

SIMATIC HMI

WinCC V7.5 SP2 WinCC/连通性软件包文档

系统手册




WinCC/连通性软件包	1
概览: WinCC/连通性软件包	2
应用	3
通过 OLE DB 提供者进行访问	4
使用 MS OLE DB 访问 SQL 数据库期间的安全设置	5
通过 OPC 访问 - 开放式互连	6
使用 OLE DB 提供者访问的实例	7
连通站	8
OPC UA 的连通站	9

在线帮助的打印输出

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	WinCC/连通性软件包	7
2	概览：WinCC/连通性软件包	9
3	应用	15
3.1	使用范例 1：本地访问 WinCC RT 数据库	15
3.2	使用范例 2：远程访问 WinCC RT 数据库	16
3.3	使用范例 3：本地访问 WinCC 归档数据库	18
3.4	使用范例 4：远程访问 WinCC 归档数据库	19
3.5	使用范例 5：本地访问 WinCC 用户归档	21
3.6	使用范例 6：远程访问 WinCC 用户归档	22
3.7	使用范例 7：通过 WinCC OLE DB Provider 进行访问	23
3.8	使用 OPC 访问 WinCC 归档、变量和消息	25
3.9	对归档数据的透明访问	27
3.10	SQL Server 的功能	29
4	通过 OLE DB 提供者进行访问	31
4.1	使用 OLE DB 提供程序访问归档数据	31
4.2	OLE DB 的基础	33
4.3	WinCC 归档连接器	35
4.4	建立到归档数据库的连接	39
4.5	查询归档数据	42
4.5.1	显示过程值归档	42
4.5.2	查询归档数据	43
4.5.3	查询过程值归档	45
4.5.4	查询报警消息归档	53
4.5.5	显示报警消息归档	55
4.5.6	用户归档的查询	58
4.5.7	显示用户归档	60
4.5.8	通过“SQL Server Import/Export”向导组态访问	61
4.5.9	满足使用报表服务的先决条件	63
4.6	消息和过程值的分析函数	65
4.6.1	消息和过程值的分析函数	65
4.6.2	显示分析查询的消息归档	68

5	使用 MS OLE DB 访问 SQL 数据库期间的安全设置	71
6	通过 OPC 访问 - 开放式互连	73
6.1	OPC 通道	73
6.1.1	WinCC OPC 通道	73
6.1.2	OPC 条目管理器	74
6.1.3	所支持的 WinCC 数据类型概述	78
6.1.4	WinCC OPC DA 客户机	79
6.1.4.1	WinCC OPC DA 客户机的功能	79
6.1.4.2	如何通过 OPC 条目管理器访问 WinCC 变量	80
6.1.4.3	在不使用 OPC 条目管理器的情况下访问 WinCC 变量	86
6.1.4.4	在 WinCC OPC DA 客户机上使用结构	88
6.1.4.5	OPC DA 通讯受到干扰时的错误处理	91
6.1.5	WinCC OPC XML 客户机	95
6.1.5.1	WinCC OPC XML 客户机的功能	95
6.1.5.2	如何通过 OPC 条目管理器访问 WinCC 变量	97
6.1.5.3	在不使用 OPC 条目管理器的情况下访问 WinCC 变量	101
6.2	OPC - 开放式互连	104
6.2.1	OPC - 开放式互连	104
6.2.2	OPC 的功能	104
6.2.3	OPC 规范及兼容性	105
6.2.4	在 WinCC 中使用 OPC	105
6.2.5	如何组态 Windows 以使用 WinCC OPC	107
6.2.6	WinCC OPC XML DA 服务器	108
6.2.6.1	WinCC OPC XML DA 服务器的功能	108
6.2.6.2	安装 WinCC OPC XML DA 服务器	109
6.2.6.3	使用 IIS 设置安全设置 (WinCC OPC XML DA 服务器)	111
6.2.6.4	测试安装 (WinCC OPC XML DA 服务器)	111
6.2.7	WinCC OPC DA 服务器	113
6.2.7.1	WinCC OPC DA 服务端的功能	113
6.2.7.2	使用多个 OPC DA 服务器	114
6.2.7.3	查询 OPC DA 服务器名称	115
6.2.7.4	OPC DA 连接的实例	116
6.2.8	WinCC OPC HDA 服务器	132
6.2.8.1	WinCC OPC HDA 服务器的功能	132
6.2.8.2	WinCC OPC HDA 服务器的数据结构	133
6.2.8.3	质量代码	139
6.2.8.4	支持的写访问	139
6.2.8.5	OPC HDA 连接实例	142
6.2.8.6	WinCC 中 OPC HDA 服务器针对非周期性记录的特殊功能	147
6.2.9	WinCC OPC A&E 服务器	150
6.2.9.1	WinCC OPC A&E 服务器的功能	150
6.2.9.2	在 OPC A&E 上映射 WinCC 消息系统	151
6.2.9.3	OPC A&E 质量代码	158

6.2.9.4	OPC A&E 连接实例.....	158
6.2.9.5	带层级访问的 OPC A&E 服务器.....	161
6.2.9.6	读取归档消息.....	171
6.2.10	WinCC OPC UA 服务器.....	176
6.2.10.1	WinCC OPC UA 服务器的工作原理.....	176
6.2.10.2	OPC UA 的安全概念.....	177
6.2.10.3	组态安全机制.....	181
6.2.10.4	支持的 OPC UA 服务和配置文件.....	185
6.2.10.5	WinCC OPC UA 服务器的名称区域.....	187
6.2.10.6	OPC UA 数据访问.....	190
6.2.10.7	OPC UA 历史访问.....	190
6.2.10.8	OPC UA 报警和条件.....	192
6.2.10.9	WinCC 消息系统的属性.....	196
6.2.10.10	WinCC OPC UA 服务器的组态.....	198
6.2.11	诊断.....	205
7	使用 OLE DB 提供者访问的实例.....	207
7.1	使用 OLE DB 提供者访问的实例.....	207
7.2	示例：使用 VB 组态对归档数据的访问.....	208
7.3	实例：使用 DataConnector Wizard 组态对归档数据的访问.....	210
7.4	实例：分析过程值归档.....	215
7.4.1	实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取过程值归档.....	215
7.4.2	实例：使用 WinCC OLE DB 提供程序和 Visual Basic.NET 读取过程值归档.....	223
7.5	实例：分析报警消息归档.....	228
7.5.1	实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取消息归档数据.....	228
7.5.2	示例：使用 WinCC OLE DB 提供程序和 Visual Basic .NET 读取消息归档数据.....	231
8	连通站.....	237
8.1	连通站的基础.....	237
8.2	连通站功能.....	240
8.2.1	使用连通站的 OPC 接口.....	240
8.2.2	使用连通站的 OLE DB 接口.....	241
8.3	在 S7 项目中组态连通站.....	244
8.4	用连通站访问 WinCC 数据.....	246
9	OPC UA 的连通站.....	249
9.1	有关 OPC UA 连通站的基本信息.....	249
9.2	使用连通站的 OPC UA 接口.....	251
9.3	在 S7 项目中组态连通站 (OPC UA).....	252
9.4	通过连通站访问 WinCC 数据 (OPC UA).....	254

索引 255

WinCC/连通性软件包

内容

使用 WinCC 连通性软件包，可通过 OPC 或 WinCC OLE DB 提供程序接收对 WinCC 数据库授权的访问。

本章将介绍

- 如何安装连通性软件包服务器和客户机；
- WinCC 的 OPC 服务器；
- 如何组态对报警消息归档的访问；
- 如何组态对过程值归档的访问。
- 如何组态连通站。

本文档的目标组

本文档供具有 Visual Basic 和 SQL Server 知识的开发人员使用。

概览：WinCC/连通性软件包

简介

使用 WinCC/连通性软件包，可启用对 WinCC 的在线和归档数据的授权的访问。

WinCC OLE DB 提供程序使得访问过程值和报警归档成为可能。可以将数据库中存储的压缩数据读取为解压缩的数据。

WinCC OLE DB 提供程序还提供了分析功能，例如归档变量的最小值、最大值等。

OPC 通过下列标准化接口在自动化领域实现了跨制造商通信：

- OPC HDA 1.20（历史数据访问）
- OPC A&E 1.10（报警和事件）
- OPC DA 2.05a, 3.00
- OPC XML DA 1.01
- OPC UA 1.03

结果，作为 OPC 服务器运行的 WinCC 可将当前过程数据和消息转发给其它 OPC 客户端（例如 Microsoft Excel）。

WinCC 连通站还使用 OPC 访问不同 WinCC 站的数据。

连通性软件包包括：

- WinCC OLE DB 提供程序，用于在 WinCC RT 计算机和归档服务器上直接访问 SQL server 数据库中的过程值和报警归档。
- 工具“归档连接器”，用于组态数据库访问。
通过该工具，可为 SQL server 连接或断开换出的 WinCC 归档数据库。这样做时，将生成各个数据库部分的概述。
归档连接器可监视文件夹和在归档中复制的自动连接。
- WinCC DataConnector，用于在过程图像中组态和访问过程值和报警归档。
- WinCC 连通站
- Server OPC HDA 1.20（历史数据访问）的许可证，以访问 WinCC 归档系统中的历史数据。
- Server OPC A&E 1.10（报警和事件）的许可证，以转发和确认消息。
- Server OPC XML DA 3.00（OPC 可扩展标记语言）的许可证。

- OPC UA 服务器 1.03 的许可证
- 用于通过 WinCC OLE DB 提供程序访问 WinCC 归档数据库或通过 Microsoft OLE DB 访问 WinCC 用户归档的许可证。

WinCC OLE DB 提供程序

WinCC OLE DB 提供程序在以下系统中可用:

- 连通性软件包客户端
- 连通性软件包服务器
- WinCC 站
- 连通站

使用 WinCC OLE DB 提供程序, 连通性软件包客户端可在过程值和消息归档中直接访问 WinCC 归档数据。

访问可发生在连通性软件包客户机的以下不同组态中:

- 在客户端计算机上存在如 WinCC Basis、Web Navigator 服务器、DataMonitor 服务器或连通性软件包服务器这样的 WinCC 软件。
连通性软件包客户端不必显式安装。
通过 WinCC 许可证提供许可。
- 客户端计算机上未安装 WinCC 软件。
这种情况下, 需要将连通性软件包客户端安装在客户端计算机上。

同时访问归档和运行系统数据库

连通性软件包提供了“ArchiveMonitor”函数。

该函数可将激活的 WinCC 项目的运行系统和归档数据库的数据自动合并到 SQL 数据库“CC_ExternalBrowsing”中, 并在其中创建“AMT”和“ARCHIVE”视图。由于这些视图为 WinCC OLE DB 提供程序提供了所有必需信息, 因此可将“CC_ExternalBrowsing”数据库用作该提供程序的数据源。

禁用 WinCC 运行系统时, 将删除在“CC_ExternalBrowsing”中另外创建的视图。

当归档服务器中不包含任何运行系统数据库时, 不能使用“CC_ExternalBrowsing”数据库访问归档数据库。

将 WinCC OLE DB 提供程序用作“链接的服务器”

可在 SQL server“链接的服务器”列表中注册 WinCC OLE DB 提供程序。

这与数据库为本地或远程无关。

说明

SQL server 设置

将 WinCC OLE DB 提供程序组态为“链接的服务器”时，必须在“链接的服务器属性”对话框的“提供程序名”域中选择“归档的 WinCC OLE DB 提供程序”条目。

在“提供程序选项”对话框中，激活“AllowInProcess”复选框。

在“SQL Server 在线手册”的“为分布式查询组态 OLE DB 提供程序”中，可找到更多详细信息。

将 WinCC OLE DB 提供程序用作链接的服务器的实例如下，该服务器名为“WinCC”：

```
select min(realvalue) from openquery(WINCC,'Tag:R,1,''0000-00-00
00:01:00.000'',''0000-00-00 00:00:00.000''')

select * from openquery(WINCC,'Tag:R,1,''0000-00-00
00:01:00.000'',''0000-00-00 00:00:00.000''')
```

访问字符串/Int64/UInt64 数据类型的变量

要访问以下变量类型的过程值，请将“TAG:R”架构替换为“TAG_EX:R”：

- 文本变量 8 位字符集/16 位字符集
- IEEE 754 64 位浮点数

在使用分析函数的查询中，使用“VARIANT”数据类型作为列名称。

WinCC 归档连接器

连通性软件包中还包括工具“归档连接器”，用于组态数据库访问。

通过该工具，可将换出的 WinCC 归档重新连接到 SQL server。

通过 WinCC OLE DB 提供程序，可使归档数据再次可用。

通过归档连接器，可实现以下功能：

- 手动连接：
 - 本地换出的 WinCC 归档数据库处于选中状态，然后连接到本地 SQL server。
- 手动断开：
 - 已连接的数据库处于选中状态，然后断开与 SQL server 的连接。
- 自动连接：
 - 用于换出的 WinCC 归档的本地文件夹处于选中状态。
 - 在复制时，所选文件夹中的所有归档均将自动连接到 SQL server。

只有使用本地 SQL Server 和连通性软件包许可证, 才能运行归档连接器。

说明

如果通过“CC_ExternalBrowsing”访问消息归档或换出的归档, 则该访问可能需要数分钟的时间。

WinCC DataConnector

DataConnector 提供以下功能:

- 使用“DataConnector 向导”, 可在“图形编辑器”中为消息或过程值归档的归档查询组态过程画面。
- 打开过程画面后, DataConnector 控件将在运行系统中执行归档查询, 并且将显示找到的数据。

查询可以使用相对或绝对时间范围。

对于消息, 可采用表格格式显示查询结果; 对于过程值, 可采用表格或趋势格式显示查询结果。

OPC

只有在 WinCC Runtime 正在运行的计算机上, 才能通过 OPC 进行访问。

WinCC OPC HDA 服务器

通过 WinCC OPC HDA 服务器, WinCC 将归档系统的历史数据提供给其它应用程序。

OPC HDA 客户端 (例如报表工具) 可指定某个时间间隔的开始和结束以及在特定情况下请求相应的数据。OPC HDA 客户端也能够特定请求在 WinCC OPC HDA 服务器上先前处理的数据, 并且可在数据传送前有效地影响数据压缩。

WinCC OPC A&E 服务器

通过 WinCC OPC A&E 服务器, 可将 WinCC 消息描绘为报警。该报警只能与伴随它的过程值一起转发到公司的生产或管理级别, 该报警在那里也要经过确认。通过过滤器, 仅传送所选数据。

WinCC OPC XML 服务器

WinCC OPC-XML 服务器以 Web 服务形式为 OPC-XML 客户端提供 OPC 过程数据。通过任何平台以及同时通过 Intranet 和 Internet, OPC-XML 客户端可访问 WinCC 运行系统数据。

WinCC OPC UA 服务器

WinCC OPC UA 服务器可提供过程值、变量归档的值和 WinCC 消息。

说明

WinCC 归档数据库: 连通性软件包许可证

要通过 OPC HDA 访问 WinCC 归档数据库、通过 OPC XML 访问过程值以及通过 OPC A&E 访问报警, 在包含该数据的计算机上需要连通性软件包许可证。

不需要安装连通性软件包服务器或客户端。

OPC DA 通过有效 WinCC RT 许可证来授权。

WinCC 连通站

连通站允许访问各种不同 WinCC 站 (如 WinCC 服务器) 的数据。

连通站在 S7 项目中组态, 后者也管理这些 WinCC 站。

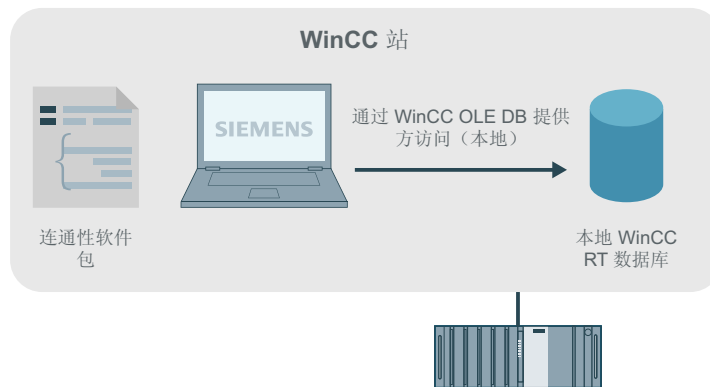
您将通过经由 OPC 服务器的 OPC 客户端访问这些 WinCC 站。

参见

使用 OPC 访问 WinCC 归档、变量和消息 (页 25)

应用

3.1 使用范例 1：本地访问 WinCC RT 数据库



原理

应用程序使用 WinCC OLE DB 提供程序访问本地 WinCC RT 数据库。
可以本地分析归档数据，例如，可以计算过程值的标准偏差。

软件要求

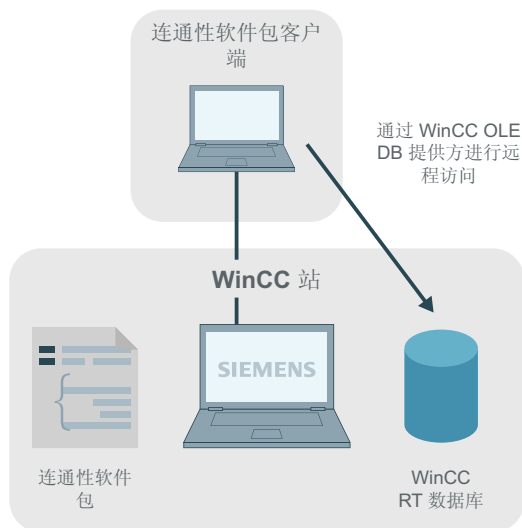
在 WinCC 站上，需要安装以下许可证：

- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包

参见

使用 OLE DB 提供者访问的实例 (页 207)

3.2 使用范例 2：远程访问 WinCC RT 数据库



原理

连通性软件包客户端远程访问 WinCC 站的 WinCC RT 数据库。通过 WinCC OLE DB 提供程序，连通性软件包客户端可读取过程值和报警消息归档的数据。

在本使用范例中，由于未访问换出的 WinCC 归档，因此归档连接器不必将 WinCC 归档连接到 SQL server。

使用“连通性软件包客户端”，可显示、分析或处理数据，例如将数据导出到 CSV 文件。

软件要求

在 WinCC 站上，需要安装以下许可证：

- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包

访问可能会发生在组态方式各不相同的连通性软件包客户端之间。

- 在客户端计算机上存在如 WinCC Basis、Web Navigator 服务器、DataMonitor 服务器或连通性软件包服务器这样的 WinCC 软件。
连通性软件包客户端不必显式安装。通过 WinCC 许可证提供许可。
- 客户端计算机上未安装 WinCC 软件。
需要将连通性软件包客户端安装在客户端计算机上。

参见

使用 OLE DB 提供者访问的实例 (页 207)

3.3 使用范例 3：本地访问 WinCC 归档数据库



原理

应用程序使用 WinCC OLE DB 提供程序访问本地归档数据库。较早的归档数据从 WinCC RT 数据库复制到同一计算机的单独目录中。

通过归档连接器，换出的 WinCC 归档被重新连接到 SQL Server。然后，使用 WinCC OLE DB 提供程序可访问这些归档。

可显示、搜索或分析本地归档数据，例如搜索过程错误或优化过程。

软件要求

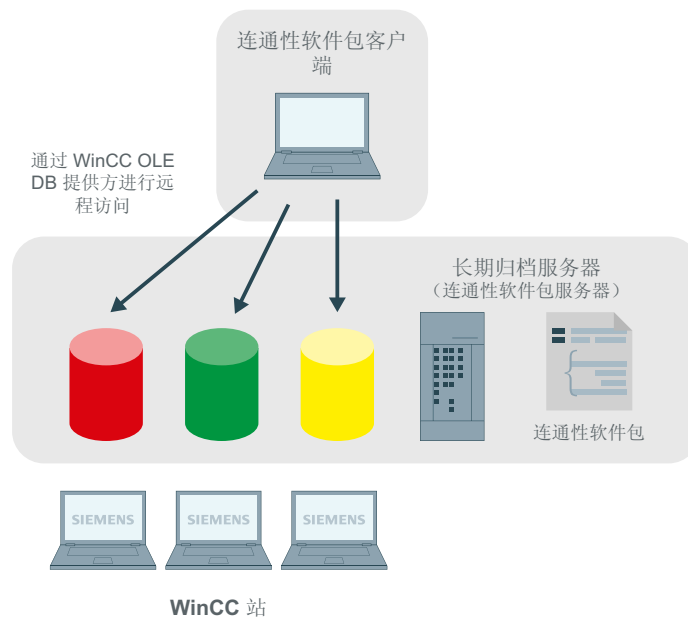
WinCC 站需要安装以下各项：

- WinCC 基本系统
- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包的许可证

参见

使用 OLE DB 提供者访问的实例 (页 207)

3.4 使用范例 4：远程访问 WinCC 归档数据库



原理

长期归档服务器用于保护过程值和消息归档的数据库文件，例如每月备份一次。

通过归档连接器，换出的 WinCC 归档被重新连接到 SQL Server。然后，使用 WinCC OLE DB 提供程序可访问这些归档。

连通性软件包客户端通过 WinCC OLE DB 提供程序访问归档。例如，通过 VB 应用程序，可分析归档以及显示特定日期的过程值。

说明

如果通过“CC_ExternalBrowsing”访问消息归档或换出的归档，则该访问可能需要数分钟的时间。

软件要求

长期归档服务器需要安装以下各项：

- 连通性软件包服务器
- WinCC 连通性软件包的许可证

3.4 使用范例 4: 远程访问 WinCC 归档数据库

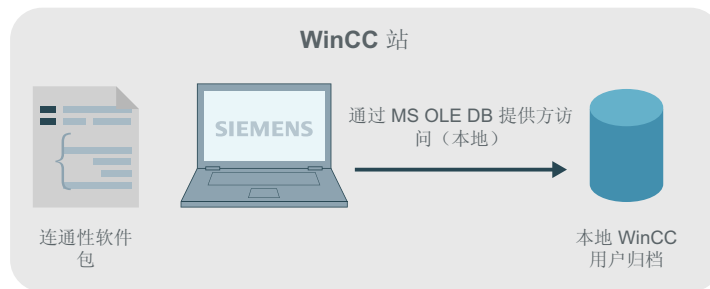
访问可发生在组态方式各不相同的连通性软件包客户端之间。

- 在客户端计算机上存在如 WinCC Basis、Web Navigator 服务器、DataMonitor 服务器或连通性软件包服务器这样的 WinCC 软件。
连通性软件包客户端不必显式安装。通过 WinCC 许可证提供许可。
- 客户端计算机上未安装 WinCC 软件。
需要将连通性软件包客户端安装在客户端计算机上。

参见

使用 OLE DB 提供者访问的实例 (页 207)

3.5 使用范例 5：本地访问 WinCC 用户归档



原理

应用程序使用 MS OLE DB 提供程序访问本地归档的 WinCC 用户归档。

例如，通过 VB 应用程序，可为本地归档数据显示、搜索及写回修改的值。

软件要求

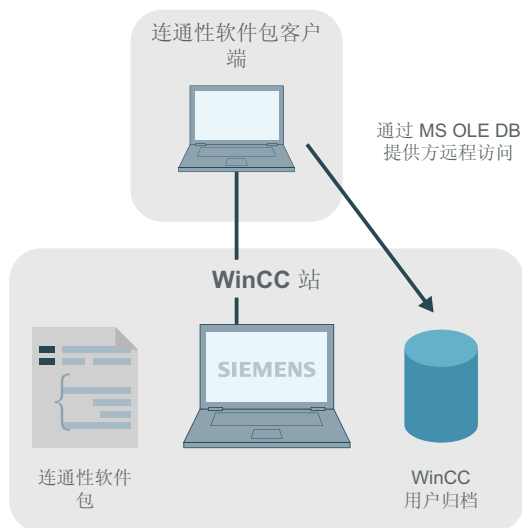
WinCC 站需要安装以下各项：

- WinCC 基本系统
- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包的许可证

参见

使用 OLE DB 提供者访问的实例 (页 207)

3.6 使用范例 6：远程访问 WinCC 用户归档



原理

连通性软件包客户端通过 MS OLE DB 提供程序访问 WinCC 用户归档。例如，通过 VB 应用程序，可为归档数据显示、搜索及写回修改的值。

软件要求

WinCC 服务器需要安装以下各项：

- WinCC 基本系统
- 有效的 WinCC RT 许可证
- WinCC 连通性软件包的许可证

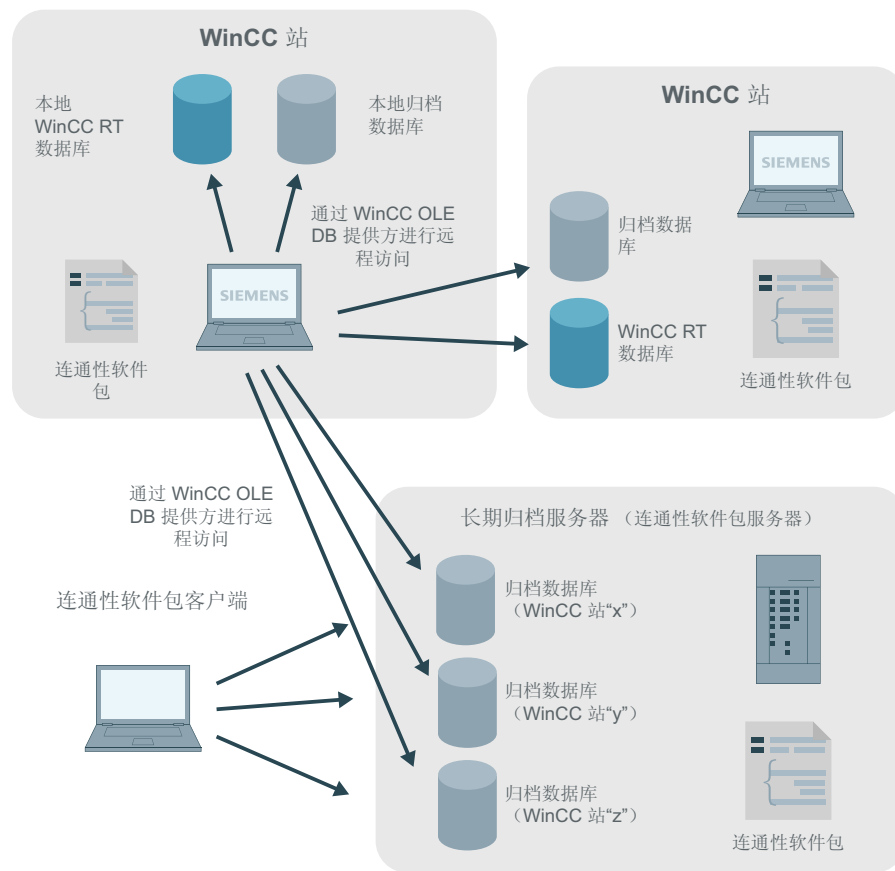
访问可发生在组态方式各不相同的连通性软件包客户端之间。

- 在客户端计算机上存在如 WinCC Basis、Web Navigator 服务器、DataMonitor 服务器或连通性软件包服务器这样的 WinCC 软件。连通性软件包客户端不必显式安装。通过 WinCC 许可证提供许可。
- 客户端计算机上未安装 WinCC 软件。
需要将连通性软件包客户端安装在客户端计算机上。

参见

使用 OLE DB 提供者访问的实例 (页 207)

3.7 使用范例7：通过 WinCC OLE DB Provider 进行访问



原理

使用“SQL Server Import/Export”向导时，可通过 WinCC OLE DB Provider 来访问 WinCC 数据库。

使用“SQL Server Import/Export”向导可从不同的源中提取数据并将其以其它格式（例如 Excel 表）导出。数据库是使用 OLE DB 链接的，并使用 WinCC OLE-DB Provider 访问 WinCC 数据库。可将数据查询保存在 DTSX-Package 中。

DTSX-Package 可绑定在脚本中，用来获取时间控制的查询以及将数据转换为目标格式。同样，也可将任务附于包中，例如，为了在执行包后立即通过邮件启动通知。

使用 WinCC 计算机可以在本地或远程建立对运行系统和归档数据库的访问。如果是长期归档服务器，则只可以在本地或远程访问归档数据库，因为长期归档服务器没有运行系统数据库。

3.7 使用范例 7：通过 WinCC OLE DB Provider 进行访问

软件要求

WinCC 站需要安装以下各项：

- WinCC 基本系统
- 有效的 WinCC RT 许可证
- WinCC 选件连通性软件包的许可证

有关详细信息，请参考“通过 SQL Server Import/Export 向导组态访问”一章。

参见

使用 OLE DB 提供者访问的实例 (页 207)

通过“SQL Server Import/Export”向导组态访问 (页 61)

3.8 使用 OPC 访问 WinCC 归档、变量和消息

简介

OPC 可启用对 WinCC 的在线和归档数据授权的访问。通过 OPC 软件接口，WinCC OPC 服务器将 WinCC 数据提供给 OPC 客户机。作为 OPC 客户机，可执行任何基于相应 OPC 规范的软件。

OPC DA

WinCC OPC DA 服务器为其它应用程序提供了 WinCC 项目的数据。这些应用程序可在本地运行或者在链接到该网络环境的计算机上运行。例如，以这种方法，WinCC 变量可导出到 Microsoft Excel。WinCC-OPC-DA 服务器及其授权是 WinCC 基本系统的组成部分。

OPC XML

在分布式系统中，WinCC 客户机具有多个 WinCC 服务器的视图。WinCC OPC-XML 服务器以 Web 服务形式为 OPC-XML 客户机提供 OPC 过程数据。可使用 HTTP 通过 Internet 访问 Web 服务。OPC XML 客户机不再限于局域网。这样，OPC XML 客户机便可通过任何平台以及通过 Intranet 或 Internet 访问 WinCC 运行系统数据。

OPC HDA

使用 OPC HDA（历史数据访问）服务器，可访问 WinCC 归档系统的历史数据。

不包含写访问的安装，仅可读取和分析 WinCC 归档数据。写访问可分析、添加、删除和更新数据。

使用 OPC HDA 客户机，可对归档数据进行分析 and 评估以及对来自不同 OPC HDA 服务器的归档进行过程控制。

说明

OPC HDA 禁止访问消息归档。通过应用程序，仅可访问过程值归档。

OPC A&E

通过 OPC A&E（报警和事件）服务器，可转发或确认 WinCC 消息。

例如，可使用 OPC A&E 客户机对来自不同 OPC A&E 服务器的报警进行分析和联合归档。

冗余系统中的 WinCC OPC 服务器

在冗余系统中，WinCC 服务器在系统运行期间相互监视，以便尽早识别停止运行的服务器。使用 OPC 软件接口，WinCC OPC 服务器允许 OPC 客户机访问 WinCC 运行系统数据。同时具有数个 WinCC-OPC 服务器视图的 OPC 客户机，可用于集中监视各种冗余系统。

作为 OPC 客户机，可执行任何基于相应 OPC 规范的软件。

有关更多信息，请参阅“OPC - 开放式互连”一章。

说明

通过 OPC 进行归档访问时出现错误 OPC_E_MAXEXCEEDED

如果 OPC 客户机在同步或异步读取数据期间请求的值多于 2,000 个，则会拒绝请求，并出现错误消息 OPC_E_MAXEXCEEDED。此限制用来限制计算机负载和调用的持续时间。

如果读取了整个时间范围，则该限制无效。

3.9 对归档数据的透明访问

简介

过程值和消息的归档用于在工业系统中注册、管理及归档过程数据。

注册的过程数据定期通过 WinCC 站换出到归档服务器（例如“Process Historian”）上。

因此，较长周期的归档过程数据分配到两个归档数据库中。

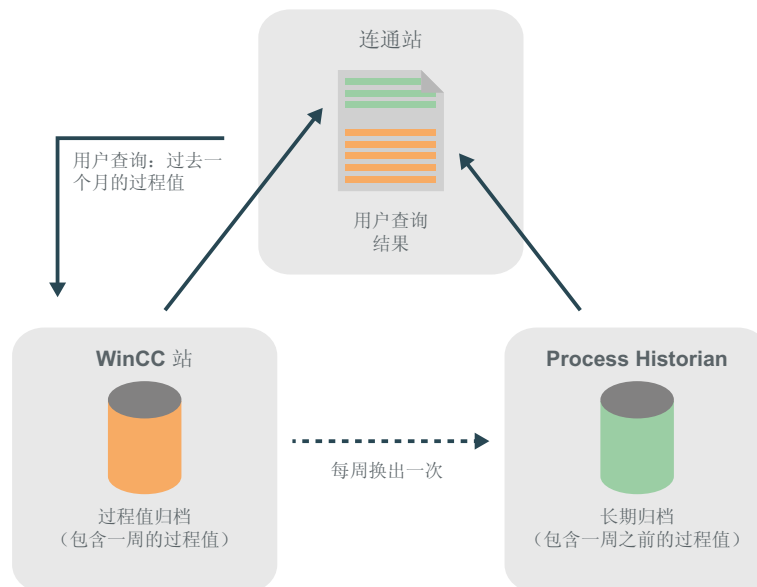
透明访问可确保将从两个归档数据库请求的过程数据结合到一起。

说明

多客户端也可用于透明访问，而不使用连通站。

工作原理

以下画面将通过用户查询上个月过程值的示例说明透明访问的原理：



用户查询的结果显示给用户，就像过程值源于某个数据源一样。

在冗余系统中的透明访问

对于冗余系统，除以下区别外，透明访问功能的原理与上述原理相同：

如果服务器在用户查询期间发生故障，则用户查询将自动被重新路由到冗余伙伴服务器。

3.9 对归档数据的透明访问

使用 OPC 进行透明访问

借助于 OPC，可对 WinCC 站的所有归档数据库进行透明访问：

OPC 服务器	服务器名称
OPC DA	WinCCConnectivity.OPCDAServer
OPC HDA	WinCCConnectivity.OPCHDAServer.1
OPC A&E	WinCCConnectivity.OPCAEServer.1

通过 OLE DB 提供程序进行透明访问

使用 OLE DB 提供程序，可以透明方式访问过程值归档。

对于数据库链接，使用以下语法进行透明访问：

数据源

<符号计算机名称>::\WinCC

目录

WinCC 项目的名称

3.10 SQL Server 的功能

简介

在下面，将有对于在 Microsoft SQL Server 中已实现的重要功能的简述。

有关 SQL Server 2016 的技术文档，请访问 Microsoft 网站，Internet:

- SQL Server 2016 文档 ([https://msdn.microsoft.com/zh-cn/library/hh995091\(v=sql.10\).aspx](https://msdn.microsoft.com/zh-cn/library/hh995091(v=sql.10).aspx))
- SQL Server 2016 概述 (<https://docs.microsoft.com/zh-cn/sql/sql-server/sql-server-technical-documentation?view=sql-server-2016>)

