组态按字访问的变量

简介


本节说明如何以按字访问方式组态 PLC 中地址区域的变量。

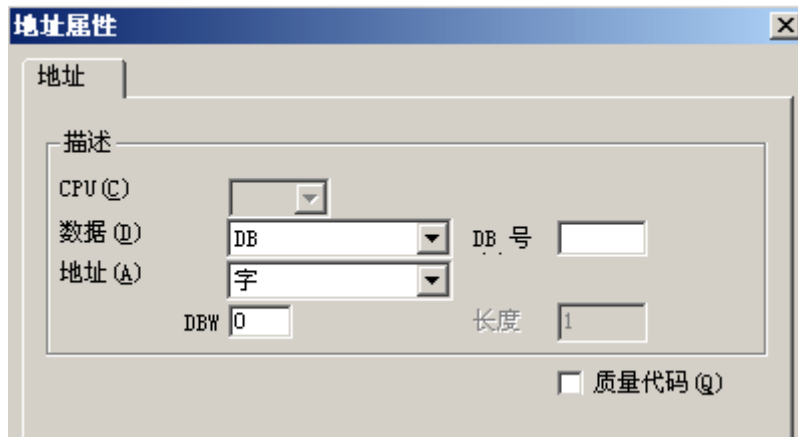
该过程也适用于长度为 4 个字节 (“双字”) 和更长的变量。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须在通道单元（例如“工业以太网”）中创建连接（例如“Test_Ind_Eth”）。

步骤

1. 选择连接“Test_Ind_Eth”。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
在“名称”(Name) 字段中输入“ETH_Var3_word”作为变量名称。
4. 在“数据类型”(Data Type) 字段中，将数据类型设置为“无符号 16 位数”(Unsigned 16-bit value)。
5. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
在“数据区”(Data area) 字段中，设置数据所在的自动化系统的数据区。如果选择“DB”作为数据区，则在已启用的“DB 号”(DB No.) 字段中输入数据块的编号。



6. “寻址” 字段中的条目“字”无法改变，因为这已经由 WinCC 变量的数据类型“无符号 16 位数”定义。
7. 在“寻址” 域中，输入地址的数字值。左侧的字段标签将取决于“数据区”(Data Area) 字段中的条目，例如：如果类型为“无符号 16 位数”，则此字段的标签显示为“DBW”。
8. 如果变量具有要在 WinCC 中使用的质量代码，则选中复选框“质量代码”(quality code)。因此，代码还必须存在于 PLC 中。只有在将数据区选作“DB”时，才能启用此复选框。
9. 单击“确定”(OK) 关闭所有打开的对话框。

参见

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

如何组态文本变量

简介

本节说明如何组态文本变量。

WinCC 仅支持 S7 字符串类型，它由一个控制字和实际用户字符串的数据组成，用于 SIMATIC S7 Protocol Suite 通道中的文本变量：


- 要在 WinCC 中组态文本变量，需在用户数据之前输入 PLC 内存中的控制字的地址。控制字的第一个字节包含字符串自定义的最大长度，第二个字节包含实际长度。
- 当在 PLC 内存中插入数据结构时，必须注意 WinCC 中为文本变量组态的长度要增加控制字的 2 个字节。如果文本变量数据结构在内存中一个接一个地直接插入，则后面的数据将会被覆盖。
- PCS7 版本从 V4.01 更新为 V5.0 SP1 时，必须重新映射。因为在 V5.0 以前的版本中，组态文本变量时也涉及用户数据的地址，而从版本 V5.0 开始，必须输入控制字的地址。
- 在进行读操作时，控制字将和用户数据一起被读取，并将判断第二个字节中的当前长度。只有长度与第二控制字节中包含的当前长度一致的用户数据才传送到 WinCC 文本变量。
- 在进行写操作时，将确定字符串的实际长度（“0”字符），且具有当前长度的控制字节将和用户数据一起发送给 PLC。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须在通道单元（例如“工业以太网”）中创建连接（例如“Test_Ind_Eth”）。

步骤

1. 选择连接“Test_Ind_Eth”。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
在“名称”(Name) 字段中输入“ETH_Var3_Text”作为变量名称。
4. 在“数据类型”域中，指定数据类型“文本变量，8 位字符集”。

- 打开“地址属性”(Address properties)对话框。
为此,请单击“地址”(Address)字段,然后单击  按钮。
在“数据区”(Data area)中,设置PCL中数据所处的数据区。如果选择“DB”作为数据区,则在已启用的“DB号”(DB No.)字段中输入数据块的编号。



- “寻址”域中的条目只能设定为“字节”或“字”,因为这些条目由 WinCC 变量的数据类型“文本变量, 8 位字符集”定义。
- 在“寻址”域中,输入地址的数字值。涉及控制字的地址。左侧的域标签将取决于“数据区”域中的条目,例如:如果类型域为“字”,则此域的标签显示为“DBW”。
- 如果变量具有要在 WinCC 中使用的质量代码,则选中复选框“质量代码”(quality code)。因此,代码还必须存在于 PLC 中。只有在将数据区选作“DB”时,才能启用此复选框。
- 单击“确定”(OK)关闭所有打开的对话框。

参见

如何组态“SIMATIC S7 Protocol Suite”通道 (页 383)

6.11.4.8 系统参数

通道单元的系统参数

引言

如果需要与 WinCC 标准设置不一样的组态,则可以使用通道单元的“系统参数”对话框进行所需要的全部改变。

下列内容可修改:

- 逻辑设备名称
- AS 中周期性读取服务的使用

逻辑设备名称

WinCC 和自动化系统之间的通讯通过逻辑设备名称执行。设备名称是在安装通讯卡时分配的，并且是单元特定的条目。该域将由缺省的单元特定条目来填充，例如，如果是通道单元“MPI”，则为“MPI”。

使用 PLC 中的周期性读取服务

可以指定是否应使用 S7-PLC 的周期性读取服务（也称为周期性变量服务）。这些周期性读取服务对即将周期性读取到单个请求中的变量进行编组，并将其传送给 PLC。PLC 一旦收到请求，将立即传送所需数据，并且每当到达周期时间时都要传送数据。

当激活周期性读取服务时，也可使用变化传送。只有当数值改变后，才可传送数据。该功能必须由相应的自动化系统支持。

说明

SIMATIC S7 和单元标签上的系统参数都是单元特定的数据，因此可以为通道中的每个通道单元单独设置。

参见

如何更改逻辑设备名称 (页 419)

如何组态系统参数 (页 417)

PLC 中的周期性读取服务 (页 416)

PLC 中的周期性读取服务

引言

在通道“SIMATIC S7 Protocol Suite”的系统参数中，能够指定是否使用 S7-AS 的周期性读取服务（也称为周期性变量服务）。这些周期性读取服务对即将周期性读取到单个请求中的变量进行编组，并将其传送给 PLC。PLC 一旦收到请求，将立即传送所需数据，并且每当到达周期时间时都要传送数据。当不再需要请求的数据时，例如画面切换时，WinCC 将删除 PLC 中的周期性读取服务。

正常情况下，应该使用 PLC 中的周期性读取服务。因此，已经激活通道单元系统参数中相应的复选框（缺省设置）。只有不想使用周期性服务时，才修改该设置。

只有当激活周期性读取服务时，才可使用变化传送。当数值改变时，数据才能从 AS 中传送且每个自动化系统周期只传送一次。该功能必须由相应的自动化系统支持。

使用周期性读取服务和变化传送会减轻 AS 和 AS-OS 二者的通讯负荷，因为读取请求不需要连续发送给 AS 并处理。

对于非周期性读取服务，要读取的变量被组合在单独的请求中，并被传送到 PLC。PLC 只能发送所需数据一次。请求周期的排列由 WinCC 执行。

CPU 中周期性读取服务的数目

周期性读取服务的数目取决于 S7-PLC 中可用的资源。对于 S7-300，最多有 4 个周期性服务可用，对于 S7-416 或 417，则最多为 32 个。该数目适用于与 PLC 进行通讯的所有成员，也就是说，如果有多个 WinCC 系统与 S7-PLC 进行通讯，则它们必须共享可用的资源。如果超过资源的最大数目，则更多的周期性读取服务访问将被拒绝。于是 WinCC 不得不使用非周期读取请求来请求该数据，并且还必须要执行周期排列。

请求脚本中的外部变量

如果选择的画面不包含任何使用函数“GetTagWord()”请求外部变量的脚本，则一旦画面已经打开，使用周期性读取服务对初始更新不会产生任何影响。如果在画面打开时使用“GetTagWord()”执行脚本，则该脚本的错误组态将导致画面切换后新的变量请求被连续发送到该通道。如果脚本中需要外部变量，则必须将“变量”作为触发事件。

如何组态系统参数

引言

本节说明如何组态通道“SIMATIC S7 Protocol Suite”的系统参数。

“系统参数”对话框由两个标签组成：

- SIMATIC S7 标签
- 单元标签

SIMATIC S7 和单元标签上的系统参数都是单元特定的数据，因此可以为通道中的每个通道单元单独设置。

对于 S7 通道的所有通道单元，这些标签是完全相同的。因此，通道单元“MPI”的对话框可用于所有实例中。

对参数值进行的任何改变都只有在 WinCC 重新启动之后才能生效。

说明

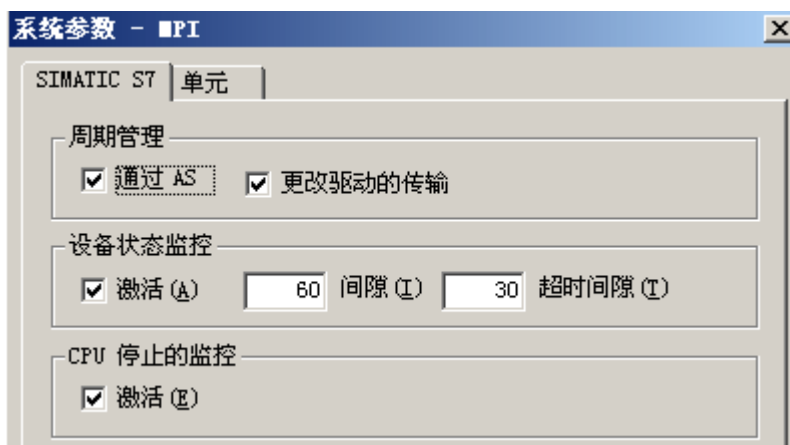
将项目复制到另一台计算机时，单元标签上的设置将保持不变，然而 SIMATIC S7 标签上的设置则可能改变。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。

步骤

1. 在变量管理器中选择通道“SIMATIC S7 Protocol Suite”。使用所期望通道单元的弹出式菜单打开“系统参数”对话框。
2. 选择 SIMATIC S7 标签。如果想要激活通道的变量周期性读取和变化传送功能，则勾选“通过 AS”和“with modification transfer”（使用变化传送）复选框。如果可用，此处将使用 PLC 中的周期性服务。详细信息请参见“PLC 周期性读取服务”。



3. 如果想要使用该功能，激活“设备状态监控”区中的复选框“激活”。在“间隙”域中，输入传送设备状态报文时间间隔的秒数。在“监视时间”域中，输入用于监控设备状态报文的响应的秒数值。
4. 如果要在 S7-CPU 处于停止状态时使 WinCC 指示通讯故障，则激活“CPU 停止的监控”区中的复选框“激活”。
5. 选择单元标签。根据所安装的通讯处理器，名称会显示在“逻辑设备名称”域中。只有在安装通讯处理器时选择了不同的名称时，才须改变该名称。详细信息请参见“更改逻辑设备名称”。



6. 如果只安装了该通讯类型的一个通讯处理器，则激活复选框“自动设置”，运行系统启动时将自动设置设备名称。
7. 如果写入请求的处理要优先于读取请求的处理，则激活复选框“写（带优先权）”。
8. 使用“确定”按钮关闭对话框。

参见

如何更改逻辑设备名称 (页 419)

PLC 中的周期性读取服务 (页 416)

如何更改逻辑设备名称

引言

通过逻辑设备名称与 S7 通讯。设备名称是在安装通讯处理器时分配的，并且是单元特定的条目。

根据已安装的通讯处理器，现在已为设备名称建立了相应的预设值。这些在下面的表格“缺省设备名称”中列出。

S7 通道所有单元的标签均完全相同，因此，描述中使用了通道单元“MPI”的对话框。

缺省设备名称

通道单元	缺省设备名称
工业以太网	CP_H1_1:
工业以太网 (II)	CP_H1_2:
MPI	MPI
命名连接	VM/
PROFIBUS	CP_L2_1:
PROFIBUS (II)	CP_L2_2:
Slot PLC	SLOT_PLC
Soft PLC	SOFT_PLC
TCP/IP	CP-TCPIP

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须创建与通道单元的连接，例如“MPI”。

步骤

1. 在变量管理器中选择所期望的通道单元。
2. 使用弹出式菜单打开“系统参数”对话框。
3. 选择单元标签。
4. 在域“逻辑设备名称”中指定设备名称。用户既可从选择列表中选择条目，也可手动输入新的名称。
所有可能的名称均由“组态 PG/PC 接口”（控制面板）工具确定。如果没有安装此工具，只能显示当前设定的设备名称。如果指定不同的逻辑设备名称，则将显示一条消息。
只有在目标站使用了组态站上尚未安装的通讯卡时，才能手动输入。
5. 单击“确定”按钮，关闭对话框。

说明

逻辑设备名称必须与设备设置中的名称的每个字母都完全一样。同时，“工业以太网”和“PROFIBUS”的缺省逻辑设备名称在名称末尾有一个冒号。

对参数值进行的任何改变都只有在 WinCC 重新启动之后才能生效。

6.11.5 特殊功能

6.11.5.1 “SIMATIC S7 Protocol Suite”通道的特殊功能

引言

SIMATIC S7 Protocol Suite 包含一些特殊功能；本章将说明这些特殊功能。

参见

软件冗余 (页 461)

通道“SIMATIC S7 Protocol Suite”的原始数据变量 (页 453)

使用 S7 功能块 AR_SEND 进行数据交换 (页 421)

6.11.5.2 使用 S7 功能块 AR_SEND 进行数据交换

使用 S7 功能块 AR_SEND 进行数据交换

简介

S7-400 AS 中的 S7 功能块 AR_SEND 用于将过程值传送到过程值归档。

操作原理

为了将 PLC 的过程值传送给 WinCC 中的过程值归档，“S7-400 PLC”具有集成的功能组件 SFB37 “AR_SEND”。

AR_SEND 组件的基本函数可向归档变量提供数据。如果使用了 AR_ID-Subnumber，则可以向多个变量提供数据。如果使用 AR_SEND 组件，则不是将过程值单个地发送到归档库，而是在 PLC 中累积多个过程值，然后以信息包方式传送。这样可减少所用网络上的负荷。

在 PLC 中，可使用的 AR_SEND 组件的数目取决于 CPI（例如 CPU 416 最多可使用 32 个 AR_SEND）。某 AR_ID 可依次分配给各 AR_SEND 组件。子编号用于增加可传送过程数据量，因为各 AR_ID 最多有 4095 个子编号。

实际上，每个 AR_SEND 组件要传送的归档变量的数目受到要传送的数据区的最大长度的限制。有关“数据块结构的结构和参数”的详细信息，请参见“过程值的数目”参数的描述。

AR_ID 和 AR_ID 子编号在 PLC 的数据和归档变量间建立了分配，并在组态 PLC 中数据库内的数据结构时，使用其它参数对其进行定义。

当自动计算其它参数时，在 WinCC 中组态该数据分配。

必须首先完成 PLC 中的 SFB 37“AR_SEND”的组态以及数据块结构的设置，因为在 WinCC 中组态是基于 PLC 中的这些数值。关于组态“AR_SEND”功能组件的信息，请参见 S7-400 PLC 文档。

AR_SEND 变量概述

变量: AR_SEND 用于...	每个 AR_SEND 的过程控制归档变量数目	预期用途
一个归档变量	a	用于为归档变量传送过程值，可在很短的时间间隔内读取过程值。
多个归档变量	对应于 AR_ID 子编号的数目	用于为多个归档变量传送过程值，可在很短的时间间隔内读取过程值。
多个归档变量（已优化）	对应于 AR_ID 子编号的数目	用于周期性地为最大数目的归档变量提供数据，每次提供一个数值。

参见

用于多个归档变量的 AR_SEND 变量 (页 435)

如何组态用于多个归档变量的 AR_SEND 变量 (页 451)

如何组态用于归档变量的 AR_SEND 变量 (页 449)

用于多个归档变量的 AR_SEND 变量（已优化） (页 447)

用于一个归档变量的 AR_SEND 变量 (页 430)

数据块 - 结构和参数

简介

在使用“AR_SEND”功能块传送数据前，要传送的数据作为 AS 中一个或多个数据块提供。数据块结构取决于各种参数，例如在使用的 AR_SEND 变量上，时间戳的使用或过程值的数据类型。

数据块中所使用的参数如下所述。在 AS 的数据块和“AR_SEND”功能块的参数化中设置各参数值。

当在 WinCC 中评估数据块时，进行参数检查。如果 WinCC 检测到数据块的结构出错或归档变量组态与所接收的数据不匹配，具有以下结构的条目会记录在 WinCC 诊断记录册中：

“日期时间、1003080、4、用户名、计算机名称、NRMS7PMC、PdeReceive: 来自连接（连接名）...的未知参数 AR_SEND + 描述错误的附加信息”

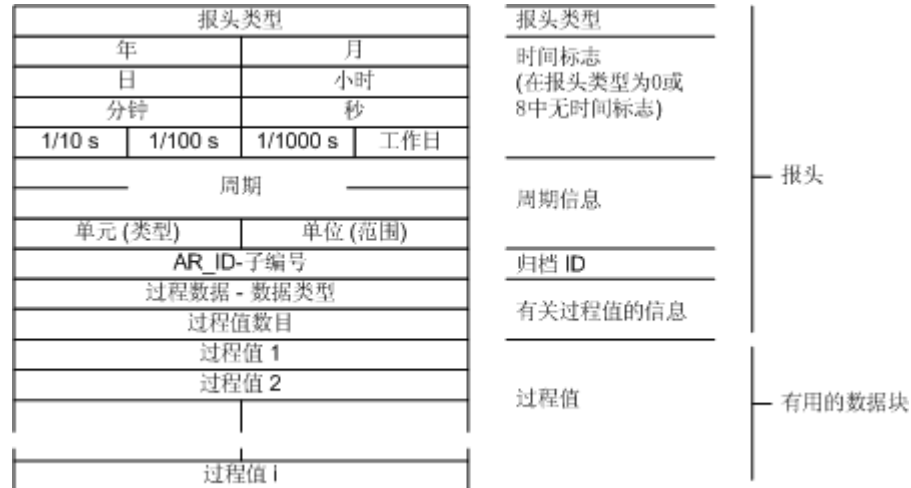
如果使用 WinCC 系统消息组态消息系统，则该诊断条目也会触发编号为 1003080 的 OS 过程控制消息。关于记录册条目的文本，请参见该消息的注释。

数据块的结构

每个数据块由报头和用户数据区组成：

- 报头包含与过程值及其周期以及时间戳有关的信息。
- 用户数据区包含实际过程值。

要传送的数据区由一个或多个数据块组成。



说明

在数据块中，每一行代表两个字节。过程值的长度根据数据类型而定，可以是一个字节或多个字节。详细信息请参见“过程值的数目”参数的描述。

参数说明

报头类型

报头类型定义了报头中所包含的信息的类型。

报头类型	时间戳	AR_ID 子编号
0	无时间戳的报头	无 AR_ID 子编号的报头
1	有时间戳的报头	无 AR_ID 子编号的报头
8	无时间戳的报头	有 AR_ID 子编号的报头
9	有时间戳的报头	有 AR_ID 子编号的报头

说明

如果报头类型为 0 和 8，报头中将省略时间戳的字节。因为这些字节也不会保留在数据块中，所以报头将减少 8 个字节。

AR_ID 子编号

指定 AS 用户数据和 WinCC 归档变量之间的分配，并在两个位置进行组态：

- 在 WinCC 中（组态过程控制归档变量时）
- 在 PLC 中设置要传送的用户数据区

子编号只与报头类型 8 或 9 有关。子编号有效值的范围为 1 到 4095。在 WinCC 中参数输出为十六进制值 (1 - 0FFF)。

时间戳

时间戳包含 SIMATIC S7 BCD 格式的日期和时间。工作日条目不由 WinCC 评估。

说明**夏时制/标准时间：自动化系统的时间设置**

自动化系统 S7 不识别夏时制/标准时间切换。通过标准化 DLL 在 WinCC 中执行时间戳到夏时制或标准时间的更正。已更正的时间和夏时制/标准时间的时间标识号可用于 WinCC 应用程序。然后将更正的时间和 ID 添加至归档（例如变量记录）。

要确保正确的时间设置，请在“WinCC 项目管理器 > 计算机属性 > 参数 > PLC 时间设置”(WinCC Explorer > Computer properties > Parameters > PLC Time Setting) 下的 WinCC 计算机属性中选择在 AS 中使用相同的时间设置。

周期

读取过程值的周期。该参数是在单位（范围）处指定的时间单位的因子。数据长度：双字。

例如：

“周期”= 10；“单位（范围）”= 4 意味着：过程值读取周期 = 10 秒

单元（类型）

指定时间信息的类型，且其对参数“过程值的数目”有影响。

数字	含义
1	按相等的时间间隔读取过程值。 启动时间在报头时间戳中给出，并且是强制性的。过程值之间的时间间隔由“单位（范围）”的时间单位和“周期”因素来定义。
2	每个过程值都有一个时间戳 不会估算报头中可能分配的时间戳。其结构对应于具有 8 个字节长度的报头中的时间戳。
3	每个过程值均拥有一个相对时间差，该时间差具有 2 个字的数据长度的时间单位。 绝对时间是报头中的时间戳的总和（= 启动时间），而相对时间差则使用在“单位（范围）”中设置的时间单位。报头的时间戳条目是强制性的。
4	每个过程值包含 AR_ID 子编号。 报头中给定的时间戳应用于过程值。报头的时间戳条目是强制性的。

单位（范围）

指定用于单位（类型）的时间单位 = 1 或 3。

数字	含义
1	保留
2	保留
3	毫秒
4	秒
5	分钟
6	小时
7	天

过程数据的数据类型

过程值直接以 S7 格式存储。

数字	S7 数据类型	WinCC 数据类型
0	BYTE	BYTE
1	WORD	WORD

数字	S7 数据类型	WinCC 数据类型
2	INT	SWORD
3	DWORD	DWORD
4	DINT	SDWORD
5	REAL	FLOAT

过程值的数目

根据“单元（类型）”中的条目，所传送的数据区可包含特定数量的过程值。该数目受到所传送的数据区的最大长度（16 KB）的限制。

将考虑使用 S7 功能“AR_SEND”和“BSEND/BRCV”与 S7-400 通信时的资源限制。这意味着使用 AR_SEND 和/或 BSEND/BRCV 功能，能够同时从 AS 传送到 WinCC 的最大数据量限制为最大 16 KB。

说明

对于 AR_SEND 变量“多个归档变量”，下列限制适用于该参数：

各种归档变量的数据块必须始终从字边界开始。因此，对于组合“数据类型过程值”= 0（字节）和“单元（类型）”= 1（具有相同时间间隔的过程值）的情况，必须输入一个偶数数目的过程值（= 字节），以用于参数“过程值的数目”。这种限制仅适用于该 AR_SEND 变量以及数据类型和“单元（类型）”的组合。

示例：

1x BSEND 最多为 16 KB

或 1 x 8 KB 的 AR_SEND + 1 x 8 KB 的 BSEND

或 1 x 10 KB 的 AR_SEND + 1 x 2 KB 的 AR_SEND + 1 x 4 KB 的 BSEND

单元 (类型)	过程值数目的含义
1	按相等的时间间隔读取过程值： 可以传送 8000 个 WORD/INT 数据类型的过程值，或 4000 个 DWORD/DINT/实型数据类型的值。
2	带时间戳的过程值： 用户数据区的每个元素由时间戳（8 个字节）和数值组成。可以传送 WORD/INT 数据类型的过程值 1600 个，或 DWORD/DINT/实型数据类型的值 1333 个。

单元 (类型)	过程值数目的含义
3	带时差的过程值： 用户数据区的每个元素由时间戳（4 个字节）和数值组成。可以传送 WORD/INT 数据类型的过程值 2666 个，或 DWORD/DINT/实型数据类型的值 2000 个。
4	过程值包含 AR_ID 子编号（具有多个变量的 AR-SEND，已优化）。 对于类型 4，每个过程值均由一个带有 AR_ID 子编号（数值范围：1 - 0x0FFF）的字和一个值组成。因此，用户数据区包括前面带“AR_ID”子编号的过程值数组。可以传送 WORD/INT 数据类型的过程值 3992 个，或 DWORD/DINT/实型数据类型的值 2660 个。

说明

数据块中给定的 AR_ID 子编号必须全部在 WinCC 中组态。如果发现存在未组态的子编号，WinCC 将停止对用户数据的说明。

各种归档变量的数据块必须始终从字边界开始。因此，对于数据类型字节和“单元（类型）”= 1（具有相同时间间隔的过程值），必须输入一个偶数数目的过程值（= 字节），以用于参数“过程值的数目”。这种限制仅适用于该 AR_SEND 变量以及数据类型和“单元（类型）”的组合。

参见

如何组态用于多个归档变量的 AR_SEND 变量 (页 451)

如何组态用于归档变量的 AR_SEND 变量 (页 449)

用于多个归档变量的 AR_SEND 变量（已优化） (页 447)

用于多个归档变量的 AR_SEND 变量 (页 435)

用于一个归档变量的 AR_SEND 变量 (页 430)

AR_SEND 变量的属性概述

引言

借助实例，这些表格将说明各种 AR_SEND 变量的属性及可能的参数值。

该表格不能显示所有可能的组合。

6.11 SIMATIC S7 Protocol Suite

“过程值的数据类型”的列“报头类型”以其在报头中的顺序显示。

说明

当组态功能块“AR_SEND”和 AS 中数据块的数据结构时，设置用于 AR_ID 和 AR_ID 子编号的数值以及一些其它参数。

用于归档变量的变量

实例/属性	实例编号	报头类型	日期/时间(报头的时间标志)	周期因子	单元(类型)	单位(范围)	AR_ID子编号	过程值数据类型	最大数目过程值	实例中的过程值结构
每个带有自己时间标志的过程值(字节)	1	0	不可用	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 个字节时间标志 + 1 个字节过程值
具有等间隔时间标志的过程值	2	1	相关的	>=1	1	3 到 7	0	0 1; 2 3; 4; 5	16000 8000 4000	1 个字过程值
每个带有自己时间标志的过程值(字)	3	1	不相关的	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 个字节时间标志 + 1 个字过程值
每个带有时差的过程值	4	1	相关的	>=1	3	3 到 7	0	0 1; 2 3; 4; 5	5332 2666 2000	8 个字节时间标志 + 1 个字节过程值

用于多个归档变量的变量

实例/属性	实例编号	报头类型	日期/时间 (报头的时间标志)	周期因子	单元 (类型)	单位 (范围)	AR_ID 子编号	过程值 数据类型	最大 数目 过程值	实例中的过 程值结构
每个带有自己时间标志的过程值 (字节)	5	8	不可用	0	2	0	1 到 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 个字节时间标志 + 1 个字节过程值
具有等间隔时间标志的过程值	6	9	相关的	>=1	1	3 到 7	1 到 4095	0 1; 2 3; 4; 5	16000 8000 4000	1 个字过程值
每个带有自己时间标志的过程值 (字)	7	9	不相关的	0	2	0	1 到 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 个字节时间标志 + 1 个字过程值
每个带有时差的过程值	8	9	相关的	0	3	3 到 7	1 到 4095	0 1; 2 3; 4; 5	5332 2666 2000	8 个字节时间标志 + 1 个字节过程值

用于多个归档变量的变量 - 已优化

实例/属性	实例编号	报头类型	日期/时间 (报头的时间标志)	周期因子	单元 (类型)	单位 (范围)	AR_ID 子编号	过程值 数据类型	最大 数目 过程值	实例中的过 程值结构
包含 AR_ID 子编号的各个过程值	9	1	相关的	0	4	0	0	1; 2 3; 4; 5	3992 2660	1 个字子编号 +1 个字过程值

参见

用于多个归档变量的 AR_SEND 变量（已优化）(页 447)

用于多个归档变量的 AR_SEND 变量(页 435)

用于一个归档变量的 AR_SEND 变量(页 430)

用于一个归档变量的 AR_SEND 变量

引言

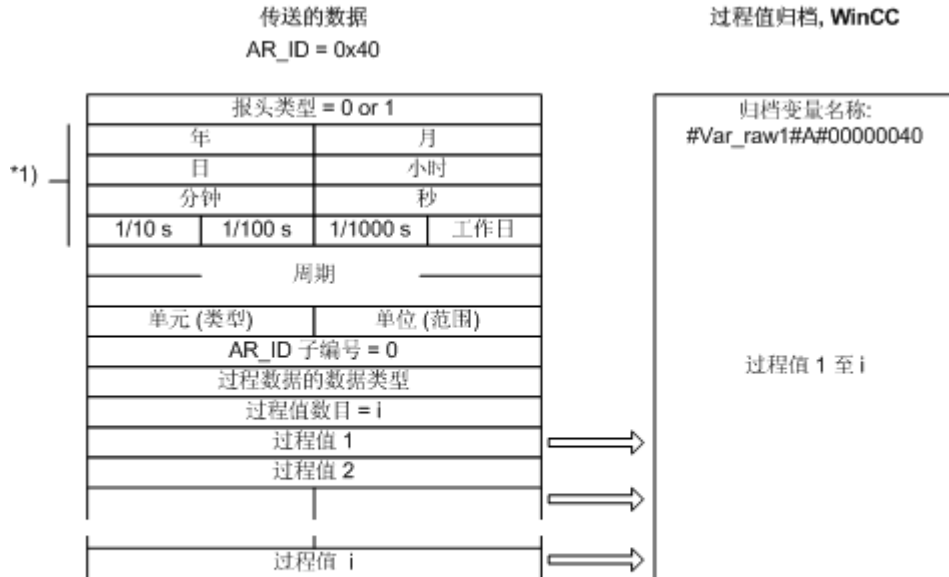
该变量可用于为归档变量提供过程值。也可用于较早的 WinCC 版本（V5.0 之前的）。

变量的属性:

- 报头类型必须为 0 或 1，即没有 AR_ID 子编号和有/没有时间标志。
- 不计算报头中的 AR_ID 子编号。
- WinCC 中的归档变量名不包含任何 AR_ID 子编号，因为只传送归档变量的过程值。

数据区结构实例

要传送的数据区由数据块组成。



*1) = 报头类型 0, 省略时间标志

参见

数据块结构实例 4: 一个归档变量; 每个过程值具有相对时间标志 (时差) (页 434)

数据块结构实例 3: 一个归档变量; 每个过程值具有自己的时间标志 (页 433)

数据块结构实例 2: 一个归档变量; 等间隔的时间标志 (页 432)

数据块结构实例 1: 一个归档变量; 每个过程值具有一个时间标志 (页 431)

AR_SEND 变量的属性概述 (页 427)

数据块结构实例 1: 一个归档变量; 每个过程值具有一个时间标志

引言

在本实例中, 将只为一个归档变量传送过程值。报头没有时间标志, 并且不保留相应的字节数。因此, 时间标志 (8 个字节) 位于每个过程值 (1 个字节) 之前。

过程值的数据类型为“字节”。

数据组件中的数据块结构

在 DB 中的地址	发送的数据块
0.0	报头类型 = 0
2.0	
4.0	周期 = 0
6.0	单元 (类型) = 2 单位 (范围) = 0
8.0	AR_ID-子编号 = 0
10.0	过程数据 - 数据类型 = 0
12.0	过程值数目 = 3 (最大 3200)
14.0	年=2001 月=10
16.0	日=05 小时=13
18.0	分钟=40 秒=00
20.0	1/10 s 1/100 s 1/1000 s 工作日
22.0	过程值 1 -
24.0	年=2001 月=10
26.0	日=05 小时=14
28.0	分钟=40 秒=00
30.0	1/10 s 1/100 s 1/1000 s 工作日
32.0	过程值 2 -
34.0	年=2001 月=10
36.0	日=05 小时=15
38.0	分钟=40 秒=00
40.0	1/10 s 1/100 s 1/1000 s 工作日
42.0	过程值 3 -

参见

数据块 - 结构和参数 (页 422)

数据块结构实例 2： 一个归档变量； 等间隔的时间标志

引言

在本实例中，将只为一个归档变量传送过程值。

使用参数“周期”= 1 和“单位（范围）”= 4 (= 秒) 创建 1 秒的等间隔时间标志。

过程值的数据类型为“WORD”。

数据组件中的数据块结构

在 DB 中的地址	发送的数据块
0.0	报头类型 = 1
2.0	年=2001 月=10
4.0	日=05 小时=13
6.0	分钟=40 秒=00
8.0	1/10 s 1/100 s 1/1000 s 工作日
10.0	周期 =1
12.0	
14.0	单元 (类型) = 1 单位 (范围) = 4
16.0	AR_ID-子编号 = 0
18.0	过程数据 - 数据类型 = 1
20.0	过程值编号 = 8 (最大 8000)
22.0	过程值 1
24.0	过程值 2
26.0	过程值 3
28.0	过程值 4
30.0	过程值 5
32.0	过程值 6
34.0	过程值 7
36.0	过程值 8

参见

数据块 - 结构和参数 (页 422)

数据块结构实例 3: 一个归档变量; 每个过程值具有自己的时间标志

引言

在本实例中, 将只为一个归档变量传送过程值。报头的时间标志不重要。因此, 时间标志 (8 个字节) 位于每个过程值 (1 个字) 之前。

过程值的数据类型为“SWORD”。

数据组件中的数据块结构

在 DB 中的地址	发送的数据块
0.0	报头类型 = 1
2.0	年=2001
4.0	日=05
6.0	分钟=40
8.0	1/10 s 1/100 s 1/1000 s 工作日
10.0	周期 =0
12.0	
14.0	单元 (类型) = 3 单位 (范围) = 4
16.0	AR_ID-子编号 = 0
18.0	过程数据 - 数据类型 = 3
20.0	过程值数目 = 3 (最大为 2000)
22.0	相对时间差, 以秒为单位
24.0	
26.0	过程值 1
28.0	
30.0	相对时间差, 以秒为单位
32.0	
34.0	过程值 2
36.0	
38.0	相对时间差, 以秒为单位
40.0	
42.0	过程值 3
44.0	

参见

数据块 - 结构和参数 (页 422)

用于多个归档变量的 AR_SEND 变量

引言

该变量允许给一个或多个归档变量提供过程值。对每个归档变量，将会分配一个 AR_ID 子编号，并且在需传送的数据区中创建数据块。

可以为每个 AR_ID 子编号传送“x”个过程值。有关“数据块结构的结构和参数”的更多信息，请参见“过程值的数目”参数的描述。

按照给定的“单元（类型）”和“单位（范围）”，从要传送的数据区取出归档变量值的时间标志。然后将其发送到 WinCC 过程值归档。

变量的属性:

- 报头类型必须是 8 或 9（有/没有时间标志，有 AR_ID 子编号）。
- 对每个 AR_ID 子编号，必须在需传送的数据区中创建数据块。
- 每个数据块中的 AR_ID 子编号必须大于 0。
- 在 WinCC 中，归档变量名具有 AR_ID 子编号。

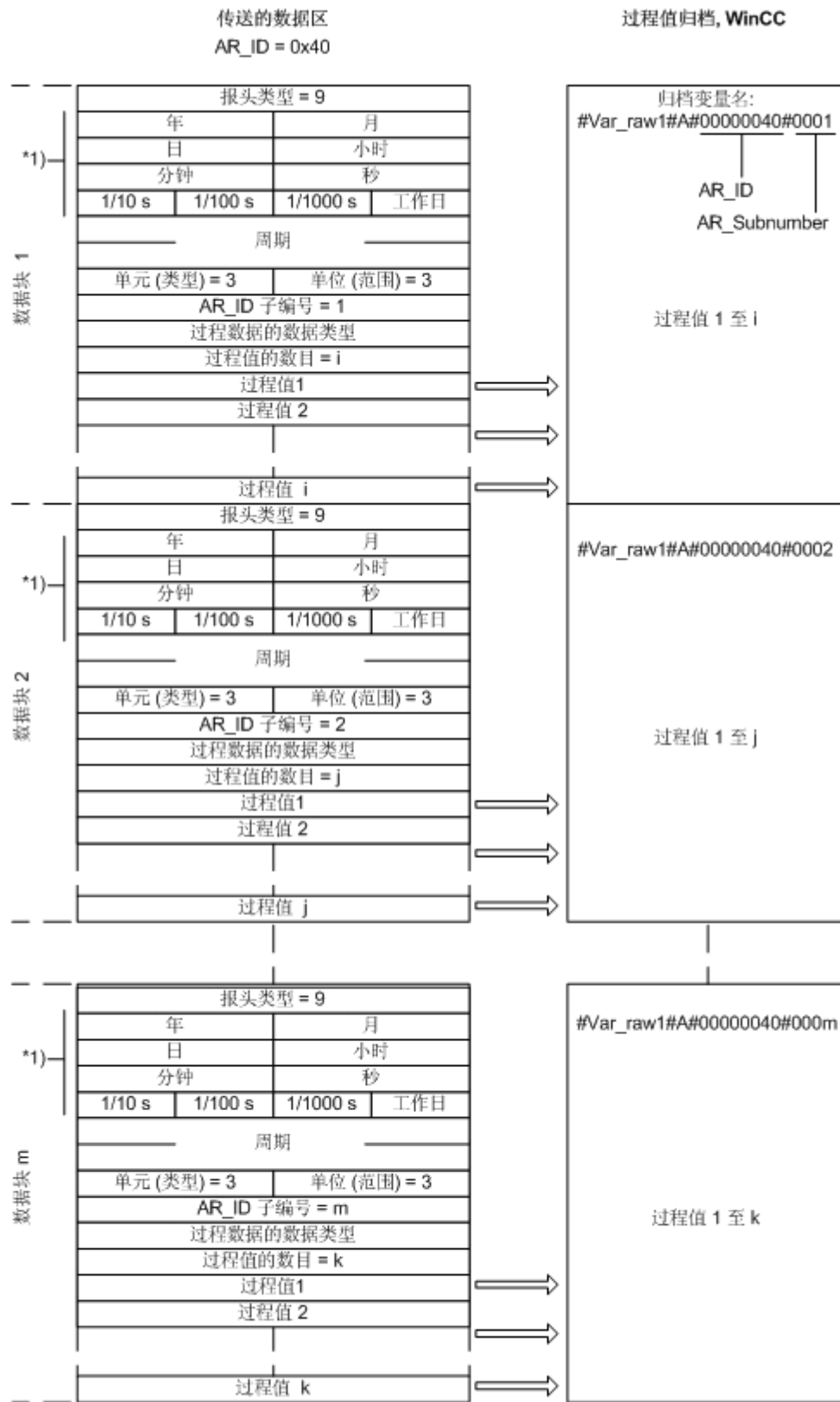
说明

数据块中给定的 AR_ID 子编号必须全部在 WinCC 中组态。如果发现存在未组态的子编号，WinCC 将停止对用户数据的说明。

各种归档变量的数据块必须始终从字边界开始。因此，对于组合“数据类型过程值”= 0（字节）和“单元（类型）”= 1（具有相同时间间隔的过程值）的情况，必须输入一个偶数数目的过程值（= 字节），以用于参数“过程值的数目”。这种限制仅适用于该 AR_SEND 变量以及数据类型和“单元（类型）”的组合。

数据区结构实例

所要传送的数据区由一个或多个数据块组成，这些数据块对应于需要数据的归档变量的数目。



*1) = 报头类型 0 或 8, 省略此时间标志

参见

数据块结构实例 8: 多个归档变量; 过程值具有相对时间标志 (时差) (页 445)

数据块结构实例 7: 多个归档变量; 每个过程值具有自己的时间标志 (页 443)

数据块结构实例 6: 多个归档变量; 等间隔的时间标志 (页 441)

数据块结构实例 5: 多个归档变量; 每个过程值具有自己的时间标志 (页 439)

AR_SEND 变量的属性概述 (页 427)

数据块结构实例 5: 多个归档变量; 每个过程值具有自己的时间标志

引言

在本实例中, 将传送用于多个归档变量的过程值。

不同归档变量的数据块在数据组件中都按顺序放置。在各个数据块中, 输入不同的 AR_ID 子编号。

报头没有时间标志, 并且不保留相应的字节数。因此, 时间标志 (8 个字节) 位于每个过程值 (1 个字节) 之前。

过程值的数据类型为“字节”。

数据组件中的数据块结构

在 DB 中的地址	发送的数据块
0.0	报头类型 = 8
2.0	周期 = 0
4.0	
6.0	单元 (类型) = 2 单位 (范围) = 0
8.0	AR_ID-子编号 = 1
10.0	过程数据-数据类型 = 0
12.0	过程值数目 = 3
14.0	年=2001 月=10
16.0	日=05 小时=13
18.0	分钟=40 秒=00
20.0	1/10 s 1/100 s 1/1000 s 工作日
22.0	过程值 1 -
24.0	年=2001 月=10
26.0	日=05 小时=14
28.0	分钟=40 秒=00
30.0	1/10 s 1/100 s 1/1000 s 工作日
32.0	过程值 2 -
34.0	年=2001 月=10
36.0	日=05 小时=15
38.0	分钟=40 秒=00
40.0	1/10 s 1/100 s 1/1000 s 工作日
42.0	过程值 3 -
44.0	报头类型 = 8
46.0	周期 = 0
48.0	
50.0	单元 (类型) = 2 单位 (范围) = 0
52.0	AR_ID-子编号 = 2
54.0	过程数据 - 数据类型 = 0
56.0	过程值数目 = 2
58.0	年=2001 月=10
60.0	日=05 小时=12
62.0	分钟=40 秒=00
64.0	1/10 s 1/100 s 1/1000 s 工作日
66.0	过程值 1 -
68.0	年=2001 月=10
70.0	日=05 小时=13
72.0	分钟=40 秒=00
74.0	1/10 s 1/100 s 1/1000 s 工作日
76.0	过程值 2 -

参见

数据块 - 结构和参数 (页 422)

数据块结构实例 6：多个归档变量；等间隔的时间标志

引言

在本实例中，将传送用于多个归档变量的过程值。不同归档变量的数据块在数据组件中都按顺序放置。在各个数据块的报头中，输入不同的 AR_ID 子编号。

使用参数“周期”= 1 和“单位（范围）”= 4（= 秒）创建 1 秒的等间隔时间标志。

过程值的数据类型为“WORD”。

数据组件中的数据块结构

在 DB 中的地址:	发送的数据块
0.0	报头类型 = 9
2.0	年=2001
4.0	日=05
6.0	分钟=40
8.0	1/10 s 1/100 s 1/1000 s 工作日
10.0	周期 = 1
12.0	
14.0	单元 (类型) = 1
16.0	AR_ID-子编号 = 1
18.0	过程数据 - 数据类型 = 1
20.0	过程值数目 = 8
22.0	过程值 1
24.0	过程值 2
26.0	过程值 3
28.0	过程值 4
30.0	过程值 5
32.0	过程值 6
34.0	过程值 7
36.0	过程值 8
38.0	报头类型 = 9
40.0	年=2001
42.0	日=05
44.0	分钟=40
46.0	1/10 s 1/100 s 1/1000 s 工作日
48.0	周期 = 1
50.0	
52.0	单元 (类型) = 1
54.0	AR_ID-子编号 = 2
56.0	过程数据 - 数据类型 = 1
58.0	过程值数目 = 5
60.0	过程值 1
62.0	过程值 2
64.0	过程值 3
66.0	过程值 4
68.0	过程值 5

参见

数据块 - 结构和参数 (页 422)

数据块结构实例 7：多个归档变量；每个过程值具有自己的时间标志

引言

在本实例中，将传送用于多个归档变量的过程值。不同归档变量的数据块在数据组件中都按顺序放置。在各个数据块中，输入不同的 AR_ID 子编号。

报头的时间标志不重要。因此，时间标志（8 个字节）位于每个过程值（1 个字）之前。

过程值的数据类型为“SWORD”。

数据组件中的数据块结构

在 DB 中的地址	发送的数据
0.0	报头类型= 9
2.0	年=0
4.0	日=0
6.0	分钟=0
8.0	1/10 s 1/100 s 1/1000 s 工作日
10.0	周期 =0
12.0	
14.0	单元 (类型) = 2 单位 (范围) = 0
16.0	AR_ID-子编号 = 1
18.0	过程数据 - 数据类型 = 2
20.0	过程值数目 = 3
22.0	年=2001
24.0	日=05
26.0	分钟=40
28.0	1/10 s 1/100 s 1/1000 s 工作日
30.0	过程值 1
32.0	年=2001
34.0	日=05
36.0	分钟=40
38.0	1/10 s 1/100 s 1/1000 s 工作日
40.0	过程值 2
42.0	年=2001
44.0	日=05
46.0	分钟=40
48.0	1/10 s 1/100 s 1/1000 s 工作日
50.0	过程值 3
52.0	报头类型= 9
54.0	年=0
56.0	日=0
58.0	分钟=0
60.0	1/10 s 1/100 s 1/1000 s 工作日
62.0	周期 =0
64.0	
66.0	单元 (类型) = 2 单位 (范围) = 0
68.0	AR_ID-子编号 = 2
70.0	过程数据 - 数据类型 = 2
72.0	过程值数目 = 2
74.0	年=2001
76.0	日=05
78.0	分钟=40
80.0	1/10 s 1/100 s 1/1000 s 工作日
82.0	过程值 1
84.0	年=2001
86.0	日=05
88.0	分钟=40
90.0	1/10 s 1/100 s 1/1000 s 工作日
92.0	过程值 2

参见

数据块 - 结构和参数 (页 422)

数据块结构实例 8: 多个归档变量; 过程值具有相对时间标志 (时差)

引言

在本实例中, 将为具有时间标志的一个归档变量传送过程值。

不同归档变量的数据块在数据组件中都按顺序放置。在各个数据块中, 输入不同的 AR_ID 子编号。

参数“单元 (类型)”= 3, 每个过程值有一个相对于报头中时间标志的时差 (4 个字节)。为每个归档变量使用参数“单位 (范围)”单独定义时差的时间单位, 因此, 也为各数据块进行了设置。

过程值的数据类型为“DWORD”。

数据组件中的数据块结构

在 DB 中的地址	发送的数据块
0.0	报头数据 = 9
2.0	年=2001
4.0	日=05
6.0	分钟=40
8.0	1/10 s 1/100 s 1/1000 s 工作日
10.0	周期 =0
12.0	
14.0	单元 (类型) = 3 单位 (范围) = 5
16.0	AR_ID-子编号 = 1
18.0	过程数据 - 数据类型 = 3
20.0	过程值数目 = 3
22.0	相对时间差, 以分钟为单位
24.0	
26.0	过程值 1
28.0	
30.0	相对时间差, 以分钟为单位
32.0	
34.0	过程值 2
36.0	
38.0	相对时间差, 以分钟为单位
40.0	
42.0	过程值 3
44.0	
46.0	报头数据 = 9
48.0	年=2001
50.0	日=05
52.0	分钟=40
54.0	1/10 s 1/100 s 1/1000 s 工作日
56.0	周期 =0
58.0	
60.0	单元 (类型) = 3 单位 (范围) = 6
62.0	AR_ID-子编号 = 2
64.0	过程数据 - 数据类型 = 3
66.0	过程值数目 = 2
68.0	相对时间差, 以小时为单位
70.0	
72.0	过程值 1
74.0	
76.0	相对时间差, 以小时为单位
78.0	
80.0	过程值 2
82.0	

参见

数据块 - 结构和参数 (页 422)

用于多个归档变量的 AR_SEND 变量（已优化）

引言

每当一次性为最大数目的归档变量提供过程值时，使用该变量。因此，将被传送的数据区只有一个数据块组成，并且每个过程值只有自己的 AR_ID 子编号及其相应数值。

在该数据块中，所有归档变量的过程值的数据类型是一样的。

变量的属性：

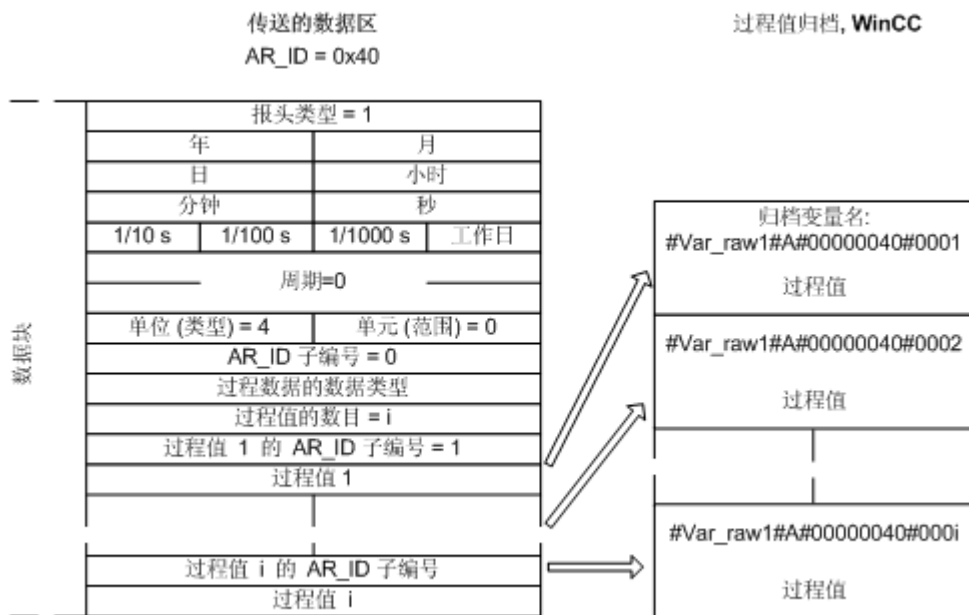
- 报头类型必须是 1（有时间标志，没有 AR_ID 子编号）。
- 在数据块中相关过程值的 AR_ID 子编号必须大于 0。不计算报头中的 AR_ID 子编号。
- “单元（类型）”参数必须是 4，也就是说，过程值包含 AR_ID 子编号。
- “单位（范围）”参数必须是 0，也就是说，报头中的时间标志适用于所有过程值，并且没有相对时间。
- 在 WinCC 中，归档变量名具有 AR_ID 子编号。

说明

如果对于过程值所包含的 AR_ID 子编号不存在任何 WinCC 归档变量与其对应，将出现一条 WinCC 诊断记录条目。继续处理其余的过程值。

数据区结构实例

要传送的数据区由一个数据块组成。



参见

AR_SEND 变量的属性概述 (页 427)

数据块结构实例 9: 多个归档变量; 已优化 (页 448)

数据块结构实例 9: 多个归档变量; 已优化

引言

在本实例中, 将为具有时间标志的一个归档变量传送过程值。时间标志适合于所有的归档变量。

相应的 AR_ID 子编号位于每个过程值的前面。

过程值的数据类型为“WORD”。

数据组件中的数据块结构

在 DB 中的地址	发送的数据块
0.0	报头类型 = 9
2.0	年=2001
4.0	日=05
6.0	分钟=40
8.0	1/10 s
10.0	1/100 s
12.0	1/1000 s
14.0	工作日
16.0	周期 =0
18.0	单元 (类型) = 4
20.0	单位 (范围) = 0
22.0	AR_ID-子编号 = 0
24.0	过程数据 - 数据类型 = 1
26.0	过程值数目 = 5 (最大为 3992)
28.0	AR_ID-子编号
30.0	过程值1
32.0	AR_ID-子编号
34.0	过程值2
36.0	AR_ID-子编号
38.0	过程值3
40.0	AR_ID-子编号
	过程值4
	AR_ID-子编号
	过程值5

参见

数据块 - 结构和参数 (页 422)

如何组态用于归档变量的 AR_SEND 变量

简介

有大量变量使用 AR_SEND 功能来交换数据。对于变量“一个归档变量”，只需使用 AR_ID。而不使用 AR_ID 子编号。

AR_ID 在组态 AS 中的数据块与功能块 SFB 37“AR_SEND”期间，用于建立 AS 中的数据与归档变量之间的分配并和其它参数一起组态。

在 WinCC 中，这种分配在过程控制变量的属性中完成。该分配是 WinCC 中所必须的唯一组态步骤，本节将对其进行说明。

WinCC 中其它的参数不需要组态，因为它们自动计算。

说明

要将该变量与 AR_ID 结合使用，报头类型必须组态为 0 或 1。AR_ID 子编号必须设置为零。在 WinCC V5.0 以前的版本中所组态的所有归档变量均可使用该变量，因为它们没有 AR_ID 子编号。


从 WinCC 版本 5.1 Hotfix 4 开始，可以为过程控制变量的归档变量名指定一个别名，或使用由系统生成的内部名称。由系统生成的名称包含已分配的原始数据变量的名称，而不是来自版本 V5.1 HF4 的原始数据标识号。在从 V5.1 HF4 移植而来的项目中，归档变量名称可以使用其原格式或进行转换。打开过程控制归档变量的属性对话框一次，随后关闭并保存来执行转换。不必指定别名。

如果项目中的所有外部变量都利用“AS-OS 传送”功能重新映射，则必须将归档变量名转换为新结构！然后保持新结构。

要求

- 必须在 AS 中首先组态“AR_SEND”功能块和数据块结构，并且该组态信息在下列过程中是可用的。
- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须在通道单元（例如“工业以太网”）中创建连接（例如“Test_Ind_Eth”）。
- 必须在“变量记录”编辑器中组态过程值归档。

步骤

1. 在通道“SIMATIC S7 Protocol Suite”中，选择要用于数据传送的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
为该变量输入名称“Var_raw_arsend”。名称不可超过 24 个字符。当组态过程控制变量时，原始数据变量名成了归档变量名的一部分，因此要考虑到此长度限制。
4. 在“数据类型”(Data type) 字段中，选择“原始数据类型”(Raw Data Type)。
5. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
6. 选中“原始数据”(Raw Data) 复选框。在“原始数据类型”(Raw Data Type) 区域中选择类型“归档数据链接”(Archive Data Link)。单击“确定”(OK) 关闭所有打开的对话框。
7. 打开“变量记录”(Tag Logging) 编辑器。在“变量记录”(Tag Logging) 编辑器的导航区域，选择“过程值归档”(Process Value Archives) 文件夹。在表格区域选择“过程控制变量”(Process-controlled tags) 选项卡并添加原始数据变量“Var_raw_arsend”。

8. 在“属性”(Properties) 区域编辑属性。
9. 在“转换 DLL”(Conversion DLL) 字段中选择条目“nrms7pmc.nll”。
10. 在“块 ID”(Block ID) 中输入十六进制数值的 AR_ID。该数值通过 AS 中的组态进行定义。请不要在“子编号”(Subnumber) 复选框内输入任何数字，因为该 AR_SEND 变量中没有使用子编号。
11. “变量名称”(Tag Name) 显示由系统生成的内部归档变量名称。它包含了分配的原始数据变量的名称和 AR_ID。如果需要，在“归档变量名称”(Archive Tag Name) 中，为该归档变量定义一个别名。如果不输入别名，在 WinCC 中将使用内部归档变量名进行过程值归档期间的管理以及对归档变量进行寻址。
12. 关闭变量记录。

如何组态用于多个归档变量的 AR_SEND 变量

简介

有大量的使用 AR_SEND 功能的变量来为多个归档变量交换数据。

- 使用变量“多个归档变量”在不同时刻为多个归档变量提供多个数值。
- 使用变量“多个归档变量 - 已优化”，同时为最大数目的归档变量每一个都提供一个数值。

这两种变量都使用 AR_ID 和 AR_ID 子编号。

AR_ID 和 AR_ID 子编号在 AS 的数据和归档变量间建立了分配。当组态数据块和功能模块 SFB 37“AR_SEND”时，它们在 AS 中和其它参数一起定义。

在 WinCC 中，这种分配在过程控制变量的属性中完成。该分配是 WinCC 中所必须的唯一组态步骤，本节将对其进行说明。

WinCC 中其它的参数不需要组态，因为它们自动计算。

说明

为使用 AR_ID 子编号，报头类型必须组态为 8 或 9。


在 WinCC 版本 V5.0 中组态的归档变量没有 AR_ID 子编号，因此只能在“一个变量”中使用。从 WinCC 版本 5.1 Hotfix 4 开始，可以为过程控制变量的归档变量名指定一个别名，或使用由系统生成的内部名称。由系统生成的名称包含已分配的原始数据变量的名称，而不是来自版本 V5.1 HF4 的原始数据标识号。在从 V5.1 HF4 移植而来的项目中，归档变量名称可以使用其原格式或进行转换。打开过程控制归档变量的属性对话框一次，随后关闭并保存来执行转换。不必指定别名。

如果项目中的所有外部变量都利用“AS-OS 传送”功能重新映射，则必须将归档变量名转换为新结构！然后保持新结构。

要求

- 必须在 AS 中首先组态“AR_SEND”功能块和数据块结构，并且该组态信息在下列过程中是可用的。
- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须在通道单元（例如“工业以太网”）中创建连接（例如“Test_Ind_Eth”）。
- 必须在“变量记录”编辑器中组态过程值归档。

步骤

1. 在通道“SIMATIC S7 Protocol Suite”中，选择要用于数据传送的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
为该变量输入名称“Var_raw_arsend”。名称不可超过 24 个字符。当组态过程控制变量时，原始数据变量名成了归档变量名的一部分，因此要考虑到此长度限制。
4. 在“数据类型”(Data type) 字段中，选择“原始数据类型”(Raw Data Type)。
5. 在通道“SIMATIC S7 Protocol Suite”中，选择要用于数据传送的连接。从连接的快捷菜单中选择“新建变量”(New Tag) 选项。将打开“变量属性”(Tag Properties) 对话框。
6. 在“名称”(Name) 字段中输入“Var_raw_arsend”作为变量的名称。名称不可超过 24 个字符。当组态过程控制变量时，原始数据变量名成了归档变量名的一部分，因此要考虑到此长度限制。
7. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
8. 选中“原始数据”(Raw Data) 复选框。在“原始数据类型”(Raw Data Type) 区域中选择类型“归档数据链接”(Archive Data Link)。单击“确定”(OK) 关闭所有打开的对话框。
9. 打开“变量记录”(Tag Logging) 编辑器。在“变量记录”(Tag Logging) 编辑器的导航区域，选择“过程值归档”(Process Value Archives) 文件夹。在表格区域选择“过程控制变量”(Process-controlled tags) 选项卡并添加原始数据变量“Var_raw_arsend”。
10. 在“属性”(Properties) 区域编辑属性。
11. 在“转换 DLL”(Conversion DLL) 字段中选择条目“nrms7pms.dll”。在“块 ID”(Block ID) 中输入十六进制数值的 AR_ID。该数值通过 AS 中的组态进行定义。对于“子编号”(Subnumber)，将 AR_ID 子编号按十六进制数值输入。该数值也通过 AS 中的组态进行定义。
12. “变量名称”(Tag Name) 显示由系统生成的内部归档变量名称。它包含了分配的原始数据变量的名称和 AR_ID。如果需要，在“归档变量名称”(Archive Tag Name) 中，为该归档变量定义一个别名。如果不输入别名，在 WinCC 中将使用内部归档变量名进行过程值归档期间的管理以及对归档变量进行寻址。
13. 关闭变量记录。

6.11.5.3 通道“SIMATIC S7 Protocol Suite”的原始数据变量

通道“SIMATIC S7 Protocol Suite”的原始数据变量

引言

- 原始数据类型变量是一种传输层的数据报文。原始数据变量的内容并不固定，因此传送的数据只能由发送方和接收方来解释。该数据类型的 WinCC 中没有更改任何格式。最大长度是 65535 字节。
- WinCC 区分两种类型的原始数据变量：一种是对用户开放的原始数据变量，另一种是用于处理 S7 功能的原始数据变量。

对用户开放的原始数据变量

对用户开放的原始数据变量用于在 WinCC 和 PLC 之间传送用户数据块，并且只处理用户数据。这种原始数据变量区分：

原始数据变量作为字节数组

用于 BSEND/BRCV 功能的原始数据变量

用于处理 S7 功能的原始数据变量

这些原始数据变量没有任何通道专用的报头，通常由消息系统使用，并用于 WinCC 中的过程数据条目。

此处不需要更详细的描述，因为这些都是通道内部的变量和功能。

参见

用于 S7 通信 BSEND/BRCV 功能的原始数据变量 (页 456)

原始数据变量作为字节数组 (页 453)

原始数据变量作为字节数组

简介

作为字节数组的原始数据变量用于在 WinCC 和 PLC 之间传送用户数据块，并且只处理用户数据。

作为字节数组的原始数据变量在通道中将和通常的过程变量一样进行处理，通过数据区的地址和长度可对其进行寻址（例如，DB 100、DW 20、长度 40 字节）。

原始数据长度限于一个可传送的数据块，必须能够使用一个 PDU（协议数据单元）完整地传送。使用通信驱动程序能够传送的数据块最大长度取决于建立连接时协商确定的 PDU 长度减去报头以及其它附加信息。因此，SIMATIC S7 中通常使用的 PDU 长度产生下列最大长度：

- S7-300：PDU 长度 240 字节，数据块最大长度 208 字节
- S7-400：PDU 长度 480 字节，数据块最大长度 448 字节

如果要传送更大的数据块，则数据必须分块。在 PLC 中，块由 S7 软件组成；在 WinCC 中，块由脚本组成。

如何将原始数据变量组态为字节数组

用于传送数据块的原始数据变量组态为“发送/接收块”类型的原始数据，带有一个地址和一份长度详细资料。

下图显示了数据组件 100 中从数据字 20 开始的 40 字节长数据区的组态实例：



将原始数据变量读取为字节数组

原始数据变量以与“标准”过程变量相同的方式进行读取。在 AS 中请求相应的数据块，并在接收到数据后传送给用户。

数据传送始终在 WinCC 启动时发生。使用该原始数据变量，不能在 AS 启动时零星地或以事件控制方式接收数据。

将原始数据变量作为字节数组写入

原始数据变量以与“标准”过程变量相同的方式进行写入。发送数据块并从 AS 接收到肯定的确认后，数据块将传送到数据管理器映像。

参见

如何将原始数据变量组态为字节数组 (页 455)

如何将原始数据变量组态为字节数组

简介


本节将说明如何将“SIMATIC S7 Protocol Suite”通道的原始数据变量组态为字节数组。

对于通道的所有通道单元，这些组态是完全相同的。本实例中使用了“MPI”通道单元及其连接。

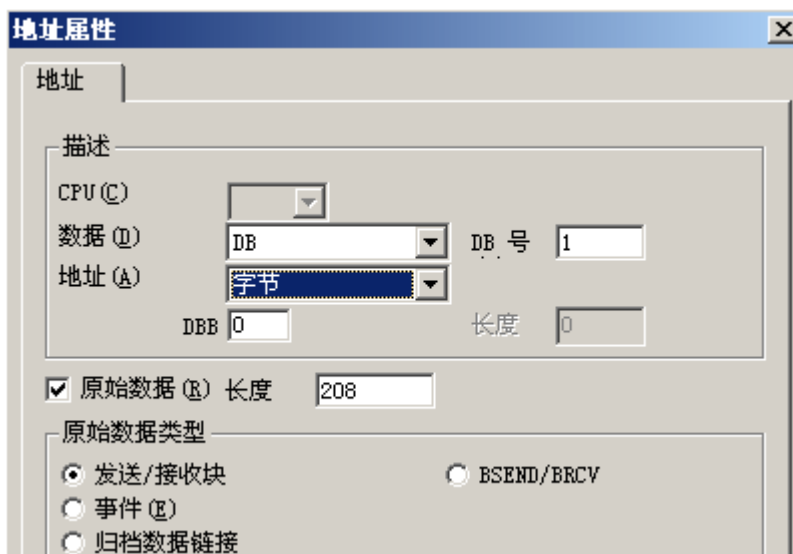
要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须创建与通道单元的连接，例如“MPI”。

步骤

1. 在通道“SIMATIC S7 Protocol Suite”中，选择要用于数据传送的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
为该变量输入名称“Var1_raw_byte”。
4. 在“数据类型”(Data Type) 字段中，选择“原始数据类型”(Raw data type)。
5. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
6. 选中“原始数据”(Raw Data) 复选框。在“原始数据类型”区域中选择类型“发送/接收块”。
该设置将决定“地址描述”(Address description) 区域中的字段显示以及“原始数据”(Raw data) 复选框旁的字段显示。
7. 在“长度”域中输入原始数据块的长度（以字节为单位）。

8. 在“数据区”(Data area)中，设置 PCL 中数据所处的数据区。如果在数据区选择了“DB”，则在激活的“DB 号”(DB No.) 字段中输入数据块号。
“Cpu”字段被禁用以连接“MPI”通道单元。



9. 在“寻址”域中设置寻址类型。如果 WinCC 变量的数据类型是“原始数据类型”，可以使用条目“字节”、“字”或“双字”。
10. 在下面的域中，输入起始地址的值。左侧的域标签取决于“数据区”和“地址”域中的条目，例如数据域选择“DB”而地址域选择“字节”，则此域的标签显示为“DBB”。
11. 单击“确定”(OK) 关闭所有打开的对话框。

说明

原始数据长度限于一个可传送的数据块，必须能够使用一个 PDU（协议数据单元）完整地传送。使用通信驱动程序能够传送的数据块最大长度取决于建立连接时协商确定的 PDU 长度减去报头以及其它附加信息。因此，SIMATIC S7 中通常使用的 PDU 长度产生下列最大长度：

- S7-300：PDU 长度 240 字节，数据块最大长度 208 字节
 - S7-400：PDU 长度 480 字节，数据块最大长度 448 字节
- 长度错误将导致读/写作业被拒绝，同时会显示出错消息。

用于 S7 通信 BSEND/BRCV 功能的原始数据变量

简介

用于“BSEND/BRCV”功能的原始数据变量将用在 WinCC 和 PLC 之间传送用户数据块，并且只处理用户数据。

该原始数据类型可用于访问 S7 通信“BSEND/BRCV”功能。

以下自动化系统支持通过命名连接实现的“BSEND/BRCV”原始数据通信：

- S7-400
- S7-300
 - CPU319-3 PN/DP V2.5 或更高版本
 - CPU317-2 PN/DP V2.6 或更高版本
 - CPU315-2 PN/DP V3.1 或更高版本对于 S7-300 控制器，推荐使用固件版本 V3.x 或更高版本。
不能通过通信处理器建立原始数据通信。

- WinAC RTX 2010

数据传送的启动总是由传送伙伴负责；因此，“BSEND/BRCV”功能也可以用于以事件控制方式或零星地传送数据块。

由于资源的原因，建议使用较少数量的 BSEND/BRCV 原始数据变量。

使用 S7 函数“AR_SEND”和“BSEND/BRCV”的资源限制

使用 AR_SEND 和/或 BSEND/BRCV 功能同时从 AS 传送到 WinCC 的最大数据量受到如下限制：

- 对于 S7-400 为 16 KB
- 对于 S7-300 为 8 KB

有关 S7-400 的示例：

- 1x BSEND 最多为 16 KB
- 1x AR_SEND (8kB) + 1x BSEND (8kB)
- 1x AR_SEND (10 KB) + 1x AR_SEND (2 KB) + 1x BSEND (4 KB)

说明

协调写入作业

如果写入工作的数据块传送给 AS，并且还没有从接收缓冲器删除或完全删除，那么下一步的写入工作将被拒绝，同时会显示出错消息。

显示这样的错误时，R_ID > 0x8000 0000 的写入工作会写入连接专用的队列，而且系统会尝试重复该写入工作，时间持续 6 秒钟。

用户负责传送的时间协调，并且需要为写入工作指定较短的时间间隔。

组态 PBK 连接以使用“BSEND/BRCV”功能

“BSEND/BRCV”功能只能通过一个称为 PBK 连接（可编程组件通信）的“硬组态连接”来使用。

要组态硬组态连接，必须在连接参数中指定连接资源（十六进制：10 ... DF）。

在 PLC 内组态连接时，将由 STEP7 分配该连接资源。

该连接必须组态为自动化系统中的被动连接端点。

读取/写入作业

硬组态连接也可用于处理“普通”读和写作业。

如果要通过该连接传送非常大的数据区，则数据块要使用多个 PDU 进行传送。

由于性能原因，建议创建 BSEND/BRCV 功能专用连接。

组态用于 BSEND/BRCV 功能的原始数据变量

用于传送“BSEND/BRCV”数据块的原始数据变量将被组态为“BSEND/BRCV”类型且带有“R_ID”的原始数据。

在发送或接收的数据量中隐含地给出了数据长度。



“R_ID”参数

必须以十六进制数字给出 32 位长的 R_ID，以便利用“BSEND/BRCV”功能。

R_ID 在 AS 中组态时指定，并且用于区分一个连接上的多个数据块传送。

与下面的通信子系统（SIMATIC 设备驱动程序）中的 R_ID 有关的发送和接收调用始终会进行通知。

所以总是给原始数据变量分配一个唯一的 R_ID。

发送“BSEND/BRCV”原始数据变量

原始数据变量“BSEND/BRCV”发送的方法与“普通”过程变量写入的方式一样。

发送数据块并从 AS 接收到肯定的确认后，数据块将传送到数据管理器映像。

接收“BSEND/BRCV”原始数据变量

“BSEND/BRCV”原始数据将根据 AS 的触发进行发送。

因此，直接读 S7 原始数据变量是不可能的。

同步

BSEND/BRCV 功能不包含任何同步功能。

如果在启动阶段没有用户登录以接收数据，由 AS 发送的数据块将在接收方跳动。

因此，用户需注意同步，例如，通过在数据字中设置标记来释放 AS 上的发送方向。

参见

如何组态“BSEND/BRCV”功能的原始数据变量 (页 459)

如何组态“BSEND/BRCV”功能的原始数据变量

简介


本节将说明如何组态用于“BSEND/BRCV”功能的“SIMATIC S7 Protocol Suite”通道原始数据变量。

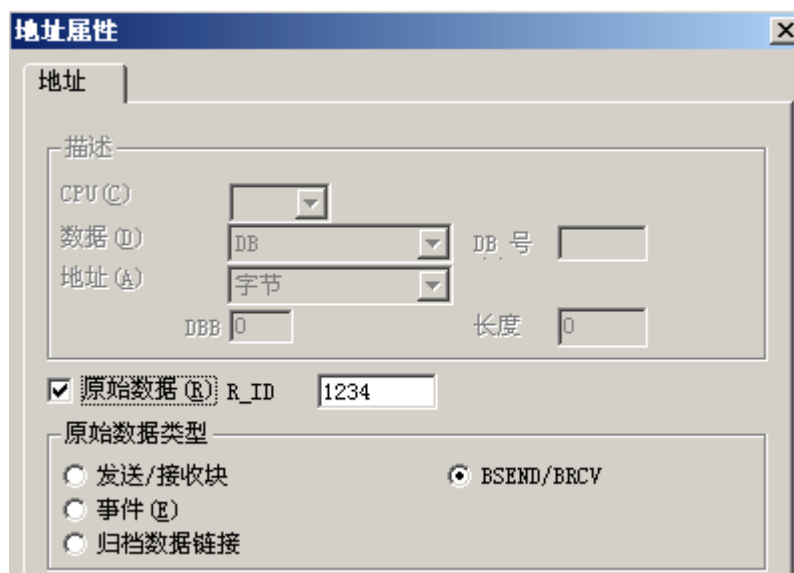
对于通道的所有通道单元，这些组态是完全相同的。本实例中使用了“MPI”通道单元及其连接。

要求

- 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
- 必须创建与通道单元的连接，例如“MPI”。

步骤

1. 在通道“SIMATIC S7 Protocol Suite”中，选择要用于数据传送的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
为该变量输入名称“Var2_raw_bsend”。
4. 在“数据类型”(Data Type) 字段中，选择“原始数据类型”(Raw data type)。
5. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
6. 选中“原始数据”(Raw Data) 复选框。在“原始数据类型”区域中选择类型“BSEND/BRCV”。“地址描述”区域中的域的显示现在被取消激活。
7. 将 ID 的十六进制数输入到“R_ID”域。组态时在 AS 中指定 R_ID



8. 单击“确定”按钮，关闭上述两个对话框。

6.11.5.4 软件冗余

软件冗余

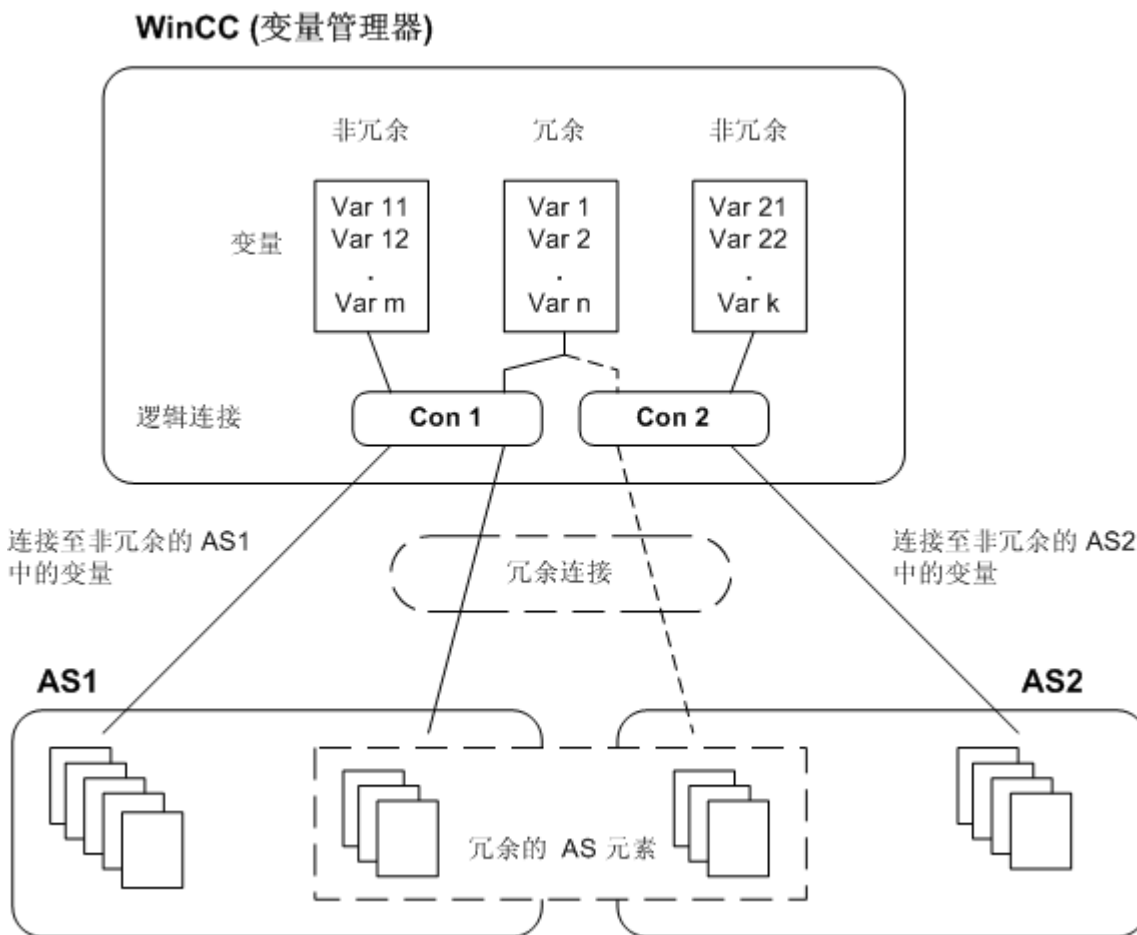
引言

软件冗余通过两个自动化系统 S7-300 或 S7-400 的冗余连接，提供一种经济的方式来监控系统中与安全性相关的、无时间紧迫性的部分。

说明

该通道的软件冗余不具有与 SIMATIC S7-400 H 的 H 层冗余相同的功能。

需要 AS 中的组态和 WinCC 中的组态来实现功能。



AS

如果两个自动化系统之间存在冗余连接，那么如果一个 AS 产生故障，另一个自动化系统可以接管监控功能。监控会覆盖整个过程或只是其中部分过程。

除了应用程序外，两个自动化系统上还安装了软件冗余程序包。该程序包不包括在 WinCC 交付范围内。为了匹配数据，需要通过 MPI、PROFIBUS-DP 或工业以太网，甚至可以使用已存在的通讯链接，在两个自动化系统间建立冗余连接。

WinCC

也可在相同通道单元的连接中组态冗余连接。组态时，只组态一个称为主连接的连接。只有通过动态向导“建立冗余连接”才能插入备用连接。

该向导同时还插入连接专用的内部变量和脚本。它控制运行期间连接之间的切换，以及相应的消息。

运行期间，可使用脚本在出错时在连接之间自动进行切换。但也可以不用脚本，而通过描述连接专用的内部变量“@<connectionname>@ForceConnectionAddress”来手动切换。

