

SIEMENS

WinCC

组态手册

第三册

订货号: 6AV6392-1CA05-0AH0
C79000-G8276-C164-01

发行: 1999 年 9 月

WinCC、SIMATIC、SINEC、STEP 是西门子注册商标。

本手册中所有其它的产品和系统名称是（注册的）其各自拥有者的商标，必须被相应地对待。

（若没有快速写入权限，不允许对本文件或其内容进行复制、传送或使用。
违犯者将要对损坏负责任。保留所有权利，包括由专利授权创建的权利，对实用新型或设计的注册。）

（我们已检查了本手册的内容，使其与硬件和软件所描述的相一致。由于不可能完全消除差错，我们也不能保证完全的一致性。然而，本手册中的数据是经常规检查的，在以后的版本中包括了必要的修正。欢迎给我们提出建议以便改进。）

目录

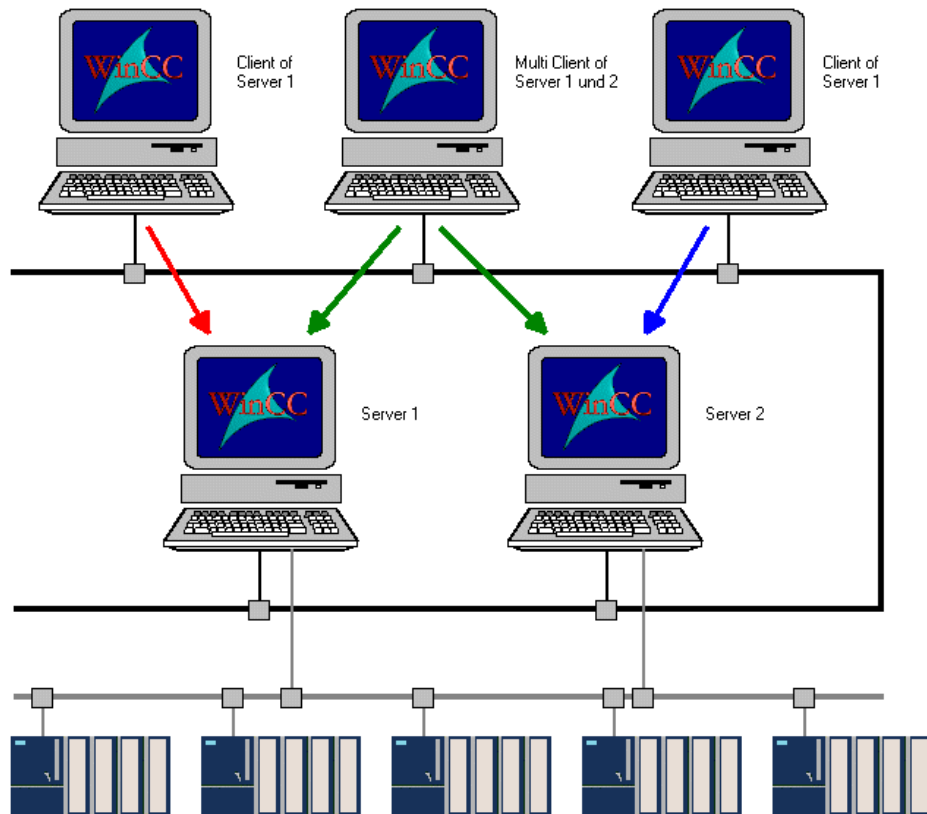
1	新功能描述	1-1
1.1	将多个客户机分配给多个服务器	1-2
1.1.1	多客户机项目的组态	1-3
1.1.2	多客户机项目的组态	1-4
1.1.2.1	画面组态	1-4
1.1.2.2	多客户机项目中变量记录 ActiveX 控件的组态	1-6
1.1.2.3	多客户机项目中报警记录 ActiveX 控件的组态	1-7
1.1.3	服务器数据(数据包)	1-8
1.1.3.1	WinCC 服务器项目中的数据包	1-8
1.1.3.2	WinCC 多客户机中的数据包	1-9
1.1.4	分布式服务器	1-13
2	多客户机系统	2-1
2.1	多客户机的应用	2-2
2.2	服务器数据(数据包)	2-3
2.2.1	WinCC 服务器项目中的数据包	2-4
2.2.2	WinCC 多客户机项目中的数据包	2-6
2.3	Project_MultiClient_Server 项目的创建	2-8
2.4	Project_MultiClient_Client 项目的创建	2-23
2.5	WinCC 项目的描述	2-35
2.5.1	服务器项目	2-36
2.5.2	多客户机项目	2-37
3	分布式服务器	3-1
3.1	常规信息	3-2
3.2	Project_DisServer_Server 项目的创建	3-3
3.3	Project_DisServer_Client 项目的创建	3-18
3.4	WinCC 项目的描述	3-31
3.4.1	服务器项目	3-32
3.4.2	客户机项目	3-33
4	冗余	4-1
4.1	常规信息	4-2
4.1.1	冗余的操作	4-3
4.1.2	冗余用户归档	4-6
4.2	Project_Redundancy_Server 项目的创建	4-9
4.3	WinCC 项目的描述	4-34
4.3.1	服务器项目	4-35
5	用户归档(PROJECT_USERARCHIVE)	5-1

5.1	用户归档	5-2
5.1.1	用户归档的创建(ex_3_chapter_01.PDL)	5-3
5.1.2	数据输入(ex_3_chapter_01.PDL)	5-14
5.1.3	表格视图的组态(ex_3_chapter_01.PDL)	5-19
5.1.4	表单视图的组态(ex_3_chapter_011.PDL)	5-27
5.1.5	使用控制变量进行工作(ex_3_chapter_012.PDL)	5-33
5.1.6	通过 WinCC 原始数据变量进行通讯	5-39
5.1.7	通过 WinCC 原始数据变量与 SIMATIC S5 进行通讯(ex_3_chapter_01a.pdl)	5-44
5.1.7.1	通讯处理器 CP 1413 的启动	5-45
5.1.7.2	PLC 的启动	5-52
5.1.7.3	在 WinCC 中组态	5-62
5.1.8	通过 WinCC 原始数据变量与 SIMATIC S7 进行通讯(ex_3_chapter_01a.pdl)	5-76
5.1.8.1	通讯处理器 CP 5412 A2 的启动	5-77
5.1.8.2	PLC 的启动	5-84
5.1.8.3	在 WinCC 中组态	5-100

1 新功能描述

1.1 将多个客户机分配给多个服务器

SIMATIC WinCC 5.0 版引入了一种新的项目类型：多客户机项目。多客户机是指拥有单独的组态数据(例如本地画面、脚本和变量)的一种客户机。SIMATIC WinCC V5.0 的体系结构允许将多客户机连接至网络上所有可用的服务器上。通过这些连接，可实现功能分配(单独的过程数据、消息与归档服务器)，或者负载分配(多种过程数据、消息与归档服务器)。



一个服务器项目最多可被 16 个多客户机访问，也就是说访问一个服务器项目的客户机与多客户机的总量不能超过 16 个。关于这一点，与 WinCC V4.02 所要求的限制相同。一个多客户机最多可访问 6 个服务器项目。因此，一个多客户机所能显示的数据点在理论上限制为 $6 * 64k$ 对变量或 $6 * 50000$ 对消息。

1.1.1 多客户机项目的组态

多客户机项目只能组态其自己的数据，而不能组态服务器项目的数据。但是，它可以访问服务器上的数据(提供所谓的服务器的视图)。

在服务器数据(数据包)一章中，对数据包的结构进行了描述。在组态阶段，需要这些数据包来使多客户机项目能够利用一个或多个服务器的相关数据。

这些相关数据是：

- 图形系统：画面
- 数据管理器：变量名称/变量类型
- 报警系统：消息服务器是/否
- 归档系统：具有相应归档变量的归档
- 文本库：文本 ID
- 组显示：服务器是/否
- 测量点列表：服务器是/否
- 画面目录管理器：服务器是/否、ID/文本

为了在组态客户机项目期间连接这些组态数据，可扩充现有的组态对话框。

实例：

- 扩充变量对话框，以允许对各种不同服务器的变量进行浏览。
- 扩充用于连接画面窗口的画面的对话框，以便也显示服务器的画面。

通常，只有那些与客户机项目相关的编辑器才能在客户机项目中启动。

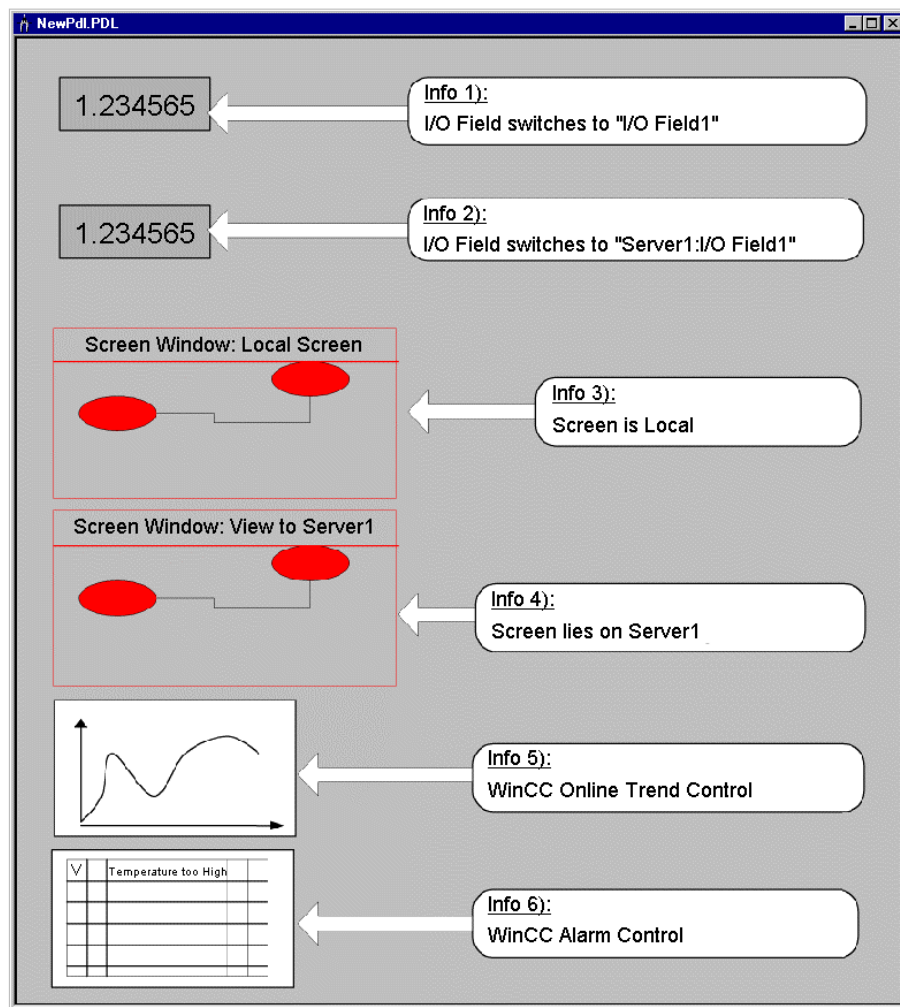
该信息可通过指定的编辑器输入到 mcp.ini 文件中，且可能以如下格式：

```
[Application_4]
Name=Tag Logging
SubDir=PDE
ExtDll=PDECSEXP.dll
MultiClient=1
```

1.1.2 多客户机项目的组态

1.1.2.1 画面组态

客户机项目拥有其自己的本地画面，但也可使用一个或多个服务器的画面。以下画面说明了多客户机的各种可能性。



- Info 1: I/O Field1 是客户机的一个内部变量。
- Info 2: Server1:I/O Field1 是 SERVER1 的一个过程变量。此处，通过名称服务可对 SERVER1::OTTO 变量进行寻址，也就是说服务器名称 SERVER1 不是物理计算机名称，而只是由名称服务分配的一个名称。
- 其重要性就是在一个画面中可同时对多个服务器进行寻址。
- Info 3: 在画面窗口中，对多客户机项目的本地画面进行显示。

- Info 4: 在画面窗口中, 对 Server1 的画面进行显示。Server1 上该画面窗口的连接通过画面窗口的服务器前缀属性来实现。
- Info 5: 变量记录 ACX (参见章节多客户机项目中变量记录 ActiveX 控件的组态)
- Info 6: 报警记录 ACX (参见章节多客户机项目中报警记录 ActiveX 控件的组态)

注意:

使用分层画面窗口技术(画面中的画面内的画面), 服务器前缀可一直遍历到下一个子位画面。如果使用了服务器的画面, 则文本列表对象中的变量连接与文本库访问也可引用该服务器。画面窗口的画面名称属性可按 Server1::Picture.pdl 的形式动态化。这样, 可自动完成画面名称或服务器前缀的分割。如果画面包含面板, 则服务器前缀可自动扩充变量名称属性。因此, 多客户机不能对面板的实现过程进行修改。

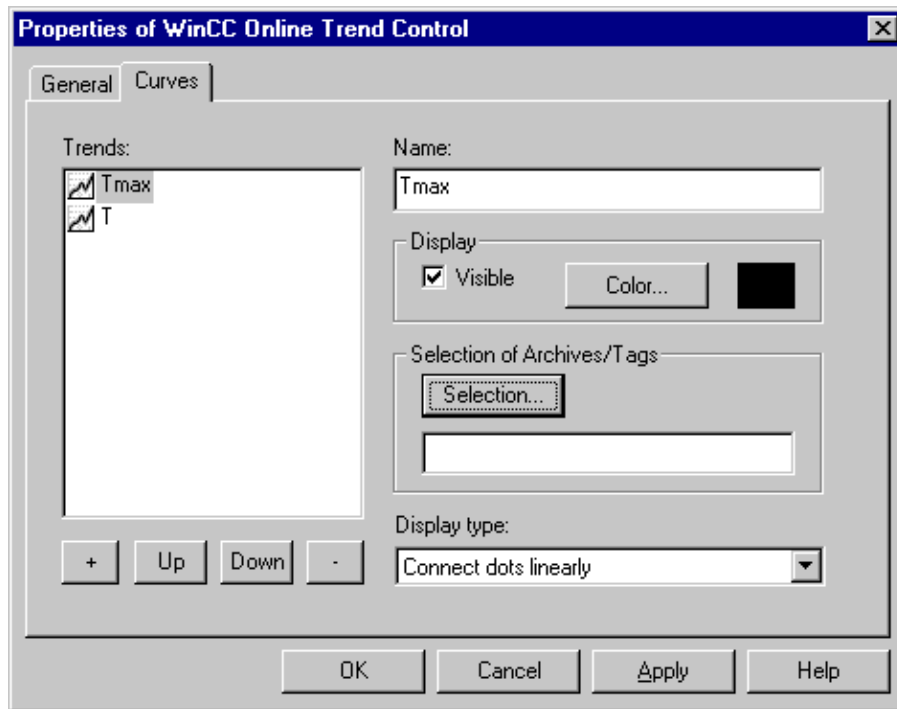
1.1.2.2 多客户机项目中变量记录 ActiveX 控件的组态

为了在多客户机项目中显示或编辑来自一个或多个变量记录服务器的数据，必须使用新的 *WinCC 在线趋势控件*和 *WinCC 在线表格控件*。

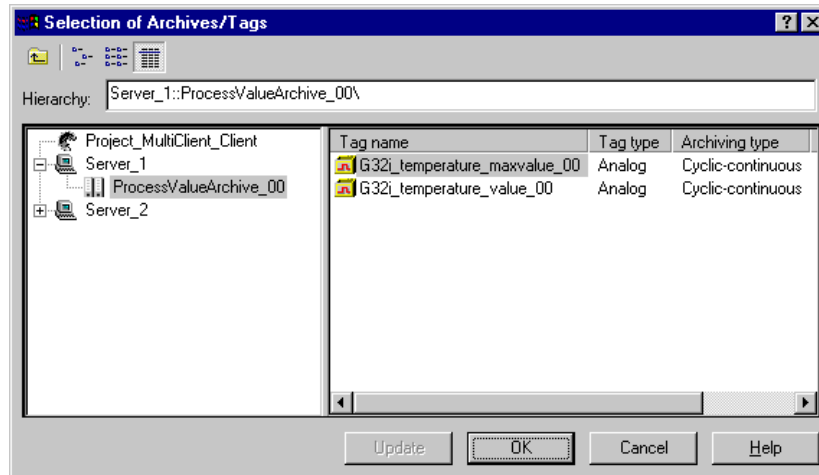
两者均能显示来自多个服务器或服务器项目的数据。

通过 *图形编辑器*，在多客户机项目中组态一个画面。在该画面中，放置一个 *WinCC 在线趋势控件*或一个 *WinCC 在线表格控件*。通过控件的组态对话框，可完成曲线(趋势控件)或列(表格控件)与服务器、归档以及归档变量或在线变量之间的连接。

在以下画面中，使用 *WinCC 在线趋势控件*的一个实例来显示组态。



在归档/变量选择对话框中，既可输入包含服务器/归档/归档变量的字符的文本，也可通过选择按钮将对话框打开，该选择按钮允许从由数据包所导入的服务器信息中选择一个服务器/归档/归档变量。

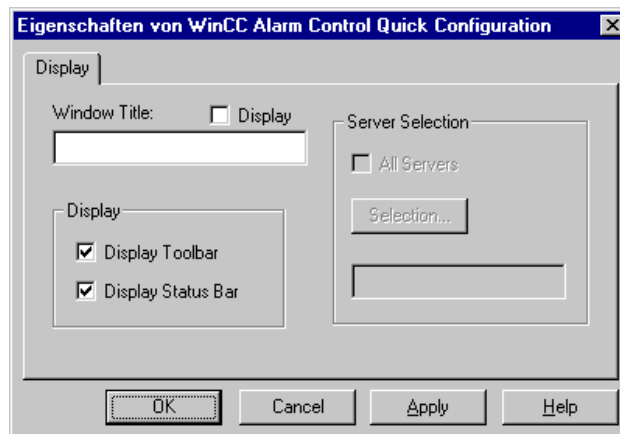


1.1.2.3 多客户机项目中报警记录 ActiveX 控件的组态

为了在多客户机项目中使用报警记录，必须使用新的 WinCC 报警控件。它提供了对来自多个服务器的消息进行显示的选项。

其组态过程如下：

通过图形编辑器，在多客户机项目中组态一个画面。在该画面中，放置一个 WinCC 报警控件。通过控件的属性对话框，可完成 WinCC 报警控件与一个或多个服务器的连接。



也有一个选项用于始终连接所有服务器(启动一个报警服务器)，它通过一个参数来进行设置。

1.1.3 服务器数据(数据包)

数据包用于为多客户机的组态者提供一个或多个 WinCC 服务器的数据引用，以及
在多客户机项目中使用这些数据引用。

此外，数据包也用于组态名称服务(NS)。

数据包包含许多对象名称，这些对象源于按对象类型(变量名称、归档名称、画面名称等)进行排序的服务器项目。名称的存储是服务器指定的，但其仍然使用统一的格式执行。这些文件从服务器项目导出。于是，可将数据包导入到多客户机项目中。用户负责对数据进行更新。

为了允许在离线时对 WinCC 服务器的数据进行访问，创建一个新的资源管理器 DLL，它可提供下列功能：

- 从服务器中导出与服务器相关的数据。
- 将与服务器相关的数据导入到多客户机项目中。
- 在选择对话框中选择可用数据。
- 组态名称服务。

1.1.3.1 WinCC 服务器项目中的数据包

在一个 WinCC 服务器项目中，有下列菜单条目可供使用：

- 生成服务器数据
- 删除服务器数据
- 服务器数据属性



生成服务器数据

如果选择了 *服务器数据* → *生成菜单*，则将在项目文件夹的数据包的子文件夹中创建一个文件名为 *Server_physical computer name.pck* 的新的复合文档。

实例：如果服务器的计算机名称是 SI10092D，则数据包将被命名为 *Server_SI10092D.pck*。

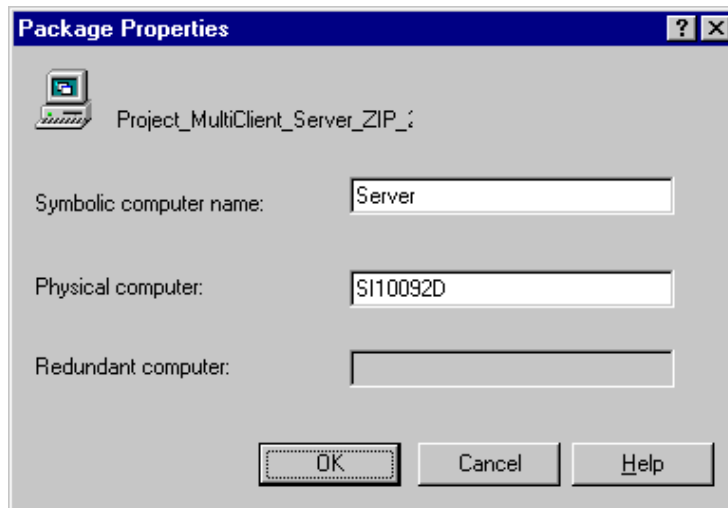
如果先前在该项目中已经导出了一个数据包，也就是说，文件 *Server_physical computer name.pck* 已存在，则该数据包将被覆盖。

删除服务器数据

如果将数据包删除，则它将从 *项目文件夹\数据包* 中删除。

服务器数据属性

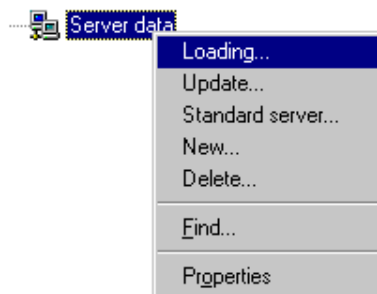
在 WinCC 服务器或 WinCC 客户机上的数据包属性中，预先组态了用于多客户机的名称服务。



1.1.3.2 WinCC 多客户机中的数据包

在 WinCC 多客户机上，有下列菜单条目可供使用：

- 导入数据包
- 导出数据包(未激活)
- 删除数据包
- 数据包属性



装载服务器数据

通过 *服务器数据* → *装载...* 和一个文件打开对话框可实现数据的导入，用户可从中选择一个数据包。这样，可将数据包复制给\项目文件夹\数据包下的项目文件夹。

随后，对用于该多客户机项目的名称服务进行组态。

如果此时还不存在具有该符号计算机名称的任何条目，则该数据包将作为新数据包进行记录，并在数据包条目下带下列信息进行显示：

- 具有 *符号计算机名称* 的服务器项目正在运行
- 数据包的名称(*Server_symbolic computer name.pck*)
- 创建日期

如果名称服务已知道数据包的符号计算机名称的条目，则在导入数据包时，可使用下列选项：取消导入、重新命名符号计算机名称或使用新的数据来覆盖名称服务中的现有符号计算机名称。

更新服务器数据

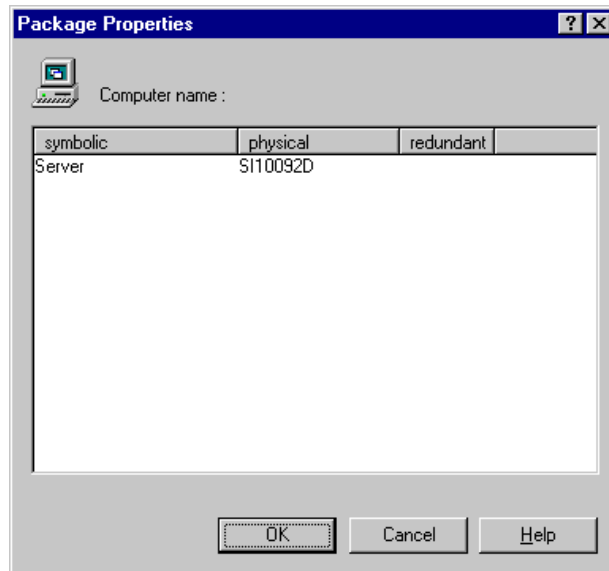
通过该菜单条目，可对所有导入的数据包进行更新。

删除数据包

通过该菜单条目，可删除所有导入的数据包。

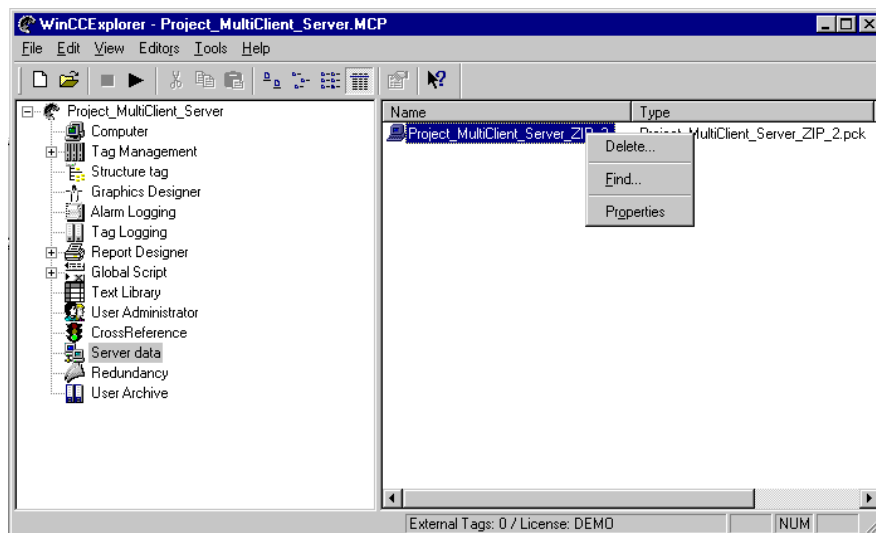
数据包属性

通过该菜单条目，可指定优先使用的服务器，若在冗余系统中，还指定其冗余伙伴。



删除一个单独的数据包

也可单独将数据包删除。



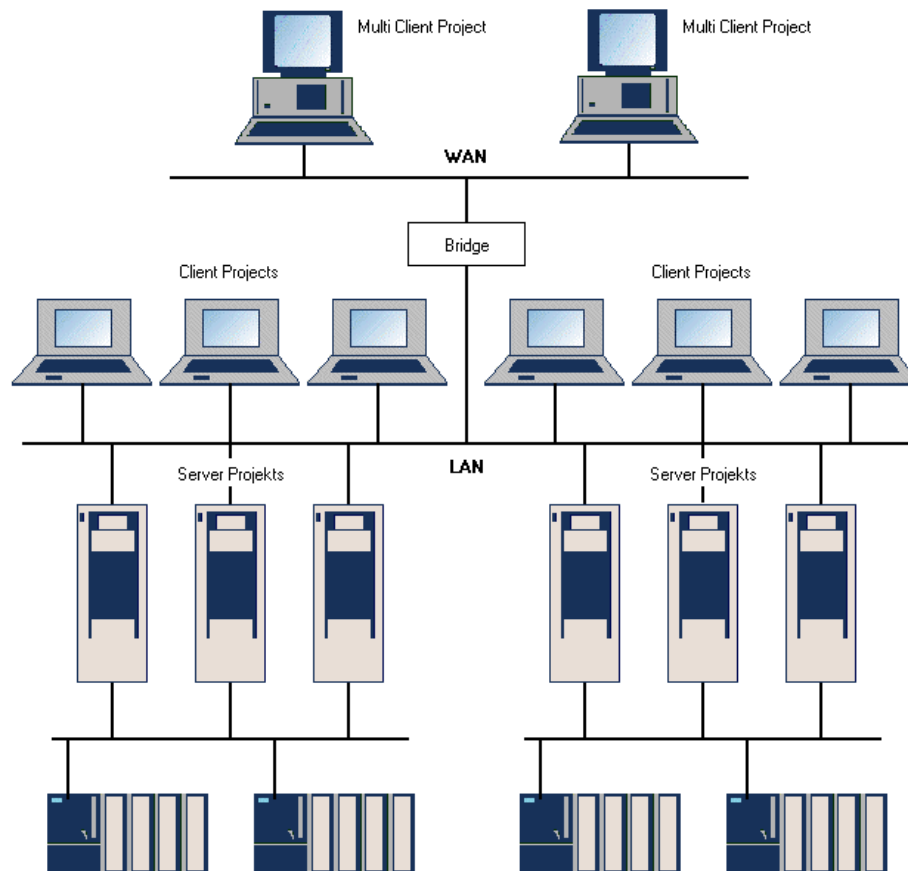
优先服务器

在项目环境中，多客户机不处理其自身的归档或消息服务器。如果在多客户机环境下调用了服务器的 RT API (MSRTCLI.DLL、PDERTCLI.DLL)，则将对指定为优先服务器的服务器进行寻址。在这种情况下，每个服务器类型均有其自身的优先服务器。优先服务器的组态是项目指定的，且只能应用于一个多客户机项目。通常，进行下列应用：如果没有使用任何前缀来指定服务器，则使用优先服务器。

1.1.4 分布式服务器

使用三层结构进行分布

Win CC 版本 5.0 的分布可能如下所示：



在底层结构中，过程计算机将通过服务器项目为其客户提供过程数据。在服务器项目中，可以任何形式对归档服务器、消息服务器和过程数据服务器进行组态。处于第二层结构的客户机对应于 WinCC 版本 4.0 中的客户机/服务器结构的客户机。这些客户机被固定分配给一个服务器项目，且只能访问该服务器项目的数据。在第三层结构中，多客户机激活多个服务器项目的视图。这包括对服务器项目中的画面和数据的访问。

2 多客户机系统

本章中创建的项目也可以直接从在线文档复制到硬盘驱动器上。缺省情况下，它们将被复制到 *C:\Communication_Manual* 文件夹中。可以选择将以下组件复制到硬盘驱动器上：



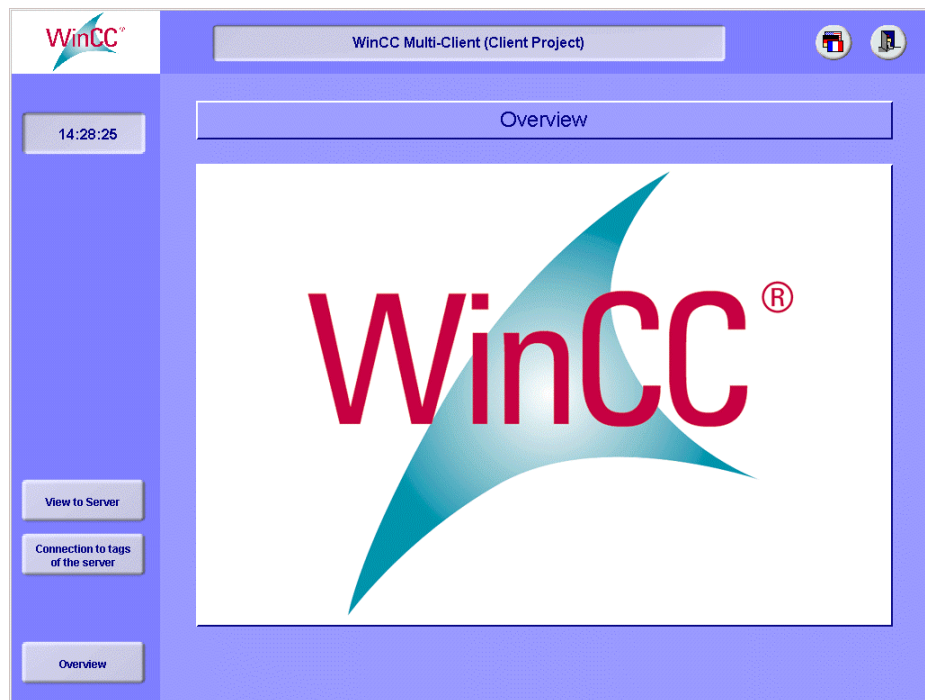
将要创建的 WinCC 服务器项目。



将要创建的 WinCC 客户机项目。

在本节中，将显示与多客户机相关的实例。

在 WinCC 项目 Project_MultiClient_Server 和 Project_MultiClient_Client 中组态了与该主题相关的实例。



2.1 多客户机的应用

多客户机是一个 WinCC 项目，在其中可对多台服务器的数据进行访问。多客户机有其自己的项目，它独立于服务器。在服务器计算机上进行服务器的组态，在多客户机计算机上进行多客户机的组态。

客户机以及多客户机可同时对一台服务器进行访问。将访问一台服务器的站、客户机和多客户机的最大数目限制为 16。

在运行系统中，多客户机最多可同时对六台服务器进行访问。例如，在一个多客户机的画面中，可看到六台不同服务器的数据。在切换画面后，多客户机可访问六台不同的服务器。此外，可对项目到多服务器的功能分布或技术分布进行组态。

服务器

服务器将完成与过程的连接、数据的存储以及过程数据的处理。服务器上的所有项目数据如画面、变量和归档均可供客户机使用。也可用一个冗余的服务器对来代替服务器。

多客户机

多客户机最多可访问六台服务器中的数据。从一台多客户机中可对过程进行控制和监视。

2.2 服务器数据(数据包)

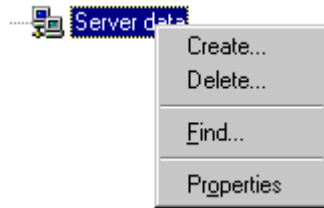
数据包用于向多客户机的组态者提供对一台或多台 WinCC 服务器的数据引用，并在多客户机项目中使用这些数据引用。此外，数据包也用于对名称服务(NS)进行组态。

数据包包含了服务器项目中的对象名称，按对象类型(变量名称、归档名称、画面名称等)进行排序。名称的存储是服务器指定的，但仍使用统一格式来执行。这些文件从服务器项目导出。于是，可将数据包导入到多客户机项目中。用户负责对数据进行更新。

2.2.1 WinCC 服务器项目中的数据包

在一个 WinCC 服务器项目中，有下列菜单条目可供使用：

- 生成服务器数据
- 删除服务器数据
- 服务器数据属性



生成服务器数据

如果选择了 *服务器数据* → *生成* 菜单，则将在项目文件夹的数据包子文件夹中创建一个文件名为 *Server_physical computer name.pck* 的新的复合文档。

实例：如果服务器的计算机名称是 SI10092D，则数据包将被命名为 *Server_SI10092D.pck*。

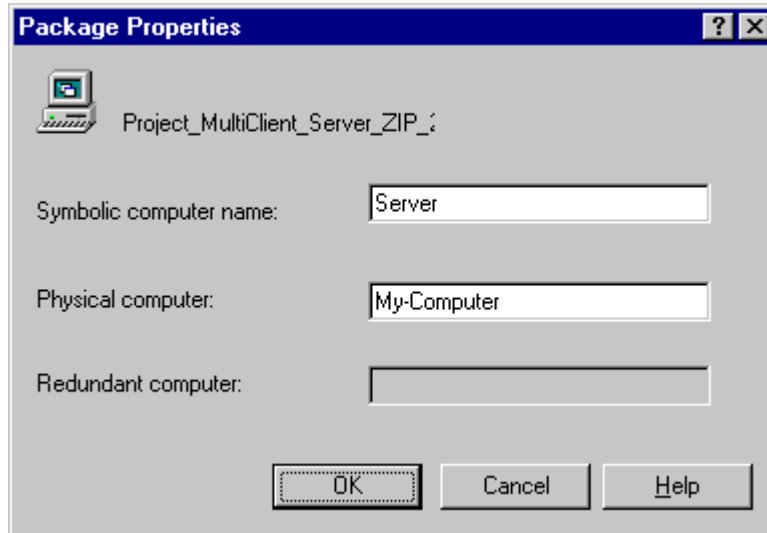
如果先前在该项目中已经导出了一个数据包，也就是说，文件 *Server_physical computer name.pck* 已经存在，则该数据包将被覆盖。