功能

“SQL Server 导入/导出”向导

使用“SQL Server 导入/导出”向导可提取数据，这些数据源自不同的源，并可传送给一个或多个目标。

有关使用范例的说明可在“使用 WinCC OLE DB 提供程序进行访问”一章和“通过‘SQL Server 导入/导出’向导组态访问”一章中找到。

作业和警报

作业代表由 SQL Server 代理按顺序执行的一系列操作。作业可由大量功能组成，例如 Transact SQL 脚本、命令行应用程序和 ActiveX 脚本。通过作业可执行重复任务或时间控制的任務。通过设置“警报”的显示，作业可自动将执行状态通知给用户。

备份

SQL Server 的“备份”组件为 SQL Server 数据库中的关键数据提供了重要保护。

这可以防止因存储介质故障、用户错误或永久丢失服务器所导致的损害。

此外，该组件在其它应用情况下也很有用，例如在不同服务器之间复制数据库，或者通过备份创建数据库的副本以及随后在不同的计算机上进行恢复。

复制

利用“复制”组件可在企业范围内复制、分布及修改数据。因此，SQL 服务器包含多个方法和选项，可用于设计复制、实现、监控以及管理，因而为数据分布和维护数据的一致性提供了所需的功能。

数据库维护计划

创建并补充对象和数据并进行使用后，必须对数据库进行维护。

因此，定期创建数据库的备份或者创建新索引以提高性能非常重要。在创建数据库时，应考

3.10 SQL Server 的功能

考虑到这些方面，以便将对用户的影响降至最低，同时将此类维护所需的时间和资源降至最低。通过使用向导和“数据库维护计划”组件，可创建一个或多个维护计划。

链接的服务器

Microsoft SQL Server 允许使用“链接的服务器”创建与 OLE DB 数据库的连接。例如，实现“链接的服务器”具有以下优点。

- 可从 OLE-DB 数据库访问数据集以及使用 Transact SQL 指令以表格格式进行显示。
- 可将命令传输到 OLE-DB 数据源以及使用 Transact SQL 指令以表格格式显示随后的数据结果。

WinCC OLE DB 提供程序作为“链接的服务器”的使用范例说明可在“WinCC/连通性软件包：简介”一章中找到。

参见

概览：WinCC/连通性软件包 (页 9)

通过“SQL Server Import/Export”向导组态访问 (页 61)

使用范例 7：通过 WinCC OLE DB Provider 进行访问 (页 23)

SQL Server 2016 概述 (<https://docs.microsoft.com/zh-cn/sql/sql-server/sql-server-technical-documentation?view=sql-server-2016>)

SQL Server 2016 文档 ([https://msdn.microsoft.com/zh-cn/library/hh995091\(v=sql.10\).aspx](https://msdn.microsoft.com/zh-cn/library/hh995091(v=sql.10).aspx))

通过 OLE DB 提供者进行访问

4.1 使用 OLE DB 提供程序访问归档数据

简介

通过 OLE DB，可使用以下选项访问 WinCC 归档数据以及使用外部接口显示它们。

使用 WinCC OLE DB 提供程序进行访问

通过 WinCC OLE DB，可访问所有 WinCC 归档数据。

根据组态情况，可采用压缩格式存储 WinCC 的过程数据。通过 WinCC OLE DB 提供程序，甚至可对这些数据进行透明访问。

使用“SQL Server 导入/导出向导”来利用标准 SQL 查询。可使用向导将解压的文件保存到中间数据库中；使用标准 SQL 查询来访问数据库。

使用 Microsoft OLE DB 进行访问

通过 Microsoft OLE DB，可访问所有 WinCC 用户归档。

说明

测试和发布的 Microsoft OLE DB 仅用于访问 WinCC 用户归档，而不用于访问报警和过程值归档。

使用 WinCC OLE DB 提供程序访问消息和过程值归档。

组态选项

为使用 WinCC OLE DB 访问数据库，可编写您自己的应用程序。为与 WinCC OLE DB 提供程序进行通讯，可通过诸如 Visual Basic、VBScript 或 VBA 创建的应用程序来使用 ADO DB。

说明

变量名称中的特殊字符

请注意，编程语言如 Visual Basic、VBScript 或 VBA 仅允许在变量名称中存在以下字符：“A...Z”、“a...z”、“0...9”以及“_”。

在 WinCC 中，如果在变量名称中使用特殊字符如“/”或“;”，则脚本将中止并出现错误消息。这种情况下，可使用“变量 ID”访问其脚本名称中包含特殊字符的变量。

4.1 使用 OLE DB 提供程序访问归档数据

通常步骤

1. 为访问归档数据，计算机上必须已经安装 WinCC Basic、连通性软件包服务器或连通性软件包客户机。
2. 对于换出的归档，可使用 WinCC 归档连接器在 SQL 数据库和换出的归档之间建立连接。

说明

在目录“<项目目录>\ArchiveManager”及其相关子目录中的 WinCC RT 归档不得与归档连接器连接或者从归档连接器断开连接，因为它们与 SQL server 的连接由 WinCC 基本系统管理。

3. 例如，通过使用 MS Excel 或者您自己的应用程序建立与该数据库的连接。定义需要的选择标准，然后读取归档数据。
4. 例如，查询结果可能以 MS Excel 显示或者作为 csv 文件导出。

参见

通过“SQL Server Import/Export”向导组态访问 (页 61)

查询归档数据 (页 43)

建立到归档数据库的连接 (页 39)

WinCC 归档连接器 (页 35)

OLE DB 的基础 (页 33)

4.2 OLE DB 的基础

简介

使用 OLE DB 接口和 WinCC 提供的相关的数据库提供者，可访问过程值和消息归档。

OLE DB

OLE DB 是用于快速访问不同数据库的公开标准。它与数据库是否为关系数据库无关。

通过数据库提供者，在 OLE DB 级别和数据库之间建立连接。

OLE DB 接口和提供者由不同的制造商提供。

WinCC OLE DB 提供者

使用 WinCC OLE DB 提供者，可直接访问存储在 MS SQL server 数据库中的 WinCC 归档数据。根据组态情况，可以压缩格式存储 WinCC 的过程数据。通过 WinCC OLE DB 提供者，甚至可对这些数据进行透明访问。

说明

如果 WinCC 关闭了整个归档并打开一个新归档，则没有来自消息和过程值归档的数据即刻通过 OLE DB 提供者读取。

Microsoft OLE DB

通过 Microsoft OLE DB，仅可访问 WinCC 用户归档。

为防止未经授权使用 MS OLE DB 进行访问，数据库的管理人员可采取适当的措施。附加信息可在“使用 MS OLE DB 访问 SQL 数据库期间的安全设置”一章中找到。

说明

测试和发布的 Microsoft OLE DB 仅用于访问 WinCC 用户归档，而不用于访问报警和过程值归档。

使用 WinCC OLE DB 提供程序访问消息和过程值归档。

4.2 OLE DB 的基础

参见

使用 MS OLE DB 访问 SQL 数据库期间的安全设置 (页 71)

使用 OLE DB 提供程序访问归档数据 (页 31)

4.3 WinCC 归档连接器

简介

WinCC“归档连接器”用于组态对归档数据库的访问。该工具是 WinCC DataMonitor 和连通性软件包的主要部分。通过归档连接器，可将已经换出的 WinCC 归档重新连接到 SQL Server。然后，DataMonitor 客户机或 WinCC OLE DB 提供程序可访问这些归档。

WinCC 归档连接器的功能如下：

- 手动连接：可选择本地数据库，然后将其连接到本地 SQL server。
- 手动断开：可选择已连接的数据库，然后断开其与 SQL server 的连接。
- 自动连接：可选择从中导出 WinCC 归档的本地目录。所有归档将自动链接到 SQL server，其从激活更改监控时开始已添加到所选目录中。

只有使用本地 SQL Server 和 WinCC DataMonitor 或 WinCC 连通性软件包的许可证，才能运行归档连接器。

只要已完成组态，即可终止归档连接器。

说明

在目录“<项目目录>\ArchiveManager”及其相关子目录中的 WinCC RT 归档不得与归档连接器连接或者从归档连接器断开连接，因为它们与 SQL server 的连接由 WinCC 基本系统管理。在 WinCC 中通过归档组态器，而不是通过 WinCC 归档连接器设置换出的 WinCC 归档的路径，例如变量记录的路径。

如果访问位于可互换介质（例如磁带或 MOD 驱动器）上的换出的归档，请注意要在更换驱动器中介质前使用归档连接器断开到该介质上这些归档的连接。更换介质后，用户应通过归档连接器检查是否已经连接了新介质上的归档。

只允许有限范围的人员访问 WinCC 归档连接器的组态。因此，应使用 Windows 用户授权“管理员”或其它 Windows 保护措施（如存储在受保护的目录中）保护对该工具的访问。

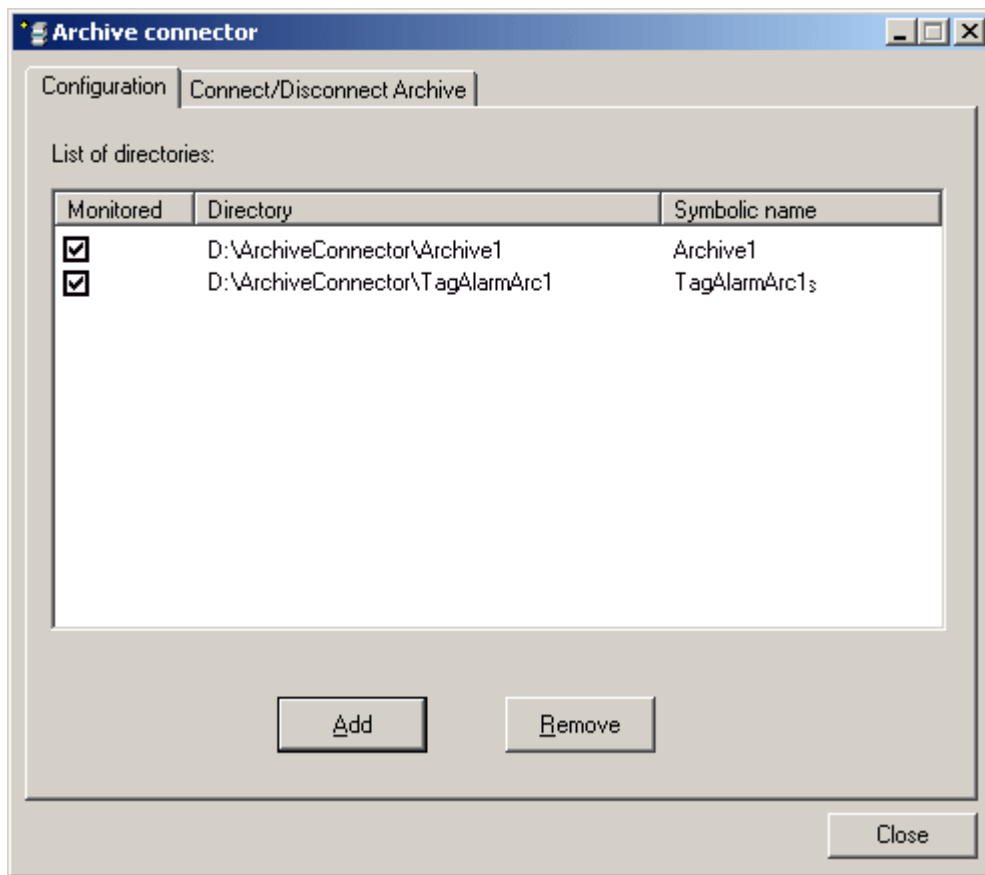
说明

使用归档连接器通过连通性软件包或 DataMonitor 访问链接的归档。

以下对象不能访问使用归档连接器链接的归档：

- WinCC 报警控件
 - WinCC 在线趋势控件
 - WinCC 在线表格控件
 - WinCC 在线功能控件
-

“组态”选项卡



在“组态”选项卡中，显示并管理归档文件夹，这些文件夹可通过 Web 或 WinCC OLE DB 提供程序进行访问。

通过按钮，可添加或删除归档文件夹。在组态期间，必须为每个归档文件夹指派一个唯一的符号名称。

DataMonitor 客户机或 WinCC OLE DB 提供程序使用符号名称访问归档。

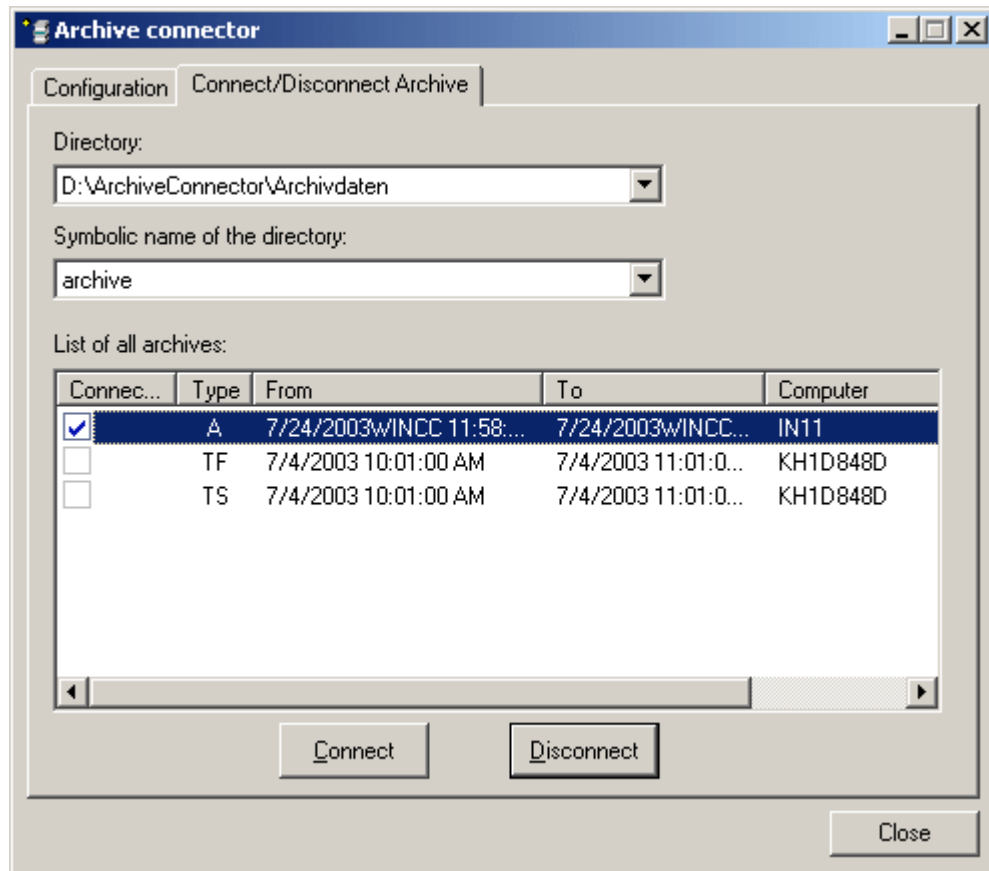
在多个计算机或项目上管理和连接导出的数据时，也使用该名称。

符号名称只能包含 SQL 语法允许的字符。

通过激活相应的复选框，所有在激活时添加到所选文件夹的归档均将自动连接到 SQL Server。

如果激活或禁用监视功能，则在关闭归档连接器之前更改将不会生效。

“连接/断开归档”选项卡



“连接/断开归档”选项卡列出所有位于归档目录中的归档。

显示每个归档的连接状态。通过按钮，可建立或终止到归档的连接。

归档类型显示在“类型”域中：

- “A”= 报警记录；
- “TF”= 变量记录（高速）；
- “TS”= 变量记录（低速）。

4.3 WinCC 归档连接器

“从”和“到”列提供有关本地时区的信息。

说明

将多个换出的归档连接到 SQL server 可能需要数秒钟。

无法使用同一名称两次连接数据库文件。

WinCC 归档连接器将最终确定以及经过备份（换出）的归档连接到 SQL server。不支持未最终确定的归档。

归档连接器的用户界面语言取决于 Windows 中区域和语言选项的设置。

说明

对于通过 SQL Server 2000 创建的归档，只有移除归档的写保护，才能连接此类归档。在 SQL Server 2000 条件下，不能再使用连通性软件包的归档连接器链接曾链接过的归档。

为连接 CD 或 DVD 上的已换出归档，可将该归档复制到能够移除写保护的介质上。

4.4 建立到归档数据库的连接

简介

对于 ActiveX 数据对象 (ADO)，通过连接对象建立应用程序和归档数据库之间的连接。

这里的一个重要参数是 `ConnectionString`。

`ConnectionString` 包含使用 OLE DB 提供程序访问数据库的所有必需信息。

4.4 建立到归档数据库的连接

ConnectionString 的结构

```
"Provider = Name of the OLE DB Provider; Catalog=Database
name;Data Source=Server name;"
```

参数	描述
提供程序	OLE DB 提供程序的名称，例如： <ul style="list-style-type: none"> WinCCOLEDBProvider
目录	WinCC 数据库的名称 使用 WinCC RT 数据库时，将使用以“R”结尾的数据库名称： <ul style="list-style-type: none"> <Database name_R> 也可以使用数据库“CC_ExternalBrowsing”。 如果已经通过 WinCC 归档连接器将换出的 WinCC 归档连接到 SQL 服务器，请使用其符号名称。 注 对于透明访问，为“Catalog”输入 WinCC 项目名称；例如： <ul style="list-style-type: none"> Catalog=WinCC_Project_Name 注 如果通过“CC_ExternalBrowsing”访问消息归档或换出的归档，则该访问可能需要数分钟的时间。
数据源	服务器名称 本地： <ul style="list-style-type: none"> “.WinCC”或“<计算机名称>\WinCC” 远程： <ul style="list-style-type: none"> <计算机名称>\WinCC 注 如果对归档服务器进行透明访问，或者通过 OLE DB 提供程序访问冗余服务器，则为“数据源”(Data Source) 输入以下内容： <ul style="list-style-type: none"> <符号计算机名称>:\WinCC。 注 使用归档变量名称可直接访问包含 8 字节值 ID 的数据库（例如归档服务器上的数据库）中的归档变量。 在这种情况下，归档服务器会返回服务器 ID 作为 ID，而不返回归档变量 ID： <ul style="list-style-type: none"> <符号计算机名称>\<Archive_Var_Name>

过程值和消息归档实例：

在以下实例中，连接对象通过随后打开到 WinCC 数据库（过程值和消息归档）的连接来创建。

```
Set conn = CreateObject("ADODB.Connection")

conn.open
"Provider=WinCCOLEDBProvider.1;Catalog=CC_OpenArch_03_05_27_14_11_46R;Data Source=.\WinCC"
```

用户归档实例：

在以下实例中，连接对象通过随后打开到 WinCC 用户归档的连接来创建。

```
Set conn = CreateObject("ADODB.Connection")

conn.open "Provider=SQLNCLI11; Integrated Security=SSPI; Persist
Security Info=false; Initial
Catalog=CC_OpenArch_03_05_27_14_11_46R; Data Source=.\WinCC"
```

说明

为了在本地访问期间提高性能，请输入“<计算机名称>\WinCC”作为数据源，而不是“.\WinCC”。

参见

实例： 使用 DataConnector Wizard 组态对归档数据的访问 (页 210)

示例： 使用 VB 组态对归档数据的访问 (页 208)

用户归档的查询 (页 58)

查询报警消息归档 (页 53)

查询过程值归档 (页 45)

4.5 查询归档数据

4.5 查询归档数据

4.5.1 显示过程值归档

简介

查询结果作为记录集返回。

在本节中，介绍用于过程值归档的记录集结构。

扩展的“TAG_EX”架构

要访问以下数据类型的变量，请使用扩展的“TAG_EX”架构：

- String
- Int64
- UInt64

出于兼容性考虑，用户还可以使用“TAG_EX”来访问通常使用“TAG”架构访问的变量。

如果使用“TAG”架构读取 String 值，则返回值“0”。这意味着所有文本变量的值都为“0”。

记录集结构

架构“TAG:R”

字段名称	类型	注释
ValueID	整型（4 字节） 或 整型（8 字节）	值的唯一标识。 长度取决于查询的类型。
TimeStamp	日期时间	时间戳
RealValue	实型（8 字节）	变量值
Quality	整型（4 字节）	值的质量代码（例如“好”或“劣”）。
Flags	整型（4 字节）	内部控制参数

架构“TAG_EX:R”

字段名称	类型	注释
ValueID	整型（4 字节） 或 整型（8 字节）	值的唯一标识。 长度取决于查询的类型。
TimeStamp	日期时间	时间戳
TimeStampExt	日期时间	外部设备的时间戳
VariantValue	变体（16 字节）	变量值
Bytes	实型（8 字节）	变量值
Quality	整型（4 字节）	值的质量代码（例如“好”或“劣”）。
Flags	整型（4 字节）	内部控制参数

参见

查询归档数据 (页 43)

建立到归档数据库的连接 (页 39)

使用 OLE DB 提供程序访问归档数据 (页 31)

OLE DB 的基础 (页 33)

4.5.2 查询归档数据

简介

查询通过命令对象转发到数据库。除“ConnectionString”之外的一个重要参数就是 CommandText。CommandText 传送查询。查询结果作为记录集返回。

说明

消息和过程值归档查询的时间范围

如果消息或过程值归档的查询指定了某个时间范围，而归档内并不存在该时间范围的消息或其它值，则不会出现信息消息或其它状态显示。如果要显示此状态，必须由用户执行错误处理。

此错误处理例程的简单描述在“示例：通过 WinCC OLE DB 提供程序读取消息归档数据”下的示例脚本中做了介绍。

4.5 查询归档数据

在以下实例中，生成每个命令对象，并且将查询作为 CommandText 传送。

在以下结构实例中，CommandText 也包含 ConnectionString，其结构在“建立到归档数据库的连接”中进行介绍。

CommandText 的结构

过程值归档:

```
Set oRs = CreateObject("ADODB.Recordset")
Set oCom = CreateObject("ADODB.Command")
oCom.CommandType = 1
Set oCom.ActiveConnection = conn
oCom.CommandText = "TAG:R,'PVArchive\Tag1','0000-00-00
00:10:00.000','0000-00-00 00:00:00.000'"
```

报警消息归档:

```
Set oRs = CreateObject("ADODB.Recordset")
Set oCom = CreateObject("ADODB.Command")
oCom.CommandType = 1
Set oCom.ActiveConnection = conn
oCom.CommandText = "ALARMVIEW:Select * FROM AlgViewEnu"
```

用户归档

```
Set oRs = CreateObject("ADODB.Recordset")
Set oCom = CreateObject("ADODB.Command")
oCom.CommandType = 1
Set oCom.ActiveConnection = conn
oCom.CommandText = "SELECT * FROM UA#Test"
```

指定记录集位置

要指定记录集的位置以便查询归档数据，需要将“CursorLocation”属性的值设置为“3”，例如，“conn.CursorLocation = 3”。即可在客户机上创建记录集。

参见

实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取过程值归档 (页 215)

显示用户归档 (页 60)

显示报警消息归档 (页 55)

显示过程值归档 (页 42)

用户归档的查询 (页 58)

查询报警消息归档 (页 53)

查询过程值归档 (页 45)

建立到归档数据库的连接 (页 39)

WinCC 归档连接器 (页 35)

OLE DB 的基础 (页 33)

使用 OLE DB 提供者访问的实例 (页 207)

4.5.3 查询过程值归档

原理

通过以下查询，可访问过程值归档。

可使用过滤标准选择数据。

查询通过命令对象转发到数据库。

说明

ValueID 的长度

ValueID 的长度可以不同：

- 对于在中央归档服务器上处理的数据库，ValueID 的长度为 8 字节，HI-DWORD 区域中包含由相应服务器分配的 ValueID，LO-DWORD 区域中包含服务器 ID。
- 对于其它所有数据库，ValueID 的长度为 4 字节，仅包含由 WinCC 服务器分配的唯一 ValueID。

在任何情况下，仍然可以兼容通过 TAG:R 和 TAG_EX:R 进行的 4 字节查询。如果数据库的 ValueID 为 8 字节，在此返回的 4 字节 ValueID 不再是唯一的。

对过程值归档的查询，最多限于 20 个变量，对每个变量的查询最多为 128 个字符。

4.5 查询归档数据

语法

请注意，查询不能包含任何空格。

8 字节长 ValueID 的请求：

```
TAG_LLVID:R,<ValueID 或 ValueName>,<TimeBegin>,<TimeEnd>[,<SQL_clause>][,<TimeStep>]
```

4 字节长 ValueID 的请求：

```
TAG:R,<ValueID 或 ValueName>,<TimeBegin>,<TimeEnd>[,<SQL_clause>][,<TimeStep>]
```

架构“TAG_EX”

要访问以下变量类型的过程值，请使用“TAG_EX:R”架构：

- 文本变量 8 位字符集/16 位字符集
- 64 位 IEEE 754 浮点数

将“TAG_LLVID:R”替换为“TAG_LLVID_EX:R”。

参数

参数	描述
ValueID	<p>数据库表的值 ID</p> <p>可以有多个名称，例如：</p> <ul style="list-style-type: none"> • "TAG:R,(ValueID_1;ValueID_2;ValueID_x),<TimeBegin>,<TimeEnd>"
ValueName	<p>ValueName 的格式为“ArchiveName\Value_Name”。</p> <p>必须将参数 <ValueName> 括在单引号中。</p> <p>可以有多个名称，例如：</p> <ul style="list-style-type: none"> • "TAG:R,('ValueName_1';'ValueName_2';'ValueName_x'), <TimeBegin>,<TimeEnd>" <p>注</p> <p>请注意，编程语言如 Visual Basic、VBScript 或 VBA 仅允许在变量名称中存在以下字符：</p> <ul style="list-style-type: none"> • “A...Z”、“a...z”、“0...9”以及“_” <p>在 WinCC 中，如果在变量名称中使用特殊字符如“,”或“;”，则脚本将中止并出现错误消息。这种情况下，可使用“变量 ID”访问其脚本名称中包含特殊字符的变量。</p>

参数	描述
TimeBegin	<p>开始时间采用以下格式：</p> <ul style="list-style-type: none"> 'YYYY-MM-DD hh:mm:ss.msc' <p>使用 <TimeStep> 时，必须将 <TimeBegin> 指定为绝对时间。</p> <p>禁止使用相对语句“0000-00-00 00:00:00.000”。</p>
TimeEnd	<p>结束时间采用以下格式：</p> <ul style="list-style-type: none"> 'YYYY-MM-DD hh:mm:ss.msc'
SQL_Clause	<p>SQL 语法中的过滤标准：</p> <ul style="list-style-type: none"> [WHERE search_condition] [ORDER BY {order_expression [ASC DESC] }] <p>“ORDER BY”标准仅可用于给出的排序顺序“{order_expression [ASC DESC] }”。</p> <p>实例：以下查询返回变量“ValueName_1”和“ValueName_2”的所有值，这些值低于 50 或高于 100。</p> <ul style="list-style-type: none"> "TAG:R,('ValueName_1';'ValueName_2'),<TimeBegin>,<TimeEnd>, 'WHERE RealValue > 100 OR RealValue < 50'"

4.5 查询归档数据

参数	描述		
TimeStep	将汇总在声明的时间间隔中的值，从开始时间 <TimeBegin> 开始。 <ul style="list-style-type: none"> 格式: 'TIMESTEP=x,y' x = 间隔 (单位为秒) y = 聚合类型, 定义间隔结果 下列值可用于汇总类型:		
	无插补	有插补	描述
	1 (FIRST)	257 (FIRST_INTERPOLATED)	第一个值
	2 (LAST)	258 (LAST_INTERPOLATED)	最后一个值
	3 (MIN)	259 (MIN_INTERPOLATED)	最小值
	4 (MAX)	260 (MAX_INTERPOLATED)	最大值
	5 (AVG)	261 (AVG_INTERPOLATED)	中值
	6 (SUM)	262 (SUM_INTERPOLATED)	总和
	7 (COUNT)	263 (COUNT_INTERPOLATED)	值数目
	无插补意味着: <ul style="list-style-type: none"> 如果在时间间隔中不存在任何值, 则不返回时间间隔结果。 有插补意味着: <ul style="list-style-type: none"> 如果在时间间隔中不存在任何值, 则将由非空的邻近时间间隔的结果线性插补得出一个值。不使用外插法。 实例: <p>例如 TIMESTEP=60,257, 对于每个 60 秒的时间间隔, 将返回此时间间隔的第一个值; 如果此时间间隔中没有任何值, 则返回邻近的时间间隔中第一个值的线性插补值。</p> <ul style="list-style-type: none"> "TAG:R,1,'2004-07-09 09:03:00.000','0000-00-00 00:10:00.000','TIMESTEP=60,257" 		

说明

<TimeBegin> 和 <TimeEnd>

<TimeBegin> 和 <TimeEnd> 不得同时为“零”，即“0000-00-00 00:00:00.000”。

性能: 使用 ValueID 代替 ValueName

为了提高性能, 请在查询期间用参数“ValueID”代替“ValueName”。

在“归档”表上可确定“ValueID”。

时间单位不到 1 秒

某些应用程序不能以 1 ms 的步长处理过程值, 这可能会导致结果不准确。

可以在 VB 脚本“SplitDateAndMs”下找到从过程值的时间戳中取消分组毫秒的示例:

- “实例: 使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取过程值归档 (页 215)”

绝对时间间隔的选择

从开始时间 <TimeBegin> 开始读取，到结束时间 <TimeEnd> 为止。

实例 A1:

从开始时间 9:03 到结束时间 9:10 读取 ValueID 1 的值。

```
"TAG:R,1,'2004-07-09 09:03:00.000','2004-07-09 09:10:00.000'"
```

相对时间间隔的选择

从记录开始时读取:

```
<TimeBegin> = '0000-00-00 00:00:00.000'
```

读取到记录结束为止:

```
<TimeEnd> = '0000-00-00 00:00:00.000'
```

<TimeBegin> 和 <TimeEnd> 不得同时为“零”，即“0000-00-00 00:00:00.000”。

说明

相对时间段: 不使用月份

使用以下格式，输入要在链接的归档数据库中进行查询的相对期间:

- 0000-00-DD hh:mm:ss.msc

如果以月表示时间范围，则其内容可能是错误的，因为一个月可包括 28 到 31 天。

实例 B1:

从“TimeBegin”到记录结束（即最后一个归档值）为止，读取绝对时间。

```
<TimeBegin> = '2003-02-02 12:00:00.000', <TimeEnd> = '0000-00-00  
00:00:00.000'
```

实例 B2:

从“TimeBegin”开始连续 10 秒读取绝对时间。

```
<TimeBegin> = '2003-02-02 12:00:00.000', <TimeEnd> = '0000-00-00  
00:00:10.000'
```

实例 B3:

从“TimeEnd”开始，反向连续 10 秒读取绝对时间。

```
<TimeBegin> = '0000-00-00 00:00:10.000', <TimeEnd> = '2003-02-02  
12:00:00.000'
```

4.5 查询归档数据

实例 B4:

从多个 valueID (1;3;5;6) 的最后一个归档值开始，读取最后一小时的值。

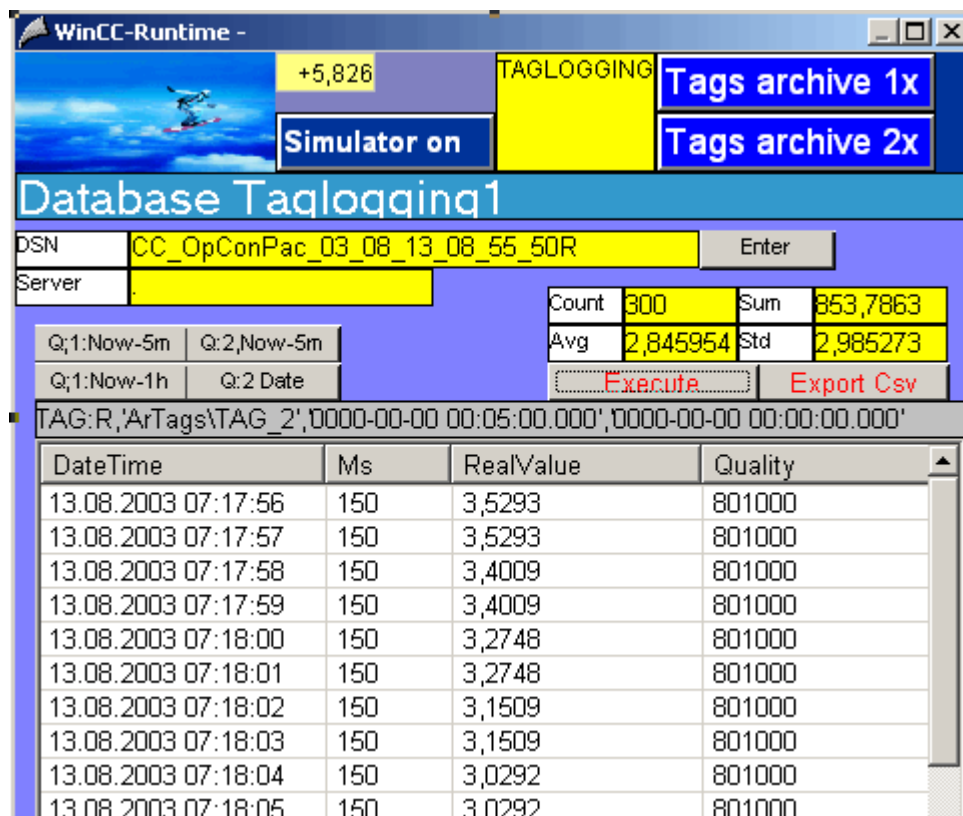
```
"TAG:R, (1;3;5;6), '0000-00-00 01:00:00.000', '0000-00-00  
00:00:00.000'"
```

实例 B5:

从“ArTags”归档的“TAG_2”变量的最后一个归档值开始，读取最后五分钟的值。

```
"TAG:R, 'ArTags\TAG_2', '0000-00-00 00:05:00.000', '0000-00-00  
00:00:00.000'"
```

下图显示了此示例的可能结果:



对变量值使用过滤器进行多个返回值的查询

实例 C1:

以下查询也使用 <SQL_Clause> 参数，并且返回所有包含 ValueID“3”、“6”并且小于 50 或大于 100 的变量值。

```
"TAG:R, (3;6), <TimeBegin>, <TimeEnd>, 'WHERE RealValue > 100 OR
RealValue < 50'"
```

带参数 <TimeStep> 的查询

实例 C2:

以下查询使用 <TimeStep> 参数，并且返回所有 ValueID“1”的值，从开始时间“TimeBegin”开始，直到 5 分钟后为止，时间间隔为“60”秒，汇总类型“5”=“无插补的平均值”。

```
"TAG:R, 1, '2004-10-13 17:00:00.000', '0000-00-00 00:05:00.000',
'TIMESTEP=60, 5'"
```