对于 AS2，如果要使用的数据来自 WinCC 中 AS2 的非冗余部分，则需要第二个固定组态连接。

使用软件冗余并不意味着只能将硬组态连接用于冗余。各个单个连接即使没有建立冗余，也都可以使用。

参见

如何在 WinCC 中清除软件冗余 (页 469)

如何组态软件冗余 (页 466)

软件冗余 - 连接专用的内部变量 (页 462)

软件冗余 - 连接专用的内部变量

引言

连接专用的内部变量用于控制冗余连接。这些变量使用动态向导“冗余连接组态”进行建立，并被集中到相关连接中的称之为“@<连接名称>”的变量组中。

这些变量可用于确定连接状态和控制连接的建立。因此，它们也可用于其它用途，例如，通过单个连接对多个 PLC 进行寻址。然而，如果在没有软件冗余程序包时要使用这些变量，则必须手工创建。

名称格式

连接专用内部变量的名称由关联连接的名称和标识符组成。

"@<连接名称>@<标识符>"

连接名称的前缀为"@", 用于将其识别为系统变量。连接名称前放置有一个标识符作为分隔符。

实例: "@CPU_3@ConnectionState"

<连接名称> = CPU_3

<标识符> = ConnectionState

说明

连接指定的内部变量作为外部变量计数（每个连接有 8 个外部变量）。

当相关的连接已准备好时，WinCC 的数据管理器只允许访问外部变量。然而，不管连接状态如何，都可以写入和读取指定连接的内部变量。

在运行系统中，可以使用“WinCC 通道诊断”工具调用一些连接专用内部变量的当前值。当选择了主连接时，变量将显示在“计数器”列中。此外，在 WinCC 项目管理器的变量管理器中，可以将变量的当前值显示为工具提示。

变量概述

下列标识符可用于软件冗余的内部变量：

ConnectionState

含义	连接状态 该变量可用于确定当前的连接状态。
类型	双字
访问	读
缺省值	0 = “错误”
数值	0 = 连接错误 1 = 连接准备就绪 2 = 连接冗余（只用于 H 系统出现冗余时）

ConnectionError

含义	错误原因 该变量包含对错误原因的描述。缺省 = 0，也就是说还没有建立连接或无错。建立连接时，再次给变量加载 0（无错）。从通道角度解释错误代码。S7 通道通过该变量传递 SIMATIC 设备驱动程序的错误代码。
类型	双字
访问	读
缺省值	0 = “无错”
数值	0 = 无错误 <> 0 = S7 错误代码

ConnectionErrorString

含义	错误原因字符串 该变量包含连接错误原因字符串。字符串以目前选定的语言输出。缺省 = “”，也就是说还没有建立连接或无错。在 S7 通道中，下列文本都以“英语”的方式输出，与所选的语言无关。
类型	TEXT8 [128]
访问	读
缺省值	“” = “无错”
数值	“No Error” “Error hhhh”= 产生错误 hhhh（这里 hhhh = 十六进制的 S7 错误代码）

ConnectionErrorCount

含义	连接错误计数器 每发生一次连接错误，该变量的值增加 1。溢出后，从 0 开始计数。
类型	双字
访问	读
缺省值	0

ConnectionEstablishMode

含义	连接建立模式 该变量可用于设置建立连接的自动模式。如果将其启用，每隔 4 秒左右，S7 通道试图重新建立失败了的连接。如果该变量的值等于 0，则在故障出现之后，不会在每 4 秒之后尝试重新建立连接，而是保持断开状态。
类型	双字
访问	写
缺省值	1
数值	写入变量会引起以下动作： 0 = 手动连接建立模式 动作：取消激活自动连接建立 <>0 = 自动连接建立模式 动作：激活自动连接建立模式

ForceConnectionState

含义	首选连接状态 该变量可用于通知通道首选连接的状态。通常该变量取数值 1，也就是说通道试图建立连接（如果适用，间隔大约为 4 秒）。如果写入变量的数值是 0，通道中断连接。
类型	双字
访问	写
缺省值	1
数值	写入到变量的数值具有下列含义： 0 = 首选连接状态：中断建立 动作：如果已建立连接，则启动断开连接 1 = 优先考虑的连接状态：中断建立 动作：如果已断开连接，则启动建立连接

ForceConnectionAddress

含义	选择连接地址 该变量定义将要用于建立连接的连接地址。
类型	双字

访问	写
缺省值	0
数值	<p>如果 ConnectionEstablishMode 设置为“自动”，则将自动建立到相应地址的连接。</p> <p>写入变量会引起以下动作：</p> <p>0 = 通过已组态的连接参数进行连接</p> <p>动作：如果 @ForceConnectionAddress 已设置为 1，则启动断开连接。</p> <p>1 = 通过其它连接参数进行连接</p> <p>动作：如果 @ForceConnectionAddress 先前为 0，则启动断开连接。</p>

AlternateConnectionAddress

含义	<p>其它连接地址</p> <p>在该变量中，可以输入其它连接地址字符串。该字符串与将在 WinCC 项目管理器中显示的连接参数字符串相同。该字符串专用于通道。系统启动（运行时）时，此处输入组态的地址，作为 S7 通道的缺省设置。如果还没有组态地址，则为 S7 通道输入文本“非法地址”。</p> <p>通过 MPI 且站地址为 3 的 S7-AS 的详细地址实例：“MPI,3 0,,0,0,02”</p>
类型	TEXT8 [255]
访问	写
缺省值	“...” = 组态的地址
数值	<p>写入变量会引起以下动作：</p> <ul style="list-style-type: none"> - 如果由于写操作而更改了地址，则连接断开并返回设置“通过其它连接参数进行连接”。 - 如果连接模式设置为“自动”，则根据刚写入的地址自动建立连接。

如何组态软件冗余

简介

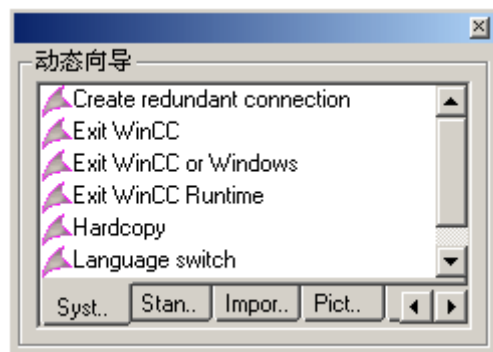
本节说明如何在 WinCC 中为通道“SIMATIC S7 Protocol Suite”的连接组态软件冗余。PLC 也必须进行组态，以便使用该功能，但本文档将不对此进行描述。

要求

1. 通道“SIMATIC S7 Protocol Suite”必须集成在项目中。
2. 必须在该通道的一个通道单元中创建连接，并为其组态冗余连接。

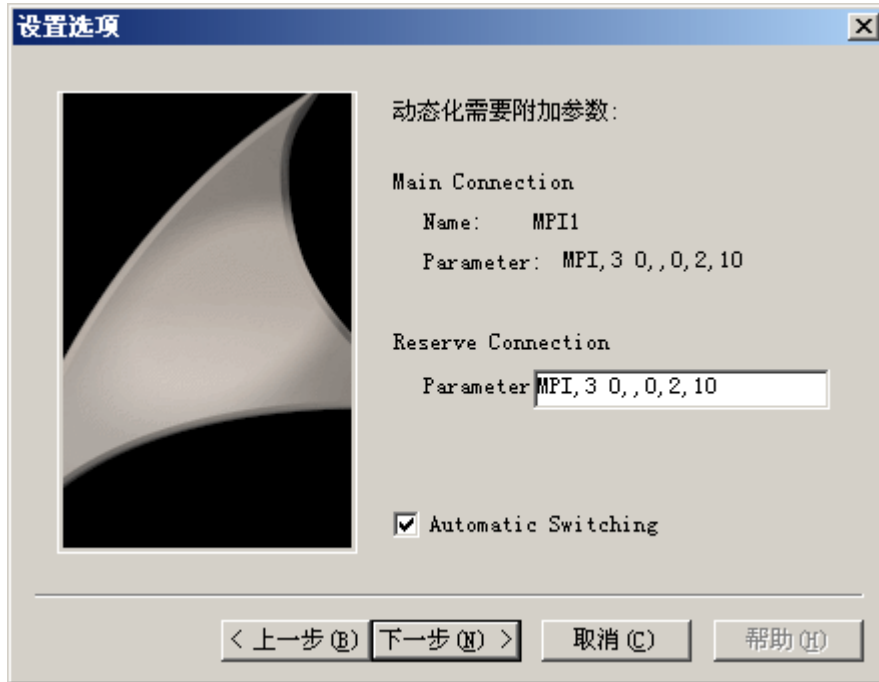
步骤

1. 在计算机的启动参数中，激活“全局脚本运行系统”、“报警记录运行系统”和“图形运行系统”模块。
有关详细信息，请参见“检查启动参数”。
2. 将 WinCC 系统消息载入报警记录。这些系统消息中包含有关软件冗余的消息。
可以在“读取报警记录中的 WinCC 系统消息”下找到有关该主题的附加信息。
3. 在图形编辑器中打开画面。在“动态向导”窗口中，选择“系统功能”标签。双击以启动动态向导“设置冗余连接”。



4. 向导的使用步骤在“欢迎”中有简短的描述。单击“下一步”，打开“设置选项”对话框。
5. 选择将用作主连接的连接，然后单击“下一步”。
向导现在将创建连接专用的内部变量，并将其保存到主连接下的变量组“@”中。

6. 输入 PLC 的地址，在“参数”域中将建立与该 PLC 之间的备用连接。
选中“自动切换”复选框以使向导为自动切换连接生成脚本。
单击“下一步”
冗余的图形（显示与两个 PLC 的 MPI 连接）将出现在画面中：



7. 在“完成！”对话框中，将再次显示所进行的全部设置。如果要进行任何更正，只需单击“返回”即可。单击“完成”。
向导随即会生成一个脚本，并将其保存在全局脚本编辑器的目录“C-Editor \ Actions \ Actions : < 计算机名称>”中的“@<连接名称>.pas”下。

说明

在下面的步骤中，将使用“建立冗余连接”向导。当完成步骤 6 时，该向导生成连接专用的内部变量。如果此时向导被取消或没有以单击“完成”来结束过程，则这些变量将保持不变。

参见

如何将 WinCC 系统消息载入报警记录 (页 470)

如何检查 WinCC 启动参数 (页 469)

如何在 WinCC 中清除软件冗余

引言

本节说明如何在 WinCC 中为通道“SIMATIC S7 Protocol Suite”的连接删除软件冗余。PLC 也必须进行组态，以便使用该功能返回到非冗余连接，但本文档不对此进行描述。

要求

- 必须取消激活 WinCC 项目。

步骤

可按两个步骤删除软件冗余：

- 在“变量管理器”中删除变量组“@<计算机名称>”及其变量。
- 在“全局脚本”中删除脚本“@<连接名称>.pas”。

步骤

1. 在变量管理器中，为软件冗余选择将组态为主连接的连接。它包括变量组“@<连接名称>”及软件冗余的连接专用的内部变量。删除该变量组。
2. 删除用于动作“@<连接名称>.pas”的脚本。为此，打开“全局脚本”中的 C 编辑器。则会显示多个子目录。
3. 选择“Actions \ Action : " directory. <计算机名称>”。在数据窗口中，删除“动作”类型的脚本“@<连接名称>.pas”。
4. 关闭“全局脚本”编辑器。

如何检查 WinCC 启动参数

步骤

1. 在 WinCC 项目管理器中，选择“计算机”。
2. 打开快捷菜单并选择“属性”。
3. 将打开“计算机列表属性”对话框。单击“属性”按钮。
4. 将打开“计算机属性”对话框。
5. 单击“启动”标签并选中条目。必要时，可以激活或取消激活运行模块，或将其它应用程序添加到启动列表中。
6. 关闭打开的对话框。

如何将 WinCC 系统消息载入报警记录

简介

本节说明如何将 WinCC 系统消息载入项目。

步骤

1. 打开报警记录。
2. 在导航区域中选择“系统消息”节点。
3. 可以针对在表格区域或“属性”(Properties)区域中使用的系统消息激活“已使用”(Used)选项。
4. 在“系统消息”节点的快捷菜单中选择“已使用更新”(Update used)命令。

当连接受干扰时的错误代码

“S7CHNdeu.chm”文件包含错误代码列表。

此文件位于安装路径 \SIEMENS\WinCC\bin 下。

错误代码的文档

6.12 SIMATIC S7-1200, S7-1500 Channel

6.12.1 “SIMATIC S7-1200, S7-1500 Channel”通道

“SIMATIC S7-1200, S7-1500 Channel”用于 WinCC 站与自动化系统 S7-1200 和 S7-1500 之间的通信。

通信采用 TCP/IP 协议。

通道还支持冗余 SIMATIC S7-1500R/H 系统。更多相关信息，请参见“通道组态 > 冗余系统 S7-1500R/H (页 491)”。

通道单元

“SIMATIC S7-1200, S7-1500 Channel”具有“OMS+”通道单元。

诊断通道

要显示运行系统中控制器的故障和错误，请使用 WinCC 系统诊断控件。

可在“通信诊断 > 诊断通道 SIMATIC S7-1200/S7-1500 (页 582)”(Communication Diagnostics > Diagnostic channel SIMATIC S7-1200/S7-1500) 下了解详细信息。

建议的通信处理器

在 WinCC 站与自动化系统 S7-1200 或 S7-1500 进行通信时，建议您使用下列通信处理器：

- CP 1612 A2
- CP 1613 A2
- CP 1623
- CP 1628

参见

通过系统诊断控件进行系统诊断 (页 582)

冗余系统 S7-1500R/H (页 491)

6.12.2 所支持的数据类型概述

简介

在组态变量时指定数据类型以及针对自动化系统 (AS) 中的数据格式的格式调整。

下表显示了通道支持的数据类型和类型转换的使用。

支持的数据类型/格式调整

数据类型	类型转换
二进制变量	否
有符号 8 位数	是
无符号 8 位数	是
有符号 16 位数	是
无符号 16 位数	是
有符号 32 位数	是
无符号 32 位数	是
浮点数 32 位 IEEE 754	是
64 位 IEEE 754 浮点数	是
文本变量, 8 位字符集	否
文本变量, 16 位字符集	否
原始数据变量	否
日期/时间	是

6.12.3 组态通道

6.12.3.1 组态“SIMATIC S7-1200, S7-1500 Channel”通道

简介

在 WinCC 中, WinCC 与自动化系统的通信需要逻辑连接。

本节介绍如何组态 "SIMATIC S7-1200, S7-1500 Channel"。

要建立通信通道，请在变量管理器的快捷菜单中选择“添加新的驱动程序 > SIMATIC S7-1200, S7-1500 Channel”(Add new driver > SIMATIC S7-1200, S7-1500 Channel)。

连接参数

S7 网络地址

地址取决于所选产品系列：

- 对于 S7-1200 或 S7-1500 产品系列，输入 TCP/IP 连接的 IP 地址。
- 对于 WinAC S7-1500 产品系列，输入 S7-1507S 的站地址。

请注意，访问点必须引用适用于所选产品系列的接口。

TCP/IP 连接

使用 TCP/IP 协议时，必须定义自动化系统的 IP 地址以实现逻辑连接。

IP 地址由四个以圆点分隔的数值组成。数值必须位于“0-255”的范围内。

说明

超时行为

使用 TCP/IP 协议时，不会立即检测到中断的连接。

回查消息可能耗时达一分钟。

站地址

站地址位于 Profibus 地址的数值范围内。

组态 S7-1507S 时，可在“索引”(Index) 下的“属性”(Properties) 对话框中找到站地址。

将访问点选为接口“PC 内部（本地）”(PC Internal (local))。

使用密码保护连接访问

对于使用“SIMATIC S7-1200, S7-1500 Channel”通道的连接，可使用密码保护对自动化系统的访问。

在自动化系统中针对此访问保护定义 1 级、2 级和 3 级。

在组态 WinCC 期间应用所需等级的已组态密码。

如未设置任何密码，则将自动采用 AS 中组态的等级。

指定并确定连接状态

可在内部变量组“ConnectionStates”中为每条连接创建以下系统变量：

- 建立/终止连接：
@<Connectionname>@ForceConnectionStateEx
- 查询连接状态：
@<Connectionname>@ConnectionStateEx

有关连接状态的更多信息，请参见《WinCC 信息系统》中的“WinCC 过程通信 > 在运行系统中为连接状态组态变量 (页 155)”。

软件冗余

如果使用 S7-1500R/H 的软件冗余，请创建以下变量：

- 建立/终止冗余连接：
@<Connectionname>@ForceConnectionState
- 查询冗余连接状态：
@<Connectionname>@ConnectionState

软件冗余的系统变量在“S7-1500R/H 的软件冗余 (页 496)”下进行说明。

组态变量

对于通过通道 "SIMATIC S7-1200, S7-1500 Channel" 实现的 WinCC 和自动化系统之间的连接，可在 WinCC 中创建多种数据类型的变量。用户可组态相应连接的过程变量，或将自动化系统的 AS 符号加载到 WinCC 变量管理器中。

变量的组态根据在自动化系统数据区中的寻址而定。

说明

仅可在 TIA Portal 中进行 AS 组态

只能在 TIA Portal 中对自动化系统的组态进行更改。

覆盖 HMI 系统或 Web 服务器的数据

在运行期间，HMI 系统或 Web 服务器的数据可能会被 S7-1500 中的数据覆盖。运行期间，两个过程（PLC 系统和 HMI 系统）并行运行，但相互独立。

如果两个系统都尝试对同一变量进行写访问，数据可能在系统端遭到覆盖。

其他信息在 Internet 中提供：

- 条目 ID 为 109478253 的常见问题解答：“为什么 HMI 系统或 Web 服务器的数据有时被 S7-1500 中的数据覆盖？ (<https://support.industry.siemens.com/cs/cn/zh/view/109478253>)”

参见

在运行系统中为连接状态组态变量 (页 155)

组态原始数据变量 (页 475)

S7-1500R/H 的软件冗余 (页 496)

<https://support.industry.siemens.com/cs/ww/en/view/109478253> (<https://support.industry.siemens.com/cs/cn/zh/view/109478253>)

6.12.3.2 组态原始数据变量

简介

"SIMATIC S7-1200, S7-1500 Channel"通道支持“原始数据变量”数据类型。

通道“SIMATIC S7-1200, S7-1500 Channel”中的原始数据变量

作为字节数组的原始数据变量用于在 WinCC 和 PLC 之间传送用户数据块，并且只处理用户数据。

原始数据变量仅支持控制器非循环读取服务，例如：通过 C 脚本请求变量。

"SIMATIC S7-1200, S7-1500 Channel"通道不支持原始数据变量的循环读取服务。

原始数据变量寻址

作为字节数组的原始数据变量在通道中将和通常的过程变量一样进行处理，通过数据区的地址和长度可对其进行寻址（例如，DB 1、DBB 10、长度 100 字节）。

如果 WinCC 变量的数据类型是“原始数据变量”，仅可使用“字节”。除原始数据范围长度外，参数均为预设值且不可更改。

数据块的长度

请遵守可由通信驱动程序发送的最大数据块长度：

- S7-1200 / S7-1500：数据块长度最大为 8000 字节

交换大数据量

如果可以使用原始数据将大量数据从控制器传送到 WinCC，则在应用示例中写入 37873547：

- 在 S7-300/400/1500 和 WinCC 之间交换大量数据 (<https://support.industry.siemens.com/cs/ww/zh/view/37873547>)

参见

组态“SIMATIC S7-1200, S7-1500 Channel”通道 (页 472)

在 S7-300/400/1500 和 WinCC 之间交换大量数据 (<https://support.industry.siemens.com/cs/ww/zh/view/37873547>)

6.12.3.3 如何组态连接

简介

组态 "SIMATIC S7-1200, S7-1500 Channel" 的必要步骤为：

1. 组态连接
2. 组态变量

要求

- "SIMATIC S7-1200, S7-1500 Channel" 的通信驱动程序已安装并集成到项目中。
- SIMATIC 项目已组态，并且在自动化系统中可用。

步骤

1. 在 WinCC 项目管理器的“变量管理”编辑器中打开 "SIMATIC S7-1200, S7-1500 Channel" 通信驱动程序的菜单结构。
2. 在通道单元“OMS+”的快捷菜单中选择“新建连接”(New connection) 条目。
3. 输入连接的名称。
4. 在连接快捷菜单中选择“连接参数”(Connection parameters) 条目。
随即打开“连接”(Connection) 对话框。
5. 从“产品系列”(product family) 中选择自动化系统。
6. 输入自动化系统的 IP 地址或站地址。
7. 选择“访问点”(Access point)。
访问点必须引用属于所选产品系列的接口。

8. 输入自动化系统的访问保护密码:
 - 单击“更改”(Change)。
 - 输入所需等级的密码。
 - 在“确认密码”(Repeat password) 字段中重复输入密码。
9. 单击“确定”(OK) 关闭对话框。
10. 要为建立连接和连接状态创建系统变量，请从连接的快捷菜单中选择“创建变量以激活/禁用”(Create tags for activation/deactivation) 条目。
在内部变量组“ConnectionStates”中创建下列变量：
 - @<Connectionname>@ForceConnectionStateEx
 - @<Connectionname>@ConnectionStateEx

S7-1500 软件控制器的限制

由于 S7-1507S 与 Simatic-Net 不兼容，因此目前不支持在软 PLC 中安装 Simatic-Net。这表示，缺少 Simatic-Net，因此无法通过附加通道与外部自动化系统进行连接。可使用 OPC UA 等其它通道。

参见

在运行系统中为连接状态组态变量 (页 155)

6.12.3.4 如何组态没有优先块访问的变量

简介

本节介绍如何在 WinCC 中组态没有对自动化系统中地址区域进行优先块访问的变量。

要求


- 在 TIA Portal 中将数据块的属性“优先块访问”(Optimized block access) 禁用。
- "SIMATIC S7-1200, S7-1500 Channel" 必须已集成到项目中。
- 必须在“OMS+”通道单元中创建了连接。

组态 8 位文本变量的注意事项

对于 "SIMATIC S7-1200, S7-1500 Channel" 中的 8 位文本变量，WinCC 仅支持 S7 字符串类型，该类型由一个控制字和用户字符串数据组成：

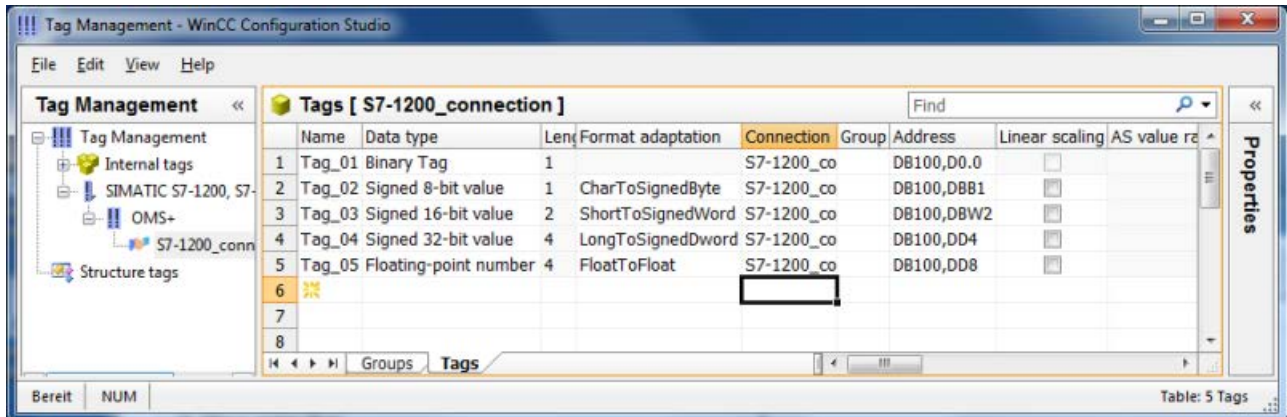
- 要在 WinCC 中组态 8 位文本变量，请在用户数据之前输入自动化系统 (AS) 存储器中存在的控制字的地址。
控制字的第一个字节包含字符串自定义的最大长度，第二个字节包含实际长度。
- 在自动化系统存储器中创建数据结构时，必须注意 WinCC 中为 8 位文本变量组态的长度要增加控制字的 2 个字节。
如果在存储器中直接逐个创建 8 位文本变量的数据结构，则后面的数据将会被覆盖。
- 在进行读操作时，控制字将和用户数据一起被读取，并将判断第二个字节中的当前长度。只有长度与第二控制字节中包含的当前长度一致的用户数据才传送到 WinCC 的 8 位文本变量。
- 在进行写操作时，字符串的实际长度是确定的（“0”字符），且具有当前长度的控制字节将和用户数据一起发送给自动化系统。

步骤

1. 选择所需连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 在“名称”(Name) 列最上方的空单元格中输入变量名。
在表格区或右侧的“属性 - 变量”(Properties - Tags) 数据区组态以下设置。
4. 选择一种支持的数据类型。
5. 在“地址”(Address) 列单击  按钮。
6. 输入变量地址。
7. 如果变量具有质量代码并且您要在 WinCC 中使用该代码，则选中“质量代码”(Quality code) 复选框。
该代码还必须存在于自动化系统中。
只有选择了“DB”数据区才能激活此复选框。
8. 单击“确定”(OK) 关闭对话框。

结果

在变量管理中组态无优先块访问的变量。



6.12.3.5 如何组态具有优先块访问的变量

简介

本节介绍如何在 WinCC 中组态具有对自动化系统中地址区域进行优先块访问的变量。

将变量从控制器导入 WinCC 项目中。

无法使用加载在线更改

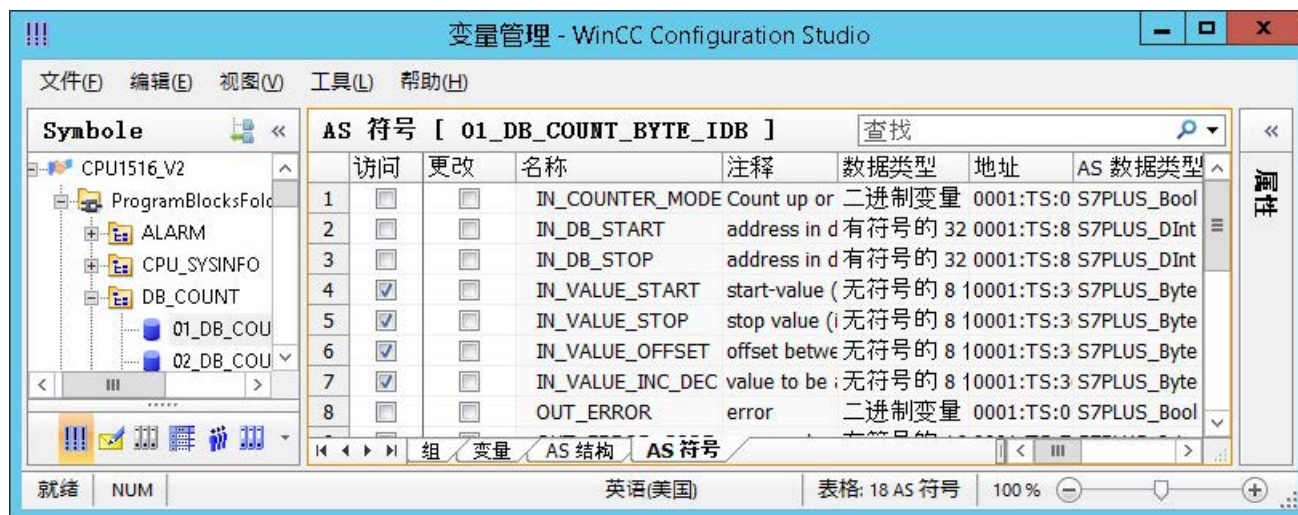
无法在运行系统中使用加载在线更改传递通过“加载到 AS”(Load to AS) 创建为 AS 符号的变量。

要求

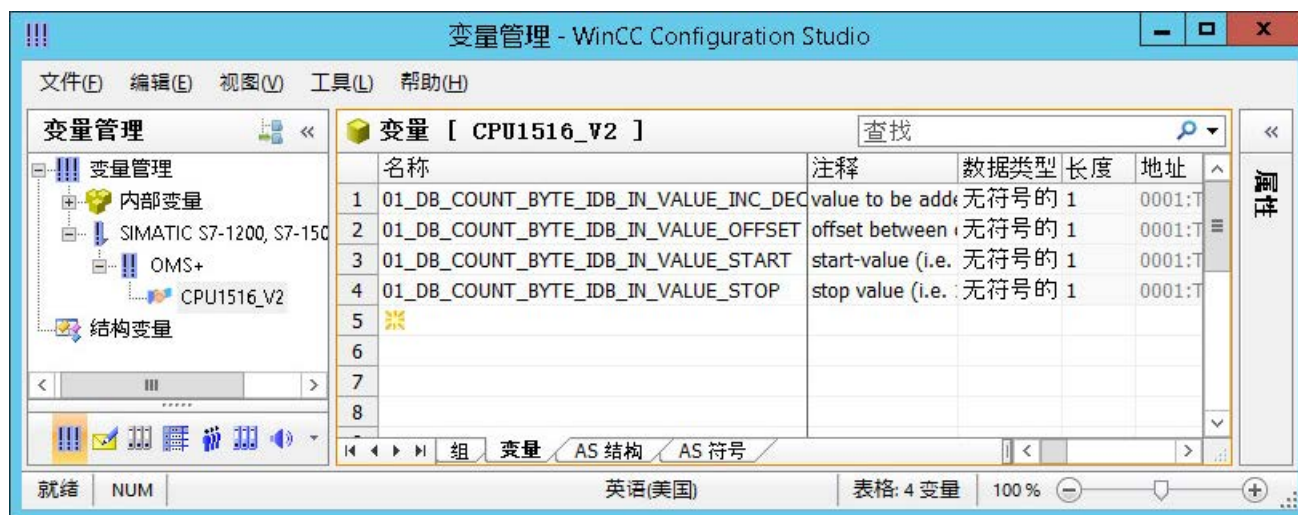
- 在 TIA Portal 中将数据块的属性“优先块访问”(Optimized block access) 激活。
- "SIMATIC S7-1200, S7-1500 Channel" 必须已集成到项目中。
- 必须在“OMS+”通道单元中创建了连接。
- 必须在运行系统中建立了连接。

操作步骤

1. 选择所需连接。
2. 在连接快捷菜单中选择“AS 符号 > 从 AS 读取”(AS Symbols > Read from AS)。将加载可用控制器数据，“系统”(Symbols) 视图随即会打开。加载的数据显示在表格区的“AS 符号”(AS Symbols) 选项卡中。如果加载的数据还包含结构，还会额外显示“AS 结构”(AS structures) 选项卡。



3. AS 符号不会自动包含到变量管理中。要将所需 AS 符号传送到“变量”(Tags) 选项卡，需激活“访问”(Access) 列中相应的复选框。变量管理中不包含选定的变量。



在不连接控制器的情况下编辑 AS 符号

可以在不连接到控制器的情况下对 AS 符号进行离线组态。

为此，请将已加载的 AS 符号保存到文件中。

1. 选择所需连接。
2. 在连接快捷菜单中选择“AS 符号 > 保存到文件”(AS Symbols > Save to file)。

然后即可将 AS 符号加载到变量管理的离线项目中。

1. 选择所需连接。
2. 在连接快捷菜单中选择“AS 符号 > 从文件加载”(AS Symbols > Load from file)。

有关离线组态的更多信息，请参见如何离线下下载 AS 符号 (页 482)。

使用控制器同步 WinCC 变量

从控制器或文件进行加载后，变量管理检查 AS 符号的属性。

地址、数据类型和变量名和 WinCC 项目中的 AS 符号属性进行比较。

- 如果符号的属性不匹配，“AS 符号”(AS Symbols) 选项卡的“已修改”(Modified) 字段会激活。
相应的属性字段会以红色突出显示。字段的工具提示包含额外信息。
- 如果未在控制器中找到 WinCC 变量，连接的 AS 符号的整行用红色突出显示。

例如，在下列情况下会发生此问题：

- 使用 WinCC V7.3 创建 WinCC 项目。移植的项目还未包含所有同步的信息。
- AS 符号的地址已在控制器中改变，例如，由于 TIA Portal 的组态变更。
- AS 符号的数据类型或名称已被更改。
- 已在控制器中删除了 AS 符号。

再次导入变量

要同步属性，更新 WinCC 项目中使用的 AS 符号。

按以下步骤执行操作：

1. 在“AS 符号”(AS Symbols) 选项卡中选择已修改的 AS 符号。
要更新移植项目，选择所有行。
2. 取消激活“已修改”(Modified) 字段。

会再次读取 AS 组态的参数并在变量管理器中使用。

说明

移植 TIA Portal 项目前：更新 AS 符号

更新 TIA Portal 版本时，请遵循以下顺序进行操作：

1. 更新所有用作 WinCC 变量的 AS 符号。
2. 移植 TIA Portal 项目。
3. 将控制器加载到 TIA Portal 中。
4. 再次更新所有用作 WinCC 变量的 AS 符号。

该操作确保进行加载后，WinCC 变量到 AS 符号的分配保持在 WinCC 项目中。

否则可能由于分配不再是最新的而无法读取变量。

参见

如何导出 AS 项目数据 (页 489)

如何组态 AS 结构 (页 487)

6.12.3.6 如何离线下载 AS 符号

简介

可离线组态以下 S7 通道：

- SIMATIC S7 Protocol Suite
- SIMATIC S7-1200, S7-1500 Channel

为此，可从现有 TIA Portal 项目中导出数据记录，并将导出文件加载至 WinCC 项目。

支持的导出格式

导入时支持以下文件格式：

格式	内容	说明
*.bin	二进制数据	从 WinCC 变量管理器导出： <ul style="list-style-type: none"> “变量管理器”(Tag Management) 视图 > 连接的快捷菜单： AS 符号 > 保存到文件 (AS Symbols > Save to file) 不受“SIMATIC S7 Protocol Suite”通道支持。
*.sdz	结构化导出	从 WinCC 变量管理器导出： <ul style="list-style-type: none"> “符号”(Symbols) 视图 > 菜单： 编辑 > 导出 (Edit > Export) 同样会从导航区导出结构信息。
*.zip	TIA Portal 导出文件	通过“SIEMENS SIMATIC SCADA Export”工具从 TIA Portal 导出

用于 TIA Portal 的“SIEMENS SIMATIC SCADA Export”

要从 TIA Portal 项目中导出数据记录，请使用“SIEMENS SIMATIC SCADA Export”工具。

在 TIA Portal 项目中，从 PLC 的快捷菜单中选择“导出到 SIMATIC SCADA”(Export to SIMATIC SCADA) 条目。

可以在工业在线支持中下载适用于各种 TIA Portal 版本的工具：

- “SIMATIC SCADA Export”下载 (ID 109748955) (<https://support.industry.siemens.com/cs/ww/zh/view/109748955>)
- “SIMATIC SCADA Export”文档 (ID 101908495) (<https://support.industry.siemens.com/cs/ww/zh/view/101908495>)

要求

- 已在 TIA Portal 中编译 AS。
- 相应的 PLC 组态数据已导出且可用，例如以 .zip 文件形式提供。
- 已在 WinCC 项目中安装了通信处理器和相关硬件驱动程序。
- 已在“SIMATIC S7-1200, S7-1500 Channel”或“SIMATIC S7 Protocol Suite”中创建了连接。
- 已打开“变量管理”编辑器。

操作步骤

1. 在连接快捷菜单中选择“AS 符号 > 从文件加载”(AS Symbols > Load from file)。



2. 选择要加载的所需数据记录。
加载可用的控制器数据。

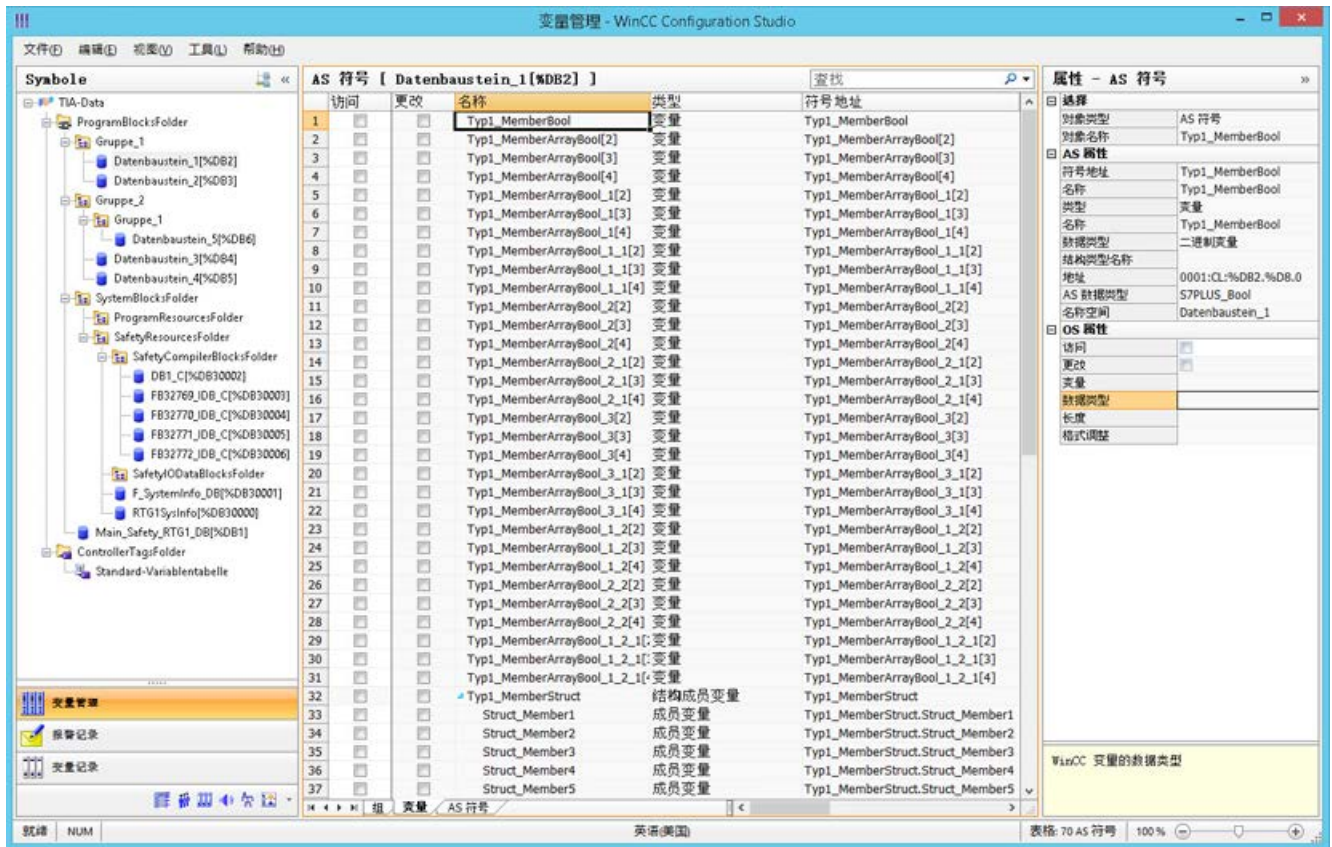
结果

已导入组态，“符号”(Symbols) 视图打开。

加载的数据显示在表格区的“AS 符号”(AS Symbols) 选项卡中，且可用于其它处理操作。

如果加载的数据还包含结构，还会额外显示“AS 结构”(AS structures) 选项卡。

编辑器关闭后，“AS 符号”(AS Symbols) 和“AS 结构”(AS structures) 选项卡会再次隐藏。



符号显示

在“变量管理器”(Tag Management)中，可使用以下按钮在默认视图和“符号”(Symbols)视图之间进行切换：

该按钮仅在加载数据记录后才可用。

导航区域

结构树中的数据表示对应于 TIA Portal 的层级。

表格区域

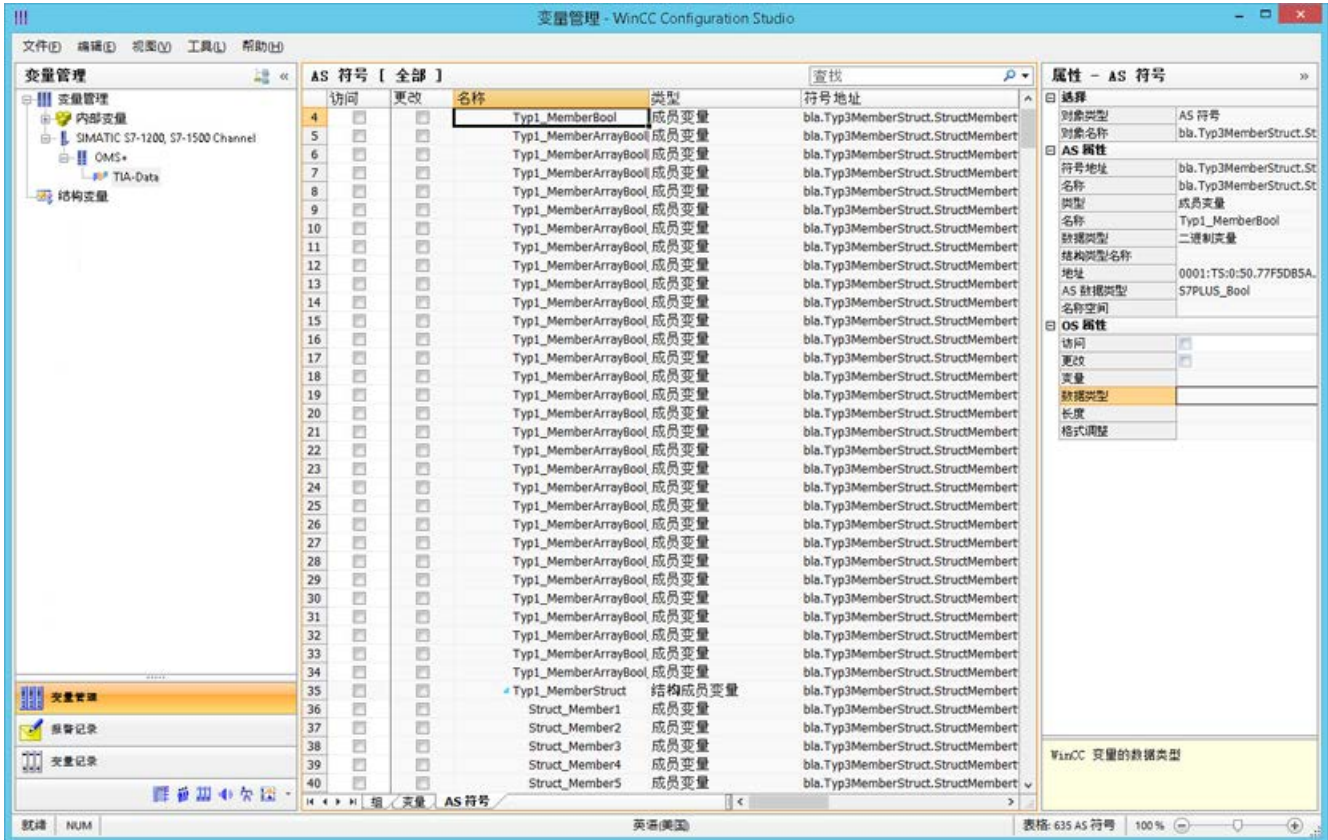
找到的 WinCC 变量与 AS 变量不匹配时，会自动选中“已修改”(Modified) 列的复选框。因此，可以按照这些内容进行过滤。

通过选中“访问”(Access) 列的复选框，可从找到的各个 AS 变量创建一个 WinCC 变量。

变量管理中的 AS 符号

还可通过“AS 符号”(AS Symbols) 选项卡访问变量管理中的 AS 符号。

与数据块特定的“符号”(Symbols)视图不同，所有可用的控制器变量均会在此处显示。该视图还显示了之前已组态但现在不再位于 AS 中的变量。



参见

如何导出 AS 项目数据 (页 489)

如何组态 AS 结构 (页 487)

“SIMATIC SCADA Export”文档 (ID 101908495) (<https://support.industry.siemens.com/cs/ww/zh/view/101908495>)

“SIMATIC SCADA Export”下载 (ID 109748955) (<https://support.industry.siemens.com/cs/ww/zh/view/109748955>)

6.12.3.7 如何组态 AS 结构

简介

如果加载 AS 符号，还会导入控制系统的结构。

该过程取决于通信通道：

- SIMATIC S7 Protocol Suite:
 - 从文件加载
- SIMATIC S7-1200, S7-1500 Channel
 - 从文件加载
 - 从 AS 加载

变量管理器中的 AS 结构

AS 结构显示在默认视图以及“符号”(Symbols) 视图的“AS 结构”(AS structures) 选项卡中。

在 WinCC 中可如下使用 AS 结构：

- 为 AS 结构变量创建 WinCC 结构类型。
结构在 WinCC 变量管理器中的“结构变量”(Structure tags) 下创建为结构类型。
还会为每个包含的“变量类型成员”创建结构类型元素。
- 为 AS 结构变量分配 WinCC 结构类型。
然后，为每个“变量类型成员”选择已选结构类型的结构类型元素。

可在变量管理器中更改 WinCC 结构类型的名称和结构类型元素。AS 结构的分配会自动调整。

要求

- 可通过下列方法之一访问 PLC 的组态数据：
 - 在运行系统中建立与 PLC 的连接。
 - 导出的组态数据以压缩文件等形式提供。
- 已在“SIMATIC S7-1200, S7-1500 Channel”或“SIMATIC S7 Protocol Suite”中创建了连接。

操作步骤

1. 通过“从 AS 读取”(Read from AS) 或“从文件加载”(Load from file) 加载 AS 符号。
已加载的消息显示在“符号”(Symbols) 变量管理器视图中。
加载的结构显示在“AS 结构”(AS structures) 选项卡上。
从 AS 加载时，不会传送结构名称。ID 显示为“结构变量类型”的名称。

Symbole		AS 结构	
WinCC 结构	名称	名称	类型
1		▶ SimData1UDT	结构变量类型
2		▶ LEN_UDT1	结构变量类型

2. 单击“AS 结构”(AS structures)。
要显示结构的元素，请单击结构名称前面的箭头。
3. 选择结构的整个行，并在快捷菜单中选择“创建结构”(Create structure) 条目。

AS 结构		
WinCC 结构	名称	类型
1	▶ SimData1UDT	结构变量类型
2	▶ BitData	变量类型成员
3	▶ IntData	变量类型成员
4	▶ RealData	变量类型成员
5	▶ ByteData	变量类型成员
6	▶ LEN_UDT1	结构变量类型
7		变量类型成员
8		结构变量类型
9		结构变量类型
10		
11		
12		
13		
14		
15		

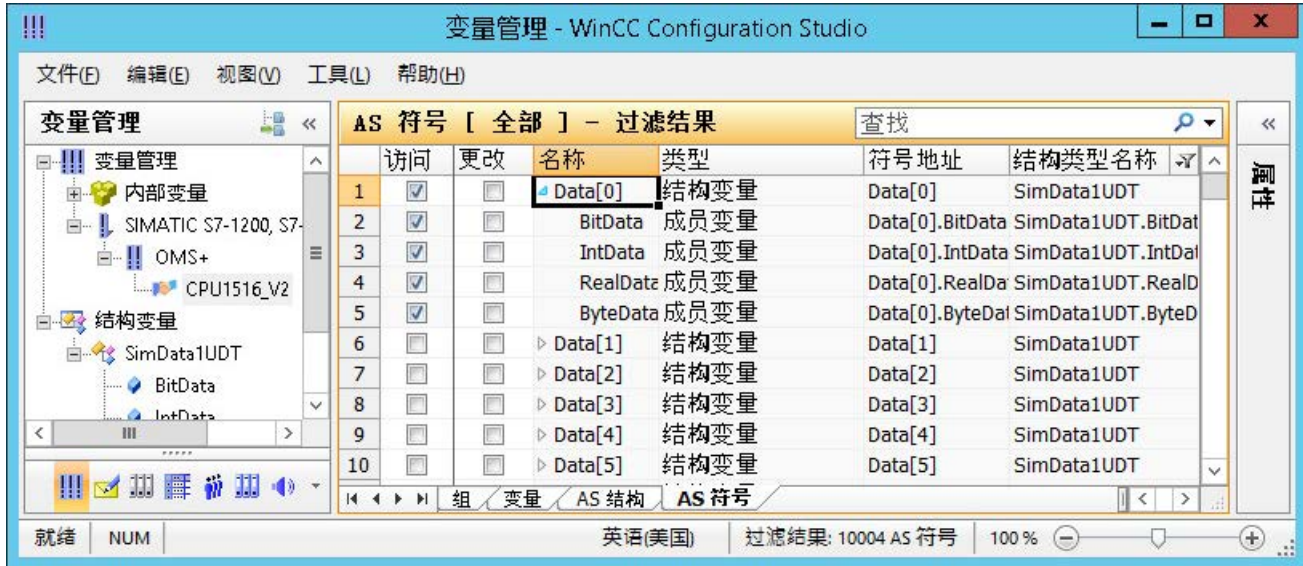
或者选择一个已在 WinCC 变量管理器中创建的结构类型。
然后将结构类型元素分配给“变量类型成员”。

AS 结构		
WinCC 结构	名称	类型
1	▶ SimData1UDT	结构变量类型
2	SimData1UDT	
3	LEN_UDT1	
4	WinCC_StrucType1	

会在 WinCC 变量管理器中为 AS 结构的每个“结构变量类型”创建结构类型。
会为每个“变量类型成员”创建结构类型元素。

4. 在“变量管理器”(Tag Management) 视图中选择“AS 符号”(AS Symbols) 选项卡。

5. 如果要只显示结构变量和成员变量，请在“结构类型名称”(Structure type name) 列中筛选出所需 AS 结构。



6. 要访问 WinCC 变量管理器中的 AS 结构变量，请激活“访问”(Access) 字段。
包含的成员变量会自动激活。
AS 结构变量将在 WinCC 变量管理器中创建为结构变量。

结果

通过 WinCC 变量管理器中的结构类型和结构变量，可访问 AS 结构变量。

举例来说，可通过这种方式访问 WinCC 面板类型中的 AS 结构，并在面板实例中表示这些结构。

参见

- 如何离线下载 AS 符号 (页 482)
- 如何导出 AS 项目数据 (页 489)
- 如何组态具有优先块访问的变量 (页 479)

6.12.3.8 如何导出 AS 项目数据

导出 AS 符号

可使用导出文件进行离线组态。

可将 AS 项目数据导出为以下格式：

通信通道	导出的数据	导出文件的格式
SIMATIC S7-1200, S7-1500 Channel	AS 符号和 AS 结构	二进制数据: *.bin 结构化导出: *.sdz
SIMATIC S7 Protocol Suite	AS 符号和 AS 结构	结构化导出: *.sdz

要求

- 已在“SIMATIC S7-1200, S7-1500 Channel”或“SIMATIC S7 Protocol Suite”中创建了连接。
- 已加载 AS 项目数据并在 WinCC 中进行了组态。

步骤：导出二进制数据

1. 在变量管理器中选择连接。
2. 从快捷菜单中选择“AS 符号 > 保存到文件”(AS Symbols > Save to file) 条目。
将打开“导出”(Export) 对话框。
3. 选择存储路径并输入文件名。
单击“导出”(Export) 按钮关闭对话框。
组态数据以二进制数据集的形式导出到 .bin 文件中。

步骤：导出结构化数据

1. 在变量管理器中选择“符号”(Symbols) 视图。
2. 选择“编辑 > 导出”(Edit > Export) 菜单命令。
3. 选择存储路径并输入文件名。
单击“导出”(Export) 按钮关闭对话框。
组态数据导出到 *.sdz 文件中。
结构化导出还包含导航区中的结构信息。

参见

如何组态具有优先块访问的变量 (页 479)

如何离线下载 AS 符号 (页 482)

如何组态 AS 结构 (页 487)

6.12.3.9 冗余系统 S7-1500R/H

CPU 冗余

CPU 冗余表示 CPU 可使用两次。SIMATIC 存储卡必须插入每一个 CPU 中。

冗余操作过程中，两个 CPU 会并行执行用户程序。

如果一个 CPU 发生故障，另一个 CPU 会对过程进行控制。

主 CPU	执行过程的 CPU 如果 R/H 系统处于运行-冗余系统状态，主 CPU 会执行过程。
备用 CPU	以下 CPU 备用 CPU 会同步执行用户程序，并会在主 CPU 发生故障的情况下继续执行过程控制。


更多相关信息，请参见 S7-1500R/H 手册的产品支持：

- S7-1500 > 冗余 CPU (<https://support.industry.siemens.com/cs/products?dtp=Manual&mt=2001&mfn=ps&pnid=25152&lc=cn-WW>)
- “冗余系统 S7-1500R/H”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>) (10/2018)

操作目的

在实践中，冗余自动化系统用于实现更高的可用性或故障安全性：

- 容错系统：
通过两个系统并行操作的方式降低生产停机的概率。
- 故障安全系统：
通过安全停机至安全空闲位置保护生命、环境和财产。

 警告
容错和故障安全系统之间的区别 S7-1500R/H 属于容错型自动化系统，但不是故障安全系统。 不得使用 S7-1500R/H 系统控制安全相关过程。

用途

在冗余操作系统中，单个自动化组件失效或故障不得影响系统操作。

举例来说，冗余系统 S7-1500R/H 的应用领域包括：

- 隧道
- 机场，例如行李处理系统
- 地铁
- 造船业
- 污水处理厂
- 高层货架仓库

S7-1500 R/H CPU 的功能

S7-1500R/H 冗余系统基于 PROFINET 环网中的介质冗余 (MRP)。

CPU 的显示画面会在不同菜单中显示控制和状态信息。快速访问诊断消息可最大限度地缩短检修过程中的系统停机时间。

CPU 支持跟踪所有 CPU 变量，可有效调试和优化驱动器和控制回路。

介质冗余 (MRP)

“介质冗余协议”(MRP) 这一功能用于确保网络和系统可用性。

冗余系统中的两个 CPU 必须位于使用 MRP 介质冗余协议的 PROFINET 环网中。

- S7-1500R 使用 PROFINET 环网同步两个 CPU。
- S7-1500H 使用通过光缆建立的冗余连接同步两个 CPU。
通过 H-CPU 的 PROFINET X1 接口的 PROFINET 环网也是 S7-1500H 必备的。

关于介质冗余和系统冗余 S2 的附加信息：

- “冗余系统 S7-1500R/H”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>): “系统概述 > S7-1500 R/H-CPU”部分
- “使用 STEP 7 V15 的 PROFINET”功能手册 (<https://support.industry.siemens.com/cs/cn/zh/view/49948856>)

参见

组态冗余控制系统 (页 493)

S7-1500R/H 的软件冗余 (页 496)

查找: S7-1500 > 冗余 CPU (<https://support.industry.siemens.com/cs/products?dtp=Manual&mt=2001&mf=ps&pnid=25152&lc=cn-WW>)

“冗余系统 S7-1500R/H”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>)

“使用 STEP 7 V15 的 PROFINET”功能手册 (<https://support.industry.siemens.com/cs/cn/zh/view/49948856>)

6.12.3.10 组态冗余控制系统

在 STEP 7 和 TIA Portal 中完全组态 CPU 冗余。

在 WinCC 中, 仅创建与 S7-1500 R/H CPU 的 IP 地址的相应连接, 并创建软件冗余脚本。

IP 地址

设备 IP 地址

为了使 CPU 和 I/O 设备的接口可被访问, 网络中的接口需要具有唯一的 IP 地址, 即设备 IP 地址。

为“软件冗余”功能使用设备 IP 地址。

MAC 地址

CPU 的每个接口及其端口都有唯一的 MAC 地址。

LLDP 协议需要使用 PROFINET 端口的 MAC 地址实现“邻域检测”等功能。

MAC 地址的编号范围是连续的。第一个和最后一个 MAC 地址印于 CPU 右侧的铭牌上。

系统 IP 地址

冗余系统 S7 1500R/H 额外支持系统 IP 地址:

系统 IP 地址 X1	两个 CPU 的 PROFINET X1 接口的系统 IP 地址
系统 IP 地址 X2	两个 CPU 的 PROFINET X2 接口的系统 IP 地址

使用系统 IP 地址与其它设备 (如 HMI 设备、CPU、PG/PC) 进行通信。

设备始终通过系统 IP 地址与冗余系统的主 CPU 进行通信。这样可确保通信伙伴可在主 CPU 发生故障后与处于 RUN-Solo 系统状态的新的主 CPU (之前的备用 CPU) 进行通信。

系统 IP 地址已在 STEP 7 中激活。

虚拟 MAC 地址

每个系统 IP 地址都包含虚拟 MAC 地址。

两个 PROFINET 接口的虚拟 MAC 地址必须彼此不同，且不得与 CPU 的 MAC 地址相同。

有关组态系统 IP 地址和虚拟 MAC 地址的信息：

- “冗余系统 S7-1500R/H”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>)：“组态 > 组态步骤”部分

系统 IP 地址/设备 IP 地址的优点

与设备 IP 地址相比，系统 IP 地址具有下列优点：

- 通信伙伴专用于与主 CPU 通信。
- 即使主 CPU 发生故障，通过系统 IP 地址进行的冗余系统 S7-1500R/H 的通信也会继续进行。

与设备 IP 地址相比，软件冗余具有下列优点：

- 冗余物理电缆（网络电缆）

有关 IP 地址的详细信息：

- “S7-1500R/H 冗余系统”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>)：“系统概述 > 系统和设备 IP 地址”部分
- “使用 STEP 7 V15 的 PROFINET”功能手册 (<https://support.industry.siemens.com/cs/cn/zh/view/49948856>)

HMI 设备

HMI 设备用于过程可视化和机器级控制。

对于冗余系统 S7-1500R/H，使用的 HMI 设备与为标准 S7-1500 系统使用的设备相同。

一个或多个 HMI 设备通过 HMI 通信与 CPU 交换数据，例如 HMI 精简面板、精智面板和移动面板。

HMI 设备的组态

HMI 设备与冗余系统的连接取决于各自的应用。

可在 STEP 7 中组态以下通信：

- HMI 设备通过系统 IP 地址与冗余系统进行通信。
- HMI 设备通过设备 IP 地址与各自的 R/H CPU 进行通信，以实现诊断等目的。

关于使用 HMI 设备的其它信息：

- “冗余系统 S7-1500R/H”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>)：“使用 HMI 设备”部分
- “SIMATIC S7-1500、ET 200MP、ET 200SP、ET 200A、ET 200pro 通信”功能手册 (<https://support.industry.siemens.com/cs/cn/zh/view/59192925>)

有关所有可用 HMI 设备的概述，请参见工业在线商城。

组态

在 WinCC 变量管理中，在“SIMATIC S7-1200, S7-1500 Channel”下组态与 S7-1500 R/H 控制器的单独连接。

CPU 冗余

系统 IP 地址用于与主 CPU 建立连接。

在 STEP 7 和 TIA Portal 中完全组态 CPU 冗余。

包括以下步骤：

1. 创建项目和 R/H CPU
2. 分配 IP 地址（设备 IP 地址）
会在 STEP 7 项目树中创建和分配 CPU 的冗余 ID。
项目树中的上方 CPU 始终具有冗余 ID 1。
3. 分配系统 IP 地址（可选）
4. 设置扫描周期监视时间或使用默认值
5. 创建 IO 设备
6. 将 IO 设备连接到两个冗余 CPU
7. 承担 S7-1500R/H 冗余系统 CPU 的 MRP 角色
STEP 7 会自动为两个 CPU 的 PROFINET X1 接口分配 MRP 角色“管理者（自动）”。
8. 为 STEP 7 中的其它环网节点定义 MRP 角色
在“设备”(Devices)表中，将 MRP“客户端”角色分配给环网的其它所有节点。
9. 为 STEP 7 项目以外的节点分配参数
对于不在 STEP 7 中的环网节点，应设置 MRP“客户端”角色。

关于组态和调试的附加信息：

- “冗余系统 S7-1500R/H”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>)

“软件冗余”功能

如果要通过设备 IP 地址使用软件冗余，还必须在 WinCC 中为冗余控制器组态脚本。

更多相关信息，请参见“S7-1500R/H 的软件冗余(页 496)”

参见

冗余系统 S7-1500R/H (页 491)

S7-1500R/H 的软件冗余 (页 496)

“冗余系统 S7-1500R/H”手册 (<https://support.industry.siemens.com/cs/cn/zh/view/109754833>)

“使用 STEP 7 V15 的 PROFINET”功能手册 (<https://support.industry.siemens.com/cs/cn/zh/view/49948856>)

“SIMATIC S7-1500、ET 200MP、ET 200SP、ET 200A、ET 200pro 通信” 功能手册 (<https://support.industry.siemens.com/cs/cn/zh/view/59192925>)

6.12.3.11 S7-1500R/H 的软件冗余

使用“软件冗余”功能通过 CPU 的物理连接来操作 CPU 冗余。根据冗余组态，用户需要一个或两个网卡或西门子 CP。

在 STEP 7 和 TIA Portal 中完全组态 CPU 冗余。更多相关信息，请参见“组态冗余控制系统 (页 493)”

软件冗余的原理

寻址

使用两个冗余控制器的设备 IP 地址，而非系统 IP 地址

系统变量

在变量管理中创建所需的系统变量，以实现冗余控制和监视。

对于每个通信连接，用户可以创建包含相应连接名称的系统变量：

- @<连接名称>@<软件冗余的系统变量>

系统变量的地址必须指向 PLC 中的空闲区域。

软件冗余脚本

例如，要监视软件冗余，请使用 ANSI-C 或 VBScript 创建脚本。此脚本循环调用，例如每 10 秒钟一次。

在此脚本中，用户可以定义主 CPU 和备用 CPU 之间的切换。

要在运行系统中将脚本作为循环动作运行，请在计算机的启动列表中启用“全局脚本运行系统”应用程序。

示例场景

如果连接中断，WinCC 将读取或设置使用的系统变量。

初始状态

- WinCC 项目正在运行。
- 与主 CPU 的连接“PLC1”已激活。
- 系统变量“@<PLC1>@AlternativeAddress”包含备用 CPU 的有效地址，例如：
 - AccessPoint=abc;IPAddress=111.111.111.111;

连接错误

1. 通过变量“@<PLC1>@ConnectionState”确定连接丢失。
2. 连接已取消激活：
 - @<PLC1>@ForceConnectionState = 0
3. 连接参数已更改：
 - @<PLC1>@UseAlternativeAddress = 1
4. 连接已重新建立：
 - @<PLC1>@ForceConnectionState = 1
5. WinCC 建立与备用 CPU 的备用连接。

诊断

使用以下系统变量评估两个控制器的操作模式：

- @<PLC1>@OpStateConfiguredAddress
- @<PLC1>@OpStateAlternativeAddress

软件冗余的系统变量

在 WinCC 变量管理中创建以下系统变量以连接到 S7-1500R/H 控制器：

变量	用途	值	说明
@<...>@ForceConnectionState	建立/终止冗余连接	1/0	激活运行系统时的行为： <ul style="list-style-type: none"> • 起始值 = 1：连接已建立。 • 起始值 = 0：连接已取消激活。 数据类型：无符号 32 位数值 访问权限：读取/写入
@<...>@ConnectionState	冗余连接状态	1/0	查询连接状态： <ul style="list-style-type: none"> • 1：连接已准备好运行。 • 0：连接中断或终止。 连接变量未归档。 数据类型：无符号 32 位数值 访问权限：读取
@<...>@ConfiguredAddress	与主 CPU 的连接	字符串	在变量管理中组态的连接的属性 数据类型：文本变量 8 位字符集；长度 = 255 访问权限：读取
@<...>@AlternativeAddress	与备用 CPU 的连接	字符串	与备用 CPU 的备用连接的属性 变量必须具有起始值，例如： <ul style="list-style-type: none"> • AccessPoint=abc;IPAddress=111.111.111.111; 随后可以更改该值。 数据类型：文本变量 8 位字符集；长度 = 255 访问权限：读取/写入
@<...>@CurrentAddress	当前地址	字符串	当前使用连接的属性 数据类型：文本变量 8 位字符集；长度 = 255 访问权限：读取
@<...>@UseAlternativeAddress	使用备用连接	1/0	确定当前使用的连接： <ul style="list-style-type: none"> • 1：备用 CPU 的备用连接 • 0：与主 CPU 的连接 数据类型：无符号 32 位数值 访问权限：读取/写入

变量	用途	值	说明
@<...>@RedundantCPUs	冗余激活	1/0	指示连接的控制器是否冗余： <ul style="list-style-type: none"> • 1: 是 • 0: 否, CPU 未冗余 变量必须具有起始值。 随后可以更改该值。 数据类型: 无符号 32 位数值 访问权限: 读取/写入
@<...>@OpStateConfiguredAddress	主 CPU 的操作模式	0/4/6/8/22	控制器的操作模式： <ul style="list-style-type: none"> • 0: not connected • 4: STOP • 6: STARTUP • 8: RUN • 22: SYNCUP 数据类型: 无符号 32 位数值 访问权限: 读取
@<...>@OpStateAlternativeAddress	备用 CPU 的操作模式		

示例脚本

```
'VBS381
Option Explicit
Function action

    Dim C1ConnectionState
    Set C1ConnectionState = HMIRuntime.Tags("@Connection1@ConnectionState")
    C1ConnectionState.Read

    Dim C1OpStateConfiguredAddress
    Set C1OpStateConfiguredAddress =
HMIRuntime.Tags("@Connection1@OpStateConfiguredAddress")
    C1OpStateConfiguredAddress.Read

    Dim C1OpStateAlternativeAddress
    Set C1OpStateAlternativeAddress =
HMIRuntime.Tags("@Connection1@OpStateAlternativeAddress")
    C1OpStateAlternativeAddress.Read

    Dim C1UseAlternativeAddress
    Set C1UseAlternativeAddress =
HMIRuntime.Tags("@Connection1@UseAlternativeAddress")
    C1UseAlternativeAddress.Read

    Dim C1ForceConnectionState
    Set C1ForceConnectionState =
HMIRuntime.Tags("@Connection1@ForceConnectionState")
    C1ForceConnectionState.Read

    'check if connection got disconnected
    If C1ConnectionState.Value = 0 Then
    'set force connection state = 0
        C1ForceConnectionState.Value = 0
        C1ForceConnectionState.Write

    'switch between plc1 and plc2
        Select Case C1UseAlternativeAddress.Value

            Case 0
                C1UseAlternativeAddress.Value = 1
                C1UseAlternativeAddress.Write

            Case 1
                C1UseAlternativeAddress.Value = 0
                C1UseAlternativeAddress.Write

        End Select

    'reconnect
```

```
C1ForceConnectionState.Value = 1
C1ForceConnectionState.Write

'example for handling operating states:stop mode
'check if plc is in stop mode; stop mode does not deliver connectionstate
= 0
  Elseif (C1OpStateConfiguredAddress.Value = 4 And
C1UseAlternativeAddress.Value = 0) Or (C1OpStateAlternativeAddress.Value =
4 And C1UseAlternativeAddress.Value = 1) Then

'set force connection state = 0
  C1ForceConnectionState.Value = 0
  C1ForceConnectionState.Write

'switch between plc1 and plc2
  Select Case C1UseAlternativeAddress.Value

    Case 0
      C1UseAlternativeAddress.Value = 1
      C1UseAlternativeAddress.Write

    Case 1
      C1UseAlternativeAddress.Value = 0
      C1UseAlternativeAddress.Write

  End Select

'reconnect
  C1ForceConnectionState.Value = 1
  C1ForceConnectionState.Write

Else

End If

Set C1ConnectionState = Nothing
Set C1OpStateConfiguredAddress = Nothing
Set C1OpStateAlternativeAddress = Nothing
Set C1UseAlternativeAddress = Nothing
Set C1ForceConnectionState = Nothing

End Function
```

参见

冗余系统 S7-1500R/H (页 491)

组态冗余控制系统 (页 493)

6.13 SIMATIC TI Ethernet Layer 4

6.13.1 WinCC 通道“SIMATIC TI Ethernet Layer 4”

简介

通讯驱动程序“SIMATIC TI Ethernet Layer 4”处理 WinCC 站与 SIMATIC TI505 自动化系统之间通过工业以太网进行的连接。其通信使用 ISO 传输协议处理。

本部分将说明如何：

- 使用“SIMATIC TI Ethernet Layer 4”通道组态数据传送。
- 组态连接和变量。

通道单元

IT 具有两个通道单元，最多可运行两个 CP 1613 A2。通道单元的功能完全相同。两个 CP 1613 A2 之间唯一的不同之处就是其逻辑设备名称不同。

逻辑设备名称可通过通道单元的系统参数进行更改。还可以在此设置 ISO 传输协议的参数。

存在下列应用能力：

- 通道单元 505 Ethernet (CP 1413-1)，适用于 SIMATIC 工业以太网的通讯模块（例如，CP 1613 A2）。
- 通道单元 505 Ethernet (CP 1413-2)，适用于 SIMATIC 工业以太网的通讯模块（例如，CP 1613 A2）。

6.13.2 变量的数据类型

简介

为逻辑连接定义所需变量。从 WinCC 的角度而言，可以访问以下数据类型：

- 二进制变量
- 无符号 8 位数（仅由 VMS 寻址支持）
- 有符号 8 位数（仅由 VMS 寻址支持）
- 无符号 16 位数
- 有符号 16 位数
- 无符号 32 位数
- 有符号 32 位数
- 浮点数 32 位 IEEE 754
- 原始数据类型

6.13.3 组态通道

6.13.3.1 组态通道“SIMATIC TI Ethernet Layer 4”

简介

组态通道“SIMATIC TI Ethernet Layer 4”需要以下步骤。

6.13.3.2 如何组态连接

简介

使用 SIMATIC TI505 自动化系统可实现通过工业以太网进行过程连接。

自动化系统中使用通信模块 CP 1434 TF。其通信使用 ISO 传输协议处理。

WinCC 系统中使用通信模块 CP 1613 A2。

因为通信是通过 ISO 传输协议进行的，因此无需在本地数据库中组态逻辑连接。

6.13 SIMATIC TI Ethernet Layer 4

对于逻辑连接，WinCC 在传送层中建立一个用于读取 (READ) 的连接和一个用于写入 (WRITE) 的连接。仅当这两个连接都已建立时，才表示逻辑连接已建立。

读功能的参数

组态连接时，在 WinCC 中为读功能定义参数。这些参数独立于在 SIMATIC TI 中使用的请求。

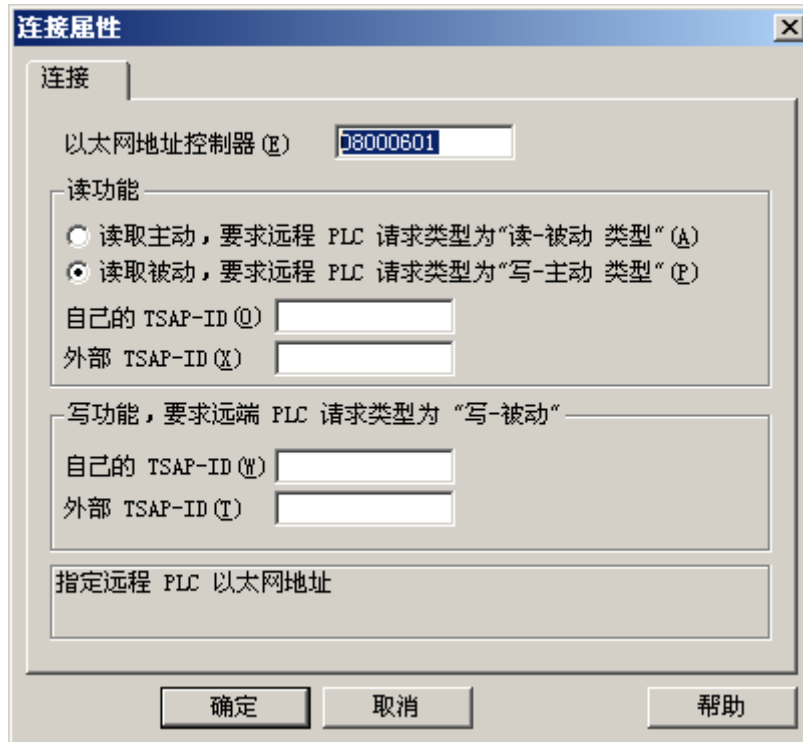
WinCC 端	SIMATIC TI 侧
FETCH-Active (请求“READ-Active”)	READ-Passive (请求“READ-Passive”)
FETCH-Passive (请求“WRITE-Passive”)	WRITE-Active (请求“WRITE-Active”)

写功能的“自身 TSAP-ID”和“外部 TSAP-ID”参数

WinCC 端	SIMATIC TI 侧
请求“WRITE Active”	请求“WRITE Passive”

步骤

1. 选择所需连接，并从快捷菜单中选择“连接参数”(Connection parameters)。将打开“连接属性”(Connection properties)对话框。



2. 在“以太网地址”域中输入工业以太网总线上 SIMATIC TI 的站地址。
3. 接下来定义 WinCC 系统中读功能的参数。这些参数独立于在 SIMATIC TI 中使用的请求。
4. 然后，在“自身 TSAP-ID”分配域中输入组态 CP 1434 TF 时在“远程参数”区中组态为“TSAP”的值。
5. 现在，在“外部 TSAP-ID”分配域中输入组态 CP 1434 TF 时在“本地参数”区中组态为“TSAP”的值。
6. 相应地定义写功能的“自身 TSAP-ID”和“外部 TSAP-ID”参数。

6.13.3.3 组态变量

组态变量

简介


对于 WinCC 与 AS 之间通过“SIMATIC TI Ethernet Layer 4”通道进行的连接，可在 WinCC 中创建不同数据类型的变量。下面将说明如何组态这些数据类型的变量。

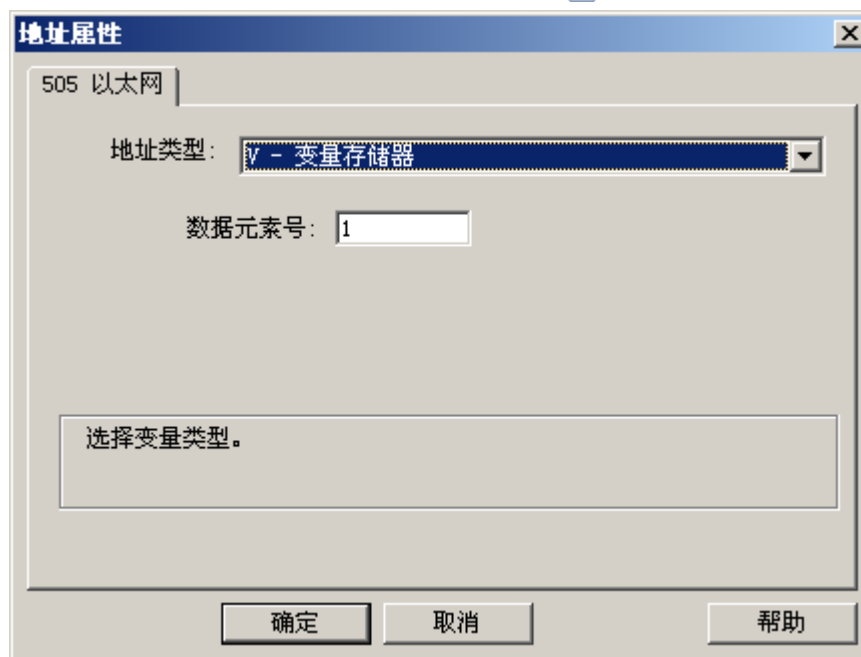
如何组态变量的地址

简介

变量地址根据 SIMATIC TI505 的地址结构输入。

步骤

1. 选择变量
2. 在“数据类型”(Data Type) 字段中选择所需的数据类型。
3. 如果是“二元”或“8 位”变量，则在“属性”(Properties) 区域会提供“位/字节变量”(Bit/Byte tag) 选项。如果要将数据写入 AS 存储器，则选中相应的复选框“访问位/字节”(Access a Bit/Byte)。
4. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



5. 在“地址类型”域中选择变量在自动化系统中的位置。视所选地址类型而定，必须进行更多定义（例如，地址类型为“V - 变量存储器”需定义“数据元素号”域）。
6. 在“只读变量”域中，可指定该变量不能由 WinCC 写入。

说明

不支持结构变量。

关于地址类型的描述，可参见 SIMATIC TI505 技术文档。

使用通道“TI Ethernet Layer 4”时只能按位或按字节执行对 AS 存储器区域的写访问。在使用二进制和“8 位”变量时，除了会打开“地址属性”对话框外，还将打开“位/字节变量”对

对话框，该对话框可用于定义对各个位或字节的写访问。为此，每次写请求时均从 AS 读取已赋址的存储器区域，从而修改相应的位和/或字节。此后，数据写回到 AS 的存储器。

说明

AS 在读取数据区中执行的更改会在写回到该数据区时被覆盖。

视变量类型而定，可以按位或按字节访问 AS 中的存储器。

如何组态按位访问变量


简介

使用通道“TI Ethernet Layer 4”时只能按位或按字节执行对 AS 存储器区域的写访问。在使用二进制和“8 位”变量时，除了会打开“地址属性”对话框外，还将打开“位/字节变量”对话框，该对话框可用于定义对各个位或字节的写访问。为此，每次写请求时均从 AS 读取已赋址的存储器区域，从而修改相应的位和/或字节。此后，数据写回到 AS 的存储器。