删除服务器数据

如果将数据包删除，则它将从\项目文件夹\数据包中删除。

服务器数据属性

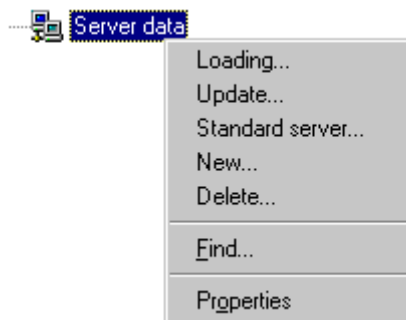
通过 WinCC 服务器上的数据包属性，预先组态了用于多客户机的名称服务。



2.2.2 WinCC 多客户机项目中的数据包

在一个 WinCC 多客户机项目中，有下列菜单条目可供使用：

- 装载服务器数据
- 更新服务器数据
- 删除服务器数据
- 服务器数据属性



装载服务器数据

通过 *服务器数据* → *装载...* 和一个文件打开对话框可实现数据的导入，用户可从中选择一个数据包。这样，可将数据包复制给\项目文件夹\数据包下的项目文件夹。

随后对用于该多客户机项目的名称服务进行组态。

如果此时还不存在具有该符号计算机名称的任何条目，则该数据包将作为新数据包进行记录，并在数据包条目下带下列信息显示：

- 具有 *符号计算机名称* 的服务器项目正在运行
- 数据包的名称(*Server_symbolic computer name.pck*)
- 创建日期

如果名称服务已知道数据包的符号计算机名称的条目，则在导入数据包时，可使用下列选项：取消输入、重新命名符号计算机名称或使用新的数据来覆盖名称服务中的现有符号计算机名称。

更新服务器数据

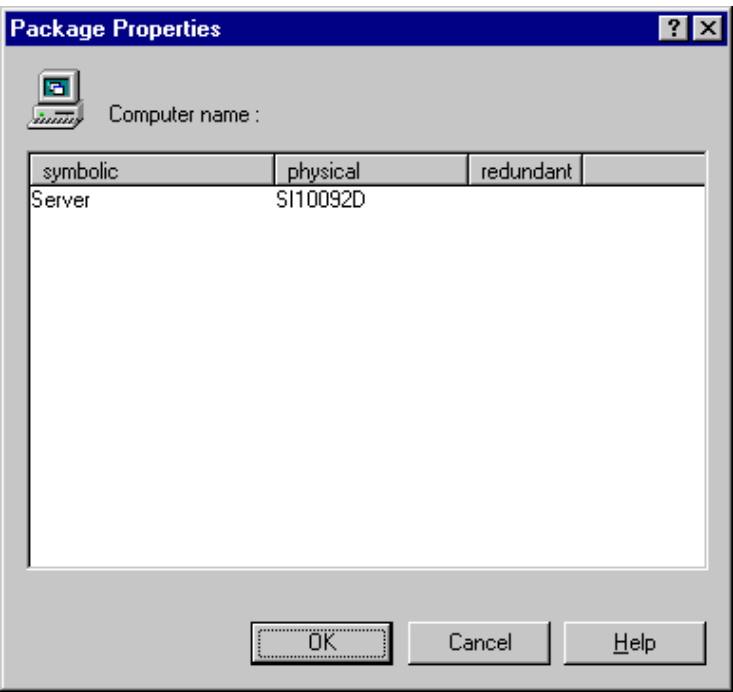
通过该菜单条目，可对所有导入的数据包进行更新。

删除服务器数据

通过该菜单条目，可删除所有导入的数据包。

服务器数据属性

通过该菜单条目，可指定优先使用的服务器，或在冗余系统中，还指定其冗余伙伴。

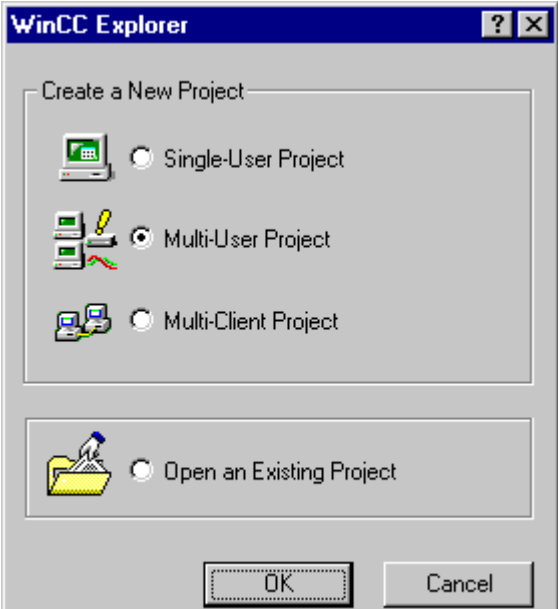


注意：
为了在运行实例项目时不出现问题，必须首先在两台服务器计算机上的服务器项目中生成数据包。随后，在数据包的属性对话框中，将第一个服务器项目的符号计算机名称重新命名为 *Server_1*，或将第二个服务器项目的相应名称重新命名为 *Server_2*。只有这时，才将服务器的数据包装载到多客户机项目中。










2.3 Project_MultiClient_Server 项目的创建

以下详细描述了创建多客户机项目 *Project_MultiClient_Server* 所需的步骤。
本项目是基于炉温控制的模拟过程，在两台服务器上运行。在图形编辑器、变量记录、报警记录和全局脚本编辑器中进行组态。

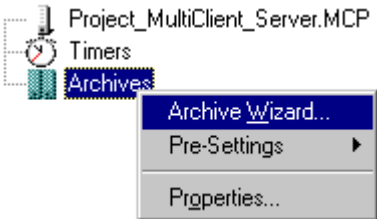
创建服务器项目

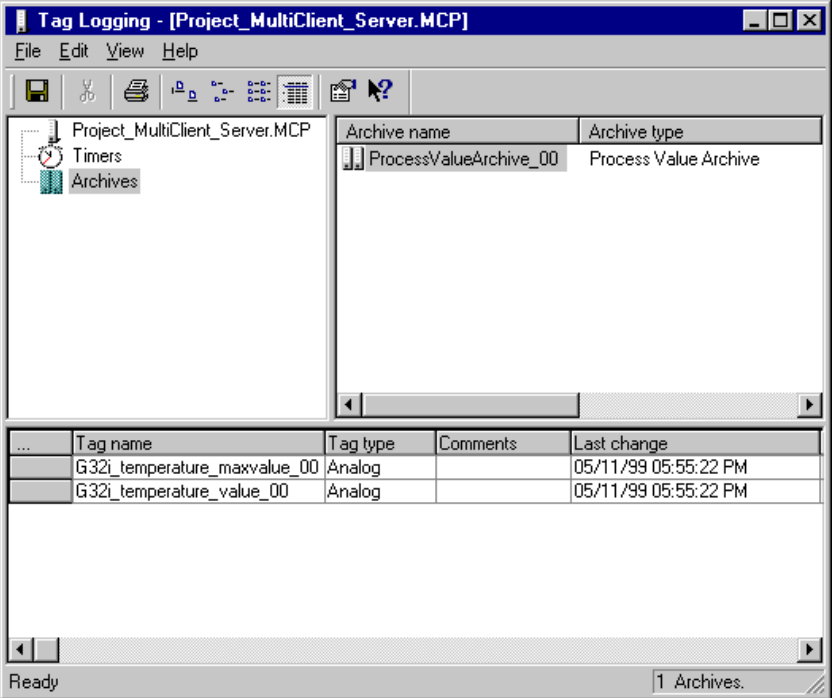
步骤	过程：创建服务器项目
1	创建新的 WinCC 项目。 通过开始 → <i>Simatic</i> → <i>WinCC</i> → 视窗控制中心来启动 WinCC 资源管理器。
2	将显示 <i>WinCC 资源管理器</i> 。 通过菜单 <i>文件</i> → <i>新建</i> ，将打开用来指定新的 WinCC 项目属性的对话框。 将本实例项目创建为 <i>多用户项目</i> 。 通过单击 <i>确定</i> 退出对话框。 

创建变量


步骤	过程：创建变量									
1	<p>在本实例中，创建三个名称如下的内部变量： 变量 <i>G32i_temperature_value_00</i> 对应于温度的实际值，变量 <i>G32i _ temperature_maxvalue_00</i> 对应于温度的设定值，变量 <i>U08i_power_value_00</i> 对应于加热量。</p> <table><tr><td></td><td><i>G32i_temperature_value_00</i></td><td>Floating-point number 32-bit IEEE 754</td></tr><tr><td></td><td><i>G32i_temperature_maxvalue_00</i></td><td>Floating-point number 32-bit IEEE 754</td></tr><tr><td></td><td><i>U08i_power_value_00</i></td><td>Unsigned 8-bit value</td></tr></table>		<i>G32i_temperature_value_00</i>	Floating-point number 32-bit IEEE 754		<i>G32i_temperature_maxvalue_00</i>	Floating-point number 32-bit IEEE 754		<i>U08i_power_value_00</i>	Unsigned 8-bit value
	<i>G32i_temperature_value_00</i>	Floating-point number 32-bit IEEE 754								
	<i>G32i_temperature_maxvalue_00</i>	Floating-point number 32-bit IEEE 754								
	<i>U08i_power_value_00</i>	Unsigned 8-bit value								

创建过程值归档


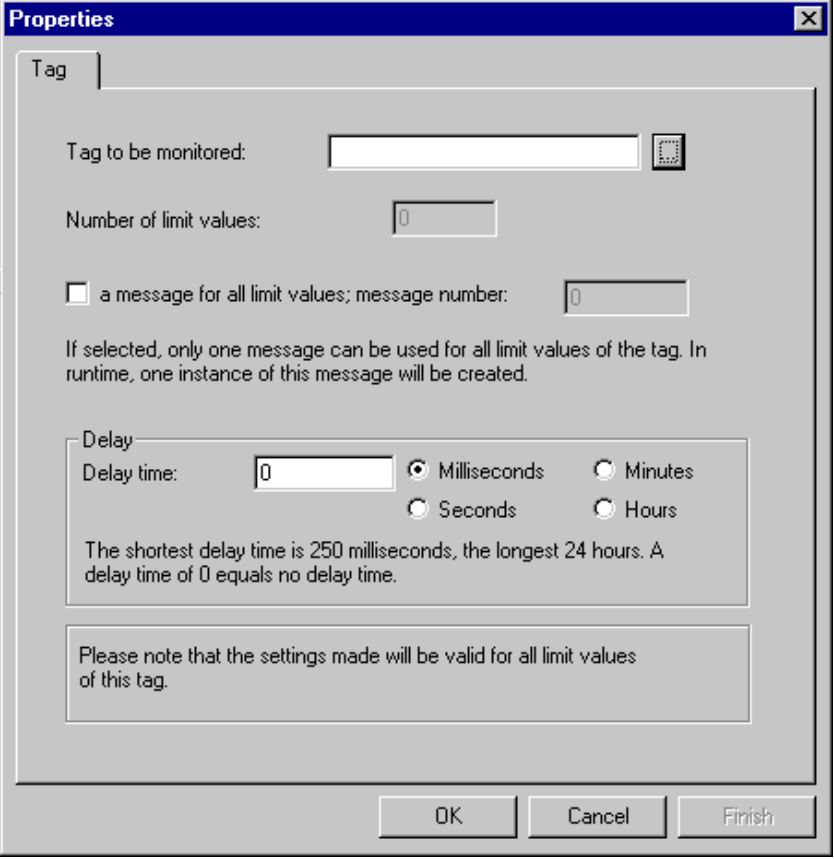


步骤	过程：创建过程值归档
1	打开变量记录编辑器。
2	<p>过程值归档的创建。为此通过  <i>归档</i> 来启动归档向导。</p> 

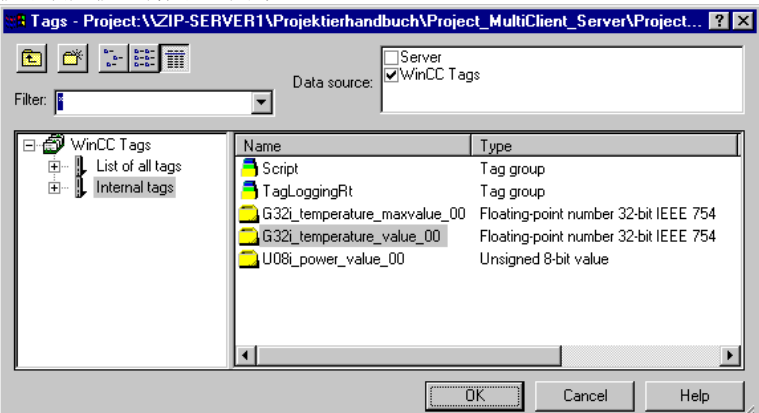

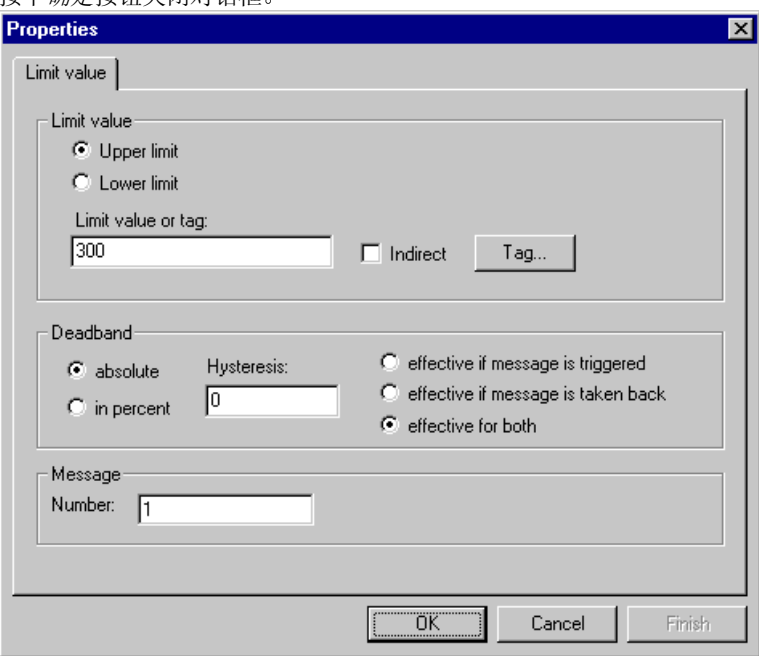
步骤	过程：创建过程值归档																
3	<p>在本实例中，归档已经命名为 <i>ProcessValueArchive_00</i>。 将变量 <i>G32i_temperature_value_00</i> 和 <i>G32i_temperature_maxvalue_00</i> 指定为归档变量。</p>  <table border="1"> <thead> <tr> <th>Archive name</th><th>Archive type</th></tr> </thead> <tbody> <tr> <td>ProcessValueArchive_00</td><td>Process Value Archive</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Tag name</th><th>Tag type</th><th>Comments</th><th>Last change</th></tr> </thead> <tbody> <tr> <td>G32i_temperature_maxvalue_00</td><td>Analog</td><td></td><td>05/11/99 05:55:22 PM</td></tr> <tr> <td>G32i_temperature_value_00</td><td>Analog</td><td></td><td>05/11/99 05:55:22 PM</td></tr> </tbody> </table> <p>Ready 1 Archives.</p>	Archive name	Archive type	ProcessValueArchive_00	Process Value Archive	Tag name	Tag type	Comments	Last change	G32i_temperature_maxvalue_00	Analog		05/11/99 05:55:22 PM	G32i_temperature_value_00	Analog		05/11/99 05:55:22 PM
Archive name	Archive type																
ProcessValueArchive_00	Process Value Archive																
Tag name	Tag type	Comments	Last change														
G32i_temperature_maxvalue_00	Analog		05/11/99 05:55:22 PM														
G32i_temperature_value_00	Analog		05/11/99 05:55:22 PM														
4	保存后退出变量记录编辑器。																

组态报警记录

步骤	过程：组态报警记录																														
1	打开报警记录编辑器。																														
2	<p>单个消息的创建。在报警记录编辑器的底部窗口中，显示已经组态的消息。通过R，可添加新的行。在本实例中，需要两条不同的消息。必须对错误类型、消息文本和出错点进行相应修改。</p> <table><tr><th>...</th><th>Number</th><th>Class</th><th>Type</th><th>MessageTag</th><th>MessageBit</th><th>Status tag</th><th>Status bit</th><th>Message text</th><th>Point of error</th></tr><tr><td></td><td>1</td><td>Error</td><td>Warning</td><td></td><td>0</td><td></td><td>0</td><td>normal</td><td>Oven</td></tr><tr><td></td><td>2</td><td>Error</td><td>Alarm</td><td></td><td>0</td><td></td><td>0</td><td>critical</td><td>Oven</td></tr></table>	...	Number	Class	Type	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error		1	Error	Warning		0		0	normal	Oven		2	Error	Alarm		0		0	critical	Oven
...	Number	Class	Type	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error																						
	1	Error	Warning		0		0	normal	Oven																						
	2	Error	Alarm		0		0	critical	Oven																						

步骤	过程：组态报警记录
3	<p>对限制值监控的组态如果 <i>限制值监控(模拟报警)</i> 条目没有显示，则必须首先将其装载。可通过 <i>报警记录</i> 中的 <i>选项</i> → <i>加载项</i> 菜单完成该操作。在所显示的对话框中，必须选择 <i>限制值监控(模拟报警)</i> 复选框。单击 <i>确定</i> 按钮关闭对话框。</p> <div data-bbox="521 394 1055 1121"><div>Add Ins</div><div>Select the Add Ins that your project requires from here:</div><div>Add Ins:</div><div><div><input checked="" type="checkbox"/> Analog Alarm</div></div><div>Description</div><div></div><div>This dialog box can also be displayed via the menu command "Extras" and the entry "Add Ins ..."</div><div><div>OK</div><div>Cancel</div></div></div>

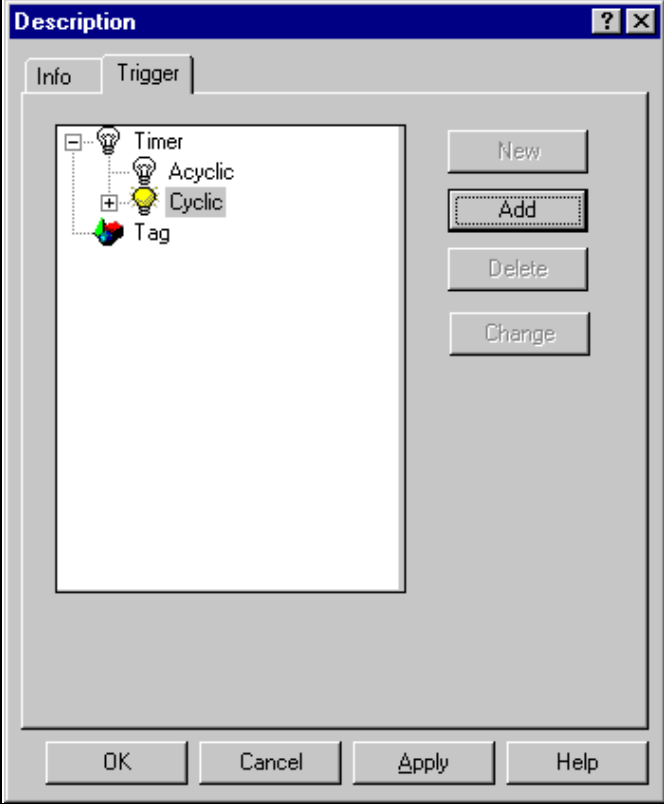
步骤	过程：组态报警记录
4	<p>通过  限制值监控条目，然后选择 → 新建...来访问变量的属性对话框。在该对话框中，可设置一个用于限制值监控的新变量。</p> <div><p>The image shows a 'Properties' dialog box for a 'Tag'. It has a 'Tag' tab. Inside, there's a 'Tag to be monitored:' field with a dropdown arrow. Below it is 'Number of limit values:' with a text box containing '0'. Then there's a checkbox 'a message for all limit values; message number:' with a text box containing '0'. A note says: 'If selected, only one message can be used for all limit values of the tag. In runtime, one instance of this message will be created.' Below that is a 'Delay' section with 'Delay time:' text box containing '0' and four radio buttons: 'Milliseconds' (selected), 'Minutes', 'Seconds', and 'Hours'. Another note says: 'The shortest delay time is 250 milliseconds, the longest 24 hours. A delay time of 0 equals no delay time.' At the bottom, a box says: 'Please note that the settings made will be valid for all limit values of this tag.' At the very bottom are 'OK', 'Cancel', and 'Finish' buttons.</p></div>
5	<p>通过  下面所显示的按钮，访问选择变量对话框。</p> <div></div>

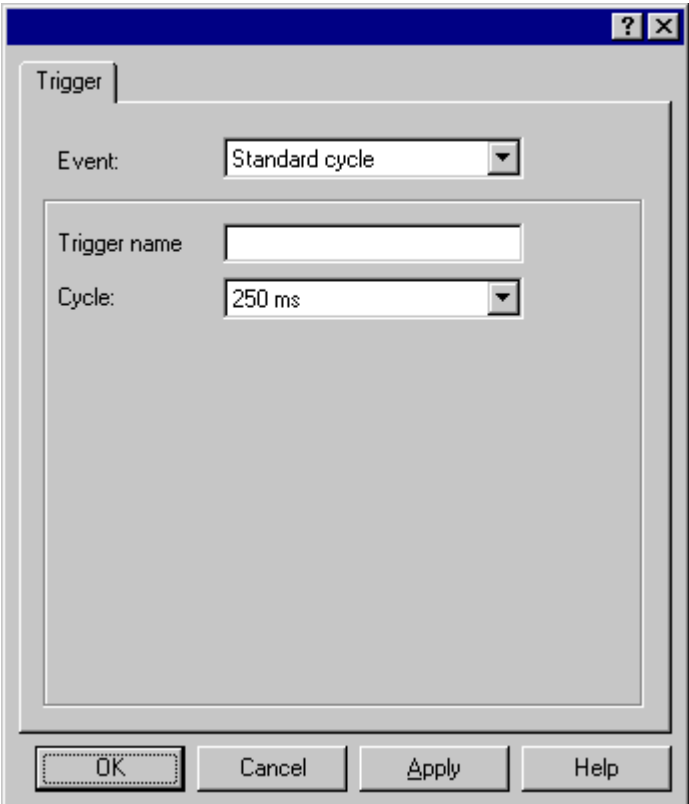
步骤	过程：组态报警记录
6	<p>在左边的窗口中，选择<i>内部变量</i>条目。右边的窗口将列出相应的变量。选择所期望的变量。在本实例中，它就是变量 <i>G32i_temperature_value_00</i>。</p> <p>按下 <i>确定</i>按钮将关闭对话框。</p> 
7	<p>同样使用确定按钮来关闭变量的<i>属性</i>对话框。于是报警记录的右边窗口将显示要监控的新变量的图标。通过  <i>G32i_temperature_value_00</i> → <i>new</i>，访问限制值的<i>属性</i>对话框。在该对话框中，可以将新的限制值分配给变量。在本实例中，将<i>上限</i>设置为 300，将消息编号设置为 1。</p> <p>按下 <i>确定</i>按钮关闭对话框。</p> 

步骤	过程：组态报警记录
8	采用先前所描述的步骤，将第二个限制值分配给变量。将上限设置为 700，将消息编号设置为 2。
9	保存后退出报警记录编辑器。

创建全局动作

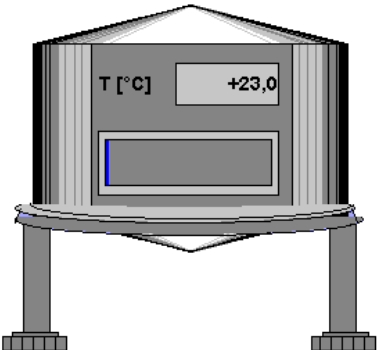
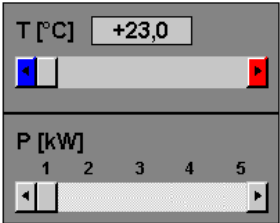
步骤	过程：创建全局动作
1	打开全局脚本编辑器。
2	创建新的全局动作。这可通过全局脚本编辑器中的文件 → 新建动作菜单来完成。
3	在本实例中，已经编写了将 C 函数模拟为趋势的 C 动作。计算设定值温度 <i>dTemp2</i> 和实际温度 <i>dTemp1</i> 之间的差值 <i>dDelta</i> 。如果该差值为正，则趋势上升。如果该差值为负，则趋势下降。 加热量 <i>nPower</i> 定义温度以多快的速度达到设定值。
4	通过编辑 → 编译来编译 C 动作。
5	通过编辑 → 信息，打开描述对话框。在触发器标签内，本实例选择了周期定时器。通过添加按钮，显示更改触发器的对话框。



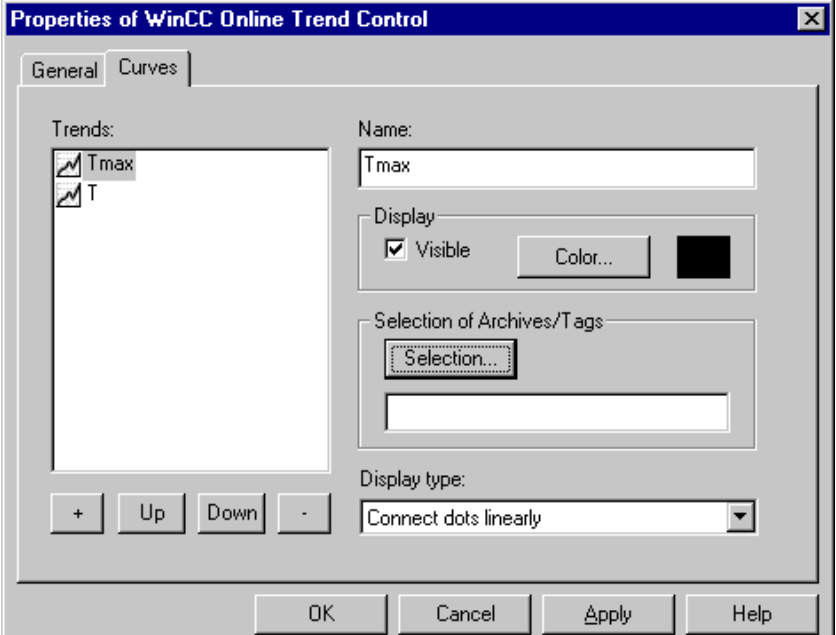
步骤	过程：创建全局动作
6	<p data-bbox="524 302 828 359">将周期时间设置为 250 毫秒。 按下 确定 关闭两个对话框。</p> <div data-bbox="524 359 1209 1157"></div>
7	<p data-bbox="524 1169 821 1194">保存后退出 全局脚本编辑器。</p>

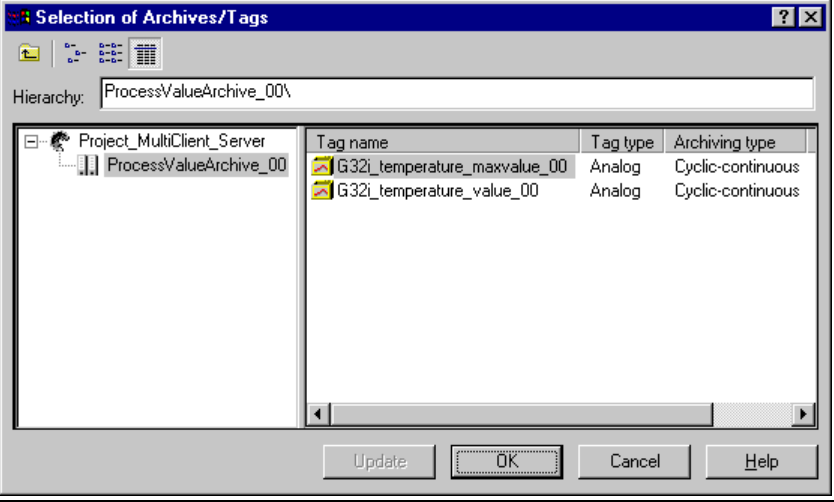
组态对象

步骤	过程：组态对象
1	在图形编辑器中创建一个新画面。在本实例中，它是画面 <i>mcs_3_chapter_01.PDL</i> 。在本画面中，将各种不同的对象连接到过程变量上。
2	<p>可通过 Windows 对象 → 滚动条对象实现输入变量的模拟。在本实例中，它们是滚动条对象 1 (<i>G32i_temperature_maxvalue_00</i>)和滚动条对象 2(<i>U08/_power_value_00</i>)以及代表温度控制的 I/O 域 1 (<i>G32i_temperature_maxvalue_00</i>)。在 I/O 域 1 中，显示设定的温度值，也可以在此处将其更改。</p> <p>在炉中显示输出变量(<i>G32i_temperature_value_00</i>)。它包括对象 I/O 域 2 和棒图 1。</p> <p>将这些对象的更新设置为一旦改变。</p>

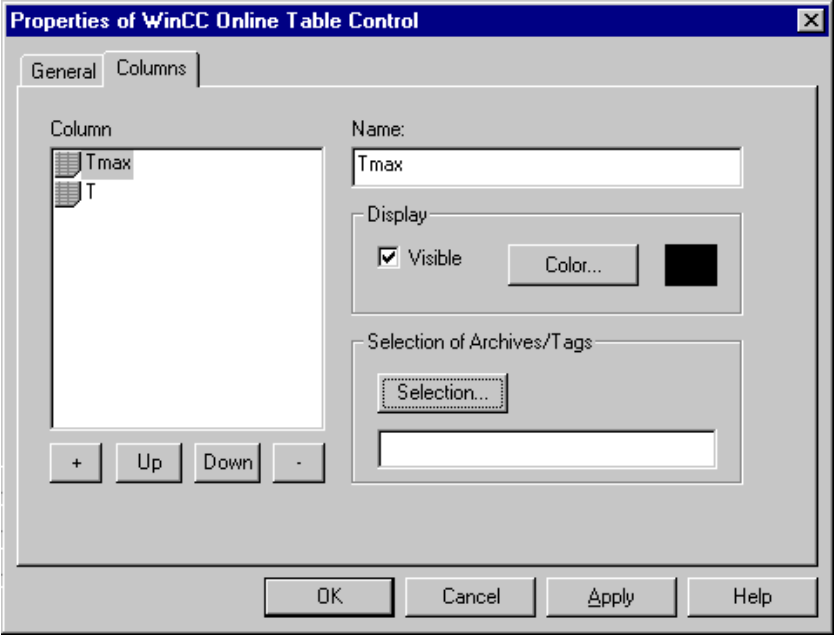


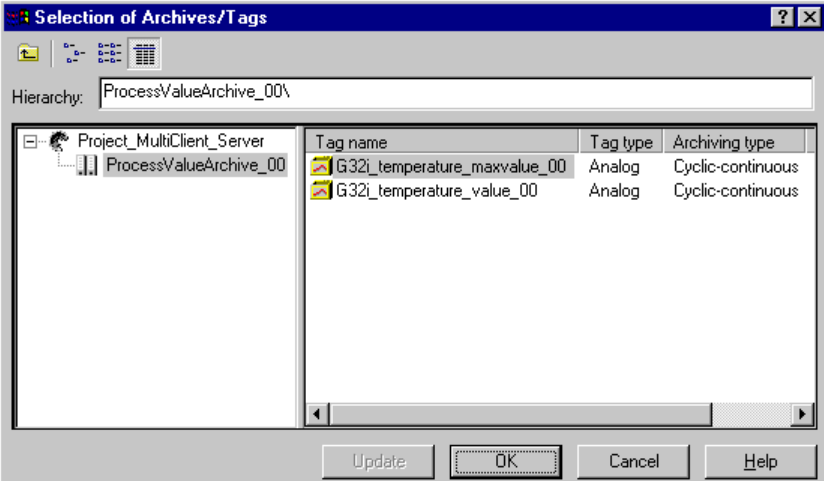
组态趋势窗口

步骤	过程：组态趋势窗口
1	在图形编辑器中另外创建一个画面。在实例中，它就是画面 <i>mcs_3_chapter_02.PDL</i> 。在该画面中，使用趋势窗口显示两个温度值。
2	<p>通过控件 → WinCC 在线趋势控件创建趋势控件。在实例中，这就是对象 <i>TlgOnlineTrend1</i>。显示对话框 <i>WinCC 在线趋势控件属性</i>。通过单击+按钮在趋势标签中添加一条新趋势。</p> <p>趋势 1 重命名为 <i>Tmax</i>，趋势 2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

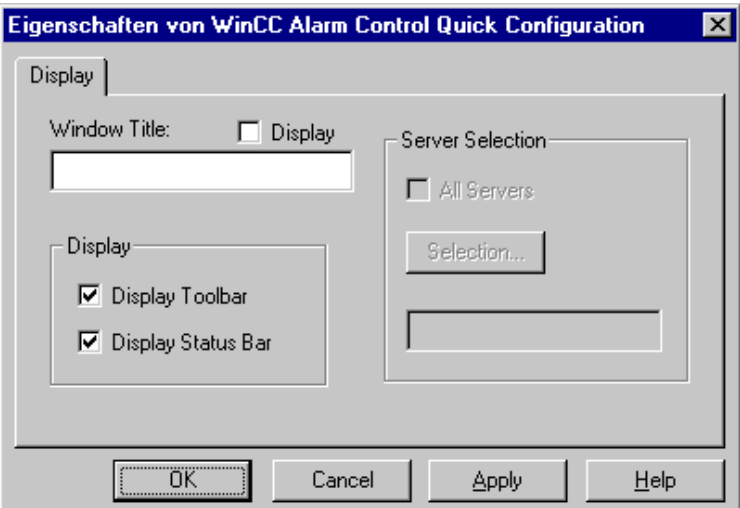
步骤	过程：组态趋势窗口
3	<p>显示 归档/变量选择对话框。该对话框允许选择归档/归档变量。</p> <p>在本实例中，将 <i>Tmax</i> 与变量 <i>G32i_temperature_value_00</i> 相连接，将 <i>T</i> 与变量 <i>G32i_temperature_maxvalue</i> 相连接。</p> 