下图显示了查询结果:

- 左侧表格显示了在某个 30 秒的归档周期内归档的归档数据。
- 右侧表格显示了查询结果。
它决定了“0”秒和“30”秒时的两个归档值的平均值，通过平均间隔的第一个时间戳（即“0”秒）显示。

TIMESTEP = 60,5

ValueID	Datetime	RealValue	Quality
1	2004-10-13 17:00:00.000	3.0	801000
1	2004-10-13 17:00:30.000	4.0	801000
1	2004-10-13 17:01:00.000	2.0	801000
1	2004-10-13 17:01:30.000	4.0	801000
1	2004-10-13 17:02:00.000	4.0	801000
1	2004-10-13 17:02:30.000	1.0	801000
1	2004-10-13 17:03:00.000	3.0	801000
1	2004-10-13 17:03:30.000	3.0	801000
1	2004-10-13 17:04:00.000	1.0	801000
1	2004-10-13 17:04:30.000	0.0	801000
1	2004-10-13 17:05:00.000	3.0	801000

ValueID	Datetime	RealValue	Quality
1	2004-10-13 17:00:00.000	3.5	801000
1	2004-10-13 17:01:00.000	3.0	801000
1	2004-10-13 17:02:00.000	2.5	801000
1	2004-10-13 17:03:00.000	3.0	801000
1	2004-10-13 17:04:00.000	0.5	801000
1	2004-10-13 17:05:00.000	3.0	801000

实例 C3:

以下查询使用 <TimeStep> 参数，并且返回所有 ValueID“1”和“2”的值，从开始时间“TimeBegin”开始，直到 2 分钟后为止，时间间隔为“15”秒，汇总类型“261”=“有线性插补的平均值”。

```
"TAG:R, (1;2), '2004-10-13 17:00:00.000', '0000-00-00
00:02:00.000', 'TIMESTEP=15.261'"
```

4.5 查询归档数据

下图显示了查询结果：

- 左侧表格显示了在某个 30 秒的归档周期内归档的归档数据。
- 右侧表格显示了查询结果。

对于在查询结果中显示的“0”秒和“30”秒时的归档值，其时间戳不变。

“15”秒时，线性插补值由在“0”秒和“30”秒时的归档值形成。

“45”秒时，线性插补值取自同一分钟的“30”秒时的归档值和下一分钟的“0”秒时的归档值。

TIMESTEP = 15, 261



ValueID	Datetime	RealValue	Quality
1	2004-10-13 17:00:00.000	3,0	801000
2	2004-10-13 17:00:00.000	12,0	801000
1	2004-10-13 17:00:30.000	4,0	801000
2	2004-10-13 17:00:30.000	14,0	801000
1	2004-10-13 17:01:00.000	2,0	801000
2	2004-10-13 17:01:00.000	16,0	801000
1	2004-10-13 17:01:30.000	4,0	801000
2	2004-10-13 17:01:30.000	17,0	801000
1	2004-10-13 17:02:00.000	4,0	801000
2	2004-10-13 17:02:00.000	14,0	801000
1	2004-10-13 17:02:30.000	1,0	801000
2	2004-10-13 17:02:30.000	13,0	801000
1	2004-10-13 17:03:00.000	3,0	801000
2	2004-10-13 17:03:00.000	13,0	801000

ValueID	Datetime	RealValue	Quality
1	2004-10-13 17:00:00.000	3,0	801000
1	2004-10-13 17:00:15.000	3,5	801000
1	2004-10-13 17:00:30.000	4,0	801000
1	2004-10-13 17:00:45.000	3,0	801000
1	2004-10-13 17:01:00.000	2,0	801000
1	2004-10-13 17:01:15.000	3,0	801000
1	2004-10-13 17:01:30.000	4,0	801000
1	2004-10-13 17:01:45.000	4,0	801000
1	2004-10-13 17:02:00.000	4,0	801000
2	2004-10-13 17:00:00.000	12,0	801000
2	2004-10-13 17:00:15.000	13,0	801000
2	2004-10-13 17:00:30.000	14,0	801000
2	2004-10-13 17:00:45.000	15,0	801000
2	2004-10-13 17:01:00.000	16,0	801000
2	2004-10-13 17:01:15.000	16,5	801000
2	2004-10-13 17:01:30.000	17,0	801000
2	2004-10-13 17:01:45.000	15,5	801000
2	2004-10-13 17:02:00.000	14,0	801000

参见

实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取过程值归档 (页 215)

显示过程值归档 (页 42)

通过“SQL Server Import/Export”向导组态访问 (页 61)

4.5.4 查询报警消息归档

简介

通过以下查询，可访问消息归档。

可使用过滤标准选择数据。查询通过命令对象转发到数据库。

有关消息状态的更多信息，请参见 WinCC 信息系统的以下位置：

- “使用 WinCC > 用于创建函数和动作的 ANSI-C 函数 > ANSI-C 函数描述 > 附录 > 结构定义 > 结构定义 MSG_RTDATA_STRUCT”。

查询消息归档时，结果按归档汇总，但不对查询的归档部分排序。

如果要对分段进行排序，则需要相应地扩展过滤条件，例如，按时间排序“ORDER BY DateTime ASC, MS ASC”。

语法

```
ALARMVIEWEX:SELECT * FROM <ViewName>[WHERE <Condition>.....,  
optional]
```

4.5 查询归档数据

参数

参数	说明
ViewName	<p>数据库表的名称。</p> <p>必须用需要的语言指定该表。</p> <p>例如，五种欧洲语言的“ViewName”是：</p> <ul style="list-style-type: none"> • ALGVIEWEXDEU: 德语消息归档数据 • ALGVIEWEXENU: 英语消息归档数据 • ALGVIEWEXESP: 西班牙语消息归档数据 • ALGVIEWEXFRA: 法语消息归档数据 • ALGVIEWEXITA: 意大利语消息归档数据 <p>例如，亚洲语言的“ViewName”是：</p> <ul style="list-style-type: none"> • ALGVIEWEXCHS: 中文（简体）消息归档数据 • ALGVIEWEXCHT: 中文（繁体）消息归档数据 • ALGVIEWEXJPN: 日语消息归档数据 • ALGVIEWEXKOR: 韩语消息归档数据 <p>注</p> <p>支持在 WinCC 基本系统中安装的语言或者在 WinCC 文本库中组态的语言。有关可接受的查询语言或相应的“ViewName”的信息可在 SQL-Server 链接报警归档的“视图”下找到。相应归档中支持的所有语言在此处显示时都带有其 ID，例如“ALGVIEWEXENU”。</p>
条件	<p>筛选标准，例如：</p> <ul style="list-style-type: none"> • DateTime>'2003-06-01' AND DateTime<'2003-07-01' • DateTime>'2003-06-01 17:30:00' • MsgNr = 5 • MsgNr 在 (4, 5) 中 • State = 2 <p>对于 DateTime，仅可使用绝对时间指示。</p>

示例 1:

读取 2003 年 7 月 5 日后记录的第 5 个消息的所有条目。

```
"ALARMVIEWEX:SELECT * FROM ALGVIEWEXENU WHERE MsgNr = 5 AND
DateTime>'2003-07-05'"
```

示例 2:

读取时间戳在 2003 年 7 月 3 日和 2003 年 7 月 5 日之间的所有消息。

```
"ALARMVIEWEX:SELECT * FROM ALGVIEWEXENU WHERE
DateTime>'2003-07-03' AND DateTime<'2003-07-05'"
```

下图显示了此实例的可能结果。

Database Alarms			
DSN	CC_OpConPac_03_07_30_15_59_32R	Enter	
Server		Count	4204
		Sum	287411
F:All	F:MsgNr=5	Avg	68,36608
F:State=2	F:'Date1'< D<'Date2'	Std	216,5581
		Execute	Csv + Hitlist
WHERE DateTime>'2003-07-03' AND DateTime <'2003-07-05'			
DateTime	MsgNr	State	Typename
08.07.2003 15:21:37	2	1	Alarm
08.07.2003 15:21:59	1	1	Alarm
08.07.2003 15:22:21	1	2	Alarm
08.07.2003 15:22:21	2	2	Alarm
08.07.2003 15:22:21	3	1	Warnung
08.07.2003 15:22:43	1	16	Alarm
08.07.2003 15:22:43	1	1	Alarm
08.07.2003 15:23:05	2	16	Alarm
08.07.2003 15:23:05	2	1	Alarm
08.07.2003 15:23:05	1	2	Alarm

参见

显示报警消息归档 (页 55)

4.5.5 显示报警消息归档

简介

查询结果作为记录集返回。

表中包含用于消息归档的“记录集”结构。

有关消息状态的信息，请参见《WinCC 信息系统》中“使用 WinCC>用于创建函数和动作的 ANSI-C 函数 > ANSI-C 函数描述 > 结构定义 > 结构定义 MSG_RTDATA_STRUCT”。

4.5 查询归档数据

“ALARMVIEWEX”的记录集结构

Position	域名称	类型	注释
1	MsgNr	整型 (4 字节)	消息号
2	State	小整型 (2 字节)	消息状态
3	DateTime	日期时间 (8 字节)	消息的时间戳 (无毫秒的日期/时间)
4	Ms	小整型 (2 字节)	消息的时间戳 (毫秒)
5	Instance	VarChar (255)	消息的实例名称
6	Flags1	整型 (4 字节)	(仅限于内部使用)
7	PValueUsed	整型 (4 字节)	使用的过程值
8 到 17	PValue1 到 PValue10	实型 (8 字节)	数字过程值 1 到 10
18 到 27	PText1 到 PText10	VarChar (255)	过程值文本 1 到 10
28	Computer name	VarChar (255)	Computer name
29	应用程序	VarChar (255)	应用程序名称
30	Comment	VarChar (255)	注释
31	Username	VarChar (255)	用户名
32	Counter	整型 (4 字节)	运行消息计数器
33	TimeDiff	整型 (4 字节)	与“进入”状态的时间差
34	Classname	VarChar (255)	消息类别的名称
35	Type name	VarChar (255)	消息类型的名称
36	Class	小整型 (2 字节)	消息类别 ID
37	Type	小整型 (2 字节)	消息类型 ID
38 到 47	Text1 到 Text10	VarChar (255)	消息文本 1 到 10
48	AG_NR	小整型 (2 字节)	PLC 号
49	CPU_NR	小整型 (2 字节)	CPU 编号
50	CrForeColor	整型 (4 字节)	“进入”状态的前景色
51	CrBackColor	整型 (4 字节)	“进入”状态的背景色
52	优先级	整型 (4 字节)	优先级
53	AP_type	整型 (4 字节)	报警循环
54	AP_name	VarChar (255)	报警循环函数名称

Position	域名称	类型	注释
55	AP_PAR	VarChar (255)	报警循环画面
56	InfoText	VarChar (255)	信息文本
57	StateText	VarChar (255)	确认状态
58	AlarmTag	整型 (4 字节)	消息变量
59	AckType	小整型 (2 字节)	确认类型
60	Params	整型 (4 字节)	参数
61	服务器名称	VarChar (255)	服务器名称

“ALARMVIEW”的记录集结构（移植项目）

Position	字段名称	类型	注释
1 到 49	参见 ALARMVIEWEX		
50	CrComeFor	整型 (4 字节)	“进入”状态的前景色
51	CrComeBack	整型 (4 字节)	“进入”状态的背景色
52	CrGoFor	整型 (4 字节)	“离开”状态的前景色
53	CrGoBack	整型 (4 字节)	“离开”状态的背景色
54	CrAckFor	整型 (4 字节)	“已确认”状态的前景色
55	CrAckBack	整型 (4 字节)	“已确认”状态的背景色
56	LocalID	整型 (4 字节)	报警的位置
57	Priority	整型 (4 字节)	优先级
58	AP_type	整型 (4 字节)	报警循环
59	AP_name	VarChar (255)	报警循环函数名称
60	AP_PAR	VarChar (255)	报警循环画面
61	InfoText	VarChar (255)	信息文本
62	TxtCame	VarChar (255)	已进入的文本
63	TxtWent	VarChar (255)	已离开的文本
64	TxtCameNWent	VarChar (255)	已进入和已离开的文本
65	TxtAck	VarChar (255)	已确认的文本
66	AlarmTag	整型 (4 字节)	消息变量
67	AckType	小整型 (2 字节)	确认类型

4.5 查询归档数据

Position	字段名称	类型	注释
68	Params	整型 (4 字节)	参数
69	服务器名称	VarChar (255)	服务器名称

参见

查询归档数据 (页 43)

建立到归档数据库的连接 (页 39)

使用 OLE DB 提供程序访问归档数据 (页 31)

OLE DB 的基础 (页 33)

4.5.6 用户归档的查询

简介

通过下列查询, 可使用 MS OLE DB 提供程序访问 WinCC 用户归档。 可以进行读取或写访问, 以便分析已保存的数据, 以及对其进行修改和保存。

可使用过滤标准选择数据。 查询通过命令对象转发到数据库。

说明

通过 MS OLE DB 提供程序访问 WinCC 用户归档时, 要考虑以下方面:

- 确保未同时通过 MS OLE DB 提供程序和 WinCC 启用写访问。 这样可防止归档中出现不一致情况。
 - 在因画面更改而选择用户归档表格控件之前, 在 WinCC 运行系统中不会显示通过 MS OLE DB 提供程序进行的更改。 将再次读取当前用户归档数据。
 - 通过 MS OLE DB 提供程序更改的用户归档不在冗余系统中同步。
 - 请注意, WinCC 更新会导致数据库模式发生变化。 安装热修补程序和服务包也会使模式发生变化。 在这种情况下, 必须对读写访问进行相应的调整。
-

语法

读取值

```
SELECT * FROM UA#<ArchiveName>[WHERE <Condition>...., optional]
```

写入值

```
UPDATE UA#<ArchiveName> SET UA#<ArchiveName>.<Column_n> = <Value>
[WHERE <Condition>....., optional]
```

插入数据集

```
INSERT INTO UA#<ArchiveName>
(ID,<Column_1>,<Column_2>,<Column_n>) VALUES (<ID_Value>,
Value_1,Value_2,Value_n)
```

删除数据集

```
DELETE FROM UA#<ArchiveName> WHERE ID = <ID_Number>
```

参数

参数	描述
ArchiveName	用户归档的名称。
条件	过滤标准，例如： LastAccess>'2004-06-01' AND LastAccess<'2004-07-01' DateTime>'2004-06-01 17:30:00' ID = 5 ID > 3

实例 1:

读取用户归档“Test”中的所有数据。

```
SELECT * FROM UA#Test
```

示例 2:

读取在 2004 年 6 月 1 日和 2004 年 7 月 1 日之间发生变化的用户归档“Test”中的所有数据。

```
SELECT * FROM UA#Test WHERE LastAccess>'2004-06-01' AND
LastAccess<'2004-07-01'
```

实例 3:

在 ID 3 的 F_STRING 域中输入“New_String”值。

4.5 查询归档数据

```
UPDATE UA#TEST SET F_STRING = 'New_String' WHERE ID = 3
```

实例 4:

插入带有 ID 100 的数据集。

```
INSERT INTO UA#Test (ID,F_Integer,F_Float,F_Double,F_String)  
VALUES (100.10, '10.0', 'AAAA')
```

实例 5:

删除带有 ID 100 的数据集。

```
DELETE FROM UA#Test WHERE ID = 100
```

参见

显示用户归档 (页 60)

4.5.7 显示用户归档

简介

每个用户归档均由具有可编辑属性的数据域组成。

每个数据域都具有多个属性，如名称、别名、类型、长度、值等。

用户归档编辑器中数据域和属性以行和列显示。因此，我们叙述时用行代替数据域，用列代替属性。

下文举例说明了用户归档可能使用的结构。

示例：用户归档的布局

字段名称	类型	注释
ID	整型	值的唯一标识
F_Integer	整型	值示例
F_Float	单精度浮点数	值示例
F_Double	双精度型	值示例
F_String	字符串	示例字符串

参见

- 查询归档数据 (页 43)
- 建立到归档数据库的连接 (页 39)
- 使用 OLE DB 提供程序访问归档数据 (页 31)
- OLE DB 的基础 (页 33)

4.5.8 通过“SQL Server Import/Export”向导组态访问

简介

使用“SQL Server Import/Export”向导时，可通过 WinCC OLE DB Provider 来访问 WinCC 数据库。

可使用向导将解压的数据保存到中间数据库中；使用标准 SQL 查询来访问数据库。

- 使用 WinCC 计算机可以在本地或远程建立对运行系统和归档数据库的访问。
- 如果是长期归档服务器，则只可以在本地或远程访问归档数据库，因为长期归档服务器没有运行系统数据库。

操作步骤

1. 启动“SQL Server Management Studio”，然后选择所需的数据库。
2. 在数据库的快捷菜单中选择“Tasks > Export Data”。
打开 SQL Server 导入/导出向导。
3. 组态数据源：
 - 在“Data source”字段中，选择条目“WinCC OLE DB-Provider for Archives”。
 - 单击“Properties”按钮。
 - “Data Link Properties”对话框就会打开。

4.5 查询归档数据

4. 组态正确的提供程序设置：
 - 在字段“Data Source”中输入以下文本作为数据源：.\WinCC
“Location”的条目留空。
 - 在“Enter the initial catalog to use”下输入所需的运行系统数据库或符号名称（已在“WinCC Archive Connector”工具中组态）。
正确拼写的名称可在“SQL Server Management Studio”的“Databases”目录中找到。
或者可为运行系统数据和归档数据输入“CC_ExternalBrowsing”数据库。

说明

为了改善本地访问期间的性能，请在“Data Source”字段中输入“<计算机名称>\WinCC”，而不是“.\WinCC”。

5. 单击“Advanced”选项卡。
 - 为“Connect timeout”属性选择所需的时间（以秒为单位）。
 - 在“Access permissions”属性中，仅选择“ReadWrite”复选框。
单击“确定”(OK) 关闭对话框，并单击“Next”。
6. 组态数据目标：
 - 例如，在“Destination”字段中，选择条目“SQL Server Native Client”。
 - 服务器名可以是任何 SQL Server 实例。
 - 在“Database”字段中输入的数据库名称可以是任何自创建的目标数据库。

说明

也可将“Database”字段留空。这样便不会创建目标表。

7. 组态查询条件：
 - 选择选项“Write a query to specify the data to transfer”，然后单击“Next”。
 - 输入所需的查询条件或使用“Browse”加载查询文件 *.sql。
例如，使用 ValueID“1”的查询“Tag:R,1,'0000-00-00 00:10:00.000','0000-00-00 00:00:00.000’”，将读取归档最后 10 分钟的值。
必要时，通过“Parse”检查查询。
有关语法的更多信息，请参见“查询过程值归档 (页 45)”部分。
8. 如果需要，请通过“Edit Mappings”编辑查询。
9. 必要时，请选择存储位置。
10. 要导出数据，请使用“Finish”关闭向导。

结果

向导即会将数据导出到目标数据库中。

如果数据导出成功，目标数据库中未压缩的数据将被保存到新创建的表“dbo.Query”中。

您可以更改表名。

即使不更改该表名，数据也不会被新的导出所覆盖。向导将创建名为“Query1”、“Query2”等等的新表。

参见

使用范例 7：通过 WinCC OLE DB Provider 进行访问 (页 23)

查询过程值归档 (页 45)

4.5.9 满足使用报表服务的先决条件

简介

可将 SQL Server 的报表服务与 WinCC 结合使用。

这可在网络上使用归档数据提供由 Microsoft Visual Studio 创建的报表。

要求

通过 WinCC 使用报表服务需要事先安装下列软件。

- Internet 信息服务
- Microsoft SQL Server 2016 的报表服务

说明

按照指示顺序正确地执行安装步骤。

安装 MS SQL Server 的报表服务

从 WinCC DVD 或从文件系统中的 WinCC 安装程序路径安装报表服务。

1. 在 Windows 控制面板中，打开“卸载或更改程序”(Uninstall or change a program) 对话框。
2. 在“Microsoft SQL Server 2016”的快捷菜单中，选择条目“卸载/更改”(Uninstall/Change)。
3. 在 WinCC 安装程序中，选择“添加”(Add) 和路径“InstData > SQL > SQL2016STDSP2 > setup”。Microsoft SQL Server 安装向导将会打开。按照屏幕说明操作。
4. 在“安装类型”(Installation type) 下，选择“添加功能到现有 SQL Server 2016 实例”(Add features to an existing SQL Server 2016 instance) 选项 和“WINCC”实例。
已安装管理工具和 SQL Server 数据工具。

4.5 查询归档数据

5. 激活“报表服务 - 本机”(Reporting Services - Native) 并遵照说明进行操作。
已安装报表服务。
6. 在 Windows 程序组“Microsoft SQL Server 2016”中，启动“报表服务的组态管理器”(Configuration Manager for Reporting Services) 并组态报表服务。

组态 Internet 信息服务

1. 打开“计算机管理”(Computer Management)。
2. 在“服务和应用程序”(Services and Applications) 下，选择“Internet 信息服务 (IIS) 管理器”(Internet Information Services (IIS) Manager)。
3. 打开功能“身份验证”(Authentication) 并在“匿名身份验证”(Anonymous authentication) 的快捷菜单中选择“编辑”(Edit)。
4. 选择“特定用户”(Specific user) 选项并输入用户名和密码。
用户名的格式如下：<Domain or computer name>\<User>

说明

建议在安装完成后重新启动计算机。

结果

使用报表服务的先决条件已经满足。

现在，可在 internet 上创建和提供报表。

4.6 消息和过程值的分析函数

4.6.1 消息和过程值的分析函数

简介

WinCC 提供不同的分析函数用于查询已归档的消息和过程值。

对于不同的集合函数由带参数的查询触发分析。集合函数的计算在“连通性软件包”服务器上执行，并且只将结果传送到客户机。

有关在下列示例中使用的 `CommandText` 和 `ConnectionString` 的附加信息可以在“建立到归档数据库的连接”和“查询归档数据”中找到。

消息的分析函数

报警日志的分析查询返回一个特定的记录集，该记录集包含每个消息的组态和运行系统数据以及聚合函数的结果。

报警日志的分析查询的已返回记录集与消息归档的标准查询的记录集不完全相同。有关更多信息，请参见“显示分析查询的报警日志”部分。

针对每个消息，会计算下列集合函数。结果列表的列说明位于括号内。

- 消息频率之和 (“`FreqOfAlarm`”)
- 从“消息进入”到“消息离开”的累积持续时间 (“`CumDurationComeGo`”)
- 从“消息进入”到“消息离开”的平均持续时间 (“`AvDurationComeGo`”)
- 从“消息进入”到首次确认的累积持续时间 (“`CumDurationComeAckn1`”)
- 从“消息进入”到首次确认的平均持续时间 (“`AvDurationComeAckn1`”)
- 从“消息进入”到第二次确认的累积持续时间 (“`CumDurationComeAckn2`”)
- 从“消息进入”到第二次确认的平均持续时间 (“`AvDurationComeAckn2`”)
- 从“消息进入”到“消息离开”的累积持续时间 (“`CumDurationComeGo`”)
- 从“消息进入”到“消息进入”的平均持续时间 (“`AvDurationComeCome`”)

说明

不支持“字符串”数据类型

如果将这些函数用于“字符串”数据类型，则分析函数将返回结果“0”。

4.6 消息和过程值的分析函数

语法

对于消息的集合函数的计算，向 WinCC OLE DB 提供程序发出下列命令。

```
"AlarmHitView:SELECT * FROM <ViewName>[WHERE <Condition>]"
```

其中：

<ViewName> = 以所需语言表示的数据库表的名称，例如，英语名称为 ALGVIEWMENU。

[WHERE <Condition>] = SQL 语法中作为 WHERE 条件的可选过滤标准。

有关参数语法的更多信息，请参见“报警日志的查询”部分。

示例

该示例提供了“ALGVIEWENU”数据库在 7/15/2004 12:00 p.m. 到 12:15 p.m. 这个时间范围内所有消息的聚合函数的结果。

ConnectionString:

```
"Provider=WinCCOLEDBProvider.1;Catalog=CC_OpenArch_03_05_27_14_11_46R;Data Source=.\WinCC"
```

CommandText:

```
"AlarmHitView:SELECT * FROM ALGVIEWENU WHERE DateTime>'2004-07-15 12:00:00' AND DateTime<'2004-07-15 12:15:00'"
```

过程值的分析函数

过程值的分析会返回集合函数的结果。在一个查询中只可以计算一个集合函数。

下列集合函数可用于过程值。

- MIN（最小值）
- MAX（最大值）
- AVG（平均值）
- SUM（所有值的和）
- COUNT（过程值的计数）
- COUNTER（带有值“1”的条目数，例如，二进制变量的查询）
- STDEV（统计标准偏差）
- VAR（统计方差）

说明**不支持文本变量**

在通过 TAG_EX:R 查询文本变量（8 位字符集/16 位字符集）时，分析函数返回结果“0”。

语法

对于过程值的集合函数的计算，向 MS SQL OLE DB 提供程序发出查询指令，并执行来自“SQL Server Master”数据库的“cp_TagStatistic”步骤。

说明

透明访问中过程值的分析函数仅与自身具有项目的客户机上的“连通站”配合使用。

下列参数将传送到“cp_TagStatistic”步骤。

```
cp_TagStatistic @P1,@P2,@P3[,@P4]
```

其中：

“@P1”= 数据库名称（例如，WinCC Runtime 数据库或具有换出归档的目录的符号名称）。对于透明访问，使用 WinCC 项目名称（而非数据库名称）。

“@P2”= 过程值的 WinCC OLE DB 提供程序字符串。

“@P3”= 所需集合函数。

“@P4”= <符号计算机名称>:\WinCC（仅透明访问需要）。

有关参数“@P1”和“@P2”语法的更多信息，请参见“过程值归档的查询”部分。

说明**针对带有亚洲归档变量名称的过程值的分析函数**

如果使用带有亚洲字符集的归档变量名称计算分析函数，则必须修改对 Unicode 字符集的请求。

在两个参数前都添加前缀“N”。

```
示例: cp_TagStatistic N'TestDB',N'TAG:R,17,"2004-05-17 12:00:00","2004-05-17  
13:00:00",'AVG'
```

示例

此查询为数据库“TestDB”的 ValueID“17”返回 5/17/2004 12:00 到 13:00 这个时间范围内过程值的平均值。

ConnectionString:

```
"Provider=SQLNCLI11;Integrated Security=SSPI;Persist Security  
Info=False;Initial Catalog=master ;Data Source=.\WinCC"
```

4.6 消息和过程值的分析函数

CommandText:

```
"cp_TagStatistic 'TestDB','TAG:R,17','2004-05-17  
12:00:00','','2004-05-17 13:00:00','','AVG'"
```

透明访问的实例

此查询为“WinCCProj”的 ValueID“7”提供 14.09.2006 10:00 到 11:00 这个时间范围内的平均过程值。

ConnectionString:

```
"Provider=SQLNCLI11;Integrated Security=SSPI;Persist Security  
Info=False;Initial Catalog=master ;Data Source=.\WinCC"
```

CommandText:

```
"cp_TagStatistic 'WinCCProj','TAG:R,7','2006-09-14  
10:00:00','','2006-09-14  
11:00:00','','AVG','Symb_WinCCProj::\WinCC'"
```

参见

- 查询归档数据 (页 43)
- 建立到归档数据库的连接 (页 39)
- 显示分析查询的消息归档 (页 68)
- 使用连通站的 OLE DB 接口 (页 241)

4.6.2 显示分析查询的消息归档

简介

消息归档的分析查询返回一个特定的记录集，该记录集包含每个消息的组态和运行系统数据以及集合函数的结果。

该记录集与消息归档的标准查询的记录集并不完全相同。

消息归档分析的记录集结构

使用分析函数“AlarmHitView”查询消息归档时，返回的结果是带有下列结构的记录集。

位置	域名称	类型	注释
1	MsgNo	整型（4 字节）	消息号
2	State	小整型（2 字节）	报警记录状态
3	DateTime	日期时间（8 字节）	消息的时间戳（不带毫秒的日期/时间）
4	Ms	小整型（2 字节）	消息的时间戳（毫秒）
5	Instance	VarChar (255)	报警记录的实例名称
6	Flags1	整型（4 字节）	（仅限于内部使用）
7	Counter	整型（4 字节）	运行报警消息计数器
8	TimeDiff	整型（4 字节）	与“进入”状态的时间差
9	ClassName	VarChar (255)	消息类别的名称。
10	Typename	VarChar (255)	消息类型的名称。
11	类别	小整型（2 字节）	消息类别 ID
12	Type	小整型（2 字节）	消息类型 ID
13 到 22	Text1 到 Text10	VarChar (255)	消息文本 1 到 10
23	AG_NR	小整型（2 字节）	PLC 的编号
24	CPU_NR	小整型（2 字节）	CPU 的编号
25	CrComeFore	整型（4 字节）	“进入”状态的前景色
26	CrComeBack	整型（4 字节）	“进入”状态的背景色
27	CrGoFore	整型（4 字节）	“离开”状态的前景色
28	CrGoBack	整型（4 字节）	“离开”状态的背景色
29	CrAckFore	整型（4 字节）	“已确认”状态的前景色
30	CrAckBack	整型（4 字节）	“已确认”状态的背景色
31	Priority	整型（4 字节）	优先级
32	AP_type	整型（4 字节）	报警循环

4.6 消息和过程值的分析函数

位置	域名称	类型	注释
33	AP_name	VarChar (255)	报警循环函数名称
34	AP_PAR	VarChar (255)	报警循环画面
35	InfoText	VarChar (255)	信息文本
36	TxtCame	VarChar (255)	已进入的文本
37	TxtWent	VarChar (255)	已离开的文本
38	TxtCameNWent	VarChar (255)	已进入和已离开的文本
39	TxtAck	VarChar (255)	已确认的文本
40	AckType	小整型 (2 字节)	确认类型
41	FreqOfAlarm	整型 (4 字节)	消息频率之和
42	CumDurationComeGo	整型 (4 字节)	从“消息进入”到“消息离开”的累积持续时间
43	AvDurationComeGo	实型 (8 字节)	从“消息进入”到“消息离开”的平均持续时间
44	CumDurationComeAckn1	整型 (4 字节)	从“消息进入”到首次确认的累积持续时间
45	AvDurationComeAckn1	实型 (8 字节)	从“消息进入”到首次确认的平均持续时间
46	CumDurationComeAckn2	整型 (4 字节)	从“消息进入”到第二次确认的累积持续时间
47	AvDurationComeAckn2	实型 (8 字节)	从“消息进入”到第二次确认的平均持续时间
48	CumDurationComeCome	整型 (4 字节)	从“消息进入”到“消息进入”的累积持续时间
49	AvDurationComeCome	实型 (8 字节)	从“消息进入”到“消息进入”的平均持续时间

参见

消息和过程值的分析函数 (页 65)

使用 MS OLE DB 访问 SQL 数据库期间的安全设置

简介

使用 MS OLE DB，用户可以访问 SQL 数据库（如 WinCC 用户归档），并且可以对其进行修改。因此，未经授权的用户也可以这样做。

要避免未经授权的访问，管理员必须采取适当的措施。

原理

可采取的一种访问保护方法是，在 SQL 服务器上创建用户或用户组以用于访问 SQL 数据库。然后为该用户或用户组指派访问 SQL 数据库的一定权限。

这可以在 SQL 服务器上使用本地或全局 Windows 用户组，以在 SQL 中组织此类访问权限。或者，可将 Windows 用户传送到 SQL 服务器 - 直接作为 Windows 用户或作为单独定义的 SQL 用户。

服务器角色

SQL 服务器安全性结构的一部分称为“服务器角色”。

它可以将用户按管理目的分组，这与 Windows 中的情况类似。

当没有相应的 Windows 用户组时，“服务器角色”用于将服务器范围的安全性设置指派给登录。

有关 SQL Server 2016 的技术文档，请访问 Microsoft 网站，Internet:

- SQL Server 2016 文档 ([https://msdn.microsoft.com/zh-cn/library/hh995091\(v=sql.10\).aspx](https://msdn.microsoft.com/zh-cn/library/hh995091(v=sql.10).aspx))
- SQL Server 2016 概述 (<https://docs.microsoft.com/zh-cn/sql/sql-server/sql-server-technical-documentation?view=sql-server-2016>)

参见

OLE DB 的基础 (页 33)

SQL Server 2016 文档 ([https://msdn.microsoft.com/zh-cn/library/hh995091\(v=sql.10\).aspx](https://msdn.microsoft.com/zh-cn/library/hh995091(v=sql.10).aspx))

SQL Server 2016 概述 (<https://docs.microsoft.com/zh-cn/sql/sql-server/sql-server-technical-documentation?view=sql-server-2016>)

通过 OPC 访问 - 开放式互连

6.1 OPC 通道

6.1.1 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

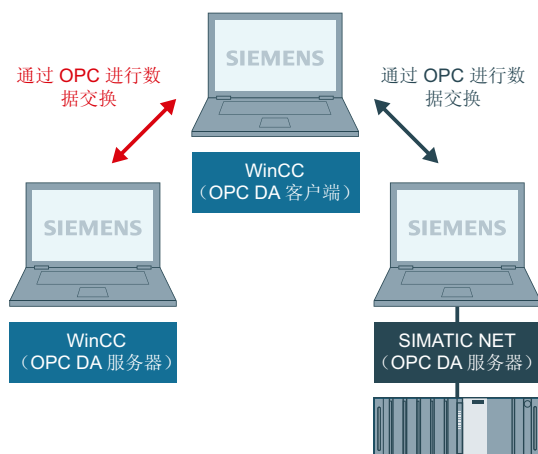
可能的应用

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。

6.1 OPC 通道



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。

6.1.2 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。
- 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