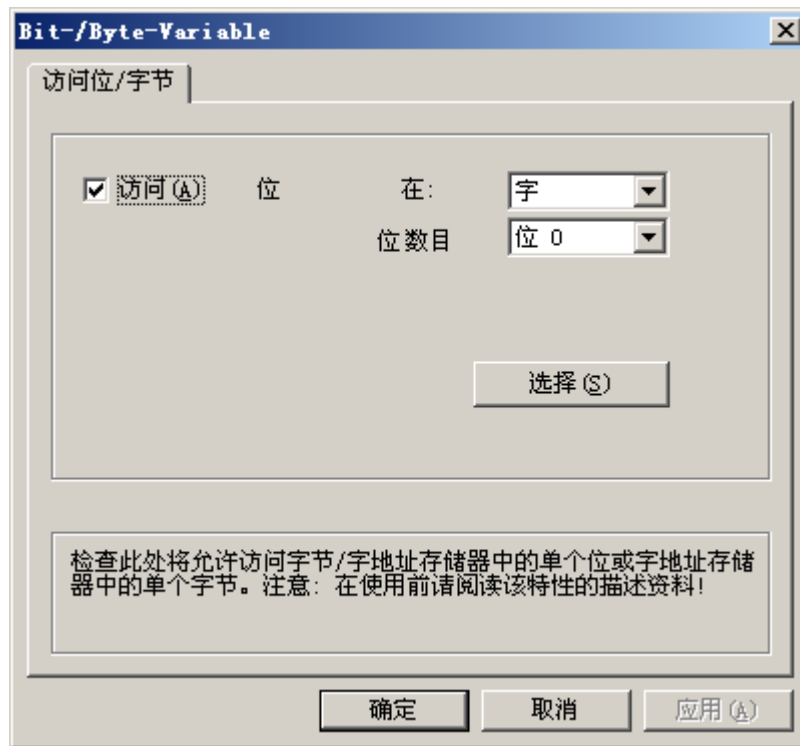
说明

AS 在读取数据区中执行的更改会在写回到该数据区时被覆盖。

步骤

1. 选择变量。
2. 在“数据类型”(Data Type) 字段中，设置“二进制变量”(Binary tag) 数据类型。
3. 打开“位/字节变量”(Bit/Byte tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。

- 4. 选择“访问位”(Access a bit) 复选框，并定义该位的地址。



- 5. 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。
- 6. 在“地址类型”(Address type) 选择域中，选择 PLC 存储器的寻址类型。
- 7. 从下方的表中选择待更改元素的编号。

说明

关于地址类型的描述，可参见 SIMATIC TI505 技术文档。

如何组态按字节访问的变量


简介

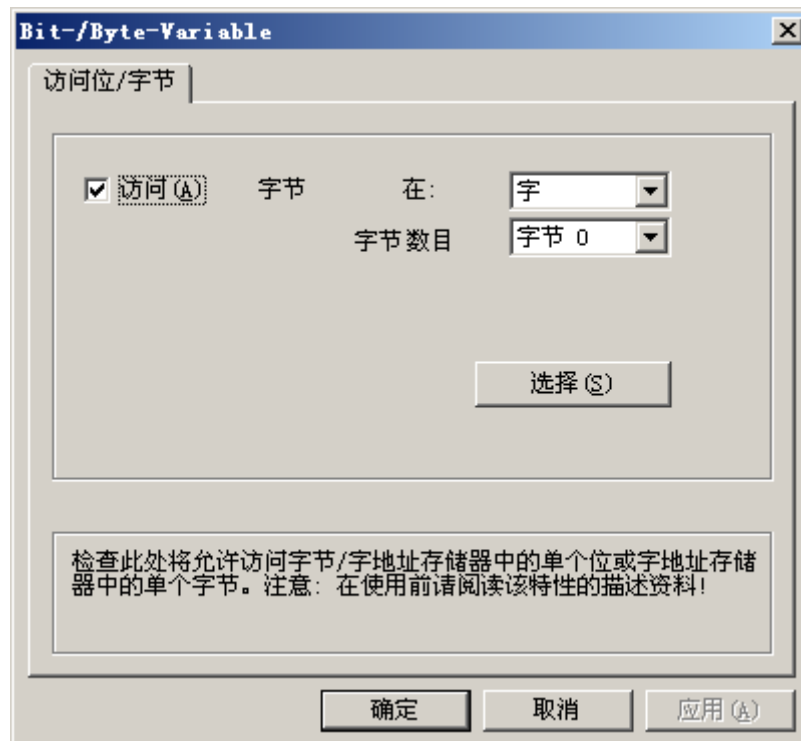
使用通道“TI Ethernet Layer 4”时只能按位或按字节执行对 AS 存储器区域的写访问。在使用二进制和“8 位”变量时，除了会打开“地址属性”对话框外，还将打开“位/字节变量”对话框，该对话框可用于定义对各个位或字节的写访问。为此，每次写请求时均从 AS 读取已赋址的存储器区域，从而修改相应的位和/或字节。此后，数据写回到 AS 的存储器。

说明

AS 在读取数据区中执行的更改会在写回到该数据区时被覆盖。

步骤

1. 选择变量。
2. 在“数据类型”(Data Type)字段中，将数据类型设置为“无符号 8 位数”(Unsigned 8-bit value)或“有符号 8 位数”(Signed 8-bit value)。
3. 打开“位/字节变量”(Bit/Byte tag)对话框。
为此，请单击“地址”(Address)字段，然后单击  按钮。
4. 选择“访问字节”(Access to a byte)复选框，并定义该字节的地址。



5. 单击“选择”(Select)按钮。将打开“地址属性”(Address properties)对话框。

6.13 SIMATIC TI Ethernet Layer 4

6. 在“地址类型”(Address type) 选择域中，选择 PLC 存储器的寻址类型。
7. 从下方的表中选择待更改元素的编号。

说明

关于地址类型的描述，可参见 SIMATIC TI505 技术文档。

6.13.3.4 系统参数

通道单元的系统参数

简介

如果需要与 WinCC 标准设置不一样的组态，则可使用通道单元的“系统参数”对话框进行所需的全部更改。

以下各项均可更改：

- 设备名称
- 传输参数

设备名称

WinCC 和自动化系统之间的通讯通过逻辑设备名称执行。这些名称是在安装通讯模块时分配的，并且是单元特定的条目。设备名称代表逻辑设备名称。默认情况下，逻辑设备名称初始定义为“/ICP_H1_1:/SCP”。

传输参数

可在传输参数中进行通道单元的特定设置，例如 PDU 大小、尝试建立等。

说明

系统参数适用于 AS 中的所有 CP。

如何更改设备名称

简介

使用 SIMATIC TI505 自动化系统可实现通过工业以太网进行过程连接。

自动化系统中使用通信模块 CP 1434 TF。其通信使用 ISO 传输协议处理。

WinCC 系统中使用通信模块 CP 1613 A2。

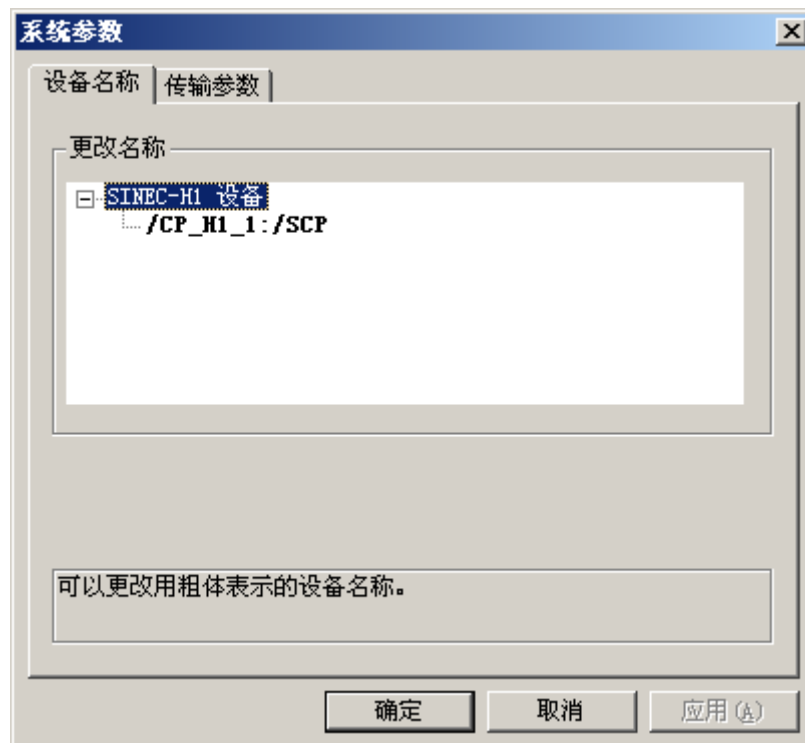
因为通信是通过 ISO 传输协议进行的，因此无需在本地数据库中组态逻辑连接。

要求

- 通道“SIMATIC TI Ethernet Layer 4”必须已集成在项目中。

步骤

1. 选择通道单元，然后通过上下文菜单打开“系统参数”(System parameters) 对话框。
2. 选择“设备名称”(Device Names) 选项卡。



3. 现在，可使用鼠标来选择用“粗体”表示的逻辑设备名称，并通过在名称域中单击鼠标对其进行更改。
默认情况下，逻辑设备名称在安装硬件驱动程序时定义为“/CP_H1_1:/SCP”。
只有在安装硬件驱动程序时定义了其它名称（不推荐），才必须也更改此处的设备名称。

如何更改传输参数

简介

使用 SIMATIC TI505 自动化系统可实现通过工业以太网进行过程连接。

自动化系统中使用通信模块 CP 1434 TF。其通信使用 ISO 传输协议处理。

WinCC 系统中使用通信模块 CP 1613 A2。

因为通信是通过 ISO 传输协议进行的，因此无需在本地数据库中组态逻辑连接。

要求

- 通道“SIMATIC TI Ethernet Layer 4”必须已集成在项目中。

步骤

1. 选择通道单元，然后通过上下文菜单打开“系统参数”(System parameters) 对话框。
2. 选择“传输参数”(Transport Parameters) 选项卡。



3. 将“PDU 大小”(PDU size) 的值设置为在 CP 1434 TF 上所组态的值。
4. 在“建立尝试”(Setup Attempts) 域中，定义尝试建立连接的频率。

5. 在“重复发送持续时间”(Duration of Send Repetitions) 区域中选择“无限”(Infinite)。
6. 在“确认时间”(Acknowledgement time) 域中输入数值“30”，这样，如果通信伙伴在此时间内未作出响应，则会在 30 秒内通知变量状态（例如，AS 处于“停止”状态）。

6.14 SIMATIC TI Serial

6.14.1 WinCC 通道“SIMATIC TI Serial”

简介

通讯驱动程序“SIMATIC TI Serial”用于建立 WinCC 站与 SIMATIC TI505 自动化设备之间的串行连接。

本章描述了

- 如何组态使用“SIMATIC TI Serial”通道的数据传输。
- 如何组态连接和变量。

通道单元

通讯驱动程序具有一个用于控制串行连接的 COM 端口的通道单元。

提供了以下功能：

- 通道单元“505 Serial Unit #1”，用于通过 TBP 协议或 NITP 协议进行串行通讯。

说明

允许通过一个单元通道运行多个逻辑连接（使用不同的 COM 端口）。

6.14.2 变量的数据类型

简介

为逻辑连接定义所需变量。从 WinCC 的角度而言，可以访问以下数据类型：

- 二进制变量
- 无符号 8 位数（仅 VMS 寻址支持）
- 有符号 8 位数（仅 VMS 寻址支持）
- 无符号 16 位数
- 有符号 16 位数
- 无符号 32 位数

- 有符号 32 位数
- 浮点数 32 位 IEEE 754
- 原始数据类型

6.14.3 组态通道

6.14.3.1 组态“SIMATIC TI Serial”通道

简介

组态通道“SIMATIC TI Serial”需要以下步骤。

6.14.3.2 如何组态连接

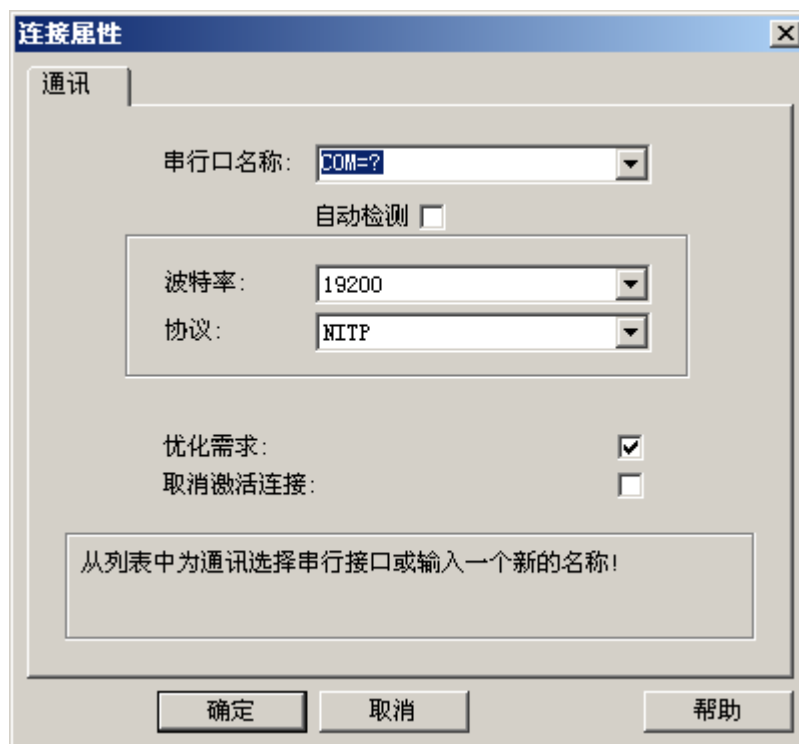
简介

使用 SIMATIC TI505 自动化系统可实现使用串行连接进行过程连接。自动化系统中使用 CPU 模块上的串行接口。

WinCC 中不需要附加通信模块。可通过系统中提供的默认 COM 端口进行通信。

步骤

1. 选择所需连接，并从快捷菜单中选择“连接参数”(Connection parameters)。“连接属性”(Connection properties)对话框随即打开。



2. 在“串行口名称”域中，选择用于串行连接的通信接口（例如，COM1、COM2 或组态的端口）。
3. 如果需要通道单元自动检测 PLC 使用的数据传输速度和协议，请选择“自动检测”域。
4. 在“波特率”(Baud rate) 和“协议”(Protocol) 字段中设置要使用的数据传输速率和协议。
5. 通过选中“优化需求”复选框，可优化数据传送以通过一个请求传输多个变量。
6. 如果选中“禁用连接”域，则会禁用逻辑连接。这通常是在调试期间临时禁用连接的一个好方法。

6.14.3.3 组态变量

组态变量

简介


对于 WinCC 与 PLC 之间通过“SIMATIC TI Serial”通道进行的连接，可在 WinCC 中创建不同类型的数据类型的变量。以下部分将对此进行说明。

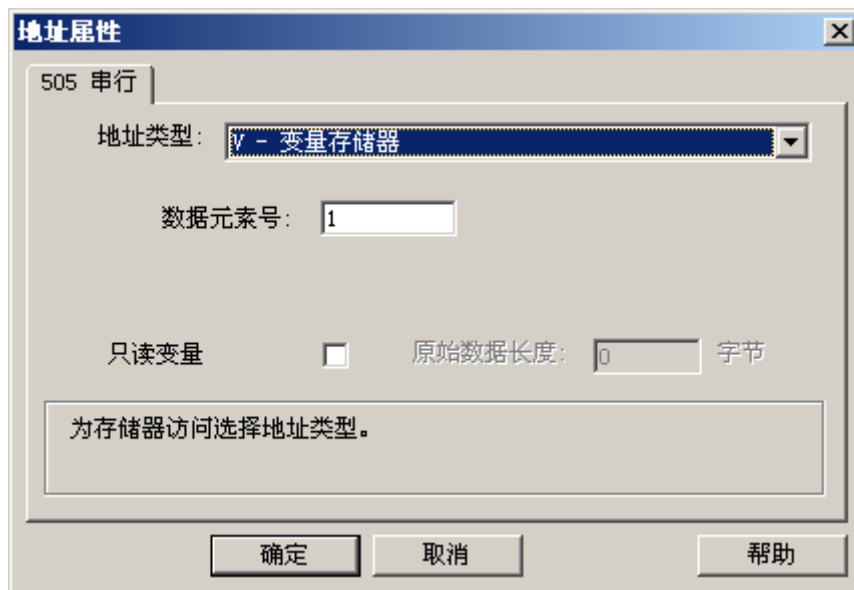
如何组态变量的地址

简介

变量地址根据 SIMATIC TI505 的地址结构输入。

步骤

1. 选择变量。
2. 在“数据类型”(Data type) 字段中设置所需数据类型（如有符号 16 位数）。
3. 如果是“二元”或“8 位”变量，则在“属性”(Properties) 区域会提供“位/字节变量”(Bit/Byte tag) 选项。如果要写入 AS 存储器，则选中相应的复选框“访问位/字节”(Access a Bit/Byte)。
4. 打开“地址属性”(Address properties) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



5. 在“地址类型”域中选择变量在自动化系统中的位置。视所选地址类型而定，必须进行更多定义（例如，地址类型为“V-变量存储器”需定义“数据元素号”域）。
6. 在“只读变量”域中，可指定该变量不能由 WinCC 写入。

说明

地址区域 V、K、X、Y 和 C 中支持结构变量。

关于地址类型的描述，可参见 SIMATIC TI505 技术文档。

6.14 SIMATIC TI Serial

使用通道“TI Serial”时只能按位或按字执行对 AS 存储器区域的写访问。在使用二进制和“8 位”变量时，除了会打开“地址属性”对话框外，还将打开“位/字节变量”对话框，该对话框可用于定义对各个位或字节的写访问。为此，每次写请求时均从 AS 读取已赋址的存储器区域，从而修改相应的位和/或字节。此后，数据写回到 AS 的存储器。

视变量类型而定，可以按位或按字节访问 AS 中的存储器。

如何组态按位访问变量


简介

使用通道“SIMATIC TI Serial”时只能按位或按字节执行对 AS 存储器区域的写访问。在使用二进制和“8 位”变量时，除了会打开“地址属性”对话框外，还将打开“位/字节变量”对话框，该对话框可用于定义对各个位或字节的写访问。为此，每次写请求时均从 AS 读取已赋址的存储器区域，从而修改相应的位和/或字节。此后，数据写回到 AS 的存储器。

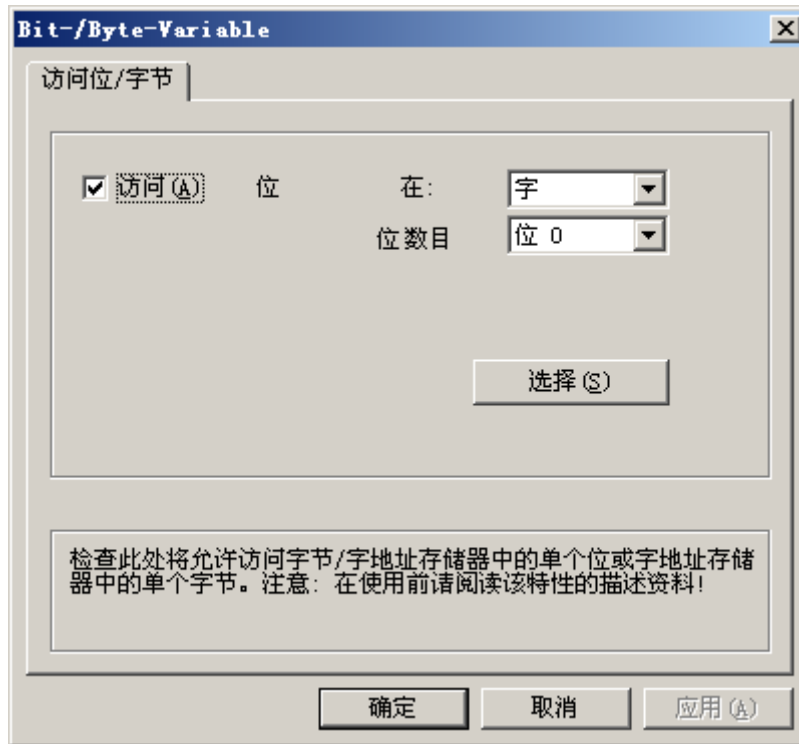
说明

AS 在读取数据区中执行的更改会在写回到该数据区时被覆盖。

步骤

1. 选择变量。
2. 在“数据类型”(Data Type) 字段中，设置“二进制变量”(Binary tag) 数据类型。
3. 打开“位/字节变量”(Bit/Byte tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。

4. 选择“访问位”(Access a bit) 复选框，并定义该位的地址。



5. 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。
6. 在选择域中，选择 PLC 存储器的寻址类型。
7. 在选择域中，选择要更改的位号。

说明

在 S5 中，可以按字节寻址标志位、输入和输出，按字寻址数据块（DB、DX）。

如何组态按字节访问的变量


简介

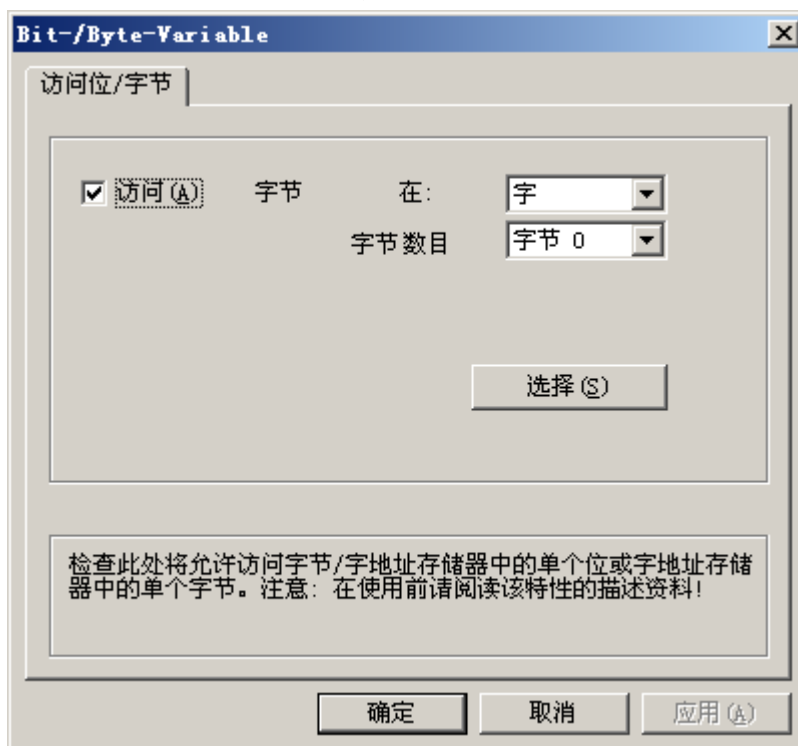
使用通道“SIMATIC TI Serial”时只能按位或按字节执行对 AS 存储器区域的写访问。在使用二进制和“8 位”变量时，除了会打开“地址属性”对话框外，还将打开“位/字节变量”对话框，该对话框可用于定义对各个位或字节的写访问。为此，每次写请求时均从 AS 读取已赋址的存储器区域，从而修改相应的位和/或字节。此后，数据写回到 AS 的存储器。

说明

AS 在读取数据区中执行的更改会在写回到该数据区时被覆盖。

步骤

1. 选择变量。
2. 在“数据类型”(Data Type) 字段中，将数据类型设置为“无符号 8 位数”(Unsigned 8-bit value) 或“有符号 8 位数”(Signed 8-bit value)。
3. 打开“位/字节变量”(Bit/Byte tag) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
4. 选择“访问字节”(Access to a byte) 复选框，并定义该字节的地址。



5. 单击“选择”(Select) 按钮。将打开“地址属性”(Address properties) 对话框。

6. 在选择域中，选择 PLC 存储器的寻址类型。
7. 在选择域中，选择要更改的字节号。

6.15 SIMOTION

6.15.1 WinCC 通道“SIMOTION”

简介

“SIMOTION”通道可连接 WinCC 站与 SIMOTION 自动化系统。使用 TCP/IP 协议通过工业以太网建立连接。

SIMOTION 是着重于运动控制应用及技术任务的自动化和驱动解决方案的系统平台。

SIMOTION 模块化系统包含 SIMOTION SCOUT 工程系统以及针对各种硬件平台的公用运行系统。

从 SIMOTION SCOUT 导出项目以访问 SIMOTION SCOUT 项目的数据。然后，通过 Simotion Mapper 从导出的文件创建 WinCC 项目。

以后可在 WinCC 中对 SIMOTION 自动化系统的组态进行某些更改，例如更改 IP 地址。进一步更改必须在 SIMOTION SCOUT 中组态，再次导出然后通过 Simotion Mapper 传送。

说明

您必须熟悉 SIMOTION SCOUT 工程系统和 WinCC 的组态以在 WinCC 中组态“SIMOTION”通道。

6.15.2 所支持的数据类型概述

简介

在变量的组态中确定数据类型以及针对自动化系统中的数据格式的格式调整。

下表显示了通道支持的数据类型和格式调整应用。

所支持的数据类型

数据类型	类型转换
二进制变量	否
有符号 8 位数	是
无符号 8 位数	是

数据类型	类型转换
有符号 16 位数	是
无符号 16 位数	是
有符号 32 位数	是
无符号 32 位数	是
浮点数 32 位 IEEE 754	是
文本变量, 8 位字符集	否
原始数据类型	否

6.15.3 组态通道

6.15.3.1 “SIMOTION”通道的组态

简介

本章将介绍如何组态“SIMOTION”通道。

说明

您必须熟悉 SIMOTION SCOUT 工程系统和 WinCC 的组态以组态“SIMOTION”通道。

请按以下步骤操作以组态“SIMOTION”通道：

1. 从 SIMOTION SCOUT 导出 SIMOTION SCOUT 项目。
2. 通过 Simotion Mapper 创建 WinCC 项目。
3. 打开 WinCC 项目。
4. 组态系统参数。

有关通道、连接和变量诊断的详细信息，请参见“诊断‘SIMOTION’通道”一章。

6.15.3.2 如何导出 SIMOTION SCOUT 项目

简介

此部分介绍如何从 SIMOTION SCOUT 导出变量和消息定义。

6.15 SIMOTION

要求:

- 您熟悉 SIMOTION SCOUT 工程系统。
- 您对要导出的 SIMOTION SCOUT 项目具有访问权限。

步骤

1. 在 SIMOTION SCOUT 中打开要导出的 SIMOTION SCOUT 项目。
2. 选择“工具”(Tools) 下的“导出 OPC 数据”(Export OPC Data)。
3. 为导出选择版本“SIMATIC NET V6.4”、所需范围以及至少“OPC-报警/事件”(OPC-Alarm/Event) 选项。

说明

Simotion Mapper 不处理“SIMATIC NET V6.4”外的其他导出版本。

4. 选择目标目录。
5. 选择通信接口。
该项目被导出。
6. 如果您正在使用路由，则输入路由信息。

该 SIMOTION SCOUT 项目被导出。“OPC_Data.sti”和“OPC_AE.xml”文件保存在目标目录中。

6.15.3.3 如何通过 Simotion Mapper 创建 WinCC 项目

简介

此部分介绍如何通过 Simotion Mapper 从导出的 SIMOTION SCOUT 项目创建 WinCC 项目。

说明

如果已为旧版本的 SIMOTION SCOUT 项目创建了 WinCC 项目，则在传送中仅更改 SIMOTION 参数。WinCC 项目中的所有其他组态设置（例如归档）保持不变。

要求:

- 您对 SIMOTION SCOUT 项目的导出文件“OPC_Data.sti”和“OPC_AE.xml”具有访问权限。
- 您对 WinCC 安装目录具有访问权限。

说明

在 SIMOTION SCOUT 编程环境中，可使用监视表过滤要导出的变量。使用监视表保持 WinCC 项目中较低的变量数。有关监视表的详细信息，请参见 SIMOTION SCOUT 在线帮助。

步骤

1. 在 WinCC 安装目录中启动“SimotionMapper.exe”程序。
2. 单击“打开”(Open)。导航到包含文件“OPC_Data.sti”和“OPC_AE.xml”的目录。在 Simotion Mapper 中读取并显示数据。
3. 在 Simotion Mapper 项目管理器中，选择 WinCC 项目中所需的组和变量。
4. 选择“创建新 WinCC 项目”(Create new WinCC project)。
5. 如果要更改“WinCC 连接名称”，单击显示的名称，然后输入新名称。
6. 指定技术报警的“第一个 TA 消息编号”。必须以此方法选择值：其不会导致与其他通信通道的消息产生冲突。默认值为 100。
Simotion Mapper 为起始于指定编号的每个 Simotion 连接创建一个消息以及合计六个模板消息。
7. 单击“开始映射”(Start mapping)。选择用于存放 WinCC 项目的目标文件夹。创建 WinCC 项目。进度条指示过程的进度。
8. 关闭 Simotion Mapper。

WinCC 项目已创建，现在可在 WinCC 中打开并编辑该项目。

说明

您可能必须在 WinCC 中设置“SIMOTION”通道的系统参数，以使用创建的 WinCC 项目。

6.15.3.4 如何通过 Simotion Mapper 更改 WinCC 项目

简介

此部分介绍如何通过 Simotion Mapper 将导出的 SIMOTION SCOUT 项目添加到现有的 WinCC 项目。这样，您可在一个 WinCC 项目中多次使用相同的 Simotion 项目。

说明

如果已为旧版本的 SIMOTION SCOUT 项目创建了 WinCC 项目，则在传送中仅更改 SIMOTION 参数。WinCC 项目中的所有其他组态设置（例如归档）保持不变。

要求:

- 您对 SIMOTION SCOUT 项目的导出文件“OPC_Data.sti”和“OPC_AE.xml”具有访问权限。
- 您对 WinCC 安装目录具有访问权限。

说明

在 SIMOTION SCOUT 编程环境中，可使用监视表过滤要导出的变量。使用监视表限制 WinCC 项目中的变量数。有关监视表的详细信息，请参见 SIMOTION SCOUT 在线帮助。

步骤

1. 打开要编辑的 WinCC 项目。
2. 在 WinCC 安装目录中启动“SimotionMapper.exe”程序。
3. 单击“打开”(Open)。导航到包含文件“OPC_Data.sti”和“OPC_AE.xml”的目录。
在 Simotion Mapper 中读取并显示数据。
4. 在 Simotion Mapper 项目管理器中，选择 WinCC 项目中所需的组和变量。
5. 选择“添加到打开的项目”(Add to the open project)。
6. 如果要再次添加已创建的组或变量，必须通过单击显示的名称更改“WinCC 连接名称”。
7. 如果不想为连接传送任何消息、组或变量，则取消选择“WinCC 连接名称”(WinCC connection name)。
8. 指定是否应覆盖变量。

9. 指定技术报警的“第一个 TA 消息编号”。必须以此方法选择值：其不会导致与其他通信通道的消息产生冲突。默认值为 100。
Simotion Mapper 为起始于指定编号的每个 Simotion 连接创建一个消息以及合计六个模板消息。

说明

请不要更改任何已映射的“第一个 TA 消息编号”。否则，可能会出现不可预见的消息行为。

10. 单击“开始映射”(Start mapping)。选择用于存放 WinCC 项目的目标文件夹。
SIMOTION SCOUT 项目添加到打开的 WinCC 项目。进度条指示过程的进度。
11. 关闭 Simotion Mapper。

用 SIMOTION SCOUT 项目扩展了 WinCC 项目，并且该 WinCC 项目随设置一起保存。

6.15.3.5 如何更改连接参数

简介

在本部分中，您将学习如何更改 SIMOTION 网络地址的连接参数。

说明

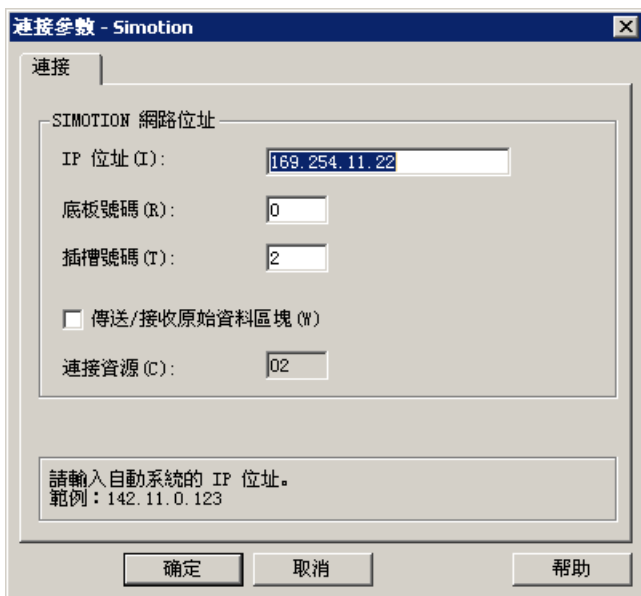
仅更改此处列出的连接参数。不创建“SIMOTION”通道的新连接。错误设置的连接可能导致 PLC 中出现控制错误。根据““SIMOTION”通道的组态 (页 523)”部分的描述组态新的连接。

要求

- SIMOTION 通信驱动程序集成在 WinCC 项目中。
- 必须在“SIMOTION”通道单元中创建了连接。

步骤

1. 在“变量管理”编辑器中打开“SIMOTION”通信驱动程序的目录结构。
2. 从通道单元“Simotion”的一个连接的快捷菜单中，选择条目“连接参数”(Connection parameters)。将打开“连接参数 - SIMOTION”(Connection parameters - SIMOTION) 对话框。



3. 在相应字段中更改 SIMOTION 网络地址的连接参数。
4. 单击“确定”(OK)，关闭每个打开的对话框。

6.15.3.6 如何更改变量地址

简介

本部分介绍如何更改“SIMOTION”通道中的变量地址。


说明

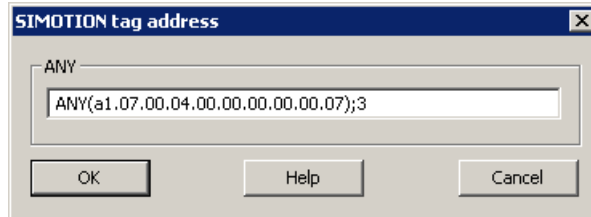
您必须十分了解如何使用 ANY 指针来更改 SIMOTION 通道中的变量地址。如果变量地址输入不正确，则可能无法建立通信连接。

要求

- “SIMOTION”通道集成到 WinCC 项目中。
- 已在“SIMOTION”通道单元中创建与变量的连接。

步骤

1. 打开“SIMOTION 变量地址”(SIMOTION tag address) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。



2. 更改变量地址。

6.15.3.7 系统参数组态

通道单元的系统参数

简介

如果需要不同的组态而不是 WinCC 默认设置，则在“系统参数”(System Parameters) 对话框中进行这些设置。

可以更改以下系统参数：

- 逻辑设备名称
- 通道使用 AS 中的周期性读取服务

逻辑设备名称

在 PLC 中安装通讯处理器后，WinCC 和 PLC 通过已分配的逻辑设备名称进行通信。

通道使用 AS 中的周期性读取服务

PLC 周期性读取服务对即将周期性读取到单个请求中的变量进行编组，并将其传送给 PLC。收到请求后，PLC 首次发送已请求的数据，然后每次经过周期时间后，再次发送。

启用周期性读取服务后，可使用变化驱动的传送功能。如果 PLC 支持变化驱动的传送，则仅当更改值时才传送数据。

如何组态系统参数

简介

此部分介绍如何组态“SIMOTION”通道的系统参数。

“系统参数”对话框由两个标签组成：

- “SIMOTION”选项卡
- “单元”(Unit) 选项卡

说明

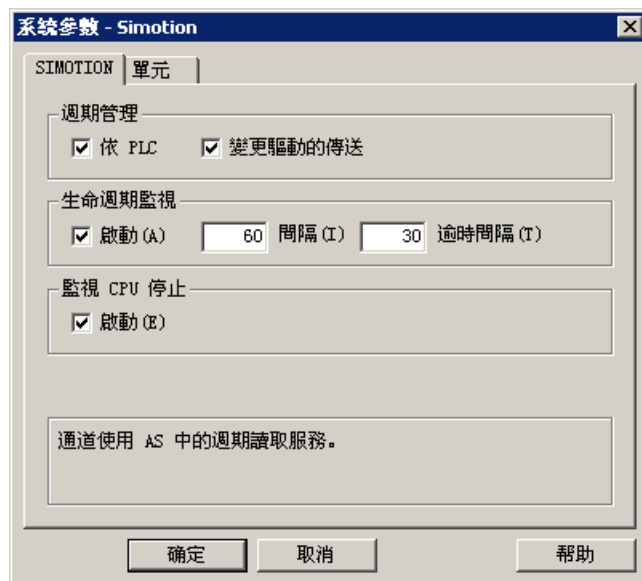
将项目复制到另一台计算机时，“单元”(Unit) 选项卡中的设置会得到保留。相反，将删除“SIMOTION”选项卡上的设置。

要求

- “SIMOTION”通道集成到 WinCC 项目中。

步骤

1. 在变量管理器中选择“SIMOTION”通道。在“Simotion”通道单元的快捷菜单中，打开“系统参数”(System Parameters) 对话框。
2. 选择“SIMOTION”选项卡。



3. 要启用周期性读取变量和变化驱动的传送，则选择“通过 PLC”(by PLC) 和“变化驱动的传送”(Change-driven transfer)。

说明

集成的 SINAMICS 伺服控制不支持“周期控制”、“设备状态监视”和“停止监视”功能。因此，“SIMOTION”通道忽略与 SINAMICS 伺服控制器的连接的相应设置。建立连接时，通道确定是否 AS 支持相应功能。

4. 如果需要，选择“设备状态监视”(Lifebeat monitoring) 功能。
确定发送设备状态监视消息的时间间隔（以秒为单位）。
确定用于监视设备状态监视消息响应的监视时间（以秒为单位）。
5. 当 SIMOTION CPU 处于停止状态时，如果要 WinCC 在通信中发出故障信号，则启用“CPU 停止监视”。
6. 选择“单元”(Unit) 选项卡。
“S7ONLINE”显示为”逻辑设备名称“的默认值。如果在安装使用的通信处理器时选择了不同名称，则必须更改设备名称。



7. 要在运行系统启动时自动设置设备名称，则选择“自动设置”(Set automatically)。
8. 要在处理期间使写作业的优先级高于读作业，则选择“写优先”(Write with priority)。
9. 单击“确定”(OK)，关闭对话框。

说明

设置更改只有在 WinCC 重启后才有效。

6.15 SIMOTION

如何更改逻辑设备名称

简介

WinCC 和 SIMOTION 通过逻辑设备名称进行通信。安装通讯处理器时分配这些逻辑设备名称。

要求

- “SIMOTION”通道集成到项目中。
- 已在“SIMOTION”通道单元中创建连接。

步骤

1. 在变量管理器中选择 SIMOTION 通道。
2. 在快捷菜单中打开“系统参数”(System Parameters) 对话框。
3. 选择“单元”(Unit) 选项卡。
4. 在“逻辑设备名称”(Logical device name) 字段中输入设备名称。可从列表中选择条目或输入新的名称。
设备名称由“设置 PG/PC 接口”工具确定。在系统控制中调用该工具。如果未安装该工具，则仅显示当前设置的设备名称。
如果指定不同的逻辑设备名称，则将显示一条消息。
仅当目标站上使用的通讯处理器未安装在组态系统上时，才输入名称。
5. 单击“确定”(OK)，关闭对话框。

说明

设置更改只有在 WinCC 重启后才有效。

6.15.4 诊断“SIMOTION”通道

6.15.4.1 “SIMOTION”通道的诊断可能性

“SIMOTION”通道及其变量的诊断和错误检测存在以下可能性。

检查通讯处理器组态

检查访问点后，可以使用应用程序“设置 PG/PC 接口”来测试通讯处理器。也可以在 SIMATIC NET 中以相同的方式检查通讯处理器。

检查连接和变量的组态

系统和连接参数的组态可能出现错误。不正确的变量寻址也可能导致变量值错误。

使用“通道诊断”诊断通道

可以通过“通道诊断”查询运行系统中通道和连接的状态。通过“错误代码”显示错误。

诊断通道变量

在运行系统的变量管理器中，可以查询当前值、当前质量代码以及变量的上次变更时间。

6.15.4.2 日志文件条目的描述

简介

通道在日志文件中输入重要的状态变化和错误。这些条目支持通信故障的分析。

文件中的每个条目都包含日期和时间标志，后面带有标记名称和描述。

记录册条目实例：

```
2009-10-28 12:10:11,467 INFO Log starting ...
2009-10-28 12:10:11,483 INFO | LogFileName : D:\SIEMENS\WINCC\Diagnosis
\Simotion_01.LOG
2009-10-28 12:10:11,483 INFO | LogFileCount : 3
2009-10-28 12:10:11,483 INFO | LogFileSize : 1400000
2009-10-28 12:10:11,483 INFO | TraceFlags : fa000000
2009-10-28 12:10:11,498 INFO SIMOTION channel DLL started!
2009-10-28 12:10:11,498 INFO SIMOTION channel with own cycle creation!
2009-10-28 12:10:11,967 INFO Connection "D445": StartRegisterEvVariable for
dwVariableCount = 89
2009-10-28 12:10:11,967 INFO Connection "D445": RegisterEvVariable for Variable
"@D445@Checksum"!
...
2009-10-28 12:10:11,983 INFO Connection "D445": EndRegisterEvVariable
```

6.15 SIMOTION

2009-10-28 12:10:12,436 INFO S7DOS release: @(#)TIS-Block Library DLL Version R8.0.0.0-REL-BASIS

2009-10-28 12:10:12,436 INFO S7DOS version: V8.0 / 0

2009-10-28 12:10:12,436 INFO SIMOTION version: V6.0 / Sep 15 2009 / 08:06:43

2009-10-28 12:10:12,436 INFO SIMOTION channel unit "Simotion" activated!

2009-10-28 12:10:12,451 ERROR Cannot connect to "SINAMICS_Integrated":Errorcode 0xFFDF 42C2!

2009-10-28 12:10:12,451 ERROR Cannot connect to "D445":Errorcode 0xFFDF 42C2!

“INFO”标记中最重要条目的描述

消息文本	含义
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	带路径日志文件的名称
LogFileCount : "n"	通道的日志文件的数目
LogFileSize : "x"	单个日志文件的大小（以字节为单位）
TraceFlags : c4000000	用十六进制数字显示跟踪功能使用的标记
SIMOTION channel DLL started!	起始消息
SIMOTION channel DLL terminated!	结束消息

“ERROR”标记中最重要条目的描述

消息文本	含义
Cannot connect to <connectionname>: Errorcode 0x0000 7<xxx>!	通信错误 激活 WinCC 后，不能立即建立与 SIMOTION 的通信。 <connectionname> = 连接的名称 <xxx> 1...fff 通道已接收所有其他错误代码，作为 S7DOS 函数调用、更低层或来自 AS 的结果。
Connectionerror <nnn> <connectionname>: Errorcode 0x0000 7xxx!	通信错误 激活 WinCC 后，不能建立与 SIMOTION 的通信。连接已中断。 <nnn> = 此连接的连接中止数 <connectionname> = 连接的名称 <xxx> 1...fff 通道已接收所有其他错误代码，作为 S7DOS 函数调用、更低层或来自 AS 的结果。
Channel API error: errorstring	通道 API 出错 通道将错误字符串“errorstring”传到 WinCC 项目管理器。根据错误关联，错误字符串显示在信息框中。有关错误字符串描述，请参见 API 错误文本。
Max. count of API errors reached - API logbook deactivated	通道 API 出错 根据错误和功能，可能在 API 上周期性发生错误。要避免这些错误信息占满记录册文件，一个 API 错误最多可以输出 32 条信息。
Cannot write storage data! Cannot read storage data / use default data Storage data illegal or destroyed / use default data! No storage data / use default data!	常规通道错误消息
单元“单元名”中的设备名从“旧 设备名”改为“新设备名”	初始化消息

说明

“SIMOTION”通道的错误代码对应于“SIMATIC S7 Protocol Suite”通道的错误代码。有关错误代码的描述，请参见“SIMATIC S7 Protocol Suite”通道帮助的“连接故障的错误代码”一章。此外，如果一致性检查失败，SIMOTION 将报告错误代码 0x000 7301。产生错误消息的原因是：从 SIMOTION SCOUT 导出的 AS 输入校验和的数据与已连接设备中的校验和不匹配。

6.16 System Info

6.16.1 “System Info”通道

内容

“System Info”通道用于解释系统信息，如时间、日期、磁盘容量，并提供定时器和计数器等功能。

该章将介绍

- 组态通道、连接和变量
- 在过程画面中显示系统信息
- 使用系统信息来触发和显示消息
- 用图表显示系统信息
- 显示多用户系统中多个服务器的系统信息

6.16.2 WinCC System Info 通道

原理

“System Info”通道用于计算系统信息，如时间、日期、磁盘容量，并提供定时器和计数器等功能。

可能的应用如下：

- 在过程画面中显示时间、日期和星期几
- 通过在脚本中判断系统信息来触发事件
- 在趋势图中显示 CPU 负载
- 显示和监视客户端系统中不同服务器上可用的驱动器空间
- 监视可用磁盘容量并触发消息

该通道无硬件要求，因为通道直接访问安装该通道的计算机的系统信息。为了使通道起作用，必须建立连接。可以进行附加连接，但不需要进行特别的操作。

6.16 System Info

有关通道和变量诊断的更多信息，请参阅“通信诊断”。

说明

授权

“System Info”通道所需的过程变量不需要许可证。因此，这些变量不算在许可证计数中。

用户权限

如果您没有管理员权限，则必须是高级用户及“Performance Monitor 用户”组的成员，才能使用“System Info”通道。

连接状态变量

“系统信息”(System Info) 通道不支持变量 @<connection_name>@ForceConnectionStateEx 和 @<connection_name>@ConnectionStateEx。

通信手册

通信手册包含通道组态的更多信息和扩展实例。本手册可在 Internet 上下载，地址如下：

- <http://support.automation.siemens.com/>

搜索项目号：

- A5E00391327

参见

在多用户和客户系统中使用 (页 556)

如何调用和评价系统信息 (页 545)

如何组态 System Info 通道 (页 544)

与其它软件组件的不同之处 (页 544)

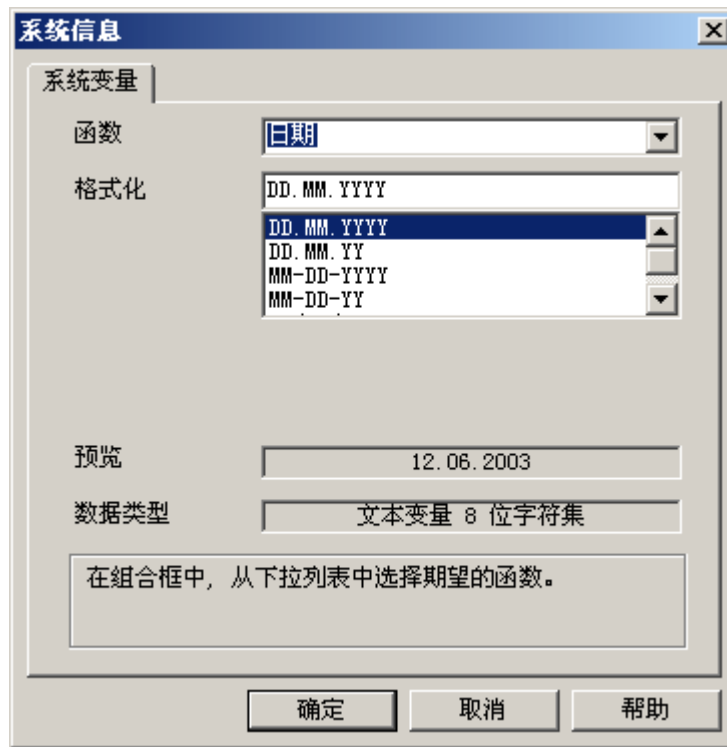
所支持的系统信息概述 (页 539)

通道和变量的诊断 (页 563)

6.16.3 所支持的系统信息概述

引言

在“System Info”对话框的“功能”域中，指定要分配给 WinCC 变量的系统信息。在“格式”域中设置显示格式。



System Info 通道支持的系统信息 - 概述

函数	数据类型	格式	预览
日期	文本变量 8 位字符集	DD.MM.YYYY DD.MM.YY MM-DD-YYYY MM-DD-YY MM/DD/YY	21.10.199 9 21.10.99 10-21-199 9 10-21-99 10/21/99
日	无符号 16 位数	DD	1...31

6.16 System Info

函数	数据类型	格式	预览
月	无符号 16 位数	MM	1...12
年	无符号 16 位数	YYYY	2000
工作日	无符号 16 位数 文本变量 8 位字符集	文本： 1 指 星期一... 7 指星期日 文本： 周一、周二、周三、 周四、周五、周六、周日	1...7 周一 ... 周日
时间	文本变量 8 位字符集； 长度 = 10 字节 长度 = 12 字节	HH:MM:SS HH:MM HH:MM AM,PM HH:MM:SS AM,PM	23:45:37 23:45 23:45 PM 23:45:37 PM
小时	无符号 16 位数	HH	0...23
分	无符号 16 位数	MM	0...59
秒	无符号 16 位数	SS	0...59
毫秒	无符号 16 位数	MSC	0...999
计数器	有符号 32 位数	ZZZZ	0...9999
CPU 负载	浮点数 32 位 IEEE 754	总负载 (%) 空闲负载 (%) 处理负载 (%)	0...100%
定时器	有符号 32 位数	TTTT	0...9999
空闲主内存	浮点数 32 位 IEEE 754	空闲容量 (KB) 空闲 (%) 空闲 (字节)	0...n kB 0...100% 0...n B
空闲 磁盘容量 (本地磁盘)	浮点数 32 位 IEEE 754	空闲容量 (MB) 空闲容量 (%)	0...n MB 0...100%

函数	数据类型	格式	预览
打印机监控	无符号 32 位数	假脱机磁盘 满容量 打印机状态 作业状态 空闲的假脱机磁盘 区域 (KB) 空闲 PRT_OUT 磁盘区域 (KB) 假脱机目录 大小 (KB) PRT_OUT 目录大小 (KB)	0...n % 0...n 0...n 0...n KB 0...n KB 0...n KB 0...n KB
交换文件的 状态	浮点数 32 位 IEEE 754	已使用 (KB) 已使用 (%) 可用 (KB)	0...n kB 0...100% 0...n kB

计数器

该函数可在脚本中用于测试。

定时器

选择该函数后，“限制从”和“到”域将扩展到“System Info”对话框。

每过一秒后，定时器数值增加或减小。要进行改变的方向由域“限制从”和“到”的开始和结束值来确定。如果起始值小于结束值，则定时器数值将增加。如果起始值大于结束值，则定时器数值将减小。

如果在运行系统中，在链接到定时器的 I/O 域中输入一个数值，则定时器的起始值和当前值都将被设置为该值。实例：组态为 0 至 60 的定时器。如果在运行系统写入“0”，定时器将复位。

取消激活后，将再次启用原来的起始值。

CPU 负载

对于与多处理器 PC 有关的格式“总负载 (%)”和“空闲负载 (%)”，输入从“0”开始的 CPU 编号。

对于格式“处理负载 (%)”，如果一个进程有多个事例，则输入事例编号。

空闲磁盘容量

系统仅可确定本地硬盘或软盘的可用空间。

打印机监控

带有“打印机状态”和“作业状态”格式的服务器名称必须输入到“打印机”域中。使用中的打印机必须支持该状态信息以便使用该系统信息。

要分析打印机状态，请遵守以下内容：

- 端口监视器负责将打印机状态发送到假脱机程序。根据所选的打印机端口，安装不同的端口监视器 DLL。在 Windows 提供的端口监视器中，只有“TCPMON.DLL”能够使用 TCP/IP 端口发送打印机状态。使用 LPT 端口的“LOCALMON.DLL”不发送打印机状态。
- 只有在提交打印作业之后，才会评估打印机状态，而不是在轮询端口状态时进行评估。

对于“空闲 PRT_OUT 驱动空间”和“PRT_OUT 目录大小”格式，通道会自动确定“目录”域的路径。

“打印机状态”格式的错误代码

状态	错误代码
PRINTER_STATUS_PAUSED	0x00000001
PRINTER_STATUS_ERROR	0x00000002
PRINTER_STATUS_PENDING_DELETION	0x00000004
PRINTER_STATUS_PAPER_JAM	0x00000008
PRINTER_STATUS_PAPER_OUT	0x00000010
PRINTER_STATUS_MANUAL_FEED	0x00000020
PRINTER_STATUS_PAPER_PROBLEM	0x00000040
PRINTER_STATUS_OFFLINE	0x00000080
PRINTER_STATUS_IO_ACTIVE	0x00000100
PRINTER_STATUS_BUSY	0x00000200
PRINTER_STATUS_PRINTING	0x00000400
PRINTER_STATUS_OUTPUT_BIN_FULL	0x00000800
PRINTER_STATUS_NOT_AVAILABLE	0x00001000
PRINTER_STATUS_WAITING	0x00002000
PRINTER_STATUS_PROCESSING	0x00004000
PRINTER_STATUS_INITIALIZING	0x00008000

状态	错误代码
PRINTER_STATUS_WARMING_UP	0x00010000
PRINTER_STATUS_TONER_LOW	0x00020000
PRINTER_STATUS_NO_TONER	0x00040000
PRINTER_STATUS_PAGE_PUNT	0x00080000
PRINTER_STATUS_USER_INTERVENTION	0x00100000
PRINTER_STATUS_OUT_OF_MEMORY	0x00200000
PRINTER_STATUS_DOOR_OPEN	0x00400000
PRINTER_STATUS_SERVER_UNKNOWN	0x00800000
PRINTER_STATUS_POWER_SAVE	0x01000000

“作业状态” 格式的错误代码

状态	错误代码
JOB_STATUS_PAUSED	0x00000001
JOB_STATUS_ERROR	0x00000002
JOB_STATUS_DELETING	0x00000004
JOB_STATUS_SPOOLING	0x00000008
JOB_STATUS_PRINTING	0x00000010
JOB_STATUS_OFFLINE	0x00000020
JOB_STATUS_PAPEROUT	0x00000040
JOB_STATUS_PRINTED	0x00000080
JOB_STATUS_DELETED	0x00000100
JOB_STATUS_BLOCKED_DEVQ	0x00000200
JOB_STATUS_USER_INTERVENTION	0x00000400
JOB_STATUS_RESTART	0x00000800

说明

“打印机状态” 和 “作业状态” 格式的错误代码对应于 Visual C 参考文件“Winspool.h”中的数值。

6.16.4 与其它软件组件的不同之处

引言

“System Info”通道的一些系统信息也可通过使用 WinCC ActiveX 控件进行解释和显示。

与 ActiveX 控件的根本差异是在“System Info”通道中，系统信息分配给 WinCC 变量。可以重复进行连续解释（例如消息、限制值），然后分别进行组态。ActiveX 控件用于指定的应用程序，也可用于多用户或客户机系统。

对于下列系统信息，ActiveX 控件和通道之间存在差异：

时间

在 WinCC 中使用 ActiveX 控件“WinCC 数字/模拟时钟控件”显示时间。该控件也支持时间的模拟显示。即使该控件在过程控制系统选件中使用，也不需要“System Info”通道。使用控件时，在其过程画面中可显示 WinCC 客户机的时间。对于“System Info”通道这是不可能的，因为它总是显示服务器的系统时间。

空闲磁盘容量

ActiveX 控件“IX Diskspace”用于显示 WinCC 中磁盘可用的空间。该 ActiveX 控件也可显示网络驱动器上可用的空间并支持其它组态选项，例如直接设置控件中的多个限制值。

6.16.5 组态通道

6.16.5.1 如何组态 System Info 通道

简介

本节说明如何组态“System Info”通道。

步骤

1. 在变量管理的导航区域中，从节点“变量管理”(Tag Management)的快捷菜单中选择条目“添加新的驱动程序”(Add new driver)。
2. 选择“系统信息”(System Info)驱动程序。随即将创建该通道，并在变量管理器中显示该通信驱动程序。
3. 选择关联的“System Info”通道单元并调用快捷菜单。在该快捷菜单中，选择“新建连接”。
4. 输入连接的名称。

5. 单击表格区下方的“变量”(Tags) 选项卡。
6. 单击“名称”(Name) 列最上方的空单元格。
输入变量名。
如果要使用该通道的示例，请按照主题“如何组态变量”的说明继续操作。
7. 在“数据类型”(Data Type) 字段中选择所需的数据类型。

6.16.6 评价和显示系统信息的实例

6.16.6.1 如何调用和评价系统信息

本节以实例说明如何以多种方式显示和评价系统信息。

参见

- 如何在状态显示中显示打印机状态 (页 553)
- 如何组态有关空闲磁盘容量的消息 (页 550)
- 如何在趋势窗口中显示 CPU 负载 (页 549)
- 如何用棒图显示空闲磁盘容量 (页 548)
- 如何在 I/O 域中显示时间 (页 547)
- 如何在“System Info”通道中组态变量 (页 545)

6.16.6.2 如何在“System Info”通道中组态变量

简介

本节说明如何在“System Info”通道中组态变量。将在实例中使用这些变量。

要求


安装“SystemInfo.chn”通道。

所使用数据类型的表格

下表显示了“System Info”通道中使用的变量类型和格式。

示例	变量名称	系统信息（功能）	格式	数据类型
I/O 字段	Sysinfo_Time	时间	时:分:秒 (HH:MM:SS)	文本变量 8 位字符集
棒图, 消息	Sysinfo_Drive _C	空闲驱动器容量 (驱动器: C)	数字 0-100% (空闲 %)	浮点数 32 位 IEEE 754
趋势	Sysinfo_CPU	CPU 负载	数字 0-100% (满载 %)	浮点数 32 位 IEEE 754
打印机状态	Sysinfo_Print erstate	打印机监控	数字 0-n (十六 进制) (打印机状态)	无符号 32 位数

步骤

1. 从关联的“System Info”通道单元的快捷菜单中选择条目“新建连接”，并创建名为“Testinfo”的连接。
2. 单击表格区下方的“变量”(Tags) 选项卡。
3. 单击“名称”(Name) 列最上方的空单元格。
输入变量的名称。
4. 打开“System Info”对话框。
为此，单击“地址”(Address) 字段，然后单击  按钮。
5. 从表中应用适用于该示例的功能和显示格式。
“数据类型”(Data type) 字段将自动适应。
6. 关闭对话框。

参见

如何组态有关空闲磁盘容量的消息 (页 550)

如何组态 System Info 通道 (页 544)

如何在状态显示中显示打印机状态 (页 553)

如何在趋势窗口中显示 CPU 负载 (页 549)

如何用棒图显示空闲磁盘容量 (页 548)

如何在 I/O 域中显示时间 (页 547)

6.16.6.3 如何在 I/O 域中显示时间

要求

用数据类型“文本变量 8 位字符集”组态“Sysinfo_Time”变量。该变量必须以“HH:MM:SS”显示格式来表达“时间”系统信息。

步骤

1. 启动图形编辑器并打开画面。
2. 在画面中添加一个 I/O 域。从“智能对象”对象列表中选择“I/O 域”对象。将打开“I/O 域组态”对话框。



3. 在“变量”域中，输入名称“Sysinfo_Time”。
4. 将更新周期设置为“1 秒”。
5. 将域类型设置为“输出”。关闭对话框。
6. 在 I/O 域的快捷菜单中单击“属性”以打开“对象属性”对话框。
7. 在“属性”标签上，选择“输出/输入”。将“数据格式”属性设置为“字符串”。
8. 关闭对话框并保存画面。
9. 单击图形编辑器工具栏中的相应按钮来激活运行系统。

说明

应仔细考虑如何选择更新周期，因为它会影响计算机负载。因此，每 250 ms 更新一次时间显示将有损于系统性能。

参见

如何启动运行系统 (页 556)

如何插入 I/O 域 (页 555)

如何在“System Info”通道中组态变量 (页 545)

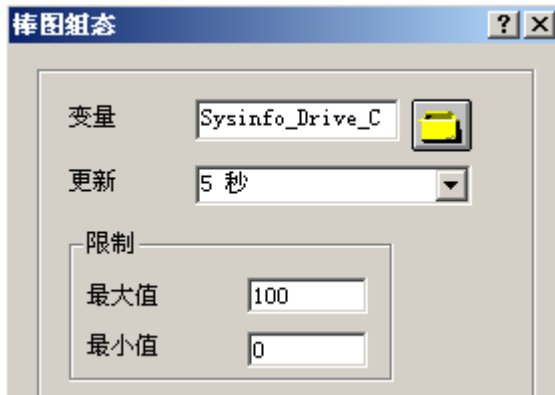
6.16.6.4 如何用棒图显示空闲磁盘容量

要求

组态数据类型为“浮点数 32 位 IEEE754”的变量“Sysinfo_Drive_C”。该变量必须以显示格式“空闲容量 (%)”表达驱动器“C”的“空闲磁盘空间”系统信息。

步骤

1. 启动图形编辑器并打开画面。
2. 在画面中插入一个棒图。为此，从对象选项板的“智能对象”中选择对象“棒图”。将打开“棒图组态”对话框。



3. 在“变量”域中，输入名称“Sysinfo_Drive_C”。
4. 将更新周期设置为“5 秒”。
5. 将最大数值设置为“100”，最小数值设置为“0”。关闭对话框。
6. 在棒图的快捷菜单中单击“属性”以打开“对象属性”对话框。
7. 在“属性”标签上，选择“坐标轴”。将属性“小数位”设置为“0”。
8. 关闭对话框并保存画面。
9. 单击图形编辑器工具栏中的相应按钮来激活运行系统。

说明

应仔细考虑如何选择更新周期，因为它会影响计算机负载。因此，每秒更新一次可用的驱动器空间显示将有损于系统性能。

参见

如何启动运行系统 (页 556)

如何插入棒图 (页 555)

如何在“System Info”通道中组态变量 (页 545)

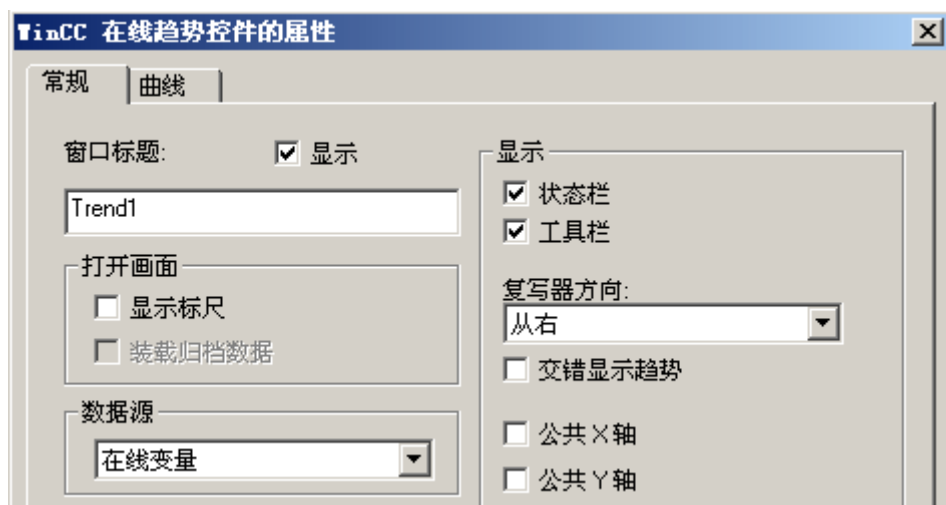
6.16.6.5 如何在趋势窗口中显示 CPU 负载

要求

组态数据类型为“浮点数 32 位 IEEE754”的变量“Sysinfo_CPU”。该变量必须以显示格式“总负载 (%)”表达“CPU 负载”系统信息。

步骤

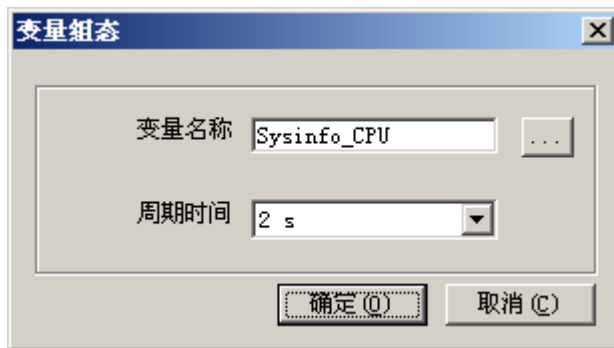
1. 启动图形编辑器并打开画面。
2. 将趋势显示插入画面。为此，可从“控件”对象选项板中选择“WinCC 在线趋势控件”对象。将打开“WinCC 在线趋势控件的属性”对话框。



3. 在“常规”标签的“窗口标题”域中输入名称“Trend1”。
4. 在“数据源”域中，选择“在线变量”。

6.16 System Info

5. 先单击“趋势”标签，然后单击“选择”按钮打开“变量组态”对话框。
6. 输入“Sysinfo_CPU”作为变量名，然后选择周期时间“2 秒”。关闭对话框。



7. 关闭“WinCC 在线趋势控件的属性”对话框并保存画面。
8. 单击图形编辑器工具栏中的相应按钮来激活运行系统。

说明

应仔细考虑如何选择更新周期，因为它会影响计算机负载。因此，每 500 毫秒更新一次 CPU 负载显示将有损于系统性能。

参见

如何启动运行系统 (页 556)

如何在“System Info”通道中组态变量 (页 545)

6.16.6.6 如何组态有关空闲磁盘容量的消息

要求

组态数据类型为“浮点数 32 位 IEEE754”的变量“Sysinfo_Drive_C”。必须以显示格式“空闲存储器 (%)”将该变量分配给驱动器“C”的“可用硬盘空间”系统信息。

步骤

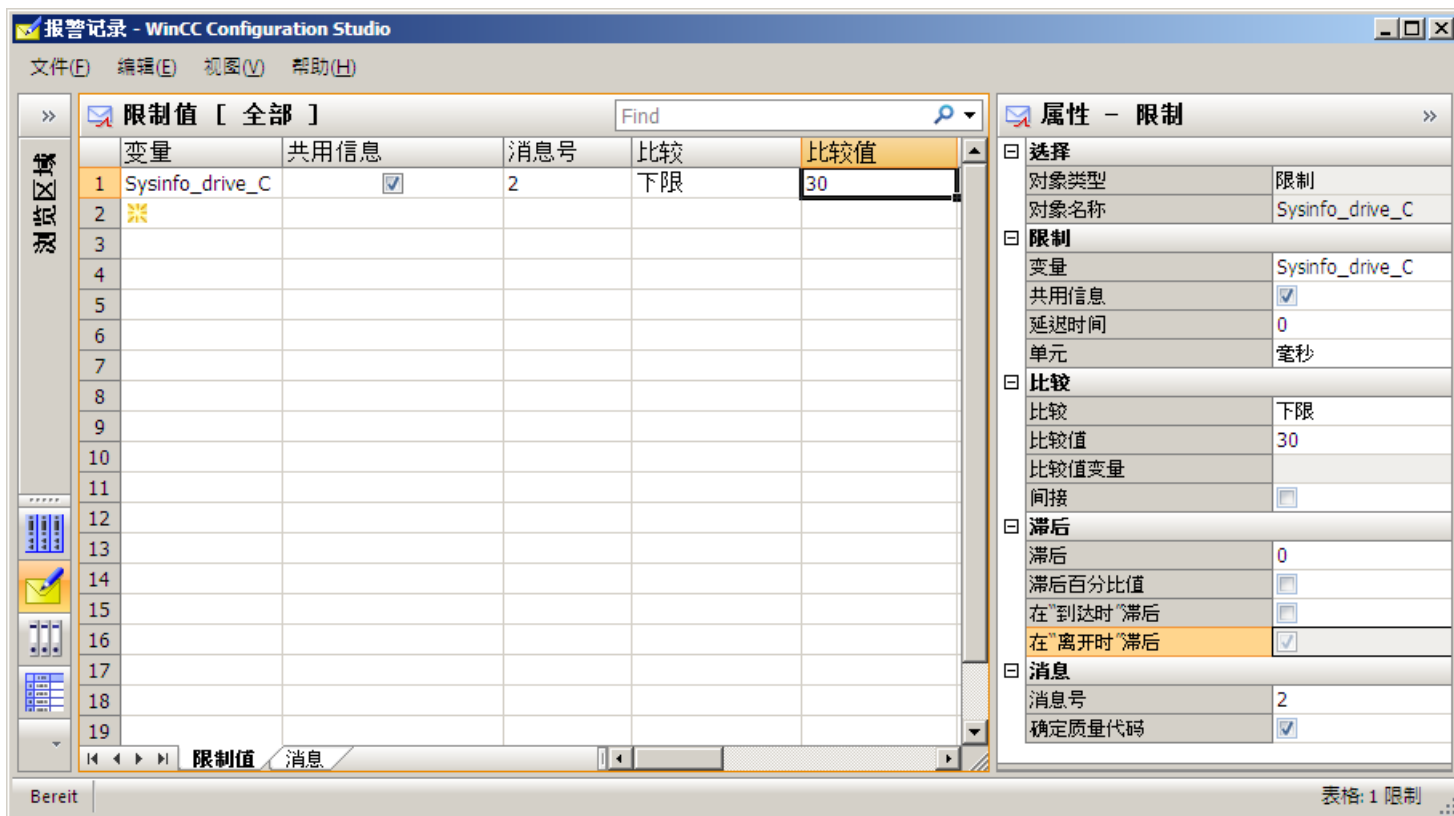
1. 打开“报警记录”编辑器并创建一条新消息。
2. 在“属性”(Properties) 区域中，定义：
 - 文本“存储空间”为“消息文本”
 - 文本“硬盘”为“错误点”



3. 在“报警记录”(Alarm Logging) 导航区域中，选择“限值监视”节点。
4. 在“限值”(Limits) 数据窗口中为变量“Sysinfo_Drive_C”创建一个新的限制值。
5. 激活选项“共享消息”(Shared message)。

6.16 System Info

6. 在“消息号”域中输入新建消息的数目。
7. 在“限值”(Limits) 数据窗口中，通过单击箭头符号来扩展条目“Sysinfo_Drive_C”。
 - 在新插入的行中选择“下限”(Low limit)。
 - 在“比较值”(Comparison value) 框中输入数值“30”。



参见

- 如何在“System Info”通道中组态变量 (页 545)
- 如何显示有关可用磁盘容量的消息 (页 552)

6.16.6.7 如何显示有关可用磁盘容量的消息

要求

- 数据类型为“浮点数 32 位 IEEE754”的变量“Sysinfo_Drive_C”。必须以显示格式“空闲存储器 (%)”将该变量分配给驱动器“C”的“空闲磁盘空间”系统信息。
- 用于监视此变量限制值的消息文本和下限值已组态。
- “报警记录运行系统”必须设置在计算机启动参数中。

步骤

1. 启动图形编辑器并打开画面。
2. 在画面中插入一个消息窗口。从“控件”对象选项板中选择“WinCC 报警控件”对象，并将其放置到画面中。
3. 激活“属性”(Properties)对话框的“消息块”(Message blocks)选项卡上的“已存在的消息块”(Existing message blocks)字段中的复选框“消息文本”和“错误点”。
4. 将已存在的消息块“消息文本”和“错误点”移动到“消息列表”(Message lists)选项卡的“消息行元素”(Elements of the message line)字段中。
5. 关闭对话框并保存画面。
6. 单击图形编辑器工具栏中的相应按钮来激活运行系统。

参见

如何检查 WinCC 启动参数 (页 555)

如何启动运行系统 (页 556)

如何在“System Info”通道中组态变量 (页 545)

6.16.6.8 如何在状态显示中显示打印机状态

简介

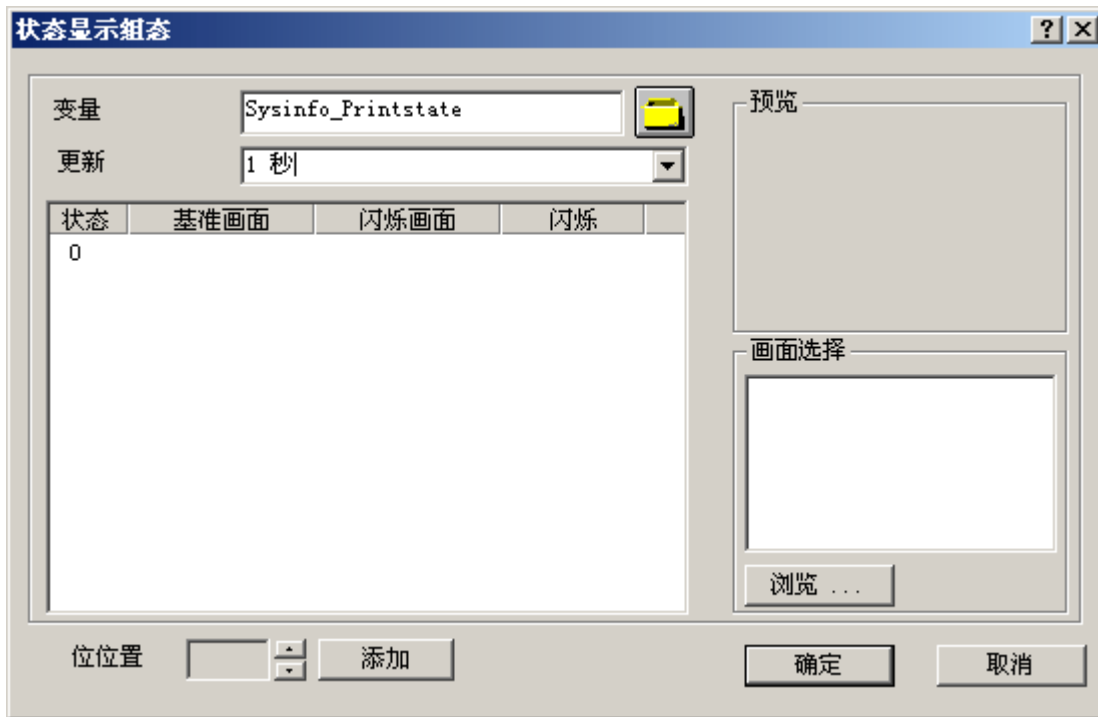
该实例基于“打印机状态”显示格式，显示可能的“打印机监控”系统信息评估。打印机或作业状态也可由组态消息评估，其由打印机或作业状态中的单一一位触发。

要求

- 组态数据类型为“无符号 32 位数”的“Sysinfo_Printerstate”变量。该变量必须以“打印机状态”显示格式分配到“打印机监控”系统信息。
- 使用中的打印机必须支持这些状态显示以能够使用该系统信息。

步骤

1. 启动图形编辑器并打开画面。
2. 将状态显示插入到画面中。从对象选项板的“智能对象”中选择“状态显示”对象。将打开“状态显示组态”对话框。



3. 在“变量”域中，输入名称“Sysinfo_Printerstate”。
4. 将更新周期设置为“1 秒”。
5. 在“状态”列中选择数值“0”。从“画面选择”区域为该状态分配一个图标，如代表打印机的图标。选择期望的图标，用鼠标将其拖到“0”行并放于“基准画面”列。如果在“画面选择”区域中没有显示画面或期望的画面，可以单击“浏览...”按钮打开一个选择对话框。
6. 可根据需要使用“添加”按钮添加附加的位置，并将其它画面分配到这些状态。
7. 关闭对话框并保存画面。
8. 单击图形编辑器工具栏中的相应按钮来激活运行系统。

在运行系统中，准备就绪的打印机显示为在步骤 5 中分配的状态为“0”的图标。如果打印机有错误或您未运行步骤 6，则不显示任何图片。如果在步骤 6 中为其它位置分配了图标，则相应显示这些图标。

说明

应仔细考虑如何选择更新周期，因为它会影响计算机负载。因此，每 250 ms 更新一次时间显示将有损于系统性能。

参见

如何在“System Info”通道中组态变量 (页 545)

如何启动运行系统 (页 556)

6.16.6.9 如何检查 WinCC 启动参数

步骤

1. 在 WinCC 项目管理器中，选择“计算机”。
2. 打开快捷菜单并选择“属性”。
3. 将打开“计算机列表属性”对话框。单击“属性”按钮。
4. 将打开“计算机属性”对话框。
5. 单击“启动”标签并选中条目。必要时，可以激活或取消激活运行模块，或将其它应用程序添加到启动列表中。
6. 关闭打开的对话框。

6.16.6.10 如何插入棒图

步骤

1. 启动图形编辑器并打开画面。
2. 在“智能对象”下的“标准”对象选项板中，选择对象“棒图”。
3. 在画面中插入一个棒图。为此，将鼠标指向工作区中要放置棒图的位置。同时按住鼠标，拖动对象到期望的大小。
4. 释放鼠标按钮后，将打开“棒图组态”对话框。在该对话框中，输入 WinCC 变量的名称并设置更新的规范和限制。此外，还可以使用“棒图方向”来设置显示棒图的方向。
5. 关闭对话框。