组态表格窗口



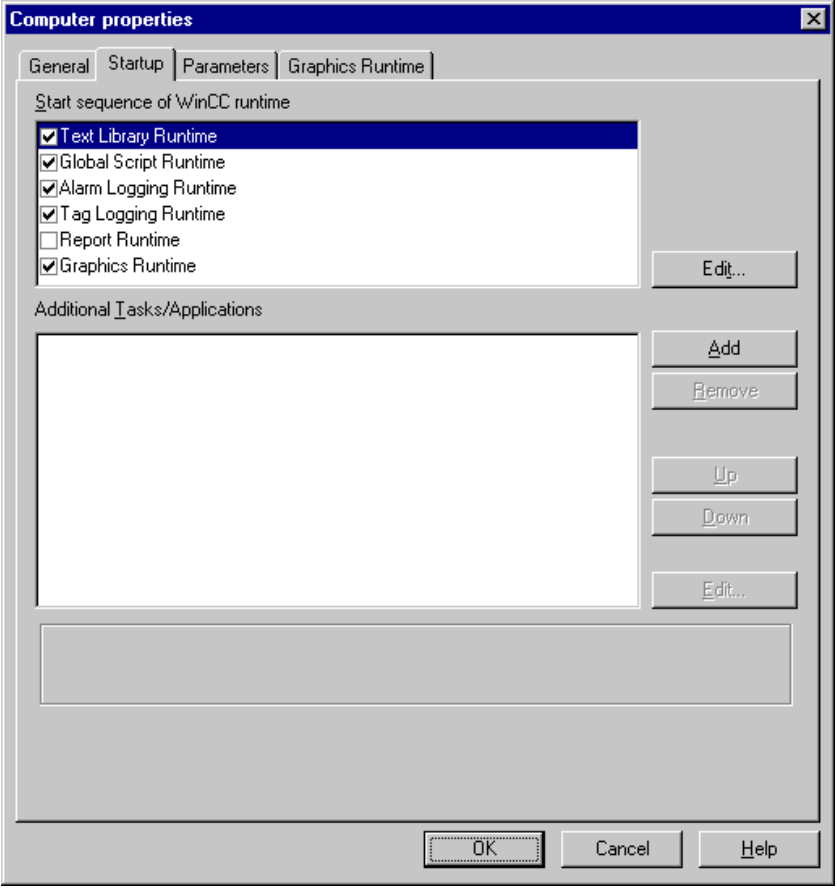
步骤	过程：组态表格窗口
1	在同一画面(<i>mcs_3_chapter_02.PDL</i>)中，使用表格窗口显示两个温度值。
2	<p>通过控件 → <i>WinCC 在线表格控件</i>，创建表格控件。在本实例中，这就是对象 <i>TlgOnlineTable1</i>。显示对话框 <i>WinCC 在线表格控件属性</i>。在列标签内，通过单击+按钮添加新列。</p> <p>列1 重命名为 <i>Tmax</i>，列2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

步骤	过程：组态表格窗口
3	<p>显示 <i>归档/变量选择</i> 对话框。该对话框允许选择归档/归档变量。</p> <p>在本实例中，将 <i>Tmax</i> 与变量 <i>G32i_temperature_value_00</i> 相连接，将 <i>T</i> 与变量 <i>G32i_temperature_maxvalue</i> 相连接。</p>  <p>The screenshot shows a dialog box titled "Selection of Archives/Tags". It has a "Hierarchy:" field containing "ProcessValueArchive_00\". Below this is a tree view showing "Project_MultiClient_Server" and "ProcessValueArchive_00\". To the right is a table with columns "Tag name", "Tag type", and "Archiving type". The table contains two rows: "G32i_temperature_maxvalue_00" and "G32i_temperature_value_00", both with "Analog" tag type and "Cyclic-continuous" archiving type. At the bottom are buttons for "Update", "OK", "Cancel", and "Help".</p>




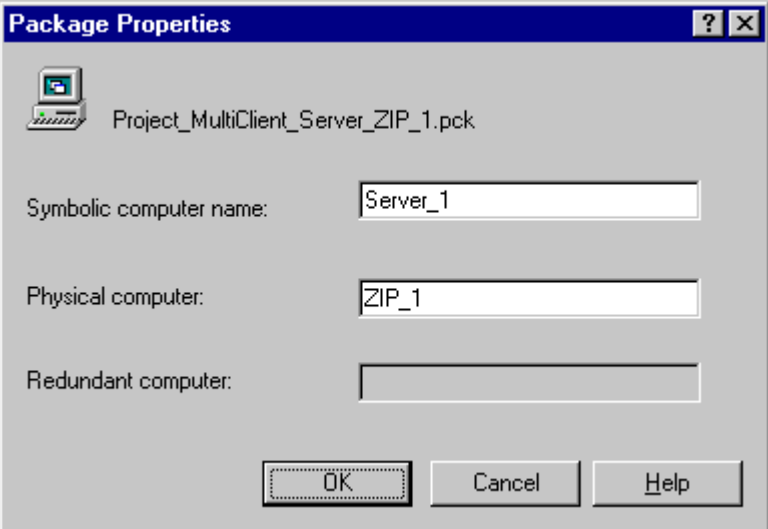
组态消息窗口

步骤	过程：组态消息窗口
1	<p>在 <i>图形编辑器</i> 中另外创建一个画面。在本实例中，它就是画面 <i>mcs_3_chapter_03.PDL</i>。在该画面中，利用消息窗口输出所组态的消息。</p>
2	<p>通过 <i>控件</i> → <i>WinCC 报警控件</i>，创建一个 <i>报警控件</i>。在实例中，这就是对象 <i>CCAlgWinCtrl1</i>。显示对话框 <i>WinCC 报警控件属性 - 快速组态</i>。按下 <i>确定</i> 关闭该对话框。</p>  <p>The screenshot shows a dialog box titled "Eigenschaften von WinCC Alarm Control Quick Configuration". It has a "Display" tab. The "Window Title:" field is empty, with a "Display" checkbox next to it. Below this is a "Display" section with checkboxes for "Display Toolbar" and "Display Status Bar", both of which are checked. To the right is a "Server Selection" section with a "All Servers" checkbox and a "Selection..." button. At the bottom are buttons for "OK", "Cancel", "Apply", and "Help".</p>

设置 WinCC 运行系统启动属性

步骤	过程: 设置 WinCC 运行系统启动属性
1	通过  WinCC 资源管理器左边的 计算机 条目, 将在右边显示计算机名称。
2	<p>通过  计算机名称 → 属性, 显示对话框 计算机属性。在 启动 标签中, 完成以下设置。单击 确定 按钮关闭对话框。</p> 

生成服务器数据

步骤	过程：生成服务器数据
1	<p>通过  RWinCC 资源管理器左边的 <i>服务器数据</i> → <i>生成</i> 来生成服务器数据。将显示一消息，声明服务器数据已成功生成。用确定按钮确认该对话框。随后，在 WinCC 资源管理器的右边将显示所生成的程序包。</p> 
2	<p>通过  R 新生成的程序包和 <i>属性</i>，显示对话框 <i>程序包属性</i>。在本实例中，将 <i>符号计算机名称</i> 重命名为 <i>Server_1</i>。按下 <i>确定</i> 按钮关闭对话框。</p> 
3	<p>采用先前所描述的步骤，在第二台服务器上生成服务器数据并将它的 <i>符号计算机名称</i> 重命名为 <i>Server_2</i>。</p>

2.4 Project_MultiClient_Client 项目的创建

以下详细描述了创建多客户机项目 *Project_MultiClient_Client* 所需的步骤。

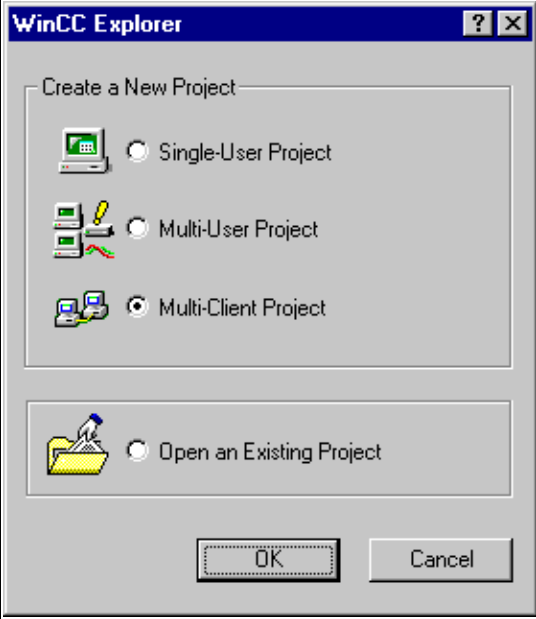
本项目引用两个先前组态的服务器的数据。

多客户机项目只能组态其自身的数据，而不能组态服务器项目的数据。但是它可以引用服务器项目上的数据(提供所谓的服务器的视图)。

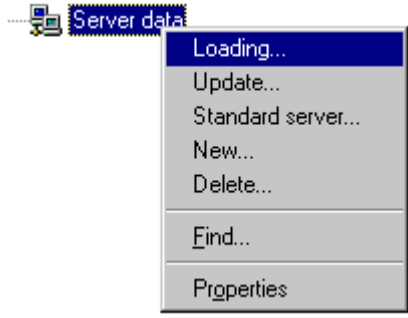
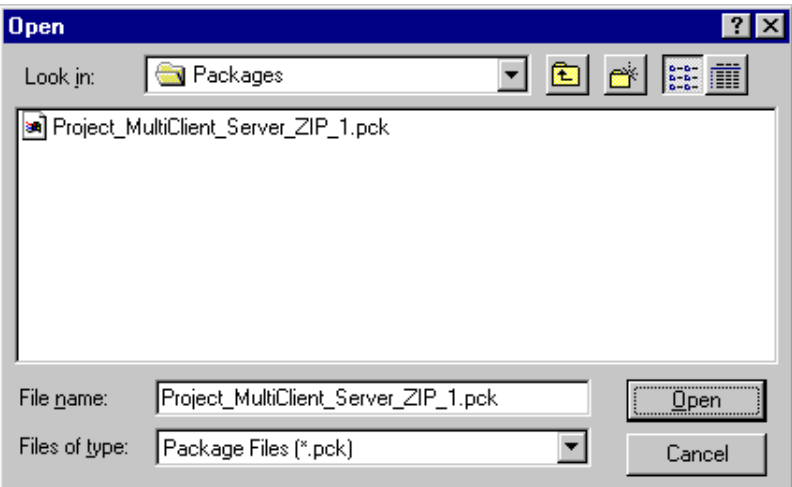
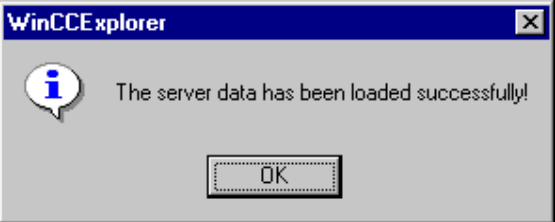
在组态阶段，需要服务器数据(数据包)来使一台或多台服务器的相关数据能被多客户机项目所使用。

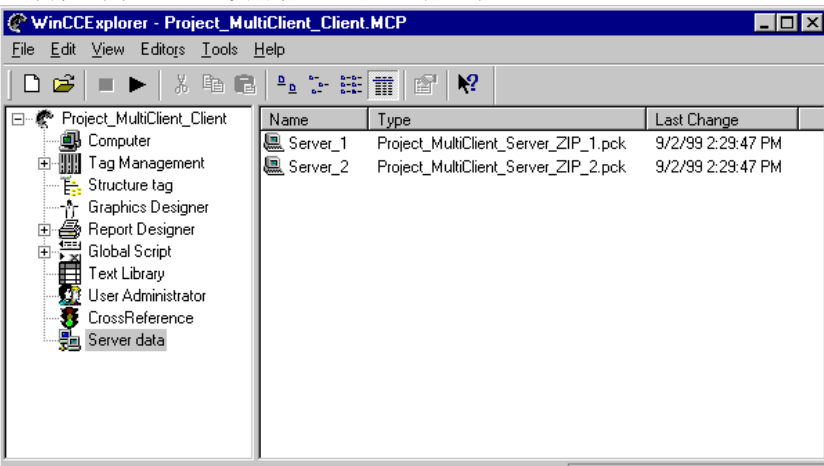
多客户机项目拥有其自己的画面，但是也可以显示一台或多台服务器的画面。

创建多客户机项目


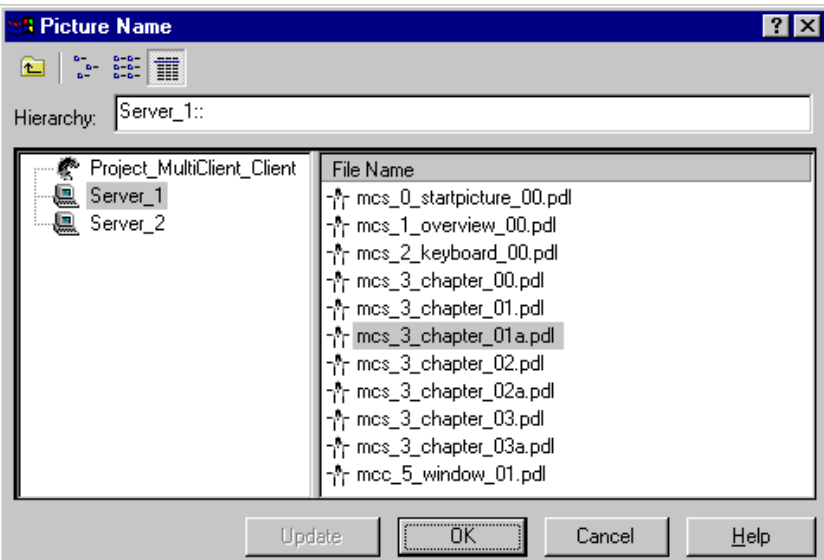
步骤	过程：创建多客户机项目
1	创建新的 WinCC 项目。 通过开始 → <i>Simatic</i> → <i>WinCC</i> → <i>视窗控制中心</i> 来启动 WinCC 资源管理器。
2	将显示 <i>WinCC 资源管理器</i> 。 通过菜单 <i>文件</i> → <i>新建</i> ，将打开用来指定新的 WinCC 项目属性的对话框。 将本实例项目创建为 <i>多客户机项目</i> 。 通过单击 <i>确定</i> 退出对话框。 <div></div>

装载服务器数据



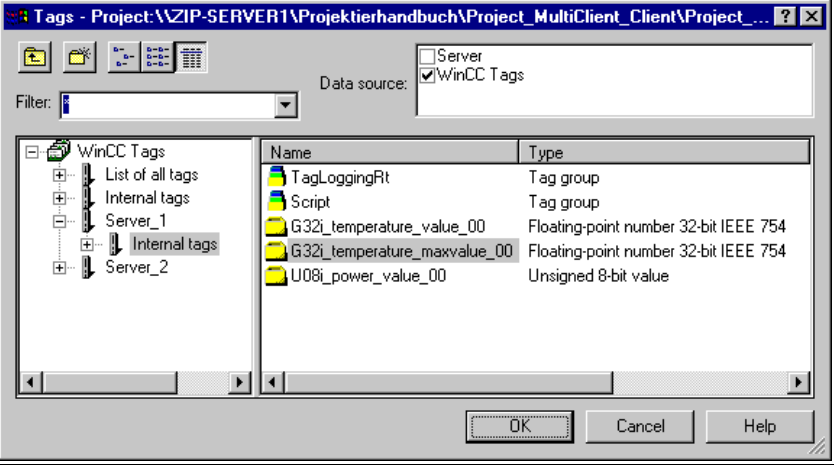
步骤	过程：装载服务器数据
1	<p>通过 WinCC 资源管理器左侧的服务器数据 → 装载来显示打开对话框。</p> 
2	<p>通过 网上邻居 条目选择服务器计算机。数据包文件位于服务器上的文件夹 <i>Project_MultiClient_Server</i> → 服务器名称 → 数据包。选择该文件并通过 打开 按钮将其载入。</p> 
3	<p>将会显示一个用于确认成功装载服务器数据的对话框。单击 确定 按钮关闭该对话框。</p> 

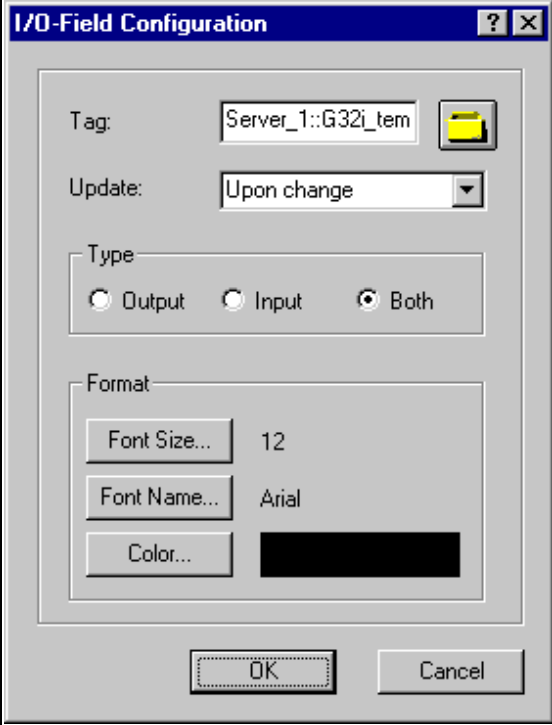
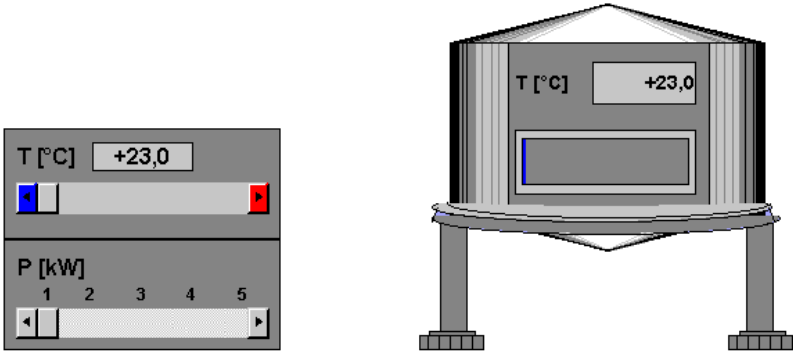
步骤	过程：装载服务器数据									
4	<p>按照刚才所描述的步骤，装载第二个服务器的数据包文件。两个已装载的数据包都将显示在 WinCC 资源管理器的右边窗口中。</p>  <table border="1" data-bbox="784 459 1344 537"><thead><tr><th>Name</th><th>Type</th><th>Last Change</th></tr></thead><tbody><tr><td>Server_1</td><td>Project_MultiClient_Server_ZIP_1.pck</td><td>9/2/99 2:29:47 PM</td></tr><tr><td>Server_2</td><td>Project_MultiClient_Server_ZIP_2.pck</td><td>9/2/99 2:29:47 PM</td></tr></tbody></table>	Name	Type	Last Change	Server_1	Project_MultiClient_Server_ZIP_1.pck	9/2/99 2:29:47 PM	Server_2	Project_MultiClient_Server_ZIP_2.pck	9/2/99 2:29:47 PM
Name	Type	Last Change								
Server_1	Project_MultiClient_Server_ZIP_1.pck	9/2/99 2:29:47 PM								
Server_2	Project_MultiClient_Server_ZIP_2.pck	9/2/99 2:29:47 PM								

组态服务器的视图

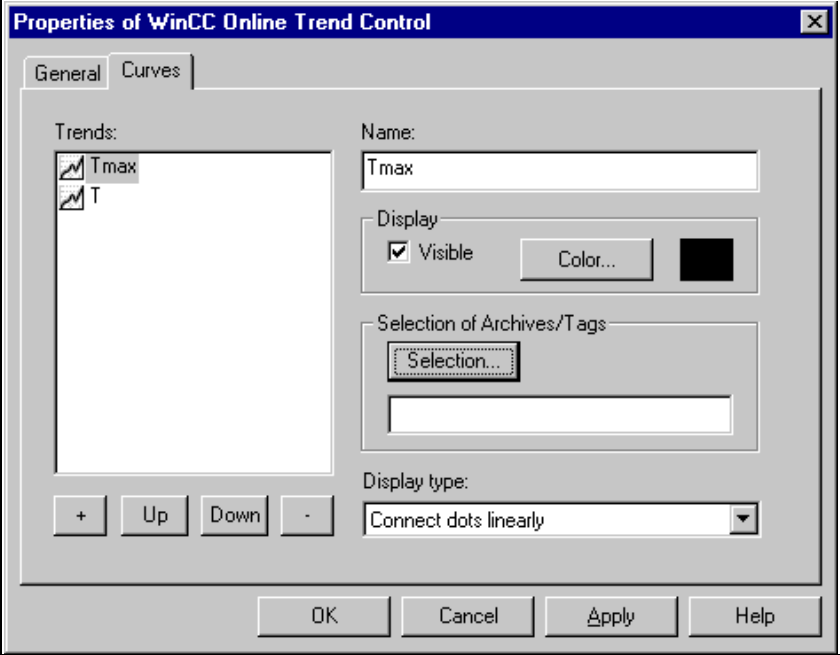
步骤	过程：组态服务器的视图
1	在图形编辑器中创建一个新画面。在本实例中，它是画面 <i>mcc_3_chapter_01.PDL</i> 。在该画面中，服务器上所组态的画面用画面窗口来显示。
2	<p>组态一个智能对象 → 画面窗口。</p> <p>在其对象属性对话框中，通过  属性 → 其它 → 画面名称来打开画面名称对话框。在该对话框中，可以指定要在画面窗口中显示的画面。为了选择服务器画面，必须首先在左边窗口中选择期望的服务器。随后，与该服务器相关的画面文件将显示在右边窗口中。选择期望的画面。在本实例中，它是 <i>Server_1</i> 的画面 <i>mcs_3_chapter_01a.PDL</i>。</p> <p>单击确定按钮关闭对话框。</p> 
3	组态另一个智能对象 → 画面窗口。在该画面窗口中，显示 <i>Server_2</i> 的画面 <i>mcs_3_chapter_01a.PDL</i> 。
4	按照步骤 1 至 3，组态另两个画面。在这些画面中，显示两台服务器的 <i>mcs_3_chapter_02a.PDL</i> 与 <i>mcs_3_chapter_03a.PDL</i> 画面。

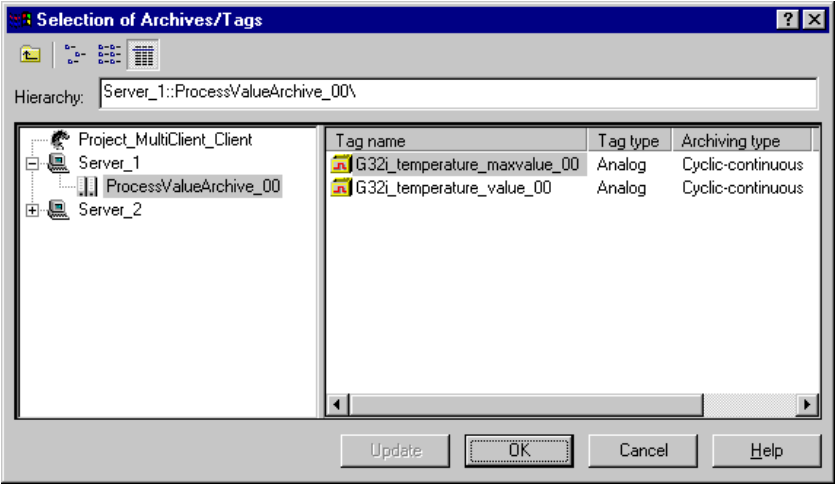
组态对象

步骤	过程：组态对象
1	在 图形编辑器 中另外创建一个画面。在本实例中，它是画面 <i>mcc_3_chapter_11.PDL</i> 。在该画面中，不同的对象与 <i>Server_1</i> 的过程变量相连。
2	<p>组态一个智能对象 → <i>I/O</i> 域。在本实例中，它是对象 <i>I/O</i> 域 1。将显示其组态对话框。</p> <p>通过  如下所示的按钮，访问 选择变量对话框。</p> 
3	<p>在左边窗口中，选择所期望的服务器的内部变量条目。右边窗口将列出相应的变量。选择所期望的变量。在本实例中，它是 <i>Server_1</i> 的变量 <i>G32i_temperature_maxvalue_00</i>。</p> <p>通过 确定按钮关闭对话框。</p> 

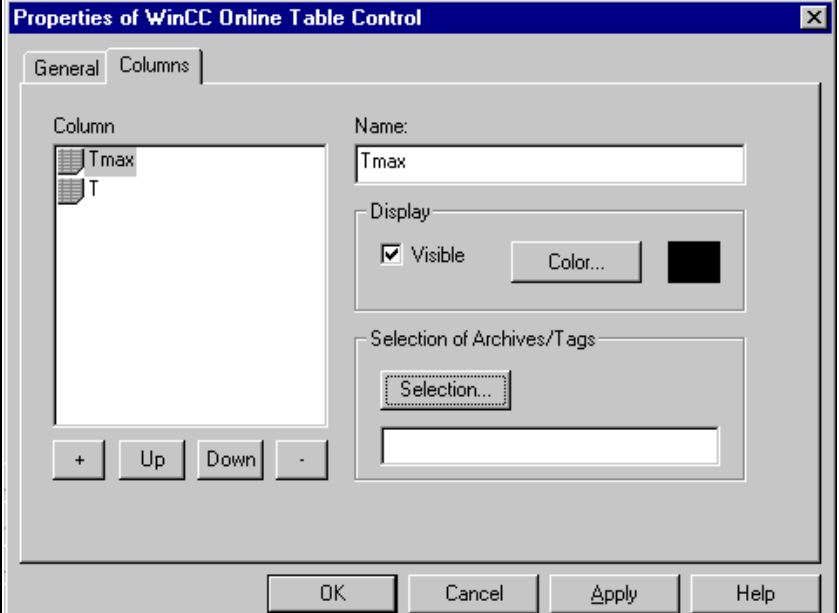
步骤	过程：组态对象
4	<p>将更新设置为一旦改变。通过确定按钮退出组态对话框。</p> 
5	<p>组态其它对象(I/O 域、滚动条对象、棒图)来显示服务器的其余变量。</p> 

组态趋势窗口

步骤	过程：组态趋势窗口
1	在图形编辑器中另外创建一个画面。在本实例中，它是画面 <i>mcc_3_chapter_12.PDL</i> 。在该画面中，用趋势窗口显示 <i>Sever_1</i> 的两个温度值。
2	<p>通过控件 → WinCC 在线趋势控件创建一个趋势控件。在本实例中，它是对象 <i>TlgOnlineTrend1</i>。显示 WinCC 在线趋势控件属性对话框。在趋势标签中，通过单击+按钮添加新趋势。</p> <p>将趋势 1 重命名为 <i>Tmax</i>，将趋势 2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

步骤	过程：组态趋势窗口
3	<p>显示 归档/变量选择对话框。通过此对话框，可以从由数据包导入的服务器数据中选择服务器/归档/归档变量。</p> <p>在本实例中，将 <i>Tmax</i> 与 <i>Server_1</i> 的变量 <i>G32i_temperature_value_00</i> 相连，将 <i>T</i> 与变量 <i>G32i_temperature_maxvalue</i> 相连。</p>  <p>The screenshot shows the 'Selection of Archives/Tags' dialog box. The 'Hierarchy' text box at the top contains the path 'Server_1::ProcessValueArchive_00\'. Below this, a tree view on the left shows the project structure: 'Project_MultiClient_Client' containing 'Server_1' and 'Server_2'. Under 'Server_1', the 'ProcessValueArchive_00' folder is expanded. To the right of the tree is a table with three columns: 'Tag name', 'Tag type', and 'Archiving type'. The table contains two entries: 'G32i_temperature_maxvalue_00' and 'G32i_temperature_value_00', both with 'Analog' tag type and 'Cyclic-continuous' archiving type. At the bottom of the dialog are four buttons: 'Update', 'OK', 'Cancel', and 'Help'.</p>
4	<p>按照刚才所描述的步骤，另外再组态一个 <i>WinCC 在线趋势控件</i>。该控件与 <i>Server_2</i> 的变量相连。</p>

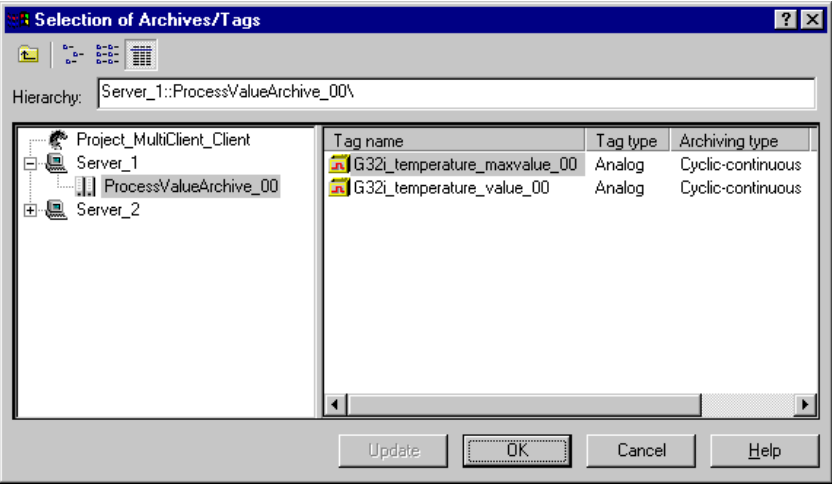
组态表格窗口

步骤	过程：组态表格窗口
1	在同一画面(mcc_3_chapter_12.PDL)中，用表格窗口显示 <i>Server_1</i> 的两个温度值。
2	<p>通过 控件 → WinCC 在线表格控件，创建一个表格控件。在本实例中，它是对象 TlgOnlineTable1。显示 WinCC 在线表格控件属性对话框。在列标签内，通过单击+按钮添加新列。</p> <p>将列 1 重命名为 <i>Tmax</i>，将列 2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

- 3

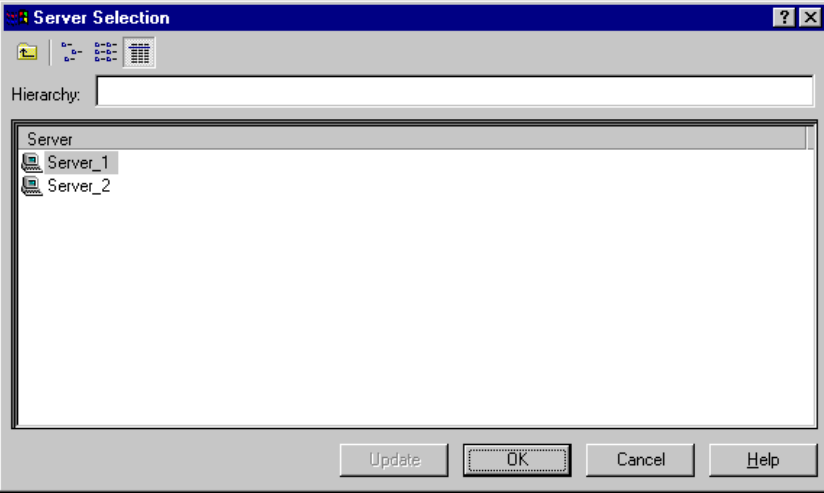
显示 归档/变量选择对话框。通过此对话框，可以从由数据包导入的服务器数据中选择服务器/归档/归档变量。

在本实例中，将 *Tmax* 与 *Server_1* 的变量 *G32i_temperature_value_00* 相连，将 *T* 与变量 *G32i_temperature_maxvalue* 相连。

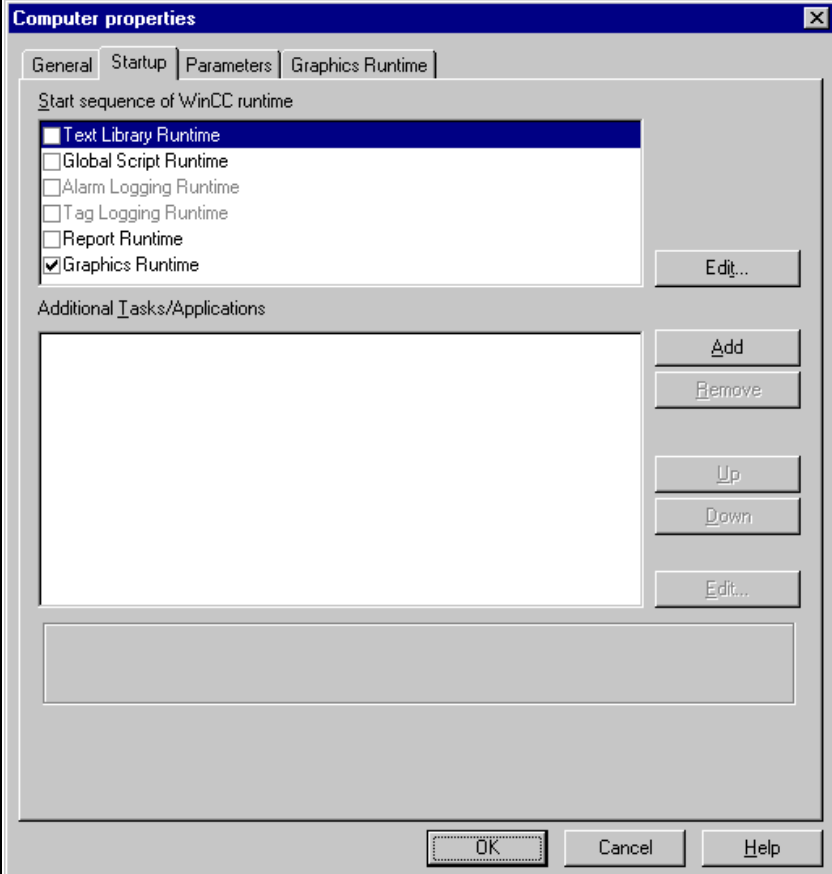

- 4

按照刚才所描述的步骤，另外再组态一个 *WinCC* 在线表格控件。该控件与 *Server_2* 的变量相连。

组态消息窗口

步骤	过程：组态消息窗口
1	在图形编辑器中另外创建一个画面。在本实例中，它是画面 <i>mcc_3_chapter_13.PDL</i> 。在该画面中，用消息窗口输出在 <i>Server_1</i> 上组态的消息。
2	通过控件 → <i>WinCC 报警控件</i> ，创建一个报警控件。在本实例中，它是对象 <i>CCAlgWinCtrl1</i> 。显示 <i>WinCC 报警控件属性 - 快速组态</i> 对话框。通过 <i>选择</i> 按钮，访问 <i>服务器选择</i> 对话框。
3	在本实例中，选择 <i>Server_1</i> 并通过 <i>确定</i> 按钮关闭对话框。 <div></div>
4	按照刚才所描述的步骤，组态一个 <i>WinCC 报警控件</i> 。该控件与 <i>Server_2</i> 相连。

设置 WinCC 运行系统启动属性

步骤	过程: 设置 WinCC 运行系统启动属性
1	通过 WinCC 资源管理器左边的计算机条目, 将在右边显示计算机名称。通过计算机名称 → 属性来显示计算机属性对话框。在启动标签中, 进行以下设置。
2	<p>在多客户机项目中, 设置图形运行系统属性。不能选择报警记录运行系统与变量记录运行系统属性。</p> <p>通过单击确定退出对话框。</p> 

2.5 WinCC 项目的描述

在两台服务器上都激活运行系统。此后，运行系统也可以在多客户机项目上激活。

如果运行系统在服务器项目上激活先于在多客户机项目上激活，则会产生通讯问题，因为多客户机项目引用服务器的数据。

2.5.1 服务器项目



在显示了总览画面后，可以通过如上所示的按钮访问设备画面。



通过如上所示的按钮，可在各画面之间进行切换。



通过此按钮，可返回至总览画面。

设备画面

在设备画面中，显示一个带温度控制的炉。通过该温度控制可以预先设置温度。炉温上升直至达到预设值。通过能量控制可以指定加热量。该值会影响炉温上升的速度。

趋势和表格窗口

在下一个画面中，显示趋势和表格窗口。趋势窗口描绘了预设温度(设定值)以及炉温(实际值)的进程。这两种数值还在表格窗口中进行显示。

消息窗口

下一个画面显示消息窗口。如果炉温超过数值 300，则产生一个警告并在消息窗口中显示。如果炉温超过数值 700，则产生一个报警并在消息窗口中显示。

2.5.2 多客户机项目

服务器的视图



在显示总览画面之后，可通过如上所示的按钮访问提供这两台服务器的视图的画面。

在第一个画面中，显示这两台服务器的设备画面。在后面的画面中，显示了两台服务器的趋势、表格和消息窗口。

与服务器的连接



通过如上所示的按钮，可访问在其中组态了多客户机对象的画面。这些对象已经与各种服务器过程变量相连接。

在第一个画面中，显示带温度控制的烘炉。显示服务器的过程变量。也可对其进行修改。在后面的画面中，组态了趋势、表格和消息窗口，它们再次与各服务器的数据相链接。

3 分布式服务器

本章中创建的项目也可以直接从在线文档复制到用户的硬盘驱动器上。缺省情况下，它们将存储在 *C:\Configuration_Manual* 文件夹中。可以选择将以下组件复制到硬盘驱动器上：