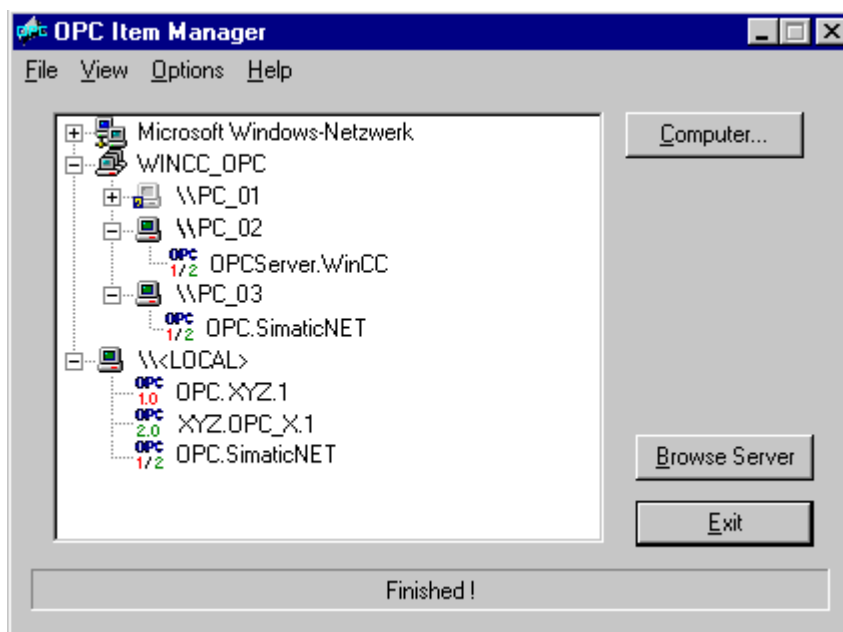
OPC DA 服务器

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”。

6.1 OPC 通道



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。

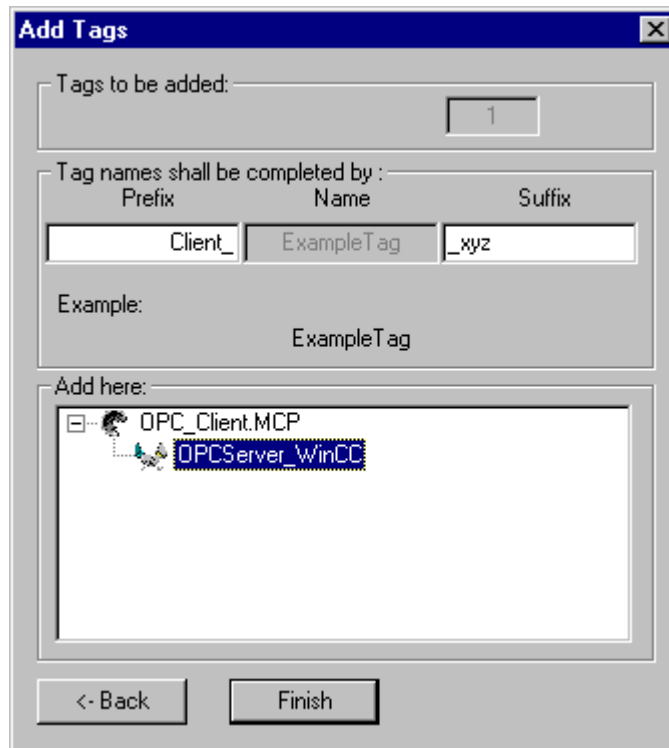
创建连接

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。

6.1 OPC 通道

参见

如何通过 OPC 条目管理器访问 WinCC 变量 (页 97)

如何通过 OPC 条目管理器访问 WinCC 变量 (页 80)

6.1.3 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。

参见

如何在 WinCC OPC DA 客户机上使用结构 (页 88)

6.1.4 WinCC OPC DA 客户机

6.1.4.1 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。

6.1 OPC 通道

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 秒周期性地地进行状态请求。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 88)

在不使用 OPC 条目管理器的情况下访问 WinCC 变量 (页 86)

通过 OPC 条目管理器组态访问 (页 82)

所支持的 WinCC 数据类型概述 (页 78)

OPC 条目管理器 (页 74)

OPC 规范及兼容性 (页 105)

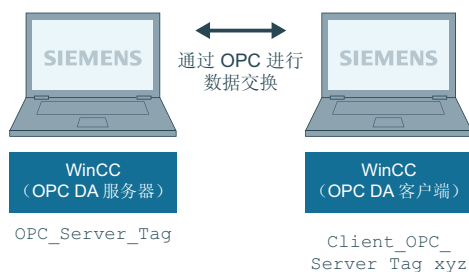
WinCC OPC DA 服务端的功能 (页 113)

6.1.4.2 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。



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 82)

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 81)

通过 OPC 条目管理器组态访问 (页 98)

WinCC 至 WinCC 的连接实例 (页 116)

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 82)

6.1 OPC 通道

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。

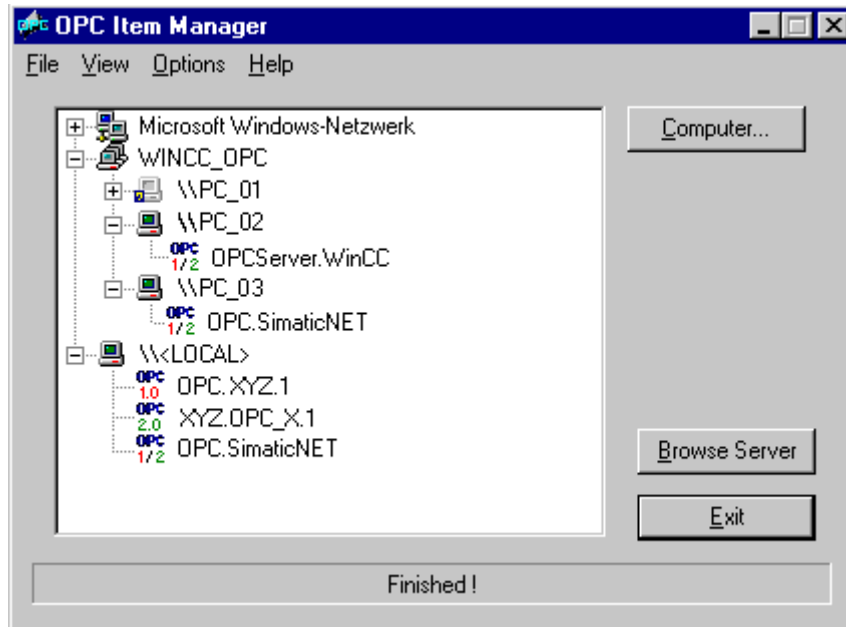
说明

“OPC”通道

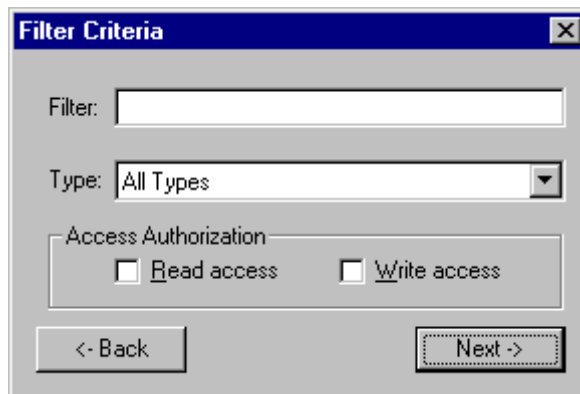
连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

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)。

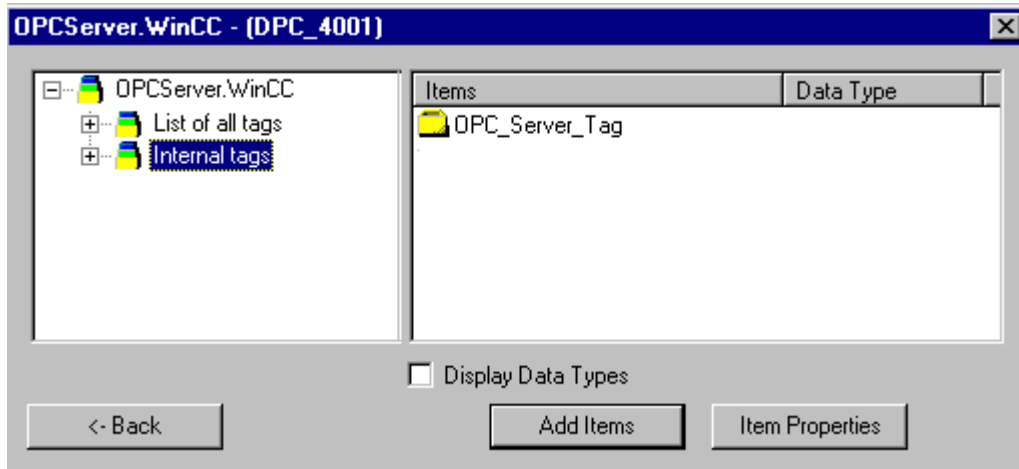


2. 从选择对话框中选择用作 WinCC OPC DA 服务器的计算机的名称。
从显示的列表中选择“OPCServer.WinCC”。
3. 单击“浏览服务器”按钮。
将打开“过滤标准”(Filter criteria) 对话框。

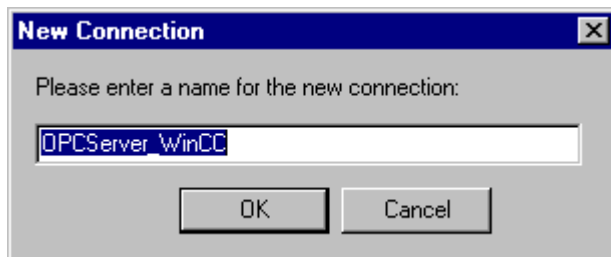


6.1 OPC 通道

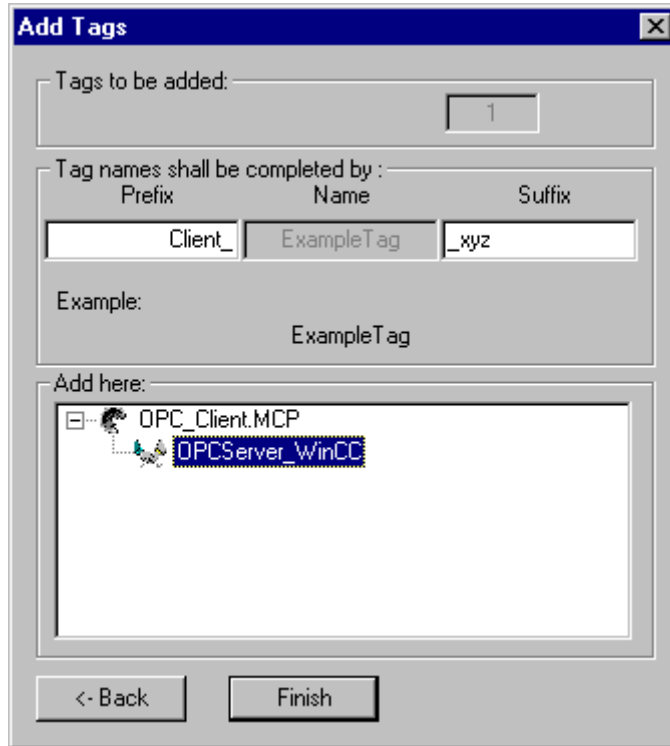
- 4. 单击 “下一步”(Next)。
将打开“OPCServer.WinCC ...”对话框。



- 5. 选择 WinCC 变量“OPC_Server_Tag”。
单击 “添加条目” 按钮。
- 6. 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。



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

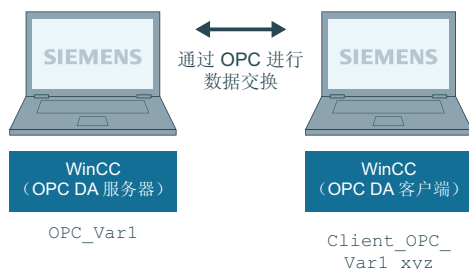
在 WinCC OPC DA 客户机上组态 OPC 通道 (页 81)

6.1 OPC 通道

6.1.4.3 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。

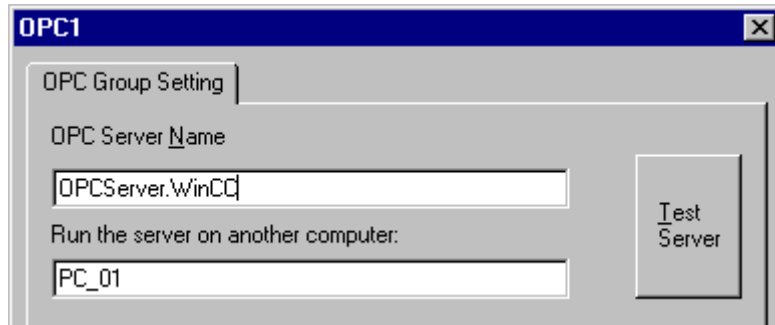
说明

“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。

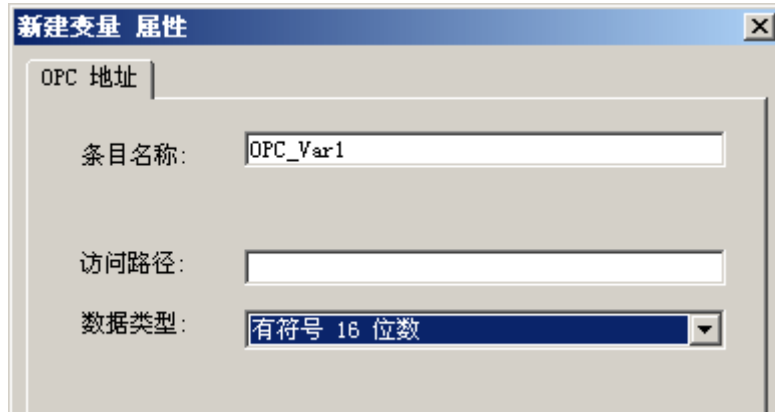
步骤

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。

6.1 OPC 通道

6.1.4.4 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 89)

在 WinCC OPC DA 服务器上组态结构和结构变量 (页 88)

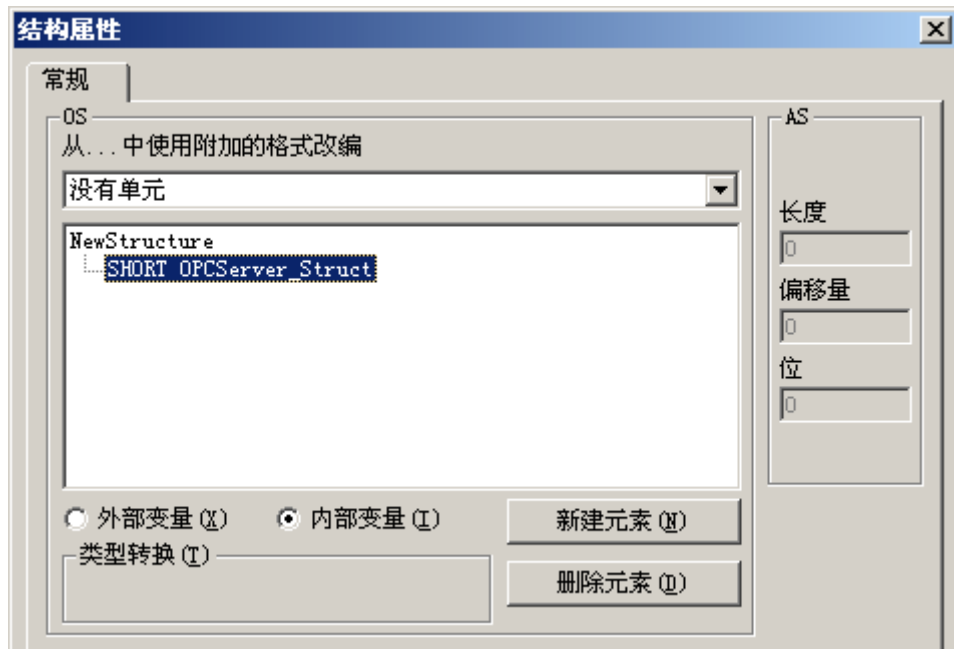
在 WinCC OPC DA 服务器上组态结构和结构变量

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 89)

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。

6.1 OPC 通道

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 81)

在 WinCC OPC DA 服务器上组态结构和结构变量 (页 88)

6.1.4.5 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 95)

WinCC 用作 OPC DA 客户机，建立连接。(页 94)

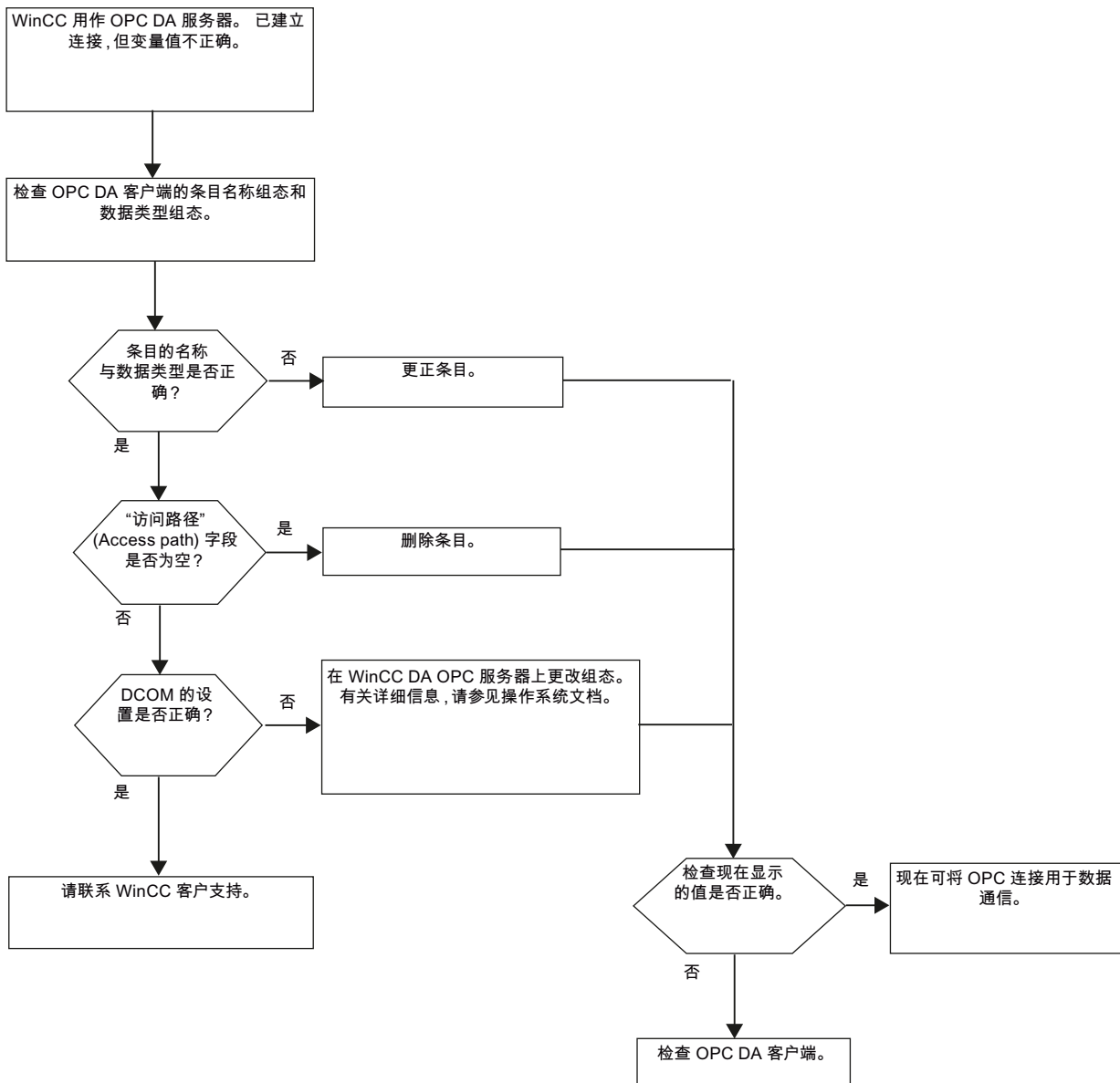
WinCC 用作 OPC DA 服务器，未建立连接。(页 93)

WinCC 用作 OPC DA 服务器，成功建立连接。(页 92)

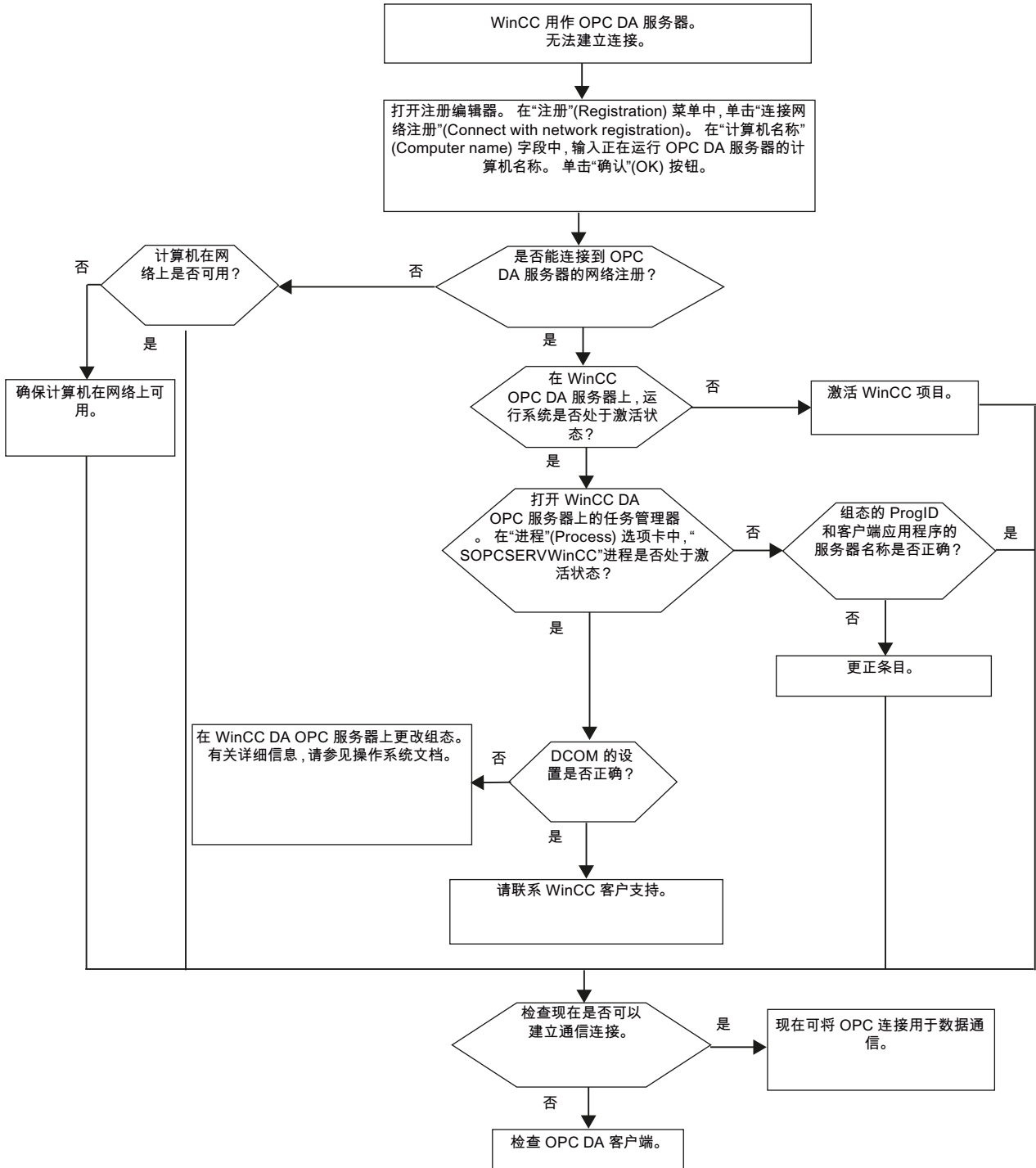
6.1 OPC 通道

WinCC 作为 OPC DA 服务器

WinCC 用作 OPC DA 服务器，成功建立连接。



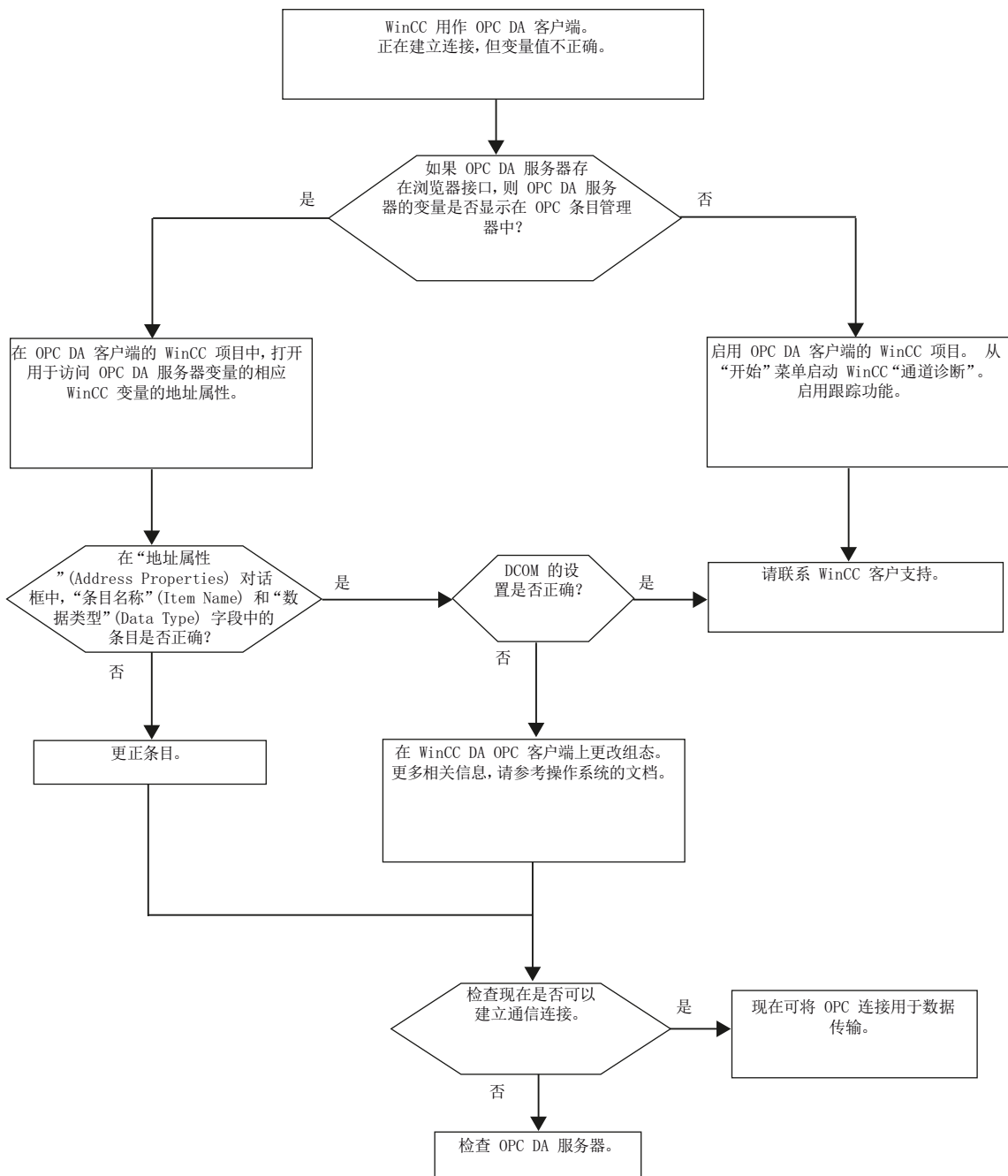
WinCC 用作 OPC DA 服务器，未建立连接。



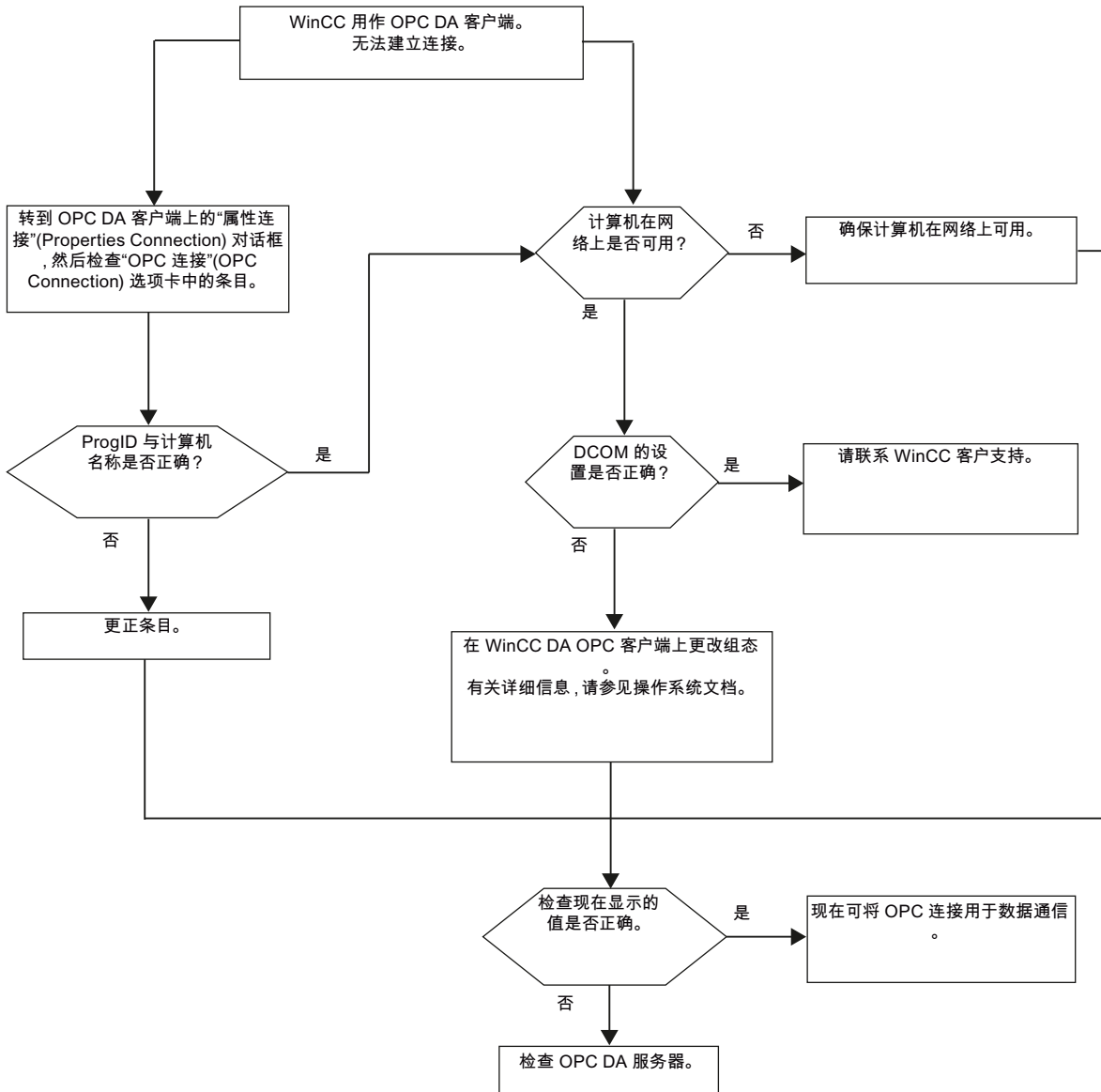
6.1 OPC 通道

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 客户机，未建立连接。



6.1.5 WinCC OPC XML 客户机

6.1.5.1 WinCC OPC XML 客户机的功能

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。

6.1 OPC 通道

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 78)

OPC 条目管理器 (页 74)

OPC 规范及兼容性 (页 105)

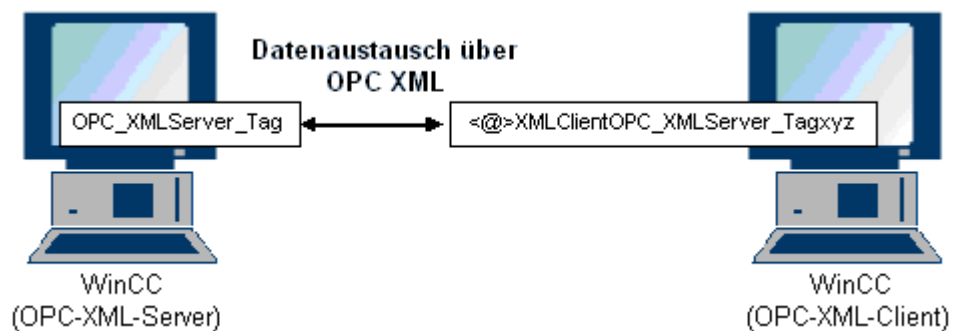
WinCC OPC XML DA 服务器的功能 (页 108)

6.1.5.2 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 81)

通过 OPC 条目管理器组态访问 (页 98)

6.1 OPC 通道

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。

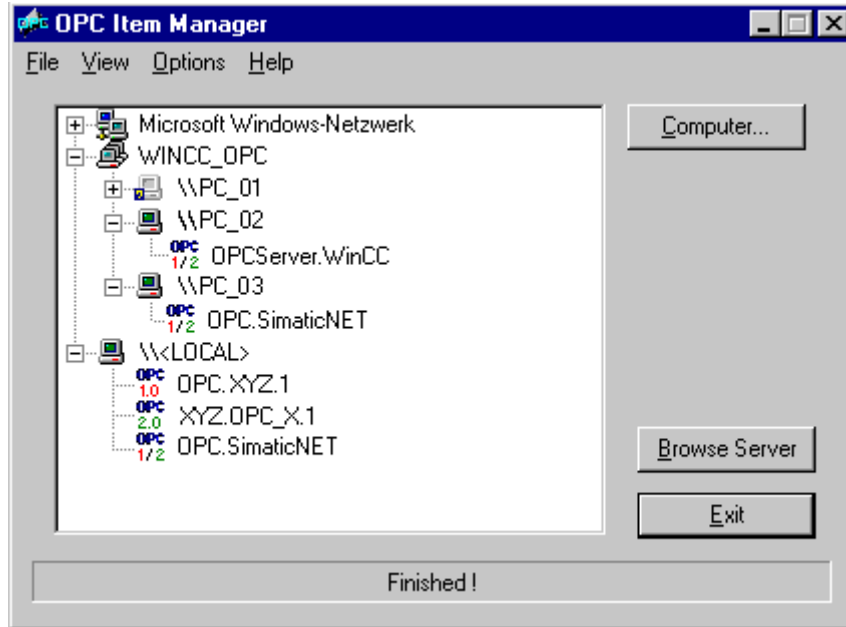
说明

“OPC”通道

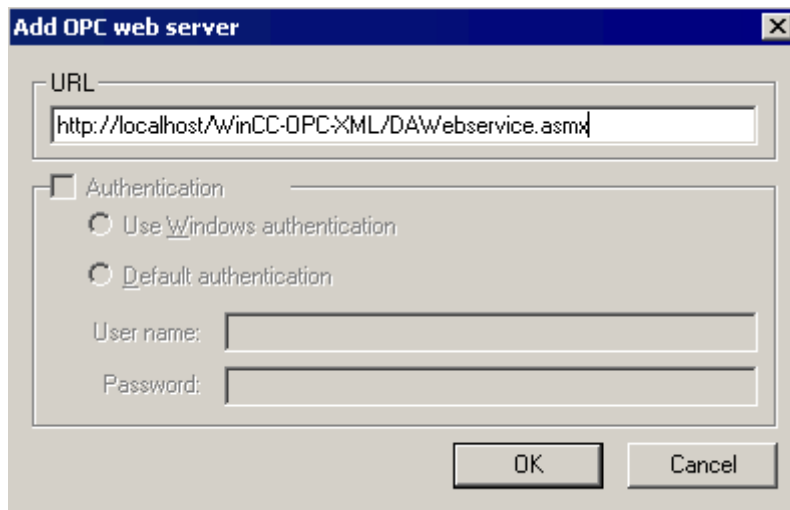
连接名称不支持 Unicode。请确保使用同一种语言对项目中的所有连接进行命名。打开计算机的控制面板为不支持 Unicode 的程序中所使用的语言设置代码页。