6.16.6.11 如何插入 I/O 域

步骤

1. 启动图形编辑器并打开画面。
2. 在“智能对象”下的“标准”对象选项板中，选择对象“I/O 域”。
3. 在画面中插入 I/O 域。为此，将鼠标置于工作区中要放置 I/O 域的位置。同时按住鼠标，拖动对象到期望的大小。

6.16 System Info

4. 释放鼠标按钮后，将打开“I/O 域组态”对话框。在该对话框中，输入 WinCC 变量的名称并设置更新和域类型设置。此外，也可以选择用于显示该值的“字体”。
5. 关闭对话框。

6.16.6.12 如何启动运行系统

要求

激活运行系统前，必须定义启动画面。

步骤

1. 保存并关闭所有可能在编辑器中打开的文件。
2. 选择 WinCC 项目管理器。
3. 通过单击工具栏上的“激活”按钮或通过选择“文件”菜单中的“激活”来激活项目。

6.16.7 特殊功能

6.16.7.1 在多用户和客户系统中使用

在多用户和客户系统中使用

引言

在多用户和客户系统中，System Info 通道可用于在客户系统中处理服务器的系统信息。这样就可以在客户系统中，用单台 WinCC 客户机来监控多台服务器。

参见

在 WinCC 客户机上监控多台服务器的系统信息 (页 557)

6.16.7.2 监控多台服务器的系统信息的实例

在 WinCC 客户机上监控多台服务器的系统信息

引言

在该实例中，单台 WinCC 客户机监控两台服务器。监控的系统信息（例如可用的磁盘空间和 CPU 负载）显示在 WinCC 客户机的过程画面中。

这需要进行下列组态：

组态第一台服务器

组态第二台服务器

在 WinCC 客户机上导入变量

在 WinCC 客户机上组态过程画面

激活项目

要求

服务器和 WinCC 客户机必须通过 Windows 网络连接。

参见

如何激活项目 (页 562)

如何在 WinCC 客户机上组态过程画面 (页 561)

如何将变量导入 WinCC 客户机 (页 560)

如何组态第二台服务器 (页 559)

如何组态第一台服务器 (页 557)

如何组态第一台服务器

简介

本节介绍了第一台服务器的组态，这对该实例来说是必要的。



1. 组态“System Info”通道的变量以显示可用的硬盘空间和 CPU 负载。
2. 生成数据包。

所使用数据类型的表格

在该“System Info”通道实例中使用的变量名和格式如下表所示。

变量	函数	数据类型	格式
Sysvar_1_Drive_C	空闲硬盘容量	浮点数 32 位 IEEE 754	0-100% (空闲 %)
Sysvar_1_CPU	CPU 负载	浮点数 32 位 IEEE 754	0-100% (满载 %)

步骤

1. 在第一台服务器上创建名为“Testinfo_1”的多用户项目。在该项目中安装驱动程序“System Info”。
2. 从关联的“System Info”通道单元的快捷菜单中选择条目“新建连接”，并创建名为“Connection1”的连接。
3. 单击表格区下方的“变量”(Tags) 选项卡。
4. 单击“名称”(Name) 列最上方的空单元格。
为该变量输入名称“Sysvar_1_Drive_C”。
5. 将“数据类型”(Data Type) 设置为“32 位浮点数 IEEE 754”(Floating-point number 32-bit IEEE 754)。
6. 打开“系统信息”(System Info) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
7. 将“功能”(Function) 字段设置为“可用磁盘空间”(Available Drive Space)，“驱动器”(Drive) 设置为“C”，“格式”(Format) 设置为“空闲容量 (%)”(Free capacity in %)。
8. 单击“名称”(Name) 列最上方的空单元格。
为该变量输入名称“Sysvar_1_CPU”。
9. 将“数据类型”(Data Type) 设置为“32 位浮点数 IEEE 754”(Floating-point number 32-bit IEEE 754)。
10. 打开“系统信息”(System Info) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
11. 将“功能”(Function) 字段中的值设置为“CPU 负载”(CPU Load)， “格式”(Format) 中的值设置为“总负载 (%)”(Total load in %)。
12. 创建数据包。通过在浏览窗口中选择“服务器数据”(Server data) 并打开快捷菜单来进行操作。
选择菜单项“创建”(Create)。对数据包已创建的消息进行确认。

参见

如何组态第二台服务器 (页 559)

如何组态第二台服务器

简介

本节介绍了第二台服务器的组态，这对该实例来说是必要的。


1. 组态“System Info”通道的变量以显示可用的硬盘空间和 CPU 负载。
2. 生成数据包。

所使用数据类型的表格


在该“System Info”通道实例中使用的变量名和格式如下表所示。

变量	函数	数据类型	格式
Sysvar_2_Drive_C	空闲硬盘容量	浮点数 32 位 IEEE 754	0-100% (空闲 %)
Sysvar_2_CPU	CPU 负载	浮点数 32 位 IEEE 754	0-100% (满载 %)

步骤

1. 在第二台服务器上创建名为“Testinfo_2”的多用户项目。在该项目中安装驱动程序“System Info”。
2. 从关联的“System Info”通道单元的快捷菜单中选择条目“新建连接”，并创建名为“Connection2”的连接。
3. 单击表格区下方的“变量”(Tags) 选项卡。
4. 单击“名称”(Name) 列最上方的空单元格。
为该变量输入名称“Sysvar_2_Drive_C”。
5. 将“数据类型”(Data Type) 设置为“32 位浮点数 IEEE 754”(Floating-point number 32-bit IEEE 754)。
6. 打开“系统信息”(System Info) 对话框。
为此，请单击“地址”(Address) 字段，然后单击  按钮。
7. 将“功能”(Function) 字段设置为“可用磁盘空间”(Available Drive Space)， “驱动器”(Drive) 设置为“C”， “格式”(Format) 设置为“空闲容量 (%)”(Free capacity in %)。
8. 单击“名称”(Name) 列最上方的空单元格。
输入“Sysvar_2_CPU”作为变量名称。
9. 在连接快捷菜单中，选择“新建变量”。
10. 将“数据类型”(Data Type) 设置为“32 位浮点数 IEEE 754”(Floating-point number 32-bit IEEE 754)。

6.16 System Info

11. 打开“系统信息”(System Info)对话框。
为此,请单击“地址”(Address)字段,然后单击  按钮。
12. 将“功能”(Function)字段中的值设置为“CPU 负载”(CPU Load),“格式”(Format)中的值设置为“总负载 (%)”(Total load in %)。关闭所有打开的对话框。
13. 创建数据包。通过在浏览窗口中选择“服务器数据”(Server data)并打开快捷菜单来进行操作。
选择菜单项“创建”(Create)。对数据包已创建的消息进行确认。

参见

如何将变量导入 WinCC 客户机 (页 560)

如何将变量导入 WinCC 客户机

引言

本节介绍了 WinCC 客户机的组态,这对该实例来说是必要的。

1. 装载第一台服务器项目的数据包。
2. 装载第二台服务器项目的数据包。

要求

该实例需要使用两个服务器项目数据包。

服务器	项目	数据包
1	Testinfo_1	Testinfo_1_<计算机名称>
2	Testinfo_2	Testinfo_2_<计算机名称>

步骤

1. 在 WinCC 客户机上创建名为“mc_info”的客户项目。
2. 在服务器数据的快捷菜单中选择“装载”。将打开“打开”对话框。
3. 选择有第一台服务器项目“Testinfo_1”的计算机。
4. 在“<项目名称>\<计算机名称>\Packages”目录中选择数据包“estinfo_1_<计算机名称>.pck”。
5. 单击“打开”按钮,并在数据包打开后确认消息。
6. 在第二台服务器上装载数据包“Testinfo_2_<计算机名称>.pck”。为此,对“要求”表格中的第二个项目采用合适的设置和名称重复步骤 2 至 5。

参见

如何在 WinCC 客户机上组态过程画面 (页 561)

如何在 WinCC 客户机上组态过程画面

引言

本节阐明了 WinCC 客户机的组态，在该实例中要求在 WinCC 客户机上的过程画面中显示服务器的系统信息。

1. 组态第一台服务器的系统信息显示
2. 组态第二台服务器的系统信息显示

要求

该实例需要在客户机项目中装载服务器项目数据包。

数据包	项目	变量
Testinfo_1_<计算机名称>	Testinfo_1	Sysvar_1_Drive_C
Testinfo_1_<计算机名称>	Testinfo_1	Sysvar_1_CPU
Testinfo_2_<计算机名称>	Testinfo_2	Sysvar_2_Drive_C
Testinfo_2_<计算机名称>	Testinfo_2	Sysvar_2_CPU

步骤

1. 在 WinCC 客户机上，启动图形编辑器并创建名为“p_serverinfo”的画面。
2. 在画面中添加一个 I/O 域。从“智能对象”对象列表中选择“I/O 域”对象。将打开“I/O 域组态”对话框。
3. 单击按钮以进行变量选择。将打开“变量”对话框。
4. 选择第一个服务器项目“Testinfo_1”的变量“Sysvar_1_Drive_C”。为此，在数据包“Testinfo_1_<计算机名称>”下打开目录结构。关闭对话框。
5. 在“I/O 域组态”对话框中，将更新周期设置为“5 秒”。
6. 将域类型设置为“输出”。关闭对话框。
7. 将第二个 I/O 域插入画面中，并组态同一个项目的变量“Sysvar_1_CPU”。为此，对“要求”表格中的相应设置重复步骤 2 至 6。
8. 重复步骤 2 至 7 来组态第二个服务器项目“Testinfo_2”的变量。
9. 关闭对话框并保存画面。

说明

应仔细考虑如何选择更新周期，因为它会影响计算机负载。因此，每秒更新数据显示将有损于系统性能。

参见

如何激活项目 (页 562)

如何激活项目

引言

本节显示如何激活服务器和 WinCC 客户机上的项目。

1. 在服务器项目“Testinfo_1”中创建启动画面并激活项目。
2. 在服务器项目“Testinfo_2”中创建启动画面并激活项目。
3. 在 WinCC 客户机上定义项目“mc_info”的启动画面并激活项目。

步骤

1. 在服务器 1 上的项目“Testinfo_1”的浏览窗口中，选择图形编辑器并使用快捷菜单来创建新画面。
2. 将该画面设置为启动画面。为此，从快捷菜单中选择“设置为启动画面”。
3. 单击工具栏上的“激活”按钮来激活项目。
4. 在服务器 2 上的项目“Testinfo_2”的浏览窗口中，选择图形编辑器并使用快捷菜单来创建新画面。
5. 将该画面设置为启动画面。为此，从快捷菜单中选择“设置为启动画面”。
6. 单击工具栏上的“激活”按钮来激活项目。
7. 在 WinCC 客户机上的项目“mc_info”的浏览窗口中，选择图形编辑器。过程画面“p_serverinfo”将显示在数据窗口中。
8. 将该画面设置为启动画面。为此，从快捷菜单中选择“设置为启动画面”。
9. 单击工具栏上的“激活”按钮来激活项目。

通讯 - 诊断

7.1 通道和变量的诊断

本节说明通道及其变量以及内部变量的诊断。

这些诊断可用于如通信故障或出现意料之外的变量值等情况。

以下文档用于说明：

- 识别通信故障的方法。
- 组态并使用下列诊断工具的方法：
 - “状态 - 逻辑连接”
 - “WinCC 通道诊断控件”
 - “性能”(Performance) 变量组的系统变量
- 诊断通道、连接及其变量的方法。
- 诊断内部变量的方法。
- 检查 WinCC 通信硬件的方法。

参见

“SIMATIC S5 PROFIBUS FDL”通道的诊断选项 (页 600)

变量的质量代码 (页 622)

使用全局动作监视变量状态 (页 631)

使用变量状态监视连接状态 (页 629)

如何检查内部变量 (页 632)

通道诊断 (页 565)

有关错误检测的常规信息 (页 564)

诊断“OPC”通道的可能性 (页 610)

“SIMATIC S7 Protocol Suite”通道 - 诊断选项 (页 588)

“System Info”通道 - 诊断选项 (页 577)

诊断通道“SIMATIC S7-1200/S7-1500” (页 582)

7.2 有关错误检测的常规信息

在运行系统中通常会首先识别出在建立通讯链接时发生的故障或错误。

使用 WinCC 变量的动态对象（不能提供当前过程值给它们）在过程画面内将显示为未激活状态。例如，这些对象可以是 I/O 域、滚动条对象或棒图。

如果故障不影响连接的部分 WinCC 变量，就表示某个 WinCC 变量是故障源。此时，应检查变量的编址或者图形编辑器中变量的拼写是否正确。

如果故障影响连接的所有 WinCC 变量，就表示错误发生在连接本身。

以下章节描述定位错误源可采取的措施及可使用的工具。

7.3 通道诊断

7.3.1 通道诊断

以下功能可用于通道及其连接的诊断：

- “状态 - 逻辑连接”(Status - Logical Connections) 功能
- “性能”(Performance) 变量组的系统变量
- WinCC“通道诊断”

参见

如何使用“状态 - 逻辑连接”功能来检查通道 (页 568)

通道诊断的原理 (页 570)

使用性能变量检查连接 (页 565)

7.3.2 使用性能变量检查连接

WinCC 提供以下系统变量来分析通信通道：

- @PRF_DMRT_CHNCON_... (变量管理)
- @PRF_ALGRT_CHNCON_... (报警记录)

可使用这些性能变量评估连接的时间特性。

使用 @<...>@ConnectionStateEx 系统变量确定连接的状态。更多信息：

- “如何使用“状态 - 逻辑连接”功能来检查通道 (页 568)”
- “在运行系统中为连接状态组态变量 (页 155)”

创建性能变量

在通信驱动程序下创建新连接后，WinCC 变量管理器会立即创建相应的性能变量。

为连接重命名时，也会自动为性能变量重命名。

变量位于“内部变量”(Internal tags) 区域中的“Performance”变量组中。

7.3 通道诊断

有关 WinCC 性能变量的更多信息：

- “使用 WinCC > 使用项目 > 对运行系统进行设置”：
 - “使用性能变量进行系统诊断”
 - “性能变量概述”

性能变量的类型

“性能”(Performance) 变量组包含以下变量类型：

变量	数据类型	访问	说明
相对变量	64 位 IEEE 754 浮点数	读取	相对于读取时间适用的值，例如当前待处理值或每秒的值。 复位变量对这些值无影响。 变量名称结尾为： <ul style="list-style-type: none"> • ..._PENDING • ..._SECOND 更新周期：1 秒
计数器变量	64 位 IEEE 754 浮点数	读取	运行系统激活后的绝对值 可以使用复位变量将值复位为“0”。 变量名称结尾为： <ul style="list-style-type: none"> • ..._AVERAGE • ..._PEAK • ..._TOTAL 更新周期：1 秒
复位变量	无符号 32 位数	读取 写入	可通过脚本设置复位变量的值，例如： <ul style="list-style-type: none"> • 0：禁用 • 1：所有相关计数器变量的值复位为“0”。 复位变量值自身也会复位为“0”。 变量名称结尾为： <ul style="list-style-type: none"> • ...RESET

性能变量概述

系统变量 ¹⁾	说明
报警记录	
@PRF_ALGRT_RESET	复位变量将复位以下性能变量的值： <ul style="list-style-type: none"> • @PRF_ALGRT_CHNCON_..._AVERAGE • @PRF_ALGRT_CHNCON_..._PEAK 还会复位未引用通信驱动程序的 @PRF_ALGRT_... 计数器变量。
@PRF_ALGRT_CHNCON_<...>_ALARM_PER_SECOND ²⁾	每秒通过连接发送的生成消息数
@PRF_ALGRT_CHNCON_<...>_ALARM_PER_SECOND_AVERAGE	每秒平均消息数
@PRF_ALGRT_CHNCON_<...>_ALARM_PER_SECOND_PEAK	每秒最大消息数
变量管理	
@PRF_DMRT_CHNCON_<...>_RESET	复位变量将复位以下性能变量的值： <ul style="list-style-type: none"> • @PRF_DMRT_CHNCON_<...>..._TOTAL • @PRF_DMRT_CHNCON_<...>_RESET 复位适用于为同一连接创建的所有计数器变量。
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_PER_SECOND ²⁾	每秒读取的字节 位舍入为字节。 不包含元数据，例如时间戳或 SetValue 回调数据。
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_TOTAL	运行系统激活后读取的字节 位舍入为字节。 不包含元数据，例如时间戳或 SetValue 回调数据。
@PRF_DMRT_CHNCON_<...>_TAG_READS_PENDING	尚未完成的已启动读取请求 该值持续增加表明系统过载。可能原因： <ul style="list-style-type: none"> • 数据源或连接过载或拥堵，导致对读取请求的处理速度不够快。
@PRF_DMRT_CHNCON_<...>_TAG_READS_PER_SECOND ²⁾	每秒读取的变量
@PRF_DMRT_CHNCON_<...>_TAG_READS_TOTAL	运行系统激活后读取的变量

7.3 通道诊断

系统变量 ¹⁾	说明
@PRF_DMRT_CHNCON_<...>_TAG _WRITES_PENDING	尚未完成的已启动写入请求 该值持续增加表明系统过载。可能原因： • 数据源或连接过载或拥堵，导致对写入请求的处理速度不够快。
@PRF_DMRT_CHNCON_<...>_TAG _WRITES_PER_SECOND ²⁾	每秒写入的变量
@PRF_DMRT_CHNCON_<...>_TAG _WRITES_TOTAL	运行系统激活后写入的变量
@PRF_DMRT_CHNCON_<...>_TAG _WRITTEN_BYTES_PER_SECOND ²⁾	每秒写入的字节 位舍入为字节。 不包含元数据，例如时间戳或 SetValue 回调数据。
@PRF_DMRT_CHNCON_<...>_TAG _WRITTEN_BYTES_TOTAL	运行系统激活后写入的字节 位舍入为字节。 不包含元数据，例如时间戳或 SetValue 回调数据。

1) <...>代表通信连接的名称，例如"@PRF_DMRT_CHNCON_S7-417_TAG_READS_PER_SECOND"。

2) 信息"PER_SECOND"与变量更新前的最后一秒有关

参见

在运行系统中为连接状态组态变量 (页 155)

如何使用“状态 - 逻辑连接”功能来检查通道 (页 568)

7.3.3 如何使用“状态 - 逻辑连接”功能来检查通道

“逻辑连接状态”(Logical connections status) 功能

通过“逻辑连接状态”(Logical connections status) 功能，WinCC 项目管理器允许用户方便地显示所有已组态连接的当前状态。

但是，只能在运行系统中显示状态。

如果仅希望查询是否建立或终止某特定连接，可使用系统变量"@<Connectionname>@ConnectionStateEx"。

要求

- 在本通道中组态一个通道，并在该通道中创建一个变量。
- 已激活 WinCC 运行系统。

步骤

1. 在 WinCC 项目管理器菜单栏的“工具”(Tools) 菜单中，选择“驱动程序连接的状态”(Status of Driver Connections) 条目。
将打开“状态 - 逻辑连接”对话框。



2. 在“更新”区域中，可以指定周期性地执行更新。
否则，可以单击“更新”按钮手动启动显示的更新。
3. 组态的连接将显示在“名称”列中。
“状态”列显示相应连接的状态。
4. 检查“状态”列中的条目。
“断开连接”(Disconnected) 状态指示组态错误或硬件故障。
更多相关信息，请参见相关通道的“通道诊断”(Channel diagnosis)。

参见

在运行系统中为连接状态组态变量 (页 155)
使用性能变量检查连接 (页 565)

7.3.4 使用“通道诊断”进行通道诊断

7.3.4.1 通道诊断的原理

简介

在运行系统中，WinCC“通道诊断”为WinCC用户提供了快速浏览激活连接的状态的方法。一方面，“通道诊断”提供了有关通道单元的状态和诊断信息，另一方面，它可用作组态诊断输出的用户界面：

- 有关诸如过程画面中通讯状态的统计或信息的输出
- 文本输出到记录册文件中以用于错误分析和由服务进行纠正
- 文本输出到跟踪文件中以帮助热线查明通讯故障的原因

诊断模块可作为ActiveX控件插入过程画面，或作为独立的应用程序在Windows中启动。

该模块只显示支持诊断的通道的状态信息。

在针对通道诊断的描述中可以找到通道变量的诊断。

记录册文件

“通道诊断”将为每个组态的WinCC通道创建一个名为<ChannelName.log>的记录册文件。该文件将记录重要的信息和错误情况。确切的文本内容取决于通道。

文件和导出文本的创建不能组态。

记录册文件包含许多信息，例如：启动和结束消息、版本信息以及有关通讯错误的信息。

文件中的每个条目由日期和时间标志、标记名称和描述组成。文件总是在输入完成后立即保存，以确保（如发生断电时）所有的信息都可用。

跟踪文件

可以为每个组态的WinCC通道创建名为<ChannelName.trc>的跟踪文件，以输出附加信息和错误消息。可以在运行系统中选择是否使用跟踪文件。当激活该功能时，将显示消息以警告链接的运行时间会受影响。

跟踪文件中的每个条目都有时间标志，后面带有标记名称和描述。

当启用跟踪功能时，所有记录在记录册中的信息也将写入跟踪文件。

记录在跟踪文件内的信息有助于热线定位通讯故障的原因。

说明

跟踪和记录册文件条目只以英文记录。

两个文件都保存在 WinCC 目录结构的“诊断”(Diagnostics) 目录下。

当前计数器数值不会输出到这些文件中。

参见

使用 ActiveX 控件进行通道诊断 (页 571)

如何使用作为 ActiveX 控件的通道诊断来检查通道 (页 571)

使用“通道诊断”来诊断通道 (页 572)

如何使用通道诊断检查通道 (页 573)

如何组态通道的跟踪功能 (页 574)

如何启动运行系统 (页 575)

7.3.4.2 使用 ActiveX 控件进行通道诊断

引言

使用 ActiveX 控件“WinCC 通道诊断控件”也可以在过程画面中显示有关通道的状态信息。

ActiveX 控件可从图形编辑器的“Control”对象选项板得到并简单地插入到画面中。这样用户即可创建诊断过程画面，在其中用户可以查看运行系统中通讯的状态以及其它消息，而无需每次重新组态安排。

参见

如何使用通道诊断检查通道 (页 573)

使用“通道诊断”来诊断通道 (页 572)

7.3.4.3 如何使用作为 ActiveX 控件的通道诊断来检查通道

引言

本节说明如何使用“WinCC Channel Diagnosis Control”ActiveX 控件组态通道的诊断。

7.3 通道诊断

要求

- 在本通道中组态一个通道，并在该通道中创建一个变量。

步骤

1. 启动图形编辑器并打开画面。
2. 将“WinCC Channel Diagnosis Control”ActiveX 控件插入画面。通过从“控件”对象选项板中选择 ActiveX 控件，将其插入画面中并拖动到期望的大小，也可以实现。
3. 保存该画面。
4. 单击图形编辑器工具栏中的相应按钮来激活运行系统。
5. 选择插入了 ActiveX 控件的画面。通道的状态信息将出现在“通道诊断”应用程序窗口的“通道/连接”标签中。
6. 单击“组态”标签。选择其中一个显示的通道并组态哪些出错消息将记录到相关的日志文件中。
7. 在“组态通道的跟踪功能”处可以找到关于激活跟踪功能的更多信息。

说明

“通道诊断”只显示支持通道诊断的通道的状态信息。

参见

如何组态通道的跟踪功能 (页 574)

7.3.4.4 使用“通道诊断”来诊断通道

简介

“通道诊断”也可作为应用程序独立于 WinCC 通过 Windows 程序组“Siemens Automation”启动。

因此通道诊断总是可以使用且不受过程画面选择的约束，就如同“WinCC 通道诊断控件”的情况一样。

当 WinCC 处于运行系统中时，状态信息只由“通道诊断”显示。

参见

- 如何使用通道诊断检查通道 (页 573)
- 使用 ActiveX 控件进行通道诊断 (页 571)

7.3.4.5 如何使用通道诊断检查通道

简介

本节说明如何将“通道诊断”作为应用程序从 Windows“开始”(Start) 菜单处启动。

说明

“通道诊断”只显示支持通道诊断的通道的状态信息。

要求

- 在本通道中组态一个通道，并在该通道中创建一个变量。

操作步骤

- 在“Siemens Automation”Windows 程序组中选择“通道诊断”(Channel Diagnosis) 条目。将打开“通道诊断”(Channel Diagnosis) 应用程序窗口。如果运行系统中当前没有 WinCC 项目，则显示消息“没有连接到 WinCC”。
- 通过 WinCC 项目管理器的工具栏激活运行系统。
- 选择插入了 ActiveX 控件的画面。通道的状态信息将出现在“通道诊断”(Channel Diagnosis) 应用程序窗口的“通道/连接”(Channels/Connection) 选项卡中。
- 在“组态”(Configuration) 选项卡上，选择显示的连接中的一个。
- 组态要记录到相关日志文件中的错误消息。在“组态通道的跟踪功能”(Configuring a Channel's Trace Function) 下可以找到关于激活跟踪功能的更多信息。

参见

- 如何组态通道的跟踪功能 (页 574)
- 如何启动运行系统 (页 575)
- 使用 ActiveX 控件进行通道诊断 (页 571)
- 使用“通道诊断”来诊断通道 (页 572)

7.3.4.6 如何组态通道的跟踪功能

简介

本节介绍如何在运行系统中组态和激活通道的跟踪功能。

有关错误和通讯状态的附加信息将记录在跟踪文件内。

跟踪文件

记录在跟踪文件内的信息有助于热线定位通讯故障的原因。

因此本节未介绍文件的进一步评估。

OPC 跟踪

更多详细信息，请参见以下常见问题解答：

- 条目 ID 99412077：
“如何组态和启用 WinCC OPC 通道的跟踪？” (<https://support.industry.siemens.com/cs/cn/zh/view/99412077>)
- 条目 ID 99412263：
“如何组态和启用 WinCC OPC DA/HDA/A&E 服务器的跟踪？” (<https://support.industry.siemens.com/cs/cn/zh/view/99412263>)

要求

- 已组态通道，并在该通道下创建连接和变量。
- 激活运行系统。

标准标记概述

标志	含义
致命错误	致命错误（例如，需要用户干预）
错误 (Error)	错误（例如消息帧错误）
警告	警告（例如引用校验和错误）
信息	信息（例如函数调用）
成功	成功执行（例如完成函数调用）
选中用户标记	允许“用户标记”复选框

操作步骤

1. 从“开始”(Start) 菜单启动 WinCC 通道诊断。
2. 在“组态”标签上，选择期望的通道。
3. 在“标记”下，激活要记录在跟踪文件内的状态和错误消息。
有关标准标志的说明，请参见“标准标志 - 概述”(Standard Flags - Overview) 表。
4. 如果要将“用户标志”记录到跟踪文件，请选中“检查用户标志”(Check User Flags) 复选框。
“用户标志”的数量和含义取决于通道。
5. 选中期望的“用户标记”的复选框。
通过单击“置位”或“复位”按钮，可以置位或复位所有“用户标记”。
6. 在“跟踪文件”部分，选中“允许”复选框。
这样可以激活该区域内的其它域。
7. 在“最多文件”域中，输入跟踪文件的最大数目。
8. 在“最大大小”域中，设置单个跟踪文件的最大大小。
9. 如果要在达到最大文件数目和文件大小后覆盖通道现有的跟踪文件（从最早的开始），请激活“覆盖”(Overwrite) 字段。
10. 单击“保存”以保存设置并激活更改。

参见

如何检查组态数据 (页 615)

如何检查通道和连接 (页 616)

Internet: WinCC V7 OPC DA 通道跟踪 (<https://support.industry.siemens.com/cs/cn/zh/view/99412077>)

Internet: OPC/OPC UA 服务器跟踪 (WinCC V7 / WinCC Professional) (<https://support.industry.siemens.com/cs/cn/zh/view/99412263>)

7.3.4.7 如何启动运行系统

要求

激活运行系统前，必须定义启动画面。

步骤

1. 保存并关闭所有可能在编辑器中打开的文件。
2. 选择 WinCC 项目管理器。
3. 通过单击工具栏上的“激活”按钮或通过选择“文件”菜单中的“激活”来激活项目。

参见

如何使用通道诊断检查通道 (页 573)

7.4 "System Info"通道的诊断

7.4.1 "System Info"通道 - 诊断选项

可以使用"System Info"通道诊断的下列选项或其变量之一：

使用“通道诊断”诊断通道

“通道诊断”可以查询运行系统中通道和连接的状态。产生的任何错误都用“错误代码”显示。

诊断通道变量

在运行系统的变量管理器中，可以查询当前值、当前质量代码和变量改变的最后时刻。

参见

如何检查变量 (页 581)

如何检查通道和连接 (页 579)

7.4.2 日志文件条目的描述

引言

在日志文件中记录通道的错误和状态的重要变化。这些条目可以用于分析通讯问题。

文件中的每个条目都有日期和时间标志，后面带有标记名称和描述。

记录册条目实例：

```
2000-03-10 12:00:21,050 INFO Log starting ...
2000-03-10 12:00:21,050 INFO | LogFileName :C:\Siemens\WinCC\Diagnose
\SYSTEM_INFO_01.LOG
2000-03-10 12:00:21,050 INFO | LogFileCount :3
2000-03-10 12:00:21,050 INFO | LogFileSize :1400000
2000-03-10 12:00:21,050 INFO | TraceFlags :fa000001
```

7.4 "System Info"通道的诊断

2000-03-10 12:00:21,050 INFO start timer

2000-03-10 12:00:21,360 ERROR Illegal tag type! tag: "Format_0" correct type:"Text Tag 8-Bit Character Set"!

"INFO"标记条目

消息文本	描述
Log starting ...	起始消息
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	带路径日志文件的名称
LogFileCount : "n"	通道的日志文件的数目
LogFileSize : "x"	单个日志文件的大小（以字节为单位）
TraceFlags :fa000001	通道使用的十六进制格式标记
start timer	起始消息

"ERROR"标记条目

消息文本	描述
Illegal tag type! tag:"tag" correct type:"data type"!	变量的数据类型不正确 变量 = 数据类型错误的变量的名称 数据类型 = 正确的数据类型

7.4.3 确定错误变量值的原因

7.4.3.1 如何确定错误变量的原因

如果在运行系统中产生意外的变量值，进行如下处理以确定原因：

1. 检查通道和连接
2. 检查通道变量

参见

如何检查变量 (页 581)

如何检查通道和连接 (页 579)

7.4.3.2 如何检查通道和连接

简介

本节描述如何在运行系统中检查"System Info"通道及其连接。

要求

- 为"System Info"通道组态连接和变量。
- 激活 WinCC 项目。

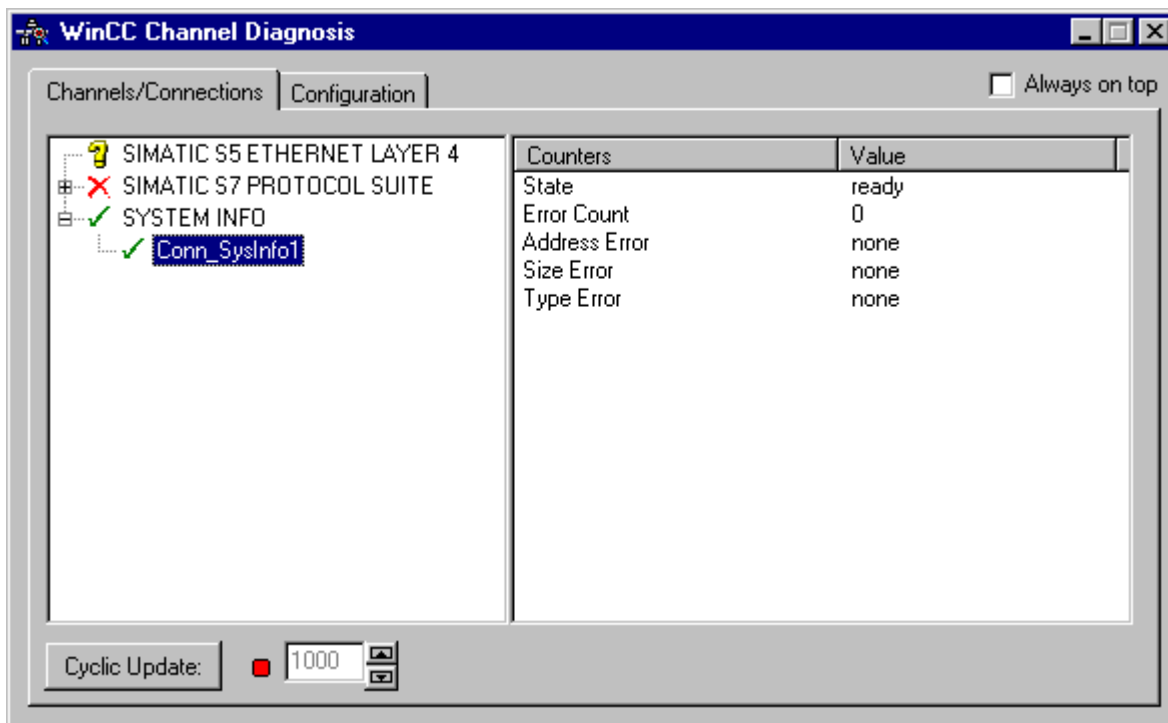
状态消息概述

图标	描述
	通道/连接无限制准备就绪
	通道/连接准备就绪，带有一些限制
	通道/没有关于连接状态的陈述
	通道/连接失败

7.4 “System Info”通道的诊断

步骤

1. 从“开始”(Start) 菜单启动 WinCC 通道诊断。
2. 将打开“通道诊断”(Channel Diagnosis) 应用程序窗口。所有安装的通道及其连接的状态信息显示在“通道/连接”(Channels/Connections) 选项卡的左侧。



3. 检查通道“System Info”前的符号及其连接。如果通道和连接状况正常，将有一个绿色的复选标记显示在相应条目的前面。有关各个图标含义的信息，请参见表格“状态消息概述”。
4. 如果通道名称和连接前面没有绿色复选标记，则选择左侧窗口中的连接。在窗口右边，检查“地址错误”、“大小错误”和“类型错误”的计数器数值。这些值表示检测到的错误。
5. 检查通道特定的日志文件。要执行此操作，请使用文本编辑器打开目录“Siemens\WinCC \Diagnose”中的文件。检查带“ERROR”标记的最新条目。有关该主题的更多信息，请参阅“日志文件条目的描述”。
6. 如果检查日志文件之后仍然不能查明错误，请激活跟踪功能并联系客户支持部门。有关该主题的更多信息，请参见“组态通道的跟踪功能”。

参见

- 如何组态通道的跟踪功能 (页 574)
- 日志文件条目的描述 (页 577)
- 如何检查变量 (页 581)

7.4.3.3 如何检查变量

简介

如果运行系统中外部变量不具有预期的数值，请使用下列步骤来检查变量。

要求

- 为"System Info"通道组态连接和变量。
- 激活 WinCC 项目。

步骤

1. 在 WinCC 项目管理器的变量管理器中，选择"System Info"通道。
2. 在数据窗口中，选择希望检查的外部变量。为此，打开目录结构直到此变量显示在表格区域中。
3. 移动鼠标指针，指向要检查的变量。将打开工具提示窗口，显示当前变量值、质量代码和最近一次数值改变的时间。
4. 检查质量代码。如果显示数值"80"，则该变量值正确。其它数值的描述可以参见“变量质量代码”。
5. 如果质量代码不等于"80"，从变量管理器中选择变量，然后单击快捷菜单中的“属性”(Properties) 来打开“变量属性”(Tag Properties) 对话框。
6. 检查是否已在“限制/报告”(Limits/Reportin) 选项卡上组态了上限或下限、启动或替换值。这些数值会影响显示。
7. 如果变量值被其中一个已组态的数值改变，则取消激活该项目并改变该限制值或替换值。

说明

变量值、质量代码等只在运行期间显示。

参见

变量的质量代码 (页 622)

7.5 诊断通道“SIMATIC S7-1200/S7-1500”

7.5.1 通过系统诊断控件进行系统诊断

概述

系统诊断可显示 S7-1200 和 S7-1500 控制器的故障和错误。

通过 WinCC 系统诊断控件，WinCC 提供了“SIMATIC S7-1200, S7-1500 Channel”通信通道中的快速错误定位概述。

可组态从控制器状态消息到系统诊断控件中诊断概览的直接导航。其中会显示控制器错误的详细信息。

系统诊断视图

WinCC 系统诊断控件中以下视图可用：

- 诊断概述
- 详细视图
- 诊断缓冲区视图

系统诊断显示也提供显示的拆分视图。您可以同时查看控制器和相关的详细信息。

区域上方显示系统诊断和诊断缓冲区视图。

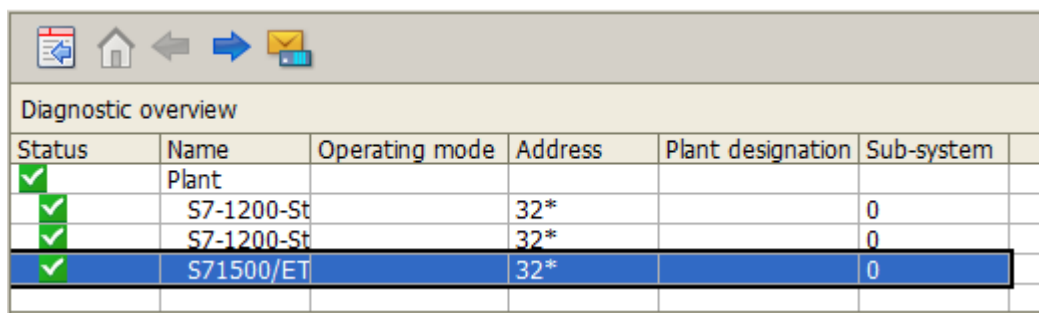
区域下方显示详细视图。

诊断概述

诊断概述显示所有可用的 S7-1200/1500 通道。

双击控制器打开详细视图。

第一列中的符号给出了关于控制器当前状态的信息。



Status	Name	Operating mode	Address	Plant designation	Sub-system
✓	Plant				
✓	S7-1200-St		32*		0
✓	S7-1200-St		32*		0
✓	S71500/ET		32*		0

详细视图

详细视图给出了关于所选控制器的详细信息。

检查详细视图中的数据是否正确。在详细视图中无法对错误文本进行排序。

下图显示了诊断概述和详细视图的拆分视图。

Status	Name	Operating mode	Address	Plant designation	Sub-system
✓	Plant				
✓	S7-1200-St		32*		0
✓	S7-1200-St		32*		0
✓	S71500/ET		32*		0

Property	Value
> Status	✓ Good
> Name	S71500/ET200MP-Station_1
> Operating mode	N/A
> Rack	
> Slot	
> Type	S71500/ET200MP-Station
> Item number	
> Address	32*
> Plant designation	
> Location identifier	
> Sub-system	0
> Station	
> Subslot	
> Sub-address	
> Software version	
> Installation	
> Additional information	
> Error Text	
> Manufacturer ID	
> Hardware version	
> Profile ID	
> Specific profile details	
> I&M data version	
> Serial number	
> Revision Counter	

诊断缓冲区视图

诊断缓冲区视图显示了控制器诊断缓冲区中的当前数据。

仅可以在诊断概述中调用诊断缓冲区视图。

要更新诊断缓冲区视图，请选择“更新”(Update)按钮。



No.	Date	Time	Event
1	4/3/2015	7:57:01 PM	
2	4/3/2015	7:54:51 PM	
3	3/20/2015	7:07:24 PM	
4	3/20/2015	6:57:44 PM	
5	3/9/2015	9:50:13 AM	
6	3/7/2015	4:50:37 PM	
7	3/7/2015	4:45:22 PM	
8	3/7/2015	4:45:22 PM	
9	3/7/2015	4:45:22 PM	
10	3/7/2015	4:45:17 PM	
11	3/7/2015	4:44:40 PM	
12	3/6/2015	3:59:29 PM	
13	3/2/2015	10:37:09 AM	
14	3/2/2015	10:39:36 AM	
15	3/1/2015	3:06:48 AM	
16	3/1/2015	2:59:38 AM	
17	2/26/2015	12:18:15 PM	

No.: 1
Date: 4/3/2015
Time: 7:57:01 PM

系统诊断视图中的按钮

按钮	功能
	打开组态对话框，可在其中更改系统诊断控件的属性。
	打开子设备，如果没有子设备，则打开详细视图。
	打开父设备，如果没有父设备，则打开诊断概述。
	打开诊断概述。
	打开诊断缓冲区视图。 仅在诊断概述中可见。
	更新诊断缓冲区视图。
	打开一个对话框，用于设置所显示诊断概述列的用户定义排序标准。

7.5 诊断通道“SIMATIC S7-1200/S7-1500”

按钮	功能
	启动显示值的打印输出。 用于打印的打印作业在组态对话框的“常规”(General) 选项卡上定义。
	此按钮用于将全部或选定的运行系统数据导出到“CSV”文件。 如果激活“显示对话框”(Display dialog) 选项，则会打开一个对话框，从中可查看导出设置并启动导出。如果具有相应权限，还可以选择要导出的文件和目录。 如果显示了一个对话框，则立即开始将数据导出到预定义文件。

参见

“SIMATIC S7-1200, S7-1500 Channel”通道 (页 471)

如何组态系统诊断 (页 586)

7.5.2 如何组态系统诊断

简介

控制器中的故障和错误显示在运行系统的各种系统诊断视图中。

为此应在“图形编辑器”中组态 WinCC SysDiagControl。

要求

- 在“SIMATIC S7-1200, S7-1500 Channel”下的“OMS+”通道单元中创建了连接。
- 在服务器的启动列表中激活了报警记录。
- 要在诊断缓冲区视图中显示 S7-1500 通道的消息和文本，必须满足下列附加条件：
 - 在 WinCC 项目中载入控制器的 AS 消息和 AS 文本列表。
 - 必须在报警记录中为 AS 文本列表选择“已使用”(Used) 选项。
 - 必须在报警记录中为“Notify_AP”类型的诊断消息组态已定义的确认原则：
必须将诊断消息分配给无需确认但拥有“已离开”(Went Out) 状态的消息类型。
- 自动更新 S7-1500 消息的要求：
 - 在 WinCC 项目中载入一次控制器的 AS 消息和 AS 文本列表。
因此，所需文本存储在文本库中。

可在“使用 WinCC > 设置消息系统 > 组态消息系统 > AS 消息”(Working with WinCC > Setting up a message system > Configuring the message system > AS messages) 下查看详细信息。

组态步骤

1. 将 WinCC 系统诊断控件插入到图形编辑器的过程画面中。
2. 在“常规”(General) 选项卡中, 组态系统诊断控件的基本属性。
 - 诊断窗口的属性
 - 控件的常规属性
 - 控件的时间基准
3. 在“列”(Columns) 选项卡中, 指定控制器数据以列或行的形式显示在系统诊断的视图中。
4. 使用排序对话框确定数据排序采用的列。
更多详细信息, 请参见以下位置的 WinCC UserArchiveControl 示例:
 - “选项 > 用户归档 > WinCC 用户归档控件 > 运行期间的操作 > 如何对用户归档数据的显示进行排序”
5. 在“参数”(Parameter)、 “显示”(Display) 和 “标记”(Marker) 选项卡中, 组态表格的显示和属性。
更多详细信息, 请参见:
 - “使用 WinCC > 设置消息系统 > 在运行期间显示消息 > 组态报警控件 > 如何组态表格显示”
6. 在相应选项卡中, 组态表格窗口的工具栏和状态栏。
更多详细信息, 请参见:
 - “使用 WinCC > 设置消息系统 > 在运行期间显示消息 > 组态报警控件 > 如何组态工具栏和状态栏”
7. 如果希望从 WinCC AlarmControl 中的 AS 消息直接跳转到 WinCC SysDiagControl, 可以通过脚本在画面中组态按钮:
 - 在画面中插入按钮。
例如, 以事件形式创建在鼠标单击时执行动作的脚本。
 - 当 WinCC AlarmControl “AlarmControl_1”和 WinCC SysDiagControl “SysDiagControl_1”在同一画面时, 可以使用下列脚本示例:
使用 C:

```
SetPropChar (lpszPictureName, "SysDiagControl_1", "NavigateTo",  
GetPropChar (lpszPictureName, "AlarmControl_1", "DiagnosticsContext"));
```


使用 VBS:

```
ScreenItems ("SysDiagControl_1").NavigateTo =  
ScreenItems ("AlarmControl_1").DiagnosticsContext
```
8. 保存组态数据。

参见

通过系统诊断控件进行系统诊断 (页 582)

7.6 “SIMATIC S7 Protocol Suite”通道的诊断

7.6.1 “SIMATIC S7 Protocol Suite”通道 - 诊断选项

有以下选项可用于错误检测和“SIMATIC S7 Protocol Suite”通道或其变量之一的诊断:

检查通讯处理器组态

除了检查相应访问点，可以使用应用程序“设置 PG/PC 接口”来测试通讯处理器。也可以在 SIMATIC NET 中以相同的方式检查通讯处理器。

检查连接和变量的组态

系统和连接参数的组态可能出现错误。无效变量值还可能由于错误引用 AS 中的变量编址而引起。

使用“通道诊断”诊断通道

“通道诊断”可以查询运行系统中通道和连接的状态。产生的任何错误都用“错误代码”显示。

诊断通道变量

在运行系统的变量管理器中，可以查询当前值、当前质量代码和变量改变的最后时刻。

参见

如何检查变量 (页 598)

如何检查通道和连接 (页 596)

如何检查连接和变量的组态 (页 595)

检查 SIMATIC NET 中的通讯处理器 (页 594)

如何检查通信处理器的组态 (页 592)

7.6.2 日志文件条目的描述

引言

在日志文件中记录通道的错误和状态的重要变化。这些条目可以用于分析通讯问题。

文件中的每个条目都有日期和时间标志，后面带有标记名称和描述。

记录册条目实例：

```
01.04.99 12:00:24.524 INFO Log starting ...
1999-04-01 12:00:24,524 INFO LogFileName :C:\Siemens\WinCC\Diagnose
\SIMATIC_S7_Protocol_Suite_01.LOG
1999-04-01 12:00:24,524 INFO LogFileCount :3
1999-04-01 12:00:24,524 INFO LogFileSize :1400000
1999-04-01 12:00:24,524 INFO TraceFlags :c4000000
1999-04-01 12:00:24,524 INFO S7 channel DLL started!
1999-04-01 12:00:26,096 ERROR Illegal tag address "nCPU3_1"!
1999-04-01 12:00:27,428 INFO S7DOS release: @(#)TIS-Block Library DLL Version
C5.0.17.3-REL5,0,17,47,3-BASIS
1999-04-01 12:00:27,428 INFO S7DOS version:V5.0 / 0
1999-04-01 12:00:27,428 INFO S7CHN version:V5.0 / Mar 1 1999 / 22:36:40
1999-04-01 12:00:27,428 INFO S7 channel unit "Industrial Ethernet" activated!
1999-04-01 12:00:27,468 ERROR Cannot connect to "CPU_4":Errorcode 0xFFDF 42C2!
1999-04-01 12:00:27,538 INFO S7 channel unit "MPI" activated!
```

“INFO”标记中最重要条目的描述

消息文本	描述
LogFileName : C:\Siemens\WinCC\Diagnose\ "channel_name".LOG	带路径日志文件的名称
LogFileCount : "n"	通道的日志文件的数目
LogFileSize : "x"	单个日志文件的大小（以字节为单位）
TraceFlags :c4000000	用十六进制数字显示跟踪功能使用的标记

7.6 "SIMATIC S7 Protocol Suite"通道的诊断

消息文本	描述
S7 channel DLL started!	起始消息
S7 channel DLL terminated!	结束消息
S7 channel unit "unitname" activated!	通道单元激活
S7 channel unit "unitname" deactivated!	通道单元取消激活
S7DOS version:versionsstring	版本信息
S7CHN version:versionsstring	版本信息

"ERROR"标记中最重要条目的描述

消息文本	描述
Cannot connect to "connectionname": Errorcode 0xhhhh ffff!	<p>通讯错误</p> <p>激活 WinCC 后，无法立即建立与自动化系统的连接。如果可以至少一次没有错误地建立连接，以后出错时，则输出下列信息。</p> <p>nnn = 本连接的断开次数 连接名称 = 连接的名称 hhh = 第一个错误代码，十六进制 S7DOS/SAPI-S7 ffff = 第二个错误代码，十六进制 S7DOS/SAPI-S7</p>
Cannot connect to "connectionname": Errorcode 0xhhhh ffff!	<p>通讯错误</p> <p>激活 WinCC 后，无法立即建立与自动化系统的连接。至少一次正确地建立了连接。</p>
Channel API error:errorstring	<p>通道 API 出错</p> <p>通道将错误字符串"errorstring"传到 WinCC 项目管理器。根据错误的重要程度，错误字符串可能显示也可能不显示在注意框中。至于错误字符串的说明，请参见 API 错误文本。</p>
Max. count of API errors reached - API logbook deactivated	<p>通道 API 出错</p> <p>根据错误和功能，可能在 API 上周周期性发生错误。要避免这些错误信息占满记录册文件，一个 API 错误最多可以输出 32 条信息。</p>

消息文本	描述
Cannot write storage data! Cannot read storage data / use default data Storage data illegal or destroyed / use default data! No storage data / use default data!	常规通道错误消息
单元“单元名”中的设备名从“旧设备名”改为“新设备名”	初始化消息
达到记录册极限大小 - 记录册被取消激活	日志文件超出其最大长度时发送的消息。监控记录册输出的长度。如果达到了指定的长度，记录册将取消激活。只有在消息输出导致超出最大文件长度时才会输出消息。如果使用编辑器改变文件长度或在 INI 文件中减少最大文件长度，不会输出任何消息！

7.6.3 确定错误变量值的原因

7.6.3.1 如何确定错误变量的原因

如果在运行系统中产生意外的变量值，进行如下处理以确定原因：

1. 检查通讯处理器的组态
2. 检查 SIMATIC NET 中的通讯处理器
3. 检查连接和变量的组态
4. 检查通道和连接
5. 检查通道变量

参见

如何检查变量 (页 598)

如何检查通道和连接 (页 596)

如何检查连接和变量的组态 (页 595)

检查 SIMATIC NET 中的通讯处理器 (页 594)

如何检查通信处理器的组态 (页 592)

7.6 “SIMATIC S7 Protocol Suite”通道的诊断

7.6.3.2 如何检查通信处理器的组态

简介

本部分说明如何使用“PG/PC 端口”程序检查通信处理器。在本示例中，“CP 5613 A3”类型的处理器用于 PROFIBUS 通信。

要求

- 安装 CP 5613 A3。
- 安装相关的通信驱动程序。
- 组态 CP 5613 A3。

步骤

1. 在控制面板中，单击“设置 PG/PC 端口”(Set PG/PC Port) 图标。将打开“设置 PG/PC 端口”(Set PG/PC Port) 对话框。
2. 检查访问点的条目。安装 CP 5613 A3 后，将自动为 Profibus 连接添加访问点“CP_L2_1:”。为该访问点选择相关条目。单击“属性”(Properties) 打开“属性 - CP5613A3.PROFIBUS.1” (Properties - CP5613A3.PROFIBUS.1) 对话框。



图 7-1 示例组态：CP 5623

3. 检查“PROFIBUS”选项卡上的条目。
4. 单击“操作状态”(Operational State) 选项卡。单击“测试”(Test) 对 CP 5613 A3 执行功能测试。测试结果随即显示在下面的输出域中。根据测试结果，可单击“重启”(Restart) 按钮执行 CP 5613 A3 的复位和完全重启。
5. 单击“确定”(OK) 关闭所有打开的对话框。

参见

如何检查连接和变量的组态 (页 595)

检查 SIMATIC NET 中的通讯处理器 (页 594)

7.6.3.3 检查 SIMATIC NET 中的通讯处理器

简介

本节说明如何使用 SIMATIC NET 软件中的“设置 PC 站”程序检查通信处理器。

在本示例中，“CP 5613 A3”类型用于与“SIMATIC S7 Protocol Suite”通道的 PROFIBUS 通信。

要求

- 安装 CP 5613 A3。
- 安装 SIMATIC NET 软件。
- 在 SIMATIC NET 下组态 CP 5613 A3。

操作步骤

1. 在 SIMATIC NET 设置中打开菜单项“设置 PC 站”(Set PC station)。将打开“组态控制台 PC 站”(Configurations Console PC Station) 对话框。
2. 检查访问点的条目。在浏览窗口中选择“访问点”(Access Point) 目录。已存在的访问点显示在数据窗口中。安装 CP 5613 A3 时，自动为 Profibus 连接插入访问点“CP_L2_1:”。在数据窗口中选择该访问点。使用快捷菜单中的“属性”(Properties) 菜单项打开“CP_L2_1 的属性:”(Properties of CP_L2_1:) 对话框。
3. 检查“分配的接口参数”(Assigned Interface Parameters) 字段内的条目。对于 PROFIBUS 网络中的 CP 5613 A3，应选择条目“CP5613A3.PROFIBUS.1”。
4. 打开浏览窗口，选择“组件”(Components) 目录，然后选择“CP5613 A3”子目录。
5. 选择“网络诊断”(Network Diagnosis) 目录。单击“测试”(Test) 对 CP 5613 A3 执行功能测试。测试结果显示在输出窗口中。根据测试结果，单击“常规”(General) 目录中的“重启”(Restart) 以执行 CP 5613 A3 的复位以及随后的完全重启。
6. 检查连接到“总线成员”(Bus Participants) 目录列表中 PROFIBUS 的成员列表。根据显示，可确定您自己工作站的功能和状态以及其它连接成员的功能和状态。
7. 关闭对话框。
8. 如果在通信处理器组态中检测到故障，只能使用 SIMATIC NET 工具修改组态。详细信息请参见 SIMATIC NET。

参见

如何检查连接和变量的组态 (页 595)

7.6.3.4 如何检查连接和变量的组态

简介

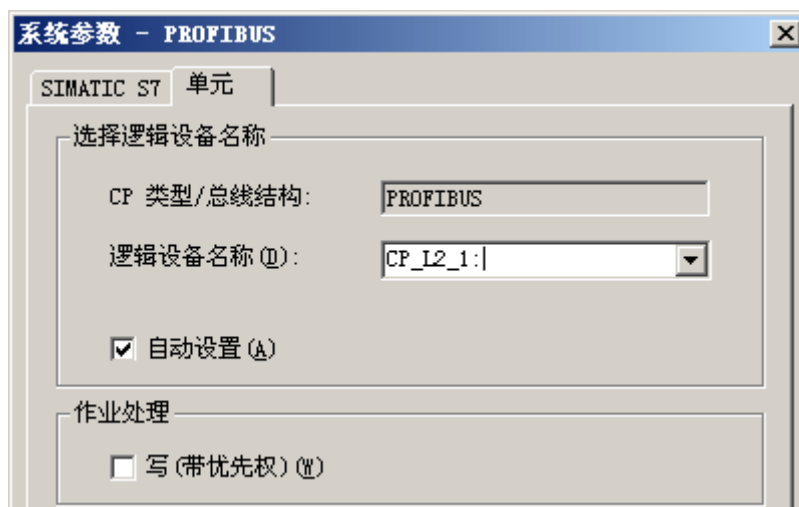
本节说明如何检查系统参数以及连接和变量的组态。在本示例中，“CP 5613 A3”通信处理器用于 PROFIBUS 通信。

要求

- 安装 CP 5613 A3。
- 安装相关的通信驱动程序。
- 组态 CP 5613 A3。
- 为“SIMATIC S7 Protocol Suite”通道组态连接和变量。
- 激活 WinCC 项目。

步骤

1. 在 WinCC 项目管理器的变量管理器中，选择“SIMATIC S7 Protocol Suite”通道。在数据窗口中，选择“PROFIBUS”通道单元。在通道单元快捷菜单上单击“系统参数”。将打开“系统参数 - PROFIBUS”对话框。
2. 在“单元”标签上，检查“逻辑设备名称”域中的条目。默认状态下，此条目设置为访问点“CP_L2_1:”在 CP 5613 A3 系统中安装通信处理器期间分配访问点。关闭对话框。



3. 在变量管理器浏览窗口中，选择“PROFIBUS”通道单元。在数据窗口中，选择要检查的连接。在快捷菜单中，单击“属性”打开“连接属性”对话框。
4. 单击“属性”按钮以打开“连接参数 - PROFIBUS”对话框。
5. 检查“连接”标签上的设置。关闭打开的对话框。

7.6 “SIMATIC S7 Protocol Suite”通道的诊断

6. 在浏览窗口中，选择经检查的连接。在数据窗口中，选择要检查的变量。在快捷菜单中，单击“属性”打开“变量属性”对话框。检查“类型转换”和“数据类型”域中的数值。
7. 单击“选择”按钮以打开“地址属性”对话框。检查自动化系统中变量的地址设置。
8. 单击“确定”(OK) 关闭所有打开的对话框。

参见

如何检查通道和连接 (页 596)

7.6.3.5 如何检查通道和连接





简介

本节说明如何在运行系统中检查“SIMATIC S7 Protocol Suite”通道及其连接。

要求

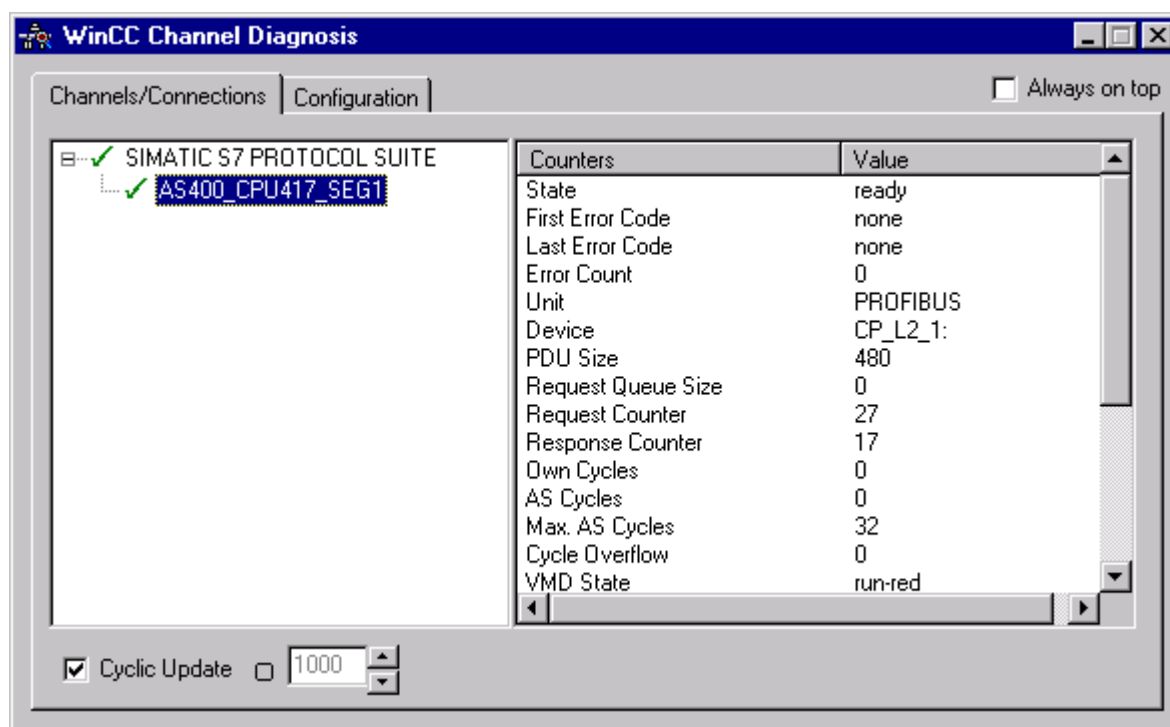
- 在 WinCC 计算机中安装通信处理器，例如用于 PROFIBUS 通信的 CP 5613 A3。
- 安装相关的通信驱动程序。
- 组态 CP 5613 A3。
- 创建 STEP7 项目。
- 为“SIMATIC S7 Protocol Suite”通道组态连接和变量。
- 激活 WinCC 项目。

状态消息概述

图标	描述
	通道/连接无限制准备就绪
	通道/连接准备就绪，带有一些限制
	通道/没有关于连接状态的陈述
	通道/连接失败

步骤

1. 从“开始”(Start) 菜单启动 WinCC 通道诊断。
2. 将打开“通道诊断”(Channel Diagnosis) 应用程序窗口。所有安装的通道及其连接的状态信息显示在“通道/连接”(Channels/Connections) 选项卡的左侧。



3. 选中名称为“SIMATIC S7 Protocol Suite”的通道和连接前面的图标。如果通道和连接状况良好，将有一个绿色的复选标记显示在相应条目的前面。有关各个图标含义的信息，请参见表格“状态消息概述”。
4. 如果通道名称和连接前面没有绿色复选标记，则选择左侧窗口中的连接。在右边的窗口中，检查计数器“第一个错误代码”和“最后一个错误代码”的条目。这些值表示检测到的错误。单击数值显示的快捷菜单可以访问直接帮助。
5. 在“组态”标签上，选择将输出到日志文件的状态和错误消息。这通过选择“SIMATIC S7 Protocol Suite”并组态出错显示得以实现。有关详细信息，请参见“组态通道的日志文件”。
6. 检查通道特定的日志文件。要执行此操作，请使用文本编辑器打开目录“Siemens\WinCC \Diagnose”中的文件。检查带“ERROR”标记的最新条目。有关该主题的更多信息，请参见“日志文件条目的描述”。
7. 如果检查日志文件之后仍然不能查明错误，请激活跟踪功能并联系客户支持部门。有关该主题的更多信息，请参见“组态通道的跟踪功能”。

7.6 “SIMATIC S7 Protocol Suite”通道的诊断

参见

如何组态通道的跟踪功能 (页 574)

日志文件条目的描述 (页 589)

如何检查变量 (页 598)

7.6.3.6 如何检查变量

简介

如果运行系统中外部变量不具有预期的数值，请使用下列步骤来检查变量。

在“SIMATIC S7 Protocol Suite”通道中，也可以使用连接专用的内部变量。该步骤也可以用于检查这些变量。

要求

- 在 WinCC 计算机中安装通信模块，例如用于 MPI 通信的 CP 5613 A3。
- 安装相关的通信驱动程序。
- 组态 CP 5613 A3。
- 创建 STEP7 项目。
- 为“SIMATIC S7 Protocol Suite”通道组态连接和变量。
- 激活 WinCC 项目。

步骤

1. 在 WinCC 项目管理器的变量管理器中，选择“SIMATIC S7 Protocol Suite”通道。
2. 在数据窗口中，选择希望检查的外部变量。为此，打开目录结构直到此变量显示在表格区域中。
3. 移动鼠标指针，指向要检查的变量。将打开工具提示窗口，显示当前变量值、质量代码和最近一次数值改变的时间。
4. 检查质量代码。如果显示数值“80”，则该变量值正确。其它数值的描述可以参见“变量质量代码”。
5. 如果质量代码不等于“80”，从变量管理器中选择变量，然后单击快捷菜单中的“属性”(Properties) 来打开“变量属性”(Tag Properties) 对话框。
6. 检查是否已在“限制/报告”(Limits/Reportin) 选项卡上组态了上限或下限、启动或替换值。这些数值会影响显示。
7. 如果变量值被其中一个已组态的数值改变，则取消激活该项目并改变该限制值或替换值。

说明

在运行系统中，可以使用“WinCC 通道诊断”工具详细查看连接专用的内部变量的当前值。选择主连接后，变量将显示在“计数器”列中。

变量值、质量代码等只在运行期间显示。

参见

变量的质量代码 (页 622)

7.7 “SIMATIC S5 Profibus FDL”通道的诊断

7.7.1 “SIMATIC S5 PROFIBUS FDL”通道的诊断选项

以下选项可用于错误检测和“SIMATIC S5 PROFIBUS FDL”通道诊断或其变量诊断：

检查通讯处理器组态

除了检查相应访问点，可以使用应用程序“设置 PG/PC 接口”来测试通讯处理器。也可以在 SIMATIC NET 中以相同的方式检查通讯处理器。

检查连接和变量的组态

系统和连接参数的组态可能出现错误。无效变量值还可能由于错误引用 AS 中的变量编址而引起。

使用“通道诊断”诊断通道

“通道诊断”可以查询运行系统中通道和连接的状态。产生的任何错误都用“错误代码”显示。

诊断通道变量

在运行系统的变量管理器中，可以查询当前值、当前质量代码和变量改变的最后时刻。

参见

如何检查变量 (页 609)

如何检查通道和连接 (页 606)

如何检查连接和变量的组态 (页 605)

检查 SIMATIC NET 中的通讯处理器 (页 604)

如何检查通讯处理器的组态 (页 602)

7.7.2 日志文件条目的描述

引言

在日志文件中记录通道的错误和状态的重要变化。这些条目可用于分析通讯问题。

文件中的每个条目都有日期和时间标志，后面带有标记名称和描述。

记录册条目实例：

```
2000-05-03 14:43:48,733 INFO Log starting ...
2000-05-03 14:43:48,733 INFO | LogFileName :d:\Siemens\WinCC\Diagnose
\SIMATIC_S5_PROFIBUS_FDL_01.LOG
2000-05-03 14:43:48,733 INFO | LogFileCount :3
2000-05-03 14:43:48,733 INFO | LogFileSize :1400000
2000-05-03 14:43:48,733 INFO | TraceFlags :fa017fff
```

"INFO"标记条目的描述

消息文本	描述
Log starting ...	起始消息
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	带路径日志文件的名称
LogFileCount : "n"	通道的日志文件的数目
LogFileSize : "x"	单个日志文件的大小（以字节为单位）
TraceFlags :fa017fff	用十六进制数字显示跟踪功能使用的标记

7.7.3 确定错误变量值的原因

7.7.3.1 如何确定错误变量的原因

如果在运行系统中产生意外的变量值，进行如下处理以确定原因：

1. 检查通讯处理器的组态
2. 检查 SIMATIC NET 中的通讯处理器
3. 检查连接和变量的组态

7.7 “SIMATIC S5 Profibus FDL”通道的诊断

4. 检查通道和连接
5. 检查通道变量

参见

- 如何检查连接和变量的组态 (页 605)
- 如何检查变量 (页 609)
- 如何检查通道和连接 (页 606)
- 检查 SIMATIC NET 中的通讯处理器 (页 604)
- 如何检查通讯处理器的组态 (页 602)

7.7.3.2 如何检查通讯处理器的组态

简介

本部分说明如何使用“PG/PC 端口”程序检查通信处理器。在本示例中，“CP 5613 A3”类型的处理器用于 PROFIBUS 通信。

要求

- 安装 CP 5613 A3。
- 安装相关的通信驱动程序。
- 组态 CP 5613 A3。

步骤

1. 在控制面板中，单击“设置 PG/PC 端口”(Set PG/PC Port) 图标。将打开“设置 PG/PC 端口”(Set PG/PC Port) 对话框。
2. 检查访问点的条目。安装 CP 5613 A3 后，将自动为 Profibus 连接添加访问点“CP_L2_1:/SCP”。为该访问点选择相关条目。单击“属性”(Properties) 打开“属性 - CP5613A3.PROFIBUS.1” (Properties - CP5613A3.PROFIBUS.1) 对话框。



图 7-2 示例组态：CP 5623

3. 检查“PROFIBUS”选项卡上的条目。
4. 单击“操作状态”(Operational State) 选项卡。单击“测试”(Test) 对 CP 5613 A3 执行功能测试。测试结果随即显示在下面的输出域中。根据测试结果，可单击“重启”(Restart) 按钮执行 CP 5613 A3 的复位和完全重启。
5. 单击“确定”(OK) 关闭所有打开的对话框。

参见

如何检查连接和变量的组态 (页 605)

检查 SIMATIC NET 中的通讯处理器 (页 604)

7.7.3.3 检查 SIMATIC NET 中的通讯处理器

简介

本节说明如何使用 SIMATIC NET 软件中的“设置 PC 站”程序检查通信处理器。

在本示例中，“CP 5613 A3”类型用于与通道“SIMATIC S5 PROFIBUS FDL”的 PROFIBUS 通信。

要求

- 安装 CP 5613 A3。
- 安装 SIMATIC NET 软件。
- 在 SIMATIC NET 下组态 CP 5613 A3。

操作步骤

1. 在 SIMATIC NET 设置中打开菜单项“设置 PC 站”(Set PC station)。将打开“组态控制台 PC 站”(Configurations Console PC Station) 对话框。
2. 检查访问点的条目。在浏览窗口中选择“访问点”(Access Point) 目录。已存在的访问点显示在数据窗口中。安装 CP 5613 A3 时，自动为 Profibus 连接插入访问点“CP_L2_1:”。在数据窗口中选择该访问点。使用快捷菜单中的“属性”(Properties) 菜单项打开“CP_L2_1 的属性:”(Properties of CP_L2_1:) 对话框。
3. 检查“分配的接口参数”(Assigned Interface Parameters) 字段内的条目。对于 PROFIBUS 网络中的 CP 5613 A3，应选择条目“CP5613A3.PROFIBUS.1”。
4. 打开浏览窗口，选择“组件”(Components) 目录，然后选择“CP5613 A3”子目录。
5. 选择“网络诊断”(Network Diagnosis) 目录。单击“测试”(Test) 对 CP 5613 A3 执行功能测试。测试结果显示在输出窗口中。根据测试结果，单击“常规”(General) 目录中的“重启”(Restart) 以执行 CP 5613 A3 的复位以及随后的完全重启。
6. 检查连接到“总线成员”(Bus Participants) 目录列表中 PROFIBUS 的成员列表。根据显示，可确定您自己工作站的功能和状态以及其它连接成员的功能和状态。
7. 关闭对话框。
8. 如果在通信处理器组态中检测到故障，只能使用 SIMATIC NET 工具修改组态。详细信息请参见 SIMATIC NET。

参见

如何检查连接和变量的组态 (页 605)

7.7.3.4 如何检查连接和变量的组态

简介

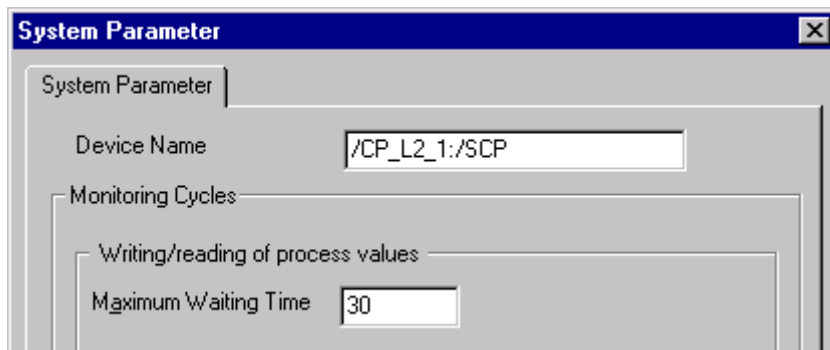
本节说明如何检查系统参数以及连接和变量的组态。

要求

- 安装 CP 5613 A3。
- 安装通信驱动程序。
- 组态 CP 5613 A3。
- 为“SIMATIC S5 PROFIBUS FDL”通道组态连接和变量。
- 激活 WinCC 项目。

步骤

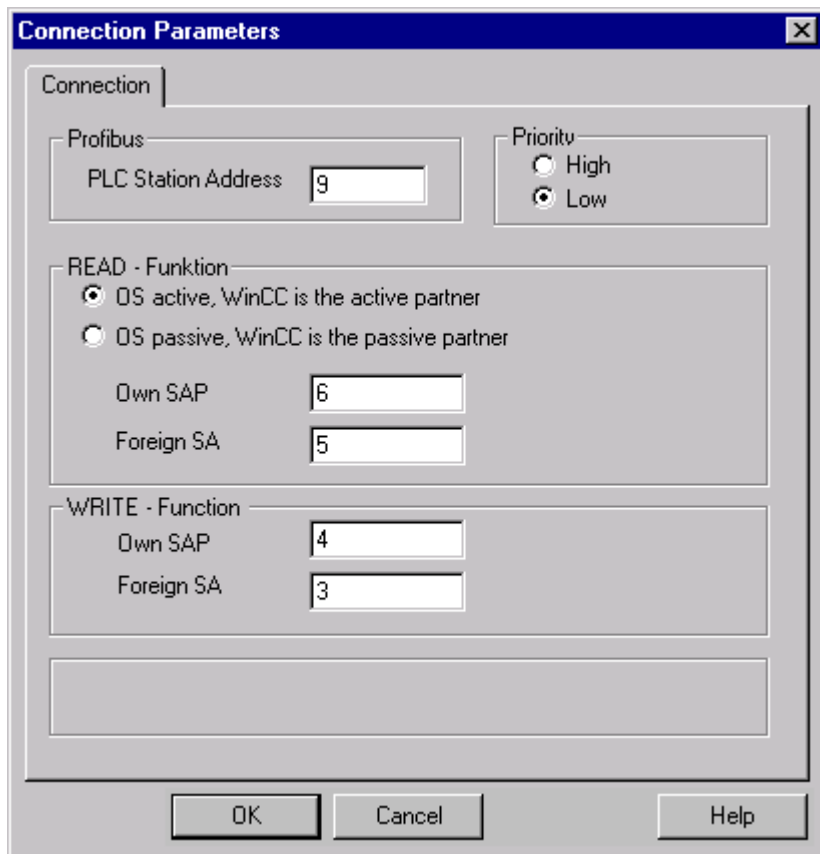
1. 在 WinCC 项目管理器浏览窗口中，单击“SIMATIC S5 PROFIBUS FDL”图标前的加号。在“FDL(CP5412/A2-1)”图标的快捷菜单中，单击“系统参数”(System Parameters)。将打开“System Parameters”（系统参数）对话框。



2. 检查“Device Name”（设备名称）域中的条目。缺省状态下，设置的访问点是“CP_L2_1:/SCP”。在 CP 5613 A3 系统中安装通信处理器期间分配访问点。关闭对话框。
3. 单击“FDL (CP5412/A2-1)”图标前的加号。在待测试变量的快捷菜单中，选择条目“属性”(Properties)。将打开“连接属性”对话框。

7.7 “SIMATIC S5 Profibus FDL”通道的诊断

- 在“连接属性”对话框中单击“属性”按钮。将打开“连接参数”对话框。



- 检查“连接”标签上的设置。关闭打开的对话框。
- 单击连接图标前的加号。在待测试变量的快捷菜单中，单击条目“属性”(Properties)。将打开“变量属性”对话框。检查“类型转换”和“数据类型”域中的条目。
- 在“变量属性”对话框中，单击“选择”按钮。将打开“地址属性”对话框。检查这些设置。
- 单击“确定”(OK) 关闭所有打开的对话框。

参见

如何检查通道和连接 (页 606)

7.7.3.5 如何检查通道和连接




简介

本节说明如何在运行系统中检查“SIMATIC S5 PROFIBUS FDL”通道及其连接。

要求

- 安装 CP 5613 A3。
- 安装通信驱动程序。
- 组态 CP 5613 A3。
- 创建 STEP5 项目。
- 为“SIMATIC S5 PROFIBUS FDL”通道组态连接和变量。
- 激活 WinCC 项目。

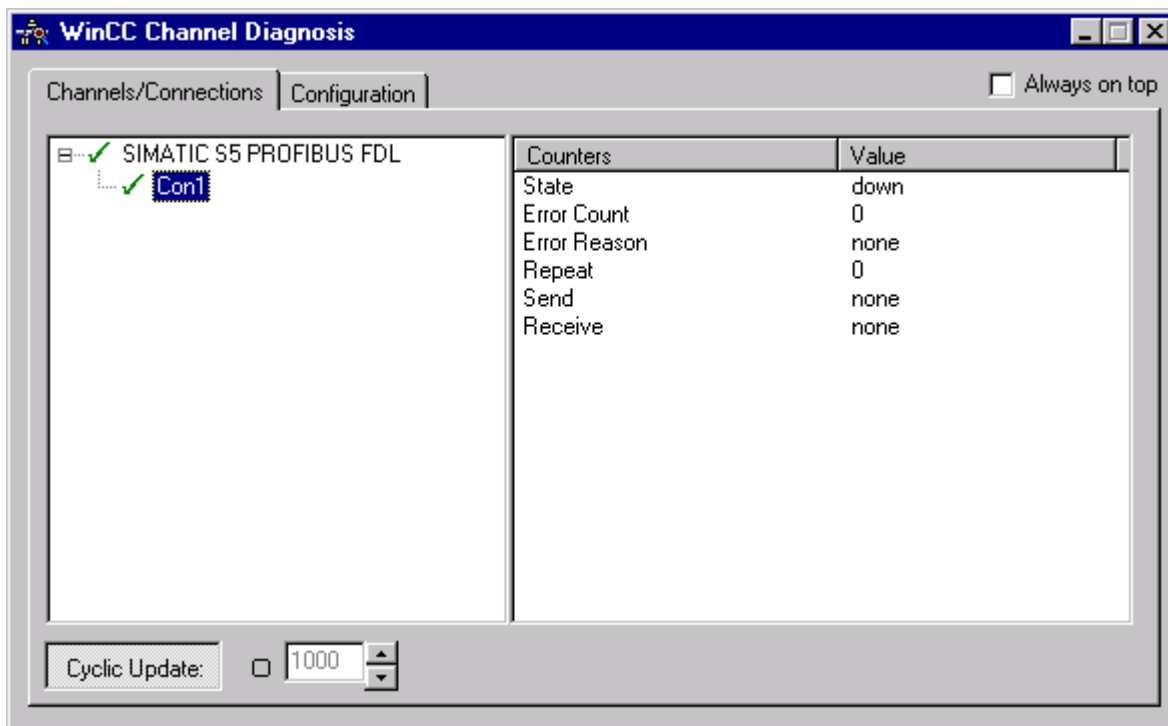
状态消息概述

图标	描述
	通道/连接无限制准备就绪
	通道/连接准备就绪，带有一些限制
	通道/没有关于连接状态的陈述
	通道/连接失败

7.7 “SIMATIC S5 Profibus FDL”通道的诊断

步骤

1. 从“开始”(Start) 菜单启动 WinCC 通道诊断。
2. 将打开“通道诊断”(Channel Diagnosis) 应用程序窗口。所有安装的通道及其连接的状态信息显示在“通道/连接”(Channels/Connections) 选项卡的左侧。



3. 检查名为“SIMATIC S5 PROFIBUS FDL”通道前的图标和连接。如果通道和连接状况正常，将有一个绿色的复选标记显示在相应条目的前面。有关各个图标含义的信息，请参见表格“状态消息概述”。
4. 如果通道名称和连接前面没有绿色复选标记，则选择左侧窗口中的连接。在右侧窗口中，检查计数器“状态”、“错误计数”、“错误原因”、“发送”和“接收”的条目。这些值表示检测到的错误。
5. 检查通道特定的日志文件。要执行此操作，请使用文本编辑器打开目录“Siemens\WinCC\Diagnose”中的文件。检查带“ERROR”标记的最新条目。有关该主题的更多信息，请参见“日志文件条目的描述”。
6. 如果检查日志文件之后仍然不能查明错误，请激活跟踪功能并联系客户支持部门。有关该主题的更多信息，请参见“组态通道的跟踪功能”。

参见

- 如何组态通道的跟踪功能 (页 574)
- 日志文件条目的描述 (页 601)
- 如何检查变量 (页 609)

7.7.3.6 如何检查变量

简介

如果运行系统中外部变量不具有预期的数值，请使用下列步骤来检查变量。

要求

- 安装 CP 5613 A3。
- 安装通信驱动程序。
- 组态 CP 5613 A3。
- 创建 STEP5 项目。
- 为“SIMATIC S5 PROFIBUS FDL”通道组态连接和变量。
- 激活 WinCC 项目。

步骤

1. 在 WinCC 项目管理器的变量管理器中，选择“SIMATIC S5 PROFIBUS FDL”通道。
2. 在数据窗口中，选择希望检查的外部变量。为此，打开目录结构直到此变量显示在表格区域中。
3. 移动鼠标指针，指向要检查的变量。将打开工具提示窗口，显示当前变量值、质量代码和最近一次数值改变的时间。
4. 检查质量代码。如果显示数值“80”，则该变量值正确。其它数值的描述可以参见“变量质量代码”。
5. 如果质量代码不等于“80”，从变量管理器中选择变量，然后单击快捷菜单中的“属性”(Properties)来打开“变量属性”(Tag Properties)对话框。
6. 检查是否已在“限制/报告”(Limits/Reportin)选项卡上组态了上限或下限、启动或替换值。这些数值会影响显示。
7. 如果变量值被其中一个已组态的数值改变，则取消激活该项目并改变该限制值或替换值。

说明

变量值、质量代码等只在运行期间显示。

参见

变量的质量代码 (页 622)

7.8 “OPC”通道的诊断

7.8.1 诊断“OPC”通道的可能性

下列可能性用于检测错误和诊断“OPC”通道或其变量之一：

检查连接和变量的组态

系统和连接参数的组态可能出现错误。无效变量值还可能由于错误引用 AS 中的变量编址而引起。

使用“通道诊断”诊断通道

“通道诊断”可以查询运行系统中通道和连接的状态。产生的任何错误都用“错误代码”显示。

诊断通道变量

在运行系统的变量管理器中，可以查询当前值、当前质量代码和变量改变的最后时刻。

参见

如何检查变量 (页 618)

如何检查通道和连接 (页 616)

如何检查组态数据 (页 615)

7.8.2 日志文件条目的描述

7.8.2.1 日志文件条目的描述

引言

在日志文件中记录通道的错误和状态的重要变化。下列章节只包含最重要的条目。这些条目可用于分析通讯问题。

下列两种类型的条目之间存在差别:

- INFO
- ERROR

条目的结构

日期/时间标记	标志名	描述
---------	-----	----

记录册中条目实例

```

2000-03-24 10:43:18,756 INFO Log starting ...
2000-03-24 10:43:18,756 INFO | LogFileName :C:\Siemens\WinCC\Diagnose\OPC.LOG
2000-03-24 10:43:18,756 INFO | LogFileCount :3
2000-03-24 10:43:18,756 INFO | LogFileSize :1400000
2000-03-24 10:43:18,756 INFO | TraceFlags :fa000007
000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC
2000-03-23 10:46:18,756 INFO Process detached at 2000-03-2410:46:18,746UTC
2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR
800706ba
2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN Unit
#1)]![OPC_No_Machine]: CoCreateInstance for server "OPCServer.WinCC" on machine
OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba -
RPC_S_SERVER_UNAVAILABLE (Der RPC-Server ist nicht verfügbar.))")