Project_DisServer_Server

将要创建的 WinCC 服务器项目。



Project_DisServer_Client

将要创建的 WinCC 客户机项目。

在本节中，将给出一个与分布式服务器相关的实例。

在 WinCC 项目 *Project_DisServer_Server* 和 *Project_DisServer_Client* 中组态了与该主题相关的实例。



3.1 常规信息

在 WinCC 中可以组态分布式系统，即由称之为多客户机的系统对 2 至 6 个服务器进行控制与操作。

通过多客户机完成分布可采用这种方法：在多客户机的画面中包含对 WinCC 服务器对象的引用。这些对象可以是变量、消息、画面或归档。

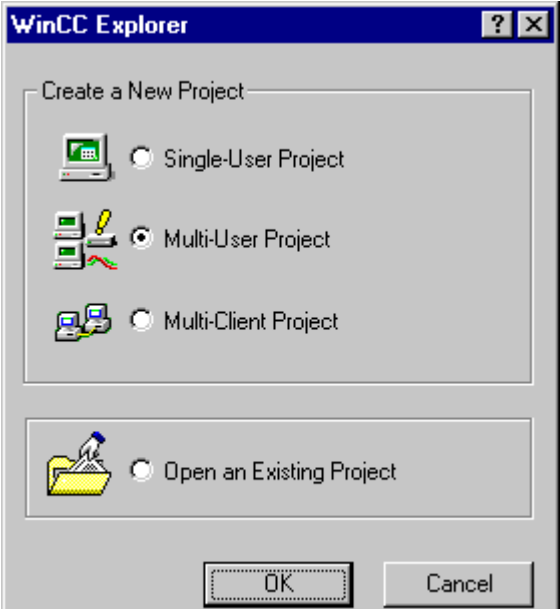
分布式系统的优点在于减轻了服务器的负载。

在实例中，显示了一个服务器项目和一个多客户机项目。于是，可在三台独立的计算机上启动服务器项目，每台计算机完成不同的功能。多客户机则获取相应服务器上的数据。




3.2 Project_DisServer_Server 项目的创建

以下详细描述了创建多客户机项目 *Project_DisServer_Server* 所需的步骤。
本项目以炉温控制的模拟为基础，在三台服务器上运行。每台服务器在运行系统中具有不同的启动属性，从而可完成不同的任务。在图形编辑器、变量记录、报警记录和全局脚本编辑器中进行组态。

创建服务器项目

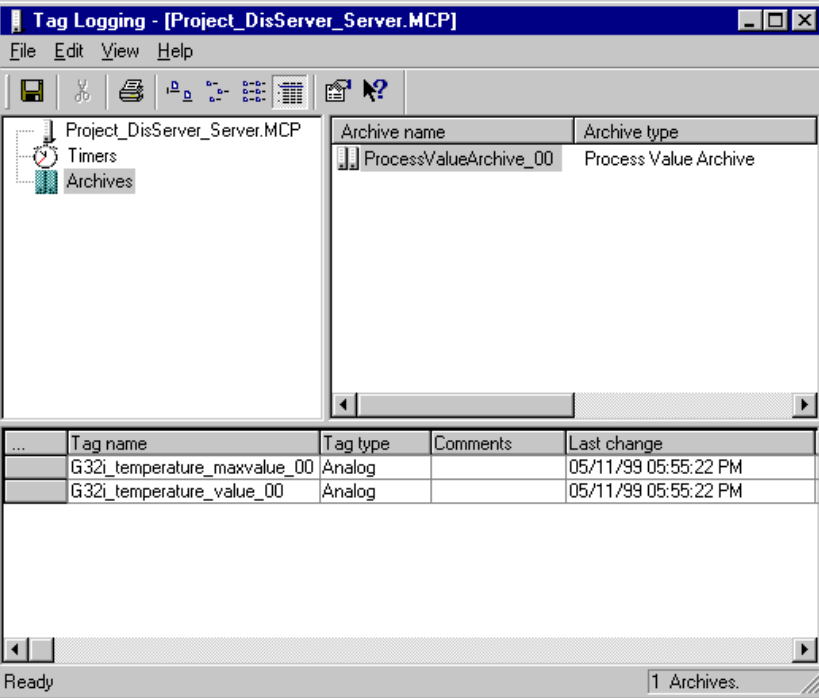
步骤	过程：创建服务器项目
1	创建新的 WinCC 项目。 通过 开始 → Simatic → WinCC → 视窗控制中心来启动 WinCC 资源管理器。
2	将显示 WinCC 资源管理器。 通过菜单 文件 → 新建，将打开用于指定新的 WinCC 项目属性的对话框。 将本实例项目创建为多用户项目。 通过单击 确定退出对话框。 

创建变量


步骤	过程：创建变量
1	<p>在本实例中，创建三个名称如下的内部变量：在对炉温控制进行模拟时需要这些变量。变量 <i>G32i_temperature_value_00</i> 对应于温度的实际值，变量 <i>G32i_temperature_maxvalue_00</i> 对应于温度的设定值，变量 <i>U08i_power_value_00</i> 对应于加热量。</p> <div><div> <i>G32i_temperature_value_00</i></div><div>Floating-point number 32-bit IEEE 754</div></div> <div><div> <i>G32i_temperature_maxvalue_00</i></div><div>Floating-point number 32-bit IEEE 754</div></div> <div><div> <i>U08i_power_value_00</i></div><div>Unsigned 8-bit value</div></div>



创建过程值归档

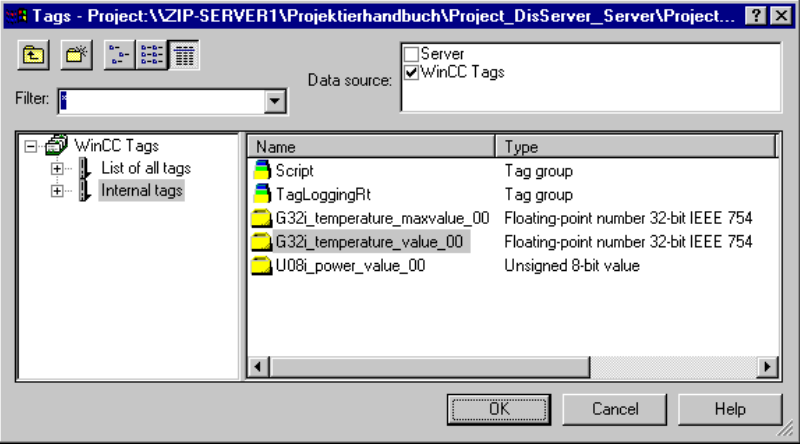

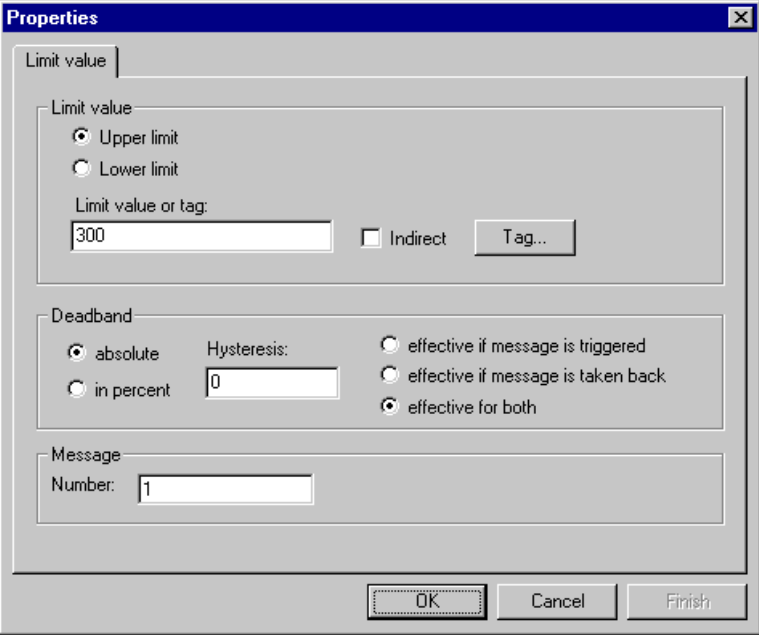
步骤	过程：创建过程值归档
1	打开变量记录编辑器。
2	<p>过程值归档的创建。为此通过  <i>归档</i> 来启动归档向导。</p> <div></div>

步骤	过程：创建过程值归档																
3	<p>在本实例中，归档已经命名为 <i>ProcessValueArchive_00</i>。 将变量 <i>G32i_temperature_value_00</i> 和 <i>G32i_temperature_maxvalue_00</i> 指定为归档变量。</p>  <table><tr><th>Archive name</th><th>Archive type</th></tr><tr><td>ProcessValueArchive_00</td><td>Process Value Archive</td></tr></table> <table><tr><th>Tag name</th><th>Tag type</th><th>Comments</th><th>Last change</th></tr><tr><td>G32i_temperature_maxvalue_00</td><td>Analog</td><td></td><td>05/11/99 05:55:22 PM</td></tr><tr><td>G32i_temperature_value_00</td><td>Analog</td><td></td><td>05/11/99 05:55:22 PM</td></tr></table> <div>Ready1 Archives.</div>	Archive name	Archive type	ProcessValueArchive_00	Process Value Archive	Tag name	Tag type	Comments	Last change	G32i_temperature_maxvalue_00	Analog		05/11/99 05:55:22 PM	G32i_temperature_value_00	Analog		05/11/99 05:55:22 PM
Archive name	Archive type																
ProcessValueArchive_00	Process Value Archive																
Tag name	Tag type	Comments	Last change														
G32i_temperature_maxvalue_00	Analog		05/11/99 05:55:22 PM														
G32i_temperature_value_00	Analog		05/11/99 05:55:22 PM														
4	保存后退出变量记录编辑器。																

组态报警记录

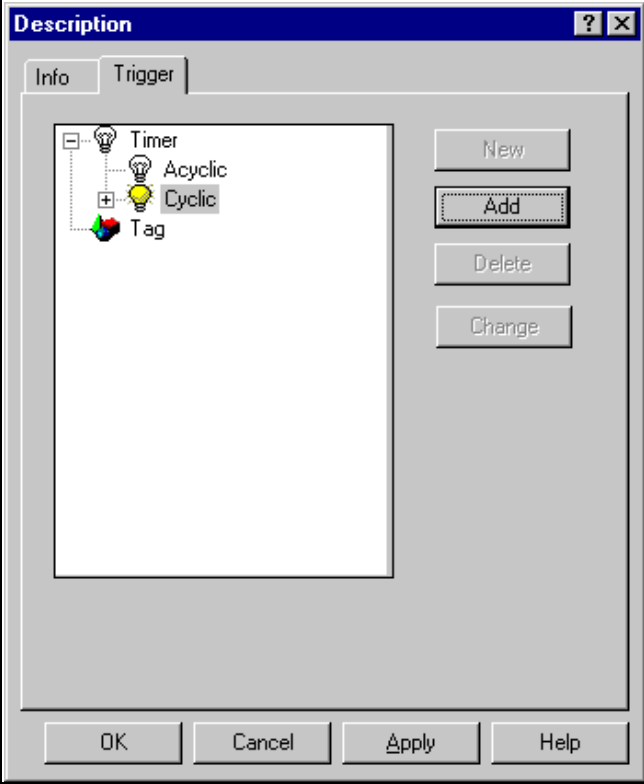
步骤	过程：组态报警记录																														
1	打开报警记录编辑器。																														
2	<p>单个消息的创建。在报警记录编辑器的底部窗口中，显示已经组态的消息。通过R，可添加新的行。在本实例中，需要两条不同的消息。必须对错误类型、消息文本和出错点进行相应修改。</p> <table border="1"> <tr> <th>...</th><th>Number</th><th>Class</th><th>Type</th><th>MessageTag</th><th>MessageBit</th><th>Status tag</th><th>Status bit</th><th>Message text</th><th>Point of error</th></tr> <tr> <td>▶</td><td>1</td><td>Error</td><td>Warning</td><td></td><td>0</td><td></td><td>0</td><td>normal</td><td>Oven</td></tr> <tr> <td></td><td>2</td><td>Error</td><td>Alarm</td><td></td><td>0</td><td></td><td>0</td><td>critical</td><td>Oven</td></tr> </table>	...	Number	Class	Type	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error	▶	1	Error	Warning		0		0	normal	Oven		2	Error	Alarm		0		0	critical	Oven
...	Number	Class	Type	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error																						
▶	1	Error	Warning		0		0	normal	Oven																						
	2	Error	Alarm		0		0	critical	Oven																						
3	<p>对限制值监控的组态如果限制值监控(模拟报警)条目没有显示，则必须首先将其装载。可通过报警记录中的选项 → 加载项菜单完成该操作。在所显示的对话框中，必须选择限制值监控(模拟报警)复选框。单击确定按钮关闭对话框。</p> <div> <div>Add Ins</div> <div>Select the Add Ins that your project requires from here:</div> <div>Add Ins:</div> <div> <input checked="" type="checkbox"/>Analog Alarm </div> <div>Description</div> <div></div> <div>This dialog box can also be displayed via the menu command "Extras" and the entry "Add Ins ..."</div> <div> <div>OK</div> <div>Cancel</div> </div> </div>																														

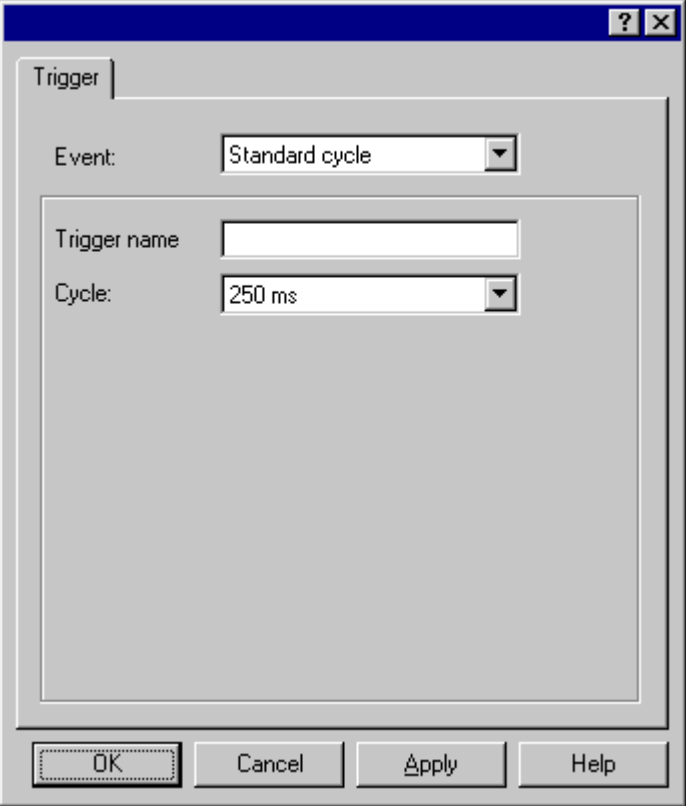
步骤	过程：组态报警记录		
4	<p>通过  限制值监控条目，然后选择 → 新建...来访问变量的属性对话框。在该对话框中，可设置一个用于限制值监控的新变量。</p> <div><div>Properties</div><div><div>Tag</div><div><div>Tag to be monitored:</div><div><input type="text"/></div><div></div></div><div><div>Number of limit values:</div><div><input type="text" value="0"/></div></div><div><div><input type="checkbox"/> a message for all limit values; message number:</div><div><input type="text" value="0"/></div></div><div><div>If selected, only one message can be used for all limit values of the tag. In runtime, one instance of this message will be created.</div></div><div><div>Delay</div><div><div>Delay time:</div><div><input type="text" value="0"/></div><div><div><input checked="" type="radio"/> Milliseconds</div><div><input type="radio"/> Minutes</div><div><input type="radio"/> Seconds</div><div><input type="radio"/> Hours</div></div></div><div><div>The shortest delay time is 250 milliseconds, the longest 24 hours. A delay time of 0 equals no delay time.</div></div><div><div>Please note that the settings made will be valid for all limit values of this tag.</div></div></div><div><div>OK</div><div>Cancel</div><div>Finish</div></div></div></div> <tr><td>5</td><td><p>通过下面所显示的按钮，访问选择变量对话框。</p><div><div>...</div></div></td></tr>	5	<p>通过下面所显示的按钮，访问选择变量对话框。</p> <div><div>...</div></div>
5	<p>通过下面所显示的按钮，访问选择变量对话框。</p> <div><div>...</div></div>		

步骤	过程：组态报警记录
6	<p>在左边的窗口中，选择 <i>内部变量</i> 条目。右边窗口将列出相应的变量。选择所期望的变量。在本实例中，它就是变量 <i>G32i_temperature_value_00</i>。</p> <p>按下 <i>确定</i> 按钮将关闭对话框。</p> 
7	<p>同样使用 <i>确定</i> 按钮来关闭变量的 <i>属性</i> 对话框。于是 <i>报警记录</i> 的右边窗口将显示要监控的新变量的图标。通过  <i>G32i_temperature_value_00</i> → <i>新建</i>，访问限制值的属性对话框。在该对话框中，可以将新的限制值分配给变量。在本实例中，将 <i>上限</i> 设置为 300，将消息编号设置为 1。</p> <p>按下 <i>确定</i> 按钮将关闭对话框。</p> 

步骤	过程：创建全局动作
8	采用先前所描述的步骤，将第二个限制值分配给变量。将上限设置为 700，将消息编号设置为 2。
9	保存后退出报警记录编辑器。

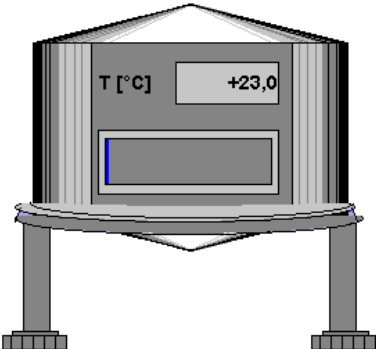
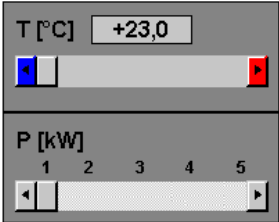
创建全局动作

步骤	过程：创建全局动作
1	打开全局脚本编辑器。
2	创建新的全局动作。这可通过全局脚本编辑器中的文件 → 新建动作菜单来完成。
3	在本实例中，已经编写了将 C 函数模拟为趋势的 C 动作。计算设定值温度 <i>dTemp2</i> 和实际温度 <i>dTemp1</i> 之间的差值 <i>dDelta</i> 。如果该差值为正，则趋势上升。如果该差值为负，则趋势下降。 加热量 <i>nPower</i> 定义温度以多快的速度达到设定值。
4	通过编辑 → 编译，对下列 C 动作进行编译。
5	通过编辑 → 信息，打开描述对话框。选择触发器标签。 在本实例中，选择了周期定时器。通过添加按钮，显示更改触发器的对话框。 

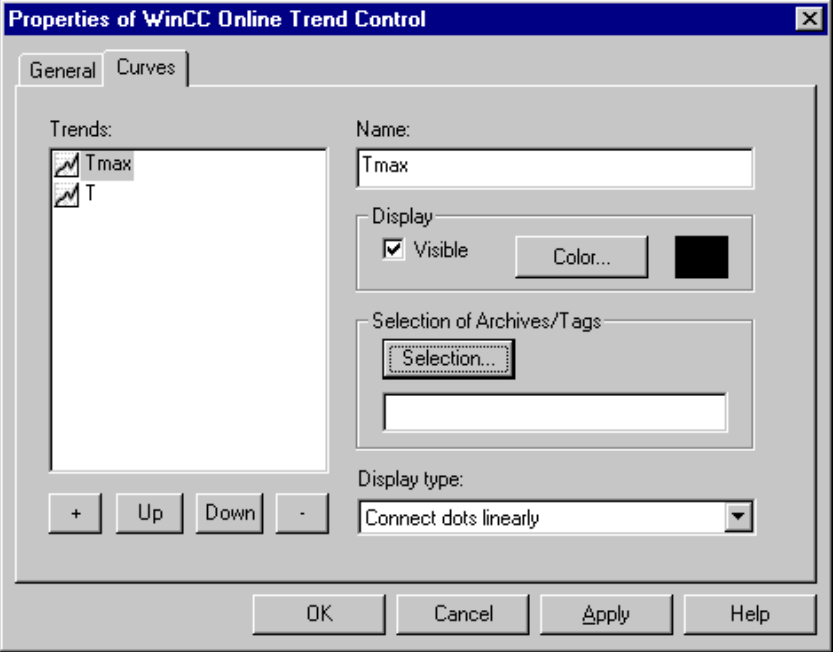
步骤	过程：创建全局动作
6	<p>将周期时间设置为 250 毫秒。 按下 确定 关闭两个对话框。</p> 
7	保存后退出全局脚本编辑器。

图形编辑器

步骤	过程：图形编辑器
1	在图形编辑器中创建一个新画面。在本实例中，它是画面 <i>dss_3_chapter_01.PDL</i> 。在本画面中，将各种不同的对象连接到过程变量上。
2	<p>可通过 <i>Windows 对象</i> → <i>滚动条对象</i> 实现输入变量的模拟。在本实例中，它们是 <i>滚动条对象 1</i> (<i>G32i_temperature_maxvalue_00</i>) 和 <i>滚动条对象 2</i> (<i>U08i_power_value_00</i>) 以及代表温度控制的 <i>I/O 域 1</i> (<i>G32i_temperature_maxvalue_00</i>)。在 <i>I/O 域 1</i> 中，显示设定的温度值，也可以在此处将其更改。</p> <p>在炉中显示输出变量(<i>G32i_temperature_value_00</i>)。它包括对象 <i>I/O 域 2</i> 和 <i>棒图 1</i>。</p> <p>将这些对象的更新设置为 <i>一旦改变</i>。</p>



组态趋势窗口

步骤	过程：组态趋势窗口
1	在图形编辑器中另外创建一个画面。在本实例中，它是画面 <i>dss_3_chapter_02.PDL</i> 。在该画面中，使用趋势窗口显示两个温度值。
2	<p>通过控件 → WinCC 在线趋势控件创建趋势控件。在实例中，这就是对象 <i>TlgOnlineTrend1</i>。显示对话框 <i>WinCC 在线趋势控件属性</i>。通过单击+按钮在趋势标签中添加一条新趋势。</p> <p>趋势 1 重命名为 <i>Tmax</i>，趋势 2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

3

显示 归档/变量选择对话框。该对话框允许选择归档/归档变量。

在本实例中，将 *Tmax* 与变量 *G32i_temperature_value_00* 相连接，将 *T* 与变量 *G32i_temperature_maxvalue* 相连接。

Selection of Archives/Tags

Hierarchy:

Server_TagLogging::

Project_DisServer_Client

Server_Data

Server_TagLogging

Server_AlarmLogging

Tag name	Tag type	Archiving type
G32i_temperature_maxvalue_00	Analog	Cyclic-continu
G32i_temperature_value_00	Analog	Cyclic-continu

Update

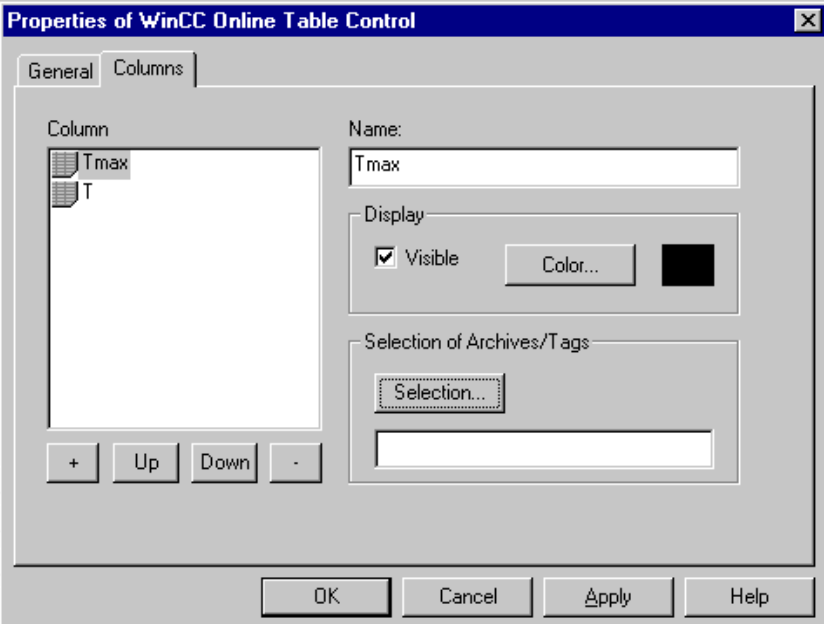
OK

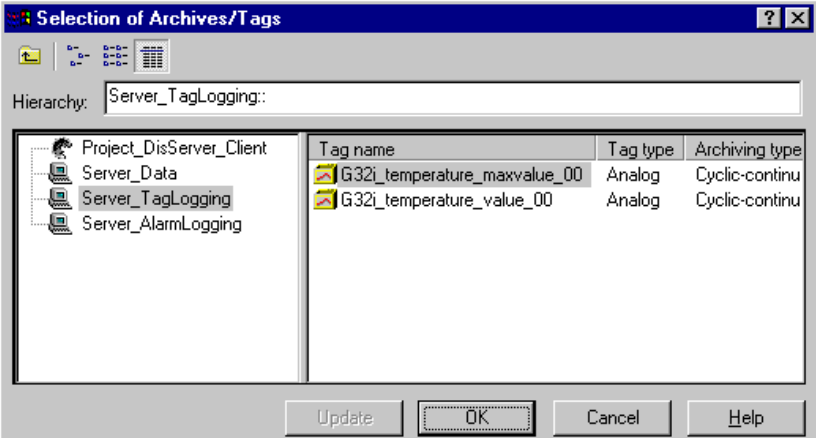
Cancel

Help

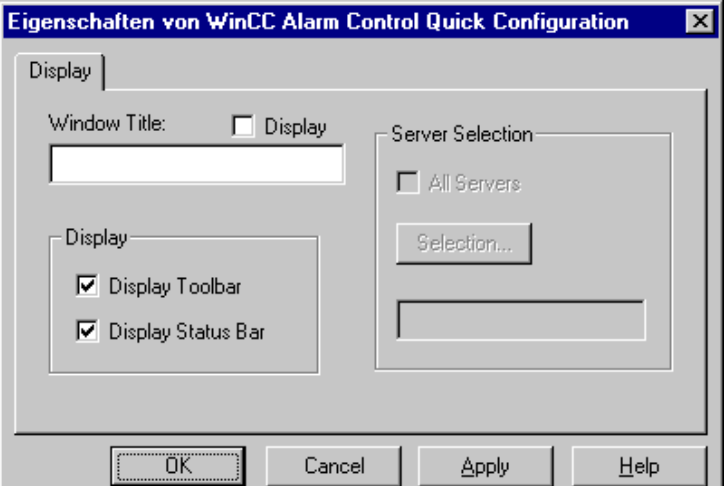
3-13

组态表格窗口

步骤	过程：组态表格窗口
1	在同一画面(<i>dss_3_chapter_02.PDL</i>)中，使用表格窗口显示两个温度值。
2	<p>通过控件 → <i>WinCC 在线表格控件</i>，创建表格控件。在本实例中，这就是对象 <i>TlgOnlineTable1</i>。显示对话框 <i>WinCC 在线表格控件属性</i>。在列标签内，通过单击+按钮添加新列。</p> <p>列 1 重命名为 <i>Tmax</i>，列 2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

步骤	过程：组态表格窗口
3	<p>显示 归档/变量选择对话框。该对话框允许选择归档/归档变量。</p> <p>在本实例中，将 <i>Tmax</i> 与变量 <i>G32i_temperature_value_00</i> 相连接，将 <i>T</i> 与变量 <i>G32i_temperature_maxvalue</i> 相连接。</p> 



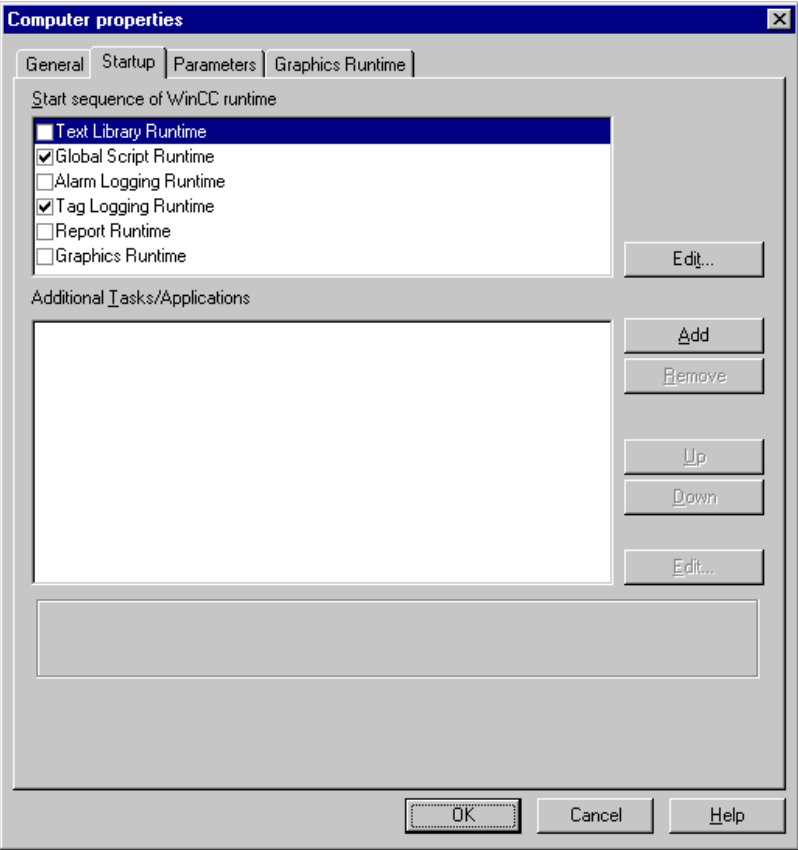
组态消息窗口

步骤	过程：组态消息窗口
1	<p>在图形编辑器中另外创建一个画面。在该画面中，使用消息窗口输出所组态的消息。在本实例中，这就是 <i>dss_3_chapter_03.PDL</i> 画面。</p>
2	<p>通过 控件 → WinCC 报警控件，创建报警控件。在实例中，这就是对象 <i>CCAlgWinCtrl1</i>。显示对话框 <i>WinCC 报警控件属性 - 快速组态</i>。按下确定按钮关闭对话框。</p> 


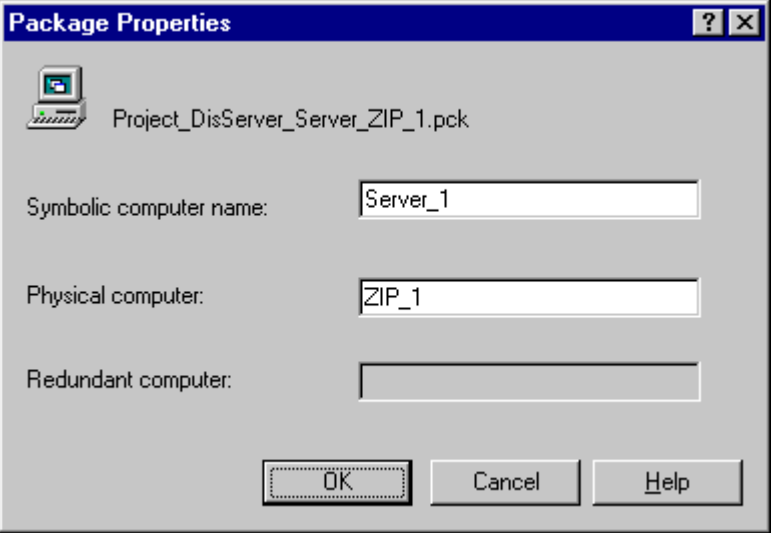
设置 WinCC 运行系统启动属性

前面已经指出，该服务器项目在三台计算机上运行，每台计算机只完成为其分配的功能，这样可减轻计算机的负载。为了实现该功能，必须对运行系统属性进行相应修改。

对保留了用于趋势和表格(变量记录)的归档的服务器，设置变量记录运行系统和全局脚本运行系统的属性：

步骤	过程：设置用于变量记录服务器的 WinCC 运行系统启动属性
1	通过  WinCC 资源管理器左边的 计算机 条目，计算机名称将显示在右边窗口中。
2	<p>通过  计算机名称 → 属性，显示对话框 计算机属性。在启动标签中，完成以下设置。单击 确定 按钮关闭对话框。</p> 
3	按照上面的步骤，为报警记录服务器设置报警记录运行系统和全局脚本运行系统属性，为数据服务器设置全局脚本运行系统和图形运行系统属性。

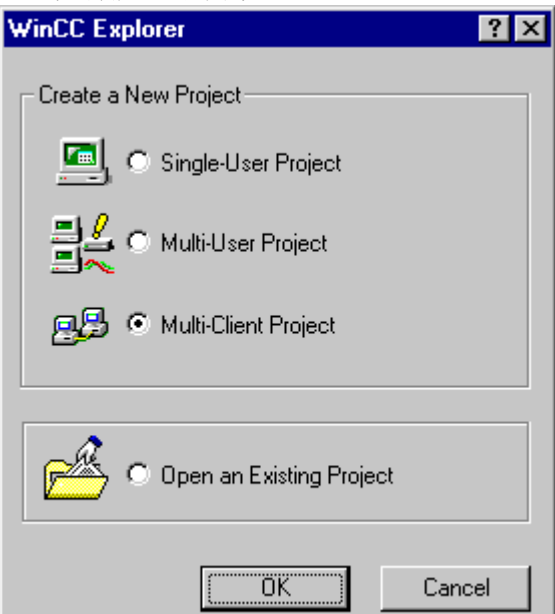
生成服务器数据

步骤	过程：生成用于变量记录服务器的服务器数据
1	<p>通过 WinCC 资源管理器左边窗口中的 <i>服务器数据</i> → 生成服务器数据。将显示一消息，说明服务器数据已成功生成。</p> <p>该对话框用确定按钮来确认。随后，所生成的程序包将显示在 WinCC 资源管理器的右边窗口中。</p> 
2	<p>通过 WinCC 新生成的程序包和属性，显示对话框 <i>程序包属性</i>。</p> <p>在本实例中，将符号计算机名称重新命名为 <i>Server_TagLogging</i>。</p> <p>按下确定按钮将关闭对话框。</p> 
3	<p>按照上面的步骤，生成其余两个服务器上的服务器数据。在报警记录服务器上，将符号计算机名称重新命名为 <i>Server_AlarmLogging</i>，而在数据服务器上，则命名为 <i>Server_Data</i>。</p>