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)。



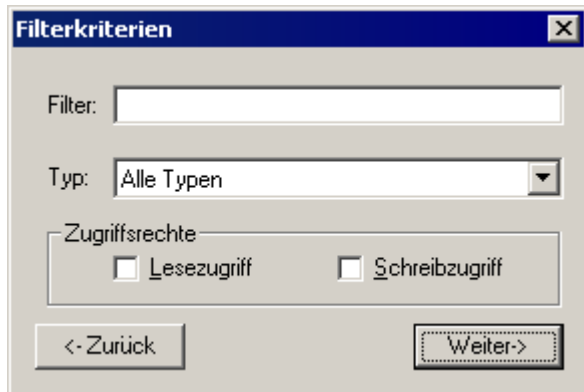
2. 单击“OPC Web 服务器”按钮。将打开“添加 OPC Web 服务器”对话框。以下列形式在“URL”域中输入 WinCC OPC XML 服务器的 URL: `<http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx>`。使用运行 OPC XML Web 服务的计算机的 IP 地址或计算机名称替代 xxx。



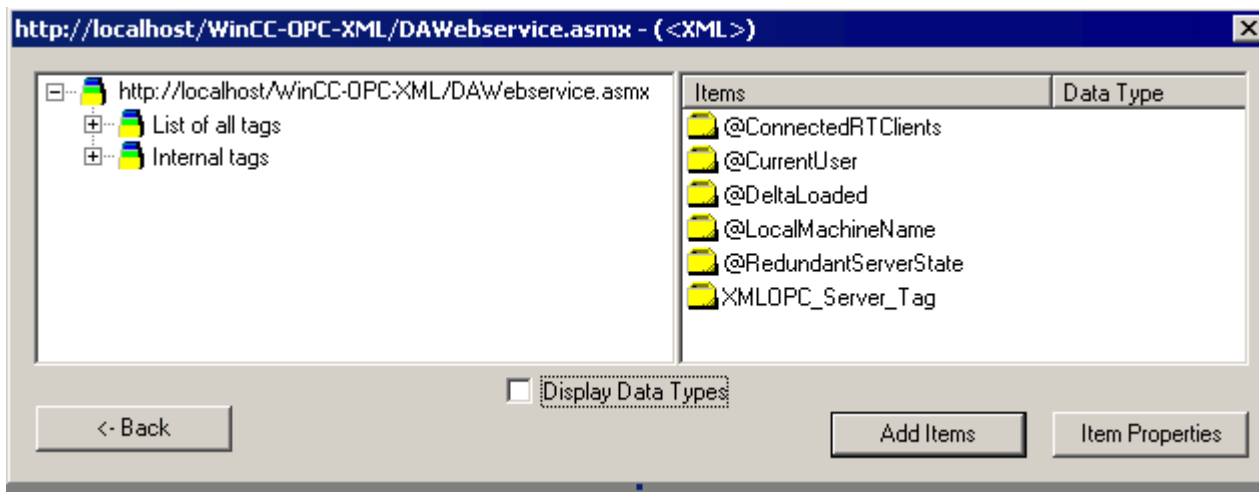
单击“确定”关闭对话框。

6.1 OPC 通道

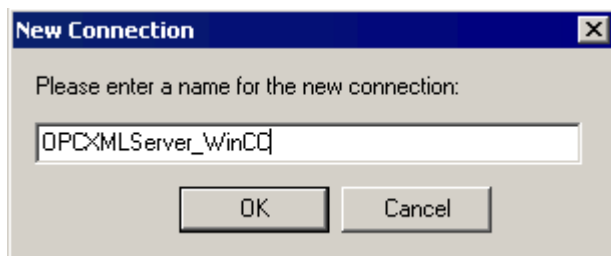
3. 显示一个列表，从其中选择“/WinCC-OPC-XML/DAWebservice.asmx>”。单击“浏览服务器”按钮。将打开“过滤标准(Filter criteria)对话框。



4. 在“过滤标准”对话框中，单击“下一步->”按钮。将打开“http:// ...”对话框。

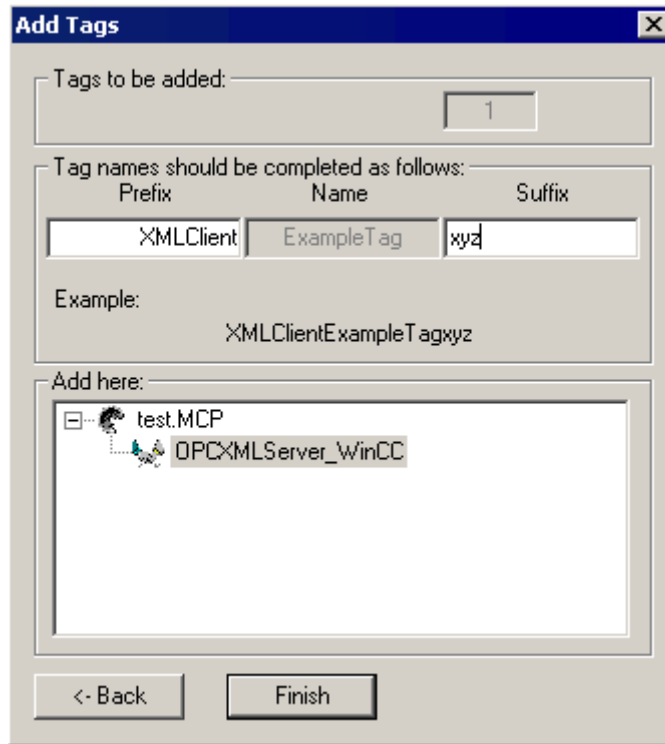


5. 在“http:// ...”对话框中，选择 WinCC 变量“XML OPC_Server_Tag”。单击“添加条目”按钮。
6. 如果已创建与 WinCC OPC XML 服务器的连接，继续步骤 7。如果尚未组态连接，则会输出相应的消息。
单击“是”(Yes)。将打开“新建连接(New Connection)对话框。



为连接输入名称“OPCXMLServer_WinCC”。单击“确定”(OK)。

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 81)

6.1.5.3 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。

6.1 OPC 通道

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

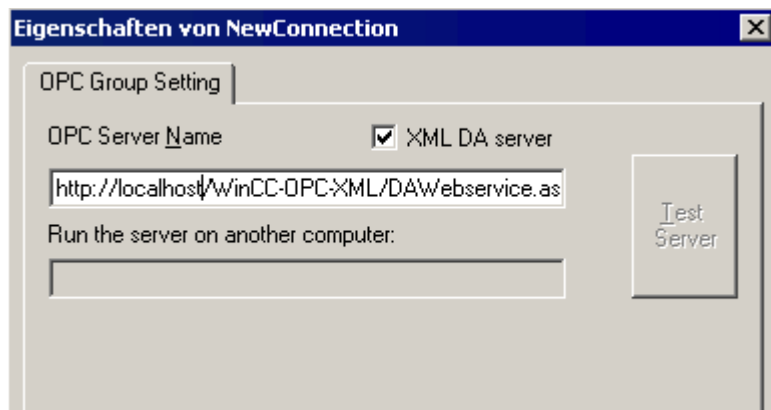
- 两台带 WinCC 项目的计算机。
- 必须可以使用 HTTP 访问这两台计算机。
- 在 WinCC OPC XML 服务器的 WinCC 项目中，组态一个名称为“XMLOPC_Server_Tag”、数据类型为“有符号 16 位数”的内部变量。
- 启用 WinCC OPC XML 服务器的 WinCC 项目。
- 必须将 OPC 通道添加到 WinCC OPC XML 客户机的 WinCC 项目。

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。

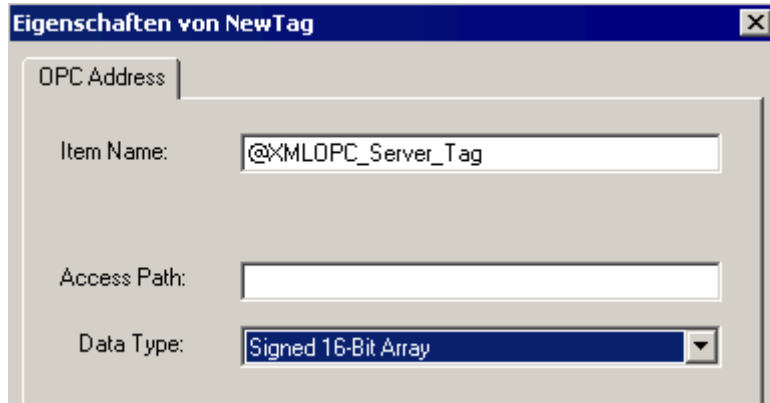
步骤

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。

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。

6.2 OPC - 开放式互连

6.2.1 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。

6.2.2 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。

6.2.3 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”。

6.2.4 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

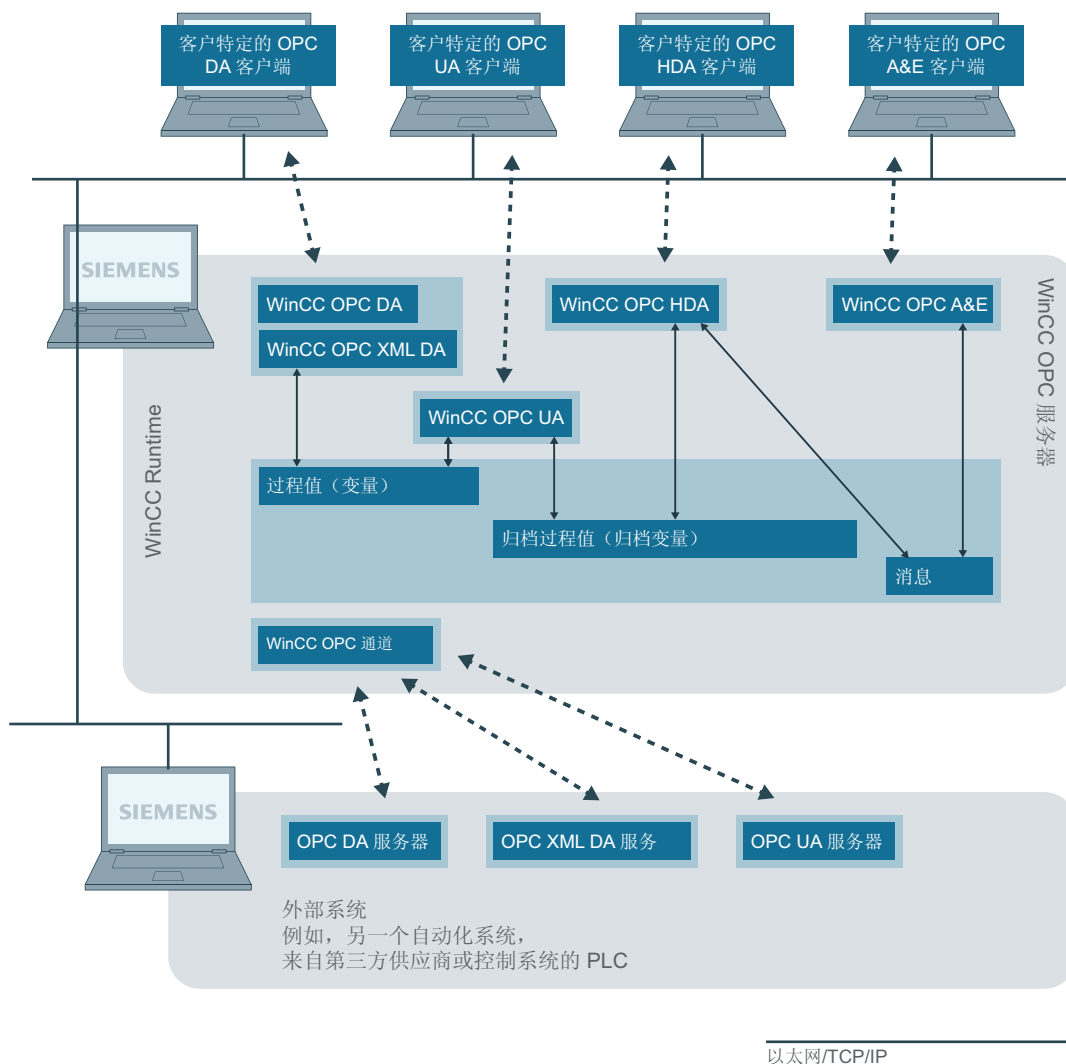
通过 DCOM 完成 WinCC OPC 服务器和 OPC 客户端之间的数据交换。安装 WinCC 后，将正确组态 WinCC OPC 服务器的 DCOM 设置。

6.2 OPC - 开放式互连

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	

6.2.5 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。

6.2 OPC - 开放式互连

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。

6.2.6 WinCC OPC XML DA 服务器

6.2.6.1 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題。

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。

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 95)

所支持的 WinCC 数据类型概述 (页 78)

www.opcfoundation.org (<http://www.opcfoundation.org>)

6.2.6.2 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。

6.2 OPC - 开放式互连

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) 时必须同时安装 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

6.2.6.3 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。

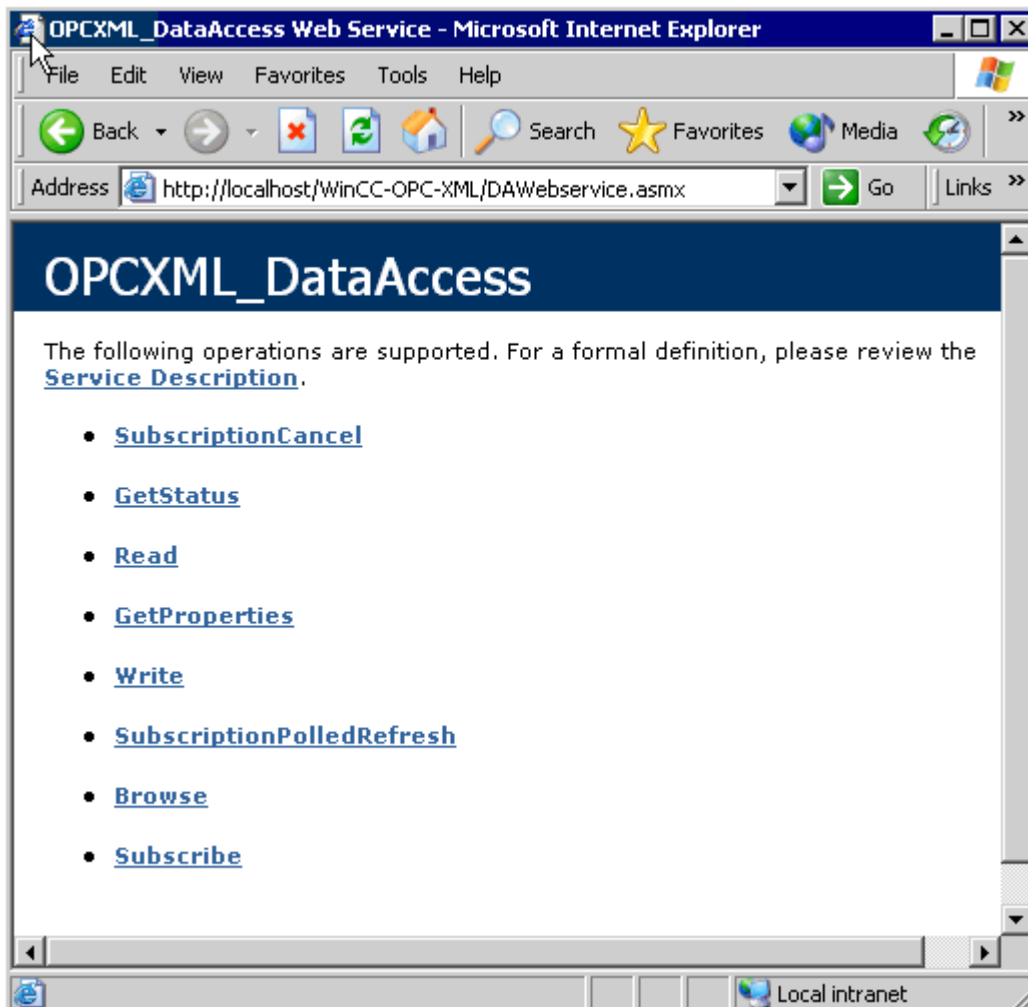
6.2.6.4 测试安装（WinCC OPC XML DA 服务器）

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。



6.2.7 WinCC OPC DA 服务器

6.2.7.1 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 115)

使用多个 OPC DA 服务器 (页 114)

6.2 OPC - 开放式互连

WinCC 至 WinCC 的连接实例 (页 116)

WinCC - SIMATIC NET FMS OPC 服务器连接的实例 (页 120)

WinCC - SIMATIC NET S7 OPC 服务器连接的实例 (页 122)

WinCC - Microsoft Excel 连接的实例 (页 127)

所支持的 WinCC 数据类型概述 (页 78)

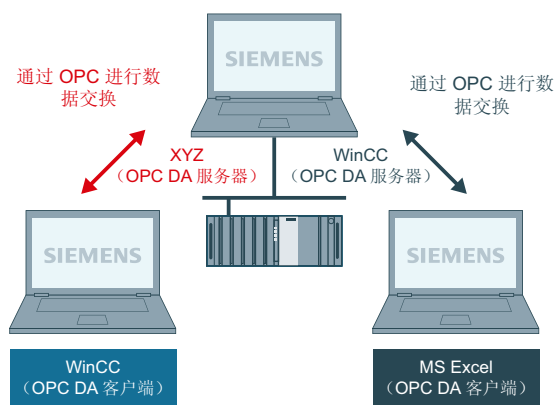
www.opcfoundation.org (<http://www.opcfoundation.org>)

6.2.7.2 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 115)

6.2.7.3 查询 OPC DA 服务器名称

简介

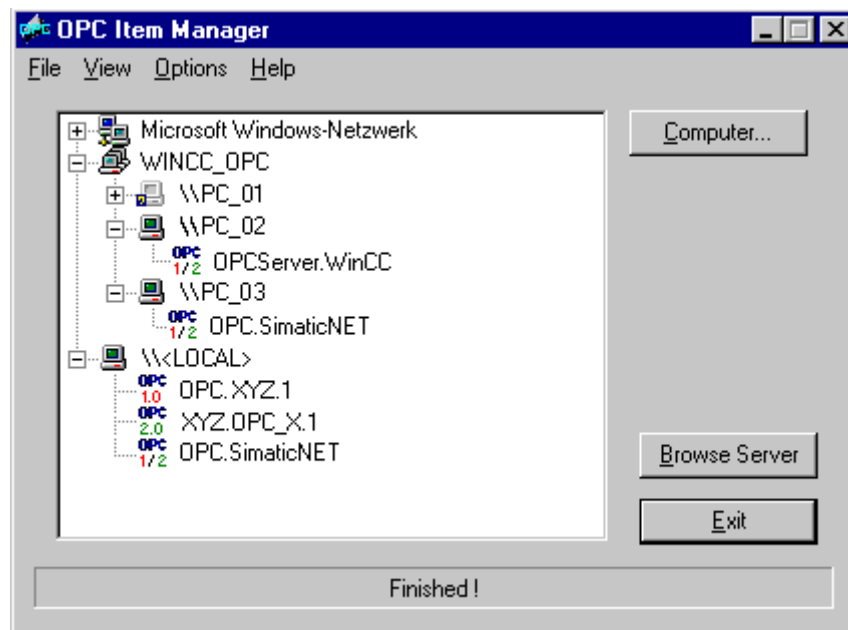
多个 OPC DA 服务器可安装在一台计算机上。OPC 条目管理器在选择窗口中显示可用于工作站的 OPC DA 服务器的名称。这些 OPC DA 服务器能够在同一计算机上运行或在已联网的计算机上运行。

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 74)

6.2 OPC - 开放式互连

6.2.7.4 OPC DA 连接的实例

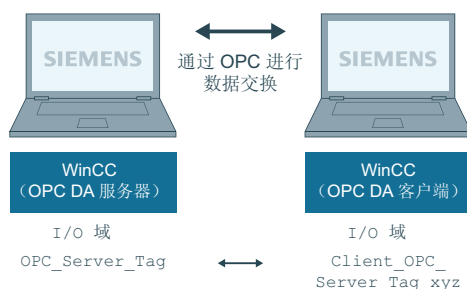
WinCC - WinCC 连接

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 117)

组态 WinCC OPC DA 客户机上的 WinCC 项目 (页 118)

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 118)

组态 WinCC OPC DA 客户机上的 WinCC 项目

引言

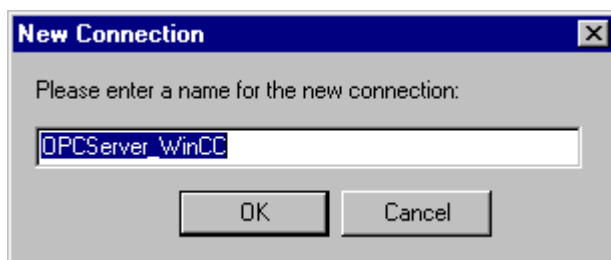
在本节中，将在 WinCC OPC DA 客户机上创建 WinCC 变量，以读取 WinCC OPC DA 服务器上的 WinCC 变量。变量值显示在 I/O 域中。

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。

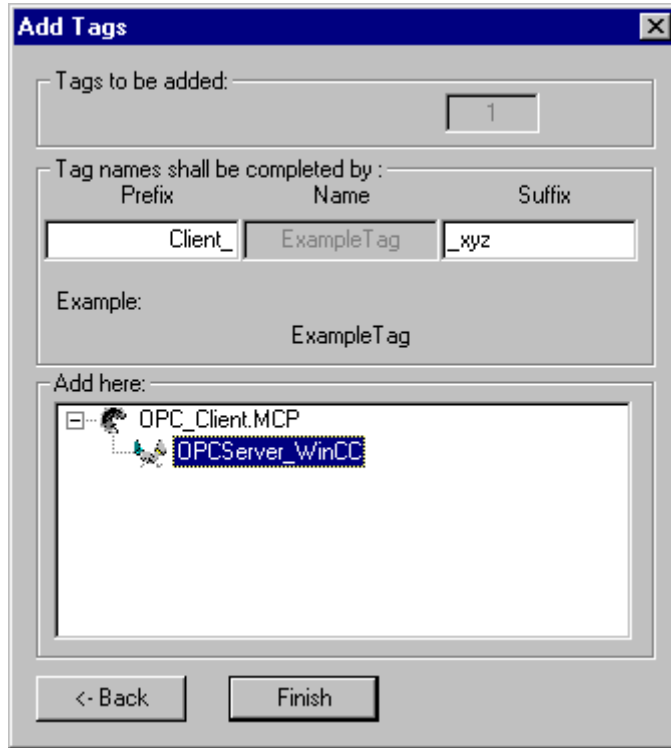
步骤

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 117)

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 81)

WinCC - SIMATIC NET FMS OPC 服务器的连接

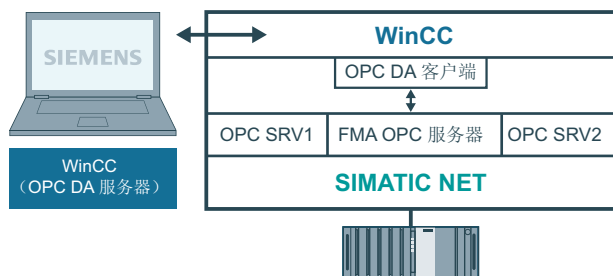
WinCC - SIMATIC NET FMS OPC 服务器连接的实例

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_SimaticNET”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。

6.2 OPC - 开放式互连

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

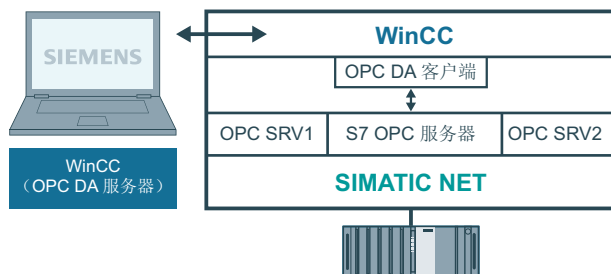
在 WinCC OPC DA 客户机上组态 OPC 通道 (页 81)

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

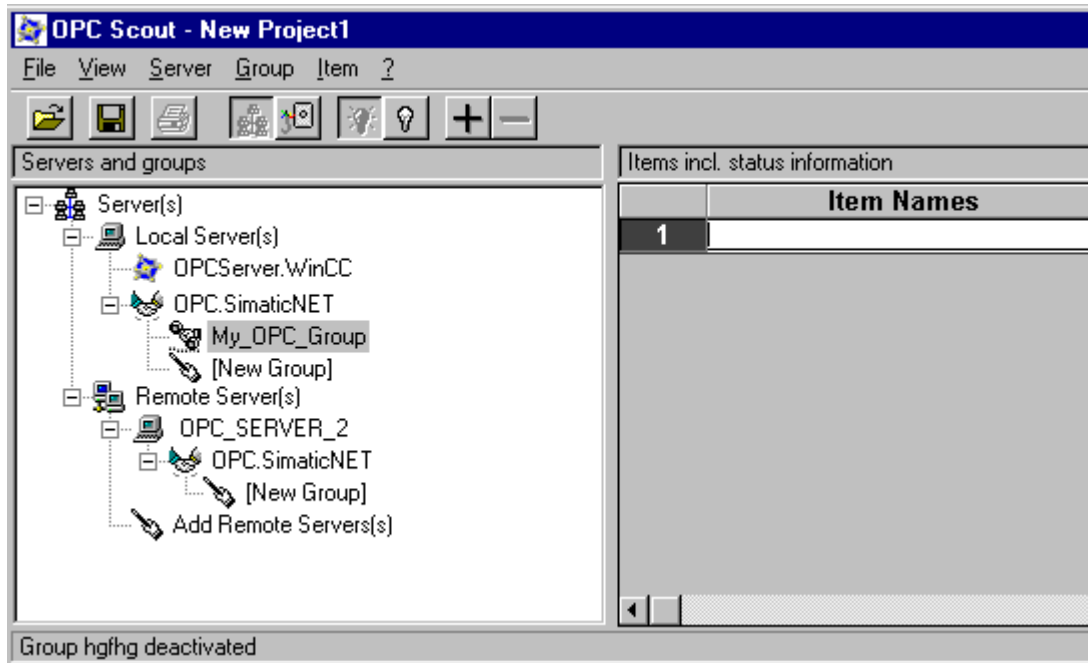
要求

- 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。

6.2 OPC - 开放式互连

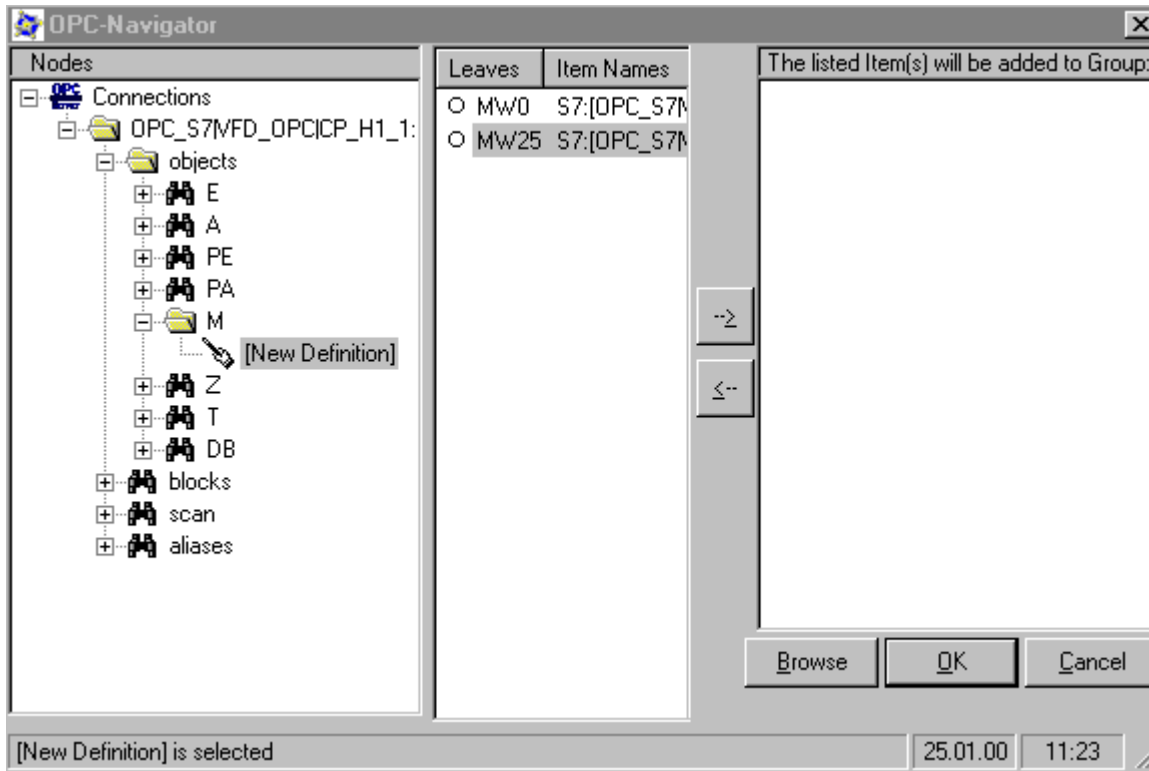
步骤

1. 通过“开始” → “程序” → “SimaticNet” → “OPCServer” → “OPCScout”打开“OPC Scout”。

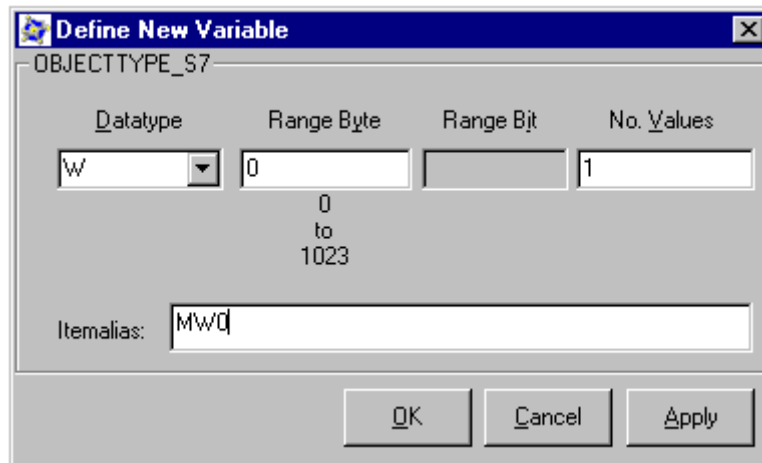


2. 选择“Local Server(s)”（本地服务器）下的“OPC.SimaticNet”。如果 SIMATIC S7 OPC 服务器不是运行在同一台计算机上，则选择“Server(s)”（服务器）快捷菜单中的“Add Remote Server(s)”（添加远程服务器）。在“Add Remote Server(s)”（添加远程服务器）对话框中输入用作 OPC 服务器的计算机名称，然后单击“OK”（确定）关闭对话框。
3. 选择“OPC.SimaticNet”快捷菜单中的“Connect”（连接）。显示“Add Group”（添加组）对话框。为该组输入一个名称。单击“OK”（确定）关闭对话框。

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 126)

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_SimaticNET”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。

- 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”。
- 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。
- 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 123)

在 WinCC OPC DA 客户机上组态 OPC 通道 (页 81)

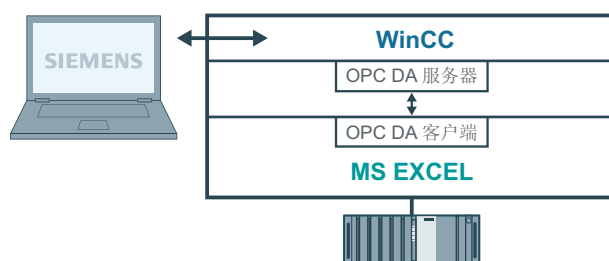
WinCC - Microsoft Excel 的连接

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：

- 在 Microsoft Excel 的 Visual Basic 编辑器中创建一个 OPC DA 客户机
- 在 Microsoft Excel 中组态对 WinCC 变量的访问

6.2 OPC - 开放式互连

参见

如何组态在 Microsoft Excel 中访问 WinCC 变量 (页 131)

在 Microsoft Excel 中创建 OPC DA 客户机 (页 128)

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

```

6.2 OPC - 开放式互连

```
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 117)

如何组态在 Microsoft Excel 中访问 WinCC 变量

引言

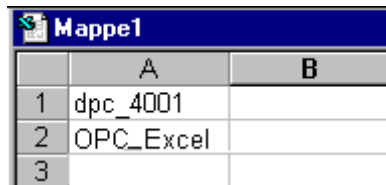
Excel OPC DA 客户机读取 WinCC OPC DA 服务器的 WinCC 变量并将变量值写入单元格中。在 WinCC OPC DA 服务器的 WinCC 项目中，变量值显示在一个 I/O 域内。如果单元格中的变量值改变，则在 WinCC OPC DA 服务器 I/O 域中的数值也随之改变。

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。

6.2.8 WinCC OPC HDA 服务器

6.2.8.1 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 139)

WinCC OPC HDA 服务器的数据结构 (页 133)

www.opcfoundation.org (<http://www.opcfoundation.org>)

6.2.8.2 WinCC OPC HDA 服务器的数据结构

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。

6.2 OPC - 开放式互连

	描述
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 136)

WinCC OPC HDA 服务器的时间格式 (页 137)

受支持属性的概述 (页 134)

受支持配件的概述 (页 135)

www.opcfoundation.org (<http://www.opcfoundation.org>)

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 133)

www.opcfoundation.org (<http://www.opcfoundation.org>)

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。
OPCHDA_MAXIMUMACTUALTIME	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。

6.2 OPC - 开放式互连

配件	描述
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 133)

WinCC OPC HDA 服务器的功能 (页 132)

www.opcfoundation.org (<http://www.opcfoundation.org>)

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 132)

www.opcfoundation.org (<http://www.opcfoundation.org>)

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

+/-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 132)

www.opcfoundation.org (<http://www.opcfoundation.org>)

6.2.8.3 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
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_DATA_LOST	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>)

6.2.8.4 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。

6.2 OPC - 开放式互连

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。
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。