```

参见

"ERROR"标记条目 (页 613)

"INFO"标记条目 (页 612)

7.8 “OPC”通道的诊断

7.8.2.2 “INFO”标记条目

引言

文件中的每个条目都有日期和时间标志，后面带有标记名称和描述。

日期/时间标志	标志名	描述
---------	-----	----

记录册中条目实例

```

2000-03-24 10:43:18,756 INFO Log starting ...
2000-03-24 10:43:18,756 INFO | LogFileName :C:\Siemens\WinCC\Diagnose\OPC.LOG
2000-03-24 10:43:18,756 INFO | LogFileCount :3
2000-03-24 10:43:18,756 INFO | LogFileSize :1400000
2000-03-24 10:43:18,756 INFO | TraceFlags :fa000007
000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC
2000-03-23 10:46:18,756 INFO Process detached at 2000-03-2410:46:18,746UTC
    
```

最重要记录册条目的描述

消息文本	描述
Log starting ...	起始消息
LogFileName : C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	带路径日志文件的名称
LogFileCount : "n"	通道的日志文件的数目
LogFileSize : "x"	单个日志文件的大小（以字节为单位）
TraceFlags :fa000007	用十六进制数字显示跟踪功能使用的标记
Process attached at 2000-03-24 09:43:18,746 UTC	通道由 WinCC 数据管理器装载。

消息文本	描述
Process detached at 2000-03-24 10:46:18,746 UTC	通道由 WinCC 数据管理器卸载。
IOPCChnShutdown::ShutdownRequest was called... Reason:system going down" IOPCChnShutdown::ShutdownRequest	取消激活 WinCC OPC 服务器的 WinCC 项目。请求 WinCC OPC 客户机与 WinCC OPC 服务器断开。

7.8.2.3 “ERROR”标记条目

简介

文件中的每个条目都有日期和时间标志，后面带有标记名称和描述。如果是“Error”标记，描述包含消息文本、错误代码和错误消息的文本。一些错误代码没有出错消息的文本。

日期/时间标记	标志名	描述 消息文本 + 错误代码 + 出错消息文本
---------	-----	----------------------------

记录册中条目实例

2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR 800706ba

2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN Unit #1)]![OPC_No_Machine]: CoCreateInstance for server \"OPCServer.WinCC\" on machine OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba - RPC_S_SERVER_UNAVAILABLE (RPC server not available.))")

最重要记录册条目的描述

错误代码	错误消息文本	可能的原因
c0040004	服务器不支持“canonicalDatatype”和“requestedDatatype”之间的转换。	访问 OPC 服务器上的 WinCC 变量失败。可以转换但是失败了。 WinCC 变量不在服务器上，或者组态的数据类型不匹配。
c0040007	该名称在服务器的名称空间中不存在。	如果使用服务器的名称空间中不存在的变量名称访问 OPC 客户端，服务器将始终返回错误代码。 示例：浏览、读取变量、写入变量、将变量插入订阅。
00000001	添加条目	访问 OPC 服务器上的 WinCC 变量失败。WinCC 变量不在服务器上，或者组态的数据类型不匹配。 数据类型 WinCC 变量 OPC 服务器 = 数据类型 WinCC 变量 OPC 客户机。
80004005	无法解析服务器名称	用作 WinCC OPC 服务器的计算机在网络上不可用。 通过“OPC”通道访问的 WinCC OPC 服务器不可用。
80040154	未注册等级	WinCC OPC 服务器未在系统中正确注册。 WinCC OPC 服务器的 WinCC 项目未激活。
80070057	参数错误	WinCC 变量不在 OPC 服务器上，或者组态的数据类型不匹配。
800706ba	RPC 服务器不可用。	在网络上找不到要在其上启动 OPC 服务器的计算机。

7.8.3 确定错误变量值的原因

7.8.3.1 如何确定无效变量的原因

如果在运行系统中产生意外的变量值，进行如下处理以确定原因：

1. 检查组态数据
2. 检查连接
3. 检查通道变量

参见

如何检查变量 (页 618)

如何检查通道和连接 (页 616)

如何检查组态数据 (页 615)

7.8.3.2 如何检查组态数据

要求

- 一台计算机用作运行 WinCC 项目的 WinCC OPC 客户端
- “OPC”通道必须集成在 OPC 客户端的 WinCC 项目中。
- 在 OPC 服务器的 WinCC 项目中组态 WinCC 变量。
- 在 OPC 客户端上组态一个连接和一个 WinCC 变量，与创建的服务器变量通信。
- 启用 OPC 服务器和 OPC 客户端上的运行系统。

操作步骤

1. 在 OPC 客户端的 WinCC 项目管理器浏览窗口中，单击“OPC”图标前的加号。
单击“OPC 组 (OPCHN 单元#1)”(OPC Groups (OPCHN Unit#1)) 图标前的加号。
2. 在待测试连接的快捷菜单中，选择条目“特性”(Properties)。
将打开“连接属性”对话框。
3. 在“OPC 连接”(OPC Connection) 选项卡中，检查“OPC 服务器名称”(OPC Server Name) 字段中 OPC 服务器的 ProgID。
 - 当连接到使用 WinCC V5.0 或更高版本的服务器时，必须输入“OPCServer.WinCC”。
 - 当连接到使用 WinCC V4.x 或更高版本的服务器时，必须输入“OE.Groups”。
4. 检查是否在“在另一计算机上运行服务器”(Run the server on another computer) 字段中输入了要用作 OPC DA 服务器的计算机名称。
5. 要测试与 OPC 服务器的连接，请单击“测试服务器”(Test Server)。
关闭对话框。
6. 单击连接图标前的加号。
7. 在待测试变量的快捷菜单中，选择条目“属性”(Properties)。
将打开“变量属性”对话框。
 - 必须为该变量输入与 OPC 服务器上变量相同的“数据类型”。
8. 在“变量属性”对话框中，单击“选择”按钮。
将打开“地址属性”对话框。

7.8 “OPC”通道的诊断

9. 检查“条目名称”和“数据类型”域中的条目。
 - “项目名称”必须与 OPC 服务器上的变量名称相匹配。
 - “数据类型”必须与 OPC 服务器上变量的数据类型相匹配。
10. 检查通道特定的日志文件。
 要执行此操作，请使用文本编辑器打开目录“Siemens\WinCC\Diagnose”中的文件。
 检查带“ERROR”标记的最新条目。
 有关此主题的详细信息，请参见“日志文件条目的描述 (页 610)”。
11. 如果检查日志文件之后仍然不能查明错误，请激活跟踪功能并联系客户支持部门。
 更多相关信息，请参见“如何组态通道的跟踪功能 (页 574)”。

参见

- 如何组态通道的跟踪功能 (页 574)
- 日志文件条目的描述 (页 610)
- 如何检查通道和连接 (页 616)

7.8.3.3 如何检查通道和连接



简介



本节说明如何在运行系统中检查“OPC”通道及其连接。

要求

- 一台计算机用作运行 WinCC 项目的 WinCC OPC 客户端
- “OPC”通道必须集成在 OPC 客户端的 WinCC 项目中。
- 在 OPC 服务器的 WinCC 项目中组态 WinCC 变量。
- 在 OPC 客户端上组态一个连接和一个 WinCC 变量，与创建的服务器变量通信。
- 启用 OPC 服务器和 OPC 客户端上的 WinCC 项目。

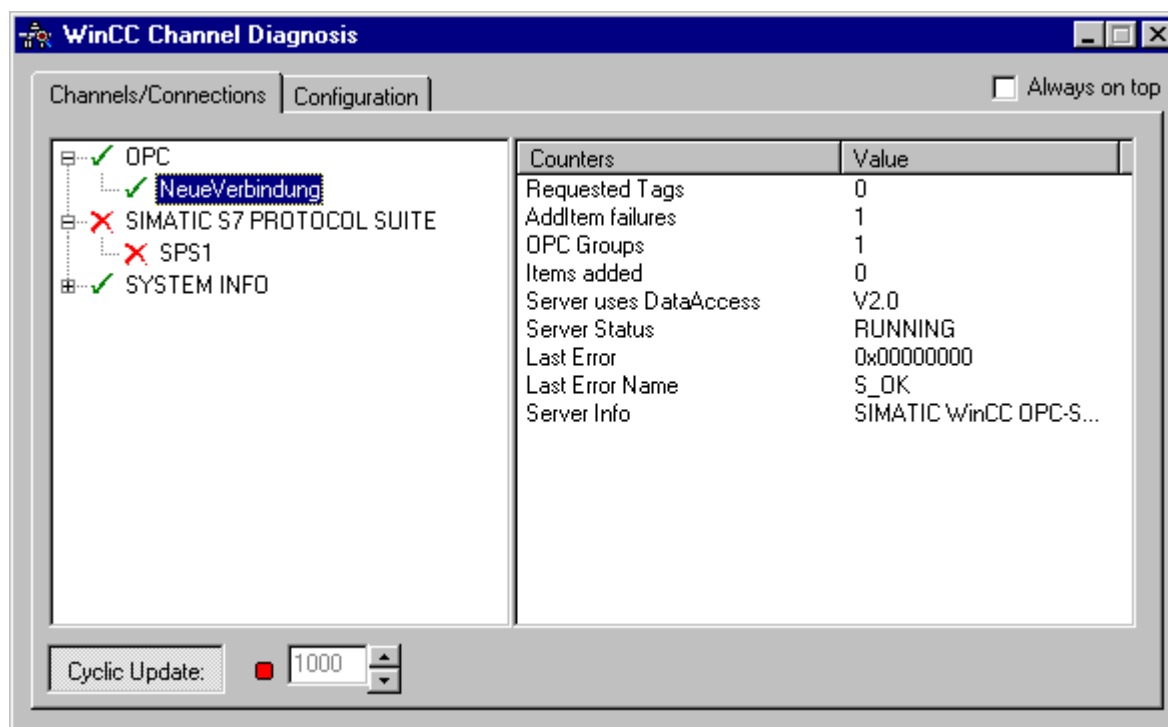
状态消息概述

图标	描述
	通道/连接无限制准备就绪
	通道/连接准备就绪，带有一些限制

图标	描述
	通道/没有关于连接状态的陈述
	通道/连接失败

步骤

1. 从“开始”(Start) 菜单启动 WinCC 通道诊断。
将打开“通道诊断”(Channel Diagnosis) 应用程序窗口。
所有安装的通道及其连接的状态信息显示在“通道/连接”(Channels/Connections) 选项卡的左侧。



2. 检查 OPC 连接前的图标。
如果连接状态良好，将有一个绿色的复选标记显示在相应条目的前面。
有关各个图标含义的信息，请参见表格“状态消息概述”。
3. 如果通道名称和连接前面没有绿色复选标记，则选择左侧窗口中的连接。
4. 在右侧窗口中，检查计数器“添加项目失败”、“服务器状态”、“最后错误”和“最后错误名称”的条目。
这些值表示检测到的错误。
5. 检查通道特定的日志文件。
要执行此操作，请使用文本编辑器打开目录“Siemens\WinCC\Diagnose”中的文件。
检查带“ERROR”标记的最新条目。
有关此主题的详细信息，请参见“日志文件条目的描述(页 610)”。
6. 如果检查日志文件之后仍然不能查明错误，请激活跟踪功能并联系客户支持部门。
更多相关信息，请参见“如何组态通道的跟踪功能(页 574)”。

7.8 “OPC”通道的诊断

参见

如何组态通道的跟踪功能 (页 574)

日志文件条目的描述 (页 610)

如何检查变量 (页 618)

7.8.3.4 如何检查变量

简介

如果运行系统中外部变量不具有预期的数值，请使用下列步骤来检查变量。

要求

- 一台计算机用作运行 WinCC 项目的 WinCC OPC 客户机。
- “OPC”通道必须集成在 OPC 客户机的 WinCC 项目中。
- 在 OPC 服务器的 WinCC 项目中组态 WinCC 变量。
- 在 OPC 客户机上组态一个连接和一个 WinCC 变量，与创建的服务器变量通信。
- 激活 OPC 服务器和客户机上的 WinCC 项目。

步骤

1. 在 WinCC 项目管理器的变量管理器中，选择“OPC”通道。
2. 在数据窗口中，选择希望检查的外部变量。为此，打开目录结构直到此变量显示在表格区域中。
3. 移动鼠标指针，指向要检查的变量。将打开工具提示窗口，显示当前变量值、质量值和最近一次修改的时间。
4. 检查质量值。如果显示值“C0”，则变量值正确。其它数值的描述可以参见“变量质量代码”。
5. 如果质量代码不等于“C0”，从变量管理器中选择变量，然后单击快捷菜单中的“属性”来打开“变量属性”对话框。
6. 检查是否已在“限制/报告”(Limits/Reportin) 选项卡上组态了上限或下限、启动或替换值。这些数值会影响显示。
7. 如果变量值被其中一个已组态的数值改变，则取消激活该项目并改变该限制值或替换值。

说明

变量值、质量代码等只在运行期间显示。

参见

变量的质量代码 (页 622)

7.9 变量质量

7.9.1 变量质量

引言

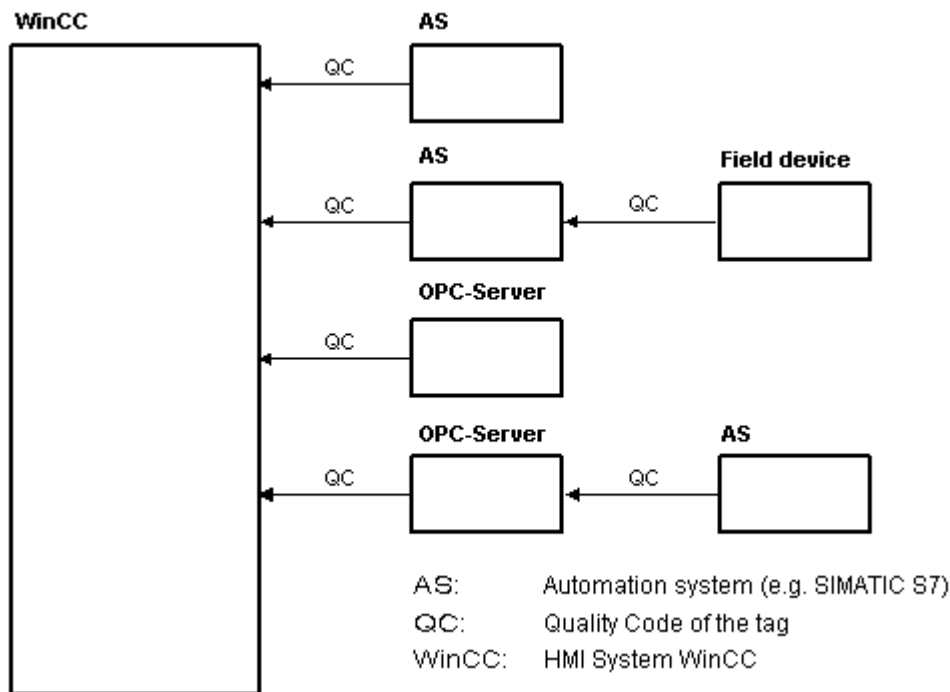
在 WinCC 中，有两个质量指标用来评估变量质量。这两个指标为变量状态和质量代码。

变量状态在 WinCC 中形成，用于指示 OS 中组态设置的质量。另外，变量状态还指示与 WinCC 通讯伙伴的连接状态。通讯伙伴可以是自动化系统或服务器计算机。

质量代码包含与变量状态相同的信息。除此以外，质量状态还包含有关连接伙伴的质量语句，用于评估或处理变量。可能的伙伴为：

- 自动化系统
- 带现场设备的自动化系统
- OPC 服务器
- 带下位自动化系统的 OPC 服务器

其中质量代码在处理链内转发。如果在处理链的某处存在一个变量的多个质量代码有待处理，那么将转发最差的代码。



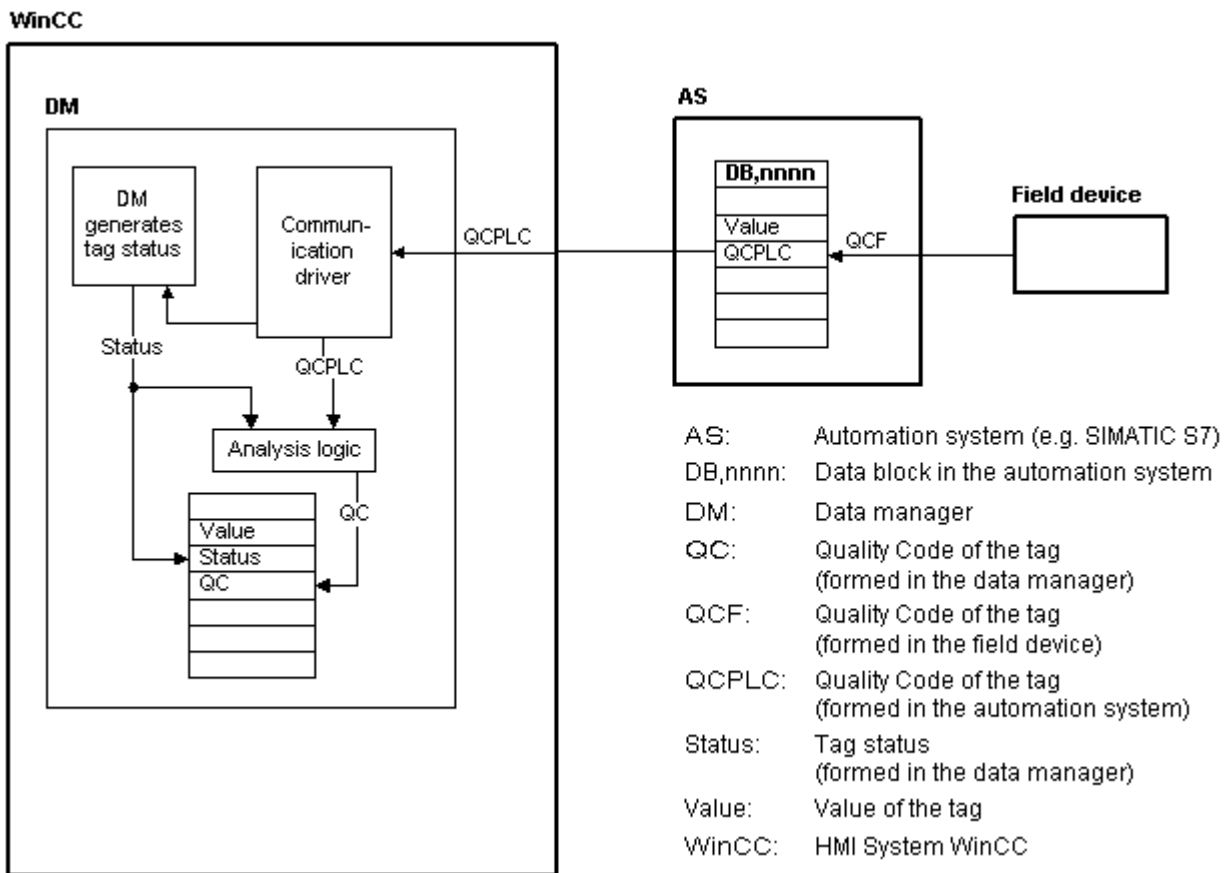
质量代码提示变量质量，与在何处形成该代码无关。

层叠的质量代码

通过使用连接了现场设备的自动化系统实例，可以概述质量代码层叠。

自动化系统读取由现场设备生成的质量代码。通过一个分析逻辑，根据优先级对等待同一个变量同时处理的质量代码进行评估。并将表示最差状态的质量代码分配给变量。必须在数据块中，将该质量代码紧跟在相关的变量值之后保存。

可通过 PCS7 库的通道模块来启动分析逻辑。如果不能使用 PCS7 库，那么必须在自动化系统中手动组态分析逻辑。



通过其中一个通讯驱动程序，WinCC 在运行系统中从自动化设备读取变量，包括相关的质量代码在内。对于每个变量，将在数据管理器中生成变量状态。可以包括如超出组态的测量范围限制以及 WinCC 和自动化设备之间的链接状态等。

通过数据管理器中的分析逻辑，从数据管理器的变量状态以及自动化设备的质量代码中生成质量代码。此处也同样传送表示最差状态的代码，并由 WinCC 将其保存为质量代码。对于在自动化系统中没有质量代码的变量，质量代码始终与变量状态相同。

7.9.2 变量的质量代码

简介

检查变量的状态和质量时需要质量代码。所显示的质量代码概述了整个数值传送和各个变量的数值处理的质量。于是例如使用质量代码可以查看当前值是起始值还是替换值。

质量代码具有优先次序。如果同时产生多个代码，那么将显示具有最差状态的代码。

质量代码的计算

可通过各种不同方法计算质量代码：

- 使用 VB 脚本计算
- 使用 C 脚本计算
- 通过动态对话框计算
- I/O 字段的“质量代码更改变量”结果的计算

说明

为了将整个数值传送和处理包含在过程变量的质量代码中，已连接的自动化系统必须支持质量代码。在 AS 中组态变量时，确保有足够的存储空间用于质量代码。例如，在 S7 系列的 AS 中，质量代码需要一个额外的字节附加到过程值上。为了防止错误，在组态变量时必须将该字节考虑进去，例如在数据块的末尾。

在变量管理中显示质量代码

可以在“变量管理”(Tag Management) 中查看变量的质量代码。

要求：

- WinCC 项目已激活。
- “变量管理”(Tag Management) 数据区域中显示了“质量代码”(Quality Code) 列。



在过程画面中显示质量代码

对于具有过程连接的图形对象中的变量值显示，质量代码可能会影响该显示。如果质量代码的值为 0x80（优）或 0x4C（初始值），变量值不会显示为灰色。所有其它值都会显示为灰色。此外，根据所设置的 WinCC 设计，对于以下对象将显示一个黄色警告三角标志：

- I/O 字段
- 棒图，3D 棒图
- 复选框，单选框
- 组显示，状态显示
- 滚动条对象

结构

质量代码具有如下的二进制结构：

QQSSSLL

Q: 质量

S: 质量的子状态

L: 限制。该数值是可选的。

说明

显示在“质量”表格中的质量代码是质量级的基本数值。使用子状态和限制元素导致中间数值超过相关的质量级。

质量

前两位指定变量的质量。

	Q	Q	S	S	S	S	L	L
	2	2	2	2	2	2	2	2
	7	6	5	4	3	2	1	0
劣 - 数值没有用。	0	0	-	-	-	-	-	-
不确定 - 数值的质量低于普通情况，但是数值可能仍然有用。	0	1	-	-	-	-	-	-
优（非层叠）- 数值的质量是优的。可能的报警条件会通过子状态指示。	1	0	-	-	-	-	-	-
优（层叠）- 数值可能用在控件中。	1	1	-	-	-	-	-	-

子状态

仅仅使用质量是不够的。单个的质量分为子状态。质量代码是二进制编码的。为了分析质量代码，这些数值必须转换为十六进制表达式。

变量的质量代码

在下表中列出了可能的质量代码。列表顶部列出了最差的质量代码，而列表底部列出了最好的质量代码。为最好的质量代码分配最低优先级，而为最差的质量代码分配最高优先级。如果此过程中的一个变量发生多种状态，则将传送最差的代码。

代码 (十六进制)	质量		Q	Q	S	S	S	S	L	L
0x23	劣	设备钝化 - 禁止诊断报警	0	0	1	0	0	0	1	1
0x3F	劣	功能检查 - 本地优先	0	0	1	1	1	1	1	1
0x1C	劣	不能用 - 该数值不可靠，因为该块还未判断，并且可能由组态设计者构造。如果该块模式是 O/S，则进行置位。	0	0	0	1	1	1	-	-
0x73	不确定	模拟值 - 开始	0	1	1	1	0	0	1	1
0x74	不确定	模拟值 - 结束	0	1	1	1	0	1	-	-
0x84	优 (非层叠)	激活更新事件 - 如果数值是优的并且块具有激活的更新事件，则进行置位。	1	0	0	0	0	1	-	-
0x24	劣	维护报警 - 提供更多诊断。	0	0	1	0	0	1	-	-

代码 (十六进制)	质量		Q	Q	S	S	S	S	L	L
0x18	劣	无通信, 没有可用的数值 - 如果自该数值上次不能使用以来从未进行过通信, 则进行置位。	0	0	0	1	1	0	-	-
0x14	劣	无通信, 具有最后有用的数值 - 如果该数值已经由现在发生故障的通信置位, 则进行置位。	0	0	0	1	0	1	-	-
0x0C	劣	设备故障 - 如果数值源受设备故障的影响, 则进行置位。	0	0	0	0	1	1	-	-
0x10	劣	传感器故障	0	0	0	1	0	0	-	-
0x08	劣	未连接 - 如果该输入需要连接而没有连接, 则进行置位。	0	0	0	0	1	0	-	-
0x04	劣	组态错误 - 根据特定制造商所检测到的, 如果该数值是因为存在有关参数化或组态的一些不一致而无用, 则进行置位。	0	0	0	0	0	1	-	-
0x00	劣	未指定 - 没有指出数值劣的原因。用于传播。	0	0	0	0	0	0	-	-
0x28	劣	与过程有关 - 替换值	0	0	1	0	1	0	-	-
0x2B	劣	与过程有关 - 无维护	0	0	1	0	1	0	1	1
0x68	不确定	需要维护	0	1	1	0	1	0	-	-
0x60	不确定	模拟数值 - 当块处于手动模式时如果由操作员写入过程值, 则进行置位。	0	1	1	0	0	0	-	-
0x64	不确定	传感器校准	0	1	1	0	0	1	-	-
0x5C	不确定	组态错误	0	1	0	1	1	1	-	-
0x58	不确定	低于正常	0	1	0	1	1	0	-	-
0x54	不确定	工程单位范围超出 - 如果数值位于该参数定义的数值设置范围之外, 则进行置位。限制定义已经超出了哪一个方向。	0	1	0	1	0	1	-	-
0x50	不确定	传感器转换不精确	0	1	0	1	0	0	-	-
0x4B	不确定	替换 (常数)	0	1	0	0	1	0	1	1
0x78	不确定	与过程有关 - 无维护	0	1	1	1	1	0	-	-
0x4C	不确定	初始值 - 复位设备或参数期间和之后的易失参数的数值。	0	1	0	0	1	1	-	-
0x48	不确定	替换值 - 预定义数值用于替换已计算的数值。这用于故障安全处理。	0	1	0	0	1	0	-	-
0x44	不确定	最后有用的数值 - 无论写入什么, 该数值都已停止操作。这用于故障安全处理。	0	1	0	0	0	1	-	-
0x40	不确定	未指定 - 没有指出数值不确定的原因。	0	1	0	0	0	0	-	-

7.9 变量质量

代码 (十六进制)	质量		Q	Q	S	S	S	S	L	L
0xE0	优 (层叠)	启动故障安全 (IFS) - 来自需要其后续程序输出块 (例如 AO) 进入故障安全的程序块的值。	1	1	1	0	0	0	-	-
0xD8	优 (层叠)	本地替换值 (LO) - 数值来自由本地键切换锁定的块或来自带有互锁逻辑激活的综合 AO/DO。正常控制故障必须传播到主机系统报警和显示目的的功能运行。这同样意味着不引入。	1	1	0	1	1	0	-	-
0xD4	优 (层叠)	不选择 (DNS) - 数值来自不应选择的块, 由于条件位于块内或超过块。	1	1	0	1	0	1	-	-
0xCC	优 (层叠)	不引入 (NI) - 数值来自不具有使用该输入的目标模式的块。	1	1	0	0	1	1	-	-
0xC8	优 (层叠)	初始化请求 (IR) - 数值是源 (返回计算输入参数) 的初始化数值, 因为低级循环中断或者模式错误。	1	1	0	0	1	0	-	-
0xC4	优 (层叠)	初始化确认 (IA) - 数值是来自源 (层叠输入、参数的远程层叠和参数的远程输出) 的初始化数值。	1	1	0	0	0	1	-	-
0xC0	优 (层叠)	好 - 没有错误或特殊情况与该数值相关。	1	1	0	0	0	0	-	-
0xA0	优 (非层叠)	启动故障安全	1	0	1	0	0	0	-	-
0x98	优 (非层叠)	不确定的紧急报警 - 如果数值是优的并且块具有优先级大于或等于 8 的不确定报警, 则进行置位。	1	0	0	1	1	0	-	-
0x94	优 (非层叠)	不确定的报告报警 - 如果数值是优的并且块具有优先级小于 8 的不确定报警, 则进行置位。	1	0	0	1	0	1	-	-
0x90	优 (非层叠)	不确定的更新事件 - 如果数值是优的并且块具有不确定的更新事件, 则进行置位。	1	0	0	1	0	0	-	-
0x8C	优 (非层叠)	激活紧急报警 - 如果数值是优的并且块具有优先级大于或等于 8 的激活报警, 则进行置位。	1	0	0	0	1	1	-	-
0x88	优 (非层叠)	激活报告报警 - 如果数值是优的并且块具有优先级小于 8 的激活报警, 则进行置位。	1	0	0	0	1	0	-	-
0xA8	优 (非层叠)	需要维护	1	0	1	0	1	0	-	-
0xA4	优 (非层叠)	要求维护	1	0	1	0	0	1	-	-

代码 (十六进制)	质量		Q	Q	S	S	S	S	L	L
0xBC	优 (非层叠)	功能检查 - 本地优先	1	0	1	1	1	1	-	-
0x80	优 (非层叠)	好 - 没有错误或特殊情况与该数值相关。	1	0	0	0	0	0	-	-

限制

质量代码可以由限制进一步细分。限制是可选的。

	Q	Q	S	S	S	S	L	L
好 - 数值可以随意移动。	-	-	-	-	-	-	0	0
下限 - 数值已经接受其下限。	-	-	-	-	-	-	0	1
上限 - 数值已经接受其上限。	-	-	-	-	-	-	1	0
常数 (上限和下限) - 数值不能变动, 不管过程如何操作。	-	-	-	-	-	-	1	1

利用 OPC 通信的质量代码

在通过“OPC”通道进行的通信中, 将转换 OPC 不支持的质量代码。

WinCC 中的质量代码	OPC 中的质量代码
0x48	0x40
0x4C	0x40
0x5C	0x40
0x60	0x40
0x80...0xD4	0xC0
0xD8	0xC0

7.9.3 变量状态

简介

在运行系统中，可以监视各个 WinCC 变量的状态。变量状态包含已组态的测量范围过限信息以及 WinCC 和自动化设备之间的链接状态。

质量代码提示变量质量，与在何处形成该代码无关。因此，将整个数值传送和数值处理的状态考虑在内。

例如，如果低于测量范围的下限，那么质量代码通信为“0x55”。在 WinCC 数据管理器或现场设备中可能发生该类超出测量范围的情况。变量状态使您能查找到是否在 WinCC 中出现了超出测量范围或在将数值传送到 WinCC 之前就已经出现了超出测量范围的情况。

例如，变量状态以代码 0x0010 报告超出限制，它表明在 WinCC 中组态了低于下限的值。如果变量状态没有报告过限信息，那么传送到 WinCC 中的质量代码已经包含过限信息。

质量代码的计算

可通过各种不同方法计算质量代码：

- 使用 C 脚本计算
- 通过动态对话框计算
- I/O 字段的“质量代码更改变量”结果的计算

WinCC 状态标记

下表给出了可能的变量状态。

标记名称	数值	描述
	0x00 00	无错
DM_VARSTATE_NOT_ESTABLISHED	0x00 01	未建立到伙伴的连接
DM_VARSTATE_HANDSHAKE_ERROR	0x00 02	信号交换错误
DM_VARSTATE_HARDWARE_ERROR	0x00 04	网络模板故障
DM_VARSTATE_MAX_LIMIT	0x00 08	超过所组态的上限

标记名称	数值	描述
DM_VARSTATE_MIN_LIMIT	0x00 10	超过所组态的下限
DM_VARSTATE_MAX_RANGE	0x00 20	超过格式处理上限
DM_VARSTATE_MIN_RANGE	0x00 40	超出格式处理下限
DM_VARSTATE_CONVERSION_ERROR	0x00 80	显示转换出错（与超过格式限制 xxx 有关）
DM_VARSTATE_STARTUP_VALUE	0x01 00	变量初始化值
DM_VARSTATE_DEFAULT_VALUE	0x02 00	变量的替换值
DM_VARSTATE_ADDRESS_ERROR	0x04 00	通道寻址出错
DM_VARSTATE_INVALID_KEY	0x08 00	没有找到变量/不可用
DM_VARSTATE_ACCESS_FAULT	0x10 00	不允许访问变量
DM_VARSTATE_TIMEOUT	0x20 00	超时/没有来自通道的回查消息
DM_VARSTATE_SERVERDOWN	0x40 00	服务器不可用。

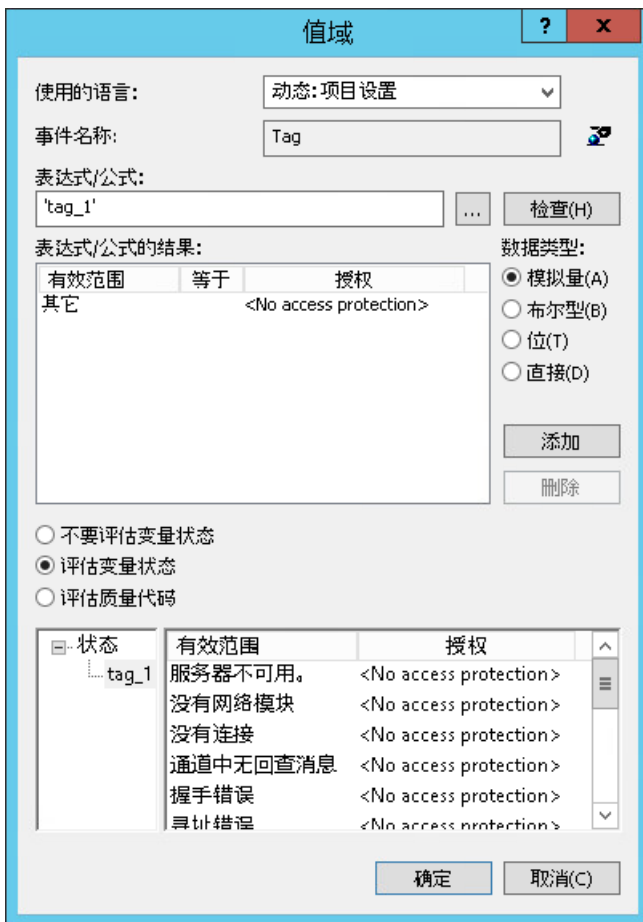
7.9.4 使用变量状态监视连接状态

在运行系统中可以监视各个 WinCC 变量的状态，从而提供关于相应连接状态的信息。

组态状态监视

该监视作为对象属性在图形编辑器中组态。

1. 在“对象属性”(Object Properties) 窗口中选择所需属性。
2. 从“动态”(Dynamic) 列的快捷菜单中选择“动态对话框”(Dynamic dialog) 条目。
“数值范围”(Value range) 对话框将打开。



3. 指定设置：
 - 监视的变量。
 - 变量值：分配有效范围和状态显示
 - 变量的状态评估
 - 状态：分配有效范围和相应的状态文本

在运行系统中，输入的状态文本之一（对应于当前变量状态）将显示在组态的对象中。

参见

在运行系统中为连接状态组态变量 (页 155)

7.9.5 使用全局动作监视变量状态

引言

监视变量状态的途径之一是使用全局脚本编辑器中的内部函数“GetTagState”和“GetTagStateWait”。与函数“GetTag”和“GetTagWait”不同，除了返回变量的数值外，它们还返回变量的状态。可以核定该状态值，并且由它触发各种事件。它也可以用于判断相关连接的状态。

在全局动作中，要监视变量的状态值通过用于该数据类型的“GetTagState”函数来确定。对于每种变量类型都存在这样一个函数。状态值“0”指示连接完好，没有错误。这种状态现在可以按期望判断。

实例：

该实例说明监视“有符号 16 位数”类型的 WinCC 变量。为了确定该变量的状态，使用函数“GetTagSWordState”。第一个函数参数是要监视的 WinCC 变量的名称。第二个参数给出返回的状态值要写入的地方。

```
#include "apdefap.h"
int gscAction( void )
{
    DWORD dwState = 0;
    GetTagSWordState("Variable_01",&dwState);
    if ( dwState == 0 )
    {
        //Connection OK
        SetTagBit("BINi_E_CONNECTION",FALSE);
    }
    else
    {
        //Connection Error
        SetTagBit("BINi_E_CONNECTION",TRUE);
    }
    return 0;
}
```

在内部变量 BINi_E_CONNECTION 中输出变量状态。在发生错误的情况下，该变量值被设置为 TRUE。例如，在错误操作时，该变量可以用于触发报警或显示出错消息。

7.9.6 如何检查内部变量

简介

如果运行系统中内部变量不具有预期的数值，使用下列步骤来检查变量。

要求

- 已组态内部变量。
- WinCC 项目已激活。

步骤

1. 在 WinCC 项目管理器中打开“变量管理”(Tag Management)。
2. 在导航区域中选择条目“内部变量”以及要检查的变量。
3. 要在数据区域中显示“质量代码”(Quality Code) 和“值”(Value) 列，可能需要转到“显示”(Show)，然后在列标题的快捷菜单中选择这些列。
4. 检查质量代码。如果显示值“80”，则该变量值正确。其它数值的描述可以参见“变量质量代码”。
5. 如果质量代码不等于“80”，请检查右侧属性下的设置。
6. 检查是否已组态上限值和下限值或起始值。这些数值可以影响显示。
7. 如果变量值被其中一个已组态的数值改变，则取消激活该项目并改变该限制值或替换值。



说明

运行系统中只显示变量值和质量代码。

参见

变量的质量代码 (页 622)

OPC - 开放式互连

8.1 OPC - 开放式互连

内容

来自不同生产商的设备和应用程序可以通过 OPC 标准软件接口以一致的方式连接起来。WinCC 可以用作 OPC 服务器或 OPC 客户端。“OPC”通道表示 WinCC 的 OPC 客户端应用程序。该部分说明了：

- WinCC 有哪些 OPC 服务器。
- 如何在 WinCC 中使用 OPC。
- 如何建立不同的 OPC DA 链接。
- 如何对 WinCC 消息系统的访问进行组态。
- WinCC 消息系统如何在 OPC A&E 上映射。
- 如何建立对 WinCC 归档系统的访问。

8.2 OPC 的功能

OPC 是一个独立于制造商的标准化软件接口，在自动化工程领域中用于数据交换。

通过 OPC 接口可在不同制造商的设备和应用程序之间实现标准连接。

OPC 基于 Windows 的 COM（组件对象模型）和 DCOM（分布式组件对象模型）技术。

OPC XML DA 额外提供了一个基于 XML、SOAP 和 HTTP Internet 标准的软件接口。

OPC UA（统一架构）是 OPC 的替代技术。OPC UA 与平台无关并且支持不同的协议作为通信媒介。

8.3 OPC 规范及兼容性

概述

访问 WinCC 中的以下对象时，OPC 指定了相应的接口：

- 过程值 (OPC Data Access 2.05a, 3.0; OPC XML Data Access 1.01; OPC UA 1.03)
- 归档的过程值 (OPC Historical Data Access 1.20; OPC UA Historical Access 1.03)
- 按时间排列的消息 (OPC Historical Alarms and Events 1.10)
- 消息 (OPC Alarms and Events 1.10; OPC UA Alarms and Conditions 1.03)

有关各个 OPC 规范的详细信息，请参见 OPC 基金会 (<http://www.opcfoundation.org>) 网站。

兼容性

这些规范的支持通常由 OPC Foundation 的“Compliance Test Tool”(CTT) 监视。加入“OPC Interoperability Workshops”可保证与其他制造商 OPC 产品的互用性。

提交的测试结果会在 OPC Foundation 网站上发布。要查看这些结果，请输入搜索项“OPC Self-Certified Products”。

8.4 在 WinCC 中使用 OPC

简介

在 WinCC 中，服务器可供以下 OPC 接口使用：

- OPC Data Access / OPC XML Data Access：访问 WinCC 数据主体
- OPC Historical Data Access：访问 WinCC 归档系统
- OPC Alarms&Events：访问 WinCC 消息系统
- OPC Unified Architecture：访问 WinCC 数据主体和归档系统

默认情况下，WinCC 包含一个 OPC 通道。该 OPC 通道可作为客户端通过 OPC DA、OPC XML DA 或 OPC UA 访问相关 OPC 服务器。

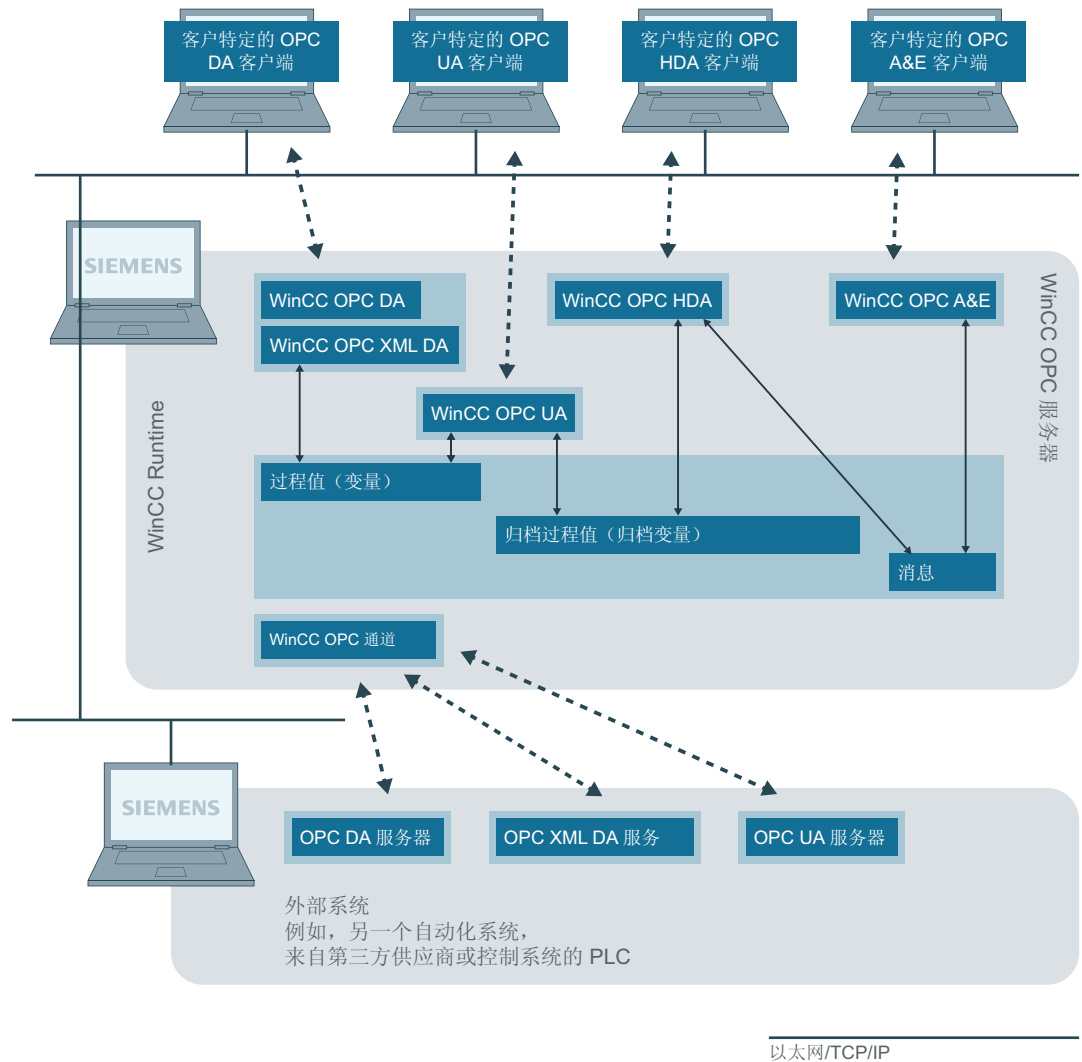
WinCC OPC 通信概念

通过 DCOM 完成 WinCC OPC 服务器和 OPC 客户端之间的数据交换。安装 WinCC 后，将正确组态 WinCC OPC 服务器的 DCOM 设置。

如果 WinCC OPC 服务器或客户端与外部 OPC 系统进行通信，则必须执行相应调整。必须在客户端上用户管理的“DCOM/工作区/COM 安全/访问权限/编辑默认值”中为用户输入“本地访问”和“远程访问”权限。

WinCC 的 OPC XML 服务器实现 Web 服务。这样便可以通过 Internet 访问 PC。因此需要定义适当的访问权限。

下图显示了 WinCC OPC 通信概念：



许可证

OPC 服务器	许可证
WinCC OPC DA 服务器	有效的 WinCC RT 许可证
WinCC OPC XML DA 服务器	有效的 WinCC RT 许可证
WinCC OPC UA 服务器	WinCC 选件连通性软件包
WinCC OPC HDA 服务器	
WinCC OPC A&E 服务器	

8.5 如何组态 Windows 以使用 WinCC OPC

简介

OPC 客户机和 OPC 服务器是 DCOM 应用程序。分布式 DCOM 应用程序只能使用同一用户帐户运行。因此，OPC 服务器必须识别 OPC 客户机的用户帐户，反之亦然。如果 WinCC OPC 服务器和 WinCC OPC 客户机一起使用，则在安装时已保证组态正确。

如果使用外部 OPC 服务器或客户机，请声明用户帐户。

有关授予用户权限的更多信息，请参阅 Windows 文档。

要求

作为管理员登录到 WinCC OPC 服务器和 OPC 客户机工作站来组态用户权限。

步骤

1. 转到“控制面板 > 系统和安全 > 管理工具 > 计算机管理 > 本地用户和组”(Control Panel > System and Security > Administrative Tools > Computer Management > Local Users and Groups)。
2. 在“用户”(Users) 快捷菜单中，选择“新用户”(New User)。在“新用户”(New User) 对话框中，输入通信伙伴的用户帐户详细信息。单击“创建”，关闭对话框。
3. 单击“用户”图标。双击相关的用户。将显示该用户的“属性”对话框。
4. 单击“成员”(Member Of) 选项卡。单击“添加”(Add)。将打开“选择组”对话框。
5. 添加组“Users”。如果是安装了 WinCC 的计算机，还要添加组“SIMATIC HMI”。单击“确定”关闭所有打开的对话框。

如何调整 Windows 防火墙设置

安装 WinCC 后，将正确组态 WinCC OPC 服务器的 Windows 防火墙设置。

如果 OPC 客户端访问不同子网中的 OPC 服务器，则必须调整 OPC 服务器所允许网络区域的组态。

8.6 WinCC OPC XML DA 服务器

8.6.1 WinCC OPC XML DA 服务器的功能

简介

WinCC 的 OPC XML DA 服务器是 Microsoft Internet 信息服务 (IIS) 的 Web 服务的一部分。

WinCC OPC XML DA 服务器以 Web 页面形式为 OPC XML 客户端提供 OPC 过程数据。可以使用 HTTP 通过 Internet 访问 Web 页面。WinCC OPC XML DA 服务器的地址是: <http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx>

WinCC OPC XML DA 服务器在 WinCC 中不可见。OPC XML 客户端请求数据时, Web 服务器便自动启动 Web 服务。

为了建立成功的 OPC 通讯, 必须遵守以下内容:

- 必须激活 WinCC OPC XML DA 服务器的 WinCC 项目。
- 必须能够通过 HTTP 访问 WinCC OPC XML DA 服务器的计算机。

许可

为了操作 WinCC OPC XML DA 服务器, 必须在每台作为 OPC XML 服务器的 WinCC 计算机上安装下列许可证:

- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包

“字符串”类型变量的特殊功能

如果使用逻辑上代表浮点值的“字符串”类型的变量, OPC 客户端要写入和读取时可能会出现问题。

描述

OPC 客户端写入到字符串变量, 并不会以字符串形式输入新值, 而是作为浮点、双精度或十进制处理。

问题

表示小数值 (德语) 的逗号可能会丢失。这会导致错误的值。

如果以浮点、双精度或十进制格式请求读取值, 还会影响对字符串变量的读访问。

8.6 WinCC OPC XML DA 服务器

纠正方法

只使用相应的浮点型变量用于浮点值。仅以字符串格式访问字符串变量。

参见

WinCC OPC XML 客户机的功能 (页 252)

所支持的 WinCC 数据类型概述 (页 235)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.6.2 安装 WinCC OPC XML DA 服务器

简介

OPC-XML 允许通过 Internet 访问过程变量。为了操作 OPC XML，必须安装各种附加的软件组件。

说明

必须遵守此处所述安装步骤的顺序。否则，在安装期间可能会出现問題。

要求

- 安装 Internet 信息服务器 (IIS)
- 已通过 WinCC 产品 DVD 安装 Microsoft .NET Framework >= V4.6.2
- 使用 WinCC 安装程序安装 WinCC OPC XML 服务器

安装 Internet 信息服务器 (IIS)

对于 Windows Server 2012 R2 / 2016 / 2019，在服务器管理器中使用“Webserver (IIS)”角色组态相关角色服务中的设置。

选择以下设置：

- Web 管理工具：
 - IIS 管理服务
 - IIS 管理控制台
 - IIS 管理脚本和工具
- WWW 服务 > 常用 HTTP 功能或共享 HTTP 功能：
 - 标准文档
 - 静态内容
- WWW 服务 > 应用程序开发功能：
 - .NET 可扩展性
 - ASP.NET
 - ISAPI 扩展项
 - ISAPI 过滤器
- WWW 服务 > 安全：
 - 请求过滤
 - 基本验证
 - Windows 验证

说明

安装 Microsoft Internet 信息服务 (IIS) 时务必同时安装 ASP.NET

安装 Microsoft Internet 信息服务 (IIS) 时，务必安装 ASP.NET。

说明

WinCC OPC XML DA 服务器的 Web 服务通过端口 80 (HTTP) 进行通信。

确保针对所需的网络区域选择并激活“WWW 服务 (HTTP)”防火墙规则。

安装 WinCC OPC XML 服务器

在安装 WinCC 过程中，可选择 WinCC OPC XML DA 服务器。有关更多信息，请参阅 WinCC 信息系统中的“安装注意事项 > 安装 WinCC”(Installation Notes > Installing WinCC) 部分。

安装期间必须进行下列设置：

- 创建虚拟目录“WinCC-OPC-XML”
- 为目录定义访问权限

8.6.3 使用 IIS 设置安全设置（WinCC OPC XML DA 服务器）

简介

Internet 信息服务使得可以通过 Internet 访问 PC。因此需要定义适当的访问权限。

说明

如果对下列设置有任何疑问或遇到任何问题，请联系您的 Intranet/Internet 管理员。

步骤

1. 转到“控制面板 > 系统和安全 > 管理工具 > 计算机管理 > 服务和应用程序 > Internet 信息服务 (IIS) 管理器”(Control Panel > System and Security > Administrative Tools > Computer Management > Services and Applications > Internet Information Services (IIS) Manager)。
2. 选择虚拟目录“WinCC-OPC-XML”。
3. 在功能视图中打开“验证”(Authentication) 功能。
随即打开一个包含已安装的验证方法的列表。
4. 要启用匿名访问，请激活“匿名验证”(Anonymous authentication)。
5. 要启用验证访问，请激活“Windows 验证”(Windows authentication)。
6. 关闭所有打开的对话框。

说明

限制和风险

请注意设置访问选项时 Microsoft 提示的限制和风险。

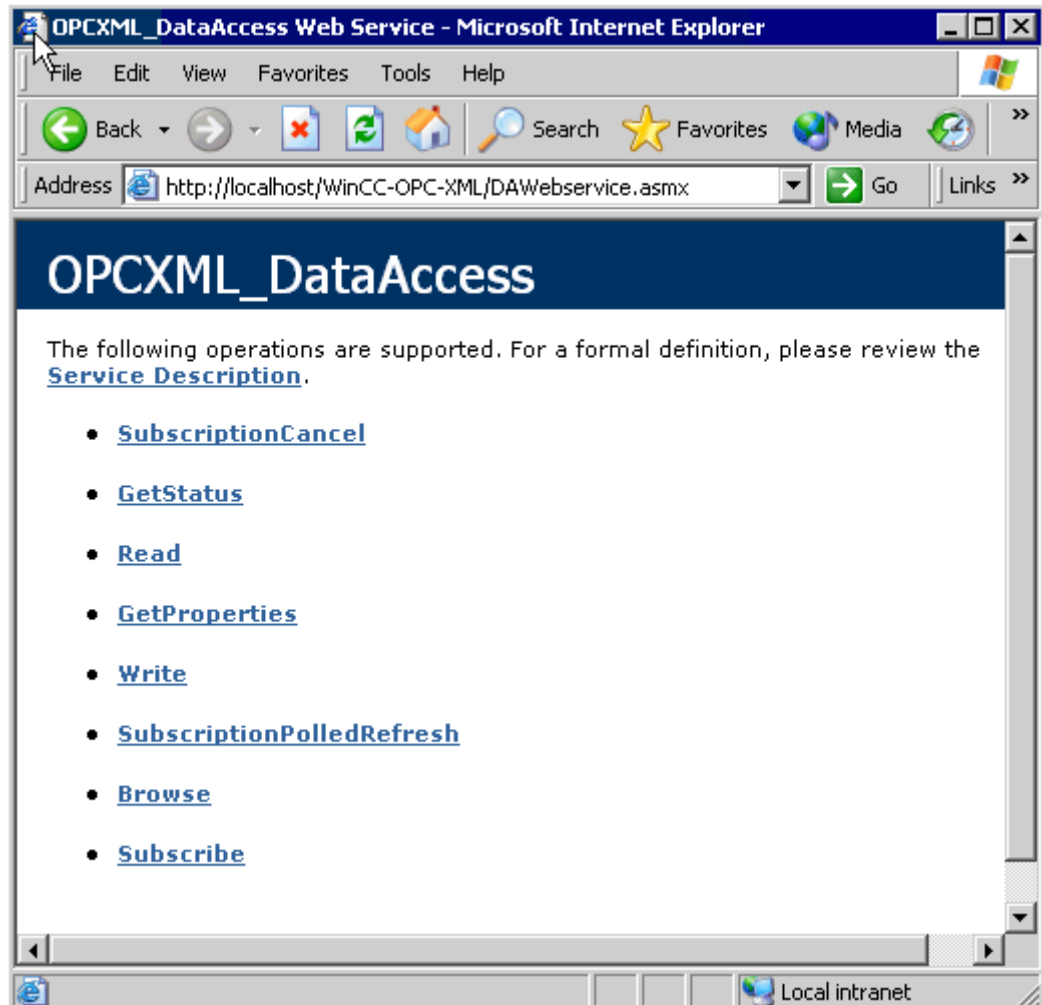
8.6.4 测试安装（WinCC OPC XML DA 服务器）

简介

OPC XML-DA 使得 OPC 过程数据可显示为 Web 页面。可以使用 HTTP 通过 Internet 访问 Web 页面。下列部分说明如何测试安装。

步骤

1. 在作为 WinCC OPC XML 服务器的计算机上启动 Internet Explorer。
2. 在地址栏中输入 URL“http://localhost/WinCC-OPC-XML/DAWebservice.asmx”。使用 <ENTER> 键确认输入。
3. 显示 OPC XML DA 功能请求时，安装成功完成。



8.7 WinCC OPC DA 服务器

8.7.1 WinCC OPC DA 服务端的功能

简介

WinCC OPC DA 服务器支持 OPC 数据访问规范 2.05a 和 3.00。这已经由兼容性测试确认。

WinCC OPC DA 服务端是一个 DCOM 应用程序。WinCC OPC DA 服务端使用该接口向 WinCC 客户端提供关于 WinCC 变量的所需信息。

如果 WinCC OPC DA 客户端正在通过连接访问 WinCC OPC DA 服务端，则服务端处于活动状态。为了建立成功的 OPC 通信，必须遵守以下内容：

- WinCC OPC DA 服务端的 WinCC 项目必须激活。
- 运行 WinCC OPC DA 服务端的计算机必须能够通过其 IP 地址访问。

安装

在安装 WinCC 过程中，可选择 WinCC OPC DA 服务端。安装后，WinCC OPC DA 服务端无需进一步组态便立即可使用。

可以在 WinCC 服务器或 WinCC 客户端上安装 WinCC OPC DA 服务端。

关于组态的注意事项

- 可在 WinCC 项目中将变量集中到变量组中。变量名称不能与组名称相同。
- 在 WinCC 中发起的每个写入请求（例如，通过 VBScript 或对象“IO 字段”）都总是被视为同步“写入”调用。WinCC OPC DA 服务器为此会使用“IOPCSyncIO::Write”接口。WinCC OPC DA 通道中不支持异步写入机制。

说明

如果计算机中“连接 -> 局域网设置”(Connections -> LAN Settings) 下的 Internet 选项被设置为自动检测设置，则通过 Web 服务访问 OPC DA 将花费较长的时间。

参见

查询 OPC DA 服务器名称 (页 648)

使用多个 OPC DA 服务器 (页 647)

WinCC 至 WinCC 的连接实例 (页 649)

WinCC - SIMATIC NET FMS OPC 服务器连接的实例 (页 653)

WinCC - SIMATIC NET S7 OPC 服务器连接的实例 (页 655)

WinCC - Microsoft Excel 连接的实例 (页 660)

所支持的 WinCC 数据类型概述 (页 235)

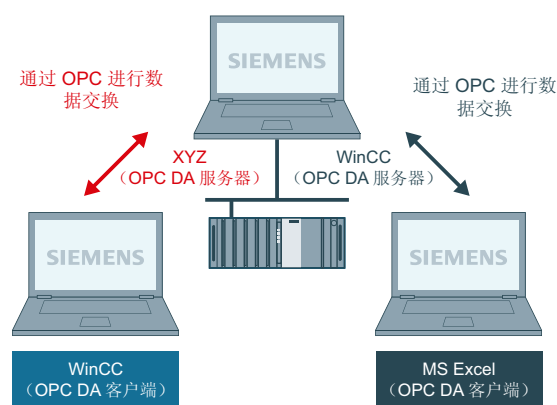
www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.2 使用多个 OPC DA 服务器

简介

在一台计算机上可以安装一个以上的 OPC DA 服务端，且任意数目的服务端都可并行地工作。通过这种方式，WinCC 的 OPC DA 服务端和另一（第三方）供应商的 OPC DA 服务端可在同一台计算机上彼此独立地运行。

WinCC OPC DA 客户端能通过第三方供应商的 OPC 服务端，访问自动化设备的过程数据。Microsoft Excel 的 OPC DA 客户端能使用 WinCC OPC DA 服务端访问 WinCC 数据。



有不同生产商提供的许多 OPC DA 服务端可用。其中每个 OPC DA 服务端都有唯一的名称 (ProgID) 以便识别。OPC DA 客户端必须使用该名称对 OPC 服务端进行寻址。

OPC 条目管理器可用于查询 OPC DA 服务端的名称。WinCC V7 的 OPC DA 服务端的名称为：“OPCServer.WinCC”。

参见

查询 OPC DA 服务器名称 (页 648)

8.7.3 查询 OPC DA 服务器名称

简介

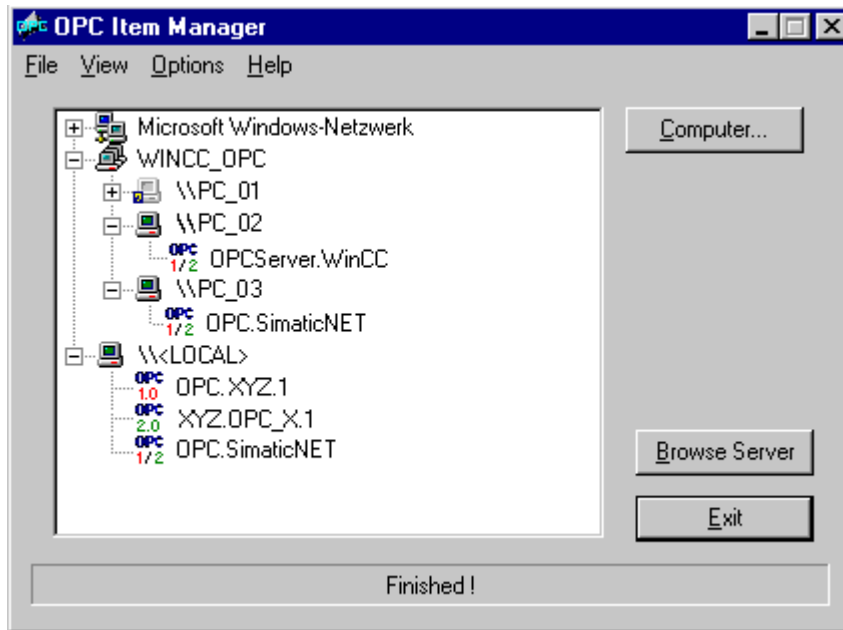
多个 OPC DA 服务器可安装在一台计算机上。OPC 条目管理器在选择窗口中显示可用于工作站的 OPC DA 服务器的名称。这些 OPC DA 服务器能够在同一计算机上运行或在已联网的计算机上运行。

要求

在 WinCC OPC DA 客户端的 WinCC 项目中添加“OPC”通道。

步骤

1. 在 WinCC OPC DA 客户端上，从“OPC 组 (OPCHN Unit#1)”通道单元的快捷菜单中选择“系统参数”。将打开“OPC 条目管理器”。
2. 在 OPC 条目管理器的浏览窗口中，选择要访问的计算机名称。
3. OPC 条目管理器在选择窗口中显示可用于计算机的 OPC DA 服务器的名称。



参见

OPC 条目管理器 (页 231)

8.7.4 OPC DA 连接的实例

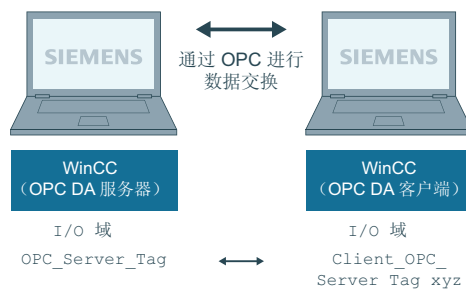
8.7.4.1 WinCC - WinCC 连接

WinCC 至 WinCC 的连接实例

引言

当建立 WinCC - WinCC 连接时，WinCC OPC DA 服务器和客户机之间的数据交换通过 WinCC 变量“OPC_Server_Tag”进行。客户机上的 WinCC 变量“Client OPC_Server_Tag_xyz”读取服务器上的 WinCC 变量“OPC_Server_Tag”。如果 WinCC OPC 服务器上的“OPC_Server_Tag”变量的数值改变，WinCC OPC DA 客户机上的 WinCC 变量“Client OPC_Server_Tag_xyz”的值也改变。客户机上的改变也会在服务器上反映出来。

变量值显示在两台计算机的 I/O 域中。



要求

- 两台带 WinCC 项目的计算机。
- 这两台计算机必须能够通过其 IP 地址被访问。

组态过程

建立 WinCC - WinCC 的连接需要以下组态：

1. 组态 WinCC OPC DA 服务器上的 WinCC 项目
2. 组态 WinCC OPC DA 客户机上的 WinCC 项目

参见

如何组态 WinCC OPC DA 服务器上的 WinCC 项目 (页 650)

组态 WinCC OPC DA 客户机上的 WinCC 项目 (页 651)

如何组态 WinCC OPC DA 服务器上的 WinCC 项目

引言

本节中介绍在 WinCC OPC DA 服务器上的 WinCC 项目中创建一个 WinCC 变量并显示在 I/O 域内。

步骤

1. 从 WinCC OPC DA 服务器上“内部变量”图标的快捷菜单中选择“新建变量”。创建一个类型为“有符号 16 位数”，名为“OPC_Server_Tag”的新变量。
2. 启动“图形编辑器”，打开一个新的画面。
3. 在画面中添加一个 I/O 域。从“智能对象”下的对象列表中选择“I/O 域”对象。将打开“I/O 域组态”对话框。



4. 在“变量”域中，输入名称“OPC_Server_Tag”。
5. 设置更新周期为“2 秒”，域类型为“I/O 域”。
6. 单击“确定”关闭对话框并保存画面。
7. 单击图形编辑器中的“激活”按钮以启用 WinCC 项目。

参见

组态 WinCC OPC DA 客户机上的 WinCC 项目 (页 651)

组态 WinCC OPC DA 客户机上的 WinCC 项目

引言

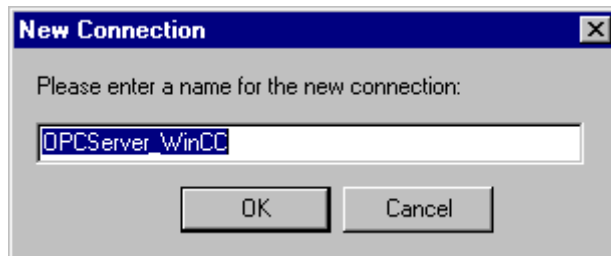
在本节中，将在 WinCC OPC DA 客户机上创建 WinCC 变量，以读取 WinCC OPC DA 服务器上的 WinCC 变量。变量值显示在 I/O 域中。

要求

- 在 WinCC OPC DA 客户机的 WinCC 项目中添加“OPC”通道。
- 在 WinCC OPC DA 服务器的 WinCC 项目中，组态一个数据类型为“有符号 16 位数”，名为“OPC_Server_Tag”的内部变量。
- 启用 WinCC OPC DA 服务器的 WinCC 项目。

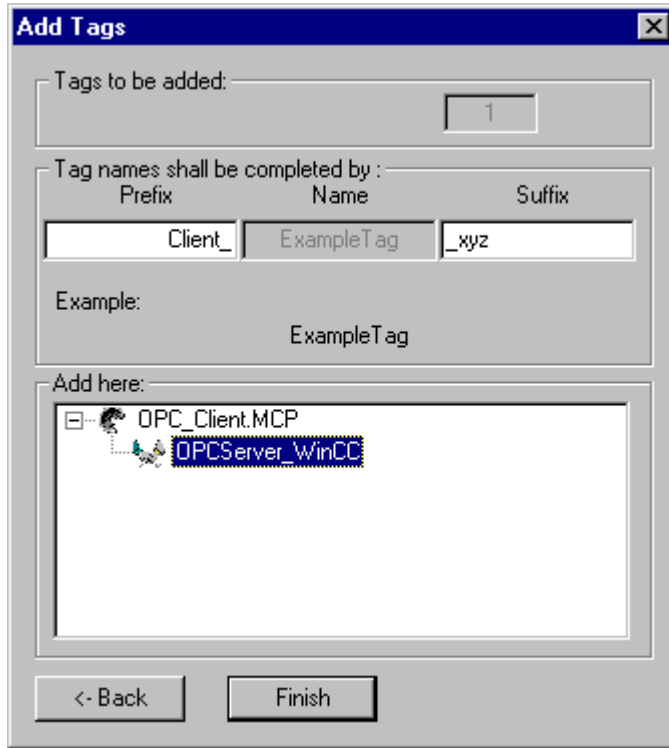
步骤

1. 在 WinCC OPC DA 客户机上，从“OPC 组 (OPCHN Unit#1)”通道单元的快捷菜单中选择“系统参数”。将打开 OPC 条目管理器。
2. 从选择对话框中选择要用作 OPC DA 服务器的计算机名称。从列表中选择“OPCServer.WinCC”。单击“浏览服务器”按钮。将打开“过滤标准”对话框。
3. 在“过滤标准”对话框中，单击“下一步->”按钮。在“OPCServer.WinCC ...”对话框中选择“OPC_Server_Tag”变量。单击“添加条目”按钮。
4. 如果到 OPC DA 服务器的连接已经存在，继续步骤 5。如果尚未组态连接，则会显示相应的消息。单击“是”。显示“New Connection”（新建连接）对话框。



输入“OPCServer_WinCC”作为连接的名称。单击“OK”（确定）。

5. 显示“Add Tags”（添加变量）对话框。
在前缀域中输入“Client_”，在后缀域中输入“_xyz”。选择连接“OPCServer_WinCC”。单击“Finish”（完成）。



6. 在“OPCServer.WinCC ...”对话框中，单击“<-返回”按钮。在“OPC 条目管理器”中，单击“退出”关闭 OPC 条目管理器。
7. 启动“图形编辑器”，打开一个新的画面。在画面中添加一个 I/O 域。从“智能对象”下的对象列表中选择“I/O 域”对象。将打开“I/O 域组态”对话框。
8. 在“变量”域中输入名称“Client_OPC_Server_Tag_xyz”。将更新周期设置为“2 秒”。将域类型设置为“I/O 域”。关闭对话框并保存画面。单击图形编辑器中的“激活”按钮以启用 WinCC 项目。
9. 在 WinCC OPC DA 服务器和客户机上的 I/O 域中，将显示组态变量的数值。在 WinCC OPC DA 服务器的 I/O 域中输入新的数值。新数值将显示在 WinCC OPC DA 客户机上的 I/O 域内。

参见

如何组态 WinCC OPC DA 服务器上的 WinCC 项目 (页 650)

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)

8.7.4.2 WinCC - SIMATIC NET FMS OPC 服务器的连接

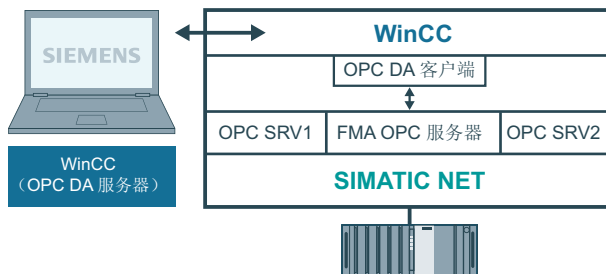
WinCC - SIMATIC NET FMS OPC 服务器连接的实例

简介

在 SIMATIC NET 的安装期间，可以选择要安装的 OPC 服务端。在以下实例中，将组态 WinCC 和 SIMATIC NET FMS OPC 服务端之间的连接。通过 SIMATIC NET FMS OPC 服务端，WinCC 可使用来自自动化设备的数据。

在本实例中，WinCC 用作 WinCC OPC DA 客户端。OPC 条目管理器显示为自动化设备组态的对象列表索引。

变量的当前值显示在一个 I/O 域中。SIMATIC NET FMS OPC 服务端上的变量值改变时，新的数值将会在 WinCC OPC DA 客户端的过程画面中反映出来。反之，在 I/O 域中输入的数值会发送到自动化设备上。



要求

- 安装有 WinCC、SIMATIC NET 软件的计算机。
- 已组态 SIMATIC NET FMS OPC 服务端。有关 SIMATIC NET S7 OPC 服务端设置的更多信息，请参阅 SIMATIC NET 文档。

组态步骤

在 WinCC OPC DA 客户端的 WinCC 项目中，需要下列组态：

1. 组态 WinCC - SIMATIC NET FMS OPC 服务端的连接

通信手册

通信手册包含通道组态的更多信息和扩展实例。本手册可在 Internet 上下载，地址如下：

- <http://support.automation.siemens.com/>

搜索项目号:

- A5E00391327

如何组态 WinCC - SIMATIC NET FMS OPC 服务器的连接

引言

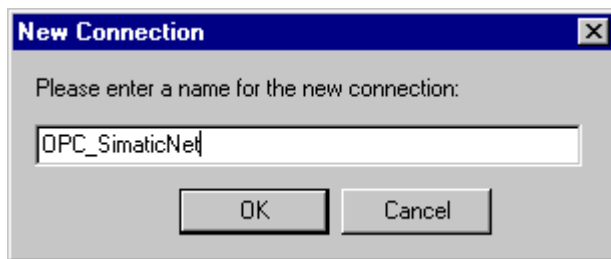
本节中，在 WinCC OPC DA 客户机的 WinCC 项目中组态一个访问 FMS 索引的 WinCC 变量。变量值显示在 I/O 域中。