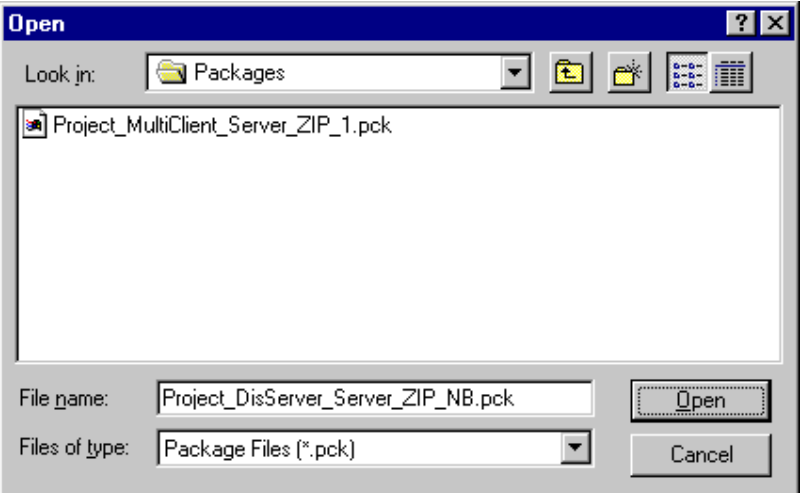
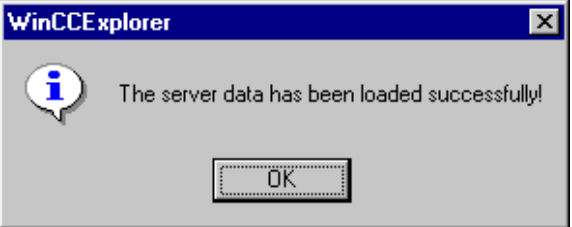
3.3 Project_DisServer_Client 项目的创建

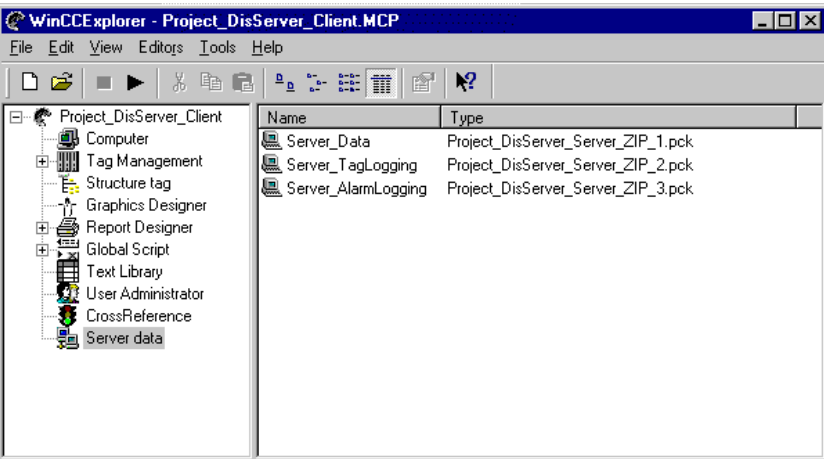
以下详细描述了创建多客户机项目 *Project_DisServer_Client* 所需的步骤。
该项目引用了先前所组态的三台服务器 *Server_Data*、*Server_TagLogging* 和 *Server_AlarmLogging* 的数据。

创建多客户机项目

步骤	过程：创建多客户机项目
1	创建新的 WinCC 项目。 通过 开始 → <i>Simatic</i> → <i>WinCC</i> → 视窗控制中心来启动 WinCC 资源管理器。
2	将显示 <i>WinCC 资源管理器</i> 。 通过菜单 文件 → 新建，将打开用来指定新的 WinCC 项目属性的对话框。 将本实例项目创建为多客户机项目。 通过单击 确定退出对话框。 

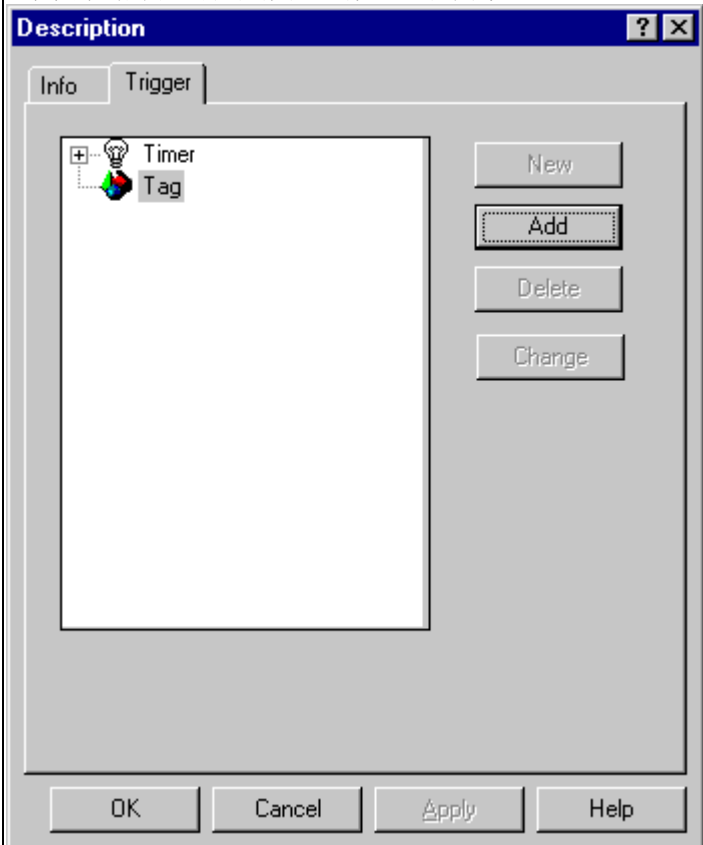


装载服务器数据

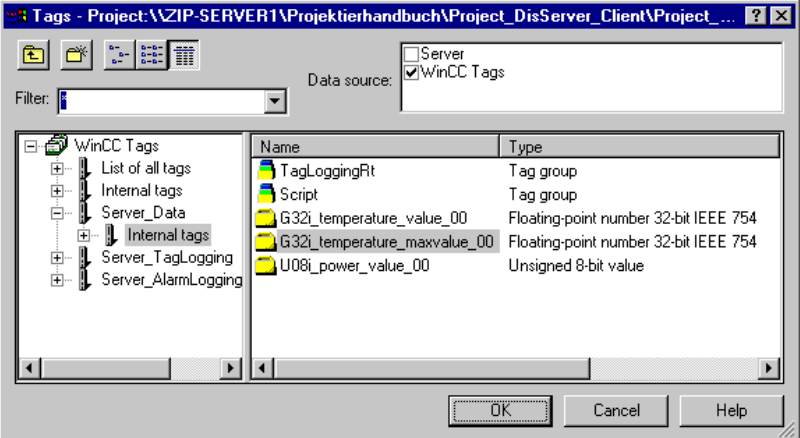
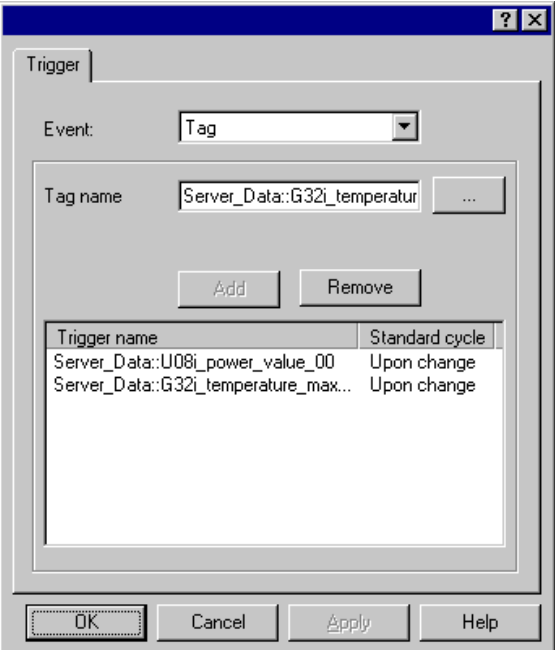
步骤	过程：装载服务器数据
1	通过  WinCC 资源管理器左边窗口上的服务器数据 → 装载，将显示“打开”对话框。
2	<p>从网上邻居中，选择服务器计算机。数据包文件位于服务器上的下列文件夹中：</p> <p>项目名称 → 变量记录服务器的名称 → 数据包</p> <p>选择该文件，并通过打开按钮装载该文件。</p> 
3	<p>显示一个对话框，对服务器数据的成功装载进行确认。使用确定按钮，确认该对话框。</p> 

步骤	过程：装载服务器数据
4	<div><p>按照刚才所描述的步骤，装载其余两个服务器的数据包文件。 所装载的数据包将显示在 WinCC 资源管理器的右边窗口中。</p></div>

创建全局动作

步骤	过程：创建全局动作
1	打开全局脚本编辑器。
2	创建新的全局动作。这可通过全局脚本编辑器中的文件 → 新建动作菜单来完成。
3	在实例中，已编制了下列 C 动作。一旦改变，该动作就把输入值(温度设定值和热容量)传送给三台服务器。
4	通过编辑 → 编译来编译 C 动作。

步骤	过程：创建全局动作
5	<p>通过 编辑 → 信息，打开描述对话框。选择触发器标签。</p> <p>在本实例中，根据两个输入值的变化对触发器进行设置。通过添加按钮，访问可在其中对于此目的变量进行组态的对话框。</p> 
6	<p>通过  下面所显示的按钮，访问选择变量对话框。</p> 

步骤	过程：创建全局动作
7	<div><p>显示一个可从中选择变量的窗口。在本实例中，选择了 <i>G32i_temperature_maxvalue_00</i> 变量，它来自于 <i>Server_Data</i> 的内部变量。单击确定按钮关闭对话框。</p></div>
8	<div><p>按照步骤 6 和步骤 7，添加 <i>U08i_power_value_00</i> 变量，该变量来自于 <i>Server_Data</i> 的内部变量。</p></div>
9	<div><p>按照上述步骤，再显示可在其中选择标准周期的对话框。在本实例中，将两个变量的标准周期都设置为 <i>一旦改变</i>。</p></div>
10	<div><p>使用确定按钮，关闭对话框。 保存后退出全局脚本编辑器。</p></div>

C 动作

```
#include "apdefap.h"

int gscAction( void )
{
    double dTemp;
    BYTE nPower;

    //Get values from Server_Data
    dTemp = GetTagDouble("Server_Data::G32i_temperature_maxvalue_00");
    nPower = GetTagByte("Server_Data::U08i_power_value_00");


    //Set values on Server_TagLogging
    SetTagDouble("Server_TagLogging::G32i_temperature_maxvalue_00", dTemp);
    SetTagByte("Server_TagLogging::U08i_power_value_00", nPower);

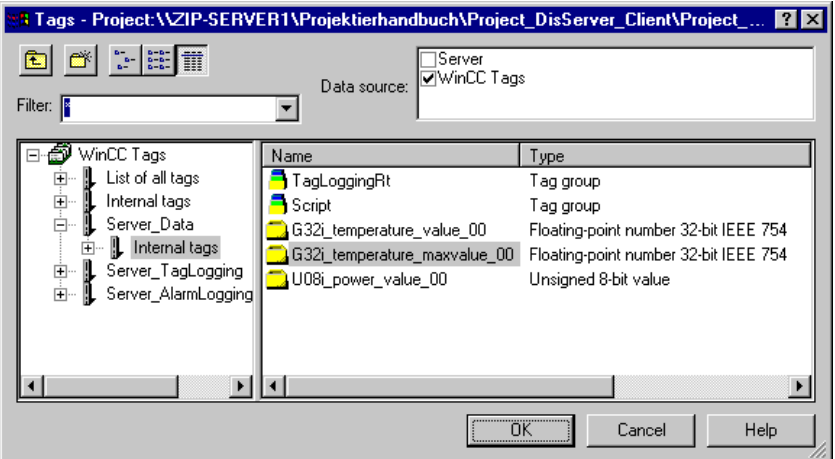
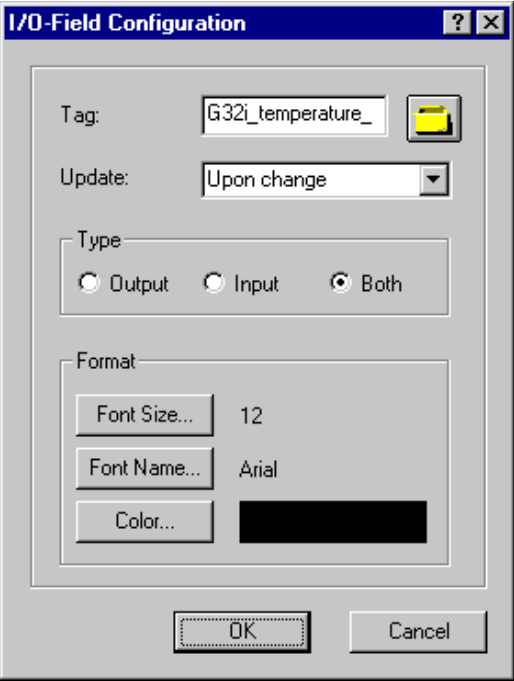
    //Set values on Server_AlarmLogging
    SetTagDouble("Server_AlarmLogging::G32i_temperature_maxvalue_00", dTemp);
    SetTagByte("Server_AlarmLogging::U08i_power_value_00", nPower);
    return 0;
}
```

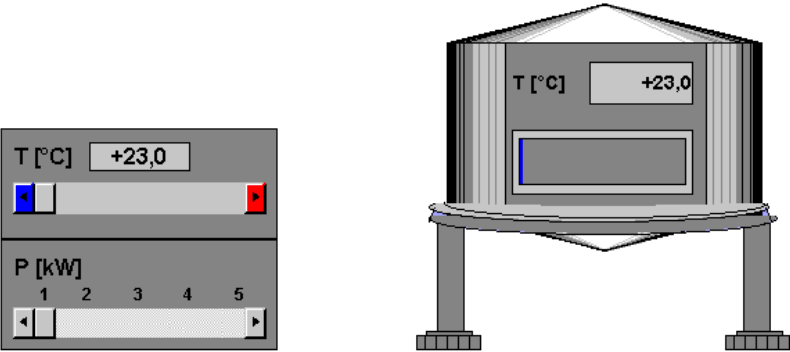
图形编辑器

在多客户机项目中，没有创建任何变量，也就是说，它使用服务器的变量进行工作。趋势和表格窗口与变量记录服务器上的归档变量相连，而消息窗口则使用报警记录服务器上的变量进行工作。其余对象(I/O 域、滚动条对象等)则与数据服务器上的变量相连。

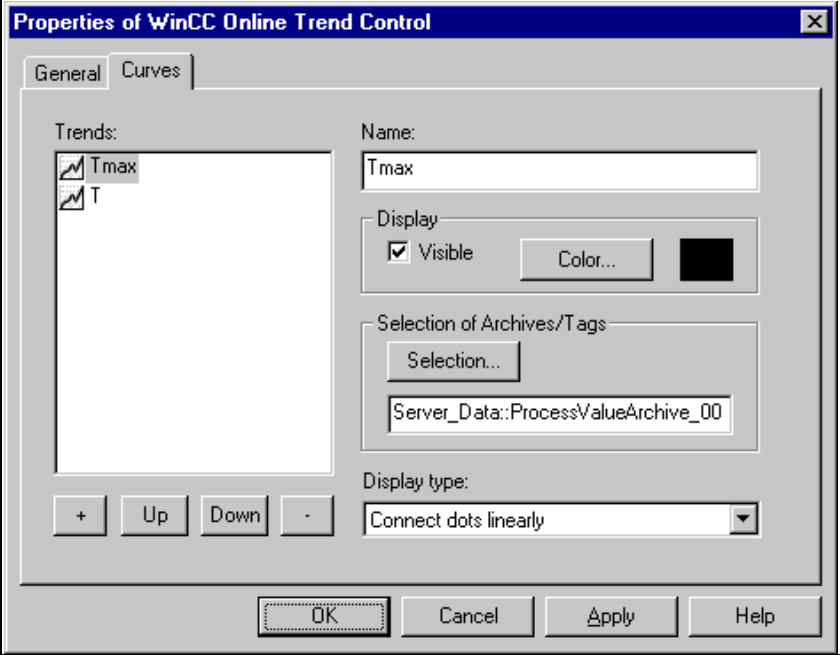
对象的组态

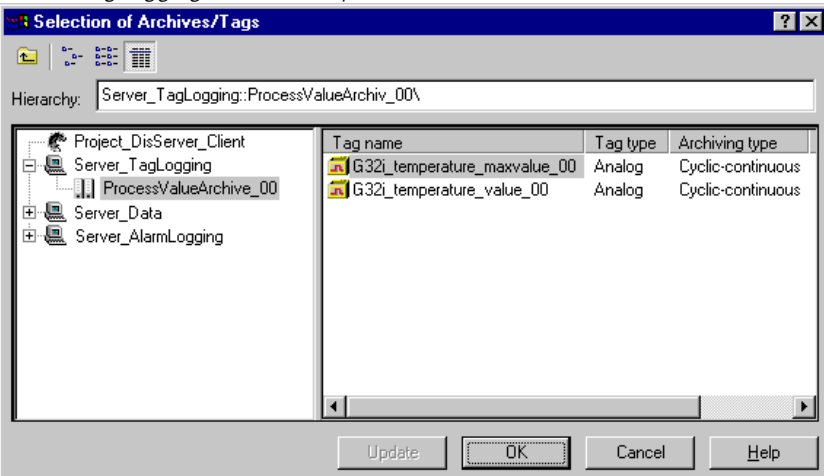
步骤	过程：对象的组态
1	在图形编辑器中创建一个新画面。在本实例中，它是画面 <i>dsc_3_chapter_01.PDL</i> 。在该画面中，各种不同的对象均与 <i>Server_Data</i> 的过程变量相连。
2	组态一个智能对象 → I/O 域。在实例中，它是对象 <i>I/O 域 1</i> 。将显示其组态对话框。  下面所显示的按钮，访问选择变量对话框。

步骤	过程：对象的组态
3	<p>在左边窗口中，选择所期望服务器的内部变量条目。右边的窗口将列出相应的变量。选择所期望的变量。在实例中，这就是 <i>Server_Data</i> 的 <i>G32i_temperature_maxvalue_00</i> 变量。</p> <p>按下确定按钮将关闭对话框。</p> 
4	<p>将更新设置为一旦改变。通过单击确定按钮可以退出组态对话框。</p> 

步骤	过程：对象的组态
5	<p data-bbox="521 296 1282 327">组态附加对象(I/O 域、滚动条对象、棒图)，以便显示服务器的其余变量。</p> <div data-bbox="548 346 1334 695"></div>

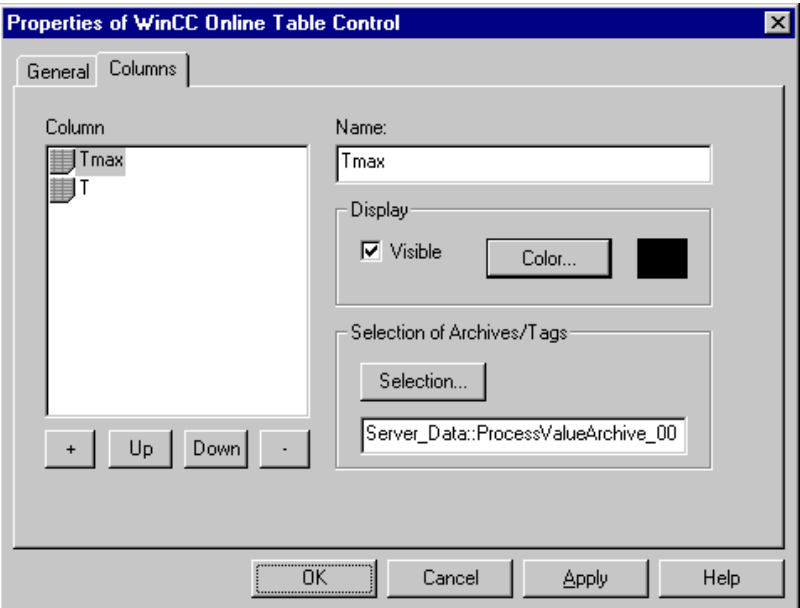
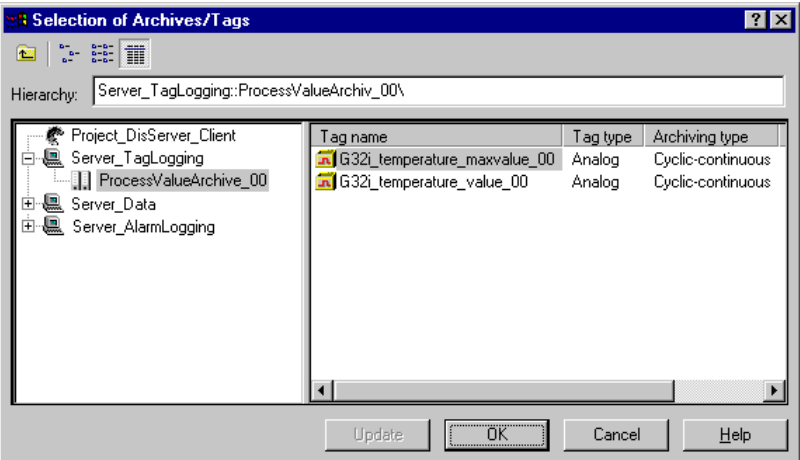
组态趋势窗口

步骤	过程：组态趋势窗口
1	在图形编辑器中另外创建一个画面。在本实例中，它是画面 <i>dsc_3_chapter_02.PDL</i> 。在该画面中，使用趋势窗口显示 <i>Sever_TagLogging</i> 的两个温度值。
2	<p>通过控件 → WinCC 在线趋势控件创建趋势控件。在实例中，这就是对象 <i>TlgOnlineTrend1</i>。显示对话框 <i>WinCC 在线趋势控件属性</i>。通过单击+按钮在趋势标签中添加一条新趋势。</p> <p>趋势 1 重命名为 <i>Tmax</i>，趋势 2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

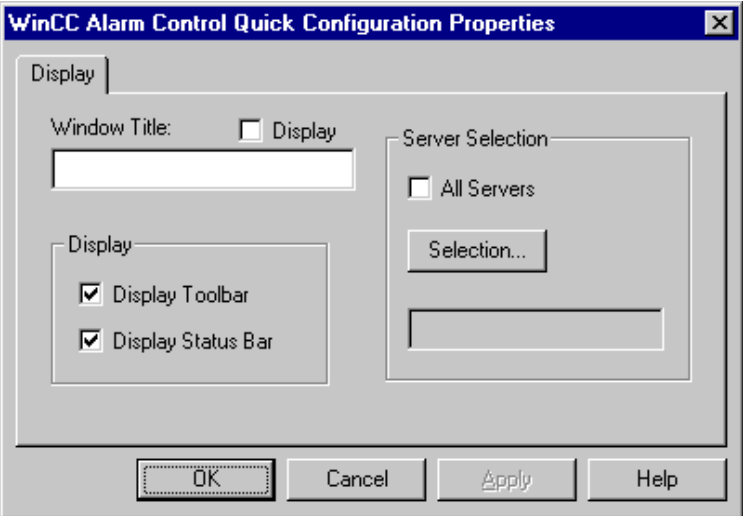
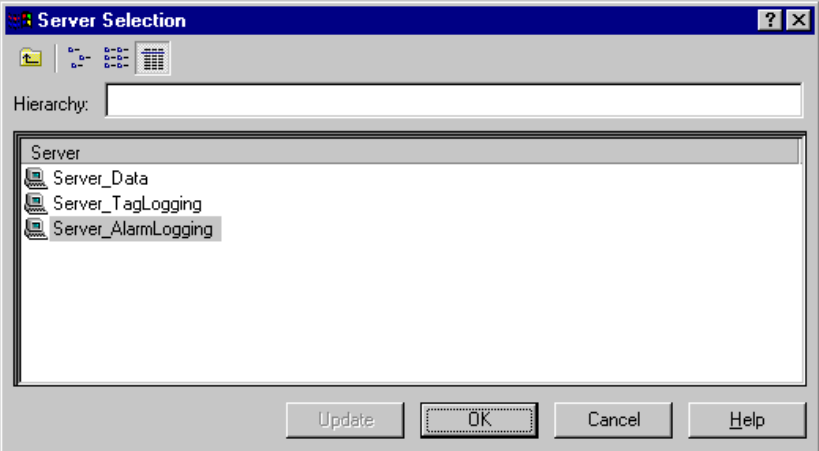
步骤	过程：组态趋势窗口
3	<p>显示 归档/变量选择对话框。从该对话框中，可从数据包所导入的服务器数据里选择服务器/归档/归档变量。</p> <p>在本实例中，将 <i>Tmax</i> 与 <i>G32i_temperature_value_00</i> 变量相连接，将 <i>T</i> 与 <i>Server_TagLogging</i> 的 <i>G32i_temperature_maxvalue</i> 变量相连接。</p> 

组态表格窗口



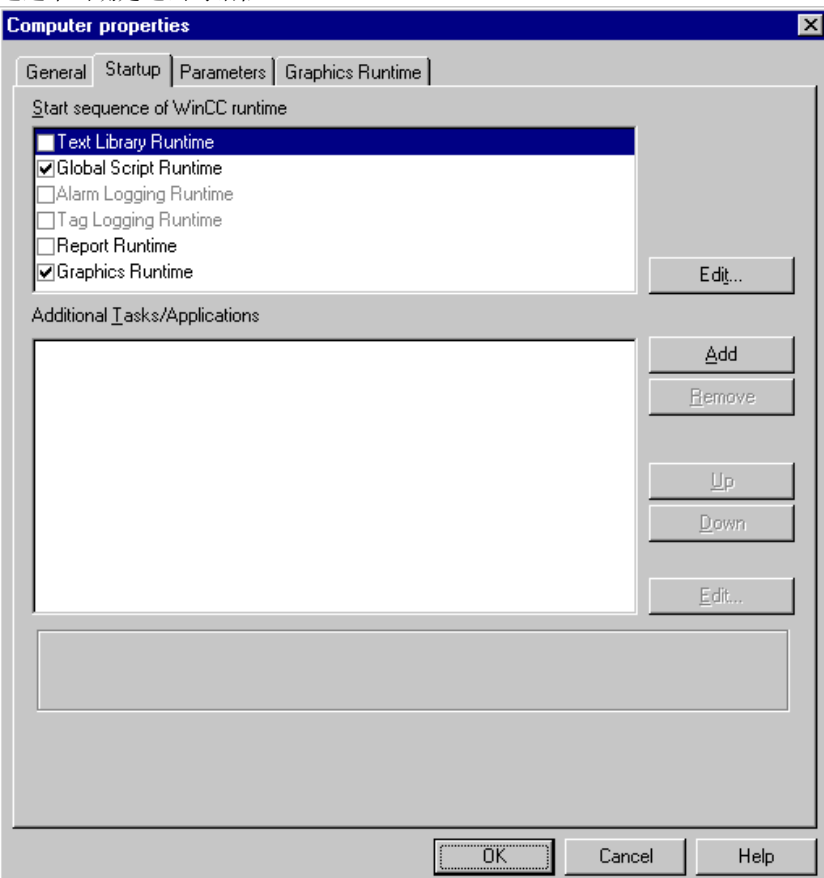
步骤	过程：组态表格窗口
1	<p>在同一画面(<i>dsc_3_chapter_02.PDL</i>)中，使用表格窗口显示 <i>Server_TagLogging</i> 的两个温度值。</p>

步骤	过程：组态表格窗口
2	<p>通过 控件 → WinCC 在线表格控件，创建表格控件。在本实例中，这就是对象 TlgOnlineTable1。显示对话框 WinCC 在线表格控件属性。在列标签内，通过单击+按钮添加新列。</p> <p>列 1 重命名为 Tmax，列 2 重命名为 T。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 
3	<p>显示 归档/变量选择对话框。在该对话框中，可从由数据包所导入的服务器数据里选择服务器/归档/归档变量。</p> <p>在本实例中，将 Tmax 与 G32i_temperature_value_00 变量相连接，将 T 与 Server_TagLogging 的 G32i_temperature_maxvalue 变量相连接。</p> 

组态消息窗口

步骤	过程：组态消息窗口
1	在图形编辑器中另外创建一个画面。在该画面中，使用消息窗口输出在 <i>Server_AlarmLogging</i> 上所组态的消息。在实例中，这就是 <i>dsc_3_chapter_03.PDL</i> 画面。
2	<p>通过控件 → WinCC 报警控件，创建报警控件。在实例中，这就是对象 CCAlgWinCtrl1。显示对话框 WinCC 报警控件属性-快速组态。通过选择按钮，访问服务器选择对话框。</p>  <p>The image shows the 'WinCC Alarm Control Quick Configuration Properties' dialog box. It has a 'Display' tab. Under 'Window Title', there is a checkbox for 'Display' and an empty text field. Under 'Server Selection', there is a checkbox for 'All Servers' and a 'Selection...' button. Below these are two checked checkboxes: 'Display Toolbar' and 'Display Status Bar'. At the bottom are 'OK', 'Cancel', 'Apply', and 'Help' buttons.</p>
3	<p>在本实例中，选择了 <i>Server_AlarmLogging</i>。然后，按下确定按钮，关闭对话框。</p>  <p>The image shows the 'Server Selection' dialog box. It has a 'Hierarchy' text field at the top. Below it is a list box titled 'Server' containing three items: 'Server_Data', 'Server_TagLogging', and 'Server_AlarmLogging'. At the bottom are 'Update', 'OK', 'Cancel', and 'Help' buttons.</p>

设置 WinCC 运行系统启动属性

步骤	过程: 设置 WinCC 运行系统启动属性
1	<p>通过  WinCC 资源管理器左边的计算机条目, 计算机名称将显示在右窗口中。通过  计算机名称 → 属性, 显示计算机属性对话框。在启动标签中, 完成以下设置。</p>
2	<p>在多客户机项目中, 设置全局脚本运行系统和图形运行系统属性。用于报警记录运行系统和变量记录运行系统的属性是灰色的, 不能对其进行设置。通过单击确定退出对话框。</p> 

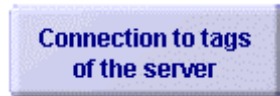
3.4 WinCC 项目的描述

激活所有三台服务器上的运行系统。在此之后，运行系统也可以在多客户机项目上激活。

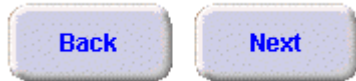
如果运行系统在多客户机项目上比在服务器项目上先激活，则会产生通讯问题，因为多客户机项目引用服务器的数据。

总览画面将显示在数据服务器和多客户机上。在其余两台服务器 (*Server_TagLogging* 和 *Server_AlarmLogging*) 上，未设置图形运行系统的属性。

3.4.1 服务器项目



在出现总览画面之后，可以通过如上所示的按钮访问设备画面。



通过如上所示的按钮，可在各画面之间进行切换。



通过此按钮，可返回至总览画面。

设备画面

在设备画面中，显示一个带温度控制的炉。通过该温度控制可以预先设置温度。炉温上升直至达到预设值。通过能量控制可以指定热容量。该值会影响炉温上升的速度。

趋势和表格窗口

在下一个画面中，显示趋势和表格窗口。趋势窗口描绘预置温度(设定值)以及炉温(实际值)的进程。两种数值在表格窗口中也都能显示。

消息窗口

下一个画面显示消息窗口。如果炉温超过数值 300，则产生一个警告信息并显示在消息窗口中。如果温度值超过 700，则产生一个报警信息并显示在消息窗口中。

3.4.2 客户机项目

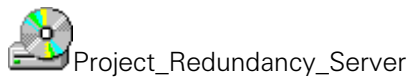
与三台服务器的连接



在总览画面出现之后，可通过上面所显示的按钮可访问画面，在该画面中组态了多客户机上的自身对象。这些对象已与各种服务器过程变量相连接。
在第一个画面中，显示设备画面。在随后的画面中，显示了趋势、表格和消息窗口。

4 冗余

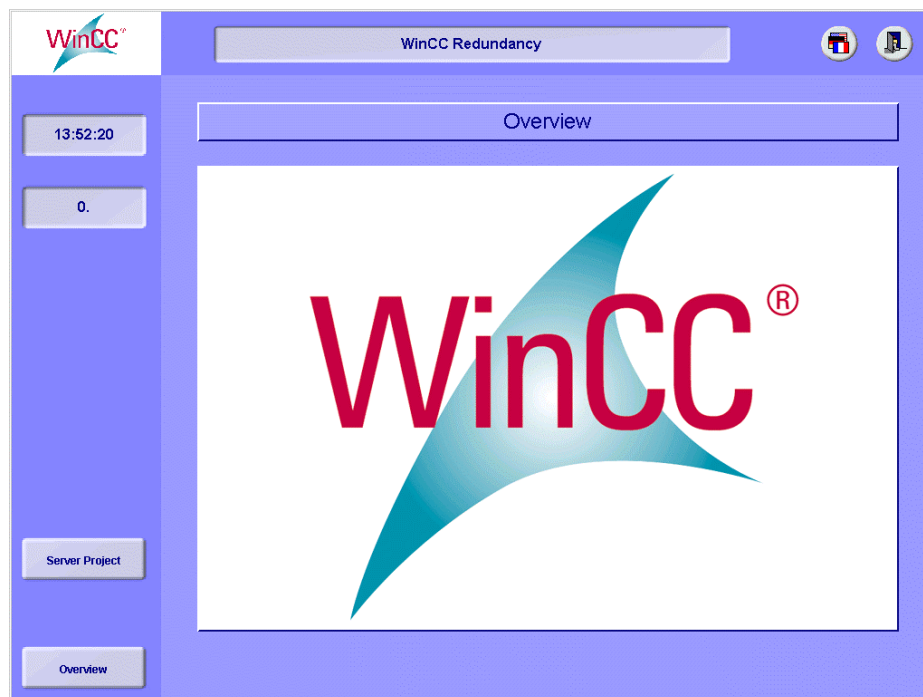
本章中创建的项目也可以直接从在线文档复制到硬盘驱动器上。缺省情况下，它将存储在 *C:\Configuration_Manual* 文件夹中。可以选择将以下组件复制到硬盘驱动器上：



将要创建的 WinCC 项目。

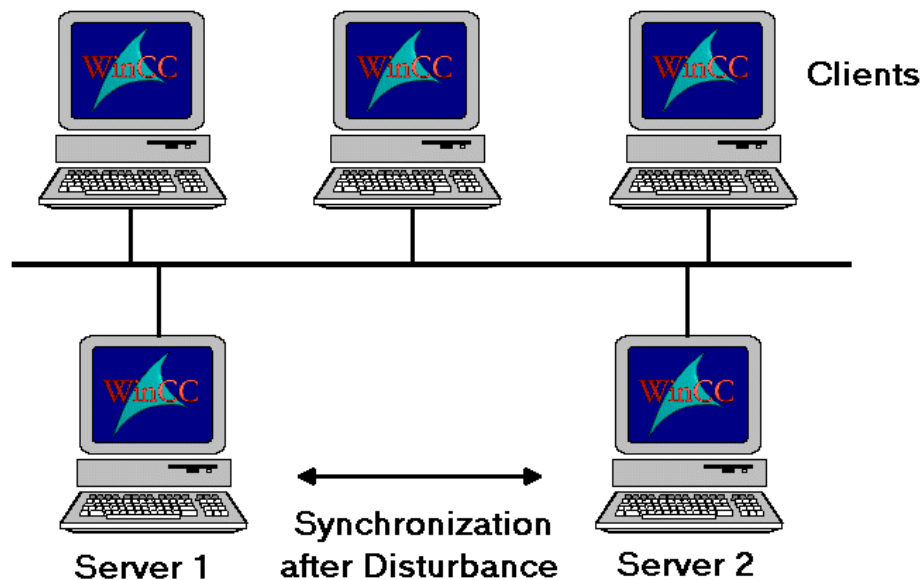
在本节中，将给出一个与冗余相关的实例。

有关该主题的实例组态在 Project_Redundancy_ServerWinCC 项目中。



4.1 常规信息

通过运行两台相互并行连接的服务器 PC，WinCC 冗余可显著提高 WinCC 和设备的利用率。



为了提早识别伙伴的故障，服务器在运行系统中互相监控。

如果两台服务器计算机中有一个出现故障，则客户机将自动从出现故障的服务器切换到另一个仍正常工作的服务器上。因此，所有客户机将仍然可以对过程进行监控和操作。

在出现故障期间，仍在运行的服务器将继续对 WinCC 项目的所有消息和过程数据进行归档。当发生过故障的服务器在线返回后，所有消息的内容、过程值以及用户归档将自动复制给已返回的服务器。这样，可将发生故障的服务器上的归档数据缺口填上。该过程也可称之为同步。

WinCC 冗余选项提供：

- 在发生故障的服务器返回之后，消息、过程值以及用户归档将自动同步。
- 在过程连接错误已更正之后，消息和过程值归档将自动同步。
- 消息归档的在线同步在一定数值范围(本地服务器的消息)内进行归档。
- 用户归档在线同步。
- 在冗余服务器之间客户机自动或者手动切换的项目切换器。
- 将项目复制到冗余伙伴服务器的项目复制器。

4.1.1 冗余的操作

正常操作时的 WinCC 归档

正常操作时，在运行中过程数据服务器全部并行运行。每个服务器站都具有单独的过程连接及其自己的数据归档。将 PLC 的过程数据和消息发送给两台冗余服务器，并由它们进行相应的处理。

服务器在运行中相互进行监控，以便提早识别伙伴的故障，并发出过程控制消息。

在一定编号范围内的用户归档和消息可以连续在线同步。

两个服务器具有同等的权限、工作中相互独立并且均可供用户使用。如果有一个服务器发生故障，总是会有一个相同的冗余服务器可用。

为了达到设备状态监控和归档同步的目的，冗余服务器站之间通过终端总线进行通讯。对于网络，使用具有 TCP/IP 或 NetBEUI 协议的 PC LAN。

服务器故障

如果服务器中有一台发生故障，则另一台仍在工作的服务器将继续接收和归档来自 PLC 的过程值与消息。这样就可以保证数据完整性。

客户机自动从出现故障的服务器切换到冗余伙伴服务器上。在极短的切换时间之后，所有操作站将可以继续使用。

触发客户机切换的因素

在服务器出现故障期间，系统将自动执行客户机从标准服务器到伙伴服务器的切换。下列因素均可触发客户机的切换：

- 与服务器的网络连接出错。
- 服务器故障

返回后触发归档同步的因素

一旦纠正了下列错误，就触发服务器之间的归档同步。

- 过程连接出错。可关闭过程连接监控。
- 与伙伴服务器的网络连接出错。
- 服务器故障
- 项目未激活。
- 项目未打开。

返回后同步

在出现故障的服务器返回在线之后，冗余服务器将执行故障期间的归档同步。通过将丢失的数据传送至出现故障的服务器，消除故障所引起的归档差别。这样，两个同样的服务器又可使用。

实现消息归档、过程值归档以及用户归档的同步。在故障所引起的时间延迟之后，发生故障的服务器将接收其数据。

不同归档类型以下列顺序同步：

- 消息归档
- 过程值归档
- 用户归档
- 归档同步作为后台功能来实现，与 WinCC 的过程控制和归档同时运行。这样可以保证设备的连续控制和监控。

注意：

与冗余一起应用“存储”选项可能会引起下列问题：

如果在服务器发生故障期间，“存储”从第二台服务器中导出并删除数据，则不能再使该数据同步。

如果“存储”导出了尚未同步的故障期间的数据，则归档同步将不能消除所导出数据中的差别。

为了避免数据丢失，在归档同步期间不要激活“存储”选项。Nach Abschluß der Archivierung wird Storage" automatisch reaktiviert.