6.2.8.5 OPC HDA 连接实例

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 144)

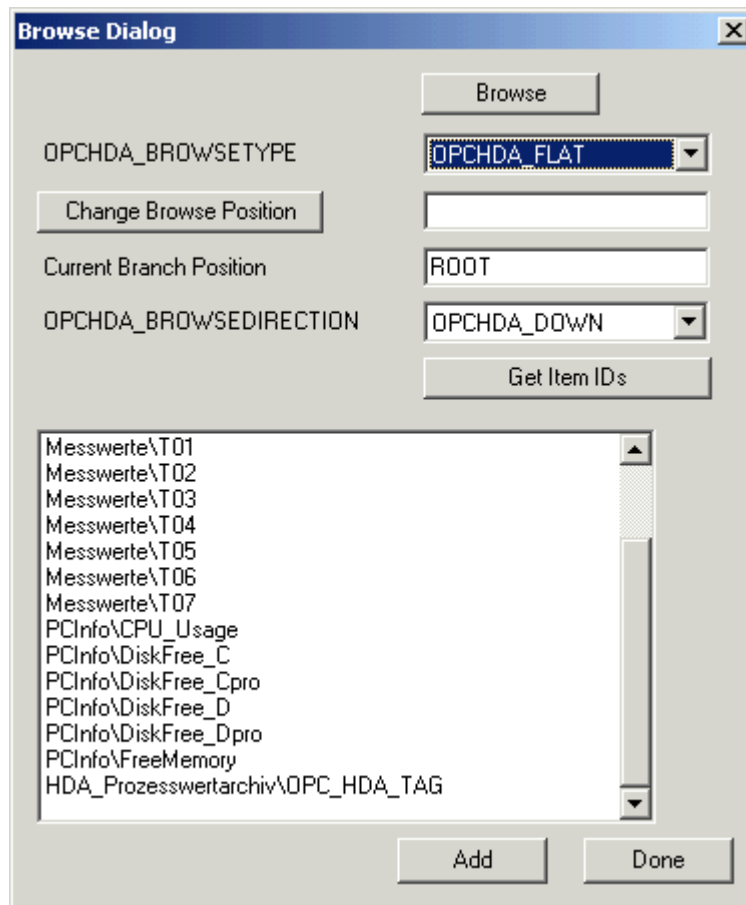
HDA 服务器浏览器 (页 143)

读取 WinCC 归档变量的数值 (页 145)

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。

6.2 OPC - 开放式互连

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 144)

www.opcfoundation.org (<http://www.opcfoundation.org>)

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

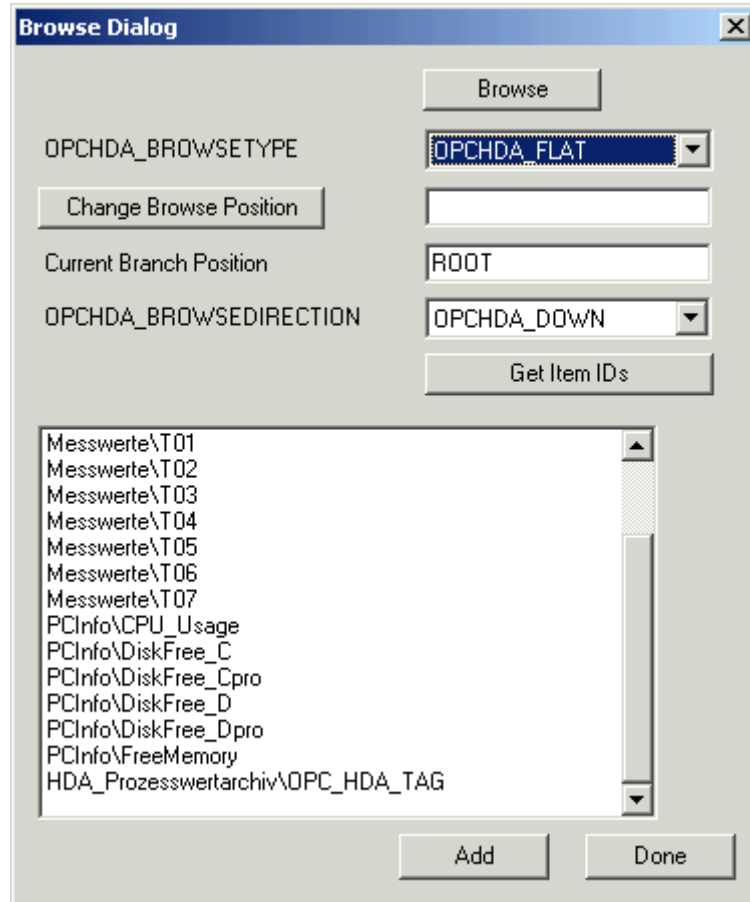
此处的 OPC HDA 客户机是来自 OPC 基金会的演示客户端。

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 145)

www.opcfoundation.org (<http://www.opcfoundation.org>)

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。

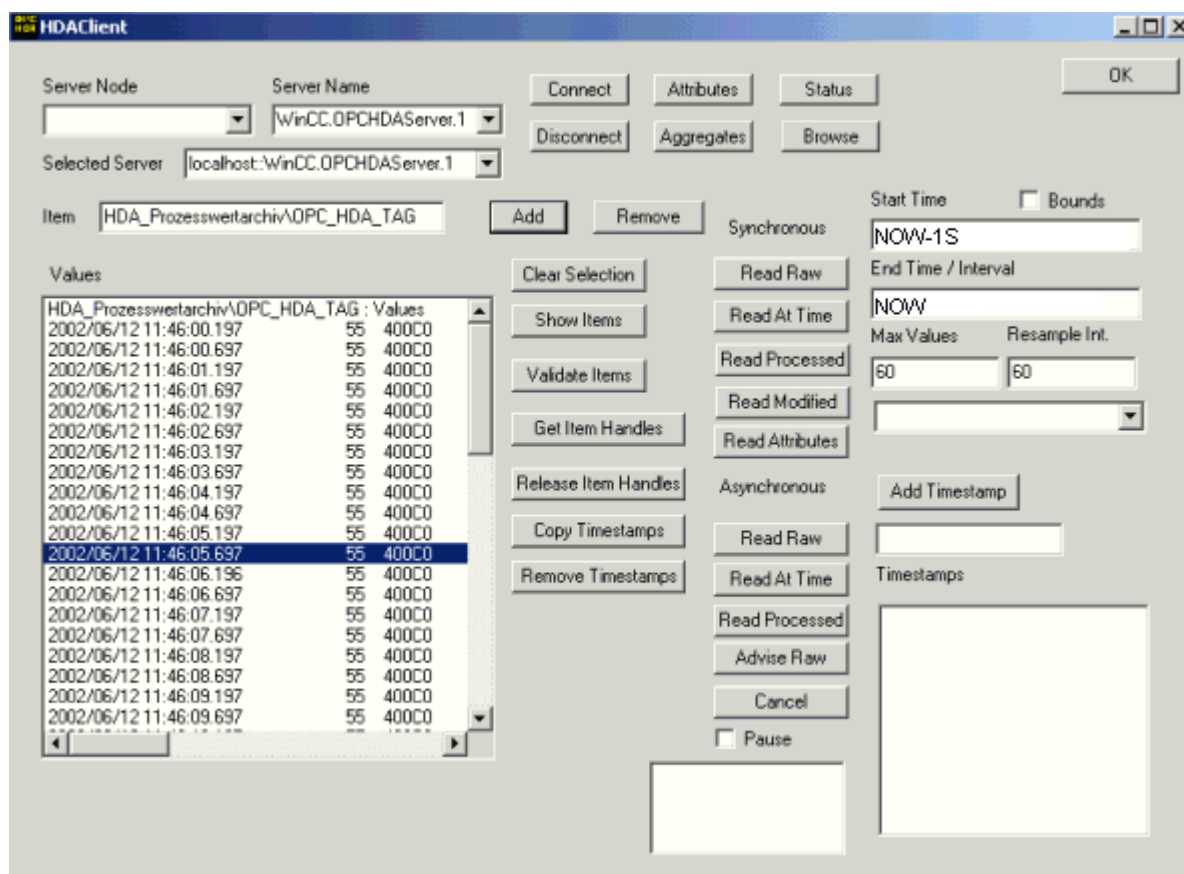
6.2 OPC - 开放式互连

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。

6.2.8.6 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

6.2 OPC - 开放式互连

- 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

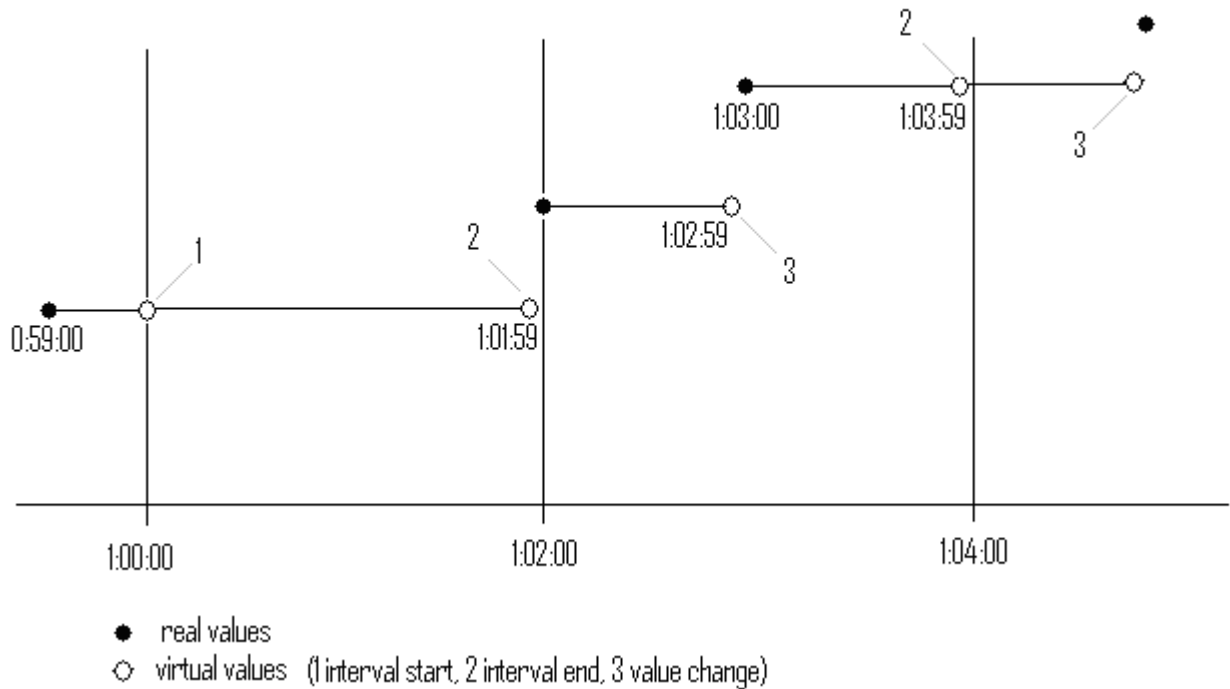
说明

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 μ s。

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



6.2 OPC - 开放式互连

6.2.9 WinCC OPC A&E 服务器

6.2.9.1 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 158)

在 OPC A&E 上映射 WinCC 消息系统 (页 151)

www.opcfoundation.org (<http://www.opcfoundation.org>)

6.2.9.2 在 OPC A&E 上映射 WinCC 消息系统

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 156)
- WinCC 消息系统的属性 (页 154)
- 映射 WinCC 消息类别和消息类型 (页 153)

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。

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。

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

6.2 OPC - 开放式互连

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

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

6.2 OPC - 开放式互连

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>)

6.2.9.3 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。

6.2.9.4 OPC A&E 连接实例

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取 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 159)

www.opcfoundation.org (<http://www.opcfoundation.org>)

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>。

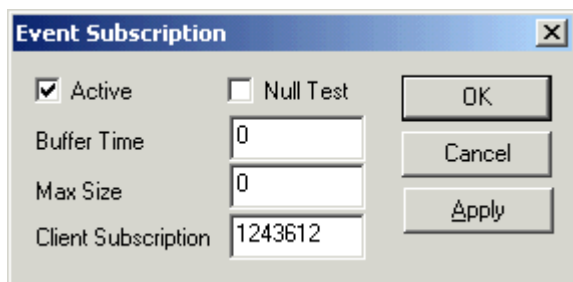
要求

- 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。

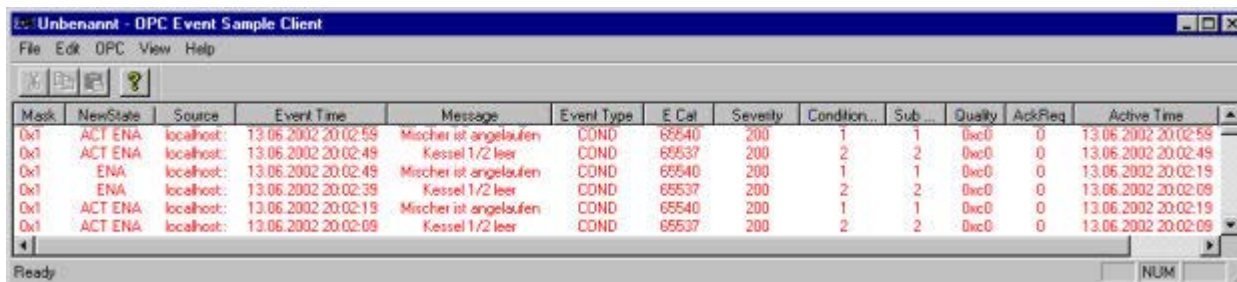
6.2 OPC - 开放式互连

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

OPC 客户机要求返回值	WinCC 使用
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>)

6.2.9.5 带层级访问的 OPC A&E 服务器

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。

6.2 OPC - 开放式互连

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

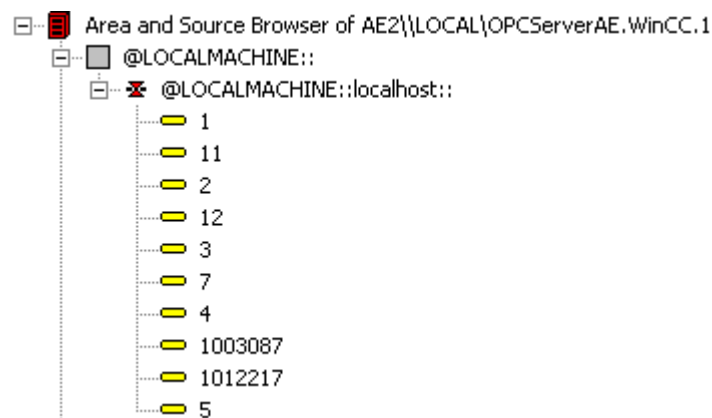
可以在“接口 > OPC - 用于过程控制的 OLE”一章中找到有关 OPC 的附加信息。

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

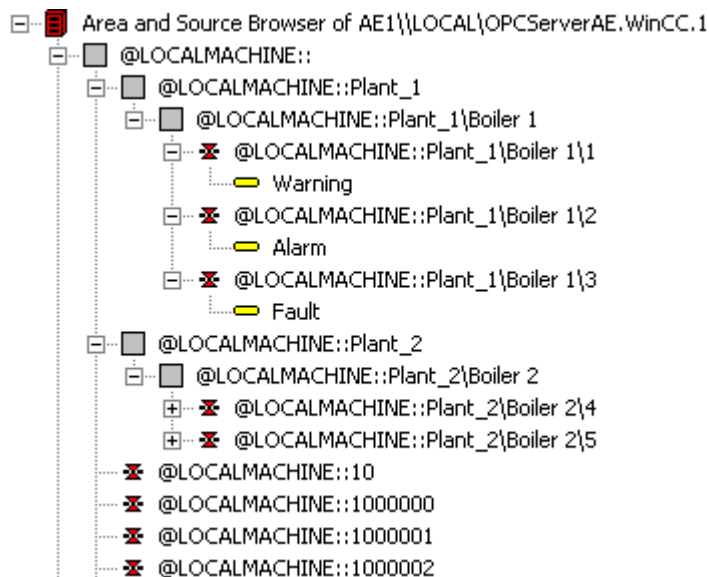
带层级访问功能的 OPC A&E 服务器支持的事件类型有条件事件、简单事件和跟踪事件。

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。

6.2 OPC - 开放式互连

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。

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
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

6.2 OPC - 开放式互连

OPC	WinCC 消息系统	事件类型
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
FOREGOLOR	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
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

6.2 OPC - 开放式互连

OPC 属性	WinCC 消息系统	数据类型
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	输出了处理了消息的计算机的名称。	VT_BSTR
USER	输出了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

导言

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

6.2 OPC - 开放式互连

WinCC	NewState	ChangeState
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
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。

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。

6.2.9.6 读取归档消息

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。

6.2 OPC - 开放式互连

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。

使用 OPC 访问归档消息的语法

语法

```
OPCHAEServer hMode=(read|advise) htStartTime=szTime
[hEndTime=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。

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。

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。

```
OPCHAEsServer 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。

6.2 OPC - 开放式互连

```
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
```

6.2.10 WinCC OPC UA 服务器

6.2.10.1 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 185)”。

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

6.2.10.2 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

6.2 OPC - 开放式互连

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 198)”。

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。

6.2 OPC - 开放式互连

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

WinCC OPC UA 服务器的组态文件 (页 198)

6.2.10.3 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:

6.2 OPC - 开放式互连

```

<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>

```

通过该组态，WINCC OPC UA 服务器支持匿名用户和 Policy“UserName”。

6.2.10.4 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

6.2 OPC - 开放式互连

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 139)”		

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

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

6.2.10.5 WinCC OPC UA 服务器的名称区域

简介

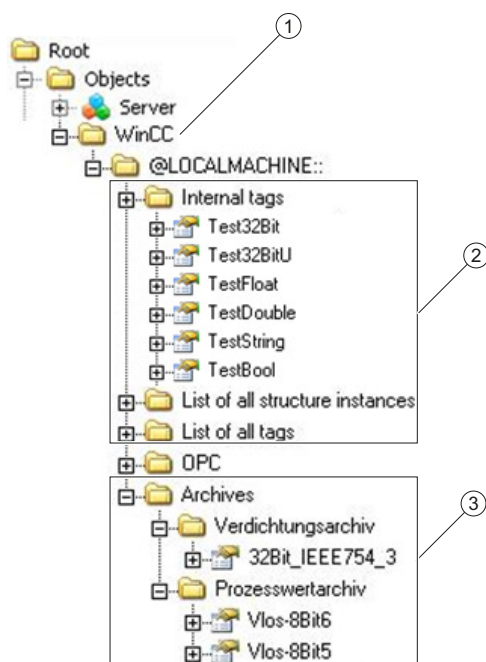
WinCC OPC UA 服务器为 OPC UA 客户端提供了一个层级名称区域以及以下运行系统数据的访问权限：

- 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：

6.2 OPC - 开放式互连



- 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
NodeId	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	-

属性	说明	注释
DataType	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	-

6.2 OPC - 开放式互连

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。

6.2.10.6 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。

6.2.10.7 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

6.2.10.8 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。

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

6.2 OPC - 开放式互连

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

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

6.2 OPC - 开放式互连

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 196)

6.2.10.9 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

6.2 OPC - 开放式互连

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 192)

6.2.10.10 WinCC OPC UA 服务器的组态

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 203)”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 177)”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 177)”。

<pre> <ServerConfiguration> <SecurityPolicies> <SecurityPolicy> <...></...> </SecurityPolicy> ... </SecurityPolicies> </pre>	<p>消息安全模式的组态。</p> <p>“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</p> <p>“Anonymous”设置仅用于测试和诊断目的</p>

<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 177)”下的实例证书上找到更多信息。

<pre><ServerConfiguration> <CertificateDescriptor></pre>	
<pre><OrganizationUnit>...</...> ...> <Organization>...</...> <Country>...</...></pre>	描述性元素 参数可以更改，且不会影响应用程序的功能。
<pre><KeyLength>...</...></pre>	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⁾
<pre><SignatureAlgorithm>... </...></pre>	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 Sha256 ¹⁾

6.2 OPC - 开放式互连

<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。

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 198)”。

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 177)”
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 177)”。
- 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>
```

6.2.11 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”。

使用 OLE DB 提供者访问的实例

7.1 使用 OLE DB 提供者访问的实例

概述

下列实例显示了显示和分析归档数据的不同方法。

参见

示例：使用 VB 组态对归档数据的访问 (页 208)

实例：使用 DataConnector Wizard 组态对归档数据的访问 (页 210)

实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取过程值归档 (页 215)

实例：使用 WinCC OLE DB 提供程序和 Visual Basic.NET 读取过程值归档 (页 223)

实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取消息归档数据 (页 228)

示例：使用 WinCC OLE DB 提供程序和 Visual Basic .NET 读取消息归档数据 (页 231)

7.2 示例：使用 VB 组态对归档数据的访问

简介

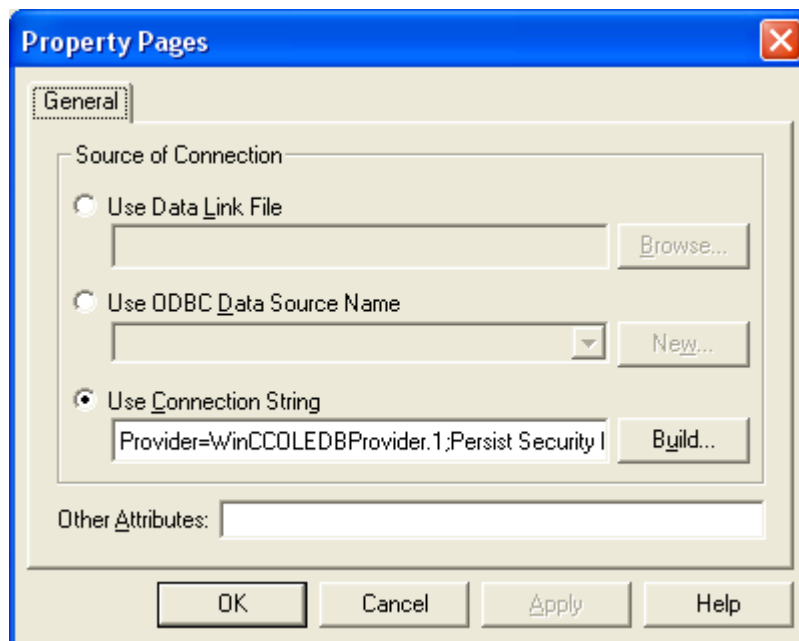
下列实例说明了如何使用 Microsoft 控件组态表格显示。也可以使用其它控件或编制自己的控件。

要求

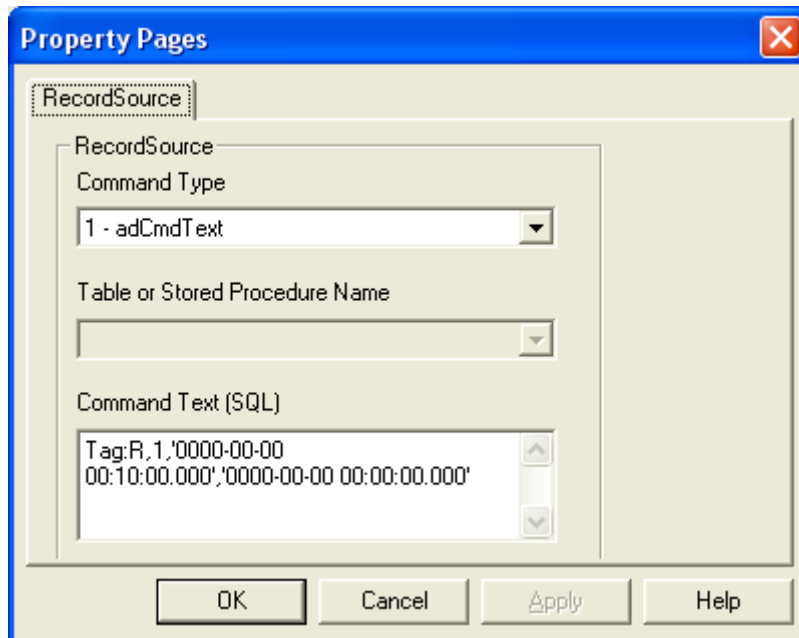
归档文件位于本地计算机上，并且必须连接到 SQL Server，如使用 WinCC 归档连接器进行连接。

步骤

1. 在“Visual Basic 编辑器”中，创建要用于显示归档数据的新窗体。
2. 在“Visual Basic 编辑器”中，选择菜单命令“工程 > 部件”(Project > Components)。
3. 激活要使用的控件，如“MS Data Grid Control”和“MS ADO Data Control”。相关的符号将显示在工具箱中。
4. 在窗体中，为归档数据的表格显示创建“MS Data Grid Control”，并且创建“MS ADO Data Control”以在表格显示内导航。
5. 标记“MS Data Grid Control”并选择属性窗口中“数据源”下的“Adoc1”条目，以便链接两个控件。
6. 标记“MS ADO Data Control”并单击“属性”窗口中“连接字符串”下的“...”按钮。这将打开“属性页”(Property pages)对话框。



7. 单击“使用连接字符序列”条目旁边的“创建...”按钮。
将打开“数据链接属性”(Data Link Properties)对话框。
8. 在“提供程序”(Provider)选项卡中,选择“用于存档的 WinCC OLE DB 提供程序”(WinCC OLE DB Provider for Archives)并单击“下一步”(Next)。
9. 在“连接”(Connection)选项卡的“数据源”(DataSource)框中,输入数据源: <computer name>\WinCC。
通过单击“测试连接”按钮测试连接。
选择“扩展”(Extended)选项卡。然后,切换到“所有”选项卡,不进行进一步的更改。此步骤是为随后创建的连接字符集正确设置所有参数的必需步骤。
10. 在“所有”(All)选项卡的“类别”(Catalog)框中,输入 WinCC 数据库的名称。数据库名称可在“SQL 企业管理器”中的“SQL Server 组”>“<计算机名称>\WinCC”>“数据库”>“<数据库名称_R>”处找到。
如果项目处于活动状态,则仅使用以“R”结尾的数据库名称。
如果已经通过 WinCC 归档连接器将导出的 WinCC 归档连接到 SQL Server,请使用数据库或导出目录的符号名称。
使用“确定”按钮确认设置。
11. 标记“MS ADO Data Control”并单击“属性”窗口中“记录源”下的“...”按钮。这将打开“属性页”(Property pages)对话框。
选择“1 - adCmdText”设置作为命令类型并在“命令文本 (SQL)”(Command text (SQL))框中输入有效的语句,如 TAG:R,1,'0000-00-00 00:10:00.000','0000-00-00 00:00:00.000'。



要访问以下变量类型的过程值,请将“TAG:R”架构替换为“TAG_EX:R”:

- 文本变量 8 位字符集/16 位字符集
- 64 位 IEEE 754 浮点数

12. 使用“确定”按钮对输入进行确认,并启动 Visual Basic Runtime。

7.3 实例：使用 DataConnector Wizard 组态对归档数据的访问

简介

下列实例显示了如何使用“WinCC 图形编辑器”的“DataConnector Wizard”（DataConnector 向导）组态过程值的表格显示。

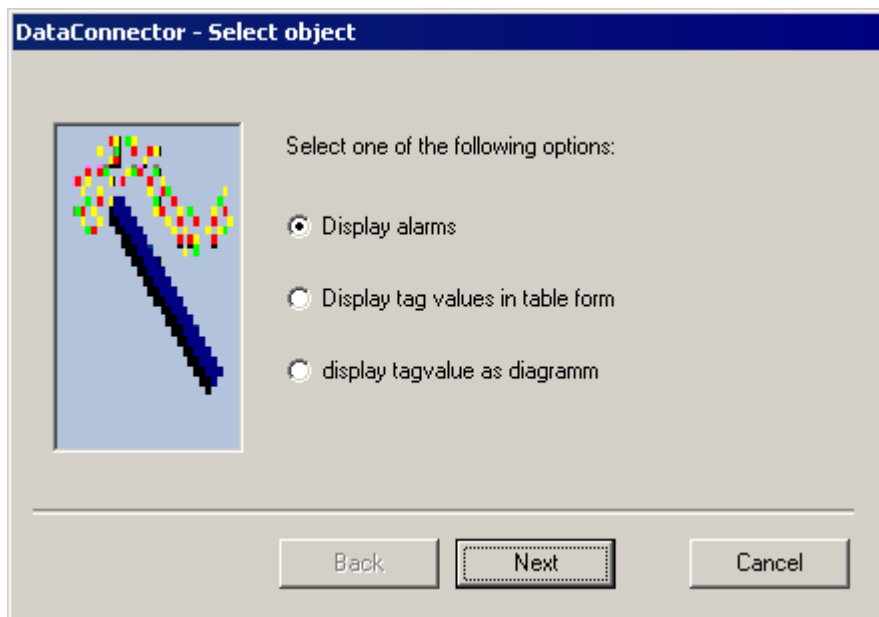
或者，也可以使用“DataConnector Wizard”（DataConnector 向导）组态消息的显示或过程值的图形显示。

要求

- 已安装“WinCC 基本系统”和“连通性软件包”。
- 归档文件已连接到所需的 SQL Server，如使用 WinCC 归档连接器进行连接。

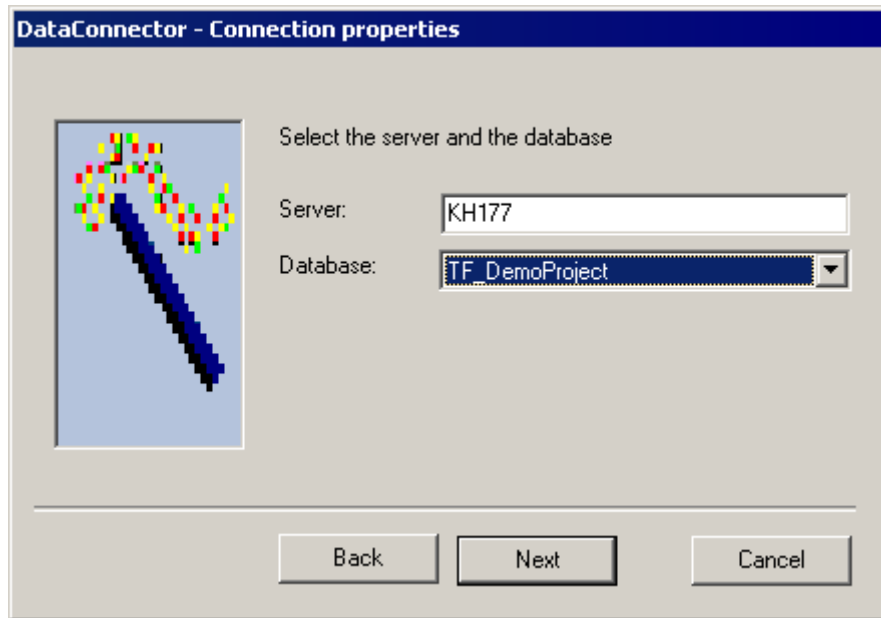
步骤

1. 打开“图形编辑器”，并在“DataConnector”菜单上选择“Create New Picture”（创建新画面）条目。将打开新的过程画面，并且显示“DataConnector - Select Object”（DataConnector - 选择对象）对话框。
2. 对于此实例，选择“Display Tag Values in Table Form”（在表格窗体中显示变量值），然后单击“Continue”（继续）按钮。



7.3 实例：使用 DataConnector Wizard 组态对归档数据的访问

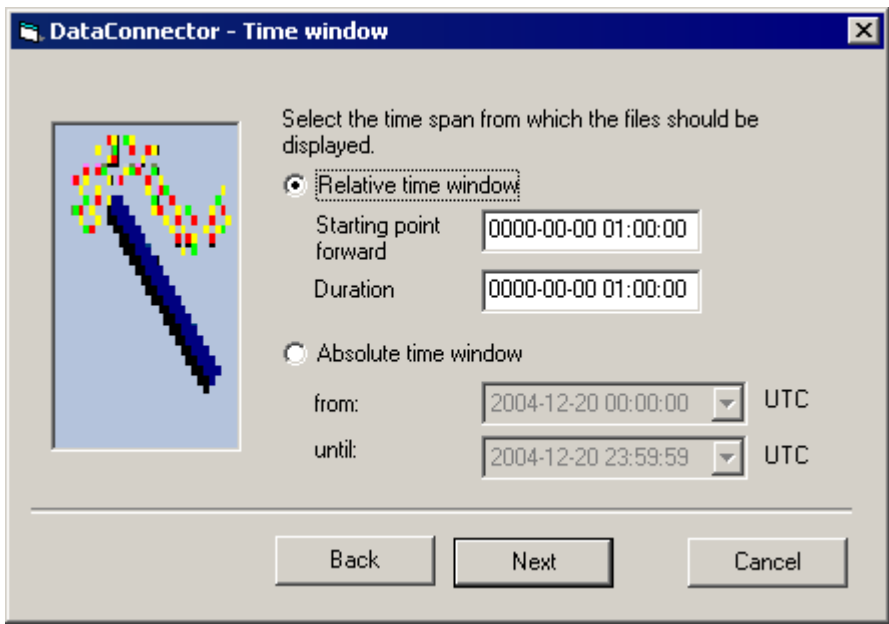
3. 将打开“DataConnector - 连接属性”(DataConnector - Connection Properties) 对话框。
在“服务器”(Server) 字段中，输入包含所需的互连数据库的服务器的名称。本地计算机的名称已指定。
在“数据库”(Database) 字段中，选择所需的数据库。所选服务器上当前连接的数据库可供选择。



单击“继续”(Continue)。

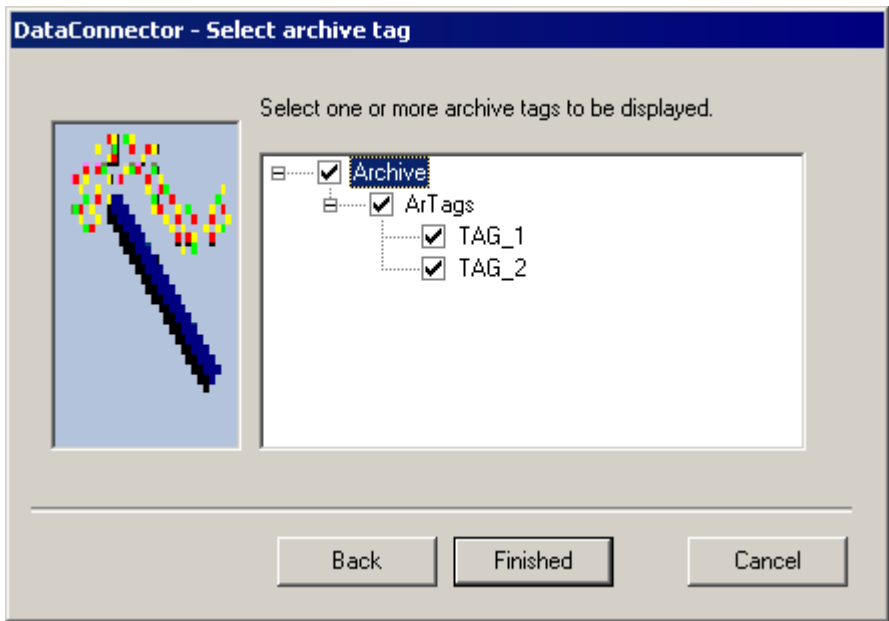
7.3 实例：使用 DataConnector Wizard 组态对归档数据的访问

- 4. 将打开“DataConnector - 时间窗口”(DataConnector - Time window) 对话框。
选择所需的时间范围，数据应在该时间范围内显示，如“相对时间窗口”(Relative time window)。已指定了前一小时的时间窗口。根据需要更改指定值，例如在“向前起点”(Starting point forward) 字段中将值更改为“0000-00-00 01:00:00”，在“持续时间”(Duration) 字段中将值更改为“0000-00-00 01:00:00”，以显示前一小时。“持续时间”(Duration) 字段中的值不能大于“向前起点”(Starting point forward) 字段中的值。



单击“继续”(Continue)。

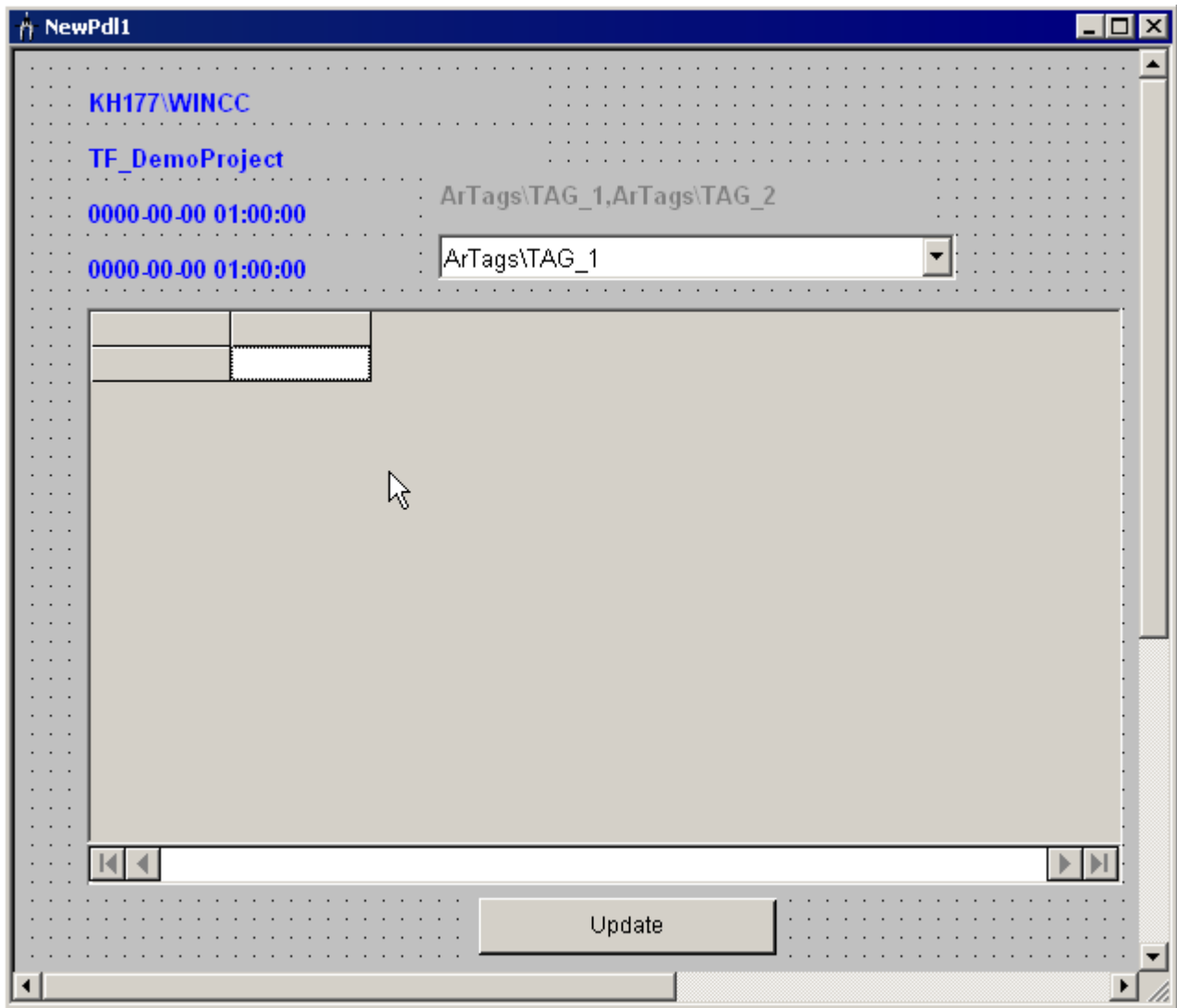
- 5. 将打开“DataConnector - 选择归档变量”(DataConnector - Select Archive Tag) 对话框。
至少选择一个归档变量。



单击“完成”(Finish) 按钮。

7.3 实例：使用 DataConnector Wizard 组态对归档数据的访问

6. “DataConnector 向导”(DataConnector Wizard) 将多个元素插入到打开的过程画面。
所选服务器、数据库和时间范围显示在顶部的静态文本字段中。
这些静态文本字段旁是包含所选归档变量的文本和归档变量的选择字段。
根据所做的选择“将变量值显示为表格”(Display tag values as tables)，数据格网控件、ADO 数据控件和按钮会插入在这些静态文本字段的下方。
保存过程画面。
为了更改组态的值，请打开此过程画面，并选择“DataConnector”菜单中的“修改现有连接”(Modify existing connection) 条目。向导将重新启动，您将可以更改组态的值。



7. 启动 WinCC Runtime，并打开这一具有数据库查询功能的过程画面。打开过程画面时，即执行此查询。
在运行系统中，找到的数据将以表格形式显示在“数据网格控件”中，并且可以使用“ADO 数据控件”在表格中进行浏览。
存在多个组态的归档变量时，可以在选择字段中选择要显示其值的变量。
显示的数据可以通过“更新”(Update) 按钮来更新，例如，是否将上一小时以相对时间范围显示。

7.3 实例：使用 DataConnector Wizard 组态对归档数据的访问

说明

使用 Web 客户机上的数据连接器控件进行归档查询

如果要在 WebNavigator 客户机上使用过程画面中的 DataConnector 控件执行归档查询，则还必须在 Web 客户机上安装连通性软件包客户机。

参见

查询报警消息归档 (页 53)

查询过程值归档 (页 45)

7.4 实例：分析过程值归档

7.4.1 实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取过程值归档

简介

在此示例中，从本地 WinCC RT 数据库读取“Tag_1”变量的最后 10 分钟的值。

数据显示在带有时间戳、值和质量代码的 ListView 中。值的输出限制在 100（常量 NMAX = 100）以内。

示例中未包含错误处理。

操作步骤

1. 创建名为“Tag_1”的 WinCC 变量。
2. 创建名为“ArTags”的过程值归档。
将 WinCC 变量“Tag_1”与过程值归档连接起来。
3. 创建 Visual Basic 项目。
将 MS Windows Common Controls 6.0 的 ListView 控件与名称“ListView1”连接起来。
ListView 中的各列由脚本创建。
将“视图”属性设置为 ListView 控件的“3 - lvwReport”选项。
4. 创建“CommandButton”。
从下面的“示例脚本”一节中复制脚本。
5. 根据项目调整脚本：
 - 更改 WinCC Runtime 数据库“CC_OpConPac_04_10_01_10_36_36R”的名称。
数据库名称可在“SQL 企业管理器”中的“SQL Server 组”>“<计算机名称>WinCC”>“数据库”>“<数据库名称_R>”处找到。
 - 如有必要，可以更改所用架构的名称。
要访问以下变量类型的过程值，请使用“TAG_EX”架构：
 - 文本变量 8 位字符集/16 位字符集
 - 64 位 IEEE 754 浮点数
6. 激活 WinCC Runtime 并启动 VB 应用程序。
7. 单击“CommandButton”。

7.4 实例：分析过程值归档

示例脚本

说明

改善性能

为提高性能，请注意以下信息：

- 用“Data Source=<计算机名称>\WinCC”指令替换“Data Source=. \WinCC”指令。
 - 对于查询，使用“ValueID”，而不是“<归档名称>\<变量名称>”。
在示例脚本中，必须用“ValueID”替换点 1.2 下的表达式“ArTags\Tag_1”。
-