要求

- 在 WinCC OPC DA 客户机的 WinCC 项目中添加“OPC”通道。

步骤

1. 在 WinCC OPC DA 客户机上，从“OPC 组 (OPCHN Unit#1)”通道单元的快捷菜单中选择“系统参数”。将打开 OPC 条目管理器。
2. 从选择对话框中选择要用作 OPC DA 服务器的计算机名称。从列表中选择“OPC.SIMATICNet”。单击“浏览服务器”按钮。将打开“过滤标准”对话框。
3. 在“过滤标准”对话框中，单击“下一步->”按钮。“OPC.SIMATICNet..”对话框打开。所有组态的 FMS 索引将显示在选择列表中。选择索引。单击“添加条目”按钮。
4. 如果到 SIMATIC NET FMS OPC 服务器的连接已经存在，继续步骤 5。如果尚未组态连接，则会显示相应的消息。单击“是”。显示“New Connection”（新建连接）对话框。



输入“OPC_SlimaticNET”作为连接的名称。单击“OK”（确定）。

5. 将打开“添加变量”对话框。
在前缀域中输入“Client_”，在后缀域中输入“_xyz”。选择连接“OPC_SimaticNET”。单击“完成”。
6. 在“OPC.SIMATICNet..”对话框中，单击“<- Back”（<- 返回）按钮。在“OPC 条目管理器”中，单击“退出”关闭 OPC 条目管理器。
7. 启动“图形编辑器”，打开一个新的画面。在画面中添加一个 I/O 域。从“智能对象”下的对象列表中选择“I/O 域”对象。将打开“I/O 域组态”对话框。

8. 在“变量”域中，输入变量的名称。将更新周期设置为“2 秒”。将域类型设置为“I/O 域”。
9. 单击“确定”关闭对话框并保存画面。单击图形编辑器中的“激活”按钮以启用 WinCC 项目。
10. 在 I/O 域中，将显示 FMS 索引的当前值。数值每 2 秒更新一次。在 I/O 域中输入数值。更改后的数值将传递到自动化设备。

参见

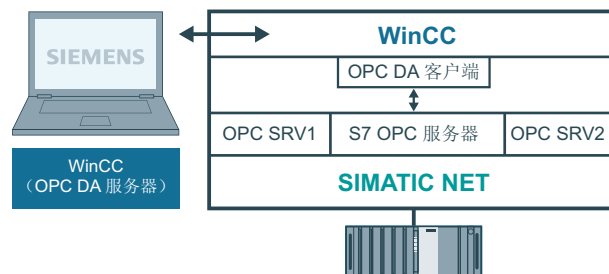
在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)

8.7.4.3 WinCC - SIMATIC NET S7-OPC 服务器的连接

WinCC - SIMATIC NET S7 OPC 服务器连接的实例

在 SIMATIC NET 的安装期间，可以选择要安装的 OPC 服务端。在下面的实例中，将组态 WinCC 到 SIMATIC NET S7 OPC 服务端的连接。通过 SIMATIC NET S7 OPC 服务端，WinCC 客户机可使用来自自动化设备的数据。

变量的当前值显示在 WinCC OPC 客户端的一个 I/O 域中。一旦 SIMATIC NET S7 OPC 服务端上的变量值改变，更改后的数值将显示在过程画面上。反之，在 I/O 域中输入的数值会发送到自动化设备上。



要求

- 安装有 WinCC、SIMATIC NET 软件的计算机。
- 已组态 SIMATIC NET S7 OPC 服务端。有关 SIMATIC NET S7 OPC 服务端设置的更多信息，请参阅 SIMATIC NET 文档。

组态步骤

为了建立 WinCC 到 SIMATIC NET S7 OPC 服务端的连接，需要下列组态：

1. 添加变量到 SIMATIC NET S7 OPC 服务端
2. 组态对 SIMATIC NET S7 OPC 服务端变量的访问

通信手册

通信手册包含通道组态的更多信息和扩展实例。本手册可在 Internet 上下载，地址如下：

- <http://support.automation.siemens.com/>

搜索项目号：

- A5E00391327

添加变量到 SIMATIC NET S7 OPC 服务器

引言

为了用 OPC 条目管理器显示变量，必须添加变量至 SIMATIC NET S7 OPC 服务器的地址空间。使用“OPC Scout”程序进行组态。使用 SIMATIC NET 安装程序安装 OPC Scout。对于本实例，将为自动化设备上的标记字“0”编址。

所用参数表

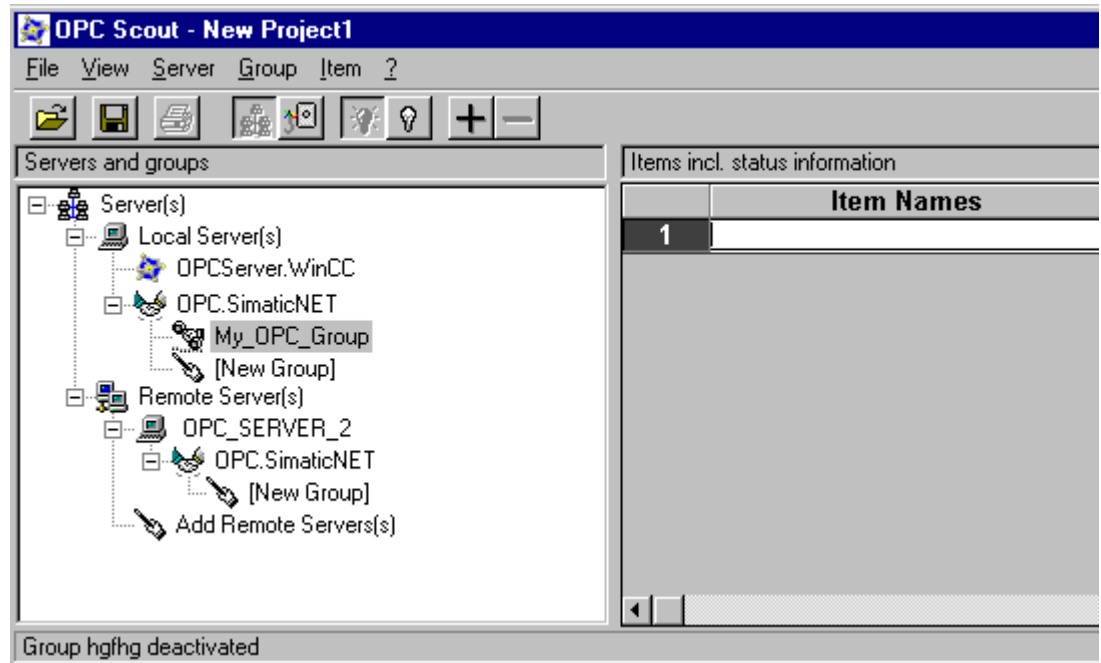
参数	数值
Data type	W
Range Byte	0
No. Values	1
Item alias	MW0

要求

- 在 SIMATIC NET 软件中组态 S7 连接。有关更多的信息，请参考 SIMATIC NET 文档。

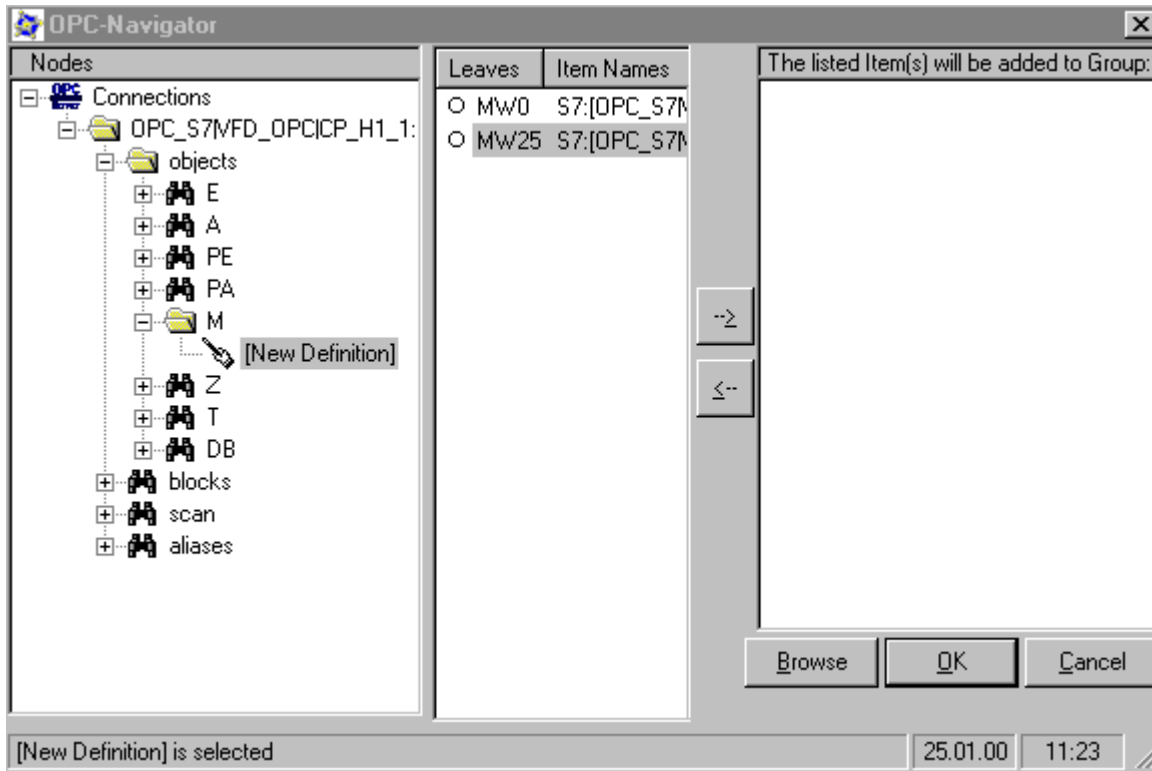
步骤

1. 通过“开始” → “程序” → “SimaticNet” → “OPCServer” → “OPCScout”打开“OPC Scout”。

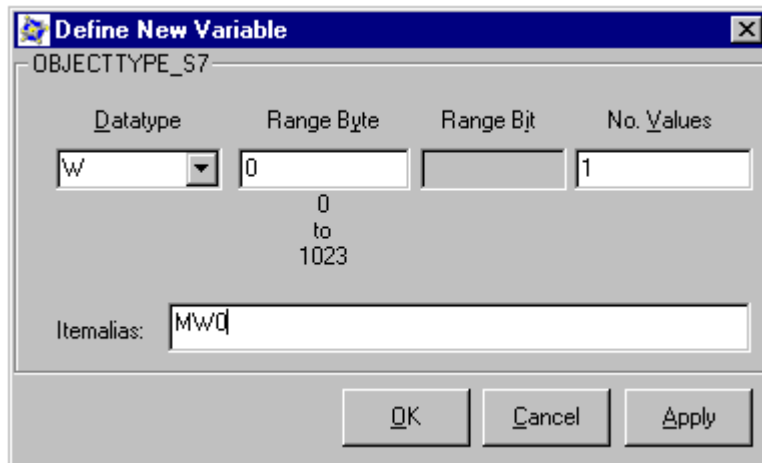


2. 选择“Local Server(s)”（本地服务器）下的“OPC.SimaticNet”。如果 SIMATIC S7 OPC 服务器不是运行在同一台计算机上，则选择“Server(s)”（服务器）快捷菜单中的“Add Remote Server(s)”（添加远程服务器）。在“Add Remote Server(s)”（添加远程服务器）对话框中输入用作 OPC 服务器的计算机名称，然后单击“OK”（确定）关闭对话框。
3. 选择“OPC.SimaticNet”快捷菜单中的“Connect”（连接）。显示“Add Group”（添加组）对话框。为该组输入一个名称。单击“OK”（确定）关闭对话框。

4. 从所添加组的快捷菜单中选择“Add Item”（添加条目）。“OPC Navigator”（OPC 浏览器）打开。



- 5. 在“OPC Navigator”（OPC 浏览器）中选择“Objects”（对象）下的“M”(标记)。双击“(New Definition)”（新的定义）打开“Define New Tag”（定义新的变量）对话框。
- 6. 在“Define New Tag”（定义新的变量）对话框中输入表格中的参数。



单击“OK”（确定）关闭“Define New Tag”（定义新的变量）对话框。

- 7. 在 OPC 浏览器的“Leaves”（叶）区域中标记变量“MW0”。单击“-->”按钮。单击 OPC 浏览器中的“OK”（确定）。

参见

组态对 SIMATIC NET S7 OPC 服务器变量的访问 (页 659)

组态对 SIMATIC NET S7 OPC 服务器变量的访问

引言

本节中，将在 WinCC OPC DA 客户机的 WinCC 项目中组态 WinCC 变量。该变量将访问 SIMATIC NET S7 OPC 服务器地址空间中的变量“MW0”。变量值显示在 I/O 域中。

要求

- 使用 OPC Scout 创建“MW0”变量。
- 在 WinCC OPC DA 客户机的 WinCC 项目中添加“OPC”通道。

步骤

1. 选择“OPC 组 (OPCHN Unit#1)” 快捷菜单中的“系统参数”。将打开 OPC 条目管理器。
2. 从选择对话框中选择用作 OPC 服务器的计算机名称。从列表中选择“OPC.SIMATICNet”。单击“浏览服务器”按钮。将打开“过滤标准”对话框。
3. 在“过滤标准”对话框中，单击“下一步->”按钮。“OPC.SIMATICNet..”对话框打开。选择“MW0”变量。单击“添加条目”按钮。
4. 如果到 SIMATIC NET FMS OPC 服务器的连接已经存在，继续步骤 5。如果尚未组态连接，则会显示相应的消息。单击“是”。显示“New Connection”（新建连接）对话框。



输入“OPC_SlimaticNET”作为连接的名称。单击“OK”（确定）。

5. 将打开“添加变量”对话框。
在前缀域中输入“Client_”，在后缀域中输入“_xyz”。选择连接“OPC_SimaticNET”。单击“完成”。
6. 在“OPC.SIMATICNet..”对话框中，单击“<- Back”（<- 返回）按钮。在“OPC 条目管理器”中，单击“退出”关闭 OPC 条目管理器。
7. 启动图形编辑器并打开画面。在画面中添加一个 I/O 域。从“智能对象”对象列表中选择“I/O 域”对象。将打开“I/O 域组态”对话框。

8.7 WinCC OPC DA 服务器

8. 在“变量”域中，输入名称“Client_MW0_xyz”。将更新周期设置为“2 秒”。将域类型设置为“I/O 域”。
9. 关闭对话框并保存画面。单击图形编辑器中的“激活”按钮以启用 WinCC 项目。
10. S7 变量的当前值显示在 WinCC OPC DA 客户机的 I/O 域中。数值每 2 秒更新一次。在 I/O 域中输入数值。更改后的数值将传递到自动化设备。

参见

添加变量到 SIMATIC NET S7 OPC 服务器 (页 656)

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 238)

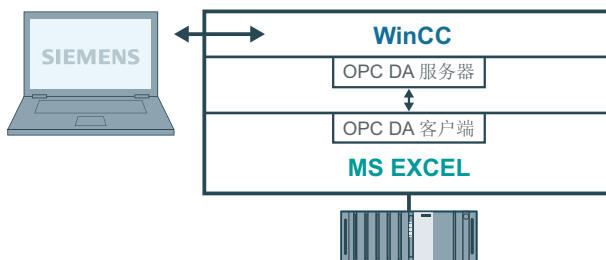
8.7.4.4 WinCC - Microsoft Excel 的连接

WinCC - Microsoft Excel 连接的实例

简介

在本实例中，将在 Microsoft Excel 中使用 Visual Basic 编辑器创建一个 OPC DA 客户机。该 OPC DA 客户机读取 WinCC OPC DA 服务器上 WinCC 项目中的 WinCC 变量，并将值写入单元格中。如果在该单元格中输入一个新值，该值将会传递到 WinCC OPC DA 服务器上。

需使用同时安装了 WinCC 和 Microsoft Excel 的计算机来进行连接。



组态步骤

必须在 Microsoft Excel 中进行下列组态：

1. 在 Microsoft Excel 的 Visual Basic 编辑器中创建一个 OPC DA 客户机
2. 在 Microsoft Excel 中组态对 WinCC 变量的访问

参见

如何组态在 Microsoft Excel 中访问 WinCC 变量 (页 664)

在 Microsoft Excel 中创建 OPC DA 客户机 (页 661)

在 Microsoft Excel 中创建 OPC DA 客户机


简介

要将 Microsoft Excel 用作 OPC DA 客户端，必须在 Microsoft Excel 的 Visual Basic 编辑器中创建特殊的脚本。

要求

具有基本的 Microsoft Excel Visual Basic 编辑器知识。

步骤

1. 打开 Microsoft Excel，将出现新的工作簿。
2. 在“Visual Basic 编辑器”的“工具”菜单中， 单击“宏”。将打开 Microsoft Excel 的 Visual Basic 编辑器。
3. 在“Visual Basic 编辑器”的“工具”菜单中选择“引用...”。将显示“引用 - VBAProject”对话框。在可用引用的列表中找到条目“Siemens OPC DAAutomation 2.0”。选中相应的复选框。单击“确定”(OK)。
4. 复制以下所显示的脚本。该脚本仅在在线帮助中可用。
5. 双击 Visual Basic 编辑器项目窗口中的“Sheet1”，打开新的代码窗口。
6. 将脚本粘贴到代码窗口中。
7. 从“文件”菜单中选择“保存”。从“文件”菜单中选择“关闭并返回到 Microsoft Excel”。

示例脚本

```

Option Explicit
Option Base 1

Const ServerName = "OPCServer.WinCC"

Dim WithEvents MyOPCServer As OpcServer
Dim WithEvents MyOPCGroup As OPCGroup
Dim MyOPCGroupColl As OPCGroups
Dim MyOPCItemColl As OPCItems
Dim MyOPCItems As OPCItems
Dim MyOPCItem As OPCItem

Dim ClientHandles(1) As Long
Dim ServerHandles() As Long
Dim Values(1) As Variant
Dim Errors() As Long
Dim ItemIDs(1) As String
Dim GroupName As String
Dim NodeName As String

'-----
' Sub StartClient()
' Purpose: Connect to OPC_server, create group and add item
'-----
Sub StartClient()
    ' On Error GoTo ErrorHandler
    '----- We freely can choose a ClientHandle and GroupName
    ClientHandles(1) = 1
    GroupName = "MyGroup"
    '----- Get the ItemID from cell "A1"
    NodeName = Range("A1").Value
    ItemIDs(1) = Range("A2").Value
    '----- Get an instance of the OPC-Server
    Set MyOPCServer = New OpcServer
    MyOPCServer.Connect ServerName, NodeName

    Set MyOPCGroupColl = MyOPCServer.OPCGroups
    '----- Set the default active state for adding groups
    MyOPCGroupColl.DefaultGroupIsActive = True
    '----- Add our group to the Collection
    Set MyOPCGroup = MyOPCGroupColl.Add(GroupName)

    Set MyOPCItemColl = MyOPCGroup.OPCItems
    '----- Add one item, ServerHandles are returned
    MyOPCItemColl.AddItem 1, ItemIDs, ClientHandles, ServerHandles, Errors
    '----- A group that is subscribed receives asynchronous notifications
    MyOPCGroup.IsSubscribed = True
Exit Sub

```

```

ErrorHandler:
  MsgBox "Error: " & Err.Description, vbCritical, "ERROR"
End Sub

'-----
' Sub StopClient()
' Purpose: Release the objects and disconnect from the server
'-----
Sub StopClient()
  '----- Release the Group and Server objects
  MyOPCGroupColl.RemoveAll
  '----- Disconnect from the server and clean up
  MyOPCServer.Disconnect
  Set MyOPCItemColl = Nothing
  Set MyOPCGroup = Nothing
  Set MyOPCGroupColl = Nothing
  Set MyOPCServer = Nothing
End Sub

'-----
' Sub MyOPCGroup_DataChange()
' Purpose: This event is fired when a value, quality or timestamp in our Group has changed
'-----
'----- 如果安装了 OPC-DA Automation 2.1, 使用:
Private Sub MyOPCGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems As Long,
ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As
Date)
  '----- Set the spreadsheet cell values to the values read
  Range("B2").Value = CStr(ItemValues(1))
  Range("C2").Value = Hex(Qualities(1))
  Range("D2").Value = CStr(TimeStamps(1))
End Sub

'-----
' Sub worksheet_change()
' Purpose: This event is fired when our worksheet changes, so we can write a new value
'-----
Private Sub worksheet_change(ByVal Selection As Range)
  '----- Only if cell "B3" changes, write this value
  If Selection <> Range("B3") Then Exit Sub
  Values(1) = Selection.Cells.Value
  '----- Write the new value in synchronous mode
  MyOPCGroup.SyncWrite 1, ServerHandles, Values, Errors
End Sub

```

参见

如何组态 WinCC OPC DA 服务器上的 WinCC 项目 (页 650)

如何组态在 Microsoft Excel 中访问 WinCC 变量

引言

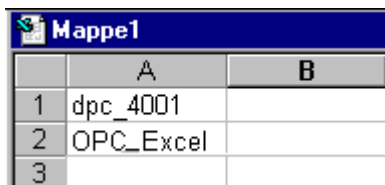
Excel OPC DA 客户机读取 WinCC OPC DA 服务器的 WinCC 变量并将变量值写入单元格中。在 WinCC OPC DA 服务器的 WinCC 项目中，变量值显示在一个 I/O 域内。如果单元格中的变量值改变，则在 WinCC OPC DA 服务器 I/O 域中的数值也随之改变。

要求

- 在 WinCC OPC DA 服务器的 WinCC 项目中，组态名称为“OPC_Excel”且数据类型为“有符号 16 位数”的内部变量。
- 将“OPC_Excel”变量的数值写入 WinCC OPC DA 服务器的 WinCC 项目的 I/O 域内。
- 启用 WinCC OPC DA 服务器的 WinCC 项目。

步骤

1. 在 Microsoft Excel 的单元格 A1 中，输入用作 OPC 服务器的计算机名称。在单元格 A2 中，输入变量名称“OPC_Excel”。



	A	B
1	dpc_4001	
2	OPC_Excel	
3		

2. 在 Excel 的“工具”菜单中，选择“宏” → “宏”。“宏”对话框将打开。从宏列表中选择条目“Sheet1.StartClient”。单击“运行”以启动 OPC 客户机。
3. 变量的值会写入单元格 B2 中，质量代码写入 C2 中，时间标志写入 D2 中。
4. 在单元格 B3 中输入新的数值。更改后的数值显示在 WinCC OPC 服务器上的 I/O 域内。
5. 在 Excel 的“工具”菜单中，选择“宏” → “宏...”。“宏”对话框将打开。从宏列表中选择条目“Sheet1.StopClient”。单击“运行”停止 OPC 客户机。

8.8 WinCC OPC HDA 服务器

8.8.1 WinCC OPC HDA 服务器的功能

简介

WinCC OPC HDA 服务器是 DCOM 应用程序，为 OPC HDA 客户端提供来自归档系统的必需数据。使用条目句柄访问数据。可以进行读访问或写访问。还可以分析数据。

WinCC OPC HDA 服务器支持 OPC 历史数据访问 1.20 规范。这已经由适应性测试确认。

下列章节说明数据结构的设计，以及 WinCC OPC HDA 服务器支持的属性、集合和功能。这不是详细的说明，而是最重要信息的摘要。有关更多信息，可参见“OPC 历史数据访问 1.20”规范。

安装

在安装 WinCC 期间，可以选择 WinCC OPC HDA 服务器。可以选择对 WinCC 归档系统的访问是否具有写功能。安装后，WinCC OPC DA 服务器无需进行组态，立即可用。

如果安装时未选择写访问，则只能读取和分析 WinCC 归档系统中的数据。如果选择了写访问，可以分析、添加、删除和更新 WinCC 归档系统中的数据。

可以在 WinCC 服务器或 WinCC 客户端上安装 WinCC OPC HDA 服务器。

许可

为了操作 WinCC OPC HDA 服务器，必须在每台作为 OPC HDA 服务器的 WinCC 计算机上安装下列许可证：

- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包

OPC HDA 客户端

所有符合 OPC 历史数据访问 1.20 规范的 OPC HDA 客户端都可以访问 WinCC OPC HDA 服务器。也可自己创建 OPC HDA 客户端。通过创建专有 OPC HDA 客户端，可以满足大多数用户特定需求。

如何使用 OPC HDA 客户端的实例包括：

- 分析和判断归档数据
- 从不同的 OPC HDA 服务器对归档进行统计过程控制

要使用 OPC HDA 客户端请求历史数值，需要在组态过程中注意以下几点：

- 选择的查询周期应能使客户端在发送下一次查询之前收到所请求的数据。周期太短可能造成接收数据时时间延迟过长。
- WinCC 服务器的 CPU 负载取决于每次查询的变量数。

使用组态的交换写访问循环归档

在运行系统中，数据在 WinCC 服务器的循环归档中进行修改。

只有在创建数据后立即进行更改时才会将更改传送给换出归档。

如果循环归档的相关归档段已经换出，则随后不会在换出归档中完成更改。在 WinCC 服务器上删除了归档段，修改的数据也将删除。

参见

质量代码 (页 672)

WinCC OPC HDA 服务器的数据结构 (页 666)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2 WinCC OPC HDA 服务器的数据结构

8.8.2.1 WinCC OPC HDA 服务器的数据结构

引言

WinCC OPC HDA 服务器上的数据是结构化的。以下列出了可用的数据结构。这不是详细的说明，而是最重要信息的摘要。有关更多信息，可参见“OPC 历史数据访问 1.20”规范。

数据结构

	描述
属性	为原始数据提供附加的质量特性。属性包括关于归档的数据类型、规范等。有关更多信息，请参见所支持属性的概述。
配件	指定时间间隔中原始数据的摘要。集合包括平均值、最小值和最大值等。有关更多信息，请参见所支持集合的概述。
开始时间/结束时间	设置时间间隔的开始点和结束点。
范围值	在开始和结束时记录的值。如果没有可用的范围值，靠近这些时间的数值将用作范围值。
原始数据	WinCC 归档系统中特定时间间隔的数据。这些数据包含时间标志和质量等级。
条目句柄	唯一指定给 WinCC 归档变量。
条目标识号	WinCC 归档变量的唯一标识符。条目标识号可用于获取条目句柄。

参见

受支持函数的概述 (页 669)

WinCC OPC HDA 服务器的时间格式 (页 670)

受支持属性的概述 (页 667)

受支持配件的概述 (页 668)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.2 受支持属性的概述

引言

下表包括了 WinCC OPC HDA 服务器支持的属性。有关更多信息，可参见“OPC 历史数据访问 1.20”规范。

属性

属性	属性标识号	描述
条目标识号	OPCHDA_ITEMID	指示要访问的 WinCC 归档变量。
条目数据类型	OPCHDA_DATA_TYPE	指示 WinCC 归档变量的数据类型。
描述	OPCHDA_DESCRIPTION	返回 WinCC 归档变量的说明。该说明在 WinCC 变量记录中定义。
工程单位	OPCHDA_ENG_UNITS	设置度量单位的显示。设定标签在 WinCC 变量记录中定义。

参见

WinCC OPC HDA 服务器的数据结构 (页 666)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.3 受支持配件的概述

引言

下表列出了 WinCC OPC HDA 服务器支持的集合。有关更多信息，可参见“OPC 历史数据访问 1.20”规范。

配件

配件	描述
OPCHDA_COUNT	返回指定时间间隔中原始数据的计数。
OPCHDA_START	返回时间间隔开始时原始数据的初始值。
OPCHDA_END	返回时间间隔结束时原始数据的最终值。
OPCHDA_AVERAGE	返回指定时间间隔中原始数据的平均值。
OPCHDA_TIMEAVERAGE	返回指定时间间隔中原始数据按时间的平均值。
OPCHDA_TOTAL	返回指定时间间隔中的总和值。
OPCHDA_STDEV	返回指定时间间隔中原始数据的标准偏差。
OPCHDA_MINIMUMACTUALTIME	返回指定时间间隔中原始数据的最小值及其时间标志。
OPCHDA_MINIMUM	返回指定间隔中原始数据的最小值。

配件	描述
OPCHDA_MAXIMUMACTUAL TIME	返回指定时间间隔中原始数据的最大值及其时间标志。
OPCHDA_MAXIMUM	返回指定间隔中原始数据的最大值。
OPCHDA_DELTA	返回指定时间间隔中原始数据中第一个值和最后一个值的差。
OPCHDA_REGSLOPE	返回指定时间间隔中原始数据回归线的斜率。
OPCHDA_REGCONST	返回起始点处原始数据的回归值。
OPCHDA_REGDEV	返回指定时间间隔内原始数据回归的标准偏离。
OPCHDA_VARIANCE	返回指定时间间隔内原始数据的变化。
OPCHDA_RANGE	返回指定时间间隔内原始数据的 OPCHDA_MAXIMUM 和 OPCHDA_MINIMUM 的差。
OPCHDA_DURATIONGOOD	返回原始数据质量好的时间段。时间段以秒为单位。
OPCHDA_DURATIONBAD	返回原始数据质量差的时间段。时间段以秒为单位。
OPCHDA_PERCENTGOOD	返回质量好的原始数据的百分比。
OPCHDA_PERCENTBAD	返回质量差的原始数据的百分比。
OPCHDA_WORSTQUALITY	返回指定时间间隔内原始数据的最差质量。

参见

WinCC OPC HDA 服务器的数据结构 (页 666)

WinCC OPC HDA 服务器的功能 (页 665)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.4 受支持函数的概述

引言

下表列出了 WinCC OPC HDA 服务器支持的函数。OPC HDA 客户机可使用这些函数进行数据交换。有关更多信息，可参见“OPC 历史数据访问 1.20”规范。

读

函数	描述
ReadRaw	返回指定时间间隔内的原始数据及其质量和时间标志。
ReadProcessed	返回计算出的数值、数值质量和指定时间间隔的时间标志。计算出的数值取决于所选的集合。
ReadAtTime	返回特定时间间隔内的原始数据及其质量和时间标志。如果没有数值可用，将插入该点的数值。
ReadAttribute	返回指定时间间隔的条目属性和时间标志。

参见

WinCC OPC HDA 服务器的功能 (页 665)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.5 WinCC OPC HDA 服务器的时间格式

引言

在 WinCC OPC HDA 服务器上通过设置开始时间和结束时间指定时间间隔。指定的时间间隔确定历史数据的观察期。在指定时间时，必须保持特定的格式。

下列选项可用于指定时间格式：

- 基于 UTC 的绝对时间
- 相对于服务器当地时间的相对时间

根据 UTC 的绝对值

缺省状态下，WinCC OPC HDA 服务器使用相同的世界时间（UTC）作为其时间基准。该时间对应于格林威治标准时间（即中欧时间减去一小时）。

时间格式

YYYY/MM/DD hh:mm:ss.msmsms

参数

YYYY = 年份

MM = 月份

DD = 日

hh = 小时

mm = 分钟

ss = 秒

ms = 毫秒

输入实例

2002/06/10 09:27:30.000

指定相对于当地时间的的时间

对于该选项，输入的时间相对于服务器的当地时间。当地时区在计算机的“日期/时间”控制面板中设置。

时间格式

关键字 +/- 误差 1 +/- 误差(n)

误差是与服务器当地时间的差。

关键字

NOW = 服务器上当前的当地时间

SECOND = 当前秒钟

MINUTE = 当前分钟

HOUR = 当前小时

DAY = 当前日

WEEK = 当前星期

MONTH = 当前月份

YEAR = 当前年份

误差

+/-S = 秒钟差

+/-M = 分钟差

+/-H = 小时差

+/-D = 日差

+/-W = 星期差

8.8 WinCC OPC HDA 服务器

+/-MO = 月份差

+/-Y = 年份差

实例:

DAY - 1D = 前一日

DAY-1D + 7H30 = 前一日的 7:30

MO-1D+5H = 上个月最后一天的 5:00

NOW-1H15M = 1 小时 15 分钟前

YEAR+3MO= 本年度的四月

参见

WinCC OPC HDA 服务器的功能 (页 665)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.3 质量代码

简介

质量代码用来判断原始数据的状态和质量。OPC 的质量代码在“数据访问自定义界面标准 3.00 版”规范的“6.8 OPC 质量标记”中有说明。

WinCC OPC HDA 服务器的质量代码

代码	OPC	描述	质量
0x00040000 0	OPCHDA_RAW	指示原始数据传输的质量。	好 劣 不确定
0x00080000 0	OPCHDA_CALCULATED	指示计算出的数据传输的质量。	好 劣 不确定
0x00100000 0	OPCHDA_NOBOUND	在开始点或结束点处未发现范围值。	劣

代码	OPC	描述	质量
0x00200000 0	OPCHDA_NODATA	指定时间间隔内没有发现任何原始数据。	劣
0x00400000 0	OPCHDA_DATALOST	所选间隔内的原始数据未完全归档。	劣

参见



www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.4 支持的写访问

引言



下表列出了 WinCC OPC HDA 服务器支持的写访问。

表格元素：

	描述
周期性归档	要归档的过程值存储在循环归档中。循环归档由数目可组态的数据缓冲区组成。定义数据缓冲区的大小和时间段（例如以天为单位）。当所有数据缓冲区都满时，第一个数据缓冲区中的过程数据被覆盖。
交换之后循环归档	为了使数据缓冲区中的过程数据不被覆盖，可以将其先交换出来（导出）。
	受 WinCC 支持。
	不受 WinCC 支持。

写访问

以后添加过程值

周期性归档	交换之后循环归档	受 WinCC 支持	描述
是	否		如果时间段包含在循环归档中，以后可以添加过程值。
是	是		相应时间段的数据缓冲区被交换至归档备份。以后，不能将过程值添加到归档备份。

8.8 WinCC OPC HDA 服务器

周期性归档	交换之后循环归档	受 WinCC 支持	描述
否	否	☹️	循环归档不可用。不能存储过程值。
不可以	可以	☹️	循环归档不可用。不能存储过程值。

在运行系统中添加过程值

周期性归档	交换之后循环归档	受 WinCC 支持	描述
是	否	😊	过程值被添加到循环归档当前有效的数据缓冲区中。

插入将来的过程值

周期性归档	交换之后循环归档	受 WinCC 支持	描述
是	否	☹️	在写访问期间，在未来不能添加值。
否	否	☹️	写访问时，在未来不能添加值。

删除过程值

周期性归档	交换之后循环归档	受 WinCC 支持	描述
是	否	😊	时间段包含在循环归档中时，可以删除过程值。
是	是	☹️	相应时间段的数据缓冲区被交换至归档备份。可以从归档备份删除过程值。
否	否	☹️	循环归档不可用。不能存储过程值。
不可以	可以	☹️	循环归档不可用。不能存储过程值。

编辑过程值

周期性归档	交换之后循环归档	受 WinCC 支持	描述
是	否	☺	时间段包含在循环归档中时，可以编辑过程值。
是	是	☹	相应时间段的数据缓冲区被交换至归档备份。可以在归档备份中编辑过程值。
否	否	☹	循环归档不可用。不能存储过程值。
不可以	可以	☹	循环归档不可用。不能存储过程值。

8.8.5 OPC HDA 连接实例

8.8.5.1 OPC HDA 连接实例

引言

在下面的实例中，将组态 WinCC 和 OPC HDA 客户机之间的连接。来自 WinCC 归档系统的数据通过 WinCC OPC HDA 服务器变得可用。OPC HDA 客户机通过条目句柄访问数据。要简化组态过程，将使用 OPC HDA 浏览器。

将使用来自 OPC 基金会的 OPC HDA 客户机。所有符合 OPC 历史数据访问 1.20 规范的 OPC HDA 客户机都可以访问 WinCC OPC HDA 服务器。

要求

- 在 WinCC OPC HDA 服务器的 WinCC 项目中，创建名称为“OPC_HDA”、数据类型为“无符号 16 位数”的内部变量。
- 创建在 WinCC 归档系统中称为“HDA_ProcessValueArchive”的过程值归档。
- 在过程值归档“HDA_ProcessValueArchive”中创建称为“OPC_HDA_Tag”的 WinCC 归档变量。将 WinCC 归档变量与内部变量“OPC_HDA”链接。
- 在运行系统列表中，启动变量记录运行系统，禁用图形运行系统。
- 启动 WinCC OPC HDA 服务器的 WinCC 项目。

组态步骤

将 WinCC 连接到 OPC HDA 客户机需要以下组态：

1. 使用 HDA 服务器浏览器组态对 WinCC 归档变量的访问
2. 读取 WinCC 归档变量的数值

参见

如何使用 HDA 服务器浏览器来组态对 WinCC 归档变量的访问 (页 678)

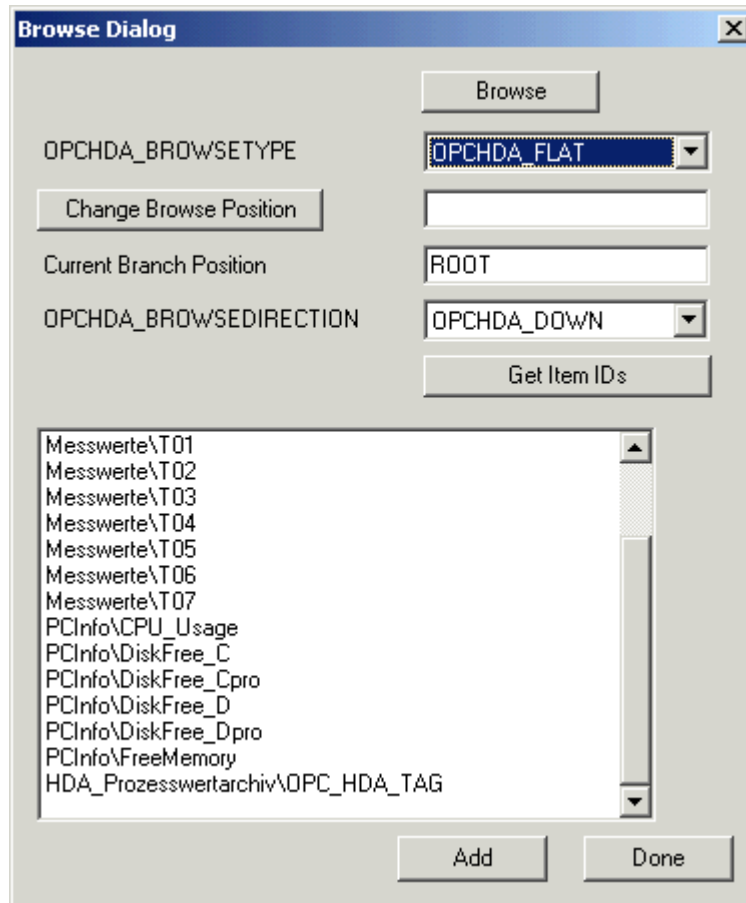
HDA 服务器浏览器 (页 677)

读取 WinCC 归档变量的数值 (页 679)

8.8.5.2 HDA 服务器浏览器

引言

OPC HDA 客户机通过条目句柄访问变量值。为了易于组态，WinCC OPC HDA 服务器支持浏览器功能。OPC HDA 客户机可以使用 HDA 服务器浏览器来搜索 WinCC OPC HDA 服务器的地址空间。数据以过程值归档分级列出。



说明

如果不通过 HDA 服务器浏览器来访问 WinCC 归档变量，就需要手动组态条目标识号。寻址 WinCC 归档变量时，计算机名称（服务器前缀）包含在路径中。条目标识号具有下列语法：Server-prefix::process_value_archive\WinCC_archive_tag。

参见

如何使用 HDA 服务器浏览器来组态对 WinCC 归档变量的访问 (页 678)
www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.5.3 如何使用 HDA 服务器浏览器来组态对 WinCC 归档变量的访问

简介

在本部分中，OPC HDA 客户端将用于访问 WinCC 归档变量。
将使用来自 OPC 基金会的 OPC HDA 客户端。
HDA 服务器浏览器将用于组态访问。

说明

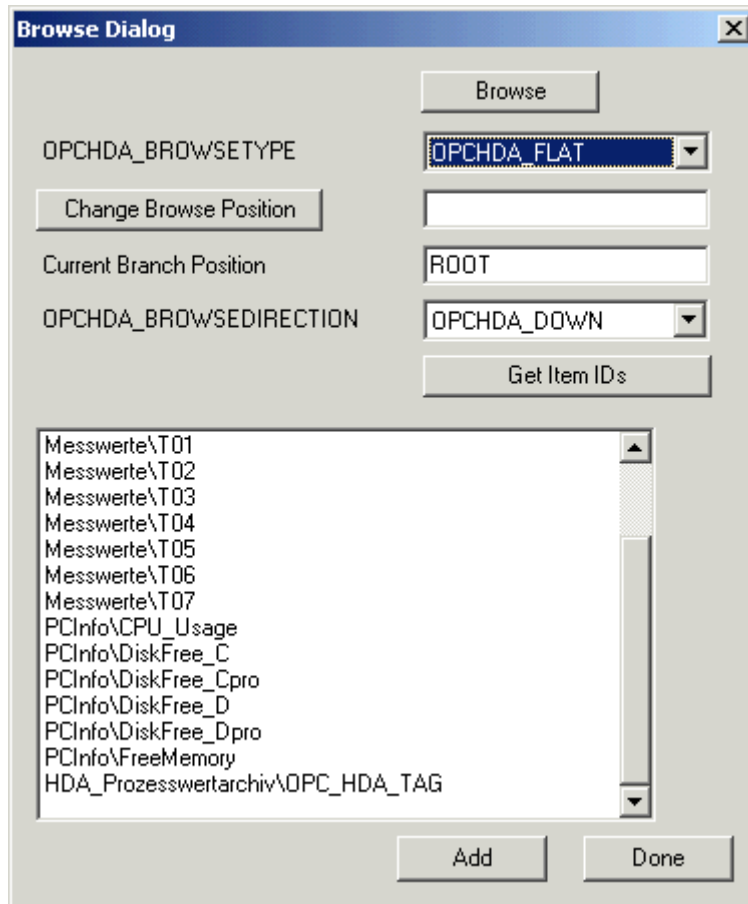
演示客户端

此处的 OPC HDA 客户机是来自 OPC 基金会的演示客户端。
可访问 Internet 获取该客户端的来源，网址为：<http://www.opcfoundation.org>。

操作步骤

1. 将“SampleClientHDA.exe”文件从 WinCC 安装路径“Siemens\WinCC\documents\English”复制到选择的文件夹中。
2. 双击“SampleClientHDA.exe”文件。
“HDA 客户端”(HDA client) 程序将打开。
3. 在“服务器名称”区域中，选择条目“OPCServerHDA.WinCC.1”。
4. 单击“连接”。
确认随后的对话框。

5. 在 HDA 客户端中单击“浏览”。
“浏览对话框”(Browse Dialog) 对话框将打开。



6. 在“OPCHDA_BROWSETYPE”域中选择“OPCHDA_FLAT”。
7. 在选择窗口中，选择“HDA_ProcessValueArchive_HDA_TAG”条目。
8. 单击“添加”，然后单击“完成”关闭对话框。

有关详细信息，请访问 <http://www.opcfoundation.org>。

参见

读取 WinCC 归档变量的数值 (页 679)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.5.4 读取 WinCC 归档变量的数值

引言

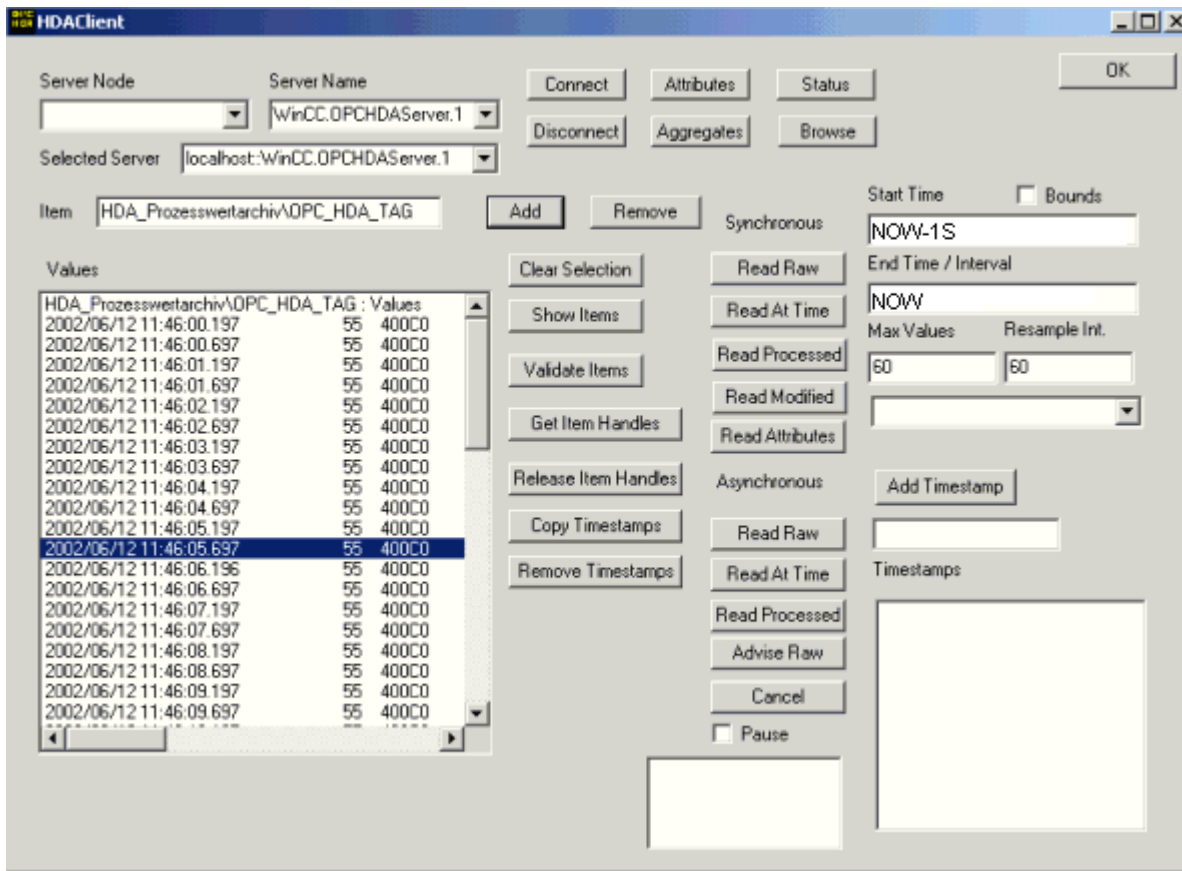
本节说明如何访问和读取 WinCC 归档变量。

要求

- OPC HDA 客户机必须在运行。

步骤

1. 在 HDA 客户机中单击“Show Items”（显示条目）。
2. 在 HDA 客户机中单击“Get Item Handles”（获取条目句柄）。
3. 双击“Value”（数值）选择域中的“HDA_ProcessValueArchive_HDA_Tag”。
4. 在“Start Time”（开始时间）域中输入“NOW-10S”。在“End Time”（结束时间）域中输入“NOW”。



5. 单击“Read Raw”（读取原始数据）。数值及其质量代码和时间标志显示在“Values”（数值）选择域中。

8.8.6 WinCC 中 OPC HDA 服务器针对非周期性记录的特殊功能

简介

在 WinCC 中，会周期性或非周期性地执行变量记录。WinCC OPC HDA 服务器会视变量记录方法而定以不同的方式工作：

- 对于所有周期性记录的值，OPC HDA 服务器都会遵照 OPC Foundation 的 HDA 规范运行。OPC 总计将进行线性内插。
- OPC Foundation 的 HDA 规范中不包括非周期性记录的变量。OPC 总计将进行增量式内插。尤其是如果变量长时间未经历变化，某个时间段内将没有可用的数据。应对以下事项加以考虑，从而在出现上述情况时仍可获得有效的数据。

说明

对于非周期性记录的变量，OPC HDA 服务器不遵从 OPC 的规定。OPC Foundation 的 HDA 规范不认可非周期性记录的变量，因此任何归档服务器都无法处理非周期性记录的变量。所支持的总计将依照 OPC HDA 规范进行计算。不支持任何非显式调用的函数。

说明

如果启用对过程值归档的写入访问，则不能再添加任何值。

组态非周期性记录的变量

为组态非周期性记录的变量，需要为变量启用“进行段更改后归档”(Archive after segment change) 设置。段发生变化时，此操作会在新记录中输入最新的有效值。

对于非周期性记录的变量 WinCC OPC HDA 服务器所支持的总计

OPC HDA 服务器支持以下总计：

- OPCHDA_MINIMUM
- OPCHDA_MAXIMUM
- OPCHDA_AVERAGE
- OPCHDA_END
- OPCHDA_INTERPOLATIVE
- OPCHDA_TIMEAVERAGE
- OPCHDA_TOTAL

8.8 WinCC OPC HDA 服务器

- OPCHDA_DURATIONGOOD
- OPCHDA_PERCENTGOOD

对于非周期性记录的变量 WinCC OPC HDA 服务器所支持的函数

- 仅限有“约束”的 ReadRaw。针对变量的 ReadRaw 在执行时必须始终施加“约束”，以找到无记录值变化区域的最后一个实际存储值。
- ReadProcessed
- DeleteRaw
- DeleteAtTime
- Insert
- InsertReplace
- Replace

计算非周期性记录变量的总计

总计的计算基于扩展的“RawData”数据记录，该记录除包含实际存储值外，还包含用于计算的虚拟数据点。WinCC OPC HDA 服务器会根据“ReadProcessed”的要求准备包含的“RawData”。计算所需的虚拟数据点由边界实际数据点组成。虚拟数据点包括以下重要点：

- “StartTime”的值
- “EndTime”的值
- 间隔限制的值

实例

“00:59:00”、“01:02:00”和“01:03:00”的值是为非周期变量记录变量存储的。OPC HDA 客户机通过“ReadProcessed”要求具有以下参数的总计：

- StartTime = 01:00:00
- EndTime = 01:04:00
- Interval = 00:02:00

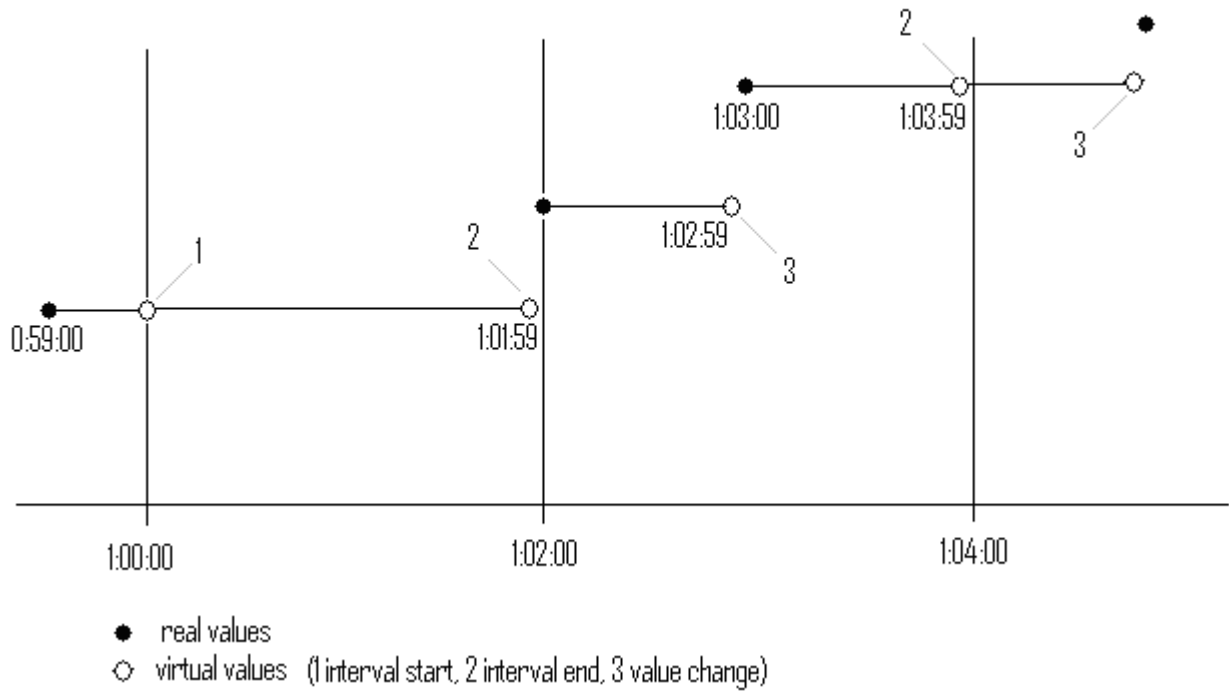
说明

在限制（“EndTime”/“Interval”）处生成虚拟值时，时间段长度始终比限制处计算的时间戳短 1 μs。

下表中使用了 1 秒的增量，以提高总览效果。下图图解了该实例。

OPC 服务器使用下面的“RawData”来计算总计：

编号	时间戳	实际存储值	生成的虚拟值
1	00:59:00	1.00	
2	01:00:00		1.00
3	01:01:59		1.00
4	01:02:00	2.00	
5	01:02:59		2.00
6	01:03:00	3.00	
7	01:03:59		3.00



8.9 WinCC OPC A&E 服务器

8.9.1 WinCC OPC A&E 服务器的功能

简介

WinCC OPC A&E 服务器是 DCOM 应用程序。OPC A&E 客户端通过订阅随时了解 WINCC 消息的状态变化情况。OPC A&E 客户端可以对订阅使用过滤器。该过滤器确定显示哪些消息和属性。

WinCC OPC A&E 服务器支持规范 OPC Alarm&Event 1.10。这已经由适应性测试确认。

下列章节说明 WinCC 消息系统在 OPC A&E 上的显示，以及 WinCC OPC A&E 服务器支持的属性。这不是详细的说明，而是最重要信息的摘要。有关更多信息，可参见“OPC 报警 & 事件 1.10”规范。

安装

可以在安装 WinCC 期间选择 WinCC OPC A&E 服务器。安装后，WinCC OPC A&E 服务器无需进行组态，立即可用。

可以在 WinCC 服务器和 WinCC 客户端上安装 WinCC OPC A&E 服务器。

许可

为了运行 WinCC OPC A&E 服务器，必须在每台作为 OPC A&E 服务器的 WinCC 服务器上安装下列许可证：

- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包

服务器类型

WinCC OPC A&E 服务器支持条件事件和简单事件。此外，还支持跟踪事件。

与条件相关的事件服务器

对于条件相关的事件服务器，事件与某个条件相关。例如，条件可以是变量超出限值。一旦超出边界值，便会在 WinCC 中生成消息。此消息在 OPC A&E 中显示为报警。

简单事件服务器

简单事件是提示 OPC A&E 客户端关于事件的消息。简单事件包括启动或退出程序等。

说明

在使用冗余系统时，请注意如下事项：

更新变量时，与内部变量互连的简单事件会发送两次。

第一个消息由主站触发，第二个消息由备用站触发。

跟踪事件服务器

如果在过程中发生了变化，OPC A&E 客户端会接收到消息。例如，这一变化可能是调节器调整。

OPC A&E 客户端

所有符合 OPC Alarms & Events 1.10 规范的 OPC A&E 客户端都可以访问 WinCC OPC A&E 服务器。也可自己创建 OPC A&E 客户端。通过创建专有 OPC 客户端，可以满足大多数用户特定需求。例如，OPC A&E 客户端可用于对来自多重 OPC A&E 服务器的报警进行分析和一般归档。

参见

OPC A&E 质量代码 (页 692)

在 OPC A&E 上映射 WinCC 消息系统 (页 685)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.2 在 OPC A&E 上映射 WinCC 消息系统

8.9.2.1 在 OPC A&E 上映射 WinCC 消息系统

导言

组态 WinCC 消息系统期间，将进行设置以确定哪些过程事件产生消息。此消息在 OPC A&E 中显示为报警。下面的表格列出了报警最重要的参数。还描述了如何通过 WinCC 消息系统使信息可用。更多信息，请参阅“报警结构”。

概述

OPC	WinCC 消息系统
Source	指示消息源。消息源的格式为“<服务器前缀>::@LOCALMACHINE::”。
Time	发布已接收、已发送和已确认消息的时间标志。发布 UTC（协调世界时）格式的时间标志。
Type	指示是简单事件、跟踪事件还是条件相关事件。WinCC - POC A&E 服务器支持简单事件、条件相关事件和跟踪事件。
Severity	指示 WinCC 消息的优先级。
EventCategory	返回消息类别。有关该主题的更多信息，请参阅“显示消息等级和类型”。
Message	指示相应消息号的消息文本。
ConditionName	指示消息号。
ChangeMask	指示消息改变后的状态。更多信息，请参阅“确认方法”。
NewState	返回消息状态。更多信息，请参阅“确认方法”。
ConditionQuality	返回消息质量。更多信息，请参阅“质量代码”。
AckRequired	指示消息是否需要确认(接收)。
ActiveTime	返回已接收消息的时间标志。
EventAttribute	列出相关消息所需的属性。更多信息，请参阅“WinCC 消息系统的属性”。
Quality	返回消息的质量代码。
Cookie	从 OPC A&E 服务器返回 cookie。cookie 对应于 WinCC 报警系统中的消息号

参见

确认方法 (页 690)

WinCC 消息系统的属性 (页 688)

映射 WinCC 消息类别和消息类型 (页 687)

8.9.2.2 映射 WinCC 消息类别和消息类型

简介

WinCC 消息系统向用户提示过程中的使用干扰和操作情况。WinCC 消息始终属于与事件类别有关的特定消息等级和消息类型。

通过“CcAeProvider.ini”文件组态 OPC 上 WinCC 消息系统的映射。

事件类别

在 WinCC OPC A&E 服务器上，为每种消息等级和类型的组合创建一个事件类别。

事件类别由类别 ID 和描述性的“类别说明”确定。类别 ID 由消息等级和消息类型的 WinCC 内部 ID 组成；类别说明由消息等级和消息类型组成。

说明

如果 OPC A&E 服务器在连通站的 WinCC 客户端上运行，则与其链接的 OS 服务器必须具有相同的消息等级和消息类型组态。否则，使用的 OPC 客户端必须直接访问 OS 服务器。

可以通过报警属性“CLASSNAME”和“TYPENAME”精准确定消息等级和消息类型的名称。

8.9.2.3 映射 WinCC 消息优先级

引言

通过 OPC 服务器的属性“Severity”来显示 WinCC 消息的优先级。

在 WinCC 消息系统中组态报警时，可以组态一个在 0 到 16 之间的优先级。OPC A&E 规范定义了 severity 的数值范围为 1 到 1000，其中 1 代表最低 severity，1000 代表最高 severity。

因此，WinCC 优先级的值相应显示为 OPC severity。在标准映射中，WinCC 优先级 0 将映射为 OPC severity 1。所有其它的优先级值都将以线性方式插入一直到 severity 1000。可在 CcAeProvider.ini 文件中组态其它优先级映射规则。

8.9.2.4 WinCC 消息系统的属性

简介

下表列出了 WinCC 消息系统的 OPC 属性。这些属性在 WinCC 消息系统中组态。某些属性仅供 WinCC 内部使用，与 OPC A&E 客户机无关。因此未列出这些属性。

属性

OPC 属性	WinCC 消息系统	数据类型
CLASSNAME	返回消息类别名称。	VT_BSTR
TYPENAME	返回消息类型名称。	VT_BSTR
FORECOLOR	返回用于显示已接收、已发送和已确认消息的文本颜色。	VT_I4
BACKCOLOR	返回用于显示已接收、已发送和已确认消息的背景色。	VT_I4
FLASHCOLOR	返回闪烁颜色。	VT_I4
FLAGS	指示消息是否需要确认（回执）。	VT_I4
TEXT01	返回用户文本块 01 的内容。	VT_BSTR
TEXT02	返回用户文本块 02 的内容。	VT_BSTR
TEXT03	返回用户文本块 03 的内容。	VT_BSTR
TEXT04	返回用户文本块 04 的内容。	VT_BSTR
TEXT05	返回用户文本块 05 的内容。	VT_BSTR
TEXT06	返回用户文本块 06 的内容。	VT_BSTR
TEXT07	返回用户文本块 07 的内容。	VT_BSTR
TEXT08	返回用户文本块 08 的内容。	VT_BSTR
TEXT09	返回用户文本块 09 的内容。	VT_BSTR
TEXT10	返回用户文本块 10 的内容。	VT_BSTR
PROCESSVALUE 01	返回过程值块 01 的内容。	VT_VARIANT
PROCESSVALUE 02	返回过程值块 02 的内容。	VT_VARIANT
PROCESSVALUE 03	返回过程值块 03 的内容。	VT_VARIANT

OPC 属性	WinCC 消息系统	数据类型
PROCESSVALUE 04	返回过程值块 04 的内容。	VT_VARIANT
PROCESSVALUE 05	返回过程值块 05 的内容。	VT_VARIANT
PROCESSVALUE 06	返回过程值块 06 的内容。	VT_VARIANT
PROCESSVALUE 07	返回过程值块 07 的内容。	VT_VARIANT
PROCESSVALUE 08	返回过程值块 08 的内容。	VT_VARIANT
PROCESSVALUE 09	返回过程值块 09 的内容。	VT_VARIANT
PROCESSVALUE 10	返回过程值块 10 的内容。	VT_VARIANT
STATETEXT	返回状态消息。	VT_BSTR
INFOTEXT	返回消息的信息文本。	VT_BSTR
LOOPINALARM	指示是否组态报警回路。	VT_I4
CLASSID	返回消息类别 ID。	VT_I4
TYPEID	返回消息类型 ID。	VT_I4
MODIFYSTATE	输出消息的状态变量值。	VT_I4
AGNR	返回产生消息的自动化设备的编号。	VT_I2
CPUNR	返回产生消息的 CPU 的编号。	VT_I2
DURATION	指示消息接收、发送和确认的时间间隔。	VT_I4
COUNTER	输出自运行系统启动以来的消息数。	VT_I4
QUITSTATETEX T	指示消息是否被确认。	VT_BSTR
QUITCOUNT	输出未确认的活动消息数。	VT_I4
PARAMETER	输出消息参数。（消息组态画面）	VT_BSTR
BLOCKINFO	返回消息块的当前内容。	VT_BSTR
ALARMCOUNT	输出未决消息数。	VT_I4
LOCKCOUNT	输出锁定消息数。	VT_I4
PRIORITY	指示已组态的消息优先级。	VT_I4
APPLICATION	输出触发了消息的应用程序。	VT_BSTR

OPC 属性	WinCC 消息系统	数据类型
COMPUTER	输出处理了消息的计算机的名称。	VT_BSTR
USER	输出处理了消息的用户的名称。	VT_BSTR
COMMENT	输出消息注释。	VT_BSTR

8.9.2.5 确认方法

简介

在 WinCC 中，确认原则是消息从“进入”到“离开”的显示及处理方法。在 WinCC OPC A&E 服务器上，该消息状态用参数“ChangeMask”和“NewState”管理。

条件、简单和跟踪事件

通常将来自 WinCC 系统的消息作为条件事件发送到客户机。为了使消息作为简单事件处理，在组态消息等级时必须满足下列条件：

- “确认进入”未激活。
- “无离开状态的消息”激活。

根据映射组态，消息等级“没有确认的系统”和消息类型“操作消息”的消息将作为 OPC 跟踪事件传送。

ChangeMask

“ChangeMask”参数将记录消息状态改变的地址。

参数值：

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

“NewState”参数指示改变后的消息状态。

参数值:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

概述

WinCC	NewState	ChangeState
已接收的消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
带回执的已发送消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
不带回执的已发送消息	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
已确认的消息（消息未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
已确认的消息(消息不再未决)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
锁定的消息	-----	OPC_CHANGE_ENABLED_STATE
解锁的消息	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
已接收的、确认的消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
已接收的、带回执的已发送消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
已接收的、不带回执的已发送消息	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
由系统确认的消息（消息未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
由系统确认的消息（消息不再未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

WinCC	NewState	ChangeState
紧急确认的消息（消息未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
紧急确认的消息（消息不再未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

参见

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.3 OPC A&E 质量代码

简介

质量代码用于判断消息的状态和质量。OPC 的质量代码在“数据访问自定义界面标准 3.00 版”规范的“6.8 OPC 质量标记”中有说明。

质量代码

代码	质量	状态
0xC0	OPC_GOOD	确定
0x40	OPC_UNCERTAIN	在不确定的情况下返回，例如如果发生延迟确认（回执）。
0x00	OPC_BAD	在与源的连接中断时返回。

8.9.4 OPC A&E 连接实例

8.9.4.1 OPC A&E 连接实例

简介

在下面的实例中，将组态 WinCC 和 OPC A&E 客户端之间的连接。WinCC OPC A&E 服务器使来自 WinCC 消息系统的数据变得可用。

OPC A&E 客户端通过订阅来获知 WinCC 消息的状态改变。

所有符合 OPC Alarms&Events 1.10 规范的 OPC A&E 客户端都可以访问 WinCC OPC A&E 服务器。

组态步骤

WinCC 和 OPC A&E 客户端之间的连接需要下列组态：

1. 组态对 WinCC 消息系统的访问

参见

如何组态访问 WinCC 消息系统 (页 693)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.4.2 如何组态访问 WinCC 消息系统

引言

本节中，OPC 基金会的 OPC A&E 客户机将访问 WinCC 消息系统。

说明

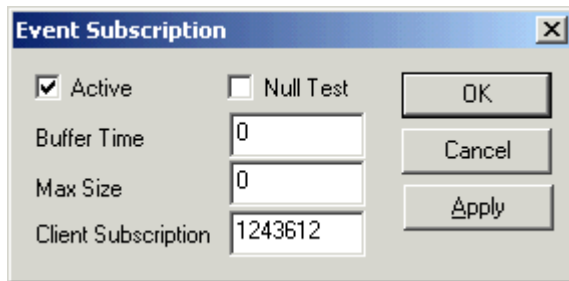
此处描述的 OPC A&E 客户机是来自 OPC 基金会的演示客户机。源代码可以参见 Internet 网址 <http://www.opcfoundation.org>。

要求

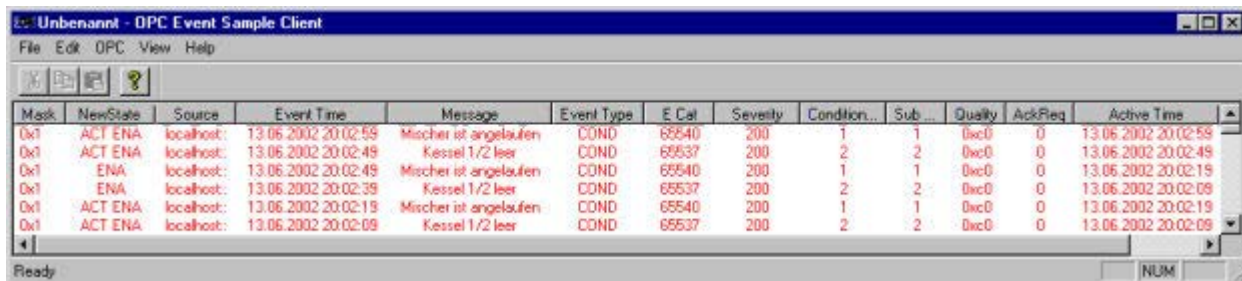
- 在 WinCC OPC A&E 服务器的 WinCC 项目中创建多个“二进制”数据类型的内部变量。
- 在 WinCC OPC A&E 服务器的 WinCC 项目中组态 WinCC 消息系统。将消息链接至内部变量。
- 用图形编辑器组态画面。将 WinCC 报警控件和 I/O 域添加到画面中。将消息变量链接到图形对象。
- 在启动列表中启用“报警记录运行系统”。
- 启用 WinCC OPC A&E 服务器的 WinCC 项目。

步骤

1. 将“SampleClientAE.exe”文件从文件夹“Siemens\WinCC\documents\english”复制到您选择的文件夹中。该应用程序仅在在线帮助中可用。
2. 在菜单栏中选择“OPC”>“连接...”。在“OPC 报警服务器”对话框中选择“OPC.WinCC-AlarmsEvent”。单击“确定”关闭对话框。
3. 从菜单栏中选择“OPC”>“事件订阅...”。将打开“事件订阅”对话框。
4. 选中对话框中标签为“激活”的复选框。在“缓冲时间”和“最大尺寸”域中输入“1000”。单击“确定”关闭“事件订阅”对话框。



5. 来自 WinCC 消息系统的消息将显示在 OPC 事件实例客户机中。



6. 从菜单栏中选择“OPC”>“过滤器”。将打开“过滤器”对话框。从“事件类别”域中选择类别。单击“确定”关闭“过滤器”对话框。
7. 符合过滤标准的消息将显示在 OPC 事件实例客户机中。

“缓冲时间”和“最大大小”参数

根据 OPC 规范，在 WinCC 中按如下组态“缓冲时间”和“最大大小”参数：

OPC 客户机要求返回值	WinCC 使用
缓冲时间 < 100 OPC_S_INVALIDBUFFERTIME	修改后的缓冲时间 = 100
100 ≤ 缓冲时间 ≤ 600000 S_OK	修改后的缓冲时间 = 缓冲时间
缓冲时间 > 600000 OPC_S_INVALIDBUFFERTIME	修改后的缓冲时间 = 600000
最大大小 = 0 OPC_S_INVALIDMAXSIZE	修改后的最大大小 = 1000
0 < 最大大小 < 10 OPC_S_INVALIDMAXSIZE	修改后的最大大小 = 10
10 ≤ 最大大小 ≤ 1000 S_OK	修改后的最大大小 = 最大大小
最大大小 = 1000 OPC_S_INVALIDMAXSIZE	修改后的最大大小 = 1000

创建用户时，可以设置参数。然而，在创建之后，不能使用 SetState() 修改已存在的用户。

更多的信息，请参阅 <http://www.opcfoundation.org>。

参见

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.5 带层级访问的 OPC A&E 服务器

8.9.5.1 OPC A&E 服务器的功能

简介

OPC-A&E 服务器使用 DCOM 服务在具有 OPC 能力的应用程序之间传送消息。OPC A&E 服务器支持规范 OPC Alarm&Event 1.10。

下一章说明 WinCC 消息系统在带层级访问的 OPC A&E 上的映射，以及 OPC A&E 服务器支持的属性。本文档包括对特定信息的概述。有关更多信息，可参见“OPC 报警 & 事件 1.10”规范。

工作原理

OPC-A&E 客户端通过订阅接收 WinCC 消息。可以使用订阅过滤器减少随订阅传送的事件数。OPC-A&E 客户端可针对每个显示消息属性的事件类别进行设置。

安装

可以在安装 WinCC 期间选择 WinCC OPC A&E 服务器。安装后，WinCC OPC A&E 服务器无需进行组态，立即可用。

可以在 WinCC 服务器和 WinCC 客户端上安装 WinCC OPC A&E 服务器。

许可

为了运行 WinCC OPC A&E 服务器，必须在每台作为 OPC A&E 服务器的 WinCC 服务器上安装下列许可证：

- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包

事件类型

带层级访问的 OPC A&E 服务器支持条件事件、简单事件和跟踪事件。

与条件有关的事件

对于与条件有关的事件，事件会与某个条件相关联。例如，条件可以是变量超出限值。这种违反限制的情况会产生一条消息，该消息在 OPC A&E 中显示为报警。

简单事件

简单事件是提示 OPC A&E 客户端关于事件的消息。简单事件包括启动或退出程序等。

说明

在使用冗余系统时，请注意如下事项：

更新变量时，与内部变量互连的简单事件会发送两次。

第一个消息由主站触发，第二个消息由备用站触发。

跟踪事件

跟踪事件连同操作员输入消息一并发送到 OPC A&E 客户端。通过在过程中进行人工干预触发操作员输入消息。

OPC A&E 客户端

所有符合 OPC Alarms & Events 1.10 规范的 OPC A&E 客户端都可以访问 OPC A&E 服务器。也可自己创建 OPC A&E 客户端。通过创建专有 OPC 客户端，可以满足大多数用户特定需求。例如，可使用 OPC A&E 客户端对来自不同 OPC A&E 服务器的报警进行分析和联合归档。无法对已归档消息进行确认；只能对当前报警和事件进行确认。

如果使用带层级访问功能的 OPC A&E，并想要使用所有功能，则可能需要修改当前使用的 OPC A&E 客户端。

说明

OPC 上的文档

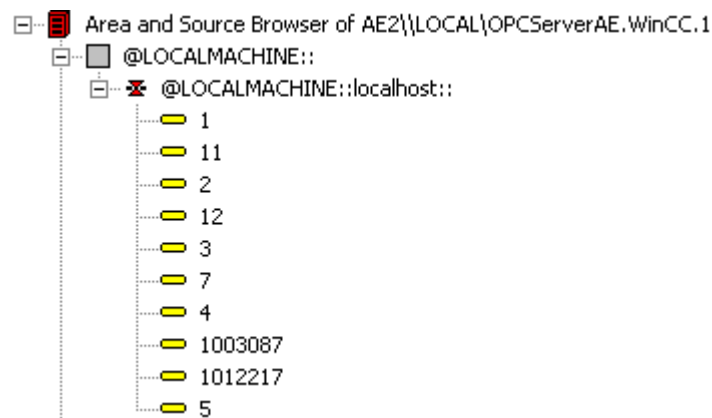
可以在“接口 > OPC - 用于过程控制的 OLE”一章中找到有关 OPC 的附加信息。

8.9.5.2 OPC A&E 与带层级访问的 OPC A&E 之间的差异

使用 OPC A&E 显示消息

OPC A&E 服务器支持用于访问消息系统的“条件事件”和“简单事件”。对于“条件事件”，将针对每个来源显示消息号。由于一台 WinCC 服务器可容纳许多消息号，因此难以维护消息概览。

下图显示的是 OPC 浏览器中的显示实例：



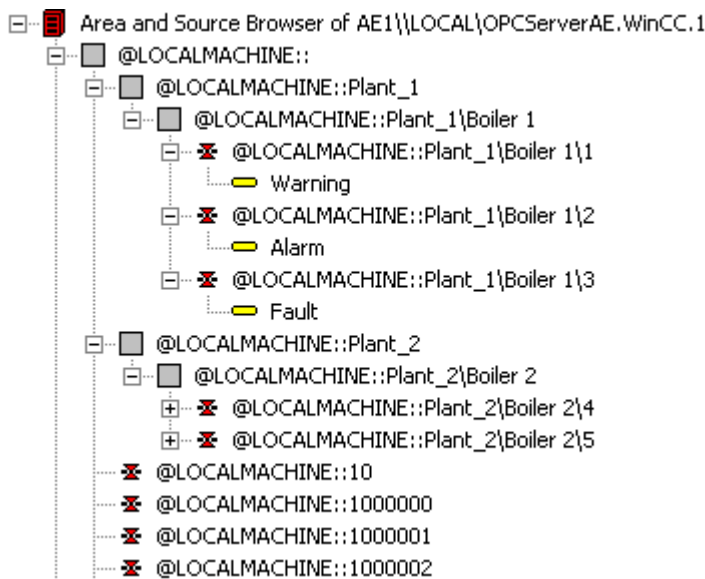
使用 OPC A&E 和层级访问显示消息

带层级访问功能的 OPC A&E 服务器支持的事件类型有条件事件、简单事件和跟踪事件。

用户文本块 2 决定“条件事件”消息的来源。使用默认设置时，用户文本块 2 对应于故障位置。要以层级形式呈现消息，必须将这些消息合并到报警记录消息的用户定义组消息中。组消息的结构由 OPC A&E 中的区域决定。

在系统中触发操作员输入消息时，会发生跟踪事件。

下图显示的是 OPC 浏览器中条件事件的显示实例。除了“区域”和“来源”外，还显示了“条件”：



切换到带层级访问功能的 OPC A&E

请在创建新项目时使用带层级访问功能的 OPC A&E 服务器。

在现有项目中，OPC A&E 服务器可像以前那样使用，也可进行转换从而实现层级访问功能。可再次撤消转换而不会丢失任何数据。

1. 将“CcAeProvider.ini”文件复制到项目文件夹中。该文件位于 WinCC 安装路径的“OPC \AlarmEvent\Hierarchical-Access”文件夹中。
2. 更新客户端或对 OS 服务器执行完整下载。