过程连接出错后的同步

如果运行期间一台服务器与一台或多台 PLC 之间产生网络错误，则在更正错误之后自动启动同步(如果已组态的话)。

在线同步(可选)

对于一定编号范围内的用户归档及报警记录消息，可执行服务器与服务器之间的直接同步(在线同步)。

注意：

对于报警记录的在线同步，必须在报警记录系统中组态短期归档。

过程数据和消息归档

在两个冗余服务器上必须组态功能完全相同的变量记录和报警记录。功能完全相同的组态意味着：

完全相同的归档，允许以附加测量点和归档的形式对其进行扩展。这些扩展将不会同步，而必须在伙伴服务器上对其进行手动更新。

WinCC 使下列归档同步：

- 位于硬盘上的归档，即过程值归档、压缩归档和消息归档。使短期以及顺序归档同步。
 - 然而，对主存储器归档不进行任何同步。
- 对于消息归档的在线同步，必须在报警记录系统中组态短期归档。

用户归档

对于用户归档，要求两台服务器上的结构相同。

- 对于要同步的用户归档的组态，在域/记录结构及其属性方面必须完全相同。

注意：

为了避免使用冗余服务器所带来的问题，应该使计算机上使用的时钟同步。

4.1.2 冗余用户归档

用户归档可以通过运算、独立程序、PLC 或者其它函数进行编辑。

同时编辑用户归档

在将记录并行地添加给相互冗余的用户归档时，必须注意下列问题：

- 由于运行系统的原因，记录的添加顺序可能发生变化。
- 即使在服务器返回后完成同步之前，也可能将附加的记录添加到先前发生故障的服务器中。
- 即使在线同步期间，也会占用一些时间，直到已在冗余归档中对记录实现了同步。

两台计算机上的归档组态必须完全相同。因此，应该使用项目复制器。

如果归档不相同，则显示下列系统消息：用于所有用户归档的同步未准备好。

唯一键

为了将一个归档的记录清除分配到冗余归档记录，需要唯一的操作键域。在该域中具有相同内容的记录可互相进行同步。该域必须包含唯一值的属性，以避免在同一归档中存在具有相同内容的 2 个记录。这可能要通过：

- 记录号码，这个号码总是记录的一部分，并且不能另外进行组态(记录编号总是唯一的)。如果使用这个记录编号，则应该没有任何其它的域包含有这个唯一值属性。
- 归档域，已为其分配了唯一值属性。如果使用这个域，而不是使用记录编号，则它必须是唯一包含唯一值属性的域。
- 例如：配方名称(文本类型)
- 配方编号(整型)
- 插入日期/创建日期(日期类型)

最后访问的域

在归档属性的组态期间必须选择这个域，因为时间标志作为同步标准使用。

在同步进行期间，带有较新时间标记的数据记录将覆盖较旧的记录，这样可使大多数当前的记录得到保持。在同时进行编辑时或在同步期间进行修改时，必须注意这点。

最后修改的日期由系统自动输入。在记录导入期间，csv 文件中所包含的修改日期将不作修改地接受。

	利用记录编号的同步	利用唯一值的同步
在服务器发生故障期间删除记录。	确实没有发生。	
在服务器发生故障期间编辑/添加记录。	在离线同步期间，将对在故障期间已经进行编辑或添加的所有数据记录进行同步。	
如果已编辑过记录，则进行在线同步。	确定	确定 不能对唯一键的域内容进行修改，即使仍然将新的记录内容添加至冗余归档中，旧的记录因为不能被识别而得到保留。
如果添加了记录，则进行在线同步。	确定 如果在服务器返回后已完成同步前，将记录分别添加到两个冗余归档上，则在冗余归档中可能存在已自动分配的记录编号。这将导致覆盖两条记录中较旧的一条记录。	确定
如果删除了记录，则进行在线同步。	确定 只有在通过记录编号进行同步，并且在 OLE 控制单元(OCX)内或者通过用户归档 API 函数进行了修改时，才能执行所删除记录的在线同步。只有那些在最后的访问域中具有比删除时间更旧时间的记录才能被删除。	不能使用。

注意：

不应将具有最后访问域(它位于当前系统时间值的上面)的值的那些记录添加给归档(例如，通过导入)：至到当前系统时间才能进行同步。

如果提供数据的服务器关机或者在完成全部记录的在线同步前出现故障，则在下次启动运行系统期间，每个归档只有最后 50 个记录可进行同步。

如果退出 WinCC 运行系统，并在 10 秒钟内重新启动(通常只对小项目才可能)，那么不将其识别为故障，且一旦返回则不执行同步。

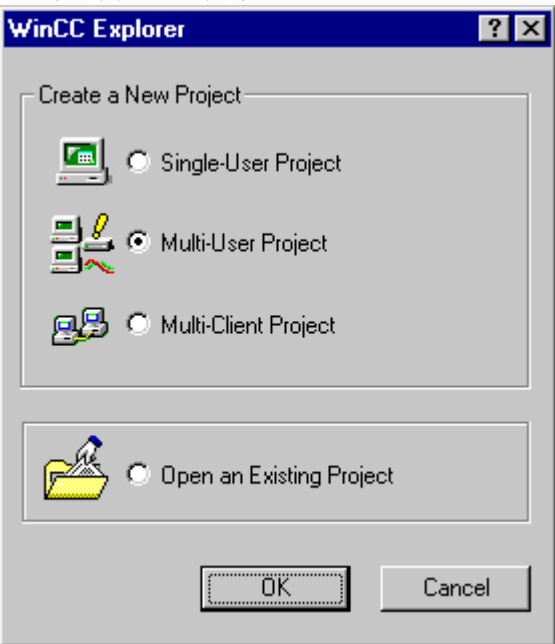
在发生连接错误的情况下，在线同步可将多达 10 条的记录存储到冗余服务器中，并在与伙伴重新建立连接之后，在实际同步激活之前，立即对其进行同步。

4.2 Project_Redundancy_Server 项目的创建










下面将详细描述创建 *Project_Redundancy_Server* 项目所必须的步骤。

本项目基于炉温控制的模拟，并在两台服务器上运行。在图形编辑器、变量记录、报警记录和全局脚本编辑器中进行组态。

创建服务器项目

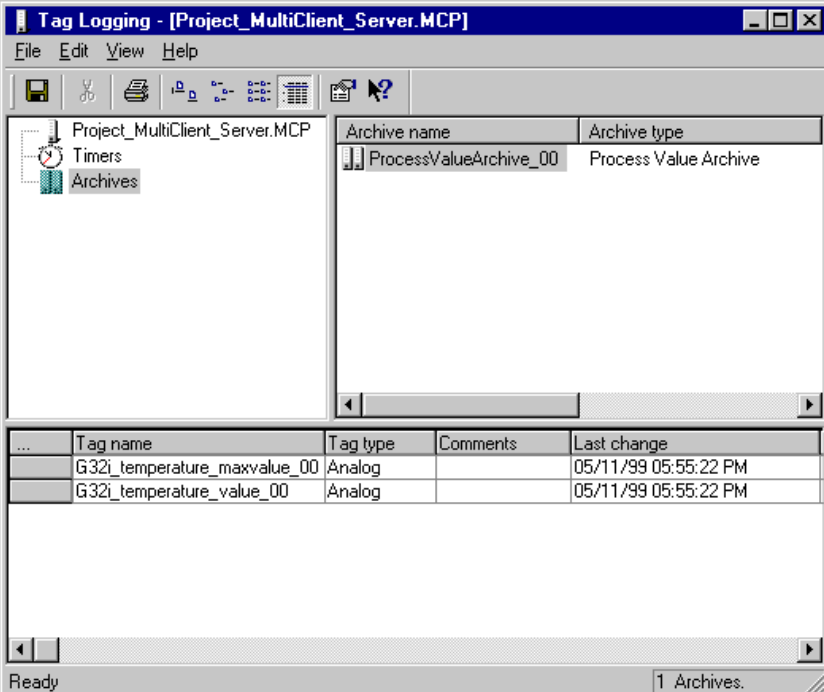
步骤	过程：创建服务器项目
1	创建新的 WinCC 项目。 通过开始 → <i>Simatic</i> → <i>WinCC</i> → <i>视窗控制中心</i> 来启动 WinCC 资源管理器。
2	将显示 <i>WinCC 资源管理器</i> 。 通过菜单文件 → 新建，将打开用于指定新 WinCC 项目属性的对话框。 将本实例项目创建为多用户项目。 通过单击确定退出对话框。 

创建变量


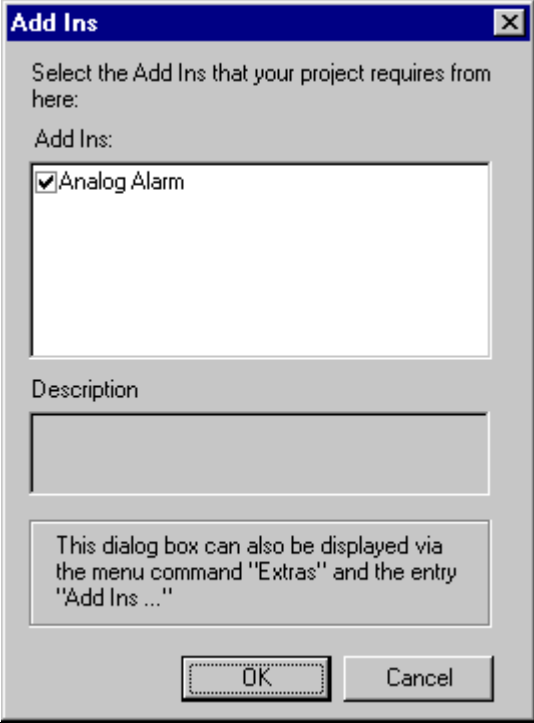
步骤	过程：创建变量									
1	<p>在本实例中，创建三个名称如下的内部变量：</p> <p>变量 <i>G32i_temperature_value_00</i> 对应于温度的实际值，变量 <i>G32i_temperature_maxvalue_00</i> 对应于温度的设定值，变量 <i>U08i_power_value_00</i> 对应于加热量。</p> <table><tr><td></td><td><i>G32i_temperature_value_00</i></td><td>Floating-point number 32-bit IEEE 754</td></tr><tr><td></td><td><i>G32i_temperature_maxvalue_00</i></td><td>Floating-point number 32-bit IEEE 754</td></tr><tr><td></td><td><i>U08i_power_value_00</i></td><td>Unsigned 8-bit value</td></tr></table>		<i>G32i_temperature_value_00</i>	Floating-point number 32-bit IEEE 754		<i>G32i_temperature_maxvalue_00</i>	Floating-point number 32-bit IEEE 754		<i>U08i_power_value_00</i>	Unsigned 8-bit value
	<i>G32i_temperature_value_00</i>	Floating-point number 32-bit IEEE 754								
	<i>G32i_temperature_maxvalue_00</i>	Floating-point number 32-bit IEEE 754								
	<i>U08i_power_value_00</i>	Unsigned 8-bit value								



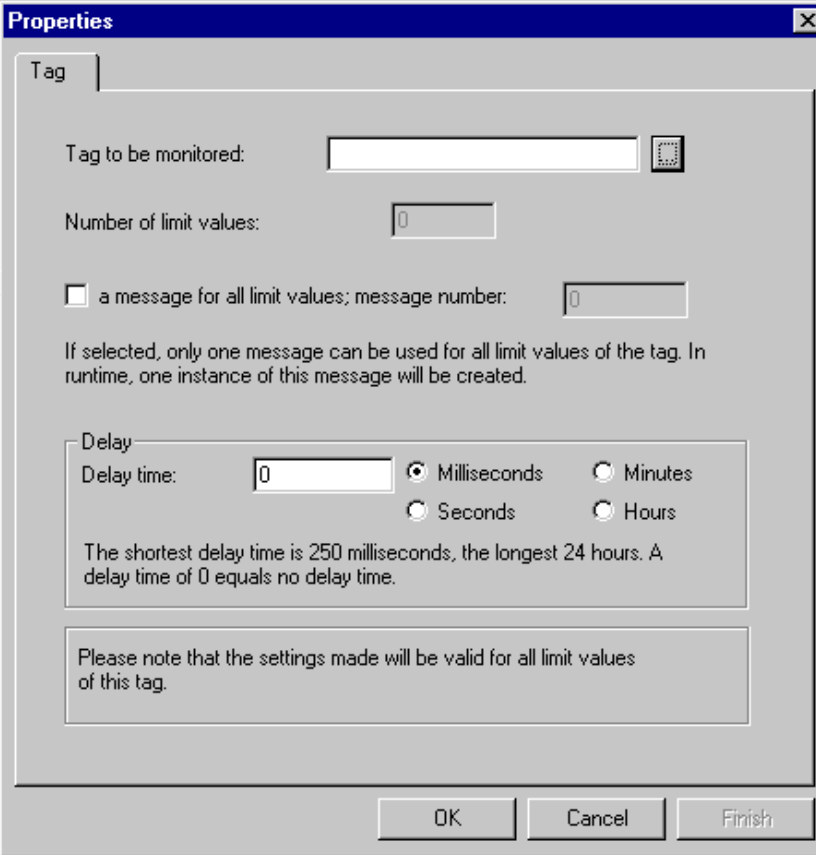


创建过程值归档

步骤	过程：创建过程值归档
1	打开 <i>变量记录编辑器</i> 。
2	<p>创建过程值归档。为此通过  <i>归档</i> 来启动归档向导。</p> 

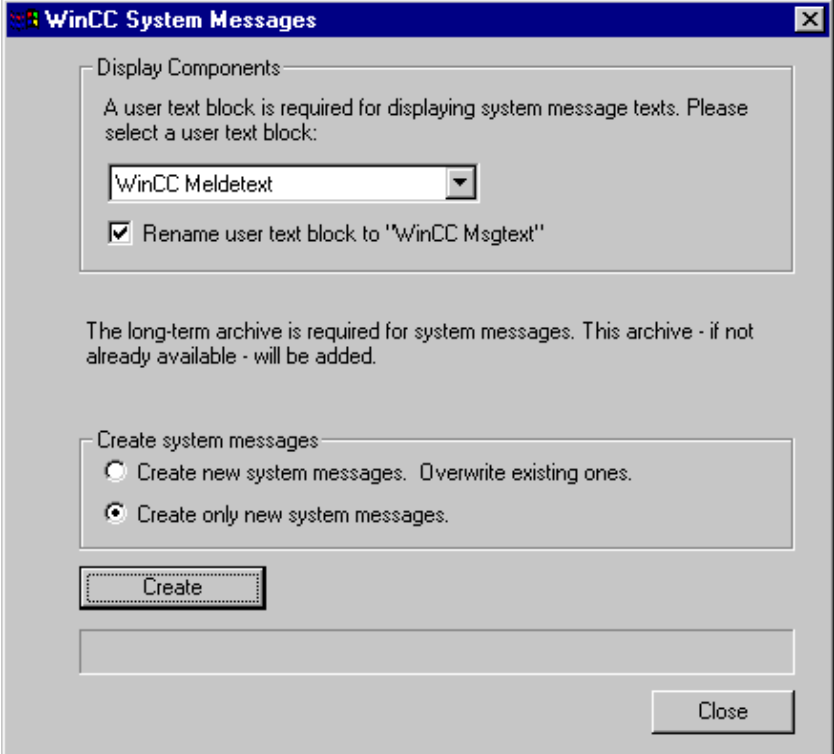
步骤	过程：创建过程值归档																
3	<p>在本实例中，归档已经命名为 <i>ProcessValueArchive_00</i>。 将变量 <i>G32i_temperature_value_00</i> 和 <i>G32i_temperature_maxvalue_00</i> 指定为归档变量。</p>  <table border="1"> <thead> <tr> <th>Archive name</th><th>Archive type</th></tr> </thead> <tbody> <tr> <td>ProcessValueArchive_00</td><td>Process Value Archive</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Tag name</th><th>Tag type</th><th>Comments</th><th>Last change</th></tr> </thead> <tbody> <tr> <td>G32i_temperature_maxvalue_00</td><td>Analog</td><td></td><td>05/11/99 05:55:22 PM</td></tr> <tr> <td>G32i_temperature_value_00</td><td>Analog</td><td></td><td>05/11/99 05:55:22 PM</td></tr> </tbody> </table> <p>Ready 1 Archives.</p>	Archive name	Archive type	ProcessValueArchive_00	Process Value Archive	Tag name	Tag type	Comments	Last change	G32i_temperature_maxvalue_00	Analog		05/11/99 05:55:22 PM	G32i_temperature_value_00	Analog		05/11/99 05:55:22 PM
Archive name	Archive type																
ProcessValueArchive_00	Process Value Archive																
Tag name	Tag type	Comments	Last change														
G32i_temperature_maxvalue_00	Analog		05/11/99 05:55:22 PM														
G32i_temperature_value_00	Analog		05/11/99 05:55:22 PM														
4	保存后退出变量记录编辑器。																


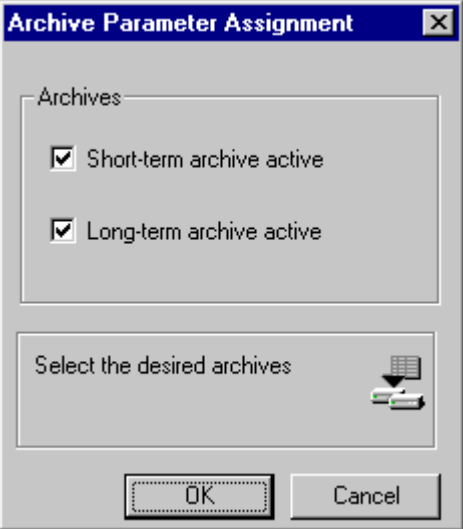
组态报警记录

步骤	过程：组态报警记录																														
1	打开报警记录编辑器。																														
2	<p>单个消息的创建。在报警记录编辑器的底部窗口中，显示已经组态的消息。通过R，可添加新的行。在本实例中，需要两条不同的消息。错误类型、消息文本和出错点必须进行相应的修改。</p> <table border="1"><thead><tr><th>...</th><th>Number</th><th>Class</th><th>Type</th><th>MessageTag</th><th>MessageBit</th><th>Status tag</th><th>Status bit</th><th>Message text</th><th>Point of error</th></tr></thead><tbody><tr><td>▶</td><td>1</td><td>Error</td><td>Warning</td><td></td><td>0</td><td></td><td>0</td><td>normal</td><td>Oven</td></tr><tr><td></td><td>2</td><td>Error</td><td>Alarm</td><td></td><td>0</td><td></td><td>0</td><td>critical</td><td>Oven</td></tr></tbody></table>	...	Number	Class	Type	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error	▶	1	Error	Warning		0		0	normal	Oven		2	Error	Alarm		0		0	critical	Oven
...	Number	Class	Type	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error																						
▶	1	Error	Warning		0		0	normal	Oven																						
	2	Error	Alarm		0		0	critical	Oven																						
3	<p>限制值监控的组态。如果限制值监控(模拟报警)条目没有显示，则首先必须将其装载。可通过报警记录中的选项 → 加载项菜单完成该操作。在所显示的对话框中，选择用于限制值监控(模拟报警)的复选框。单击确定按钮关闭对话框。</p> <div><p>Add Ins</p><p>Select the Add Ins that your project requires from here:</p><p>Add Ins:</p><p><input checked="" type="checkbox"/> Analog Alarm</p><p>Description</p><p>This dialog box can also be displayed via the menu command "Extras" and the entry "Add Ins ..."</p><p>OK Cancel</p></div>																														

步骤	过程：组态报警记录
4	<p>通过  限制值监控条目，然后选择  新建...来访问变量的属性对话框。在该对话框中，可设置一个用于限制值监控的新变量。</p>  <p>The image shows a 'Properties' dialog box for a 'Tag'. It has a 'Tag' tab. Inside, there's a 'Tag to be monitored:' field with a dropdown arrow. Below it is 'Number of limit values:' with a text box containing '0'. Then, a checkbox 'a message for all limit values; message number:' with a text box containing '0'. A note says: 'If selected, only one message can be used for all limit values of the tag. In runtime, one instance of this message will be created.' Below that is a 'Delay' section with 'Delay time:' text box containing '0', and four radio buttons: 'Milliseconds' (selected), 'Minutes', 'Seconds', and 'Hours'. A note states: 'The shortest delay time is 250 milliseconds, the longest 24 hours. A delay time of 0 equals no delay time.' At the bottom, a box says: 'Please note that the settings made will be valid for all limit values of this tag.' At the very bottom are 'OK', 'Cancel', and 'Finish' buttons.</p>
5	<p>通过使用  单击如下所示的按钮，打开选择变量对话框。</p> 

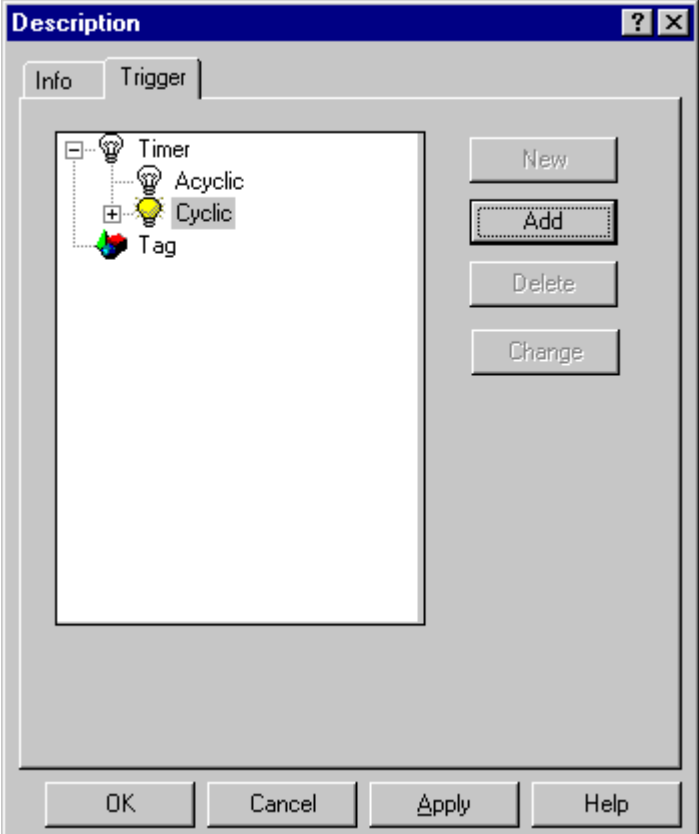
步骤	过程：组态报警记录
6	<p>在左边的窗口中，选择 <i>内部变量</i> 条目。右边的窗口将列出相应的变量。选择所期望的变量。在本实例中，它就是变量 <i>G32i_temperature_value_00</i>。</p> <p>按下 <i>确定</i> 按钮关闭对话框。</p>
7	<p>同样使用 <i>确定</i> 按钮关闭变量的 <i>属性</i> 对话框。于是 <i>报警记录</i> 的右边窗口将显示要监控的新变量的图标。通过 <i>G32i_temperature_value_00</i> → <i>新建</i>，访问限制值的 <i>属性</i> 对话框。在该对话框中，可以将新的限制值分配给变量。在本实例中，将 <i>上限</i> 设置为 <i>300</i>，将消息编号设置为 <i>1</i>。</p> <p>按下 <i>确定</i> 按钮关闭对话框。</p>

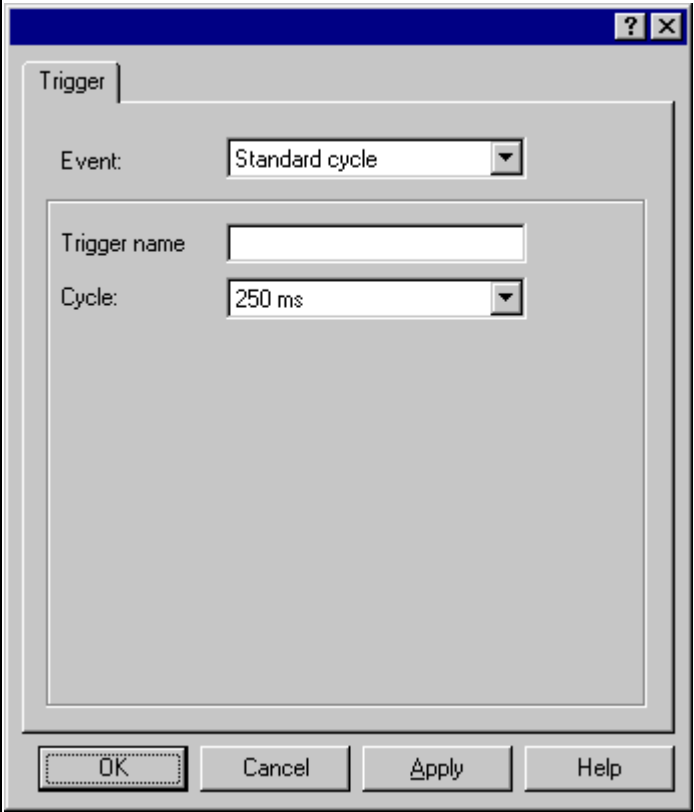
步骤	过程：组态报警记录
8	按照先前所描述的步骤，将第二个限制值分配给变量。将上限设置为 700，将消息编号设置为 2。
9	<p>创建 WinCC 系统消息。通过选项 → WinCC 系统消息菜单，访问 WinCC 系统消息对话框。通过创建按钮，将生成这些 WinCC 系统消息。使用确定按钮关闭对话框。</p>  <p>The image shows the 'WinCC System Messages' dialog box. It has a title bar with the WinCC logo and the text 'WinCC System Messages'. Inside, there's a section titled 'Display Components' with a text block selection dropdown set to 'WinCC Meldetext' and a checked checkbox 'Rename user text block to "WinCC Msgtext"'. Below this is a note about the long-term archive. Then, a 'Create system messages' section has two radio buttons: 'Create new system messages. Overwrite existing ones.' (selected) and 'Create only new system messages.'. At the bottom, there is a 'Create' button and a 'Close' button.</p>

步骤	过程：组态报警记录
10	<p>激活短期和长期归档。通过  R 归档 → 添加/删除，打开归档参数分配对话框。在该对话框中，激活短期归档和长期归档(顺序归档)。 按下确定按钮关闭对话框。</p>  <p>The dialog box titled 'Archive Parameter Assignment' contains a section labeled 'Archives' with two checked checkboxes: 'Short-term archive active' and 'Long-term archive active'. Below this is a section labeled 'Select the desired archives' with a list box and a 'Select' button. At the bottom are 'OK' and 'Cancel' buttons.</p>
11	保存后退出报警记录编辑器。

创建全局动作

步骤	过程：创建全局动作
1	打开全局脚本编辑器。
2	创建新的全局动作。这可通过全局脚本编辑器中的文件 → 新建动作菜单来完成。
3	在本实例中，已经编写了将 C 函数模拟为趋势的 C 动作。计算设定温度 $dTemp2$ 和实际温度 $dTemp1$ 之间的差值 $dDelta$ 。如果该差值为正，则趋势上升。如果该差值为负，则趋势下降。 加热量 $nPower$ 定义温度以多快的速度达到设定值。
4	通过编辑 → 编译来编译 C 动作。


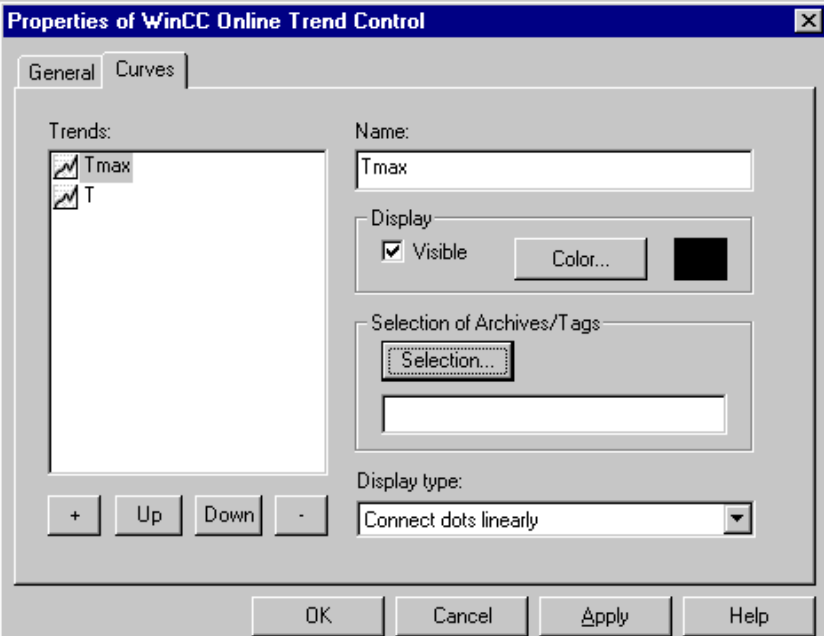
步骤	过程：创建全局动作
5	<p data-bbox="524 306 1356 365">通过 编辑 → 信息，打开描述对话框。在 触发器 标签内，本实例选择 周期定时器。通过 添加 按钮，显示更改触发器的对话框。</p> <div data-bbox="524 369 1219 1197"></div>

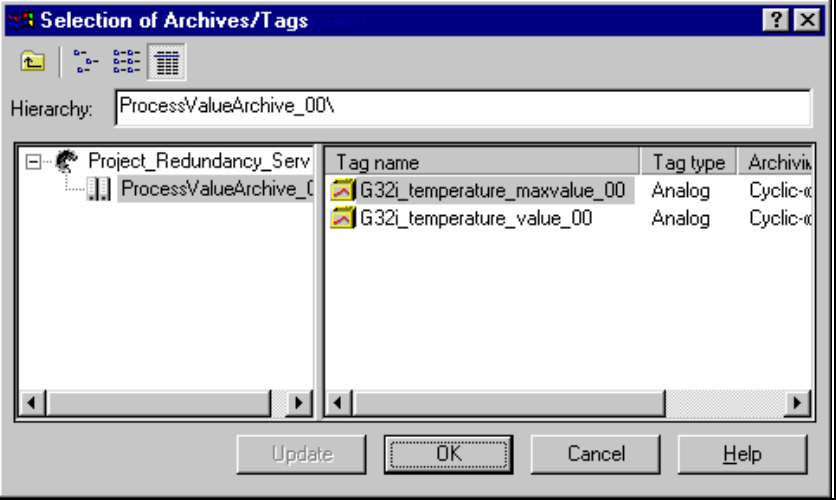
步骤	过程：创建全局动作
6	<p>将周期时间设置为 <i>250 ms</i>。 两个对话框都用 <i>确定</i> 按钮关闭。</p> 
7	保存后退出 <i>全局脚本编辑器</i> 。

组态对象

步骤	过程：组态对象
1	在图形编辑器中创建一个新画面。在本实例中，它是画面 <i>red_3_chapter_01.PDL</i> 。在该画面中，将各种不同的对象与过程变量相连。
2	<p>各用一个 <i>Windows 对象</i> → <i>滚动条对象</i> 来实现输入变量的模拟。在本实例中，它们是 <i>滑块对象 1</i> (<i>G32i_temperature_maxvalue_00</i>) 和 <i>滚动条对象 2</i> (<i>U08i_power_value_00</i>) 以及代表温度控制的 <i>I/O 域 1</i> (<i>G32i_temperature_maxvalue_00</i>)。在 <i>I/O 域 1</i> 中，显示设定的温度值，并且在此可以对其进行修改。</p> <p>在炉中显示输出变量(<i>G32i_temperature_value_00</i>)。它包括对象 <i>I/O 域 2</i> 和棒图 1。</p> <p>将这些对象的更新设置为一旦改变。</p> <div><div><div><div>T/°C</div><div>+23,0</div><div><div></div><div></div></div></div><div><div>P/kW</div><div>12345</div><div><div></div><div></div></div></div></div><div><div><div>T/°C</div><div>+23,0</div><div></div></div><div></div></div></div>