```
Option Explicit
Const NMAX = 100
Private Sub Command1_Click()
Dim sPro As String
Dim sDsn As String
Dim sSer As String
Dim sTagSchema As String
Dim sCon As String
Dim sSql As String
Dim conn As Object
Dim oRs As Object
Dim oCom As Object

' 1.1 Make connection string for ADODB
sPro = "Provider=WinCCOLEDBProvider.1;"
sDsn = "Catalog=CC_Test_19_05_21_10_23_19R;"
sSer = "Data Source=.\WinCC"
sCon = sPro + sDsn + sSer

' 1.2 Define the Tag Schema Name (TAG or TAG_EX)
sTagSchema = "TAG"

' 1.3 Define command text in sSql (relative time)
sSql = sTagSchema + ":R,'ArTags\Tag_1','0000-00-00
00:10:00.000','0000-00-00 00:00:00.000'"

MsgBox "Open with:" & vbCrLf & sCon & vbCrLf & sSql & vbCrLf

' 2.1 Make connection
Set conn = CreateObject("ADODB.Connection")
conn.ConnectionString = sCon
```

7.4 实例：分析过程值归档

```
conn.CursorLocation = 3
conn.Open

' 2.2 Use command text for query
Set oRs = CreateObject("ADODB.Recordset")
Set oCom = CreateObject("ADODB.Command")
oCom.CommandType = 1
Set oCom.ActiveConnection = conn
oCom.CommandText = sSql

' 2.3 Fill the recordset
Set oRs = oCom.Execute

' 3.1 Fill standard listview object with recordset according to
schema
If sTagSchema = "TAG" Then
    Display oRs
Else
    DisplayEx oRs
End If

Set oRs = Nothing
conn.Close
Set conn = Nothing

End Sub

' Display recordset for 'TAG' schema
Sub Display(oRs_in)

Dim oItem As ListItem
Dim strDateTime As String
```

```
Dim iMS As Long

Dim m, n, s, nRec
m = oRs_in.Fields.Count
n = 0
nRec = NMAX

' 3.2 Fill standard listview object with recordset
ListView1.ColumnHeaders.Clear

ListView1.ColumnHeaders.Add , , CStr(oRs_in.Fields(1).Name),
2500      ' DateTime
ListView1.ColumnHeaders.Add , , "MS", 1500
' Millisecond
ListView1.ColumnHeaders.Add , , CStr(oRs_in.Fields(0).Name),
1500      ' VarID
ListView1.ColumnHeaders.Add , , CStr(oRs_in.Fields(2).Name),
1500      ' RealValue
ListView1.ColumnHeaders.Add , , CStr(oRs_in.Fields(3).Name),
1500      ' Quality
ListView1.ColumnHeaders.Add , , CStr(oRs_in.Fields(4).Name),
1500      ' Flags

If (m > 0) Then
oRs_in.MoveFirst
n = 0
Do While (Not oRs_in.EOF And n < nRec)
n = n + 1
SplitDateTimeAndMs oRs_in.Fields(1).Value, strDateTime, iMS
'Split Millisecond from DateTime
s = FormatDateTime(strDateTime, 2) & " " &
FormatDateTime(strDateTime, 3)
Set oItem = ListView1.ListItems.Add()
```

7.4 实例：分析过程值归档

```
oItem.Text = s
oItem.SubItems(1) = iMS
oItem.SubItems(2) = oRs_in.Fields(0).Value
oItem.SubItems(3) = FormatNumber(oRs_in.Fields(2).Value, 4)
oItem.SubItems(4) = Hex(oRs_in.Fields(3).Value)
oItem.SubItems(5) = Hex(oRs_in.Fields(4).Value)

oRs_in.MoveNext
Loop
oRs_in.Close
Else
End If

End Sub

' Display recordset for 'TAG_EX' schema
Sub DisplayEx(oRs_in)

Dim oItem As ListItem
Dim strDateTime As String
Dim iMS As Long

Dim m, n, s, nRec
m = oRs_in.Fields.Count
n = 0
nRec = NMAX

' 3.3 Fill standard listview object with recordset
ListView1.ColumnHeaders.Clear
ListView1.ColumnHeaders.Add , , CStr(oRs_in.Fields(1).Name),
2500 ' DateTime
```

```
Listview1.ColumnHeaders.Add , , "MS", 1500
' Millisecond

Listview1.ColumnHeaders.Add , , CStr(oRs_in.Fields(0).Name),
1500      ' VarID

Listview1.ColumnHeaders.Add , , CStr(oRs_in.Fields(3).Name),
1500      ' VariantValue

Listview1.ColumnHeaders.Add , , CStr(oRs_in.Fields(4).Name),
1500      ' Quality

Listview1.ColumnHeaders.Add , , CStr(oRs_in.Fields(5).Name),
1500      ' Flags

If (m > 0) Then
oRs_in.MoveFirst
n = 0
Do While (Not oRs_in.EOF And n < nRec)
n = n + 1
SplitDateTimeAndMs oRs_in.Fields(1).Value, strDateTime, iMS
'Split Millisecond from DateTime
s = FormatDateTime(strDateTime, 2) & " " &
FormatDateTime(strDateTime, 3)
Set oItem = Listview1.ListItems.Add()
oItem.Text = s
oItem.SubItems(1) = iMS
oItem.SubItems(2) = oRs_in.Fields(0).Value
oItem.SubItems(3) = oRs_in.Fields(3).Value
oItem.SubItems(4) = Hex(oRs_in.Fields(4).Value)
oItem.SubItems(5) = Hex(oRs_in.Fields(5).Value)

oRs_in.MoveNext
Loop
oRs_in.Close
Else
End If
```

7.4 实例：分析过程值归档

```
End Sub

Sub SplitDateTimeAndMs(dtDateTimeWithMs_in, strDateTime_out,
strMs_out)

Dim diff
Dim dtTemp
Dim lTemp As Long

lTemp = 86400000 '24 * 60 * 60 * 1000

diff = (Cdbl(dtDateTimeWithMs_in) -
Cdbl(CDate(CStr(dtDateTimeWithMs_in)))) * lTemp
If (diff < 0) Then
diff = 1000 + diff
dtTemp = Cdbl(dtDateTimeWithMs_in) - diff / lTemp
strDateTime_out = CDate(dtTemp)
Else
strDateTime_out = CDate(dtDateTimeWithMs_in)
End If
strMs_out = "00" & Round(diff)
strMs_out = Right(strMs_out, 3)

End Sub
```

参见

- 显示过程值归档 (页 42)
- 查询过程值归档 (页 45)
- 查询归档数据 (页 43)
- 建立到归档数据库的连接 (页 39)

7.4.2 实例：使用 WinCC OLE DB 提供程序和 Visual Basic.NET 读取过程值归档

简介

在此实例中，从本地 WinCC RT 数据库读取“Tag_1”变量的最后 10 分钟的值。

数据将与时间戳、值和质量代码一起显示在 ListView 中。

值的输出限制在 100（常量 MaxRows = 100）以内。

该实例中包含简单的错误处理。

操作步骤

1. 创建名为“Tag_1”的 WinCC 变量。
2. 创建名为“ArTags”的过程值归档。
3. 将 WinCC 变量“Tag_1”与过程值归档连接起来。
4. 在 Visual Studio 中，创建“Windows 应用程序”类型的新 Visual Basic 项目。
5. 切换到“设计器视图”模式，并将 Windows 窗体“Form1”的大小扩大到约 500x600。
6. 将“ListView 控件”从工具箱拖动到窗体的左上角，并将其扩大到约 400x450。
7. 将按钮从工具箱拖动到窗体中间且位于“控件”下。
该按钮将被自动命名为“按钮 1”。
8. 双击按钮，然后将下面列出的 VB.NET 实例代码添加到方法“Private Sub Button1_Click”中的指令“End Sub”之前。
9. 选择菜单中的“生成 > 生成解决方案”，或按下 <CTRL+Shift+B> 编译已完成的 VB.NET 工程。
10. 根据项目调整脚本：
 - 更改 WinCC Runtime 数据库“CC_CPSample_07_10_09_13_44_26R”的名称。
数据库名称可在 SQL Server Management Studio 中的“对象浏览器 > <计算机名称> \WinCC > 数据库 > <数据库名称_R>”处找到。
 - 如有必要，可以更改所用架构的名称。
要访问以下变量类型的过程值，请使用“TAG_EX”架构：
 - 文本变量 8 位字符集/16 位字符集
 - 64 位 IEEE 754 浮点数
11. 启动 WinCC Runtime，然后选择菜单中的“调试 -> 开始调试”(Debug->Start Debugging) 或按 <F5> 启动 VB 应用程序。
12. 单击“Button1”按钮。

7.4 实例：分析过程值归档

示例脚本

```
Const MaxRows = 100

' 1.1 Create connection string for ADO.NET access
Dim ProviderName As String
Dim CatalogName As String
Dim DataSourceName As String
Dim ConnectionString As String
Dim CommandString As String
Dim TagSchema As String

ProviderName = "Provider=WinCCOLEDBProvider.1;"
CatalogName = "Catalog=CC_CPSample_07_10_09_13_44_26R;"
DataSourceName = "Data Source=.\WinCC"

ConnectionString = ProviderName + CatalogName + DataSourceName

' 1.2 Define the Tag Schema Name (TAG or TAG_EX)
TagSchema = "TAG"

' 1.3 Define command string for OLE DB access (relative time
range, last 10 minutes)
CommandString = TagSchema + ":R,'ArTags\Tag_1','0000-00-00
00:10:00.000','0000-00-00 00:00:00.000'"

Try

Dim DBConnection As System.Data.OleDb.OleDbConnection
Dim DBCommand As System.Data.OleDb.OleDbCommand
Dim DBReader As System.Data.OleDb.OleDbDataReader

' 2.1 Open an OLE DB connection
```



```
DBConnection = New
System.Data.OleDb.OleDbConnection(ConnectionString)

DBConnection.Open()

' 2.2 Execute command
DBCommand = New System.Data.OleDb.OleDbCommand(CommandString,
DBConnection)

DBReader = DBCommand.ExecuteReader()

' 3.1 Prepare list view control
Dim DBTimeStamp As System.DateTime
Dim DBValue As String
Dim DBQualityCode As Integer
Dim DBFlags As Integer
Dim ColumnEntry As String

Me.ListView1.View = View.Details
Me.ListView1.GridLines = True
Me.ListView1.TabIndex = 0
Me.ListView1.Width = 400

Me.ListView1.Columns.Add("TimeStamp",
HorizontalAlignment.Center)
Me.ListView1.Columns(0).Width = 150
Me.ListView1.Columns(0).Text = "Time Stamp"

Me.ListView1.Columns.Add("Value", HorizontalAlignment.Center)
Me.ListView1.Columns(1).Width = 60
Me.ListView1.Columns(1).Text = "Value"

Me.ListView1.Columns.Add("QC", HorizontalAlignment.Center)
```

7.4 实例：分析过程值归档

```
Me.ListView1.Columns(2).Width = 60
Me.ListView1.Columns(2).Text = "Quality Code"

Me.ListView1.Columns.Add("Flags", HorizontalAlignment.Center)
Me.ListView1.Columns(3).Width = 60
Me.ListView1.Columns(3).Text = "Flags"

Dim SingleRow As ListViewItem

Dim Count As Integer
Count = 0

' 3.2 Fetch subsequent rows of the result set
While (DBReader.Read And Count < MaxRows)

    DBTimeStamp = DBReader.GetDateTime(1)

    If TagSchema = "TAG" Then
        DBValue = DBReader.GetValue(2)
        DBQualityCode = DBReader.GetInt32(3)
        DBFlags = DBReader.GetInt32(4)
    Else
        DBValue = DBReader.GetValue(3)
        DBQualityCode = DBReader.GetInt32(4)
        DBFlags = DBReader.GetInt32(5)
    End If

' 3.3 Fill list view columns
ColumnEntry = String.Format("{0:dd.MM.yy HH:mm:ss.}{1:D2}",
DBTimeStamp, DBTimeStamp.Millisecond)
SingleRow = New ListViewItem(ColumnEntry)
```

```
ColumnEntry = String.Format("{0:F}", DBValue)
SingleRow.SubItems.Add(ColumnEntry)

ColumnEntry = String.Format("{0:X2}", DBQualityCode)
SingleRow.SubItems.Add(ColumnEntry)

ColumnEntry = String.Format("{0:X4}", DBFlags)
SingleRow.SubItems.Add(ColumnEntry)

Me.ListView1.Items.Add(SingleRow)
Count = Count + 1

End While

' 4 Clean up
DBReader.Close()
DBConnection.Close()

' 5 Exception (error) handling
Catch ex As Exception
MessageBox.Show(ex.Message, "Error Occurred !")

End Try
```

7.5 实例：分析报警消息归档

7.5.1 实例：使用 WinCC OLE DB 提供程序和 Visual Basic 6 读取消息归档数据

简介

在本实例中，读取报警消息归档数据的时间间隔是 10 分钟。

数据显示在带有时间戳、消息号、状态和消息类型的 ListView 控件中。

该实例中包含简单的错误处理。

步骤

1. 创建 Visual Basic 项目。
将 MS Windows Common Controls 6.0 ListView 控件与名称“ListView1”连在一起。
2. ListView 中的各列由脚本创建。
3. 创建“CommandButton”。
从下面的“实例脚本”一节中复制脚本。
4. 在脚本中，将 WinCC 运行系统数据库的名称“CC_OpenArch_03_05_27_14_11_46R”更改为您自己项目的名称。
数据库名称可在“SQL 企业管理器”中的“SQL Server 组”>“<计算机名称>/WinCC”>“数据库”>“<数据库名称_R>”处找到。
5. 此示例使用英语数据库表“AlgViewEXEnu”。
如果要读取其它语言的报警消息归档数据，则需要修改此条目。
6. 激活 WinCC Runtime。
7. 启动 VB 应用程序。
8. 单击“CommandButton”。

实例脚本

说明

改善性能

为了提高性能，用“Data Source=<计算机名称>\WinCC”指令替换“Data Source=. \WinCC”指令。
值的输出限制在 100 (n >100) 以内。

所显示的时间是系统时间。

```
Dim sPro As String
Dim sDsn As String
Dim sSer As String
Dim sCon As String
Dim sSql As String

Dim conn As Object
Dim oRs As Object
Dim oCom As Object
Dim oItem As ListItem

Dim m, n, s

On Error GoTo ErrorHandlerA

' 1.1 Make connection string for ADODB
sPro = "Provider=WinCCOLEDBProvider.1;"
sDsn = "Catalog=CC_OpenArch_03_05_27_14_11_46R;"
sSer = "Data Source=.\WinCC"
sCon = sPro + sDsn + sSer

' 1.2 Define command text in sSql
sSql = "ALARMVIEWEX:Select * FROM AlgViewEXenu WHERE
DateTime>'2003-07-30 11:30:00'
AND DateTime<'2003-07-30 11:40:00'"
'sSql = "ALARMVIEWEX:Select * FROM AlgViewEXenu WHERE MsgNr = 5"
'sSql = "ALARMVIEWEX:Select * FROM AlgViewEXenu"
MsgBox "Open with:" & vbCr & sCon & vbCr & sSql & vbCr

' 2.1 Make connection
Set conn = CreateObject("ADODB.Connection")
```

7.5 实例：分析报警消息归档

```
conn.ConnectionString = sCon
conn.CursorLocation = 3
conn.Open

' 2.2 Use command text for query
Set oRs = CreateObject("ADODB.Recordset")
Set oCom = CreateObject("ADODB.Command")
oCom.CommandType = 1
Set oCom.ActiveConnection = conn
oCom.CommandText = sSql

' 2.3 Fill the recordset
Set oRs = oCom.Execute
m = oRs.Fields.Count

' 3.0 Fill standard listview object with recordset
ListView1.ListItems.Clear
ListView1.ColumnHeaders.Clear
ListView1.ColumnHeaders.Add , , CStr(oRs.Fields(2).Name), 140
ListView1.ColumnHeaders.Add , , CStr(oRs.Fields(0).Name), 60
ListView1.ColumnHeaders.Add , , CStr(oRs.Fields(1).Name), 60
ListView1.ColumnHeaders.Add , , CStr(oRs.Fields(34).Name), 100
If (m > 0) Then
oRs.MoveFirst
n = 0
Do While Not oRs.EOF
n = n + 1
If (n < 100) Then
s = Left(CStr(oRs.Fields(1).Value), 23)
Set oItem = ListView1.ListItems.Add()
oItem.Text = CStr(oRs.Fields(2).Value)
oItem.SubItems(1) = CStr(oRs.Fields(0).Value)
```

```
oItem.SubItems(2) = CStr(oRs.Fields(1).Value)
oItem.SubItems(3) = CStr(oRs.Fields(34).Value)
End If
oRs.MoveNext
Loop
oRs.Close
Else

End If
Set oRs = Nothing
conn.Close
Set conn = Nothing

ErrorHandlerA:
MsgBox Err.Description
```

参见

建立到归档数据库的连接 (页 39)
显示报警消息归档 (页 55)
查询报警消息归档 (页 53)
查询归档数据 (页 43)

7.5.2 示例：使用 WinCC OLE DB 提供程序和 Visual Basic .NET 读取消息归档数据

简介

在本实例中，读取报警消息归档数据的时间间隔是 10 分钟。
数据显示在带有时间戳、消息号、状态和消息类型的 ListView 控件中。
该实例中包含简单的错误处理。

7.5 实例：分析报警消息归档

步骤

1. 在 Visual Studio 2005 中创建新的 Visual Basic 工程，类型为“Windows 应用程序”。
2. 切换到“设计器视图”模式，并将 Windows 窗体“Form1”的大小扩大到 ca. 600x600。
3. 将“ListView 控件”从工具箱拖动到窗体的左上角，并将其扩大到 ca. 500x450。
4. 将按钮从工具箱拖动到窗体中间且位于“控件”下。
该按钮将被自动命名为“按钮 1”。
5. 双击按钮，然后将下面列出的 VB.NET 实例代码添加到方法“Private Sub Button1_Click”中的指令“End Sub”之前。
6. 选择菜单中的“生成 > 生成解决方案”，或按下 <CTRL+Shift+B> 编译已完成的 VB.NET 工程。
7. 在脚本中，将 WinCC 运行系统数据库的名称“CC_CPSample_07_10_09_13_44_26R”更改为自己的项目。
数据库名称可在 SQL Server Management Studio 中的“对象浏览器 > <计算机名称> \WinCC > 数据库 > <数据库名称_R>”处找到。
8. 激活 WinCC Runtime。
9. 选择菜单中的“调试 > 开始调试”(Debug > Start Debugging) 或按 <F5> 启动 VB 应用程序。
10. 单击“Button1”按钮。

实例脚本

```
Const MaxRows = 100
' 1.1 Create connection string for ADO.NET access
Dim ProviderName As String
Dim CatalogName As String
Dim DSN As String
Dim ConnectionString As String
Dim CommandString As String

ProviderName = "Provider=WinCCOLEDBProvider.1;"
CatalogName = "Catalog=CC_CPSample_07_10_09_13_44_26R;"
DSN = "Data Source=.\WinCC"

ConnectionString = ProviderName + CatalogName + DSN

' 1.2 Define command string for ADO.NET access
CommandString = "ALARMVIEWEX:Select * From AlgViewEXenu WHERE DateTime > '2007-10-10 12:00:00' AND DateTime > '2007-10-10 12:10:00'"
'CommandString = "ALARMVIEWEX:Select * From AlgViewEXenu WHERE MsgNr = 1"
'CommandString = "ALARMVIEWEX:Select * From AlgViewEXenu"

Try

Dim DBConnection As System.Data.OleDb.OleDbConnection
Dim DBCommand As System.Data.OleDb.OleDbCommand
Dim DBReader As System.Data.OleDb.OleDbDataReader

Dim DBConnection As System.Data.OleDb.OleDbConnection
Dim DBCommand As System.Data.OleDb.OleDbCommand
Dim DBReader As System.Data.OleDb.OleDbDataReader

' 2.1 Open an OLE DB connection
DBConnection = New System.Data.OleDb.OleDbConnection(ConnectionString)

DBConnection.Open()

' 2.2 Execute command
DBCommand = New System.Data.OleDb.OleDbCommand(CommandString, DBConnection)

DBReader = DBCommand.ExecuteReader()

' 3.1 Prepare list view control
Dim ColumnEntry As String

Me.ListView1.View = View.Details
Me.ListView1.GridLines = True
Me.ListView1.TabIndex = 0
Me.ListView1.Width = 500
```

7.5 实例：分析报警消息归档

```
Me.ListView1.Columns.Add("Number", HorizontalAlignment.Center)
Me.ListView1.Columns(0).Width = 90
Me.ListView1.Columns(0).Text = "Number"

Me.ListView1.Columns.Add("Date/Time", HorizontalAlignment.Center)
Me.ListView1.Columns(1).Width = 130
Me.ListView1.Columns(1).Text = "Date/Time"

Me.ListView1.Columns.Add("State", HorizontalAlignment.Center)
Me.ListView1.Columns(2).Width = 60
Me.ListView1.Columns(2).Text = "State"

Me.ListView1.Columns.Add("Type", HorizontalAlignment.Center)
Me.ListView1.Columns(3).Width = 150
Me.ListView1.Columns(3).Text = "Type"

Dim SingleRow As ListViewItem
Dim MsgDateTime As DateTime
Dim MsgMilliSecs As Integer
Dim MsgNumber As Integer
Dim MsgState As Integer
Dim MsgTypeName As String

Dim Count As Integer
Count = 0

' 3.2 Fetch subsequent rows of the result set
While (DBReader.Read And Count < MaxRows)

MsgNumber = DBReader.GetInt32(0)
MsgDateTime = DBReader.GetDateTime(2)
MsgMilliSecs = DBReader.GetInt16(3)
MsgState = DBReader.GetInt16(1)
MsgTypeName = DBReader.GetString(34)

' 3.3 Fill list view columns
ColumnEntry = String.Format("{0:D10}", MsgNumber)
SingleRow = New ListViewItem(ColumnEntry)

ColumnEntry = String.Format("{0:dd.MM.yy HH:mm:ss.}{1:D2}", MsgDateTime, MsgMilliSecs)
SingleRow.SubItems.Add(ColumnEntry)

ColumnEntry = String.Format("{0:D5}", MsgState)
SingleRow.SubItems.Add(ColumnEntry)

ColumnEntry = MsgTypeName
SingleRow.SubItems.Add(ColumnEntry)

Me.ListView1.Items.Add(SingleRow)

Count = Count + 1
```

```
End While

' 4 Clean up
DBReader.Close()
DBConnection.Close()

' 5 Exception (error) handling
Catch ex As Exception
    MessageBox.Show(ex.Message, "Error Occured !")