8.9.5.3 在 OPC A&E 上映射 WinCC 消息系统

映射 WinCC 消息系统

简介

因进行组态而产生的 WinCC 消息系统定义过程中的哪个事件将会生成消息。此消息在 OPC A&E 中显示为事件通知。

在带层级访问功能的 OPC A&E 上映射 WinCC 消息系统

WinCC 用户文本块 2 的 OPC 源和 WinCC 用户文本块 1 的 OPC 消息在 WinCC 中用作映射 WinCC 消息系统的默认设置。

概述

下表显示最重要的事件通知属性及来自 WinCC 消息系统的相应信息。

使用已组态属性的事件显示在表的第三列中：

- “S”表示简单事件
- “C”表示条件事件
- “T”表示跟踪事件

OPC	WinCC 消息系统	事件类型
Area	组消息的结构决定 OPC A&E 中的区域。如果没有为消息组态组消息，则只有与服务器前缀对应的 OPC 区域可用。	S、C、T
Source	指示消息源。消息源的格式为“<服务器前缀>::Area\用户文本块 2”。本地计算机的服务器前缀为“@LOCALMACHINE”。服务器前缀始终显示服务器层级中的顶级 Areas。	S、C、T
Time	发布已接收、已发送和已确认消息的时间戳。以 UTC（协调世界时）格式发布时间戳。	S、C、T
Type	指示该事件是简单事件、跟踪事件还是条件事件。	S、C、T
Severity	返回消息的优先级。	S、C、T
EventCategory	指示消息类别。“事件类别”由“类别 ID”和“类别说明”组成。“类别 ID”对应于消息类别的内部 ID。“类别说明”对应于消息类别的名称。	S、C、T
Message	指示相应消息号的消息文本。	S、C、T
Condition	指示消息类型。	C

OPC	WinCC 消息系统	事件类型
Sub-condition	对应于“Condition”参数。	C
ChangeMask	指定条件变化。有关详细信息，请参阅“确认理论”。	C
NewState	指示条件的当前状态。有关详细信息，请参阅“确认理论”。	C
ConditionQuality	返回消息质量。有关详细信息，请参阅“质量代码”。	C
AckRequired	指示消息是否需要确认。	C
EventAttribute	列出相关消息所需的属性。有关详细信息，请参阅“WinCC 消息系统的属性”。	C
Quality	返回消息的质量代码。	C
Cookie	不包括客户机的任何可用信息	C
ActorID	指示确认消息的用户。	T

说明

如果将不带通配符的文本指定为该区域的过滤器，则只返回该区域的消息。如果想要包括位于该指定区域以外区域中的源，则需要使用通配符。

说明

如果如下所示运行 OPC A&E 服务器，则**必须**在所连接的 OS 服务器上对消息类别和消息类型进行完全相同的组态：

- 在 WinCC 客户机上
- 在连通站上

如果 OS 服务器未进行完全相同的组态，所使用的 OPC 客户机必须直接访问相应的 OS 服务器。

映射消息优先级

导言

消息优先级由 OPC A&E 服务器映射到属性“Severity”。

在消息传递系统中组态报警时，可以组态“0”和“16”之间的优先级。OPC A&E 规范为严重性定义的取值范围是“1”到“1000”。在这种情况下，“1”代表严重性最低，“1000”代表严重性最高。

因此，优先级的值相应显示为 OPC 严重性。在标准映射中，优先级“0”指定给 OPC 严重性“1”，优先级“16”指定给 OPC 严重性“1000”。所有其它优先级值在“0”与“1000”之间进行线性插值。

WinCC 消息系统的属性

简介

下表列出了 WinCC 消息系统的 OPC 属性。这些属性在 WinCC 消息系统中组态。某些属性仅供 WinCC 内部使用，与 OPC A&E 客户机无关。因此，这些属性未包含在该表中。

属性

OPC 属性	WinCC 消息系统	数据类型
CLASSNAME	输出消息类别名称。	VT_BSTR
TYPENAME	输出消息类型名称。	VT_BSTR
FORECOLOR	输出已激活、已禁用和已确认消息的文本颜色。	VT_I4
BACKCOLOR	输出已激活、已禁用和已确认消息的背景色。	VT_I4
FLASHCOLOR	输出闪烁颜色。	VT_I4
FLAGS	指示必须确认该消息	VT_I4
TEXT01	输出用户文本块 01 的内容。	VT_BSTR
TEXT02	输出用户文本块 02 的内容。	VT_BSTR
TEXT03	输出用户文本块 03 的内容。	VT_BSTR
TEXT04	输出用户文本块 04 的内容。	VT_BSTR
TEXT05	输出用户文本块 05 的内容。	VT_BSTR
TEXT06	输出用户文本块 06 的内容。	VT_BSTR
TEXT07	输出用户文本块 07 的内容。	VT_BSTR
TEXT08	输出用户文本块 08 的内容。	VT_BSTR
TEXT09	输出用户文本块 09 的内容。	VT_BSTR
TEXT10	输出用户文本块 10 的内容。	VT_BSTR
PROCESSVALUE01	输出过程值块 01 的内容。	VT_VARIANT
PROCESSVALUE02	输出过程值块 02 的内容。	VT_VARIANT
PROCESSVALUE03	输出过程值块 03 的内容。	VT_VARIANT
PROCESSVALUE04	输出过程值块 04 的内容。	VT_VARIANT
PROCESSVALUE05	输出过程值块 05 的内容。	VT_VARIANT
PROCESSVALUE06	输出过程值块 06 的内容。	VT_VARIANT
PROCESSVALUE07	输出过程值块 07 的内容。	VT_VARIANT
PROCESSVALUE08	输出过程值块 08 的内容。	VT_VARIANT

8.9 WinCC OPC A&E 服务器

OPC 属性	WinCC 消息系统	数据类型
PROCESSVALUE09	输出过程值块 09 的内容。	VT_VARIANT
PROCESSVALUE10	输出过程值块 10 的内容。	VT_VARIANT
STATETEXT	输出状态消息。	VT_BSTR
INFOTEXT	输出消息信息文本。	VT_BSTR
LOOPINALARM	指示是否组态报警回路。	VT_I4
CLASSID	输出消息类别 ID。	VT_I4
TYPEID	输出消息类型 ID。	VT_I4
MODIFYSTATE	输出消息的状态变量值。	VT_I4
AGNR	输出产生消息的 AS 的编号。	VT_I2
CPUNR	输出产生消息的 CPU 的编号。	VT_I2
DURATION	输出激活、禁用和确认消息之间的时间间隔。	VT_I4
COUNTER	输出自运行系统启动以来的消息数。	VT_I4
QUITSTATETEXT	指示消息是否被确认。	VT_BSTR
QUITCOUNT	输出未确认的活动消息数。	VT_I4
PARAMETER	输出消息参数。（消息组态画面）	VT_BSTR
BLOCKINFO	输出消息块的当前内容。	VT_BSTR
ALARMCOUNT	输出未决消息数。	VT_I4
LOCKCOUNT	输出锁定消息数。	VT_I4
PRIORITY	指示已组态的消息优先级。	VT_I4
APPLICATION	输出触发了消息的应用程序。	VT_BSTR
COMPUTER	输出处理了消息的计算机的名称。	VT_BSTR
USER	输出处理了消息的用户的名称。	VT_BSTR
COMMENT	输出消息注释。	VT_BSTR
HIDDEN-COUNT	输出隐藏消息数。	VT_I4
BIG COUNTER	输出自运行系统启动以来的消息数。	VT_CY
OS-HIDDEN	输出消息的隐藏状态。	VT_BOOL
OS-EVENTID	输出消息的已组态消息 ID。	VT_I4

确认方法

引言

WinCC 中的确认策略是如何显示和处理消息从“进入”到“离开”的过程。在 OPC A&E 服务器上，该消息状态用“ChangeMask”和“NewState”参数来显示。

条件事件、简单事件和跟踪事件

来自系统的消息将作为有确认的条件事件发送到客户机。

为了使消息作为简单事件接受处理，消息的消息等级必须符合下列条件：

- “确认进入”未激活。
- “无离开状态的消息”激活。

在 WinCC 中，消息等级“系统，不需要确认”和消息类型“操作员输入消息”的消息作为跟踪事件传送。

说明

具有“系统，不需要确认”消息等级和“过程控制系统”消息类型的消息作为“系统消息”事件类别的简单事件进行传送。

ChangeMask

“ChangeMask”参数将记录消息状态改变的地址。

参数值：

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

“NewState”参数指示改变后的消息状态。

参数值:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

概述

WinCC	NewState	ChangeState
已接收的消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
有确认的离开消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
无确认的离开消息	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
已确认的消息（消息未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
已确认的消息(消息不再未决)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
锁定的消息	-----	OPC_CHANGE_ENABLED_STATE
解锁的消息	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
已确认的进入消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
有确认的进入、离开消息	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
无确认的进入、离开消息	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
由系统确认的消息（消息未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
由系统确认的消息（消息不再未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

WinCC	NewState	ChangeState
紧急确认的消息（消息未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
紧急确认的消息（消息不再未决）	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

说明

历史报警和事件不会被确认。OPC A&E 历史事件接口仅有读取访问权限。

8.9.5.4 OPC A&E 质量代码**简介**

质量代码用于判断消息的状态和质量。OPC 的质量代码在“数据访问自定义界面标准 3.00 版”规范的“6.8 OPC 质量标记”中有说明。

质量代码

代码	质量	状态
0xC0	OPC_GOOD	确定
0x40	OPC_UNCERTAIN	在不确定的情况下返回，例如：发生延迟确认（回执）时。
0x00	OPC_BAD	在与源的连接中断时返回。

8.9.6 读取归档消息

8.9.6.1 访问归档消息

简介

可以使用 OPC 客户机，通过 OPC A&E 服务器访问归档消息。支持使用以下两种方法来访问归档消息：

- 输出过去某个时间段的归档消息
- 输出过去某个时间段的归档消息，但不指定截止时间。输出归档消息后，其它所有新生成的消息会自动发送到 OPC 客户机。

说明

读取归档消息后，无法使用消息的已返回“ActiveTime”来确认该消息或跟踪消息转变。为确保这一点，OPC A&E 客户机必须检查带有额外标记“OPC_HAE_HISTORICAL_EVENTFLAG”的消息的“EventType”。归档消息的“ActiveTime”是不正确的。可以在“标识归档消息”下找到有关额外标记的信息。

查询“历史报警和事件”功能

除了标准过滤器外，WinCC 的扩展 OPC A&E 服务器还随附了以下过滤器：

滤波器	过滤器的值	描述
OPC_HAE_FILTER_BY_TIMEFRAME	0x80000000	与用于 OPC 历史数据访问的“ReadRaw”函数对应
OPC_HAE_FILTER_BY_STARTTIME	0x40000000	与用于 OPC 历史数据访问的“AdviseRaw”函数对应

源过滤器和历史报警请求

要请求归档消息，OPC 客户机必须支持“SetFilter”的订阅功能。如果将关键字“OPCHAEServer”也插入到订阅的“源过滤器”序列中，则 OPC 服务器还会发送归档消息。除此关键字外，还可使用其它参数定义要读取的消息：

- 方法
- 时间段
- 有限制或没有限制

在过滤器中指定的源列表除了“OPCHAEServer”源以外，还可以包括其它源名称。在此情况下，订阅将仅传送给定来源的历史事件。源名的顺序不连贯。

组态源过滤器后，所选时间段可从客户机上通过“Refresh”调用进行调用。

8.9.6.2 使用 OPC 访问归档消息的语法

语法

```
OPCHAEServer hMode=(read|advise) htStartTime=szTime  
[htEndTime=szTime] [bBounds=(TRUE|FALSE)]
```

参数

hMode = [read|advise]

该参数是必需的。定义如何读取归档消息和事件。

Read: 输出过往定期的归档消息和事件（如果是 OPC 历史数据访问，则可与 ReadRaw 相比）。

下面是设置用于在最后 30 分钟进行读取的过滤器的实例：

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

Advise: 输出特定时间段的归档消息和事件。接收所有归档消息后，将按照激活订阅的情况下所采用的方式发送新消息（相当于 OPC 历史数据访问情况下的 AdviseRaw）。

在下列实例中，将读取最后 30 分钟的消息（订阅必须激活）：

```
OPCHAEServer hMode=advise htStartTime=NOW-30M
```

说明

参数“htStartTime”和“htEndTime”支持以下符号：

- 相对符号，例如 NOW
- 符号值，例如 NOW、YEAR、MONTH
- 根据 XML 符号指定绝对 UTC 数据/时间值：2006-09-01T10:00:00.000Z

使用符号表示法对应于 OPC 历史数据访问语法。

htStartTime =

该参数是必需的。定义从归档中读取消息和事件的开始时间。

htEndTime =

该参数是可选的。定义从归档中读取消息和事件的结束时间。如果“hMode = read”，使用缺省设置“NOW”。

bBounds = [TRUE|FALSE]

该参数是可选的。定义如何处理接近开始时间和结束时间的消息。此函数相当于 OPC 历史数据访问。

bBounds=FALSE:

- 传送的第一个消息的时间标志 \geq htStartTime
- 传送的最后一个消息的时间标志 \geq htEndTime

bBounds=TRUE:

- 传送的第一个消息的时间标志 \leq htStartTime
- 传送的最后一个消息的时间标志 \geq hEndTime

缺省设置为 FALSE。

8.9.6.3 归档消息的读方法

简介

可以使用以下两种读模式之一来读取归档消息:

- 读取
- 建议

“读取”读模式

使用“读取”模式读取过去某个已定义时间段内的归档消息。依据从中读取报警的各个 OS 服务器，始终按时间顺序读取消息。通过设置开始和结束时间，可以指定是最先还是最后输出最早的一条消息。如果开始时间早于结束时间，将最后输出最早的一条消息。

如果要使用“读取”模式，需要对订阅运行下列功能:

1. SetFilter
2. 刷新

具有刷新标识符的事件包仅含有历史事件。这些事件还可以按队列排列。

历史消息的最后一个刷新包含有“最后刷新”标识符。

将拒绝在“刷新”期间“SetFilter”。如果在“刷新”期间激活订阅，则不会影响刷新过程。将继续以刷新标识符传送历史事件。

根据激活订阅的标准行为来传送最近生成的事件：

- 考虑除“历史”源“OPCHAEServer”外的设置过滤器值
- 无刷新标识符

这会使客户端根据刷新标识符区分接收的事件。事件包从不同时包含历史事件和新事件。

- 具有刷新标识符的事件包仅含有历史事件。这些事件还可以按队列排列。
- 无刷新标识符的事件包仅包含新生成的事件。

“建议”读模式

使用“建议”模式读取从过去某个已定义时间段开始的归档消息。读取所有归档消息后，将如同订阅已激活的情况一样，发送新消息。将根据每台 OS 服务器，按时间顺序传送归档消息：先传送从开始时间开始的归档消息。然后传送新归档的消息。

请注意，对于“建议”模式不得指定结束时间。

激活的订阅将用于“建议”模式。如果对激活的订阅运行“SetFilter”功能，则会立即传送历史报警。

如果对未激活的订阅运行“SetFilter”功能，则只会在激活订阅后传送归档消息。如果想要对未激活的订阅使用“建议”模式，请按如下步骤操作：

1. SetFilter
2. 使用 SetState 将订阅设置为激活

如果将订阅设置为“未激活”，将结束传送。订阅激活时，将拒绝“SetFilter”。

对于“建议”模式下激活的“历史”订阅，“刷新”的工作方式与标准订阅情况下的方式相同：

所有排队等候的条件相关事件都将传送到具有刷新标识符的包中。

在“建议”模式下读取历史报警时，“刷新”调用没有任何影响。

8.9.6.4 识别归档消息

一般步骤

使用 EventType 中的附加标记来区分归档消息。该标记通过 OR 链接来链接到有效的 EventType。

名称	EventType	EventType (归档消息)
OPC_SIMPLE_EVENT	0x01	0x81
OPC_CONDITION_EVENT	0x04	0x84
OPC_TRACKING_EVENT	0x02	0x82
OPC_HAE_HISTORICAL_EVENTFLAG		0x80

实例

实例 1

下列源过滤器用于以“读取”模式输出前 30 分钟的归档消息和事件。每台 OS 服务器日期最早的消息将第一个输出。也会发送下限值。

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

实例 2

下列源过滤器用于以“读”模式输出 2006 年 9 月 1 日 10:00 至 12:00 的归档事件。每台 OS 服务器的最新消息将第一个输出。也会发送此时间段的极限。

```
OPCHAEServer hMode=read htStartTime=2006-09-01T12:00:00.000Z
htEndTime=2006-09-01T10:00:00.000Z bBounds=TRUE
```

实例 3

下列源过滤器用于以“建议”模式输出前 30 分钟的归档消息和事件。读取归档消息后，将按照操作已激活订阅的方式发送新近生成的消息。

```
OPCHAEServer hmode=advise htStartTime=NOW-30M
```

8.10 WinCC OPC UA 服务器

8.10.1 WinCC OPC UA 服务器的工作原理

工作原理

WinCC OPC UA 服务器提供以下值：

- 过程值
- 变量归档的值
- WinCC 消息

WinCC OPC UA 服务器作为 Windows 服务安装，并自动启动。WinCC OPC UA 服务器仅支持“UA-TCP UA-SC UA Binary”通信配置文件。所使用的端口号可调整。

支持的规范

OPC Unified Architecture 是用于传送过程值、归档数据和消息的规范。WinCC OPC UA 服务器支持 OPC UA 规范 1.03。有关支持的 UA 功能的详细信息，请参见“支持的 OPC UA 服务和配置文件 (页 720)”。

安装

安装 WinCC 后，无需进一步组态，即可立即使用 WinCC OPC UA 服务器。
WinCC OPC UA 服务器可在 WinCC 服务器或 WinCC 客户端上使用。

WinCC OPC UA 服务器的 URL

通过以下 URL 访问 WinCC OPC UA 服务器：

- "opc.tcp://[HostName]:[Port]"

参数	说明
HostName	计算机名称的占位符。自动使用。
Port	端口号。默认设置为“4862”。

Discovery 服务器

“Discovery 服务器”由 OPC 基金会提供。默认情况下，“Discovery 服务器”作为 Windows 服务安装在 HMI 设备上。

在“Discovery 服务器”上，可通过 OPC UA 服务器获得在“Discovery 服务器”上注册的 UA 客户端信息。

根据组态，当运行系统启动时，WinCC OPC UA 服务器在一个或多个已组态的可用“Discovery 服务器”上注册或不在任何服务器上注册。之后，会周期性重复注册。如果结束运行系统，WinCC OPC UA 服务器会自动从“Discovery 服务器”注销。

WinCC 地址区域中支持的语言

WinCC OPC A&E 服务器支持以下几种语言的 WinCC 地址区域：

- 德语
- 英语
- 法语
- 意大利语
- 西班牙语

8.10.2 OPC UA 的安全概念

简介

OPC UA 安全概念主要基于：

- 相关应用程序和用户的验证和授权
- 确保应用程序之间交换的消息的完整性和保密性

证书是用于验证 OPC UA 应用程序的方法。

每个应用程序都有自己的实例证书，通过该证书在公共密钥基础设施中标识自身。实例证书也称为“应用程序证书”。

WinCC OPC UA 服务器的证书

为确保安全运行，每个 WinCC OPC UA 服务器都需要带有私钥的自有证书 - 服务器证书。

证书仅在相应的计算机上有效，且只能由该计算机上安装的 WINCC OPC UA 服务器使用。

创建服务器的自签名证书并将其存储在服务器的证书文件夹中。

此服务器证书的私钥也存储在证书文件夹中。必须将带私钥的文件夹的访问权限限制给：

- 服务器自身
- 系统管理员

注意
访问带私钥的文件夹
出于安全原因，除了服务器和系统管理员以外，其它用户或应用程序均不能访问 WINCC OPC UA 服务器的私钥。

工厂管理员可替换安装期间生成的服务器证书和相应的私钥。

根据适用的系统安全概念，新的服务器证书可以自签名，也可由认证机构颁发。

WINCC OPC UA 服务器使用的证书是由“OpcUaServerWinCC.xml”组态文件中的设置决定：更多相关信息，请参见“WinCC OPC UA 服务器的组态文件 (页 733)”。

服务器证书的存储

“WinCC OPC UA 服务器”应用程序存储在以下路径中：

存储路径	应用程序	组态文件
<安装目录>WinCC\opc\UAServer\	OpcUaServerWinCC.exe	OpcUaServerWinCC.xml

WinCC OPC UA 证书存储在 WinCC 安装路径的以下文件夹中：

WinCC OPC UA 服务器	证书	opc\UAServer\PKI\CA\certs
	私钥	opc\UAServer\PKI\CA\private

可在组态文件中更改存储位置。

受信任的客户端证书

WinCC OPC UA 服务器仅支持与受信任的客户端进行安全通信。客户端在以下情况下受到信任：

- 客户端具有有效的自签名证书，且该证书存储在 WinCC OPC UA 服务器的受信任证书存储器中时
- 或有效的客户端证书由认证机构颁发时。
来自认证机构的有效证书必须位于 WinCC OPC UA 服务器的受信任证书存储器中。这种情况下，只需要来自认证机构的证书。客户端证书不需要位于受信任证书的证书存储器中。

客户端证书存储

通过 WINCC OPC UA 服务器组态文件指定受信任证书的存储设置：

参数	含义
StoreType	证书存储类型。存储位置可以是“目录”或“Windows”。
StorePath	受信任客户端的证书存储在此文件夹下。

使用“目录”存储的组态示例

```
<TrustedCertificateStore|
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\Trusted</StorePath>
  <ValidationOptions />
</TrustedCertificateStore>
```

在此例中，WINCC OPC UA 服务器信任所有在“...PKI\TrustList\Certs”文件夹中存储服务器证书的客户端。

使用“Windows”存储的组态示例

```
<TrustedCertificateStore>
  <StoreType>Windows</StoreType>
  <StorePath>UA Applications</StorePath>
  <ValidationOptions />
</TrustedCertificateStore>
```

对于这种存储选项，客户端的证书必须位于操作系统证书存储器的“<本地计算机>\UA 应用程序”下。

验证客户端证书链所需的认证机构证书存储在认证机构的证书存储器中。在这里也使用 WINCC OPC UA 服务器组态文件指定存储设置：

参数	含义
StoreType	证书存储类型。存储位置可以是“目录”或“Windows”。
StorePath	受信任的认证机构的证书存储在此文件夹下。

说明

来自认证机构存储器的证书并非自动受到信任。

对于受信任的认证机构，其证书必须位于受信任证书存储器中。

使用“目录”存储的组态示例

```
<IssuerCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\CA</StorePath>
  <ValidationOptions />
</IssuerCertificateStore>
```

在此例中，受信任的认证机构的证书位于“...\PKI\CA\Certs”文件夹。

使用“Windows”存储的组态示例

```
<IssuerCertificateStore>
  <StoreType>Windows</StoreType>
  <ValidationOptions />
</IssuerCertificateStore>
```

“StorePath”参数不相关。来自认证机构的证书必须存储在 Windows 证书存储器中，以符合操作系统要求。

如果证书位于以下两个位置之一，则受到信任：

- <本地计算机>\受信任的根认证机构
- <本地计算机>\第三方根认证机构

说明

存储的重要信息

- 服务器证书的存储位置必须为“目录”。
- 受信任客户端证书的存储位置和来自认证机构的证书的存储位置必须具有相同的 StoreType，即，必须都为“目录”或“Windows”。

客户端证书不被接受

如果 UA 客户端访问 WINCC OPC UA 服务器时没有受信任的证书，WINCC OPC UA 服务器不会允许安全通信，并将该客户端证书复制到存放被拒绝证书的文件夹。

通过 WINCC OPC UA 服务器组态文件指定被拒绝证书的存储设置，例如

```
<RejectedCertificatesStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\OPCUA\rejected</StorePath>
</RejectedCertificatesStore>
```

说明

这里也仅支持 StoreType“目录”。

要启用与此客户端的安全通信，必须将被拒绝的证书移至受信任证书的证书存储器中。

参见

设置通过证书进行认证。(页 272)

WinCC OPC UA 服务器的组态文件(页 733)

8.10.3 组态安全机制

简介

在通信级别应确保：

- UA 应用程序真实性
- 所交换消息的保密性
- 所交换消息的完整性

所使用的安全机制（例如加密和签名算法）在标准化安全策略中进行定义。

WinCC OPC UA 服务器支持的安全策略通过“ServerConfiguration”和“SecuredApplication”中的服务器组态文件进行设置。

ServerConfiguration

“ServerConfiguration”下的 XML 元素“SecurityPolicies”包含用于服务器的所有可用的“Security Profile”和“Message Security Mode”组合的列表。

Security Profile	Message Security Mode	说明
http://opcfoundation.org/UA/SecurityPolicy#None	None	不安全通信
http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15	Sign 或 SignAndEncrypt	安全通信，已签名或已加密并签名的消息
http://opcfoundation.org/UA/SecurityPolicy#Basic256	Sign 或 SignAndEncrypt	安全通信，已签名或已加密并签名的消息
http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256 ¹⁾	Sign 或 SignAndEncrypt	安全通信，已签名或已加密并签名的消息

1) 使用安全策略“Basic256Sha256”的要求：具有签名算法“Sha256”且最小密钥长度 = 2048 的实例证书。

说明

确保安全通信

安全通信需要服务器和客户端服务器证书，以及正确组态的证书存储器。

具有最大功能范围的组态文件示例

```

<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfound">...</SecuredApplication>
  <ServerConfiguration>
    <SecurityPolicies>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</ProfileUri>
        <MessageSecurityModes>None</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
        <MessageSecurityModes>Sign</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
        <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
        <MessageSecurityModes>Sign</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
        <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
      </SecurityPolicy>
    </SecurityPolicies>
  </ServerConfiguration>
</OPCUA_Server_WinCCUA>

```

SecuredApplication

根据 OPC UA 规范，使用“SecuredApplication”下的“SecurityProfileUris”元素显式启用和禁用安全机制。

下图显示了一个禁用不安全通信的 SecuredApplication:

```
<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
    <ApplicationName>OPCUA Server for Simatic WinCC UA Runtime</ApplicationName>
    <ApplicationUri>urn:[HostName]:Siemens.Automation.WinCCUA.Rt</ApplicationUri>
    <ProductName>Simatic WinCC UA</ProductName>
    <ApplicationType>Server</ApplicationType>
    <BaseAddresses>...</BaseAddresses>
    <ApplicationCertificate>...</ApplicationCertificate>
    <TrustedCertificateStore>...</TrustedCertificateStore>
    <TrustedCertificates>...</TrustedCertificates>
    <IssuerCertificateStore>...</IssuerCertificateStore>
    <IssuerCertificates>...</IssuerCertificates>
    <RejectedCertificatesStore>...</RejectedCertificatesStore>
    <SecurityProfileUris>
      <SecurityProfile>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
        <Enabled>>true</Enabled>
      </SecurityProfile>
      <SecurityProfile>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
        <Enabled>>true</Enabled>
      </SecurityProfile>
    </SecurityProfileUris>
  </SecuredApplication>
```

因此，WinCC OPC UA 服务器支持运行系统中的两个安全策略“Basic128Rsa15”和“Basic256”。虽然具有“Message Security Modes Sign”和“SignAndEncrypt”，但不是不安全通信。

建立通信时，UA 客户端从此列表中选择所需的 Policy。

用户标识

除了通信级别的安全机制以外，WinCC OPC UA 服务器还支持使用 UserTokenPolicy “UserName”进行客户端应用程序的用户验证。

建立通信时，客户端应用程序必须提供有效的用户名和密码组合。WinCC OPC UA 服务器在操作系统的用户管理中验证该组合。

UserTokenPolicy 在 WINCC OPC UA 服务器的组态文件中进行设置。

```

<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfound">...</SecuredApplication>
  <ServerConfiguration>
  <SecurityPolicies>...</SecurityPolicies>
  <UserTokenPolicies>
    <UserTokenPolicy>
      <TokenType>Anonymous</TokenType>
    </UserTokenPolicy>
    <UserTokenPolicy>
      <TokenType>UserName</TokenType>
    </UserTokenPolicy>
  </UserTokenPolicies>
  </ServerConfiguration>
</OPCUA_Server_WinCCUA>
    
```

通过该组态，WINCC OPC UA 服务器支持匿名用户和 Policy“UserName”。

8.10.4 支持的 OPC UA 服务和配置文件

OPC UA 服务

WinCC OPC A&E 服务器支持以下描述的功能。

下表概述了 OPC UA 服务器 1.0.9 支持的功能：

OPC UA Service Sets	Services	注释
Discovery Service Set	FindServers GetEndpoints	-
Secure Channel Service Session Service Set	全部	-
View Service Set	Browse BrowseNext RegisterNodes UnregisterNodes	确定映射的 WinCC 数据：过程值和归档数据
Attribute Service Set	Read Write HistoryRead HistoryUpdate*)	仅 WinCC 变量 仅 WinCC 变量 仅归档的变量 仅归档的变量

OPC UA Service Sets	Services	注释
Subscription Service Set	CreateSubscription SetPublishingMode Publish RePublish DeleteSubscription	
MonitoredItem Service Set	CreateMonitoredItems SetMonitoringMode DeleteMonitoredItems	仅 WinCC 变量的“Value”属性 访问 WinCC 消息时的 Event Notifier
Method Service Set	Call	Acknowledge ConditionRefresh
*): 有关限制的信息, 请参见“支持的写访问 (页 673)”		

OPC UA 配置文件和 Conformance Units

WinCC OPC UA 服务器支持以下 OPC UA 配置文件 1.03, 且没有限制:

- 6.5.5 基础服务器行为方面
- 6.5.16 标准事件订阅服务器方面
- 6.5.18 A & C 基础条件服务器方面
- 6.5.30 方法服务器方面
- 6.5.36 历史原始数据服务器方面
- 6.5.42 历史数据更新服务器方面
- 6.5.44 历史数据插入服务器方面
- 6.5.45 历史数据删除服务器方面
- 6.5.131 UA-TCP UA-SC UA 二进制
- 6.5.149 SecurityPolicy - Basic256
- 6.5.148 SecurityPolicy - Basic128Rsa15
- 6.5.147 SecurityPolicy - 无
- 6.5.150 SecurityPolicy - Basic256Sha256

8.10 WinCC OPC UA 服务器

WinCC OPC A&E 服务器支持下表中列出的 OPC UA 配置文件，但是有限制：

Profile	"Group"	不支持 "Conformance Unit"
6.5.2 Core Server Facet	Attribute Services	Attribute Write Index
6.5.11 Standard DataChange Subscription Server Facet	Monitored Item Services	DeadBand Filter
6.5.12 Enhanced DataChange Subscription Server Facet	Monitored Item Services	-
6.5.14 Data Access Server Facet	Data Access	Data Access Analog Data Access Multistate Data Access PercentDeadBand Data Access Semantic Changes Data Access Two State
6.5.55 Standard UA Server Profile	Attribute Services	Attribute Write StatusCode & Timestamp

8.10.5 WinCC OPC UA 服务器的名称区域

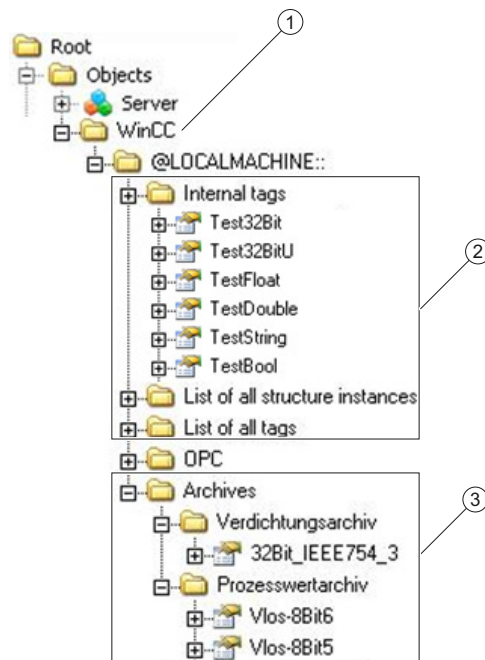
简介

WinCC OPC UA 服务器为 OPC UA 客户端提供了一个层级名称区域以及以下运行系统数据的访问权限：

- 过程值（WinCC 变量和 WinCC 变量组）
- 数据日志包含的记录变量
- WinCC 消息

WinCC OPC UA 服务器的名称区域附加在“Objects”默认文件夹中。

以下画面显示了本地 PC (“@LOCALMACHINE::”) 上激活的 WinCC 项目的 WinCC OPC UA 服务器的名称区域：



- ① WinCC 特定名称区域的起始节点。
- ② 显示 WinCC 变量；结构对应于 WinCC 中变量的结构。
- ③ 显示数据日志

WinCC 变量的显示

变量组、通信驱动程序和连接由“FolderType”类型的 OPC UA 对象显示。这些文件夹中的每一个都具有“Organizes”类型的对从属对象和变量的引用。

内部和外部 WinCC 变量由“DataItemType”类型的 OPC UA 变量显示。如果额外记录了 WinCC 变量，则显示的 OPC UA 变量额外具有“HasHistoricalConfiguration”类型的引用来进行日志组态。“Historizing”和“AccessLevel”属性分别设置。

下表显示了表示 WinCC 变量的 OPC UA 变量的最重要属性。在“§5.6”下的“OPC UA Part 3 - Address Space Model 1.03 Specification”文档中可以找到完整的属性列表：

属性	说明	注释
Nodetd	WinCC 变量的唯一标识	-
BrowseName	WinCC 变量名称	-
DisplayName	WinCC 变量名称	-
Value	变量值和状态	-

8.10 WinCC OPC UA 服务器

属性	说明	注释
Data Type	对应于 WinCC 变量类型的 OPC UA 数据类型，例如： <ul style="list-style-type: none"> • Int32；有符号的 32 位值 • UInt32；无符号的 32 位值 	-
AccessLevel	“CurrentRead” / “CurrentWrite” “HistoryRead” / “HistoryWrite”	对应于 WinCC 变量组态
ValueRank	始终为“Scalar”	-

写保护和读保护

可防止客户端访问 WinCC OPC UA 服务器变量。

在 WinCC 项目的“变量管理”(Tag Management) 中，可激活“选项”(Options) 组中变量属性区域中的以下设置：

属性	运行系统中的特性
OPC 写保护	客户端仅可对变量值进行读访问。
OPC 读保护	客户端无法对变量值进行读写操作。

无 WinCC 结构类型的映射

WinCC 结构无法作为 OPC UA 服务器的类型进行映射。

只能将 OPC UA 类型与 WinCC 结构变量链接。

记录变量的显示

过程值和压缩日志由“FolderType”类型的 OPC UA 对象显示。这些文件夹中的每一个都具有“Organizes”类型的对相关记录变量的引用。

过程值或压缩日志的记录变量由“BaseDateVariableType”类型的 OPC UA 变量显示。记录变量始终具有对日志组态的“HasHistoricalConfiguration”类型的引用。

下表显示了表示 WinCC 记录变量的 OPC UA 变量的最重要属性。在“§5.6”下的“OPC UA Part 3 - Address Space Model 1.03 Specification”文档中可以找到完整的属性列表：

属性	说明	注释
NodeId	记录变量的唯一标识	-
BrowseName	归档变量的名称	-
DisplayName	归档变量的名称	-
Description	节点说明	-

属性	说明	注释
Value	不可用	对于记录变量，无法读取和更改此属性。
DataType	对应于 WinCC 变量类型的 OPC UA 数据类型，例如： <ul style="list-style-type: none"> • Double: 64 位浮点数 • UInt32: 无符号的 32 位值 	-
AccessLevel	"HistoryRead" / "HistoryWrite"	-
ValueRank	始终为"Scalar"	-

访问 WinCC 消息

WinCC 命名空间的起始节点是一个 Event Notifier，它允许 OPC UA 客户端通过 Subscriptions 接收运行系统中 WinCC 消息的状态更改。

8.10.6 OPC UA 数据访问

内部和外部 WinCC 变量由"DataItemType"类型的 OPC UA 变量显示。其它 DataAccess 变量类型（如"AnalogItem"或"DiscreteType"）不受支持。

WinCC OPC A&E 服务器支持对 OPC UA 变量属性（如"DataType"或"AccessLevel"）进行读取访问。"Value"属性仅支持写访问和订阅。

8.10.7 OPC UA 历史访问

简介

"OPC Historical Access"允许访问归档，并包括"Historical Data"和"Historical Events"服务。WinCC OPC UA 服务器仅支持"Historical Data"服务。

WinCC OPC UA 服务器通过"Services"为 OPC 客户端提供对变量归档原始数据的访问。

- HistoryRead (READRAW)
- HistoryUpdate (INSERTDATA, REPLACEDATA, UPDATEDATA, DELETE_RAW)

使用 OPC UA 客户端可以读取变量归档中的归档变量值，也可以有限制地将归档变量值写入变量归档。根据变量归档的组态，归档变量可以包含原始数据或已处理的过程值。

归档变量的特性

WinCC 中的一个过程变量可以位于多个变量归档中。在这种情况下，过程变量链接到一个相应的归档变量。

归档组态的属性/Properties

下表显示了“HistoricalConfigurationType”类型的 OPC UA 变量组态的 Properties：在“Description”属性中，将显示 WinCC 中组态的归档变量注释。在“OPC UA Part 11 - Historical Access 1.03 Specification”文档的“§5.2.2”下可以找到完整的属性列表：

Property	说明/值	注释
Definition	WinCC 过程变量名称	对于过程值归档
Stepped	True	-

不支持以下可选 Properties：

- MaxTimeInterval
- MinTimeInterval
- ExceptionDeviation
- ExceptionDeviationFormat

Service“HistoryUpdate”的限制

只能对过程值归档使用 Service“HistoryUpdate”。

下表列出了 WinCC OPC UA 服务器支持的函数：支持哪些函数取决于 WinCC OPC UA 服务器的组态以及过程值归档组态。有关更多信息，请参见“§5.5”下的“OPC UA Part 11 - Historical Access 1.03 Specification”文档。

Service	函数	说明
HistoryUpdate	INSERTDATA	插入新归档值
	REPLACEDATA	替换现有归档值
	UPDATEDATA	替换插入归档值
	DELETE_RAW	删除归档值

8.10.8 OPC UA 报警和条件

简介

自 WinCC 7.3 起，可以通过 OPC UA 服务器对 WinCC 消息系统的消息进行访问。

OPC UA 服务器使用 WinCC-Event-Notifications 通过 Subscriptions 和 Monitored Event Items 将 WinCC 消息状态更改转发给 OPC UA 客户端，但不会在名称空间中保持 Condition 实例。

要使用的 Event Notifier 节点是 WinCC 名称区域的起始节点。

UA 客户端可以过滤消息，并定义返回的消息属性列表。

OPC UA 服务器支持“OPC UA Alarms & Conditions 1.03”规范。

以下部分概述了 WinCC 消息系统到 OPC UA 的映射。有关规范的详细信息，请参见“Part 9: Alarms and Conditions 1.03 Specification”。

WinCC 消息系统到 OPC UA 事件类型的映射

BaseEventType

“BaseEventType”是派生出 OPC UA Event 类型“WinCCEventType”和“WinCCAlarmConditionType”的基本类型。

说明

过滤器显示所有 WinCC 消息

当您过滤“BaseEventType”时，将收到所有 WinCC 消息。

WinCC 消息映射到以下 OPC UA 事件类型：

WinCCEventType

此类型基于“BaseEventType”，并使用以下确认原则映射“简单的 WinCC 消息”：

- “无离开状态的消息” 激活
- “确认进入” 未激活

此类型的消息的示例有启动和停止电机。

WinCCAlarmConditionType

此类型基于“AlarmConditionType”，映射所有无法映射到“WinCCEventType”的消息，例如可确认消息以及状态为“已到达”和“已离开”的消息。

8.10 WinCC OPC UA 服务器

对于“WinCCAlarmConditionType”类型的消息，该事件链接到条件。例如，只要超出变量限值，WinCC 就会生成一个消息。OPC UA 中的此消息等同于一个 Alarm Condition。

WinCC 消息属性

两个 Event 类型为基本类型增加了 WinCC 特定的消息属性。该属性作为 UA Event Properties 进行 1:1 映射，更多详细信息请参见“WinCC 消息系统的属性”。

消息类别和消息类型

WinCC 消息系统将过程中的干扰和操作情况通知给用户。WinCC 消息始终属于相应的 UA Events 的“CLASSID”、“TYPEID”、“CLASSNAME”和“TYPENAME”属性中指定的特定消息类别和消息类型。

优先级

在 WinCC 消息系统中组态消息时，可以组态“0”和“16”之间的优先级。OPC UA 规范为 Severity 定义的取值范围是“1”到“1000”。“1”代表最低严重性，“1000”代表最高 Severity。

因此，优先级的值必须相应映射到 OPC 严重性。在标准映射中，优先级“0”指定给 OPC-Severity “1”，而优先级“16”指定给 OPC-Severity “1000”。所有其它值在“0”与“1000”之间进行线性插值。

OPC UA 映射规则

组态 WinCC 消息系统期间，将进行设置以确定哪些过程事件产生消息。此消息通常在 OPC UA 中显示为一个 Event。

下表显示了 Events 的最重要 Properties，以及 WinCC 消息系统提供信息的方式。

OPC UA 属性	映射 WinCC 消息系统
对于所有事件类型：	
EventID	唯一的消息标识
EventType	事件类型：WinCCAlarmConditionType 节点或 WinCCEventType 节点的节点 ID
SourceNode	不相关
SourceName	指示消息源。下面详细说明了映射。
Message	相应消息编号的消息文本。
Time	事件的时间。时间戳以 UTC 格式给出
Severity	WinCC 消息的优先级

OPC UA 属性	映射 WinCC 消息系统
仅限于 WinCCAlarmConditionType:	
ConditionName	设置输出文本以及消息。文本输出取决于映射规则集： <ul style="list-style-type: none"> “模式 1”和“模式 2”：消息编号 “模式 3”：消息类别，例如“过程控制消息”
Quality	返回消息质量
ConditionClassId	“ProcessConditionClassType”节点的节点 ID
ConditionClassName	“ProcessConditionClassType”
Retain	“TRUE”，带未决消息
NodeId	ConditionId: 对 UA-Condition 进行唯一标识，例如报警。需要确认，即使不支持 Condition 实例
EnabledState	“TRUE”表示消息已启用
ActiveState/Id	“TRUE”表示消息已到达
AckedState/Id	“TRUE”表示消息已确认
ClientUserId	指示用户已登录

说明

OPC UA 服务器不支持以下 OPC UA Condition 和 Alarm Properties :

- BranchId
- LastSeverity
- InputNode
- ConfirmedState
- SuppressedState
- ShelvingState
- SuppressedOrShelved
- MaxTimeShelved

消息状态/确认状态

下表显示了 WinCC 消息状态到相应 WinCCAlarmConditionType - Properties 的映射:

消息状态	EnabledState/Id	ActiveState/Id	AckedState/Id
锁定的消息	FALSE	-	-
启用的消息	TRUE		
已接收的消息	TRUE	TRUE	FALSE

8.10 WinCC OPC UA 服务器

消息状态	EnabledState/Id	ActiveState/Id	AckedState/Id
带确认的已发送消息	TRUE	FALSE	TRUE
不带确认的已发送消息	TRUE	FALSE	FALSE
已确认的消息（消息未决）	TRUE	TRUE	TRUE
已确认的消息(消息不再未决)	TRUE	FALSE	TRUE
已接收的、确认的消息	TRUE	TRUE	TRUE
已接收的、带确认的已发送消息	TRUE	FALSE	TRUE
已接收的、不带确认的已发送消息	TRUE	FALSE	FALSE
由系统确认的消息（消息未决）	TRUE	TRUE	TRUE
由系统确认的消息（消息不再未决）	TRUE	FALSE	TRUE
紧急确认的消息（消息未决）	TRUE	TRUE	TRUE
紧急确认的消息（消息不再未决）	TRUE	FALSE	TRUE

WinCC 消息系统映射设置

就消息的 Properties“SourceName”和“Message”的映射而言，OPC UA 服务器的组态也适用于 OPC UA 服务器。

- 对于带层级访问的 OPC A&E 服务器：

SourceName	指示消息源。Source 的格式为“<服务器前缀>::Area\UserTextBlock 2”。本地计算机的服务器前缀为“@LOCALMACHINE”。
Message	返回相应消息编号的消息文本

- 对于不带层级访问的 OPC A&E 服务器：

SourceName	指示消息源。Source 的格式为“<服务器前缀>::localhost:”。本地计算机的服务器前缀为“@LOCALMACHINE”。
Message	返回相应消息编号的消息文本

报警组

在 WinCC 7.3 中，WinCC 报警组不显示在名称区域中。

支持的事件方法

确认

使用标准 OPC UA 信息模型中“`AcknowledgeableConditionType`”节点的“`Acknowledge`”方法确认 WinCC 消息。

只能确认“`WinCCAlarmConditionType`”类型的消息。

ConditionRefresh

使用标准 OPC UA 信息模型中“`ConditionType`”节点的“`ConditionRefresh`”方法建立仍未决的消息。

过滤器

OPC UA 客户端可以为 Monitored Event Items 定义过滤器。

但是，OPC UA 服务器不支持以下操作员：

- `FilterOperator_Cast`
- `FilterOperator_BitwiseAnd`
- `FilterOperator_BitwiseOr`
- `FilterOperator_RelatedTo`
- `FilterOperator_InView`

参见

WinCC 消息系统的属性 (页 731)

8.10.9 WinCC 消息系统的属性

概述

下表列出了 WinCC 消息系统的可组态属性。将属性 1:1 映射为 UA Event Properties 。

WinCC 消息属性	含义	数据类型
CLASSNAME	消息类别的名称	String
TYPENAME	消息类型的名称	String
FORECOLOR	进入消息、离开消息和已确认消息的前景色。	Int32
BACKCOLOR	进入消息、离开消息和已确认消息的背景色。	Int32
FLASHCOLOR	闪烁颜色	Int32
FLAGS	指示必须确认该消息	Int32
TEXT01...TEXT10	用户文本块 #1....#10 的内容	String
PROCESSVALUE01...PROCESSVALUE10	过程值块 #1....#10 的内容	
STATETEXT	状态消息	String
INFOTEXT	消息的信息文本	String
LOOPINALARM	指示是否已组态报警回路	Int32
CLASSID	消息类别 ID	Int32
TYPEID	消息类型 ID	Int32
MODIFYSTATE	消息状态变量的值	Int32
AGNR	输出生成消息的自动化系统的编号	Int16
CPUNR	输出生成消息的 CPU 的编号	Int16
DURATION	输出进入状态和离开状态与消息确认之间的时间段	Int32
COUNTER	自运行系统启动以来的消息数	Int32
QUITSTATETEXT	指示消息是否已确认	String
QUITCOUNT	处于打开状态的未确认消息数	Int32
PARAMETER	消息的组态参数	Int32
BLOCKINFO	消息块的当前内容	String
ALARMCOUNT	未决消息数	Int32
LOCKCOUNT	已锁定的消息数	Int32
PRIORITY	消息的优先级	Int32
APPLICATION	输出触发了消息的应用程序	String
COMPUTER	输出处理了消息的计算机的名称	String
USER	输出处理了消息的用户的名称	String
COMMENT	消息注释	String

WinCC 消息属性	含义	数据类型
HIDDEN-COUNT	隐藏的消息数	Int32
OS-HIDDEN	表示消息已隐藏	布尔型
OS_EVENTID	WinCC 消息数	Int32
BIG_COUNTER	消息计数器	Int64

参见

OPC UA 报警和条件 (页 727)

8.10.10 WinCC OPC UA 服务器的组态

8.10.10.1 WinCC OPC UA 服务器的组态文件

简介

WinCC OPC UA 服务器使用组态文件“OPCUAServerWinCC.xml”进行组态。

组态文件分为多个部分。本部分将介绍组态文件的布局。

“如何组态 OPC UA 服务器 (页 738)”一章介绍了如何组态 WinCC OPC UA 服务器。

组态文件的路径

WinCC OPC UA 服务器存在两个组态文件“OPCUAServerWinCC.xml”：

组态文件	存储路径
服务器特定组态文件	<WinCC 安装路径>\opc\UAServer\
项目特定的组态文件	<WinCC 项目文件夹>\OPCUAServer

编辑组态文件

您需要以下授权才能对组态文件进行更改：

服务器特定组态文件	Windows 管理员权限
项目特定的组态文件	用户必须是“SIMATIC HMI”用户组的成员。

说明

相同参数：文件的优先级

一些参数包含在两个组态文件中。

如果参数不匹配，则项目特定的组态文件的设置具有更高优先级。

结构：<SecuredApplication> 部分

在该部分中，根据 OPC UA 技术规范/第 6 部分/§“安全设置管理” 设置 OPC UA 应用程序安全性。

可在“OPC UA 的安全概念 (页 712)”下的 URL 上找到更多信息。

<SecuredApplication>	
<BaseAddresses> <...></...> </BaseAddresses>	WinCC OPC UA 服务器 URL 的组态。
<SecurityProfileUris> <SecurityProfile> <...></...> </SecurityProfile> ... </SecurityProfileUris>	支持的安全策略的组态 “none”设置仅用于测试和诊断目的
<ApplicationCertificate> <TrustedCertificateStore> > <TrustedCertificates> <...>	符合 OPC UA 技术规范/第 6 部分的默认证书组态的版本。 (可选) 这些参数仅包含在服务器特定的组态文件中。
</SecuredApplication>	

实例：OPC UA 应用程序安全性

```

<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
    <ApplicationName>OPCUA Server for Simatic WinCC UA Runtime</ApplicationName>
    <ApplicationUri>urn:[HostName]:Siemens.Automation.WinCCUA.Rt</ApplicationUri>
    <ProductName>Simatic WinCC UA</ProductName>
    <ApplicationType>Server</ApplicationType>
    <ApplicationCertificate>...</ApplicationCertificate>
    <TrustedCertificateStore>...</TrustedCertificateStore>
    <IssuerCertificateStore>...</IssuerCertificateStore>
    <RejectedCertificatesStore>...</RejectedCertificatesStore>
  </SecuredApplication>
</OPCUA_Server_WinCCUA>
  
```

结构：<ServerConfiguration> 部分

在此部分中设置服务器特定参数。

有关消息安全模式的详细信息，请参见“OPC UA 的安全概念 (页 712)”。

<pre><ServerConfiguration></pre>	
<pre><SecurityPolicies> <SecurityPolicy> <...></...> </SecurityPolicy> ... </SecurityPolicies></pre>	<p>消息安全模式的组态。 “none”设置仅用于测试和诊断目的</p>
<pre><UserTokenPolicies> <UserTokenPolicy> <...></...> </UserTokenPolicy> ... </UserTokenPolicies></pre>	<p>用户标识组态 “Anonymous”设置仅用于测试和诊断目的</p>

8.10 WinCC OPC UA 服务器

<pre><FastInsert> <Users> <...></...> </Users> <Clients> <...></...> </Clients> </FastInsert></pre>	优化的 WinCC 归档写访问的组态
<pre></ServerConfiguration></pre>	

结构: <CertificateDescriptor> 部分

可在 <ServerConfiguration> 部分的 <CertificateDescriptor> 标题下指定 WinCC OPC UA 服务器的证书参数。

这些参数仅包含在服务器特定的组态文件中。

可在“OPC UA 的安全概念 (页 712)”下的实例证书上找到更多信息。

<pre><ServerConfiguration> <CertificateDescriptor></pre>	
<pre><OrganizationUnit>...</...> <Organization>...</...> <Country>...</...></pre>	<p>描述性元素</p> <p>参数可以更改，且不会影响应用程序的功能。</p>
<pre><KeyLength>...</...></pre>	<p>用于创建证书的私钥的长度</p> <p>该长度取决于签名算法。</p> <ul style="list-style-type: none"> • 1024: 通过 OPC UA 进行安全通信的最小长度 • 2048: 使用 Sha256 时的最小长度¹⁾
<pre><SignatureAlgorithm>... </...></pre>	<p>用于签名证书的签名算法</p> <ul style="list-style-type: none"> • 可能值: Sha1、Sha224、Sha256、Sha384、Sha512 • 常用值: Sha1、Sha256 • 默认值: 私钥长度为 2048 的 Sha256¹⁾

<pre><LifetimeInMonths>...</ ...></pre>	<p>证书的有效期限按月计 在指定的时间到期后，将无法再使用此证书对服务器进行操作。</p> <ul style="list-style-type: none"> • 默认值：60
<pre></ CertificateDescriptor> </ServerConfiguration></pre>	

1) 要使用安全策略“Basic256Sha256”建立安全连接，服务器以及 OPC UA 客户端需要具有以下值的证书：

- KeyLength: 至少 2048
- SignatureAlgorithm: Sha256

实例：用于控制证书的参数

```
<ServerConfiguration>
  <CertificateDescriptor>
    <OrganizationUnit>DF PL DER HMI</OrganizationUnit>
    <Organization>Siemens AG</Organization>
    <Country>DE</Country>
    <KeyLength>2048</KeyLength>
    <SignatureAlgorithm>SHA256</SignatureAlgorithm>
    <LifetimeInMonths>60</LifetimeInMonths>
  </CertificateDescriptor>
```

更改服务器证书的存储路径

如有需要，WinCC OPC UA 服务器证书的存储位置可由工厂管理员进行调整。
只能在服务器特定的组态文件中更改这些参数。

参数	值	含义
StoreType	Directory	证书存储类型。 存储位置必须为“目录”。
StorePath	[ApplicationPath]\PKI \WINCC-OPC-UA-Server	证书和私钥存储在此文件夹下。

实例：服务器证书的存储路径

```
<ApplicationCertificate>  
  <StoreType>Directory</StoreType>  
  <StorePath>[ApplicationPath]\PKI\OPCUA</StorePath>  
  <SubjectName>OPCUA Server for Simatic WinCC UA Runtime</SubjectName>  
  <Thumbprint />  
</ApplicationCertificate>
```

新建服务器证书

用户需要具备管理员权限才能在 OPC UA 服务器上创建新证书。

1. 创建备份。
2. 删除相应文件夹中的现有证书和相关的私钥。
3. 在组态文件中，更新证书参数并保存 XML 文件。
4. 使用管理员权限在 Windows 中打开 DOS 窗口“cmd.exe”。
5. 要创建证书，请转到 OPC UA 应用程序的安装路径。
6. 输入以下调用：
 - OpcUaServerWinCC.exe /CreateCertificate
在存储路径中创建新证书和私钥。

8.10.10.2 如何组态 OPC UA 服务器**要求**

已创建 WinCC 项目。

打开组态文件

1. 打开 Windows 资源管理器。导航到目录“<WinCC 项目文件夹>OPCUAServer”。
2. 打开“OPCUAServerWinCC.xml”组态文件。有关详细信息，请参见“WinCC OPC UA 服务器的组态文件 (页 733)”。

更改 WinCC OPC UA 服务器的端口号

1. 如果需要，更改 <BaseAddresses> 下的端口号 4862。
不要使用已经分配给其他应用程序的端口号。
参数 [HostName] 为计算机名称的占位符，在运行期间确定。
示例：
<BaseAddresses>
<ua:String>opc.tcp://[HostName]:5210</ua:String>
<BaseAddresses>

指定安全设置

1. 指定通信的安全设置。有关详细信息，请参见“OPC UA 的安全概念 (页 712)”
2. 在 <SecurityProfileUri> 下，组态支持的“安全策略”。
 - 使用“true”启用设置。
 - 使用“false”禁用设置。

示例：

```
<SecurityProfile>
  <ProfileUri>http://opcfoundation.org/UA/
  SecurityPolicy#None</ProfileUri>
  <Enabled>>false</Enabled>
</SecurityProfile>
```

3. 在 <SecurityPolicies> 下，组态关联的“消息安全模式”。要禁用某一设置，请删除整个条目 <SecurityPolicy>... </Security Policy>。
示例：

```
<SecurityPolicy>
  <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</
  ProfileUri>
  <MessageSecurityModes>None</MessageSecurityModes>
</SecurityPolicy>
```

说明

客户端与服务器之间的不安全通信

“none”设置仅用于测试和诊断目的。

要在生产模式下实现安全的客户端/服务器通信，至少需要使用以下设置：

- SecurityPolicy::Basic128Rsa15
MessageSecurityMode::Sign
-

指定用户标识

1. 指定用于在 <UserTokenPolicies> 下设置连接的用户标识。有关详细信息，请参见“OPC UA 的安全概念 (页 712)”。

要禁用某一设置，请删除整个条目。

示例

```
<UserTokenPolicy>
  <TokenType>Anonymous</TokenType>
</UserTokenPolicy>
```

组态优化的 WinCC 归档写访问

1. 如果需要，在 <FastInsert> 下组态优化的 WinCC 归档写访问。
 - 设为 “true” 可以激活所有 OPC UA 客户端对 WinCC 归档的优化写访问。
 - 设为 “false” 可以设置特定 Windows 用户或 OPC UA 客户端对 WinCC 归档的优化写访问。

在 <Users> 下指定 Windows 用户。

在 <Clients> 下指定 OPC UA 客户端。将客户端证书中输入的“Common Name”用作 ClientName。

示例：

```
<EnabledByDefault>>false</EnabledByDefault>
<Users>
  <User>domain\user1</User>
</Users>
<Clients>
  <Client>ClientName1</Client>
</Clients>
```

8.11 诊断

跟踪文件

所有服务器都可激活诊断数据的输出，以用于测试和故障排除。

服务器的数据被写入跟踪文件。

设置

可在相应服务器的组态文件中指定诊断数据的输出。

有关更多信息，请参阅“SIMATIC 客户支持”。

REST 接口

9.1 WinCC 中的 REST 接口

WinCC REST 服务（代表状态传输）

WinCC REST 服务是用于监视端口的自托管服务。为了监视组态的 URL 和选定的端口，该服务使用 Microsoft C++ SDK。

通过 REST 通信，WinCC 可以访问运行系统数据和组态数据：

- 通过 REST 协议，您可以将运行系统数据以变量值的形式发送到云端。更多信息：
 - “接口 > WinCC/Cloud Connector > 通过 REST 将数据传输到云端 (页 777)”
- “WinCC 作为 REST”服务通过灵活的身份验证机制（例如 HTTPS）支持安全通信。
- “WinCC 作为 REST”服务允许外部应用程序访问 WinCC 数据。
外部应用程序可以通过 REST 接口读取和写入 WinCC 组态数据和变量值。

支持以下操作：

- 查询数据
- 编辑数据

要进行访问，请使用常规的 HTTP 方法和 JSON 资源显示格式。

限制：所支持的数据类型

访问变量时不支持以下数据类型：

- 原始数据变量
- 文本参考

API 结构

为了发送 WinCC 运行系统数据和组态数据，WinCC 使用 HTTP 请求和 HTTP 响应。

9.1 WinCC 中的 REST 接口

这些请求和响应包含以下元素：

Header	请求的元数据
Body	<ul style="list-style-type: none"> • 数据作为请求主体发送到 WinCC API • 数据作为响应主体从 WinCC API 发送
Service Endpoint	服务端点： 具有 WinCC API 服务的网络地址的基础 URL

HTTP 方法

方法	说明
GET	从服务器读取资源。 用于从 WinCC 项目中读取运行系统值和组态数据。
POST	从服务器读取资源。 寻址 Body 中的资源。 用于发送多个值。
PUT	写访问 通过 URL 寻址资源。 用于将单个 WinCC 值发送到云端。

错误代码

响应代码	说明	注释
200	OK	WinCC API 已成功处理请求。
400	Bad Request	请求无效
401	Unauthorized	无访问权限
402	Payment Required	未找到有效的 WinCC 许可证
404	Not Found	未找到资源 URL 与 WinCC API 的资源类型不匹配。 请更正 URL 的格式。

响应代码	说明	注释
416	RangeNotSatisfiable	请求无效 URL 结构不正确。无法满足请求的范围。 原因包括，在调用“GET”时缺少所寻址资源的名称。
423	Locked	资源被锁定。 原因包括，当前正在读取寻址的数据。 等待一段时间后再次发送请求。
500	InternalServerError	内部错误 服务器报告意外错误。
501	NotImplemented	请求无效 服务器不支持请求的功能。

参见

通过 REST 将数据传送到云端 (页 777)

方法概述 (页 748)

组态数据过滤器 (页 765)

WinCC Cloud Connector 中的 REST 设置 (页 786)

如何组态 WinCC REST 服务 (页 746)

9.2 如何组态 WinCC REST 服务

在“计算机属性”(Computer properties)对话框的“REST 设置”(REST settings)选项卡中组态 REST 服务器连接。

要通过 HTTPS 使用安全通信，请选择 REST 服务器以及其中安装的证书。

WinCC 用户和授权

变量管理：变量访问的授权

对于每个变量，可在“变量管理”中限制值的读写权限。

在变量属性的“WinCC 云”(WinCC Cloud)组中，在以下字段内选择所需授权：

- 读取权限
- 写入权限

用户管理器：分配权限

要通过 WinCC REST 服务进行访问，可在“用户管理器”中创建 WinCC 用户。

访问变量时，WinCC 用户需要具备在变量属性中定义的授权。

REST 设置

设置	说明
使用的主机名称	WinCC 服务器
本地端口	用于访问的端口号
端口证书	单击此按钮选择安装的证书。
WinCC REST 服务 URL	HTTPS 路径是由设置构成的。

操作步骤

1. 打开“计算机属性”(Computer properties)对话框。
2. 在“REST 设置”(REST settings)选项卡中,选中“通过 REST 发送变量值”(Send tag values via REST)选项。



3. 指定连接数据:
 - 使用的主机名称
 - 本地端口服务器的路径显示在“WinCC REST 服务 URL”(WinCC REST Service URL)字段中。
4. 要建立已通过验证的连接,请单击“端口证书”(Port certificate)按钮。“证书”(Certificates)对话框随即打开。
5. 选择证书,然后单击“确定”(OK)进行确认。
6. 单击“确定”(OK)关闭该对话框。
7. 如果应用程序“REST Service”尚未在计算机启动列表中激活,系统会提示用户将其激活。单击“是”(Yes)确认激活。如果该应用程序已被禁用,则不会在 WinCC Runtime 中执行 REST 服务。

参见

WinCC 中的 REST 接口 (页 743)

方法概述 (页 748)

9.3 方法概述

PUT: 支持的请求

- 将值写入到一个变量 (页 763)
- 将值写入到多个变量 (页 763)

GET / POST: 支持的请求

- 读取一个变量的运行系统值 (页 749)
- 读取所有变量的运行系统值 (页 749)
- 读取一个连接的组态数据 (页 750)
- 读取所有连接的组态数据 (页 751)
- 读取一个变量的组态数据 (页 752)
- 读取所有变量的组态数据 (页 753)
- 读取一个变量组的组态数据 (页 755)
- 读取所有变量组的组态数据 (页 755)
- 读取一个结构类型的组态数据 (页 757)
- 读取所有结构类型的组态数据 (页 757)
- 读取一个结构类型的实例 (页 760)
- 读取多个结构类型的实例 (页 761)

参见

WinCC 中的 REST 接口 (页 743)

组态数据过滤器 (页 765)

如何组态 WinCC REST 服务 (页 746)

9.4 GET / POST 方法

9.4.1 读取一个变量的运行系统值

说明

读取运行系统中过程变量或内部变量的变量值。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/TagManagement/Value/<Name of Tag>

本体

无

返回值：示例

```
{ "variableName": "Tag2",  
  "dataType": 2,  
  "value": "0",  
  "timestamp": "2020-03-20T09:12:38.5830Z",  
  "qualitycode": "0x4C - uncertain - initial value",  
  "errorCode": 0 }
```

9.4.2 读取所有变量的运行系统值

说明

读取运行系统中所有已注册过程变量和内部变量的变量值。

调用

GET / POST

9.4 GET / POST 方法

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/Values

本体

```
{
  "variableNames" : [ "Tag name 1", "Tag name 2", "Tag name 3",
"Tag name 4" ]
}
```

返回值：示例

```
[
  { "variableName": "Tag1",
    "dataType": 1,
    "value": "false",
    "timestamp": "2020-03-20T09:12:38.5830Z",
    "qualitycode": "0x4C - uncertain - initial value",
    "errorcode": 0},
  { "variableName": "Tag2",
    "dataType": 2,
    "value": "0",
    "timestamp": "2020-03-20T09:12:38.5830Z",
    "qualitycode": "0x4C - uncertain - initial value",
    "errorcode": 0},
  { "variableName": "Tag10",
    "dataType": 5,
    "value": "This variable contains text values",
    "timestamp": "2020-03-20T09:12:38.5830Z",
    "qualitycode": "0x4C - uncertain - initial value",
    "errorcode": 0},
  { "variableName": "Tag",
    "error": "Not Found" }
]
```

9.4.3 读取一个连接的组态数据

说明

读取在通信驱动程序下所创建连接的组态数据。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/Connection/
<ConnectionName>

本体

无

返回值：示例

```
{  
  "connectionName": "NewConnection_Name",  
  "channelUnit": "OMS+",  
  "channelName": "SIMATIC S7-1200, S7-1500 Channel",  
  "lastChange": "2020-01-30 15:08:31.000"  
}
```

9.4.4 读取所有连接的组态数据**说明**

读取在“变量管理”(Tag Management) 中通信驱动程序下所创建的所有连接的组态数据。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/Connections

本体

无

返回值：示例

```
{ "connections": [
  { "connectionName": "Internal Tag",
    "channelUnit": "Internal Tag",
    "channelName": "Internal Tag",
    "lastChange": "2020-01-30 15:01:30.297" },
  { "connectionName": "NewConnection_Name",
    "channelUnit": "OMS+",
    "channelName": "SIMATIC S7-1200, S7-1500 Channel",
    "lastChange": "2020-01-30 15:08:31.000" }
]}
```

替代寻址

URL

<https://<Host>:<Port>/WinCCRestService/tagManagement/Connections?itemLimit=20>

返回值：示例

```
{ "continuationPoint": 2,
  "moreAvailable": false,
  "connections": [
    { "connectionName": "Internal Tag",
      "channelUnit": "Internal Tag",
      "channelName": "Internal Tag",
      "lastChange": "2020-01-30 15:01:30.297" },
    { "connectionName": "NewConnection_Name",
      "channelUnit": "OMS+",
      "channelName": "SIMATIC S7-1200, S7-1500 Channel",
      "lastChange": "2020-01-30 15:08:31.000" }
  ]
}
```

9.4.5 读取一个变量的组态数据

说明

读取过程变量或内部变量的组态数据。

调用

GET / POST

Url

https://<Host>:<Port>/WinCCRestService/tagManagement/variable/
<Name of Tag>

本体

无

返回值：示例

```
{ "variableName": "Tag2",  
  "dataType": 2,  
  "comment": "comment of Tag2",  
  "canRead": true,  
  "canWrite": true,  
  "lastChange": "2020-03-03 13:48:29.000" }
```

9.4.6 读取所有变量的组态数据**说明**

读取所有已注册过程变量和内部变量的组态数据。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/variables

本体

无

返回值：示例

```
{ "variables":  
  [  
    { "variableName": "@SCRIPT_COUNT_TAGS",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"},  
    { "variableName": "@SCRIPT_COUNT_REQUESTS_IN_QUEUES",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"},  
    .... ]
```

替代寻址

URL

[https://<Host>:<Port>/WinCCRestService/tagManagement/variables?
itemLimit=2&continuationPoint=0](https://<Host>:<Port>/WinCCRestService/tagManagement/variables?itemLimit=2&continuationPoint=0)

返回值：示例

```
{ "continuationPoint": 3,  
  "moreAvailable": true,  
  "variables": [  
    { "variableName": "@SCRIPT_COUNT_TAGS",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"},  
    { "variableName": "@SCRIPT_COUNT_REQUESTS_IN_QUEUES",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"}  
  ] }
```

9.4.7 读取一个变量组的组态数据

说明

读取在通信驱动程序或内部组下所创建变量组的组态数据。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/Group/
<GroupName>

本体

None

返回值：示例

```
{
  "groupName": "ProcessHistorian",
  "connectionName": "Internal Tag",
  "lastChange": "2020-06-08 14:13:14.467"
}
```

9.4.8 读取所有变量组的组态数据

说明

读取在通信驱动程序或内部组下所创建变量组的组态数据。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/Groups

本体

无

返回值：示例

```
{ "groups": [
  { "groupName": "Script",
    "connectionName": "Internal Tag",
    "lastChange": "2020-06-08 14:13:14.463" },
  { "groupName": "TagLoggingRt",
    "connectionName": "Internal Tag",
    "lastChange": "2020-06-08 14:13:14.467" },
  { "groupName": "ProcessHistorian",
    "connectionName": "Internal Tag",
    "lastChange": "2020-06-08 14:13:14.467" },
  { "groupName": "Performance",
    "connectionName": "Internal Tag",
    "lastChange": "2020-06-08 14:13:14.470" },
  { "groupName": "Connection1_Group_1",
    "connectionName": "NewConnection_1",
    "lastChange": "2020-06-08 15:23:57" },
  { "groupName": "Connection2_Group_1",
    "connectionName": "NewConnection_2",
    "lastChange": "2020-06-08 15:24:39" }
]}
```

替代寻址

URL

<https://<Host>:<Port>/WinCCRestService/tagManagement/Connections?itemLimit=2&continuationPoint=0>

返回值：示例

```
{ "continuationPoint": 3, "moreAvailable": true,
  "groups": [
    {
      "groupName": "Script",
      "connectionName": "Internal Tag",
      "lastChange": "2020-06-08 14:13:14.463" },
    { "groupName": "TagLoggingRt",
      "connectionName": "Internal Tag",
      "lastChange": "2020-06-08 14:13:14.467" }
  ]
}
```

9.4.9 读取一个结构类型的组态数据

说明

读取在“结构变量”(Structure tags)下所创建的结构类型的组态数据。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement /
StructureType/<Name of Structure>

本体

无

返回值：示例

```
{ "typeName": "NewStructure_1",  
  "typeMembers": [  
    { "memberName": "NewElement_1",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:06.000" },  
    { "memberName": "NewElement_2",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:07.000" },  
    { "memberName": "NewElement_3",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:08.000" },  
    { "memberName": "NewElement_4",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:09.000" }  
  ] }
```

9.4.10 读取所有结构类型的组态数据

说明

读取在“结构变量”(Structure tags)下所创建的所有结构类型的组态数据。

9.4 GET / POST 方法

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/
StructureTypes

本体

无

返回值：示例

```
{ "structures":
[
{ "typeName": "NewStructure_1", "typeMembers": [
{ "memberName": "NewElement_1",
"dataType": 2,
"lastChange": "2020-03-19 08:46:06.000"},
{ "memberName": "NewElement_2",
"dataType": 2,
"lastChange": "2020-03-19 08:46:07.000"},
{ "memberName": "NewElement_3",
"dataType": 2,
"lastChange": "2020-03-19 08:46:08.000"},
{ "memberName": "NewElement_4",
"dataType": 2,
"lastChange": "2020-03-19 08:46:09.000"} ] },
{ "typeName": "NewStructure_2", "typeMembers": [
{ "memberName": "NewElement_1",
"dataType": 2,
"lastChange": "2020-03-19 08:46:12.000"},
{ "memberName": "NewElement_2",
"dataType": 2,
"lastChange": "2020-03-19 08:46:13.000"},
{ "memberName": "NewElement_3",
"dataType": 2,
"lastChange": "2020-03-19 08:46:14.000"} ] },
{ "typeName": "NewStructure_3", "typeMembers": [
{ "memberName": "NewElement_1",
"dataType": 2,
"lastChange": "2020-03-19 08:46:18.000"},
{ "memberName": "NewElement_2",
"dataType": 2,
"lastChange": "2020-03-19 08:46:20.000"},
{ "memberName": "NewElement_3",
"dataType": 2,
"lastChange": "2020-03-19 08:46:21.000"}
] }
] }
```

替代寻址

URL

```
https://<Host>:<Port>/WinCCRestService/tagManagement/  
StructureTypes?itemLimit=2&continuationPoint=0
```

返回值：示例

```
{
  "continuationPoint":1026,
  "moreAvailable":true,
  "structures":[
    {
      "typeName":"NewStructure_1",
      "typeMembers":[
        {
          "memberName":"NewElement_1",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:06.000"},
        {
          "memberName":"NewElement_2",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:07.000"},
        {
          "memberName":"NewElement_3",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:08.000"},
        {
          "memberName":"NewElement_4",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:09.000"}
      ]
    },
    {
      "typeName":"NewStructure_2",
      "typeMembers":[
        {
          "memberName":"NewElement_1",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:12.000"},
        {
          "memberName":"NewElement_2",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:13.000"},
        {
          "memberName":"NewElement_3",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:14.000"}
      ]
    }
  ]
}
```

9.4.11 读取一个结构类型的实例**说明**

读取在“结构变量”(Structure tags)下所创建的结构类型的结构变量元素。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/
StructureVariable/<Name of structure type>

本体

无

返回值：示例

```
{ "structureVariables":  
  [  
    { "variableName": "StructureInstance1_1",  
      "typeName": "NewStructure_1",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:15.000"},  
    { "variableName": "StructureInstance1_2",  
      "typeName": "NewStructure_1",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000"},  
    { "variableName": "StructureInstance1_3",  
      "typeName": "NewStructure_1",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000"}  
  ]  
}
```

9.4.12 读取多个结构类型的实例

说明

读取在“结构变量”(Structure tags)下所创建的结构类型的多个结构变量元素。

调用

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagManagement/
StructureVariables

本体

```
{  
  "typeNames" : [ "NewStructure_1", "NewStructure_2" ]  
}
```

返回值：示例

```
{ "structureVariables":  
  [  
    { "variableName": "StructureInstance1_1",  
      "typeName": "NewStructure_1",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:15.000" },  
    { "variableName": "StructureInstance1_2",  
      "typeName": "NewStructure_1",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000" },  
    { "variableName": "StructureInstance1_3",  
      "typeName": "NewStructure_1",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000" },  
    { "variableName": "StructureInstance2_1",  
      "typeName": "NewStructure_2",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:32.000" },  
    { "variableName": "StructureInstance2_2",  
      "typeName": "NewStructure_2",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:36.000" },  
    { "variableName": "StructureInstance2_3",  
      "typeName": "NewStructure_2",  
      "connectionName": "Internal Tag",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:36.000" }  
  ]  
}
```

9.5 PUT 方法

9.5.1 将值写入到一个变量

说明

向变量同步写入值。

调用

PUT

URL

https://<Host>:<Port>/WinCCRestService/TagManagement/Value/<Name of Tag>

本体

```
{"value":"text value"}
```

返回值：示例

```
{"variableName":"Tag11",  
"errorcode":0}
```

9.5.2 将值写入到多个变量

说明

向所列变量同步写入值。

指定要为每个变量写入的值。

调用

PUT

URL

https://<Host>:<Port>/WinCCRestService/TagManagement/Values

本体：示例

```
[  
  {"variableName":"Tag name 1","value":true},  
  {"variableName":"Tag name 2","value":6},  
  {"variableName":"Tag name 3","value":8},  
  {"variableName":"Tag name 4","value":10}  
]
```

返回值：示例

```
[  
  {"variableName":"Tag name 1","errorCode":0},  
  {"variableName":"Tag name 2","errorCode":0},  
  {"variableName":"Tag name 3","error":"Not Found"},  
  {"variableName":"Tag name 4","errorCode":0}  
]
```

9.6 组态数据过滤器

可通过过滤器界定请求。

这将减少数据传送量。

插入过滤器

过滤器将作为 URL 的一部分进行传送：

- <URL>?<过滤条件>

示例：

- https://<URL>/WinCCRestService/tagManagement/variables?canWrite

过滤器字符串不区分大小写。

链接

可同时使用多个不同过滤器。

对于逻辑运算，使用操作数"&"，例如：

- <URL>?<过滤条件_1>&<过滤条件_2>&<过滤条件_3>

连接的过滤器越多，对搜索结果的限制就越多。不允许使用 OR 逻辑运算。

每个过滤器参数只能在一个请求中使用一次。

无效过滤器组合的示例：

- variables?variableName=mot*&variableName=!motor

过滤条件：参数

参数	对象	说明
canRead	变量	返回已登录用户具有读取访问权限的变量的名称。
canWrite	变量	返回已登录用户具有写入访问权限的变量的名称。
variableName	变量	返回包含所搜索字符串的所有变量的名称。 可使用占位符"?"和"*"。
connection	变量：连接数据	提供在创建变量、结构变量或变量组时所用连接的相关信息。
structureType	结构变量	返回结构类型的所有结构类型元素的名称。
structureVariable	结构变量	返回结构类型的所有结构变量元素的名称。

9.6 组态数据过滤器

参数	对象	说明
group	变量组	返回变量组中所有变量的名称。
channel	通信通道	返回通道下所有连接的名称。
changed_After	组态数据	返回在指定日期后更改的组态数据。 日期格式: YYYY-MM-DD hh:mm:ss:ms 示例: 2015-12-31 16:30:00:000
itemLimit	组态数据	限制返回元素的数量。 这可提高大型 WinCC 项目的性能。
continuationPoint	组态数据	继续执行通过“itemLimit”限制的请求。 如果找到的元素数量大于通过“itemLimit”设置的值, 则将在达到最大值时停止输出。可通过再次请求继续输出, 直至返回所有已找到的元素。
* ?	通配符	只能在过滤变量名称 (variableName) 时使用通配符。 <ul style="list-style-type: none"> ? : 字符串中的任意单个字符 * : 字符串开头或末尾的任意数量的字符
!	操作数: 取反 (NOT)	对于每个请求只能使用一个 NOT 过滤器。
&	操作数: 逻辑运算 (AND)	可为每个请求使用多个 AND 逻辑运算。

示例: 通配符

请求名称中包含“Motor”的变量:

- /variables?variableName=*motor*

请求名称以“parfum”或“perfum”开头的变量:

- /variables?variableName=p?rfum*

例如: channel

请求通道“SIMATIC S7-1200, S7-1500 Channel”下的连接名称:

- /connections?channel=SIMATIC S7-1200, S7-1500 Channel

引用通信通道在 WinCC“变量管理”(Tag Management) 中显示的名称。

示例: itemLimit / continuationPoint

请求应返回已登录用户具有写入访问权限的所有变量。

通过“itemLimit”将请求限制为最多 90 个变量，但总共找到 250 个变量。

- /variables?canWrite&itemLimit=90

响应本体包含前 90 个变量。

要查询后续的 90 个变量，重新进行此查询并添加“continuationPoint”：

- /variables?canWrite&itemLimit=90&continuationPoint

响应本体包含后续的 90 个变量。

重复这个请求，查询最后 70 个变量。

例如: !(NOT)**changed_After: 在指定的时间之前**

要查询特定时间点及此前创建的所有连接，选择对“changed_After”取反：

- /connections?changed_After!=2015-11-26 10:30:02.000

connection: 排除连接名称

要请求以下变量的连接数据：

- 变量名称包含字符串“_S7_”
- 变量并非在连接“S7_1”下创建。

过滤器：

- /variables?Connection!=S7_1&variableName=*_S7_*

参见

WinCC 中的 REST 接口 (页 743)

方法概述 (页 748)

WinCC/Cloud Connector

10.1 WinCC/Cloud Connector

利用 WinCC/Cloud Connector，可将变量从 WinCC 站传送到云。

可以使用存储在云中的数据进行进一步分析或输出变量值，例如通过仪表盘。

云提供商

WinCC/Cloud Connector 支持以下提供商：

- Siemens MindSphere - MindConnect IoT Extension
- Amazon: AWS
- Microsoft: Azure
- Alibaba Cloud: Polardb, AnalyticDB（阿里云，中国）
- 通用 MQTT
- REST 协议

Cloud Connector 工作原理

“Cloud Connector”使用“MQTT”或“REST”协议传送变量值。

要提高通信安全性，请使用带认证握手的加密连接。

只能通过 Cloud Connector 发送数据。无法在 WinCC 中接收数据。

Message Queue Telemetry Transport (MQTT)

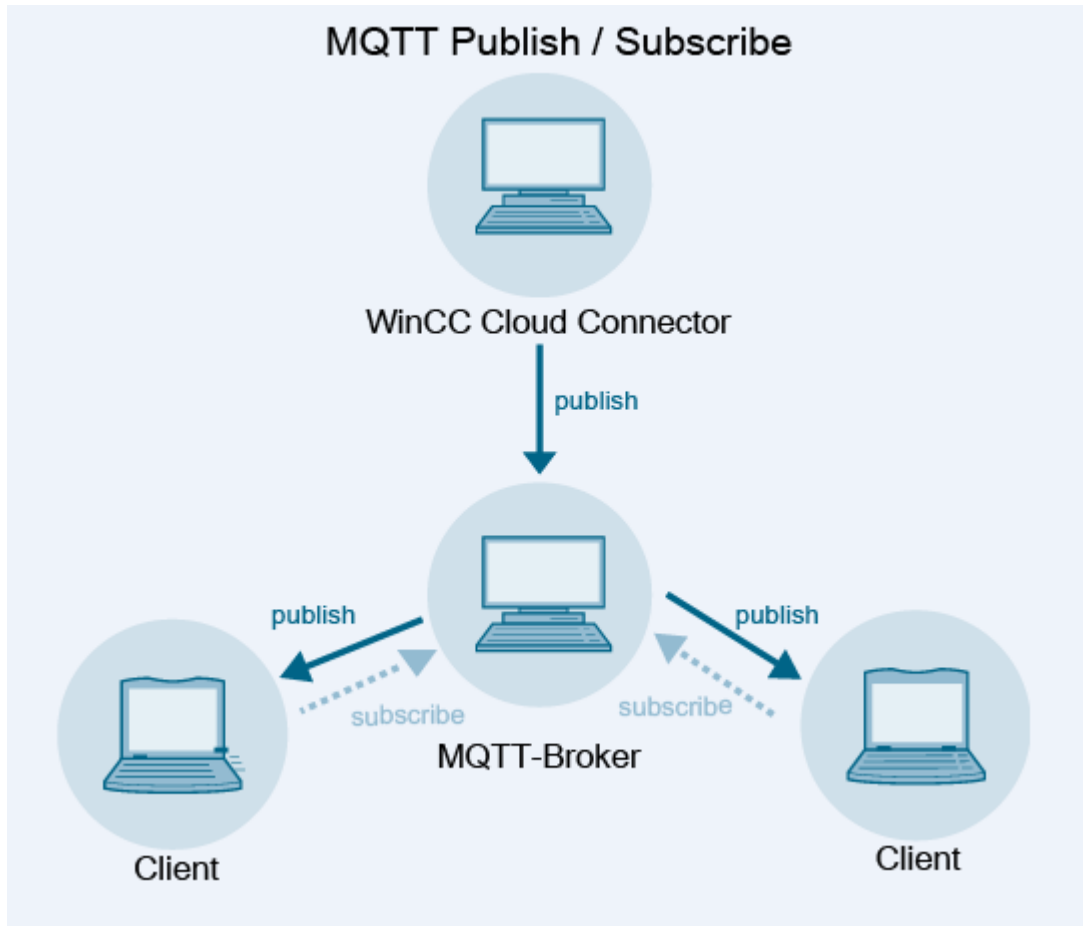
通过 MQTT 协议将中央服务器（MQTT 代理）用于数据传输。

发送和接收设备之间的数据交换仅通过 MQTT 代理进行。

更多关于 MQTT 协议的信息，请参见 Internet：

- <https://mqtt.org/> (<https://mqtt.org/>)

通过 MQTT 进行通信



Representational State Transfer (REST)

REST 协议使用 HTTP 或 HTTPS 在分布式客户端-服务器系统中进行通信。

通过 REST 进行数据交换

与 MQTT 相似，数据会发送至中央服务器。

可对任何提供商进行寻址。会在提供商处使用用户名和密码进行身份验证。

WinCC 作为 REST 服务

“WinCC 作为 REST”服务允许外部应用程序访问 WinCC 数据。

有关通过 HTTP 方法访问 WinCC 的 REST 接口的更多信息：

- “接口 > REST 接口 (页 743)”

参见

WinCC Cloud Connector 中的 REST 设置 (页 786)

WinCC 中的 REST 接口 (页 743)

MQTT 协议 (<https://mqtt.org/>)

10.2 Cloud Connector 许可

许可证

需要单独的 WinCC/Cloud Connector 许可证:

- SIMATIC WinCC Cloud Connect

如果没有许可证，则最多可以传送 5 个变量以进行测试。

10.3 通过 MQTT 将数据传送到云端

云提供商

使用 MQTT 协议通过 Cloud Connector 写入数据。

目前支持以下云提供商：

- Siemens MindSphere - MindConnect IoT Extension (MQTT)
- Amazon Web Services (MQTT)
- Microsoft Azure (MQTT)
- Alibaba Cloud: PolarDB, AnalyticDB (MQTT)

服务 CCCloudConnect

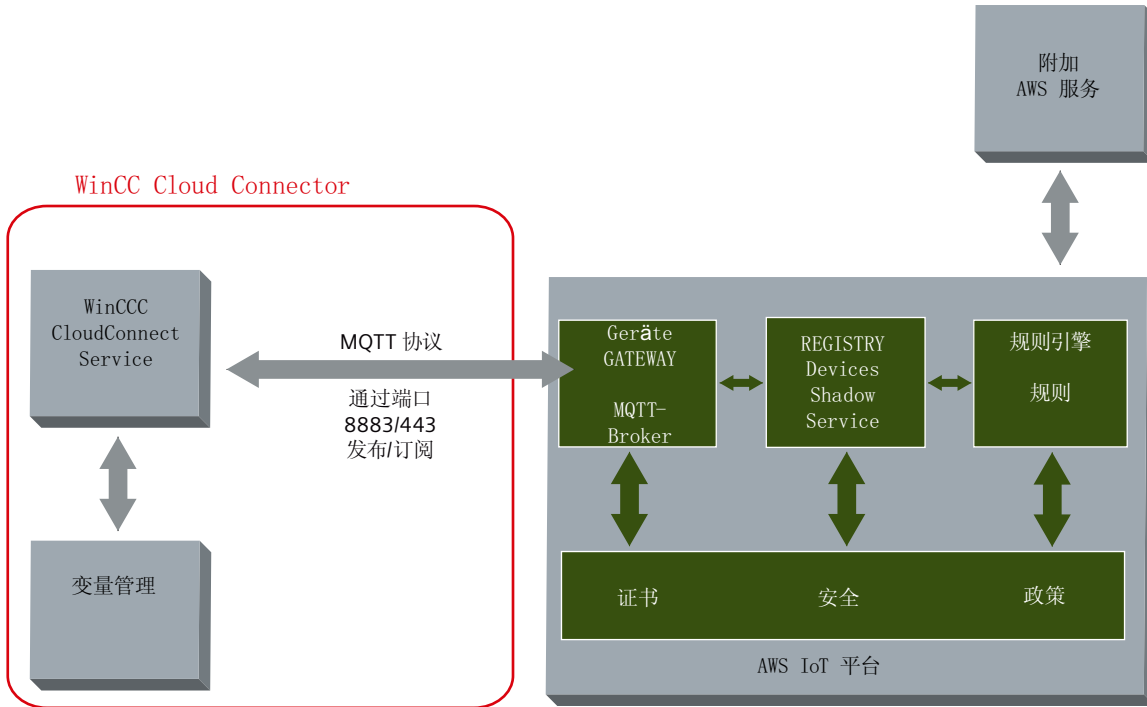
Windows 服务 CCCloudConnect 用于在 WinCC 项目和云系统之间建立连接。

CCCloudConnect 服务是一个 MQTT 客户端，可连接到云的 MQTT 代理，以通过标准端口 8883 或 443 发送数据。

在 WinCC 中，CCCloudConnect 服务会记录 WinCC 变量的数值更改。数值会被写入云端。如果 CCCloudConnect 从变量管理接收值更改，则该服务将创建消息。服务会将此消息传送给 MQTT 代理。

示例：WinCC/Cloud Connector 和 AWS-MQTT

下图显示了 Cloud Connector 和 AWS 平台之间的数据传输：



MQTT 主题

命名约定

将为每个被送至 MQTT 代理的变量创建单独的 MQTT 主题。

每个想要从代理接收此主题的客户都必须了解该主题。

WinCC 变量的命名约定为：

- <站名称>/<WinCC 项目名称>/<变量名称>

想要接收这些值的 MQTT 客户端必须使用适当的路径来订阅 MQTT 主题。

可在 Cloud Connector 设置中更改默认站名称“WinCC”。

MindSphere 中的 MQTT 主题

如果选择提供商“MindSphere - MindConnect IoT Extension”，则会应用 WinCC 中的变量名称。

会使用 MindSphere 中的设备名称作为站名称。

示例

已创建名称为“MyWinCCProject”的 WinCC 项目。

已在 WinCC 项目中为云激活了两个名为“MyTag1”和“MyTag2”的变量。

CCCloudConnect 为这些变量发送以下 MQTT 主题：

- WinCC/MyWinCCProject/MyTag1
- WinCC/MyWinCCProject/MyTag2

MindSphere 示例

已创建名称为“WinCCStation1”的设备。

首次在设备“WinCCStation1”上进行传输后，会显示变量“MyTag1”和“MyTag2”。

队列

通过 Cloud Connector 传输数据时，将根据队列原则发送消息。

最先添加的消息也将最先发送。

数据类型

大部分数据类型都允许进行传输。

例外

- 不支持结构化数据类型，例如 STRUCT 或 ARRAY。
- 其它限制取决于相应的云端提供商。

变量类型 “日期/时间”(Date/Time)

“日期/时间”变量类型的格式取决于使用的云。

MindSphere 中的变量类型

在 MindSphere 中，WinCC 变量不支持以下数据类型：

- 文本变量，8 位字符集
- 文本变量，16 位字符集
- 文本参考
- 日期/时间

时间戳

时间戳由 WinCC 站生成并发送至云。

云提供商为时间戳使用协调世界时 (UTC)。

10.3 通过 MQTT 将数据传送到云端

连接中断的数据缓冲区

为应对短暂的云连接中断而不会丢失数据，将创建一个数据缓冲区以通过 MQTT 进行通信。要发送的变量值临时存储在此数据缓冲区中。

- 最大大小：1 000 个变量

恢复连接时，数据缓冲区中的变量值将立即保存到云中。

禁用 WinCC Runtime

禁用 WinCC Runtime 也会关闭与云的连接。

发送到云的最后一消息存储在诊断文件“CCCloudConnect.log”中。

参见

WinCC Cloud Connector 中的 MQTT 设置 (页 781)

云连接诊断 (页 797)

WinCC 变量管理中的设置 (页 779)

10.4 通过 REST 将数据传送到云端

使用 REST 协议通过 Cloud Connector 写入数据。

Representational State Transfer (REST)

与通过 CCCloudConnect Windows 服务进行通信相反，连接是通过 HTTP 或 HTTPS 建立的。

可在变量值发生变化时读取变量值并将其写入到云端，也可定期读写。

数值以 HTTP 请求和 HTTP 响应的形式传送。

通过 REST 进行数据交换

WinCC 通过 REST 将数值发送到云端，而不是通过 MQTT。

与 MQTT 相似，数据会发送至中央服务器。

可对任何提供商进行寻址。

会在提供商处使用用户名和密码进行身份验证。

队列

通过 Cloud Connector 传输数据时，将根据队列原则发送消息。

最先添加的消息也将最先发送。

数据类型

大部分数据类型都允许进行传输。

例外

- 不支持结构化数据类型，例如 STRUCT 或 ARRAY。
- 其它限制取决于相应的云端提供商。

时间戳

时间戳由 WinCC 站生成并发送至云。

云提供商为时间戳使用协调世界时 (UTC)。

禁用 WinCC Runtime

禁用 WinCC Runtime 也会关闭与云的连接。

10.4 通过 REST 将数据传送到云端

发送到云的最后一消息存储在诊断文件“CCCloudConnect.log”中。

参见

WinCC 中的 REST 接口 (页 743)

10.5 WinCC 变量管理中的设置

在 WinCC 变量管理中，指定要传送的变量。

云提供商分别支持不同的变量类型。有关限制的信息，请参见“通过 MQTT 将数据传送到云端 (页 773)”。

启用云传送

要将变量值发送到云，请激活“变量管理”(Tag Management) 中的“WinCC 云”(WinCC Cloud) 变量属性。

选择所需采集周期。

只有使用 WinCC REST 服务时，授权的分配才有意义。

The screenshot shows the WinCC variable management interface. On the left, a table lists variables with their names, data types, and lengths. The variable 'var_text' is selected. On the right, the '属性 - 变量' (Properties - Variable) panel is open, showing various configuration options for the selected variable.

名称	数据类型	长度	
1	DB10_Value	32-位浮点数 IEEE 754	4
2	DB20_Value	32-位浮点数 IEEE 754	4
3	DB30_Value	无符号的 32 位值	4
4	DB40_Value	无符号的 32 位值	4
5	var_bool	二进制变量	1
6	var_byte	无符号的 8 位值	1
7	var_text	文本变量 8 位字符集	10
8			
9			
10			
11			
12			
13			
14			

属性 - 变量	
选择	
对象类型	变量
对象名称	var_text
常规	
分配	
线性标定	
限制值	
使用替换值	
选项	
各种	
结构变量元素	
WinCC 云	
WinCC 云	<input checked="" type="checkbox"/>
WinCC 云周期	5 s
读授权	改变画面
写授权	过程控制

设置周期时间

可以为每个将被传送至云的变量单独设置“WinCC 云端周期”(WinCC Cloud Cycle)。

该组态对应于 WinCC 变量记录中的周期时间设置。

- 如果不对周期时间进行任何设置，则默认设置周期时间为一分钟。
- 可行的最短周期时间为一秒。
- 从变量周期列表中为周期时间选择“启用值更改”(On value change) 或固定值：
 - 1 / 2 / 5 / 10 秒
 - 1 / 5 / 10 分钟
 - 1 小时

说明

ODK/VBA

周期时间不能通过 VBA 或 ODK 更改。

参见

通过 MQTT 将数据传送到云端 (页 773)

WinCC Cloud Connector 中的 MQTT 设置 (页 781)

云连接诊断 (页 797)

如何组态 MindSphere 连接 (页 792)

如何通过 REST 组态云连接 (页 795)

如何通过 MQTT 组态云连接 (页 788)

10.6 WinCC Cloud Connector 中的 MQTT 设置

简介

在“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration)对话框中，组态所用云的 URL 和访问设置。

要使用 MQTT 协议，请在“MQTT 设置”(MQTT Settings)选项卡中选择“通过 MQTT 将变量值发送到云”(Send tag values to cloud via MQTT)。

The screenshot shows the "WinCC Cloud Connector 组态" dialog box with the "MQTT 设置" tab selected. The "通过 MQTT 将变量值发送到云端" checkbox is checked. Under "提供程序设置", the "云提供商" dropdown is set to "Generic (MQTT)". The "代理地址(R)" field is empty, and the "代理端口(P)" field contains "8883". Under "设备设置", the "站名称(S)" field contains "WinCC". The "仅发送更改值" checkbox is unchecked. Under "安全性", there are three fields for "CA 证书(C)", "客户端证书", and "客户端密钥(L)", each with a browse button (...). A "测试连接" button is located at the bottom right.

云提供商

选择“通用 MQTT”(Generic MQTT) 或提供商。

10.6 WinCC Cloud Connector 中的 MQTT 设置

支持以下提供商：

- Siemens MindSphere - MindConnect IoT Extension (MQTT)
- Amazon Web Services (MQTT)
- Microsoft Azure (MQTT)
- Alibaba Cloud: PolarDB, AnalyticDB (MQTT)

应用示例

有关如何为各家云供应商组态数据连接的具体示例，请访问 Internet：

- 应用示例 109760955：“云中的 WinCC 数据连接 (<https://support.industry.siemens.com/cs/cn/zh/view/109760955>)”

代理地址

MQTT 云的终点。

MQTT 代理地址由云提供商提供。

代理端口

支持标准端口：

- 8883
- 443

站名

为客户端分配唯一的名称。

在变量传送期间，客户端的名称将用作 MQTT 主题的路径。

MindSphere 中的站名称

如果使用提供商“MindSphere - MindConnect IoT Extension (MQTT)”，站名会用作设备名称。

在 MindSphere 中注册设备时，名称会被指定为设备 ID。

仅发送已更改的值

如果选择此选项，则仅将更改的数据发送到云。

这样可减少数据传输工作量。

CA 证书

可以从云提供商处获取 CA 证书。“CA”代表“证书机构”。

将证书本地保存在 WinCC 站上。

WinCC 默认路径：

- \\Program Files (x86)\\Siemens\\WinCC\\CloudConnector\\Certificate

阿里巴巴证书

要访问阿里云，请使用提供商的自有证书。

AWS 证书

Amazon Web Services 使用 AWS IoT 生成的证书或具有 CA 认证的证书进行设备标识。

AWS IoT 证书由以下 CA 证书签发：

- RSA 2048 bit key: VeriSign Class 3 Public Primary G5 root CA certificate
- RSA 2048 bit key: Amazon Root CA 1
- RSA 4096 bit key: Amazon Root CA 2
- ECC 256 bit key: Amazon Root CA 3
- ECC 384 bit key: Amazon Root CA 4

要使用 AWS IoT 服务器证书验证设备，AWS 建议在 WinCC 站上安装所有五个证书。

Azure 证书

Microsoft Azure 云使用临时证书。

MindSphere 证书

WinCC 安装期间会安装 MindSphere 的 CA 证书。

如果选择提供商“MindSphere - MindConnect IoT Extension (MQTT)”，则会隐藏“安全”(Security) 区域。

会出现“MindSphere”区域，并会显示 MindSphere 证书。

客户端证书

在 WinCC 站上本地存储从云提供商处接收的证书。

10.6 WinCC Cloud Connector 中的 MQTT 设置

证书的文件扩展名为：

- .cer
- .crt
- .pem

在输入字段中输入证书的路径。

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Certificate

MindSphere 证书

如果选择提供商“MindSphere - MindConnect IoT Extension (MQTT)”，则会隐藏“安全”(Security) 区域。

客户端密钥

用户从云提供商处获取客户端密钥。，并将密钥本地存储在 WinCC 站上。

客户端密钥也称为“客户端/设备密钥”，其文件扩展名为“.key”。

在输入字段中输入客户端密钥的路径：

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Private

MindSphere 证书

如果选择提供商“MindSphere - MindConnect IoT Extension (MQTT)”，则会隐藏“安全”(Security) 区域。

注册/取消注册

仅当选择提供商“MindSphere - MindConnect IoT Extension (MQTT)”时，才会显示这些按钮。

“注册”(Register) 按钮会启动 MindSphere 中的设备注册过程。

成功注册后，该按钮会以灰色显示，“取消注册”(Unregister) 按钮会激活。

要更改 WinCC 站的设置，请单击“取消注册”(Unregister)。仍会在 MindSphere 中创建设备，且现有数据会保留下来。

用户名/密码

仅当选择提供商“MindSphere - MindConnect IoT Extension (MQTT)”时，才会显示这些字段。

用户名和密码是在 MindSphere 注册期间指定的，并会显示在 Cloud Connector 中。

在 MindSphere 中更改密码时，必须在 Cloud Connector 中应用新密码。

参见

通过 MQTT 将数据传送到云端 (页 773)

如何通过 MQTT 组态云连接 (页 788)

云连接诊断 (页 797)

WinCC 变量管理中的设置 (页 779)

如何组态 MindSphere 连接 (页 792)

如何通过 REST 组态云连接 (页 795)

WinCC Cloud Connector 中的 REST 设置 (页 786)

应用示例：与云的 WinCC 数据连接 (<https://support.industry.siemens.com/cs/cn/zh/view/109760955>)

10.7 WinCC Cloud Connector 中的 REST 设置

简介

在“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration)对话框中，组态所用云的 URL 和访问设置。

要使用 REST 协议，请在“REST 设置”(REST Settings)选项卡中选择“通过 REST 将变量值发送到云”(Send tag values to cloud via REST)。

提供商设置

输入提供商的连接数据：

设置	说明
服务地址	提供商的 HTTP 地址
维护端口	用于访问的端口号 默认情况下会设置 HTTP 端口“8080”。
服务路径	服务器目录路径
发送方法	传输方法： <ul style="list-style-type: none"> • PUT：发送一个变量的所有值 • POST：发送所有变量的变量值

仅发送已更改的值

如果选择此选项，则仅将更改的数据发送到云。

这样可减少数据传输工作量。

基本验证

在 REST 服务器中组态用于访问保护的用户名和密码。

在“用户名”(User name) 和“密码”(Password) 字段中输入 REST 服务器的访问数据。

参见

WinCC Cloud Connector 中的 MQTT 设置 (页 781)

WinCC/Cloud Connector (页 769)

如何通过 REST 组态云连接 (页 795)

WinCC 中的 REST 接口 (页 743)

10.8 如何通过 MQTT 组态云连接

简介

在“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration)对话框中指定所用云的 URL 和访问设置。

应用示例

有关如何为各家云供应商组态数据连接的具体示例，请访问 Internet：

- 应用示例 109760955：“云中的 WinCC 数据连接 (<https://support.industry.siemens.com/cs/cn/zh/view/109760955>)”

其它方法

如果选择“MindSphere (MindConnect IoT Extension)”作为云提供商，请遵循以下说明：

- 如何组态 MindSphere 连接 (页 792)

要通过 REST 组态云连接，请按以下说明操作

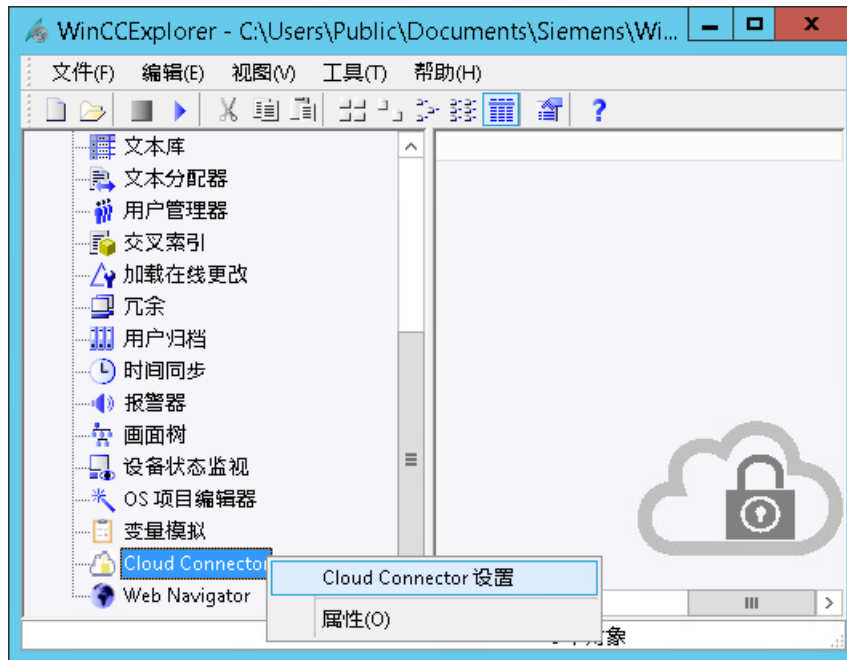
- 如何通过 REST 组态云连接 (页 795)

要求

- 在 WinCC 变量管理中为要传输的变量启用了“WinCC 云”(WinCC Cloud) 选项。

操作步骤

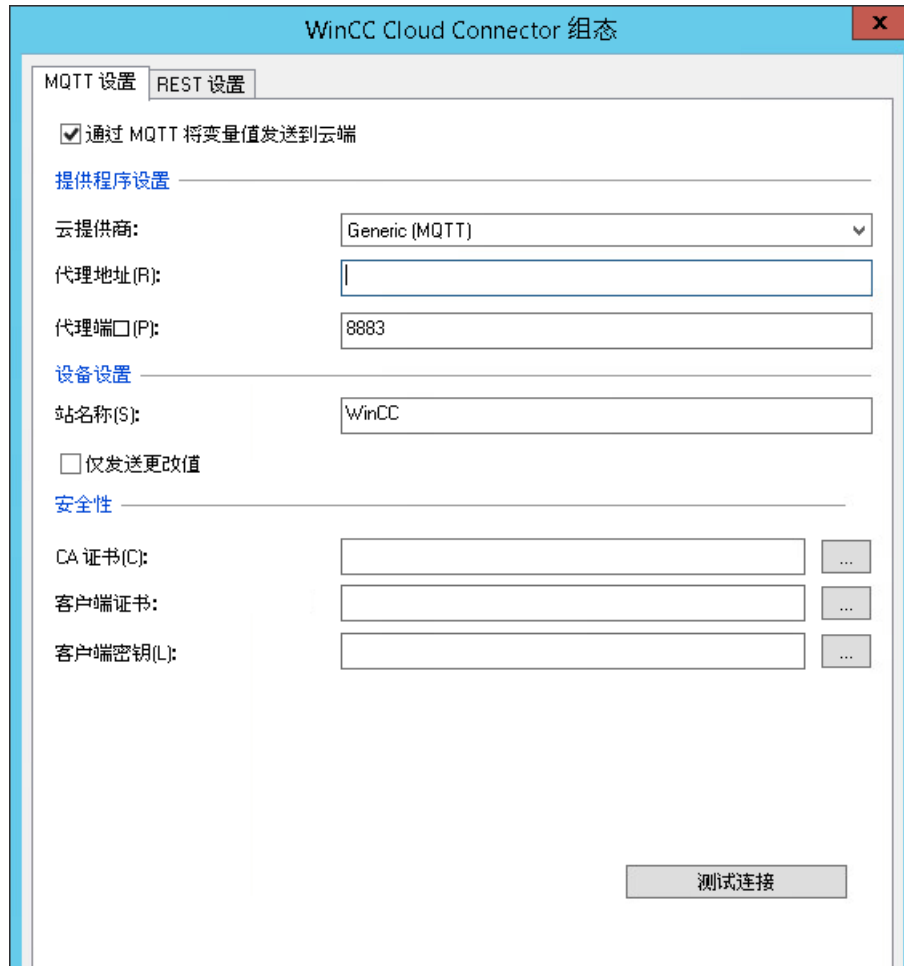
1. 打开 WinCC 项目管理器。
2. 在“Cloud Connector”的快捷菜单中，选择“Cloud Connector 设置”(Cloud Connector Settings) 条目。



“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration) 对话框将打开。

10.8 如何通过 MQTT 组态云连接

- 在“MQTT 设置”(MQTT Settings) 选项卡中，选择“通过 MQTT 将变量值发送到云”(Send tag values to cloud via MQTT) 选项。



- 指定连接数据：
 - 云提供商
 - 代理地址
 - 代理端口
 - 站名为了减少数据传输工作量，请选择“仅发送已更改的值”(Only send changed values) 选项。
- 选择证书和客户端密钥。
- 要测试连接设置，请单击“测试连接”(Test connection) 按钮。
- 单击“确定”(OK) 关闭该对话框。
- 如果应用程序“Cloud Connector”尚未在计算机启动列表中激活，系统会提示用户将其激活。单击“是”(Yes) 确认激活。如果该应用程序已被禁用，则不会在 WinCC Runtime 中执行 Cloud Connector 服务。

参见

WinCC Cloud Connector 中的 MQTT 设置 (页 781)

通过 MQTT 将数据传送到云端 (页 773)

如何组态 MindSphere 连接 (页 792)

如何通过 REST 组态云连接 (页 795)

WinCC 变量管理中的设置 (页 779)

应用示例：与云的 WinCC 数据连接 (<https://support.industry.siemens.com/cs/cn/zh/view/109760955>)

10.9 如何组态 MindSphere 连接

简介

在“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration)对话框中指定所用云的 URL 和访问设置。

与 MindSphere 连接的过程中，会使用随 WinCC 安装的 CA 证书。

为了进行额外验证，MindSphere 会创建用户名和密码。

应用示例

有关如何为各家云供应商组态数据连接的具体示例，请访问 Internet：

- 应用示例 109760955：“云中的 WinCC 数据连接 (<https://support.industry.siemens.com/cs/cn/zh/view/109760955>)”

要求

- 在 WinCC 变量管理中为要传输的变量启用了“WinCC 云”(WinCC Cloud) 选项。

操作步骤

1. 打开 WinCC 项目管理器。
2. 在“Cloud Connector”的快捷菜单中，选择“Cloud Connector 设置”(Cloud Connector Settings) 条目。
“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration) 对话框将打开。

- 在“MQTT 设置”(MQTT Settings) 选项卡中, 选择 “通过 MQTT 将变量值发送到云”(Send tag values to cloud via MQTT) 选项。

WinCC Cloud Connector 组态

MQTT 设置 REST 设置

通过 MQTT 将变量值发送到云端

提供程序设置

云提供商: MindSphere (MindConnect IoT Extension)

代理地址(R):

代理端口(P): 8883

设备设置

站名称(S): WinCC

仅发送更改值

MindSphere

CA 证书(C): Program Files (x86)\Siemens\WinCC\bin\CertRootCA2G3.crt ...

注册 WinCC 作为 MindConnect IoT Extension 的设备, 以建立连接。
注册 MindConnect 设备时, 站名对应于设备 ID。

注册

注册后无法再更改组态。

取消注册

用户名(U):

密码(A):

测试连接

- 指定连接数据:
 - 云提供商: “MindSphere (MindConnect IoT Extension)”
 - 代理地址
 - 站名
 默认端口 8883 无法更改。
 为了减少数据传输工作量, 请选择 “仅发送已更改的值”(Only send changed values) 选项。
- 在浏览器中打开 MindSphere 组态。
- 切换至“MindConnect IoT Extension”编辑器。
在 “设备”(Devices) 下选择设备注册。
- 要创建新设备, 请输入站名称。
Cloud Connector 中的站名称与“MindConnect IoT Extension”编辑器中的设备名称必须匹配。
 - WinCC 站作为设备创建。
 - 显示 “等待连接”(Waiting for connection) 状态。
- 转到“WinCC Cloud Connector 组态”(WinCC Cloud Connector configuration) 对话框并单击 “注册”(Register) 按钮。

10.9 如何组态 MindSphere 连接

9. 返回浏览器窗口中的“MindConnect IoT Extension”编辑器。
要完成注册，请单击“接受”(Accept)按钮。
 - 如果已接受注册，“注册”(Register)按钮会在 Cloud Connector 中以灰色显示。
 - “取消注册”(Unregister)按钮已激活。
 - 用户名和密码通过“MindConnect IoT Extension”编辑器设置，并显示在 Cloud Connector 中。
在“MindConnect IoT Extension”编辑器中更改密码时，必须在 Cloud Connector 中应用新密码。
10. 要测试连接设置，请单击 Cloud Connector 中的“测试连接”(Test connection)按钮。
11. 单击“确定”(OK)关闭对话框。
12. 如果应用程序“Cloud Connector”尚未在计算机启动列表中激活，系统会提示用户将其激活。
单击“是”(Yes)确认激活。
如果该应用程序已被禁用，则不会在 WinCC Runtime 中执行 Cloud Connector 服务。

参见

WinCC Cloud Connector 中的 MQTT 设置 (页 781)

WinCC 变量管理中的设置 (页 779)

如何通过 MQTT 组态云连接 (页 788)

如何通过 REST 组态云连接 (页 795)

应用示例：与云的 WinCC 数据连接 (<https://support.industry.siemens.com/cs/cn/zh/view/109760955>)

10.10 如何通过 REST 组态云连接

简介

可使用 REST 协议向 HTTP 提供商发送数据。

在“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration)对话框中指定所用云的 URL 和访问设置。

其它方法

要通过 MQTT 组态云连接，请按以下说明操作：

- 如何通过 MQTT 组态云连接 (页 788)

如果选择“MindSphere (MindConnect IoT Extension)”作为云提供商，请遵循以下说明：

- 如何组态 MindSphere 连接 (页 792)

要求

- 在 WinCC 变量管理中为要传输的变量启用了“WinCC 云”(WinCC Cloud) 选项。

操作步骤

1. 打开 WinCC 项目管理器。
2. 在“Cloud Connector”的快捷菜单中，选择“Cloud Connector 设置”(Cloud Connector Settings) 条目。
“WinCC Cloud Connector 组态”(WinCC Cloud Connector Configuration) 对话框将打开。

10.10 如何通过 REST 组态云连接

- 在“REST 设置”(REST Settings) 选项卡中，选择“通过 REST 将变量值发送到云”(Send tag values to cloud via REST) 选项。



- 指定连接数据：

- 服务地址
- 维护端口
- 服务路径
- 发送方法

为了减少数据传输工作量，请选择“仅发送已更改的值”(Only send changed values) 选项。

- 在“用户名”(User name) 和“密码”(Password) 字段中输入 REST 服务器的访问数据。如果在 REST 服务器中更改密码，则必须在 Cloud Connector 中应用新密码。
- 单击“确定”(OK) 关闭该对话框。
- 如果应用程序“Cloud Connector”尚未在计算机启动列表中激活，系统会提示用户将其激活。单击“是”(Yes) 确认激活。如果该应用程序已被禁用，则不会在 WinCC Runtime 中执行 Cloud Connector 服务。

参见

如何通过 MQTT 组态云连接 (页 788)

WinCC Cloud Connector 中的 MQTT 设置 (页 781)

WinCC 变量管理中的设置 (页 779)

如何组态 MindSphere 连接 (页 792)

WinCC Cloud Connector 中的 REST 设置 (页 786)

10.11 云连接诊断

简介

WinCC 支持通过性能变量以及日志文件中的消息输出进行诊断。

用于连接监视的性能变量

创建 WinCC 项目时，会在内部“Performance”变量组中创建系统变量。更多信息：

- “使用 WinCC > 使用项目 > 对运行系统进行设置 > 使用性能变量进行系统诊断”

可选择下列系统变量进行性能分析：

系统变量	说明
@PRF_CLDCN_RESET	复位变量将复位以下性能变量的值： <ul style="list-style-type: none"> • @PRF_CLDCN_TAG_FAILED_WRITES_TOTAL • @PRF_CLDCN_TAG_WRITES_TOTAL
@PRF_CLDCN_TAG_FAILED_WRITES_TOTAL	未通过云确认的已传送变量的数量
@PRF_CLDCN_TAG_WRITES_PER_SECOND	每秒已传送变量的数量
@PRF_CLDCN_TAG_WRITES_TOTAL	通过连接传送的变量总数

诊断文件

文件“CCCloudConnect.log”在 WinCC 安装路径的“Diagnostics”文件夹中创建。

Eclipse Mosquitto

Eclipse Mosquitto 错误代码在诊断文件中用“MOSQ”指定。

有关详细信息，请参见“libmosquitto API”文档：

- <https://mosquitto.org/> (<https://mosquitto.org/>)

参见

通过 MQTT 将数据传送到云端 (页 773)

WinCC Cloud Connector 中的 MQTT 设置 (页 781)

10.11 云连接诊断

WinCC 变量管理中的设置 (页 779)

Internet: <https://mosquitto.org/> (<https://mosquitto.org/>)

索引

“

- “MPI”通道单元
 - 组态连接, 397
- “SIMATIC S5 PROFIBUS FDL”通道, 600
 - 诊断选项, 588
- “SIMATIC S7 Protocol Suite”通道, 588
 - 诊断选项, 588
- “Slot PLC”通道单元, 403
 - 组态连接, 404
- “Soft PLC”通道单元, 405
 - 组态连接, 405
- “System Info”通道, 577
 - 诊断选项, 577
- “TCP/IP”通道单元, 407
 - 组态连接, 408
- “命名连接”通道单元, 398
 - 组态连接, 400

5

- 505 Ethernet (CP 1413-x)
 - 通道单元, 502

A

- A&E 服务器, 684, 685, 695, 697, 699
 - 层级访问, 697
 - 跟踪事件, 684, 695
 - 简单事件, 695
 - 条件事件, 695
 - 映射 WinCC 消息类别和消息类型, 699
 - 映射 WinCC 消息系统, 685, 699
 - 与条件相关的事件, 684
- AR_SENDAR_SEND-FunctioninWinCC, 421
- AS 符号, 384, 391, 482, 489
 - AS 结构, 389, 487
 - 导入, 384, 482
- AS 数据类型, 158
 - 类型转换, 170
- AS511, 360

C

- Cloud Connector, 769
- CP1434 TF, 511, 512

- CPU 负载, 539
 - 系统信息示例, 549

E

- ERROR 标记, 577, 589

I

- I/O 域组态, 650
 - OPC DA 的实例, 650
- I/O 字段
 - S5 Profibus FDL 的实例, 358
 - 系统信息示例, 547
- INFO 标记, 577, 589, 601, 610

M

- MELSEC FX3U 系列, (请参见 Mitsubishi 以太网)
- MELSEC iQ-R 系列, (请参见 Mitsubishi 以太网)
- MELSEC Q 系列, (请参见 Mitsubishi 以太网)
- Mitsubishi 以太网, 208
 - 变量, 218
 - 地址类型, 216
 - 数据类型, 208, 216
 - 通道单元, 209, 210, 212, 213, 215
 - 支持的控制器, 209
 - 组态变量, 209, 216, 218
 - 组态连接, 209, 210, 212, 213, 215
- MPI 通道单元, 396
- MQTT 协议, 769

N

- NAT 路由器, 271

O

- OPC
 - HDA 服务器浏览器, 677
 - OPC A&E 服务器的功能, 695
 - OPC UA, 261
 - OPC 条目管理器, 231
 - OPC 条目管理器功能概述, 231
 - OPC 通讯受到干扰时的错误处理, 248, 309
 - OPC1, 243, 258
 - OPCServer.WinCC-(DPC_4001), 239

- ProgID, 648
- WinCC OPC A&E 服务器的功能, 684
- WinCC OPC DA 服务端的功能, 646
- WinCC OPC DA 客户端的功能, 236
- WinCC OPC HDA 服务器的功能, 665
- WinCC OPC UA 服务器的工作原理, 711
- WinCC OPC XML 服务器的功能, 641
- WinCC OPC XML 客户端的功能, 252
- WinCC 中的 OPC, 230, 638
- WinCC 作为 OPC DA 客户端, 230
- WinCC 作为 OPC DA 客户机, 248
- WinCC 作为 OPC UA 客户端, 309
- WinCC 作为 OPC XML 客户端, 230
- 变量选择, 231
- 查询 ProgID, 231
- 创建连接, 231
- 地址属性, 243, 258
- 访问变量, 237
- 跟踪, 741
- 规范, 637
- 过滤标准, 239
- 兼容性, 637
- 结构属性, 245
- 实例, 654, 656, 659, 660, 661, 664
- 实例: , 649, 650, 651, 675
- 使用 HDA 服务器浏览器组态对 WinCC 归档变量的访问, 678
- 使用多个 OPC DA 服务端, 647
- 示例, 653, 655
- 所支持的 WinCC 数据类型, 235
- 所支持的 WinCC 数据类型概述, 235
- 添加变量, 231, 239
- 通道诊断, 610
- 通信概念, 638
- 新建连接, 239
- 新用户, 640
- 用 OPC 条目管理器组态对 WinCC 变量的访问, 239
- 在 OPC 计算机上设置用户帐户, 640
- 在 WinCC OPC DA 客户端上使用结构, 245
- 在 WinCC OPC DA 客户机上使用结构, 246
- 在 WinCC OPC DA 客户机上组态 OPC 通道, 238
- 在不使用 OPC 条目管理器的情况下组态对 WinCC 变量的访问, 243, 258
- 组态对 WinCC 消息系统的访问, 693
- 组态结构, 245
- 组态结构变量, 245
- OPC A 和原始数据, 692
- OPC A&E 服务器, 641, 684, 687, 693, 695, 697, 699
 - 质量代码, 692, 705
 - 组态对 WinCC 消息系统的访问, 693
- OPC A&E 服务器的功能
 - 跟踪事件, 695
 - 简单事件, 695
 - 条件事件, 695
- OPC A&E 服务器上的消息类别, 699
- OPC A&E 服务器上的消息类型, 699
- OPC A&E 客户端
 - OPC A&E 的示例, 693
- OPC DA 服务端, 646
 - WinCC OPC DA 服务端的功能, 646
 - 使用多个 OPC DA 服务端, 647
- OPC DA 客户端, 236
 - WinCC OPC DA 客户端的功能, 236
- OPC HDA 服务器, 665
 - WinCC OPC HDA 服务器的时间格式, 670
 - 范围值, 666
 - 工作原理, 665
 - 配件, 668
 - 属性, 667
 - 数据结构, 666, 667
 - 条目标识号, 666
 - 条目句柄, 666
 - 写访问, 673
 - 原始数据, 665
 - 支持的函数, 669
 - 质量代码, 672, 692
- OPC UA
 - Conformance units, 721
 - WinCC 变量的显示, 723
 - WinCC 作为 OPC UA 客户端, 261
 - 访问变量, 263
 - 记录变量的显示, 724
 - 配置文件, 721
 - 数据类型, 262
- OPC UA 服务器
 - URL, 711
 - 安全策略, 712
 - 工作原理, 711
 - 实例证书, 712
 - 受信任的客户端证书, 714
 - 通信配置文件, 711
 - 应用程序证书, 712
 - 证书, 712
 - 支持的规范, 711
 - 组态文件, 733
- OPC UA 客户端, 261
- OPC UA 历史访问, 725
- OPC UA 数据访问, 725
- OPC XML, 644
 - IIS 中的安全性设置, 644
 - 测试安装, 644
 - 访问变量, 254
- OPC XML 服务器, 641
 - 安装, 642

- OPC XML 客户端, 252
 - WinCC OPC XML 客户端的功能, 252
- OPC 接口, 88
 - 在客户端/服务器系统中, 88
- OPC 条目管理器, 231, 239, 648
 - OPC 条目管理器功能概述, 231
- OPC 通道, 610
 - 在 WinCC OPC DA 客户机上组态 OPC 通道, 238
 - 诊断选项, 610
- OPC 通信, 309
 - 处理 OPC UA 通信中的错误, 309
- OPC 通讯, 248
 - OPC DA 通讯受到干扰时的错误处理, 248
- OPC1, 243, 258
- OPCScout new project1
 - OPC DA 的实例, 656
- OPCServer.WinCC-(DPC_4001), 239

- P**
- PROFIBUS 通道单元, 401
 - 组态连接, 402
- PROFIBUS 系统参数, 595
- ProgID, 231
 - 查询, 648
 - 查询 ProgID, 231
- PSK 密钥, 19, 75, 77

- R**
- REST 服务, 777
- REST 协议, 769

- S**
- S5 AS511, 361
 - 变量地址, 363
 - 定义按字节访问的变量, 367
 - 数据类型, 360
 - 组态, 361
 - 组态按位访问的变量, 366
 - 组态变量, 363
- S5 Ethernet Layer 4, 315
 - iNA960 消息, 333
 - SCI 消息, 334
 - 变量地址, 319
 - 传输参数, 326, 328
 - 连接参数, 317
 - 连接受干扰时的错误代码, 329
 - 内部错误代码和常量, 329
 - 设备名称, 326, 327
 - 数据类型, 316
 - 通道单元的系统参数, 326
 - 原始数据变量, 324
 - 组态, 316
 - 组态按位访问的变量, 320
 - 组态按字访问的变量, 323
 - 组态按字节访问的变量, 321
 - 组态变量, 319
- S5 Profibus FDL, 337
 - 被动连接, 337
 - 变量参数, 341
 - 变量属性, 344, 345, 346, 350
 - 地址属性, 344, 345, 346, 350
 - 读/写监视时间, 347
 - 访问 ARRAY OF BYTE 数据类型的 S5 变量, 338
 - 访问 BIT 数据类型的 S5 变量, 338
 - 访问 BYTE 数据类型的 S5 变量, 338
 - 访问 S5 变量, 338
 - 服务访问点, 337
 - 更改读/写监视时间, 349
 - 更改设备名称, 348
 - 块, 355
 - 类型转换, 339
 - 连接参数, 341, 342, 352, 354
 - 连接属性, 342, 352, 354
 - 启动块, 355
 - 设备名称, 347
 - 示例, 355, 358
 - 数据处理块的组态, 355
 - 数据类型, 338
 - 数据区, 338
 - 特殊功能, 350
 - 添加新驱动程序, 341
 - 通道单元 FDL (CP5412/A2-1), 337
 - 通道单元的系统参数, 347
 - 通道特性, 339
 - 通道诊断, 600
 - 系统参数, 348, 349
 - 循环块, 355
 - 原始数据变量作为字节数组, 350
 - 支持的数据类型, 338
 - 主动连接, 337
 - 组态 SIMATIC S5 Profibus FDL, 341
 - 组态按位访问的变量, 344
 - 组态按字访问的变量, 346
 - 组态按字节访问的变量, 345
 - 组态被动数据传送, 354
 - 组态变量, 344
 - 组态连接, 342
 - 组态通道 SIMATIC S5 Profibus FDL, 341
 - 组态通讯类型, 351

- 组态原始数据变量, 350
 - 组态主动数据传送, 352
 - S5 Serial 3964R, 369
 - 变量的数据类型, 369
 - 组态, 370
 - 组态按位访问的变量, 374
 - 组态按字节访问的变量, 375
 - 组态变量, 372
 - S5 变量, 338
 - 访问 ARRAY OF BYTE 数据类型的 S5 变量, 338
 - 访问 BIT 数据类型的 S5 变量, 338
 - 访问 S5 变量, 338
 - 访问字节数据类型的变量, 338
 - S7 Plus, (请参见: SIMATIC S7-1200、S7-1500 通道)
 - S7 Protocol Suite, 377
 - AR_SEND 变量的属性概述, 427
 - AR_SEND 功能数据块结构和参数, 422
 - 更改逻辑设备名称, 419
 - 将原始数据变量组态为字节数组, 455
 - 连接故障的错误代码, 470
 - 软件冗余, 461
 - 软件冗余 - 连接专用的内部变量, 462
 - 使用 FB S7ProtocolSuite:AR_SEND 变量进行数据交换, 421
 - 数据块结构 AR_SEND 的实例, 431, 432, 433, 434, 439, 441, 443, 445, 448
 - 数据类型, 381
 - 所支持的数据类型, 381
 - 通道单元, 378, 393
 - 通道诊断, 588
 - 系统参数, 415
 - 用于 BSEND/BRCV 功能的原始数据变量, 456
 - 用于多个归档变量的 AR_SEND 变量, 435
 - 用于多个归档变量的 AR_SEND 变量 (已优化), 447
 - 用于一个归档变量的 AR_SEND 变量, 430
 - 原始数据变量, 453
 - 原始数据变量作为字节数组, 453
 - 在 WinCC 中删除软件冗余, 469
 - 周期性读取服务, 416
 - 组态, 383
 - 组态按位访问的变量, 410
 - 组态按字访问的变量, 412
 - 组态按字节访问的变量, 411
 - 组态连接, 394, 397, 400, 402, 404, 405, 408
 - 组态软件冗余, 466
 - 组态文本变量, 414
 - 组态系统参数, 417
 - 组态用于“BSEND/BRCV”功能的原始数据变量, 459
 - 组态用于多个归档变量的 AR_SEND 变量, 451
 - 组态用于归档变量的 AR_SEND 变量, 449
 - S7+, (请参见: SIMATIC S7-1200、S7-1500 通道)
 - S7-1200 控制器, 471
 - S7-1500 控制器, 471
 - S7-1500 R/H-CPU, 491, 493, 496
 - S7DOS, 153
 - SCADA 导出, 385, 483
 - SelfDiagnosis, 112
 - ServiceMode, 93
 - SIMATIC S5 Profibus FDL, 337
 - S5 Profibus FDL, 337
 - 特殊功能, 350
 - 原始数据变量, 350
 - SIMATIC S5 Programmers Port AS511, 360
 - SIMATIC S5 Serial 3964R, 369, 370
 - SIMATIC S7 Protocol Suite, 377
 - SIMATIC S7-1200, S7-1500 channel, 471
 - 通道诊断, 582
 - 组态系统诊断, 586
 - SIMATIC S7-1200, S7-1500 Channel
 - 数据类型, 472
 - 组态, 472, 475, 476
 - 组态连接, 476
 - SIMATIC S7-1200、S7-1500 通道
 - 导入, 384, 482
 - SIMATIC SCADA 导出, 385, 483
 - SIMATIC Shell, 73, 77
 - SIMATIC TI Ethernet Layer 4, 502
 - 传输参数, 510
 - 通道, 502
 - SIMOTION 通道, 522
 - 数据类型, 522
 - 系统参数, 529, 530, 532
 - 诊断, 533
 - 组态, 523, 524, 526, 527, 528
 - System Info, 537
 - 空闲驱动器容量, 539
 - 实例, 561, 562
 - 实例: , 560
 - 使用的数据类型, 545, 557
 - 示例, 545, 552, 557, 559
 - 所支持的系统信息概述, 539
 - 通道诊断, 577
 - 与其它软件组件的不同之处, 544
 - 组态, 544
 - System Info 系统信息
 - 在多用户系统中使用, 556
- ## T
- TI Ethernet Layer 4, 502
 - 按位访问, 507
 - 按字节访问, 509
 - 变量的地址, 506

- 传输参数, 510
- 连接参数, 502, 503
- 设备名称, 510
- 数据类型, 503
- 系统参数, 502, 510, 511, 512
- 组态, 503
- 组态按位访问的变量, 507
- 组态按字节访问的变量, 509
- 组态变量, 505
- TI Serial, 514
 - 变量的地址, 517
 - 变量地址, 517
 - 连接参数, 515
 - 数据类型, 514
 - 组态, 515
 - 组态按位访问的变量, 518
 - 组态按字节访问的变量, 520
 - 组态变量, 516
- TIA Portal
 - AS 结构, 389, 487
 - 导入, 384, 482
- TTY 接口, 360

- U**
- URL
 - OPC UA 服务器, 711

- W**
- Web 客户端, 19
- WinAC Basis, 405
- WinAC Pro, 403
- WinCC, 16
 - ServiceMode, 93
 - WinCC 中的 OPC, 230, 248, 309, 638
 - WinCC 中的 OPC UA, 261
 - 服务模式, 93
 - 客户端服务器系统, 16
 - 作为 OPC DA 客户端, 230
 - 作为 OPC DA 客户机, 248
 - 作为 OPC UA 客户端, 261, 309
 - 作为 OPC XML 客户端, 230
- WinCC - Microsoft Excel 的连接, 660
 - OPC DA 的实例, 660
- WinCC - OPC HDA 客户机连接, 675
 - OPC HDA 的实例, 675
- WinCC - SIMATIC NET FMS OPC 服务器的连接, 653
 - OPC DA 的实例, 653
- WinCC - SIMATIC NET S7 OPC 服务端的连接, 655
 - OPC DA 的实例, 655
- WinCC - WinCC 连接, 649
 - OPC DA 的实例, 649
- WinCC Explorer-OPC_Client.MPC, 648
- WinCC OPC A&E 服务器
 - 层级访问, 697
- WinCC OPC A&E 服务器的功能
 - 简单事件, 684
- WinCC OPC UA 服务器, 711
 - Discovery 服务器, 712
 - 组态, 738
- WinCC Projects, 73
 - WinCC Projects 对话框, 67
 - 调用, 67
 - 远程激活, 85
 - 远程取消激活, 86
- WinCC 变量 @RM_MASTER, 135
- WinCC 变量 @RM_MASTER_NAME, 135
- WinCC 服务, 93
- WinCC 服务模式, 93, 101
- WinCC 过程通信, 153
- WinCC 类型转换, 158
- WinCC 数据类型, 158, 162
 - WinCC 数据类型, 162
 - 类型转换, 162, 170
- WinCC 通道诊断, 579, 596, 606, 616
- WinCC 通信, 153
 - 原理, 153
- WinCC 消息系统
 - OPC A&E 服务器上的 WinCC 消息等级, 687
 - 属性, 688, 701
 - 映射 WinCC 消息等级和消息类型, 687
 - 在 OPC A&OPC-A&WinCC 消息系统上, 685
 - 组态对 WinCC 消息系统的访问, 693
- WinCC 云, (请参见: Cloud Connector)
- WinCC 中的 AR_SEND 功能
 - 变量的属性概述, 427
 - 数据块结构的实例, 431, 432, 433, 434, 439, 441, 443, 445, 448
 - 数据块结构和参数, 422
 - 用于多个归档变量的变量, 435
 - 用于多个归档变量的变量 (已优化), 447
 - 用于归档变量的变量, 430
 - 组态用于多个归档变量的变量, 451
 - 组态用于归档变量的变量, 449
- Windows
 - 任务栏, 98
 - 托盘区域, 98

- 安**
- 安全通信, 77

安装, 91

文件服务器, 91

安装冗余, 120

按

按时间顺序发送消息, 115

按位访问, 185, 320, 409

S5 AS511, 366

S5 Ethernet Layer 4, 320

S5 Profibus FDL, 344

S5 Serial 3964R, 374

S7 Protocol Suite, 410

TI Ethernet Layer 4, 507

TI Serial, 518

按字访问, 409

S7 Protocol Suite, 412

按字节访问, 185, 321, 409

S5 AS511, 367

S5 Ethernet Layer 4, 321

S5 Profibus FDL, 345

S5 Serial 3964R, 375

S7 Protocol Suite, 411

TI Ethernet Layer 4, 509

TI Serial, 520

棒

棒图

系统信息示例, 548

变

变量, 158, 319, 409, 545, 618, 620

HDA 服务器浏览器, 677

OPC DA 的实例, 656, 659, 664

OPC HDA 的实例, 678, 679

OPC 条目管理器, 239

PowerTag, 158

System Info 实例, 545, 560

变量选择, 231

测试变量组态, 595, 605

导出, 391, 489

导入, 384, 482

检查, 581, 598, 609, 618

检查内部变量, 632

添加变量, 231

通过二进制写入组态变量, 186

外部, 158

文本变量的长度信息, 158

在自动化系统中对外部变量寻址, 158

指定线性转换, 161

质量, 620

质量代码, 620, 622

状态, 620, 628

组态, 319, 344, 505, 516

组态 HDA 服务器浏览器, 678

组态 OPC 条目管理器, 239, 243

组态 PC 条目管理器, 258

组态 powertag, 161

组态按位访问, 320, 344, 374, 410, 507, 518

组态按字访问, 323, 346, 412

组态按字节访问, 321, 345, 375, 411, 509, 520

组态变量地址, 372

组态外部变量, 161

组态文本变量, 161

变量的地址, 506

TI Ethernet Layer 4, 506

变量的质量代码, 620, 622

变量地址, 319

S5 Ethernet Layer 4, 315

TI Serial, 514

变量名称

后缀, 160

前缀, 160

变量属性, 161, 344, 345, 346, 350

S5 Profibus FDL 的实例, 358

变量同步, 118

变量状态, 620, 628, 630, 631

使用全局动作监视变量状态, 631

在“动态”(Dynamic)对话框中监视变量状态, 630

标

标准标记, 574

标准服务器, 24

组态, 49

标准项目, 93, 101

切换到服务项目, 105

不

不间断电源, 115

操

操作系统, 32

冗余系统, 115

组态操作员授权, 32

操作员特权, 32

操作系统, 32

客户端, 32

传

- 传输参数, 153, 326
 - SIMATIC S5 Ethernet Layer 4, 326
 - SIMATIC TI Ethernet Layer 4, 510
 - TI Ethernet Layer 4, 510
- 传送更改, 416

串

- 串行电缆, 115

磁

- 磁盘容量, 539
 - 空闲驱动器容量, 539, 544
 - 系统信息示例, 548

错

- 错误点, 550

打

- 打开
 - 项目, 82
- 打印机监控, 539
 - 系统信息示例, 553

导

- 导出, 35
 - 数据包, 35
- 导出工具, (请参见: 西门子 SIMATIC SCADA 导出)
- 导入, 46
 - 数据包, 46

地

- 地址类型
 - Mitsubishi 以太网, 216
- 地址属性, 243, 258, 344, 345, 346, 350
 - S5 Profibus FDL 的实例, 358

电

- 电子服务器
 - WinCC OPC A&OPC 的功能, 641

定

- 定时器, 539
- 定义新变量, 656
 - OPC DA 的实例, 656

读

- 读/写监视时间, 347, 349
 - 更改, 349

多

- 多用户系统, 24, 556
 - 通道的使用, 556
 - 组态, 27
 - 组态客户端, 39
- 多用户项目, 29
 - 创建, 29
- 多站系统, 16

二

- 二进制写入机制, 185
 - 一般步骤, 185

范

- 范围值, 666

方

- 方案, 24
 - 客户端和服务, 24
 - 中央归档服务器, 25

分

- 分布式系统, 16, 24
 - 创建客户端项目, 44
 - 客户端组态, 42
 - 数据包导入, 46

- 远程组态, 73
- 在客户端显示消息, 62
- 在客户机上组态画面变化, 58
- 在客户机显示服务器画面, 56
- 在客户机组态数据输出, 60
- 在客户机组态消息顺序报表, 63
- 组态, 27

服

- 服务 CCCloudConnect, 773
- 服务访问点, 337
- 服务模式, 93
- 服务器, 24, 27
 - SOFTNET 驱动程序的特性, 69
 - 标准服务器, 24
 - 操作员权限, 32
 - 创建新项目, 29
 - 从客户端编辑画面, 84
 - 从客户端激活项目, 85, 86
 - 对多个客户机的项目访问, 80
 - 服务器-服务器通讯, 24
 - 关闭, 71
 - 具有多个网卡的通信特性, 69
 - 启动, 67, 69
 - 文件服务器, 24
 - 远程组态, 32
 - 远程组态 (要求), 31
 - 在服务器项目中组态客户端, 39
 - 组态, 27
 - 组态标准服务器, 49
 - 组态首选服务器, 52
- 服务器-服务器通讯, 24
- 服务器前缀, 54, 56, 58, 60
- 服务器上使用的 SOFTNET 驱动程序, 69
- 服务器数据, 35
 - 数据包导出, 35
 - 数据包导入, 46
 - 组态标准服务器, 49
 - 组态首选服务器, 52
- 服务器证书, 272
- 服务项目, 93, 101
 - 登录, 108
 - 定义, 104
 - 激活交互式操作, 109
 - 脚本, 96
 - 结束交互式操作, 109
 - 连通站, 97
 - 启动列表, 97
 - 切换到标准项目, 104
 - 使用, 96
 - 未发布的组件, 97

- 限制, 96
- 要求, 102
- 诊断信息, 97
- 注销, 108
- 自动启动, 105
- 组态, 95

跟

- 跟踪功能, 574
 - 组态, 574

工

- 工业以太网通道单元, 393
 - 组态连接, 394

关

- 关闭, 71
 - 服务器的, 71
 - 客户机的, 71

归

- 归档存储器, 136, 138

过

- 过程画面
 - System Info 实例, 561
- 过滤标准, 239

后

- 后缀, 160

画

- 画面, 54
 - 编辑服务器项目, 84
 - 服务器前缀, 54
 - 在客户机从多个服务器显示, 56
 - 在客户机上组态画面变化, 58
 - 组态起始画面, 54

缓

缓冲时间, 693

混

混合组态, 22

激

激活, 85

项目, 85

激活冗余服务器, 120

计

计数器, 539

计算机

状态, 98

计算机列表, 31

接受客户机, 31

记

记录册条目, 577, 589, 601, 610

记录册文件, 570

加

加密通信, 19

加载在线更改

对冗余系统的用户归档的更改, 126

兼

兼容性, 637

脚

脚本

服务项目, 96

结

结构

使用 WinCC OPC 客户机, 246

在 WinCC OPC DA 客户端上使用结构, 245

组态结构, 245

结构变量, 245

AS 结构, 389, 487

组态结构变量, 245

结构类型

AS 结构, 389, 487

结构属性, 245

截

截断, 765

介

介质冗余, 492

可

可编程控制器, 339

客

客户端, 19, 42

Web 客户端, 19

编辑服务器画面, 84

操作员权限, 32

创建新项目, 44

打开服务器上的项目, 82

导入数据包, 46

到服务器的视图, 39

多站系统, 39

启动, 68

取消激活服务器项目, 86

瘦客户端, 19

显示多个服务器的消息, 62

远程服务器项目激活, 85

在多用户系统中定义启动属性, 39

在分布式系统中组态, 42

在服务器上同时启动, 22

在服务器项目中组态, 39

组态, 42

组态计算机属性, 39

客户端/服务器方案, 24

客户端/服务器系统, 16

OPC 接口的应用程序, 88

标准服务器, 24

多用户系统, 24

多站系统, 16

分布式系统, 16, 24

服务器-服务器通讯, 24
 关闭服务器, 71
 混合组态, 19
 客户端和服务器的可能数量, 16
 客户端类型, 19
 启动服务器, 67
 启动客户端, 68
 冗余, 16, 24
 数量结构, 22
 文件服务器, 24
 要求, 16
 用途, 16
 运行系统中的特性, 65
 组态, 19
 客户端项目, 44
 客户端证书, 272
 客户机
 对服务器项目的访问, 80
 关闭, 71
 进入服务器的计算机列表, 31
 显示多个服务器, 60
 显示多个服务器的画面, 56
 组态画面变化, 58
 组态起始画面, 54
 组态消息顺序报表, 63
 客户机/服务器系统
 关闭客户机, 71

块

块, 355
 循环块, 355

类

类型转换, 158, 170, 339

离

离线组态, 384, 482

连

连接, 153, 393, 616
 被动连接, 337
 测试, 616
 测试连接组态, 595, 605
 创建连接, 231
 创建新连接, 160
 检查, 579, 596, 606

建立/终止, 156
 运行系统中的状态, 156, 568
 主动连接, 337
 组态, 342, 394, 397, 400, 402, 404, 405, 408
 连接 WinCC - OPC A&E 客户端, 693
 OPC A&E 的示例, 693
 连接参数, 153, 317, 341, 342, 352, 354, 393, 605
 S5 AS511, 360, 361
 S5 Ethernet Layer 4, 315
 S7 Protocol Suite, 377
 SIMATIC S5 Serial 3964R, 369
 TI Ethernet Layer 4, 502, 503
 TI Serial, 514
 通道单元, 503
 连接监控, 236
 连接属性, 160, 342, 352, 354
 连接状态, 98
 测试, 568
 连通站
 服务项目, 97

路

路由器, 77

逻

逻辑设备名称, 415
 修改, 419

启

启动, 67, 68
 服务器的, 67, 69
 客户端的, 68
 启动块, 355
 启动列表
 服务项目, 97
 启动特性, 67, 68
 服务器的, 67
 客户端的, 68

起

起始屏幕, 54
 在客户机组态, 54

前

前缀, 160

情

情况, 135

取

取值范围, 170

确

确认策略, 703

确认方法, 690

任

任务栏, 98

日

日期, 539

日志文件, 577, 589, 601, 610

冗

冗余, 24, 112

CPU 冗余, 491

按时间顺序发送消息, 115

操作系统, 115

服务器之间串行连接, 123, 135

激活服务器, 112

切换归档, 112

冗余 PLC, 491, 493

软件冗余, 496

设置, 120

时间同步, 115

首选服务器, 24

系统消息, 136

要求, 115

应用程序正常检查, 112, 141

组态首选服务器, 52

冗余服务器之间串行连接, 123

冗余系统

复制项目, 127, 129

组态, 122

冗余系统的要求, 115

软

软件冗余, 461, 496

连接专用的内部变量, 462

在 WinCC 中删除, 469

在 WinCC 中组态, 466

设

设备名称, 326, 347, 510

TI Ethernet Layer 4, 510

更改, 348

设置, 120

时

时间, 539, 544

系统信息示例, 547

时间戳, 422

时间同步, 115

实

实例: , 631

实例证书, 272

首

首选服务器, 24

组态, 52

瘦

瘦客户端, 19

属

属性 - CP5613, 592, 594, 602, 604

数

数据, 56

显示多个服务器, 56

显示多个服务器中的数据, 60

数据包, 35, 46

隐含更新, 35

在 WinCC 项目管理器中显示, 35, 46

- 在文件系统中保存, 35
- 组态导出, 35
- 组态导入, 46
- 数据传输, 351
 - 组态被动数据传送, 354
 - 组态主动数据传送, 352
- 数据类型, 158, 235, 262, 316, 381, 472, 539
 - Mitsubishi 以太网, 208, 216
 - OPC UA, 262
 - S5 AS511, 360
 - S5 Ethernet Layer 4, 316
 - S5 Profibus FDL, 338
 - S5 Serial 3964R, 369
 - S7 Protocol Suite, 381
 - SIMATIC S7-1200, S7-1500 Channel, 472
 - TI Ethernet Layer 4, 503
 - 类型转换, 170
 - 使用的数据类型, 539, 545
 - 所支持的 WinCC 数据类型, 235
 - 所支持的 WinCC 数据类型概述, 235
- 数据区, 338
- 数量结构, 22
- 数值变量类型, 158
 - 线性标定, 158

锁

- 锁定消息, 115
 - 被动, 115
 - 主动, 115
- 锁定消息 (被动), 115
- 锁定消息 (主动), 115
- 锁定消息类别, 115
 - 被动, 115

特

- 特殊功能, 350

添

- 添加变量, 231, 239
 - OPC DA 的实例, 651
- 添加新驱动程序, 341

条

- 条目标识号, 666
- 条目句柄, 666

调

- 调试, 741

通

- 通道, 153, 315, 369, 377, 616
 - Mitsubishi 以太网, 208
 - S5 PROFIBUS FDL 的实例, 355
 - SIMATIC S5 Ethernet Layer 4, 315
 - SIMATIC S5 Programmers Port AS511, 360
 - SIMATIC S5 Serial 3964R, 369
 - SIMATIC S7 Protocol Suite, 377
 - SIMATIC S7-1200, S7-1500 channel, 471
 - SIMATIC TI Ethernet Layer 4, 502
 - TI Serial, 514
 - 测试, 616
 - 检查, 579, 596, 606
 - 建立/终止连接, 156
 - 逻辑连接状态, 156, 568
 - 组态 SIMATIC S5 PROFIBUS FDL, 341
 - 通道 S5Serial3964R 的设备状态监控
 - 设备状态监控, 370
 - 通道单元, 153, 315, 341, 378
 - 505 Ethernet (CP 1413-x), 502
 - 505 Serial Unit #1, 514
 - FDL (CP5412/A2-1), 341
 - Mitsubishi 以太网, 208
 - MPI, 396
 - PROFIBUS, 401
 - S5 Transport (CP1413-x), 317
 - S5 Transport (TCP/IP), 317
 - S5-AS511, 360, 361
 - S5-RK512 (3964R), 369
 - SIMATIC S7-1200, S7-1500 channel, 471
 - SIMATIC S7 Protocol Suite S7 Protocol Suite 通道选择: 选择通道单元, 378
 - Slot PLC, 403
 - Soft PLC, 405
 - TCP/IP, 407
 - 工业以太网, 393
 - 连接参数, 503
 - 系统参数, 326, 347, 415, 510, 511
 - 指定连接, 398
 - 通道单元 FDL (CP5412/A2-1), 337, 341
 - 通道诊断, 570
 - “SIMATIC S5 PROFIBUS FDL”通道, 600
 - “SIMATIC S7 Protocol Suite”通道, 588
 - “System Info”通道, 577
 - OPC 通道, 610

- PROFIBUS 系统参数, 595
- SIMATIC S7-1200, S7-1500 channel, 582
- WinCC 通道诊断, 579, 596, 606, 616
- 测试通道, 579, 596, 606, 616
- 跟踪功能的标准标记, 574
- 记录册条目的 ERROR 标记, 577, 589
- 记录册条目的 INFO 标记, 577, 589, 601, 610
- 检查变量, 581, 598, 609, 618
- 检查连接, 579, 596, 606, 616
- 检查内部变量, 632
- 检查通信处理器的组态, 592, 594, 602, 604
- 检查组态数据, 615
- 连接参数, 605
- 连接状态, 568
- 逻辑连接状态, 568
- 日志文件, 577, 589, 601, 610
- 实例, 631
- 使用全局动作监视变量状态, 631
- 属性 - CP5613, 592, 594, 602, 604
- 通道, 使用, 571
- 通道使用, 573
- 通道诊断, 570, 571, 572
- 通道诊断, 使用, 571
- 通道诊断使用, 570
- 系统参数, 605
- 在“动态”(Dynamic)对话框中监视变量状态, 630
- 诊断选项, 577, 588, 600, 610
- 状态消息概述, 579, 596, 606, 616
- 组态通道的跟踪功能, 574
- 作为 Windows 应用程序的通道诊断, 572
- 通道诊断 通道诊断
 - 使用通道诊断进行通道诊断, 571
- 通道诊断跟踪文件, 570
- 通道诊断通道
 - 测试, 571, 573
- 通道诊断通道诊断
 - 通道诊断通道诊断, 572
- 通配符, 765
- 通信, 151
 - AS 数据类型, 161, 170
 - IPv4 协议, 153
 - PowerTag, 158
 - WinCC 过程通信, 153
 - WinCC 和自动化系统之间的通信, 153
 - WinCC 类型转换, 158
 - WinCC 数据类型, 158
 - WinCC 通信的原理, 153
 - 按 AS 数据类型排序的格式改编, 170
 - 按 WinCC 数据类型排序的格式改编, 162
 - 变量属性, 161
 - 创建新连接, 160
 - 基础知识, 151
 - 类型转换, 158, 170
 - 连接, 153
 - 连接属性, 160
 - 取值范围, 170
 - 数据类型, 158, 170
 - 数值变量类型的线性标定, 158
 - 通道单元, 153
 - 通过二进制写入组态变量, 186
 - 通信驱动程序, 153
 - 位/字节变量, 186
 - 文本变量的长度信息, 158
 - 在自动化系统中寻址, 158
 - 组态 powertag, 161
- 通信处理器, 592, 594, 602, 604
 - 检查通信处理器的组态, 592, 594, 602, 604
- 通信驱动程序, 153, (请参见通道)
 - 建立/终止连接, 156
- 通讯
 - 二进制写入机制, 185
 - 二进制写入机制的原理, 185
 - 位/字节访问, 185
- 通讯类型, 351
 - 组态, 351
- 托**
- 托盘区域, 98
- 网**
- 网络, 77, 151
 - 路由器, 77
 - 子网, 77
- 位**
- 位/字节变量, 186
- 位/字节访问, 185
- 文**
- 文本变量, 158, 409
 - 组态, 161, 414
- 文件服务器, 24, 91
 - 安装, 91
 - 组态, 91
- 西**
- 西门子 SIMATIC SCADA 导出, 385, 483

系

- 系统变量, 143
- 系统参数, 153, 326, 415, 529, 530, 532, 605
 - S5 Ethernet Layer 4, 315
 - S5 Profibus FDL, 337
 - SIMATIC S7 Protocol Suite, 377
 - TI Ethernet Layer 4, 502, 510, 511, 512
 - 通道单元, 510, 511
 - 组态, 417, 419
- 系统错误, 65
- 系统特性, 65
- 系统消息, 146
- 系统信息
 - I/O 域组态, 547
 - 棒图组态, 548
 - 错误点, 550
 - 示例: , 547, 548, 549, 550, 553
 - 消息文本, 550
 - 要监视的变量, 550
 - 状态显示组态, 553
- 系统诊断, 586
 - 设备视图, 582
 - 详细视图, 583
 - 诊断缓冲区视图, 584
- 系统诊断视图, 585

夏

- 夏时制/标准时间, 422

显

- 显示, 56
 - 从多个服务器, 56
 - 来自多个服务器的数据, 60

项

- 项目, 29, 44
 - OPC DA 的实例, 651
 - WinCC 状态, 98
 - 创建新客户端项目, 44
 - 打开远程, 82
 - 复制, 127, 129
 - 激活, 85
 - 取消激活, 86
 - 远程访问, 80
 - 在服务器上创建新项目, 29

- 项目复制器, 127, 129

消

- 消息, 62
 - System Info 实例, 552
 - 导入, 384, 482
 - 系统信息示例, 550
 - 显示多个服务器中的数据, 62
 - 在客户机组态消息顺序报表, 63
- 消息顺序报表, 63
- 消息文本, 550

新

- 新建连接, 239
- 新用户, 640

性

- 性能, 22

要

- 要监视的变量, 550

隐

- 隐含更新, 35, 46

应

- 应用程序正常检查, 112, 141

用

- 用户帐户, 640
 - 公布 OPC 计算机, 640

原

- 原始数据, 665
- 原始数据变量, 350, 453
 - 用于 BSEND/BRCV 功能, 456
 - 组态, 350
 - 组态为字节数组, 455
 - 组态以使用 BSEND/BRCV 功能, 459
 - 作为字节数组, 350, 453

远

- 远程组态, 73
 - 编辑画面, 84
 - 打开项目, 82
 - 对服务器项目的访问, 80
 - 激活项目, 85
 - 取消激活项目, 86

运

- 运行系统, 65
 - 报表, 65
 - 出现系统错误时的特性, 65
 - 归档, 65
 - 脚本, 65
 - 图形, 65
 - 文本库, 65
 - 系统特性, 65
 - 消息, 65
 - 用户管理器, 65

诊

- 诊断
 - 连接状态, 156, 568
 - 通道, 533
- 诊断功能
 - SIMATIC S7-1200, S7-1500 channel, 582
- 诊断信息
 - 服务项目, 97
- 诊断选项, 577, 588, 600, 610
 - "SIMATIC S5 PROFIBUS FDL"通道, 600
 - "SIMATIC S7 Protocol Suite"通道, 588
 - "System Info"通道, 577
 - OPC 通道, 610

证

- 证书, 272

支

- 支持的系统信息, 539
 - 概述, 539

质

- 质量代码, 672, 692, 705

中

- 中央归档服务器, 25

周

- 周期性读取服务, 416

状

- 状态显示, 98, 156, 579, 596, 606, 616
 - 系统信息示例, 553

子

- 子网, 77, (见子网)

自

- 自动启动, 105

字

- 字节数组, 350

组

- 组态, 19, 27, 42, 91, 122
 - TI Ethernet Layer 4, 503
 - WinCC OPC UA 服务器, 738
 - 安全设置, 739
 - 变量, 505
 - 端口号, 738
 - 多用户系统, 27
 - 分布式系统, 27
 - 服务器, 27
 - 客户端, 42
 - 客户端服务器系统, 19
 - 文件服务器, 91
 - 用户标识, 739
 - 优化的 WinCC 归档写访问, 740
 - 远程, 73

组态按位访问

 变量, 507

组态按位访问的变量

 TI Ethernet Layer 4, 507

组态按字节访问

 变量, 509

组态按字节访问的变量

 TI Ethernet Layer 4, 509

组态变量

 TI Ethernet Layer 4, 505

组态数据, 615

组态文件

 WinCC OPC UA 服务器的 URL, 733

 WinCC 记录写访问, 733

 安全设置, 733

 布局, 733

 存储位置, 733

 端口号, 733

 用户标识, 733

 证书, 733

最

 最大大小, 693

作

 作为 ActiveX 控件, 571