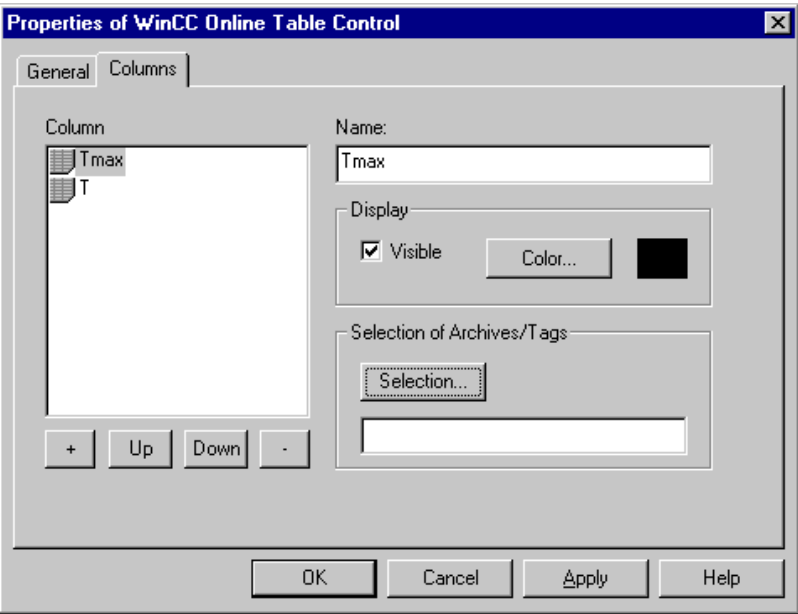
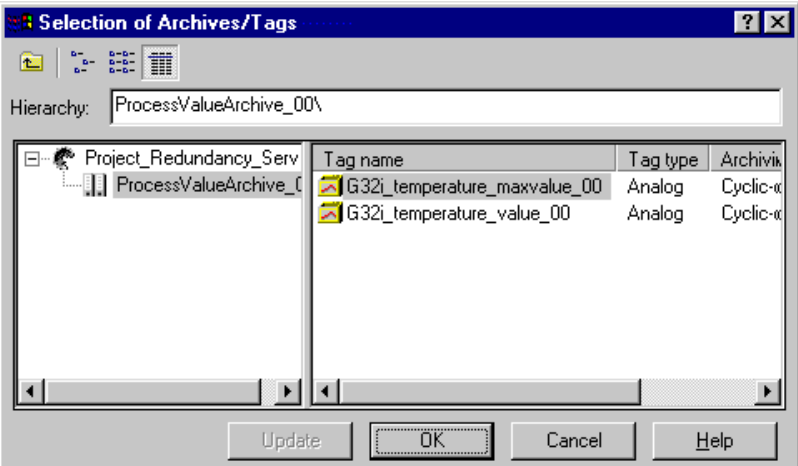
组态趋势窗口

步骤	过程：组态趋势窗口
1	在图形编辑器中另外创建一个画面。在实例中，它就是画面 <i>mcs_3_chapter_02.PDL</i> 。在该画面中，使用趋势窗口显示两个温度值。
2	<p>通过控件  WinCC 在线趋势控件创建趋势控件。在实例中，它就是对象 <i>TlgOnlineTrend1</i>。显示 WinCC 在线趋势控件属性对话框。在趋势标签中，通过单击+按钮添加一条新趋势。</p> <p>将趋势 1 重命名为 <i>Tmax</i>，将趋势 2 重命名为 <i>T</i>。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 

步骤	过程：组态趋势窗口
3	<div><p>显示 归档/变量选择对话框。该对话框允许选择归档/归档变量。</p><p>在本实例中，将 <i>Tmax</i> 与变量 <i>G32i_temperature_value_00</i> 相连接，将 <i>T</i> 与变量 <i>G32i_temperature_maxvalue</i> 相连接。</p></div>


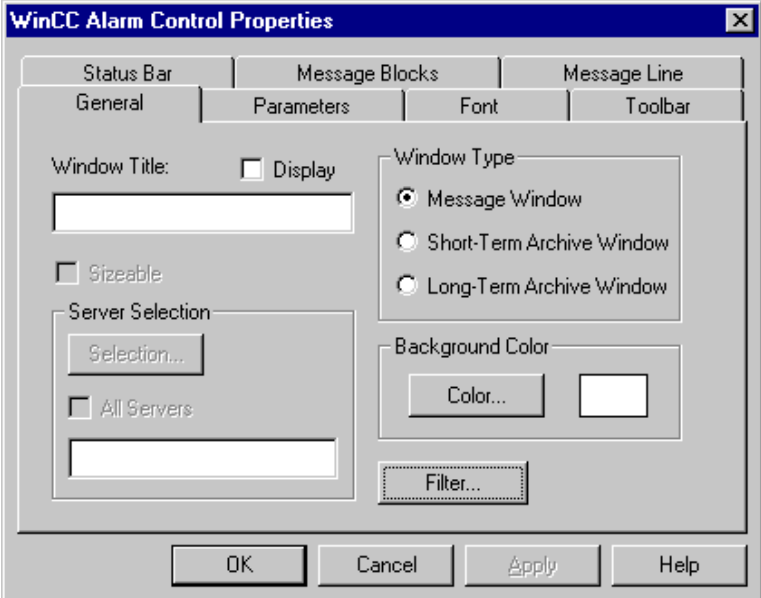
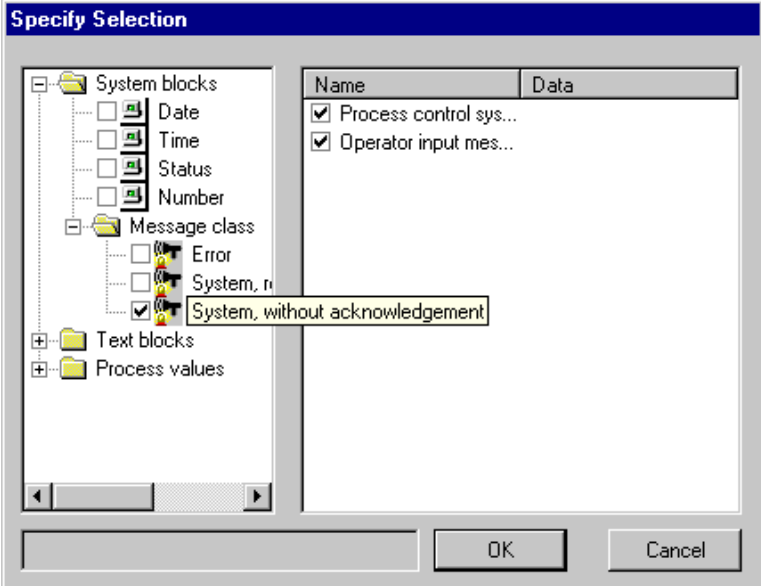
组态表格窗口

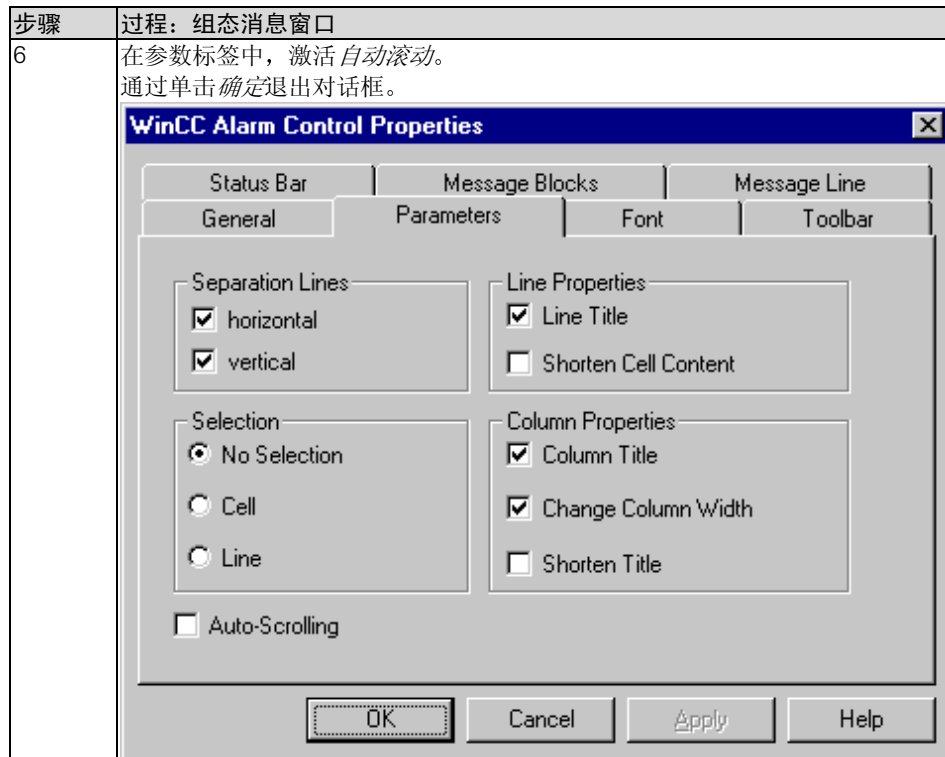
步骤	过程：组态表格窗口
1	在同一画面(<i>red_3_chapter_02.PDL</i>)中，使用表格窗口显示两个温度值。

步骤	过程：组态表格窗口
2	<p>通过 控件 → WinCC 在线表格控件，创建表格控件。在本实例中，它就是对象 TlgOnlineTable1。显示 WinCC 在线表格控件属性对话框。在列标签内，通过单击+按钮添加新列。</p> <p>将列 1 重命名为 Tmax，将列 2 重命名为 T。</p> <p>在归档/变量选择域内，通过选择按钮打开选择所期望归档变量的对话框。</p> 
3	<p>显示 归档/变量选择对话框。该对话框允许选择归档/归档变量。</p> <p>在本实例中，将 Tmax 与变量 G32i_temperature_value_00 相连接，将 T 与变量 G32i_temperature_maxvalue 相连接。</p> 


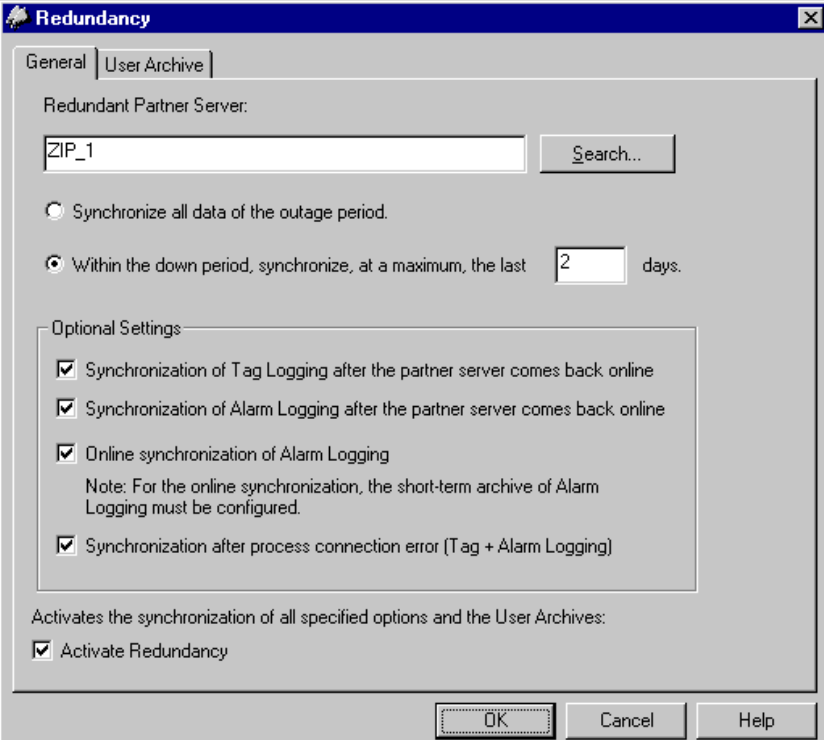
组态消息窗口

步骤	过程：组态消息窗口
1	在图形编辑器中另外创建一个画面。在本实例中，它是画面 <i>red_3_chapter_03.PDL</i> 。在该画面中，利用消息窗口输出所组态的消息。
2	<p>通过控件 → WinCC 报警控件，创建一个 WinCC 报警控件。在实例中，它就是对象 <i>CCAlgWinCtrl1</i>。显示 WinCC 报警控件属性 - 快速组态对话框。用确定按钮关闭该对话框。</p> 
3	按照刚才所描述的步骤，另外再组态一个 WinCC 报警控件。在实例中，它是 <i>CCAlgWinCtrl2</i> 对象。



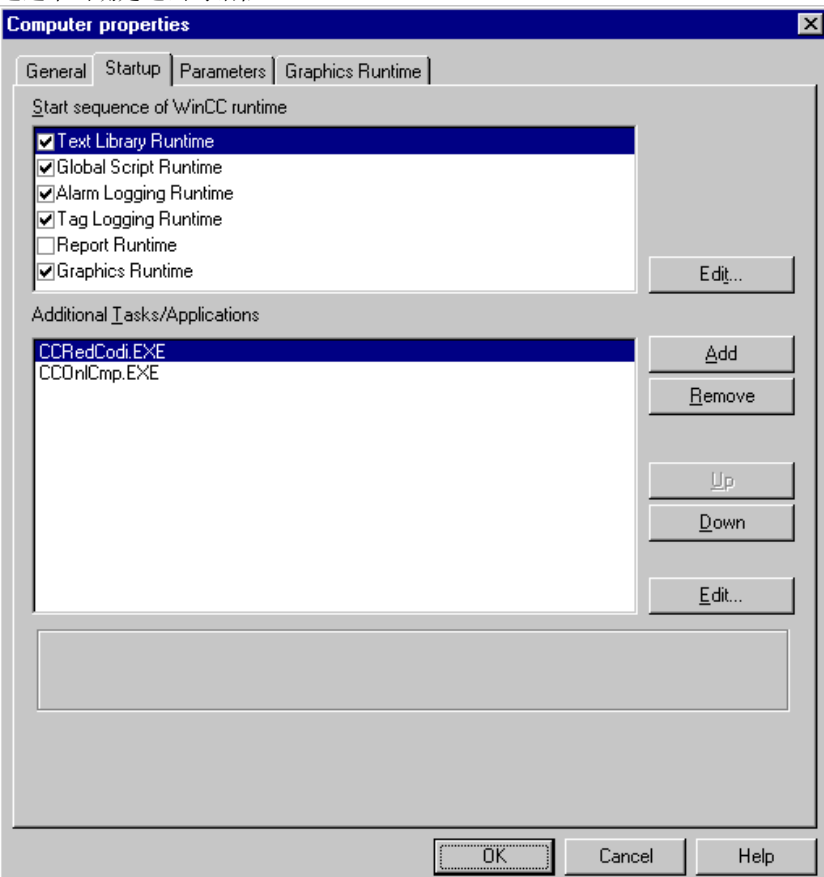
步骤	过程：组态消息窗口
4	<p>通过  刚才所创建的 WinCC 报警控件，将显示 WinCC 报警控件属性对话框。在窗口类型下的常规信息标签中，选择 Short_Term Archive Window。通过 Select 按钮，进入 Define Selection 对话框。</p> <div><p>The image shows the 'WinCC Alarm Control Properties' dialog box. It has several tabs: 'Status Bar', 'Message Blocks', 'Message Line', 'General', 'Parameters', 'Font', and 'Toolbar'. The 'General' tab is active. It contains fields for 'Window Title' (with a 'Display' checkbox), 'Window Type' (with radio buttons for 'Message Window', 'Short-Term Archive Window', and 'Long-Term Archive Window'), 'Background Color' (with a 'Color...' button), and 'Server Selection' (with a 'Selection...' button and an 'All Servers' checkbox). There is also a 'Filter...' button. At the bottom are 'OK', 'Cancel', 'Apply', and 'Help' buttons.</p></div> <p>Error!</p> <p>Bookmark not defined.</p>
5	<p>在所显示的对话框中，可指定要显示的消息。 选择系统，未确认复选框。单击确定按钮关闭对话框。</p> <div><p>The image shows the 'Specify Selection' dialog box. It has a tree view on the left and a list on the right. The tree view shows a hierarchy: 'System blocks' (expanded) containing 'Date', 'Time', 'Status', and 'Number'; 'Message class' (expanded) containing 'Error', 'System, n', and 'System, without acknowledgement' (which is selected); 'Text blocks'; and 'Process values'. The list on the right has two columns: 'Name' and 'Data'. It contains two entries: 'Process control sys...' and 'Operator input mes...', both with checked checkboxes. At the bottom are 'OK' and 'Cancel' buttons.</p></div>




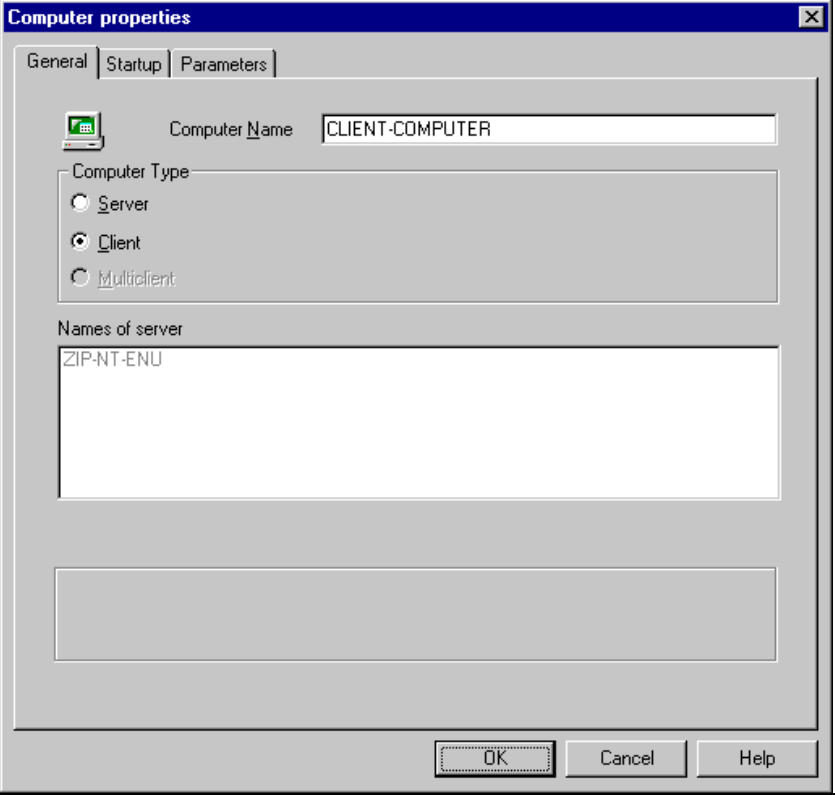
创建冗余

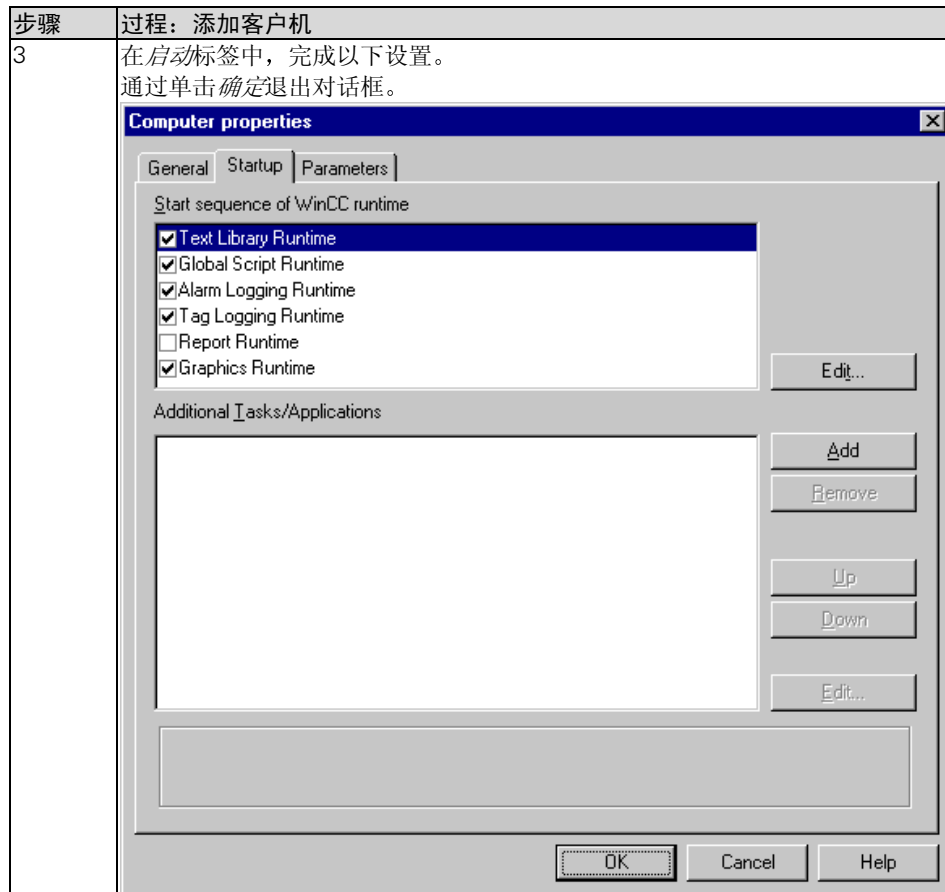
步骤	过程：创建冗余
1	通过  WinCC 资源管理器左边窗口上的冗余 → 打开，打开冗余对话框。
2	<p>在 常规 标签中，将冗余伙伴服务器的名称输入到冗余伙伴服务器域中。通过 搜索 按钮，显示 选择冗余伙伴 对话框，它用于搜索相应的计算机。</p> <p>选择 激活冗余 复选框。</p> <p>在 可选设置 域中，选择所有的复选框。</p> <p>通过单击 确定 退出对话框。</p> 

设置服务器的 WinCC 运行系统启动属性

步骤	过程：设置 WinCC 运行系统启动属性
1	通过使用  单击 WinCC 资源管理器左边的计算机条目，将在右窗口中显示计算机名称。
2	<p>通过  计算机名称 → 属性，显示计算机属性对话框。在启动标签中，完成以下设置。</p> <p>通过单击确定退出对话框。</p> 

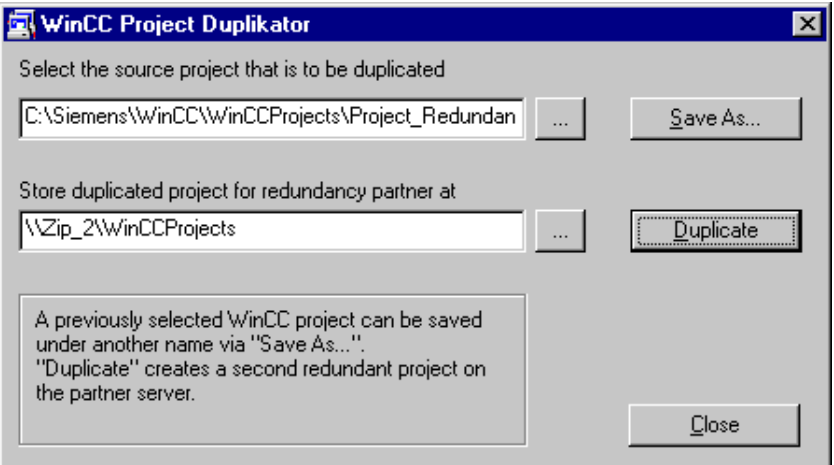
添加客户机

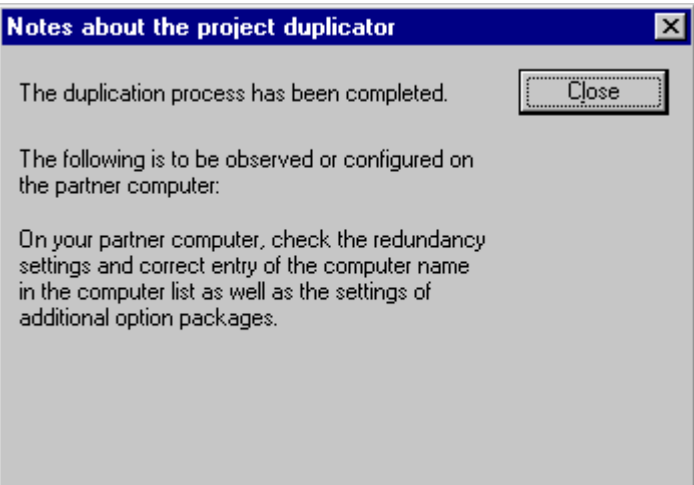
步骤	过程：添加客户机
1	添加一个新的计算机。通过  WinCC 资源管理器左边的 计算机 → 新计算机，显示 计算机属性对话框。
2	在 常规信息标签中的 计算机名称下，指定相应客户机计算机的名称。将客户机指定为 计算机类型。 



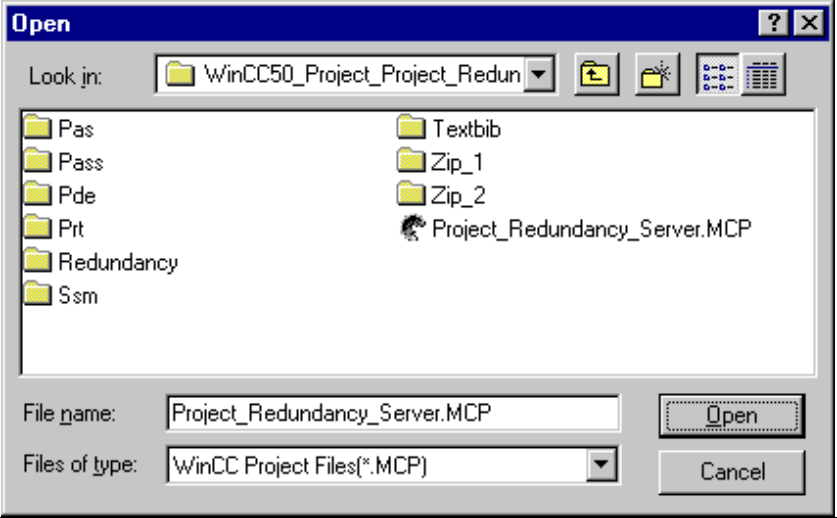
复制项目

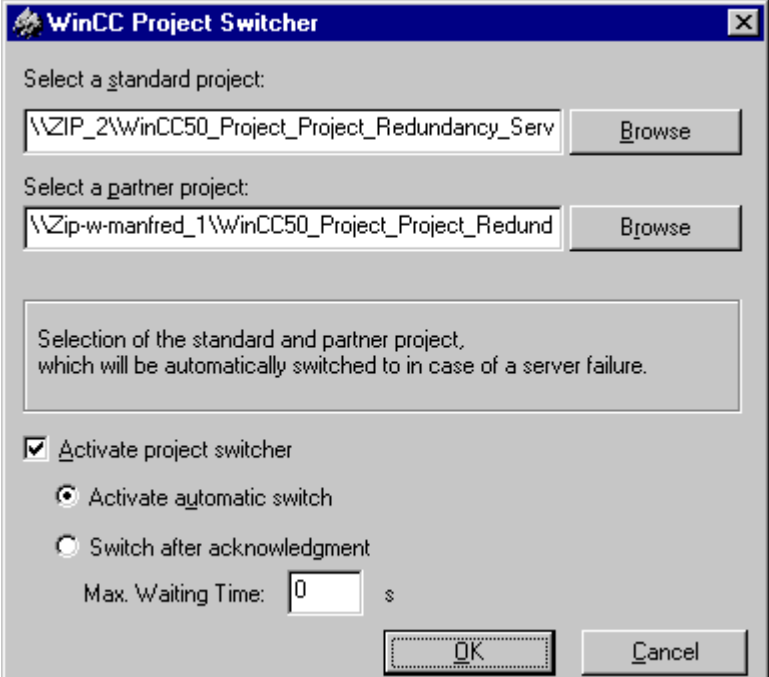
在两台服务器上，必须组态功能完全相同的项目。WinCC 项目复制器使得可以将与项目相关的所有数据复制到冗余服务器中。
WinCC 项目复制器自动创建冗余伙伴项目。将所有相关的项目数据(画面、脚本、归档等)复制给目标计算机并完成所有的设置，以使目标计算机可作为冗余服务器。

步骤	过程：复制项目
1	打开 WinCC 项目复制器。 通过 开始 → <i>Simatic</i> → <i>WinCC</i> → 工具 → 项目复制器来启动它。
2	将显示 <i>WinCC 冗余项目复制器</i> 对话框。 在 <i>选择要复制的源项目</i> 输入域中，选择源项目。 在将用于 冗余伙伴的复制项目存储在域中，指定包含目标项目文件夹的目标计算机。 通过 <i>复制</i> 按钮，启动复制过程。 

步骤	过程：复制项目
3	<p data-bbox="529 306 1182 359">在完成复制过程之后，显示关于 项目复制器的注意事项对话框。用 确定按钮关闭该对话框。</p> <div data-bbox="529 359 1219 842"><p data-bbox="537 369 922 401">Notes about the project duplicator</p><p data-bbox="545 432 976 464">The duplication process has been completed.</p><p data-bbox="1089 432 1146 464">Close</p><p data-bbox="545 495 1000 558">The following is to be observed or configured on the partner computer:</p><p data-bbox="545 579 1016 684">On your partner computer, check the redundancy settings and correct entry of the computer name in the computer list as well as the settings of additional option packages.</p></div>

在客户机上设置项目切换器

步骤	在客户机上设置项目切换器
1	<p>在客户机计算机上打开 WinCC。</p> <p>通过 开始 → Simatic → WinCC → 视窗控制中心来启动 WinCC 资源管理器。</p>
2	<p>将显示 WinCC 资源管理器。</p> <p>通过 文件 → 打开菜单，显示 打开对话框，用于选择一个 WinCC 项目。</p> <p>在“网上邻居”下，选择已经组态了服务器项目 <i>Project_Redundancy_Server</i> 的计算机。文件夹 <i>WinCC50_Project_Project_Redundancy_Server</i> 包含文件 <i>Project_Redundancy_Server.mcp</i>。选择该文件，并通过 打开按钮将其打开。</p> 

步骤	在客户机上设置项目切换器
3	<p>打开 WinCC 项目切换器。</p> <p>通过 开始 → Simatic → WinCC → 工具 → 项目切换器来启动它。</p> <p>在选择缺省项目输入域中，指定缺省情况下客户机与其相连的相关服务器的缺省项目。搜索按钮将有助于项目选择。</p> <p>在选择伙伴项目输入域中，指定相关服务器的缺省项目，当出现故障时要切换至该项目。搜索按钮将有助于项目选择。</p> <p>选择激活项目切换器复选框，然后选择自动切换条目。</p> <p>通过单击确定退出对话框。</p> 

4.3 WinCC 项目的描述

将两个服务器项目上的运行系统都激活。此后，客户机项目上的运行系统将自动激活。

在启动服务器计算机期间，冗余组件将确定伙伴服务器是否已经激活。如果伙伴服务器已经激活，则在服务器计算机上将设置从站标识。如果在启动期间还没有激活伙伴服务器，则在服务器计算机上将设置主站标识。如果出现了网络连接错误或伙伴服务器未激活，则主站标识将重新设置。为了将服务器计算机标记为主站，设置了内部 WinCC 变量@RM_MASTER。如果服务器计算机为从站，则重新设置变量@RM_MASTER。@RM_MASTER_NAME 变量含有服务器计算机的名称，即服务器 1。其它应用程序或脚本可对这些变量进行计算和修改。冗余组件只对上面所涉及的变量进行设置。两台服务器始终完全相同。

4.3.1 服务器项目



在出现总览画面之后，可以通过如上所示的按钮访问设备画面。



通过如上所示的按钮，可在各画面之间进行切换。



通过此按钮，可返回至总览画面。

设备画面

在设备画面中，显示一个带温度控制的炉。通过该温度控制可以预先设置温度。炉温上升直至达到预设值。通过能量控制可以指定热容量。该值会影响炉温上升的速度。

趋势和表格窗口

在下一个画面中，显示趋势和表格窗口。趋势窗口描绘预置温度(设定值)以及炉温(实际值)的进程。这两种数值在表格窗口中也都显示出来。

消息窗口

下一个画面对两个消息窗口都进行了显示。如果炉的温度值超过 300，则产生一个警告信息并显示在顶部消息窗口中。如果温度值超过 700，则产生一个报警信息并显示在底部消息窗口中。

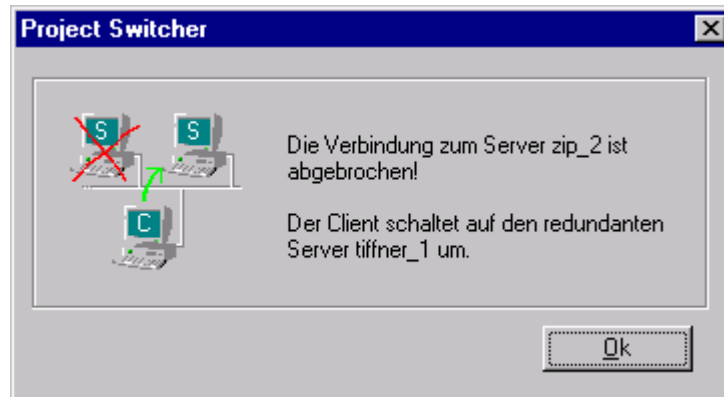
在底部消息窗口中，显示了系统消息，通过冗余可将其输出。

这些消息可以是：

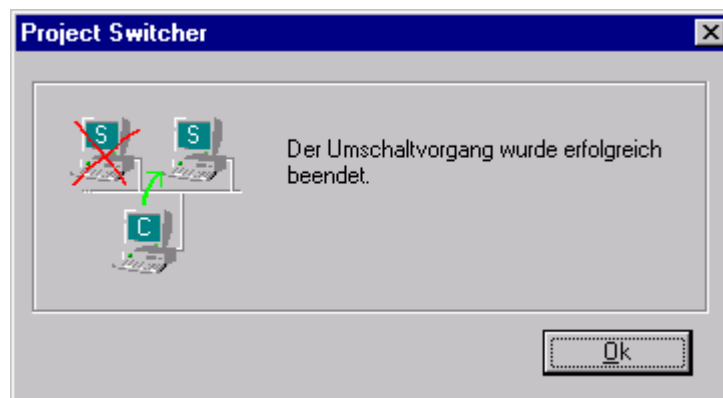
- 过程连接出错
- 报警记录同步
- 变量记录同步

在主服务器出现故障期间对过程进行切换

如果与主服务器的网络连接中断，将显示下列对话框：



客户机项目将自动关闭，并切换到伙伴服务器。随后，运行系统将在客户机上再次激活。在切换过程完成之后，显示下列对话框：



5 用户归档(Project_UserArchive)

本章中创建的项目也可以直接从在线文档复制到硬盘驱动器上。缺省情况下，它将存储在 *C:\Configuration_Manual* 文件夹中。



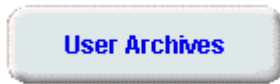
Project_UserArchive


此 WinCC 项目用于对用户归档编辑器的操作进行说明。同时，也对用户归档的组态过程和应用进行了解释。

下面显示了 Project_UserArchive 项目的起始页面。



5.1 用户归档



在运行系统中，对 *Project_UserArchive* 项目中与该主题相关的实例进行访问可通过使用  选择上面所显示的按钮来进行。这些实例均在画面 *ex_3_chapter_01.pdl* 至 *ex_3_chapter_01c.pdl* 中。