End Try
```

7.5 实例：分析报警消息归档

连通站

8.1 连通站的基础

简介

借助连通站，您可使用服务器数据包从中央计算机访问 WinCC 站，而无需使用 WinCC 软件。

可以通过两个不同的接口访问 WinCC 站：

- 连通站的 OPC 接口
- 连通性软件包的 OLE DB 接口

访问选项包括不同的功能。

连通站的 OPC 接口

连通站提供了多种接口，通过这些接口可以用 OPC 客户机访问下列内容：

- OPC DA 服务器：变量，例如过程值
- OPC HDA 服务器：归档过程值
- OPC A&E 服务器：消息

授权

要在未安装 WinCC 的计算机上使用连通站的 OPC 接口，需要“WinCC 连通站”许可证。

如果只使用 WinCC 安装的 OPC 接口，则只需要“连通性软件包”许可证。

下表显示了各项规则：

	连通站上独立于 WinCC 的安装	使用 WinCC 安装 OPC
OPC DA	“WinCC 连通站”许可证	不需要许可证
OPC HDA	“WinCC 连通站”许可证	“WinCC 连通性软件包”许可证
OPC A&E	“WinCC 连通站”许可证	“WinCC 连通性软件包”许可证

连通性软件包的 OLE DB 接口

运行连通站所需的“连通性软件包”服务器或“连通性软件包”客户机提供了附加的 OLE DB 接口。

8.1 连通站的基础

可以通过此类 OLE DB 接口访问消息和变量。

对归档数据的透明访问

连通站支持对 WinCC 站的归档数据库进行透明访问。

安装和组态

要安装连通站，需要具备下列条件：

- 具有连通性软件包服务器或连通性软件包客户机的 PC

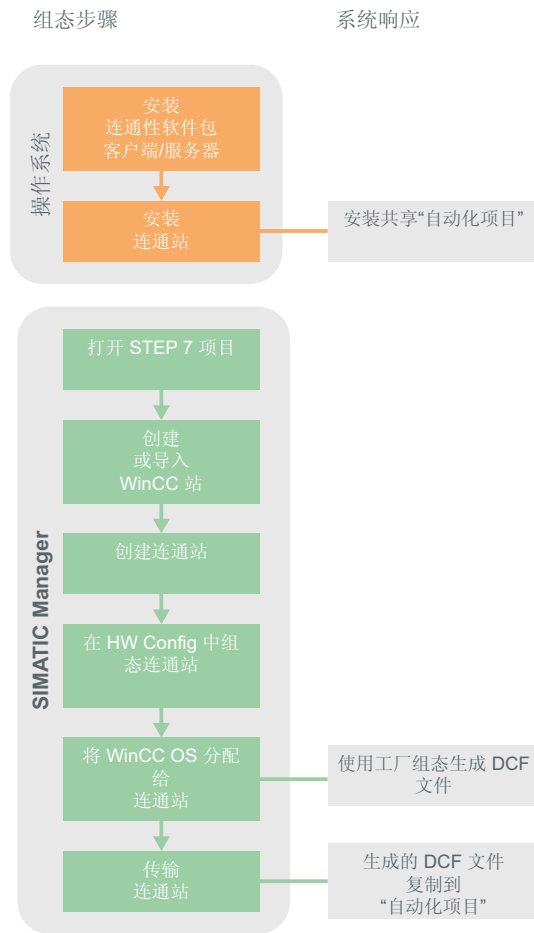
为了将某计算机组态为连通站，请在该计算机上运行连通性软件包客户机安装程序。

从 DVD 安装光盘的“安装软件”(Installing Software) 菜单中，选择“连通站”(Connectivity Station) 条目。

在安装后，可以使用下列程序设置对 WinCC 站进行访问：

- STEP 7 的 SIMATIC Manager

下图概述了连通站的组态步骤：



使用多个网络适配器时的操作设置

请确保您已在 SIMATIC Shell 的设置中选择了正确的网卡。

单击 SIMATIC Shell 图标，然后在快捷菜单中选择“SIMATIC Shell 设置”(Settings of SIMATIC Shell)。

您可能需要在“选择终端总线”(Selection of terminal bus) 下选择其它网卡。

参见

OLE DB 的基础 (页 33)

对归档数据的透明访问 (页 27)

OPC 的功能 (页 104)

使用连通站的 OPC 接口 (页 240)

使用连通站的 OLE DB 接口 (页 241)

8.2 连通站功能

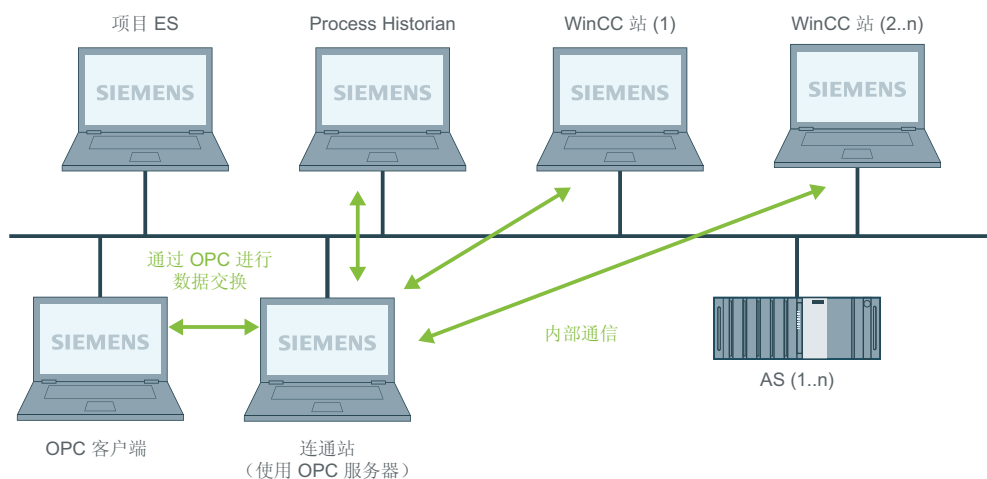
8.2.1 使用连通站的 OPC 接口

原理

连通站包括 WinCC OPC 服务器，通过该服务器借助于 OPC 客户端可以访问含服务器数据包的 WinCC 站。

可以在连通站或单独的计算机上以本地方式使用 OPC 客户端。

OPC 客户端需要连通站的 DCOM 访问权限。



ES = 使用 STEP 7 或 NCM PC Manager 的工程站
AS = 自动化系统

OPC 服务器的服务器名称

通过 OPC 在 OPC 客户端和连通站之间进行数据交换。

下列 WinCC OPC 服务器被集成到连通站中：

WinCC OPC 服务器	服务器名称
OPC DA	WinCCConnectivity.OPCDAServer
OPC HDA	WinCCConnectivity.OPCHDAServer.1
OPC A&E	WinCCConnectivity.OPCAEServer.1

支持的 OPC 标准

连通站支持下列 OPC 标准：

- OPC DA 2.05a Specification
- OPC DA 3.00 Specification
- OPC XML DA 1.01 Specification
- OPC AE 1.10 Specification
- OPC HDA 1.20 Specification

对归档数据的透明访问

借助 OPC，可对 WinCC 站的所有归档数据库进行透明访问。

参见

连通站的基础 (页 237)

在 WinCC 中使用 OPC (页 105)

使用 OPC 访问 WinCC 归档、变量和消息 (页 25)

对归档数据的透明访问 (页 27)

8.2.2 使用连通站的 OLE DB 接口

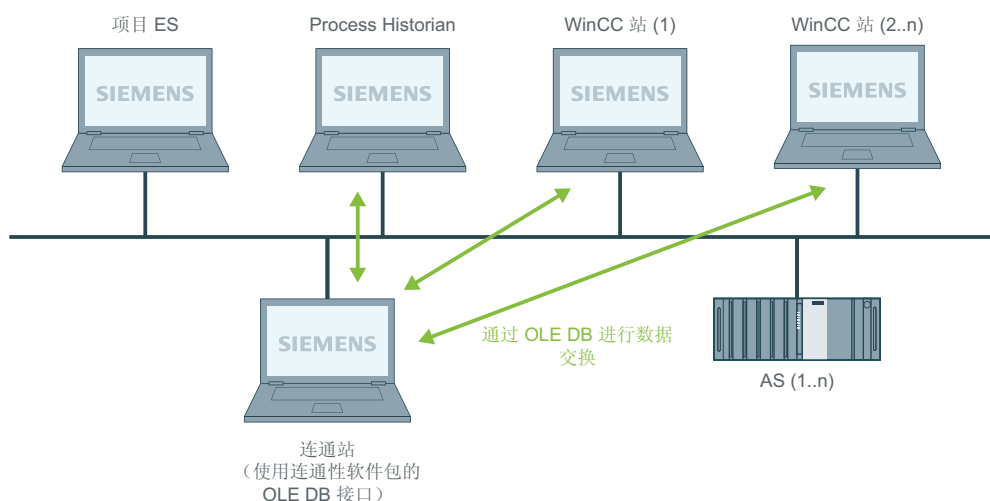
原理

如果安装了连通性软件包服务器或连通性软件包客户机，则只能使用连通站。

连通性软件包服务器/客户机的 WinCC OLE DB 提供者提供了 OLE DB 接口。

可以通过此类 OLE DB 接口访问消息和变量。

8.2 连通站功能



ES = 使用 STEP 7 或 NCM PC Manager 的工程站
AS = 自动化系统

使用 OLE DB 访问数据库

要使用 WinCC OLE DB 访问数据库，可能需要编写您自己的应用程序。

为与 WinCC OLE DB 提供程序进行通信，可通过诸如 Visual Basic、VBScript 或 VBA 创建的应用程序来使用 ADO DB。

说明

包含自有项目的客户端上的连通站

如果将客户端上的连通站与自身项目配合使用，并使用 OLE-DB 接口，请遵守下列事项：
您只能访问在连通站的组态阶段输入的 WinCC 站。
连通站不使用客户端上可用的服务器数据包。

对归档数据的透明访问

使用 OLE DB，只能以透明方式访问过程值归档。

如果通过 WinCC OLE DB 提供程序进行透明访问，在一个项目中仅可使用一个中央归档服务器，例如“Process Historian”。

如果要在两个项目中使用一个连通站，而且这两个项目都包含归档服务器，则在更改先前的项目前，需要删除该项目的 DCF 文件。

可在连通站的以下路径中找到 DCF 文件：

- 安装目录\Siemens\AutomationProjects
 \

说明

使用项目名称作为“目录”

对于透明访问，为“Catalog”输入 WinCC 项目名称；例如：“Catalog=WinCC_Project_Name”。

参见

连通站的基础 (页 237)

对归档数据的透明访问 (页 27)

消息和过程值的分析函数 (页 65)

8.3 在 S7 项目中组态连通站

简介

为了设置连通站，将新的“SIMATIC PC 站”添加到系统组态。

通过组态“SPOSA 应用程序”，此“SIMATIC PC 站”接收连通站的属性。

根据项目类型，在“SPOSA 应用程序”中创建下列对象之一。

- 对于“STEP 7”项目类型，这是“连接站_(n)”对象。
- 对于“PCS 7”项目类型，这是“Open_PCS7_Station_(n)”对象。

在下面说明中，使用了“STEP 7”项目类型和相应的对象名称。

要求

- 连通站的所需软件已安装在连通站 PC 上。
- 系统组态中包含 WinCC 站和服务器数据包。

操作步骤

连通站的组态过程由下列步骤组成：

1. 创建和组态 SIMATIC PC 站
2. 分配 WinCC 站
3. 传送连通站

创建和组态 SIMATIC PC 站

1. 从 SIMATIC Manager 的浏览窗口中选择项目目录。
2. 从快捷菜单中选择“插入新对象 > SIMATIC PC 站”(Insert New Object > SIMATIC PC STATION) 菜单命令。
插入了一个新“SIMATIC PC 站”(SIMATIC PC Station) 对象，并且在浏览窗口中保持选定。
3. 从 SIMATIC PC 站的快捷菜单中选择“打开对象”(Open object) 菜单命令。
“HW Config”编辑器将打开新 PC 站的视图，它是一个空机架。
4. 在“SIMATIC PC 站 > HMI”(SIMATIC PC Station > HMI) 目录的模块目录中选择“SPOSA 应用程序”(SPOSA Application) 条目。
将所选模块拖动至机架的一个空槽。
5. 选择菜单命令“站 > 保存和转换”(Station > Save and translate)。
6. 选择菜单命令“站 > 退出”(Station > Exit) 关闭“HW Config”。

分配 WinCC 站

1. 使用“SPOSA 应用程序”，浏览到新 PC 站目录中的“连通站”(Connectivity Station) 图标。
2. 从“连通站”(Connectivity Station) 快捷菜单中选择“分配 OS 服务器...(Assign OS Server...)”菜单命令。
此时将打开“为 Open_TIA_Station 分配 OS 服务器”(Assignment of OS Server for Open_TIA_Station) 对话框。
3. 在表格的“OS 信息”(OS Information) 列中选择连通站要访问其数据的 WinCC 站。
单击“确定”(OK) 关闭对话框。
4. 在浏览窗口中选择“连通站”(Connectivity Station) 图标。
5. 从快捷菜单中选择“对象属性”(Object properties) 菜单命令。
将显示“属性 - TIA 应用程序”(Properties - TIA application) 对话框：此时将打开“连通站”(Connectivity Station)。
6. 转到“目标系统”(Destination System) 选项卡，并在“路径”(Path) 字段中输入连通站 PC 的路径。
或者，用户可以通过“浏览...(Browse...)”按钮打开选择对话框，然后搜索网络中的计算机。
 - 输入 PC 名称后，单击“应用”(Apply) 按钮。
 - “自动化项目”(Automation Projects) 目录将附加到计算机名称后。
 - 单击“确定”(OK) 关闭对话框。

传送连通站

1. 检查在浏览窗口的新的 PC 站的目录中是否单击了“连通站”(Connectivity Station) 图标。
2. 从快捷菜单中选择“目标系统 > 加载”(Target System > Load) 菜单命令。
3. 退出此过程后确认消息。

结果

在 S7 项目的硬件组态中，针对连通站补充并组态了计算机。

所需的项目数据已经传送到此计算机。

8.4 用连通站访问 WinCC 数据

引言

连通站允许通过 OPC 客户机访问不同 WinCC 站的数据。为此，OPC 客户机必须真正地连接到连通站。

通过 OPC 在 OPC 客户机和连通站之间进行数据交换。下列 WinCC OPC 服务器被集成到连通站中。

OPC 服务器	服务器名称
OPC DA	WinCCConnectivity.OPCDAServer
OPC HDA	WinCCConnectivity.OPCHDAServer.1
OPC A&E	WinCCConnectivity.OPCAEServer.1

要求

- 连通站所需的软件已安装在连通站 PC 上。
- 连通站已在 S7 项目中组态，并且项目数据已传送到连通站的计算机。
- 有符合 OPC 规范的 OPC 客户机供使用。

通常步骤

下面的说明描述了基本步骤。

1. 在 PC 上启动 OPC 客户机。
2. 根据所需的数据选择 OPC 服务器。
与上表的服务器名称相同的名称适用于 WinCC OPC 服务器。
3. 定义应传送哪些数据，如变量。

结果

OPC 客户机连接到连通站并接收 WinCC 数据。

参见

使用 OLE DB 提供者访问的实例 (页 207)

使用 OPC 访问 WinCC 归档、变量和消息 (页 25)

建立到归档数据库的连接 (页 39)

OPC - 开放式互连 (页 104)

8.4 用连通站访问 WinCC 数据

OPC UA 的连通站

9.1 有关 OPC UA 连通站的基本信息

简介

借助连通站，您可使用服务器数据包从中央计算机访问 WinCC 站，而无需使用 WinCC 软件。可通过连通站的 OPC UA 接口访问 WinCC 站。

连通站的 OPC 接口

连通站提供 OPC UA 接口，用户可通过该接口使用 OPC 客户端访问下列内容：

- 过程值
- 变量归档的值
- WinCC 消息

授权

要在未安装 WinCC 的计算机上使用连通站的 OPC 接口，需要“WinCC 连通站”许可证。

如果只使用 WinCC 安装的 OPC 接口，则只需要“连通性软件包”许可证。

下表显示了各项规则：

	连通站上独立于 WinCC 的安装	使用 WinCC 安装 OPC
OPC UA	“WinCC 连通站”许可证	“WinCC 连通性软件包”许可证

对归档数据的透明访问

连通站支持对 WinCC 站的归档数据库进行透明访问。

安装和组态

要安装连通站，需要具备下列条件：

- 具有连通性软件包服务器或连通性软件包客户端的 PC

为了将某计算机组态为连通站，请在该计算机上运行连通性软件包客户端安装程序。

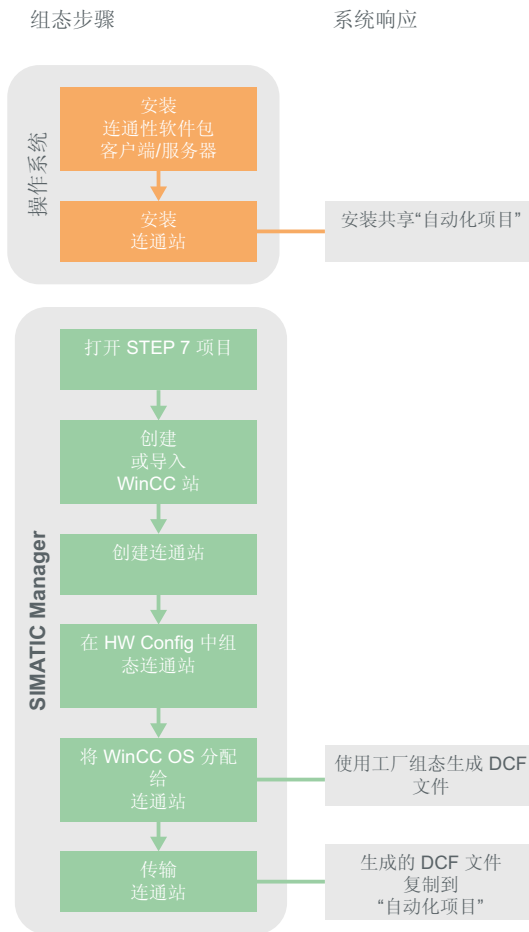
9.1 有关 OPC UA 连通站的基本信息

从 DVD 安装光盘的“安装软件”(Installing Software) 菜单中，选择“连通站”(Connectivity Station) 条目。

在安装后，可以使用下列程序设置对 WinCC 站进行访问：

- STEP 7 的 SIMATIC Manager

下图概述了连通站的组态步骤：



使用多个网络适配器时的操作设置

请确保您已在 SIMATIC Shell 的设置中选择了正确的网卡。

单击 SIMATIC Shell 图标，然后在快捷菜单中选择“SIMATIC Shell 设置”(Settings of SIMATIC Shell)。

您可能需要在“选择终端总线”(Selection of terminal bus) 下选择其它网卡。

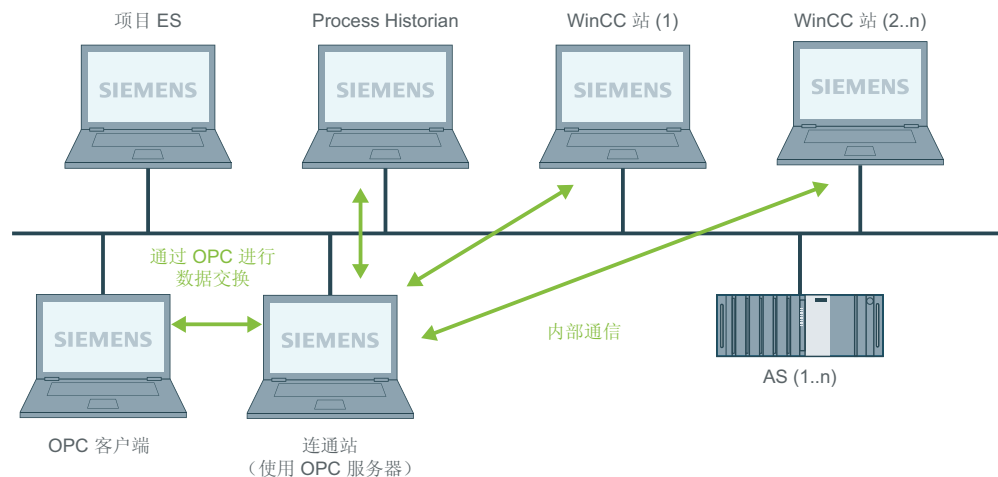
9.2 使用连通站的 OPC UA 接口

原理

连通站包括 WinCC OPC 服务器，通过该服务器借助于 OPC 客户端可以访问含服务器数据包的 WinCC 站。

可以在连通站或单独的计算机上以本地方式使用 OPC 客户端。

通过 OPC UA 在 OPC 客户端和连通站之间进行数据交换。



ES = 使用 STEP 7 或 NCM PC Manager 的工程站
AS = 自动化系统

服务器名称

WinCC OPC 服务器“WinCCConnectivity.OPCUAServer”集成在连通站中。

支持的 OPC 标准

WinCC OPC UA 服务器支持 OPC UA 规范 1.03。

9.3 在 S7 项目中组态连通站 (OPC UA)

简介

为了设置连通站，将新的“SIMATIC PC 站”添加到系统组态。

通过组态“SPOSA 应用程序”，此“SIMATIC PC 站”接收连通站的属性。

根据项目类型，在“SPOSA 应用程序”中创建下列对象之一。

- 对于“STEP 7”项目类型，这是“连接站_(n)”对象。
- 对于“PCS 7”项目类型，这是“Open_PCS7_Station_(n)”对象。

在下面说明中，使用了“STEP 7”项目类型和相应的对象名称。

要求

- 连通站的所需软件已安装在连通站 PC 上。
- 系统组态中包含 WinCC 站和服务器数据包。

操作步骤

连通站的组态过程由下列步骤组成：

1. 创建和组态 SIMATIC PC 站
2. 分配 WinCC 站
3. 传送连通站

创建和组态 SIMATIC PC 站

1. 从 SIMATIC Manager 的浏览窗口中选择项目目录。
2. 从快捷菜单中选择“插入新对象 > SIMATIC PC 站”(Insert New Object > SIMATIC PC STATION) 菜单命令。
插入了一个新“SIMATIC PC 站”(SIMATIC PC Station) 对象，并且在浏览窗口中保持选定。
3. 从 SIMATIC PC 站的快捷菜单中选择“打开对象”(Open object) 菜单命令。
“HW Config”编辑器将打开新 PC 站的视图，它是一个空机架。
4. 在“SIMATIC PC 站 > HMI”(SIMATIC PC Station > HMI) 目录的模块目录中选择“SPOSA 应用程序”(SPOSA Application) 条目。
将所选模块拖动至机架的一个空槽。
5. 选择菜单命令“站 > 保存和转换”(Station > Save and translate)。
6. 选择菜单命令“站 > 退出”(Station > Exit) 关闭“HW Config”。

分配 WinCC 站

1. 使用“SPOSA 应用程序”，浏览到新 PC 站目录中的“连通站”(Connectivity Station) 图标。
2. 从“连通站”(Connectivity Station) 快捷菜单中选择“分配 OS 服务器...(Assign OS Server...)”菜单命令。
此时将打开“为 Open_TIA_Station 分配 OS 服务器”(Assignment of OS Server for Open_TIA_Station) 对话框。
3. 在表格的“OS 信息”(OS Information) 列中选择连通站要访问其数据的 WinCC 站。
单击“确定”(OK) 关闭对话框。
4. 在浏览窗口中选择“连通站”(Connectivity Station) 图标。
5. 从快捷菜单中选择“对象属性”(Object properties) 菜单命令。
将显示“属性 - TIA 应用程序”(Properties - TIA application) 对话框：此时将打开“连通站”(Connectivity Station)。
6. 转到“目标系统”(Destination System) 选项卡，并在“路径”(Path) 字段中输入连通站 PC 的路径。
或者，用户可以通过“浏览...(Browse...)”按钮打开选择对话框，然后搜索网络中的计算机。
 - 输入 PC 名称后，单击“应用”(Apply) 按钮。
 - “自动化项目”(Automation Projects) 目录将附加到计算机名称后。
 - 单击“确定”(OK) 关闭对话框。

传送连通站

1. 检查在浏览窗口的新的 PC 站的目录中是否单击了“连通站”(Connectivity Station) 图标。
2. 从快捷菜单中选择“目标系统 > 加载”(Target System > Load) 菜单命令。
3. 退出此过程后确认消息。

结果

在 S7 项目的硬件组态中，针对连通站补充并组态了计算机。

所需的项目数据已经传送到此计算机。

9.4 通过连通站访问 WinCC 数据 (OPC UA)

简介

连通站允许通过 OPC 客户端访问不同 WinCC 站的数据。为此，OPC 客户端必须真正地连接到连通站。

通过 OPC UA 在 OPC 客户端和连通站之间进行数据交换。WinCC OPC 服务器“WinCCConnectivity.OPCUAServer”集成在连通站中。

要求

- 连通站所需的软件已安装在连通站 PC 上。
- 连通站已在 S7 项目中组态，并且项目数据已传送到连通站的计算机。
- 有符合 OPC 规范的 OPC 客户端供使用。

通常步骤

下面的说明描述了基本步骤。

1. 在 PC 上启动 OPC 客户端。
2. 根据所需的数据选择 OPC 服务器。
上表中的名称适用于 WinCC OPC 服务器。
3. 定义应传送哪些数据，如变量。

结果

OPC 客户端连接到连通站并接收 WinCC 数据。

索引

A

A&E 服务器, 150, 151, 161, 163, 164
 层级访问, 163
 跟踪事件, 150, 161
 简单事件, 161
 条件事件, 161
 映射 WinCC 消息类别和消息类型, 164
 映射 WinCC 消息系统, 151, 164
 与条件相关的事件, 150
AlarmHitView, 65, 68
ALARMVIEW, 53

C

CommandText, 43
ConnectionString, 39
cp_TagStatistic, 65

D

DataConnector, 9

I

I/O 域组态, 117
 OPC DA 的实例, 117

M

MS OLE DB, 33

O

OLE DB
 透明访问, 242
OLE DB 接口
 连通性软件包, 241
OPC, 9
 HDA 服务器浏览器, 143
 OPC A&E 服务器的功能, 161
 OPC 条目管理器, 74
 OPC 条目管理器功能概述, 74
 OPC 通讯受到干扰时的错误处理, 91
 OPC1, 86, 101

OPCServer.WinCC-(DPC_4001), 82
ProgID, 115
WinCC OPC A&E 服务器的功能, 150
WinCC OPC DA 服务端的功能, 113
WinCC OPC DA 客户端的功能, 79
WinCC OPC HDA 服务器的功能, 132
WinCC OPC UA 服务器的工作原理, 176
WinCC OPC XML 服务器的功能, 108
WinCC OPC XML 客户端的功能, 95
WinCC 中的 OPC, 73, 105
WinCC 作为 OPC DA 客户端, 73
WinCC 作为 OPC DA 客户机, 91
WinCC 作为 OPC XML 客户端, 73
 变量选择, 74
 查询 ProgID, 74
 创建连接, 74
 地址属性, 86, 101
 访问变量, 80
 跟踪, 205
 规范, 105
 过滤标准, 82
 兼容性, 105
 结构属性, 88
 实例, 121, 123, 126, 127, 128, 131
 实例: , 116, 117, 118, 142
 使用 HDA 服务器浏览器组态对 WinCC 归档变量的访问, 144
 使用多个 OPC DA 服务端, 114
 示例, 120, 122
 所支持的 WinCC 数据类型, 78
 所支持的 WinCC 数据类型概述, 78
 添加变量, 74, 82
 通信概念, 105
 透明访问, 241
 新建连接, 82
 新用户, 107
 用 OPC 条目管理器组态对 WinCC 变量的访问, 82
 在 OPC 计算机上设置用户帐户, 107
 在 WinCC OPC DA 客户端上使用结构, 88
 在 WinCC OPC DA 客户机上使用结构, 89
 在 WinCC OPC DA 客户机上组态 OPC 通道, 81
 在不使用 OPC 条目管理器的情况下组态对 WinCC 变量的访问, 86, 101
 组态对 WinCC 消息系统的访问, 159
 组态结构, 88
 组态结构变量, 88
OPC A 和原始数据, 158

- OPC A&E 服务器, 108, 150, 153, 158, 159, 161, 163, 164
 - 质量代码, 158, 171
 - 组态对 WinCC 消息系统的访问, 159
 - OPC A&E 服务器的功能
 - 跟踪事件, 161
 - 简单事件, 161
 - 条件事件, 161
 - OPC A&E 服务器上的消息类别, 164
 - OPC A&E 服务器上的消息类型, 164
 - OPC A&E 客户端
 - OPC A&E 的示例, 158
 - OPC DA 服务端, 113
 - WinCC OPC DA 服务端的功能, 113
 - 使用多个 OPC DA 服务端, 114
 - OPC DA 客户端, 79
 - WinCC OPC DA 客户端的功能, 79
 - OPC HDA 服务器, 132
 - WinCC OPC HDA 服务器的时间格式, 137
 - 范围值, 133
 - 工作原理, 132
 - 配件, 135
 - 属性, 134
 - 数据结构, 133, 134
 - 条目标识号, 133
 - 条目句柄, 133
 - 写访问, 139
 - 原始数据, 132
 - 支持的函数, 136
 - 质量代码, 139, 158
 - OPC UA
 - Conformance units, 186
 - WinCC 变量的显示, 188
 - 记录变量的显示, 189
 - 配置文件, 186
 - OPC UA 服务器
 - URL, 177
 - 安全策略, 177
 - 工作原理, 176
 - 实例证书, 178
 - 受信任的客户端证书, 179
 - 通信配置文件, 176
 - 应用程序证书, 178
 - 证书, 178
 - 支持的规范, 176
 - 组态文件, 198
 - OPC UA 历史访问, 190
 - OPC UA 数据访问, 190
 - OPC XML, 111
 - IIS 中的安全性设置, 111
 - 测试安装, 111
 - 访问变量, 97
 - OPC XML 服务器, 108
 - 安装, 109
 - OPC XML 客户端, 95
 - WinCC OPC XML 客户端的功能, 95
 - OPC 标准
 - 支持 ~ 连通站支持, 241
 - OPC 服务器
 - 服务器名称, 240
 - OPC 接口
 - 连通站, 240, 251
 - OPC 条目管理器, 74, 82, 115
 - OPC 条目管理器功能概述, 74
 - OPC 通道
 - 在 WinCC OPC DA 客户机上组态 OPC 通道, 81
 - OPC 通讯, 91
 - OPC DA 通讯受到干扰时的错误处理, 91
 - OPC1, 86, 101
 - OPCScout new project1
 - OPC DA 的实例, 123
 - OPCServer.WinCC-(DPC_4001), 82
- P**
- ProgID, 74
 - 查询, 115
 - 查询 ProgID, 74
- S**
- SQL Server Import/Export, 23
 - SQL Server 导入/导出向导, 61
 - SQL 查询
 - 至 WinCC 数据库, 61
 - SQL 数据库, 71
 - SQL 数据库的访问保护, 71
- T**
- TIMESTEP, 45
- U**
- URL
 - OPC UA 服务器, 177
- V**
- Visual Basic, 208
 - 访问归档数据, 208

Visual Basic .NET, 223, 231
访问归档数据, 223, 231

W

WinCC

WinCC 中的 OPC, 73, 91, 105
作为 OPC DA 客户端, 73
作为 OPC DA 客户机, 91
作为 OPC XML 客户端, 73

WinCC - Microsoft Excel 的连接, 127
OPC DA 的实例, 127

WinCC - OPC HDA 客户机连接, 142
OPC HDA 的实例, 142

WinCC - SIMATIC NET FMS OPC 服务器的连接, 120
OPC DA 的实例, 120

WinCC - SIMATIC NET S7 OPC 服务端的连接, 122
OPC DA 的实例, 122

WinCC - WinCC 连接, 116
OPC DA 的实例, 116

WinCC DataConnector, 9

WinCC Explorer-OPC_Client.MPC, 115

WinCC OLE DB, 33

Microsoft, 33
WinCC, 33
基础, 33

WinCC OLE DB 提供程序, 9, 31

SQL Server, 9
访问归档数据, 31
链接的服务器, 9

WinCC OPC A&E 服务器
层级访问, 163

WinCC OPC A&E 服务器的功能
简单事件, 150

WinCC OPC UA 服务器, 176

Discovery 服务器, 177
组态, 203

WinCC OPC 服务器
服务器名称, 240

WinCC 归档连接器, 9, 35

WinCC 消息系统

OPC A&E 服务器上的 WinCC 消息等级, 153
属性, 154, 166
映射 WinCC 消息等级和消息类型, 153
在 OPC A&OPC-A&WinCC 消息系统上, 151
组态对 WinCC 消息系统的访问, 159

WinCC 站

访问 ~ 通过连通站, 246, 254

报

报表服务

要求, 63

变

变量

HDA 服务器浏览器, 143
OPC DA 的实例, 123, 126, 131
OPC HDA 的实例, 144, 145
OPC 条目管理器, 82
R, 45
变量选择, 74
添加变量, 74
组态 HDA 服务器浏览器, 144
组态 OPC 条目管理器, 82, 86
组态 PC 条目管理器, 101

标

标准 SQL 查询

至 WinCC 数据库, 61

查

查询, 45, 53, 58
过程值归档, 45
消息归档, 53
用户归档, 58

带

带归档数据的报表

在网络上可用, 63

地

地址属性, 86, 101

电

电子服务器

WinCC OPC A&OPC 的功能, 108

定

- 定义新变量, 123
 - OPC DA 的实例, 123

范

- 范围值, 133

分

- 分析的集合函数, 65
- 分析函数, 65
 - 报警日志, 65
 - 报警消息归档, 68
 - 过程值归档, 65
 - 具有消息归档分析的记录集, 68

服

- 服务器名称
 - WinCC OPC 服务器, 240

归

- 归档连接器, 9
- 归档数据, 31, 43, 53, 58
 - 查询, 43, 53, 58
 - 使用 DataConnector Wizard 组态访问, 210
 - 使用 Visual Basic .NET 组态访问, 223, 231
 - 使用 Visual Basic 组态访问, 208
 - 通过 WinCC OLE DB Provider 进行访问, 23
 - 通过 WinCC OLE DB 提供程序进行访问, 31
 - 作为网络上的报表, 63
- 归档数据库, 39
 - 连接, 39

过

- 过程值归档, 42, 43, 45, 215
 - 查询, 45
 - 查询归档数据, 43
 - 记录集, 42
 - 使用 DataConnector Wizard 组态访问, 210
 - 使用 Visual Basic .NET 访问, 223
 - 使用 Visual Basic 访问, 208, 215
 - 显示数据, 42
 - 在 Visual Basic Runtime 中显示, 215

- 过滤标准, 82

缓

- 缓冲时间, 159

基

- 基础知识
 - 连通站, 237, 249

记

- 记录集, 42, 55

兼

- 兼容性, 105

结

- 结构
 - 使用 WinCC OPC 客户机, 89
 - 在 WinCC OPC DA 客户端上使用结构, 88
 - 组态结构, 88
- 结构变量, 88
 - 组态结构变量, 88
- 结构属性, 88

连

- 连接
 - 创建连接, 74
 - 连接 WinCC - OPC A&E 客户端, 158
 - OPC A&E 的示例, 158
 - 连接监控, 79
 - 连通性软件包, 9
 - MS OLE DB 提供程序, 9
 - OPC, 9
 - WinCC OLE DB 提供程序, 9
 - 归档连接器, 9, 35
 - 使用 OLE DB 接口, 241
 - 数据库访问, 35
- 连通站
 - 访问 WinCC 站, 246, 254
 - 基础知识, 237, 249
 - 使用 OPC 接口, 240, 251
 - 许可证, 237, 249
 - 正在安装, 238, 249

支持的 OPC 标准, 241
组态, 244, 252

链

链接的服务器, 9

确

确认策略, 168
确认方法, 156

实

实例: , 207

使

使用

连通性软件包的 OLE DB 接口, 241
连通站的 OPC 接口, 240, 251

使用案例, 15, 16, 18, 21, 22

访问本地 WinCC RT 数据库, 15
访问本地 WinCC 归档数据库, 18
访问本地 WinCC 用户归档, 21
通过 WinCC OLE DB Provider 进行访问, 23
远程访问 WinCC RT 数据库, 16
远程访问 WinCC 用户归档, 22

示

示例, 215, 228
读取过程值归档, 215
读取消息归档, 228

授

授权

连通站, 237, 249

数

数据库访问, 35
数据类型, 78
所支持的 WinCC 数据类型, 78
所支持的 WinCC 数据类型概述, 78

添

添加变量, 74, 82
OPC DA 的实例, 118

条

条目标识号, 133
条目句柄, 133

调

调试, 205

通

通过 WinCC OLE DB 进行访问, 61
通过连通站
访问 WinCC 站, 246, 254

透

透明访问
借助于 OPC, 241
使用 OLE DB, 242

项

项目
OPC DA 的实例, 118

消

消息归档, 43, 53, 228
查询归档数据, 43, 53, 228
记录集, 55
使用 DataConnector 组态访问, 210
使用 Visual Basic .NET 访问, 231
使用 Visual Basic 访问, 208, 228
显示数据, 55
消息归档数据
使用 Visual Basic .NET 访问, 231

新

新建连接, 82
新用户, 107

应

应用程序

- 使用 OPC 访问, 25
- 通过 WinCC OLE DB 提供程序进行访问, 29

应用方案, 19

- 远程访问 WinCC 归档数据库, 19

用

用户归档, 58, 60

- 查询归档数据, 58
- 显示数据, 60

用户帐户, 107

- 公布 OPC 计算机, 107

原

原始数据, 132

在

在可移动介质上换出 WinCC 归档, 35

正

正在安装

- 连通站, 238, 249

质

质量代码, 139, 158, 171

组

组态

- WinCC OPC UA 服务器, 203
- 安全设置, 204
- 端口号, 203
- 连通站, 244, 252
- 用户标识, 204
- 优化的 WinCC 归档写访问, 205

组态文件

- WinCC OPC UA 服务器的 URL, 198
- WinCC 记录写访问, 198
- 安全设置, 198
- 布局, 198
- 存储位置, 198

端口号, 198

用户标识, 198

证书, 198

最

最大大小, 159