常规信息

该用户归档可利用数据库来将数据存储到自定义的表格结构中。

使用多种不同的方法可将该数据传送到一个低层 PLC 中。为了与 PLC 进行通讯，可以使用 WinCC 变量和 WinCC 原始数据变量。不需通讯时也可组态用户归档。

此时，可使用 4 种用于单个的数据域的数据类型。它们是：

- 整型
- 长整型
- 字符串
- 日期/时间

使用一个 OLE 控件，可在运行系统中以列表形式显示用户归档的数据。通过该 OLE 控件，可对用户归档的数据进行编辑，可写入到 PLC 中，可从 PLC 中读取，可导出到 CSV 文件中以及从 CSV 文件中导出。所显示的数据可根据确定的标准和条件进行过滤和排序。

OLE 控件具有可自由组态的工具栏和状态栏。可把各种不同的显示格式分配给各个表格列。

通过应用查看，来自于不同归档的数据可组合在一起进行显示。

5.1.1 用户归档的创建(ex_3_chapter_01.PDL)

任务定义


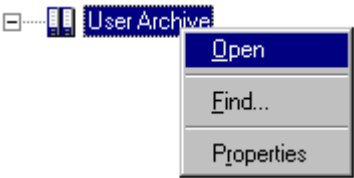


创建一个归档，其数据记录由 5 个整型域及一个用于记录数据记录名称的文本域所组成。通过记录可以使归档数据应用于整个项目记录。

概念的实现

为了对数据进行归档，在用户归档编辑器中创建归档。使用 WinCC 变量，组态归档的通讯。在该归档中，创建了 6 个所需域类型的数据域。给各个数据域均分配了一个变量(在本实例中，该变量是一个内部变量)，它允许归档与其余项目进行通讯。

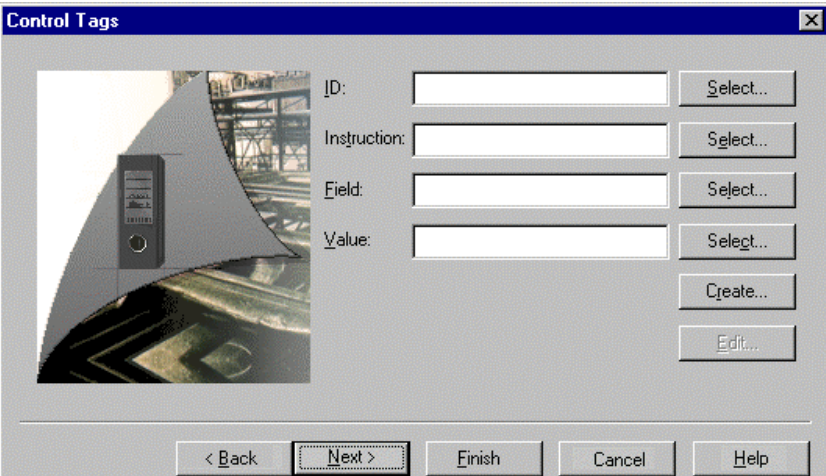
注意：
以下实例是对组态用户归档的实现过程的常规描述。没有建立任何与低层 PLC 的连接，取而代之的是使用了内部 WinCC 变量来演示操作。
对于具有已完整组态的连接实例可参见后续章。

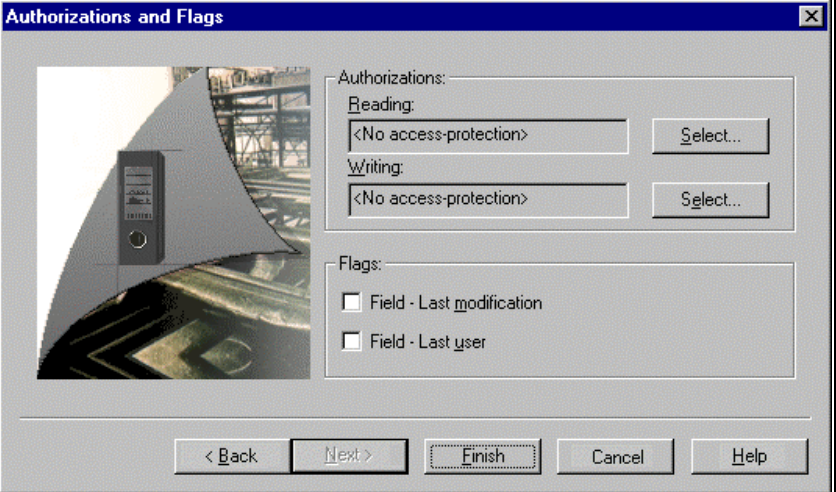
创建归档



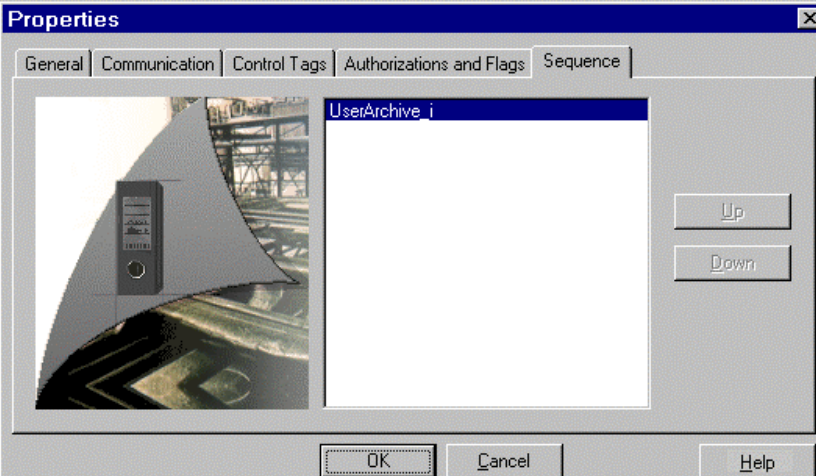
步骤	过程：创建归档
1	为归档的每个数据记录创建一个内部变量。由这些变量来执行归档和其余系统之间的通讯。 在实例中，使用了无符号 32 位数类型的变量 S32i_ex_UAi_01 至 S32i_ex_Uai_05。此外，创建一个 8 位字符集文本变量类型的变量来存储数据记录名。在本实例中，它是变量 T08i_ex_UAi_01。
2	打开用户归档编辑器。这可从 WinCC 资源管理器中通过  用户归档条目，然后从弹出菜单中选择打开来完成。 
3	创建一个新归档。通过  归档条目，然后从弹出菜单中选择新建归档，则可启动向导。该向导将指导用户创建新归档。 

步骤	过程：创建归档
4	<p>填写向导的 <i>常规信息</i> 页面。在这里，提供了与所要创建的归档相关的常规信息。</p> <p>必须输入 <i>归档名称</i>。归档名称不能超出 20 个字符的长度。不可使用任何 ANSI-C 编程语言或 SQL 数据库查询语言的关键字。在本实例中，将名称 <i>UserArchive_i</i> 输入到 <i>归档名称</i> 域内。</p> <p>在 <i>别名</i> 域内，输入对归档描述得更详细的文本。该域可接受所有字符与数字。填写该域是可选的。在实例中，别名域是空白的。</p> <p>而且，必须选择归档类型。如果选择了受限制，则随后必须指定数据记录的最大数目，它包含在归档中。在本实例中，将无限制设置为归档类型。</p> <p>通过单击 <i>下一步</i> 可继续下一页的操作。</p> 

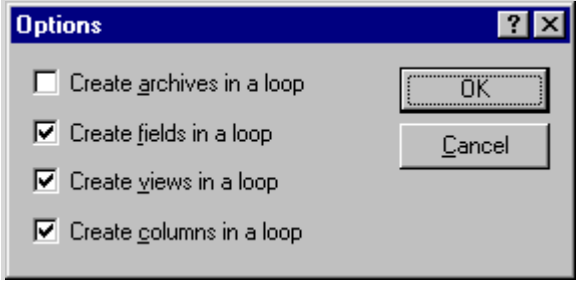
步骤	过程：创建归档
5	<p>填写向导的通讯页面。在这里，指定了所创建的归档如何与其它项目进行通讯。</p> <p>如果选择了 <i>通过 WinCC 原始数据变量进行通讯</i>，则必须在该页面中指定一个 WinCC 原始数据变量。在这种通讯类型下，使用设置变量可传送完整的数据记录。而且，必须输入一个 <i>PLCID</i>，它包含最多 8 个字符。通过该 ID，可将 PLC 所发送的原始数据块分配给一个确定的归档。</p> <p>如果选择了 <i>通过 WinCC 变量进行通讯</i>，则必须将一个 WinCC 变量分配给各个归档数据域。在创建对应的数据域期间将执行该操作。</p> <p>此外，还可以组态带有 <i>无任何通讯</i> 选项的归档。</p> <p>在本实例中，选择类型 <i>通过 WinCC 变量进行通讯</i>。</p> <p>通过单击 <i>下一步</i> 继续下一页的操作。</p> 

步骤	过程：创建归档
6	<p>填写向导的<i>控制变量</i>页面。在这里，可以设置 4 个内部或外部 WinCC 变量。通过这些变量，可以在运行系统中控制归档。</p> <p>在本实例中，没有使用控制变量选项。实例采用控制变量进行工作(ex_3_chapter_012.PDL)包含关于该主题的详细描述。</p> <p>通过单击 <i>下一步</i> 继续下一页的操作。</p> <div></div>



步骤	过程：创建归档
7	<p>填写向导的 <i>权限与标志</i> 页面。</p> <p>在权限域中，存在用于对访问类型读与写的授权等级进行设置的选项。通过下面所显示的按钮，可打开一个对话框，它包含有一个用用户管理员编辑器创建的所有授权等级的列表。</p> <p>从该列表中，可选择所期望的授权等级。<i>Pictu_3_Chapter_02.pd</i> 章包含了关于用户的组态和带有 <i>用户管理器</i> 编辑器的授权等级的详细描述。</p> <p>在标志域中，可将预定义的列插入归档。可使用存储最后一次访问数据记录的日期的列。此外，可使用存储最后一次访问数据记录的用户名称的列。</p> <p>在本实例中，没有在本页内进行任何设置。</p> <p>通过单击 <i>完成</i> 可关闭此向导页。</p> 


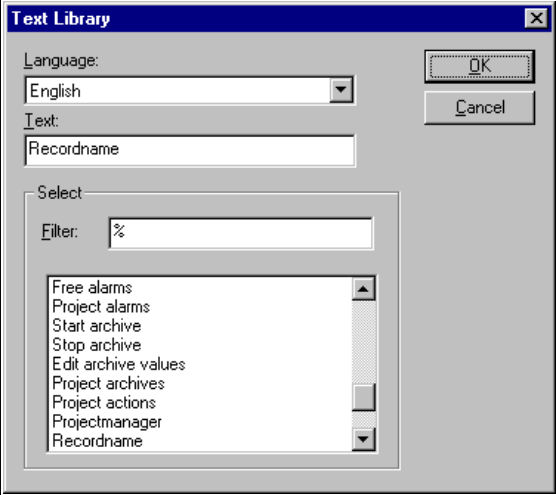
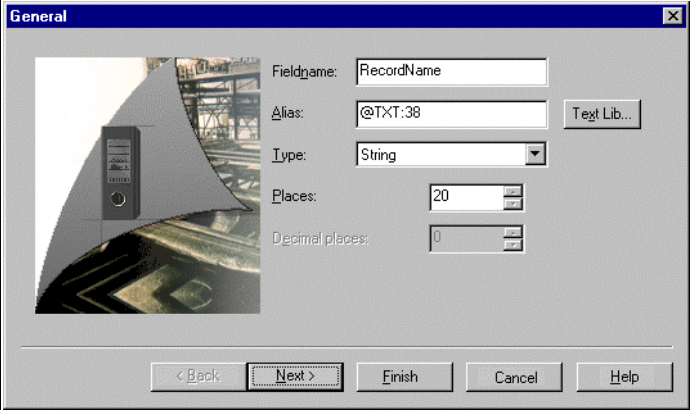
步骤	过程：创建归档
8	<p>现在即可在<i>归档</i>条目下的目录结构内显示刚才创建的归档。</p>  <p>通过  新建归档 <i>UserArchive_i</i> 的该条目，然后从弹出菜单中选择<i>属性</i>或在选择条目时单击工具栏上的属性按钮，可打开归档的属性对话框。该对话框显示了向导的所有页面和以标签形式呈现的条目。此外，还提供了一个用于设置归档顺序的标签。然而，在这个组态阶段只出现一个单独的归档。</p> 

注意：
在缺省情况下，已设置了用户归档编辑器来询问在完成归档之后是否创建一个新的归档。对该特性进行修改可通过在 *编辑* → *选项* 菜单中撤消相应复选框的选定来进行。这一点同样适用于域、视图和列。



数据域的组态

步骤	过程：数据域的组态
1	<p>创建归档的单个域。通过  新建归档 UserArchive_i 的条目，然后从弹出菜单中选择新建域，则可启动向导。该向导将指导用户创建一个新的域。</p> 
2	<p>填写向导的 <i>常规信息</i> 页面。在这里，提供了与所要创建的域相关的常规信息。必须输入一个 <i>域名称</i>。该名称将受到与应用到归档名称上的限制一样的约束。在本实例中，使用了 <i>域名 RecordName</i>。</p> <p>在别名域内，可输入一个不受任何限制的名称。在别名域中所输入的名称在运行系统中将被用作列标题。否则，使用域名作为列标题。可直接输入别名或在文本 库中的文本里对其进行引用。如果用多种语言创建项目，则第二个选项也将有用处。</p>


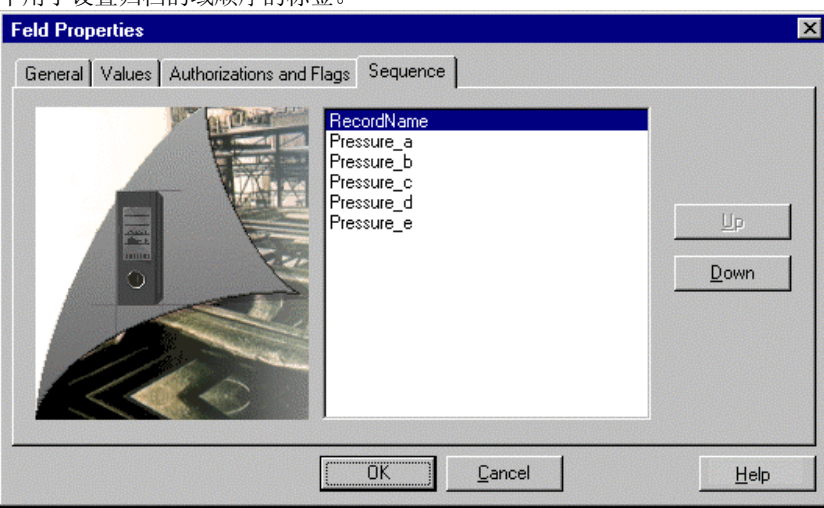


步骤	<p>过程：数据域的组态</p> <p>通过下面所显示的按钮，可打开一个对话框，它将用于将文本输入文本库或从文本库中选择文本。</p> <p>文本域中所输入的文本在按下 确定 按钮之后，将被输入文本库中。将引用该文本作为 <i>别名</i>。</p> <p>在 选择 域中，可对已经存在的文本进行搜索。在这种情况下，所搜索的文本中包含的某个字或文本段将被指定为 <i>过滤器</i>。% 字符可用作占位符。如果只输入了 % 字符，则文本库的全部内容均将显示在以下列表域内。通过  列表域，可对其内容进行更新。</p> <p>在本实例中，输入文本域名称。文本库对话将随着按下 确定 按钮而关闭。</p> <div data-bbox="522 653 1075 1144">The 'Text Library' dialog box has a title bar with a close button. It contains a 'Language' dropdown menu set to 'English', with 'OK' and 'Cancel' buttons to its right. Below is a 'Text' label followed by a text input field containing 'Recordname'. A 'Select' section contains a 'Filter' input field with a '%' character. Below the filter is a list box containing the following items: 'Free alarms', 'Project alarms', 'Start archive', 'Stop archive', 'Edit archive values', 'Project archives', 'Project actions', 'Projectmanager', and 'Recordname'. The list box has scroll bars.</div> <p>此外，必须选择域的类型。目前，可使用的类型有 <i>整型</i>、<i>长整型</i>、<i>字符串</i> 与 <i>日期/时间</i>。在本实例中，设置了 <i>类型字符串</i>。对该类型，也必须设置字符串的 <i>长度</i>。在本实例中，所设置的值为 <i>10</i>。</p> <p>通过单击 下一步 继续下一页的操作。</p> <div data-bbox="522 1266 1208 1673">The 'General' dialog box features a preview image on the left showing a control panel. On the right, there are several fields: 'Fieldname' with 'RecordName', 'Alias' with '@TXT:38' and a 'Text Lib...' button, 'Type' with a dropdown set to 'String', 'Places' with a value of '20', and 'Decimal places' with a value of '0'. At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.</div>
----	--

步骤	过程：数据域的组态
3	<p>填写向导的数值页面。对于数字域类型，可设置一个最大值、最小值与起始值。对于文本域，可将所输入的起始值作为缺省值。</p> <p>如果已经组态了一个通过 WinCC 变量进行通讯按钮，则必须为各个域设置一个相应的变量。在本实例中，选择了先前已创建的文本变量 T08i_ex_UAi_01。</p> <p>通过单击下一步继续下一页的操作。</p> 

步骤	过程：数据域的组态
4	<p>填写向导的权限与标志页面。</p> <p>在权限域中，存在用于对访问类型读与写的授权等级进行设置的选项。这里所设置的授权等级指的是单个域。它将允许对单个域的数据进行保护，以避免对其进行未授权的访问。</p> <p>在标志域中，可将各种不同的属性分配给域。可分配下列属性：</p> <ul style="list-style-type: none"> 该域必须包含一个数值，它不能为 0。 域必须包含一个唯一的值，列的值必须互不相同。 该域由索引支持。 <p>在本实例中，没有在本页内进行任何设置。</p> <p>通过单击完成可关闭此向导页。</p>

5	<p>在本实例中，创建了 5 个附加的域。它们均属整型。这些域的名称分别为 Pressure_01 至 Pressure_05，其别名分别为 Pressure1 到 Pressure5。至于变量，已为各个域设置了某个先前已创建的无符号 32 位数变量。其余的域设置则对应于第一个域。</p> <p>如果在左边的目录结构中选择了归档 UserArchive_i，则新创建的域将显示在列表视图中。</p>
---	--

Name	Alias	Type	Length	Tag name
RecordName	@TXT:38	String	20	T08i_ex_UAi_01
Pressure_e	@TXT:44	Number (integer)		S32i_ex_UAi_05
Pressure_d	@TXT:43	Number (integer)		S32i_ex_UAi_04
Pressure_c	@TXT:42	Number (integer)		S32i_ex_UAi_03
Pressure_b	@TXT:41	Number (integer)		S32i_ex_UAi_02
Pressure_a	@TXT:40	Number (integer)		S32i_ex_UAi_01

步骤	过程: 数据域的组态
6	<p>通过  R 域的条目, 然后从弹出菜单中选择 <i>属性</i>, 则可打开域的属性对话框。该对话框显示了向导的所有页面和以标签形式呈现的条目。此外, 还提供了一个用于设置归档的域顺序的标签。</p>  <p>The screenshot shows the 'Field Properties' dialog box with the 'Sequence' tab selected. It contains a list of record names: RecordName, Pressure_a, Pressure_b, Pressure_c, Pressure_d, and Pressure_e. There are 'Up' and 'Down' buttons to the right of the list for reordering. The dialog also has 'General', 'Values', and 'Authorizations and Flags' tabs, and 'OK', 'Cancel', and 'Help' buttons at the bottom.</p>
7	<p>保存所作的组态。这可通过如下显示的工具栏按钮或 <i>项目</i> → <i>保存</i> 菜单来完成。</p>  <p>请确保 <i>用户归档编辑器</i> 不处于输入归档数据的模式, 也就是说, 不能按下下面所显示的按钮。只有这时才能把组态数据传送给数据库。</p> 

5.1.2 数据输入(ex_3_chapter_01.PDL)

任务定义


本实例基于前面的实例用户归档的创建(ex_3_chapter_01.PDL)。为先前已创建的归档提供数据。在这种情况下，将利用在用户归档编辑器中直接输入数据的选项和从外部文件中导入数据的选项。


在用户归档编辑器中输入数据

步骤

过程：在用户归档编辑器中输入数据

1

打开用户归档编辑器。为了将数据记录输入到归档中，必须使用在左边的树形结构中选择归档。



2


必须将归档切换到数据输入模式。这可通过 编辑 → 运行系统数据菜单来完成。只有在这种模式下才能修改归档的数据。只能在保存所进行的所有修改之后才能切换到该模式。


在表格窗口底部，显示了一个空白表格，表格的列与已组态的归档的数据结构相对应。

	ID	RecordNam	Pressure_a	Pressure_b	Pressure_c	Pressure_d	Pressure_e
...							

3

将数据输入到表格中。

为了将数据输入到一个域中，必须使用对其进行选择。如果表格窗口是激活的窗口，则通过单击下面所显示的工具栏按钮，也可输入一条新的数据记录。





可使用 ENTER 键来完成数值的输入。

	ID	RecordNam	Pressure_a	Pressure_b	Pressure_c	Pressure_d	Pressure_e
1	1	Record_01	563	782	798	565	567
2	2	Record_02	589	808	823	589	590
3	3	Record_03	615	843	848	613	613
4	4	Record_04	641	860	873	637	636
...							


4

如果表格窗口是激活的窗口，且已按下了下面所显示的工具栏按钮，则可将表格窗口切换到只对数据进行简单修改和编辑的模式。





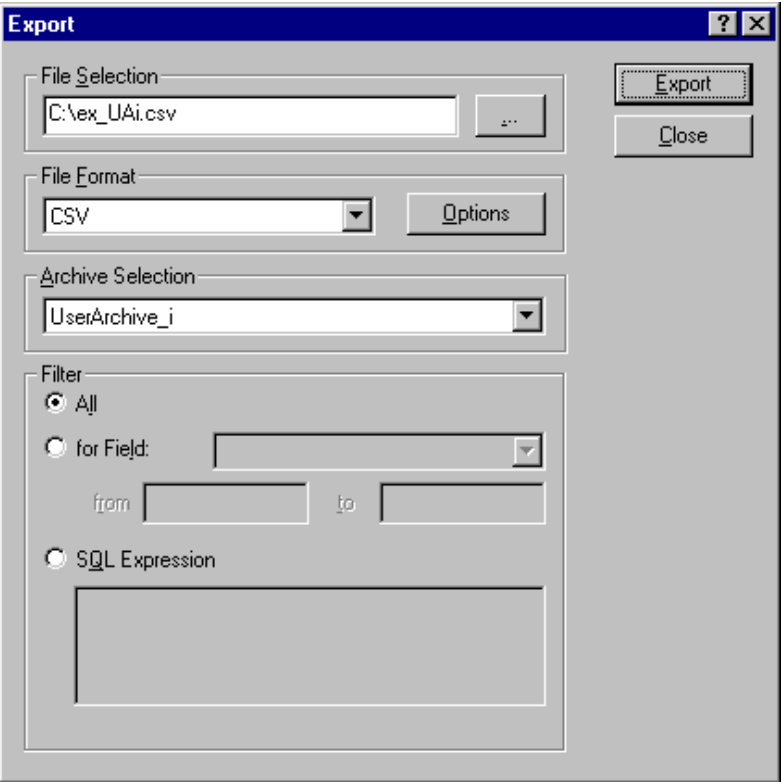

现在即可使用条目选择一个域。

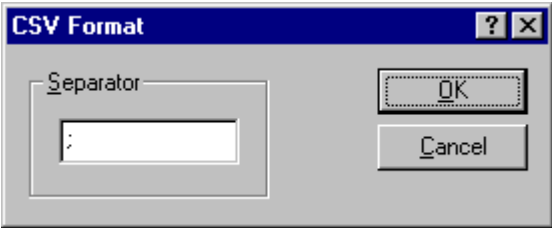
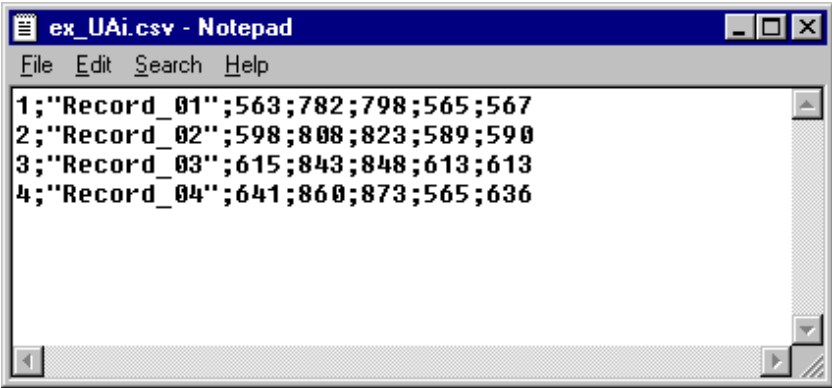
如果表格窗口是激活的窗口，且已按下下面所显示的工具栏按钮，则当前所选择的数据记录将被删除。





一次只能删除一条数据记录。

从 CSV 文件导入数据

步骤	过程：从 CSV 文件导入数据
1	<p>确定导入文件的结构。 为此，可将先前所输入的数据记录导出到一个 CSV 文件中。这通过如下所示的工具栏按钮来完成。</p> <p></p> <p>请确保用户归档编辑器没有处于输入归档数据的模式，也就是说，不能按下如下所示的按钮。只有这样，才可能导出和输入数据。</p> <p></p> <p>将显示导出对话框。</p> <div></div> <p>在文件选择域中，指定归档的数据记录将导出至的那个文件的名称。通过下面所显示的按钮，也可选择一个已存在的文件。</p> <p></p>

步骤	<p>过程：从 CSV 文件导入数据</p> <p>在本实例中，在文件选择域里设置 <i>ex_UAi.csv</i> 文件。</p> <p>保留 CSV (用逗号分隔的数值)文件格式。通过选项按钮，可改变文件所使用的分隔符。在本实例中，保留的缺省符号是分号。</p>  <p>在归档选择域中，指定归档 <i>UserArchive_i</i>。</p> <p>对过滤器，保留全部，因而可导出归档的所有数据。也可将过滤器设置为某个域的数值或使用 SQL 表达式进行选择。</p> <p>通过导出按钮，可启动所选文件的数据的导出过程。</p>
2	<p>现在即可使用任何文本编辑器打开刚才创建的文件。在本实例中，已使用 Microsoft 编辑器打开了文件。这样，就获得了导入所需要的文件结构。</p>  <p>在创建一个用于进行数据导入的基础文件时，请确保删除文本域两边的单引号。否则，在导入时会发生错误。在进行导入时，现有的数据记录将不会被覆盖。可使用第一列中所显示的 ID 来识别数据记录。导入文件应只包含带有 ID 的数据记录，目前还未对其进行分配。</p>

步骤	过程: 从 CSV 文件导入数据																																																						
3	<p>现在即可使用选项打开 Excel 下获得的文件, 并作进一步的处理。在 Excel 的打开对话框中, 将文件类型指定为文本文件, 而不是 Microsoft Excel 文件。</p> <p>Files of type: Text Files (*.prn; *.txt; *.csv)</p> <p>于是, 归档的各个域将分别在表格列中显示。现在即可如下所示方便地添加新的数据记录。</p> <table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th><th>G</th><th>H</th></tr><tr><td>1</td><td>5</td><td>Record_05</td><td>563</td><td>782</td><td>798</td><td>565</td><td>567</td><td></td></tr><tr><td>2</td><td>6</td><td>Record_06</td><td>598</td><td>808</td><td>823</td><td>589</td><td>590</td><td></td></tr><tr><td>3</td><td>7</td><td>Record_07</td><td>615</td><td>834</td><td>848</td><td>613</td><td>613</td><td></td></tr><tr><td>4</td><td>8</td><td>Record_08</td><td>641</td><td>860</td><td>873</td><td>637</td><td>636</td><td></td></tr><tr><td>5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <p>在本实例中, 通过文件 → 另存为...菜单, 然后将其重新命名为 <i>im_UAi.csv</i> 来保存文件。这样, Excel 将自动删除文本域两边的单引号。关闭 Excel。最后使用 否来回答对保存的询问, 因为数据已被保存到一个新的文件中。</p>		A	B	C	D	E	F	G	H	1	5	Record_05	563	782	798	565	567		2	6	Record_06	598	808	823	589	590		3	7	Record_07	615	834	848	613	613		4	8	Record_08	641	860	873	637	636		5								
	A	B	C	D	E	F	G	H																																															
1	5	Record_05	563	782	798	565	567																																																
2	6	Record_06	598	808	823	589	590																																																
3	7	Record_07	615	834	848	613	613																																																
4	8	Record_08	641	860	873	637	636																																																
5																																																							
4	<p>将数据导入到用户归档编辑器。这通过如下所示的工具栏按钮来完成。</p>  <p>将打开 导入对话框。</p>  <p>在文件选择域中, 可通过下面所显示的按钮来指定包含数据的文件。</p>  <p>在本实例中, 在文件选择域里选择先前所创建的文件 <i>im_UAi.csv</i>。 在文件格式域中, 保留了 <i>CSV</i>。通过选项按钮, 必须指定在文件中使用的分隔符。在本实例中, 保留的缺省符号是分号。 在归档选择域中设置目标归档, 在本实例中就是 <i>UserArchive_i</i>。 通过 导入按钮, 可启动数据导入。</p>																																																						
5	<p>将所导入的数据直接写入数据库。因此, 不需要再保存数据。</p>																																																						

注意:

在进行导入时, 请确保文件在 Excel 中不再打开。否则, 将报告出错。如果文件包含已存在的数据记录或文件的数据结构与归档的结构不符, 也将报告同样的出错信息。



5.1.3 表格视图的组态(ex_3_chapter_01.PDL)

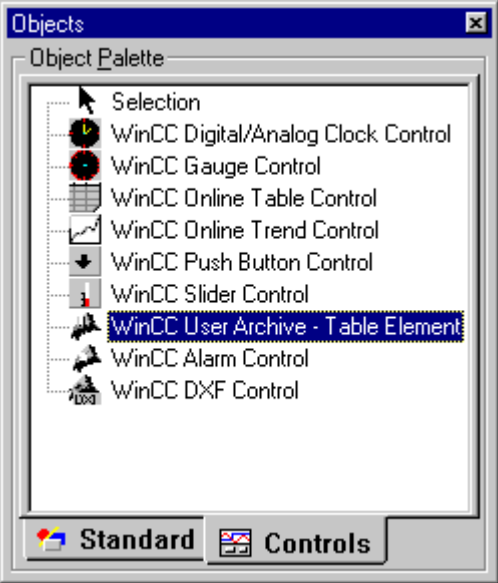

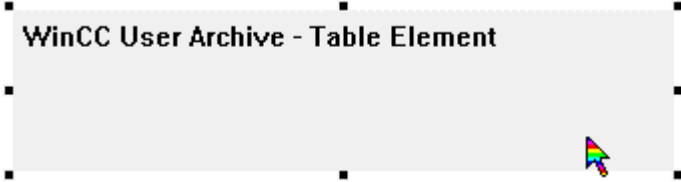
任务定义

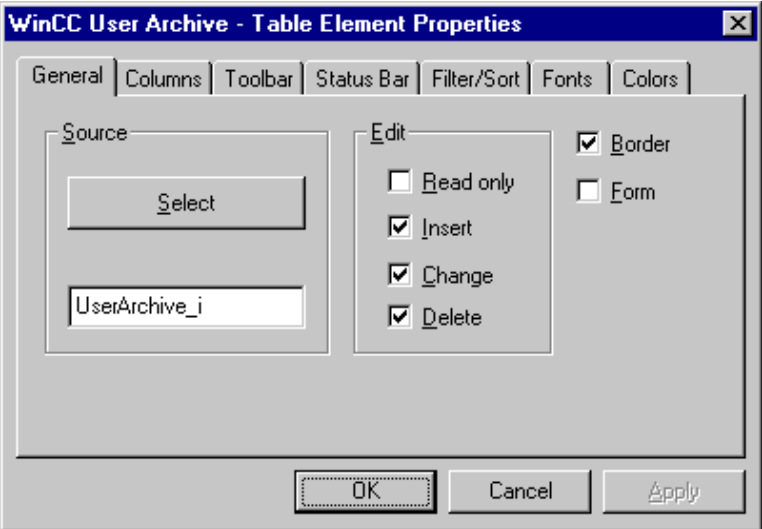
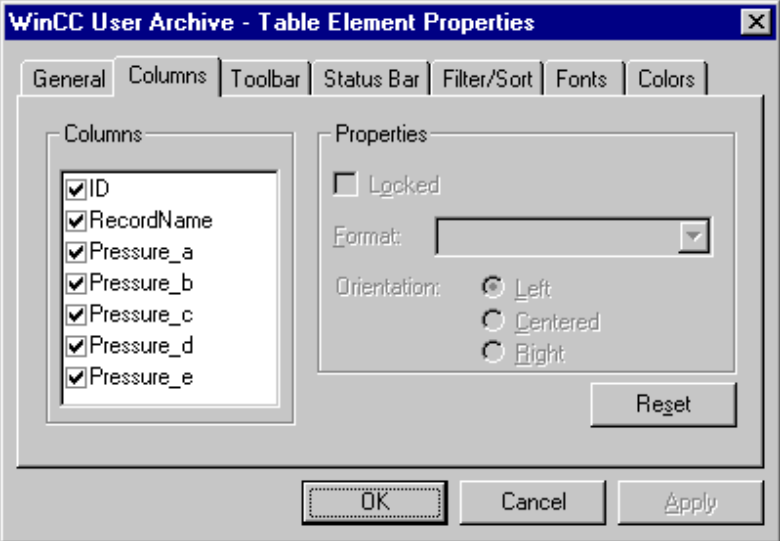
本实例基于先前的两个实例用户归档的创建(ex_3_chapter_01.PDL)与数据输入(ex_3_chapter_01.PDL)。在这些实例中所组态的归档数据可供用户在运行系统中使用。归档数据以表格形式进行显示。

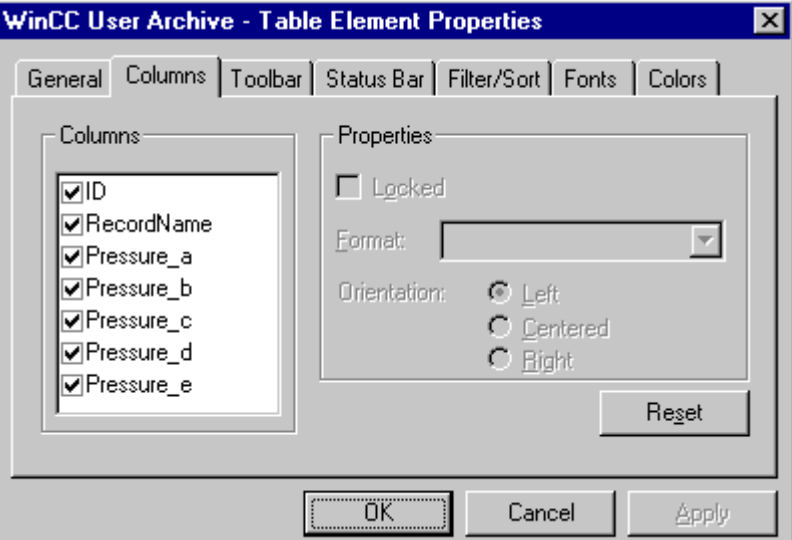
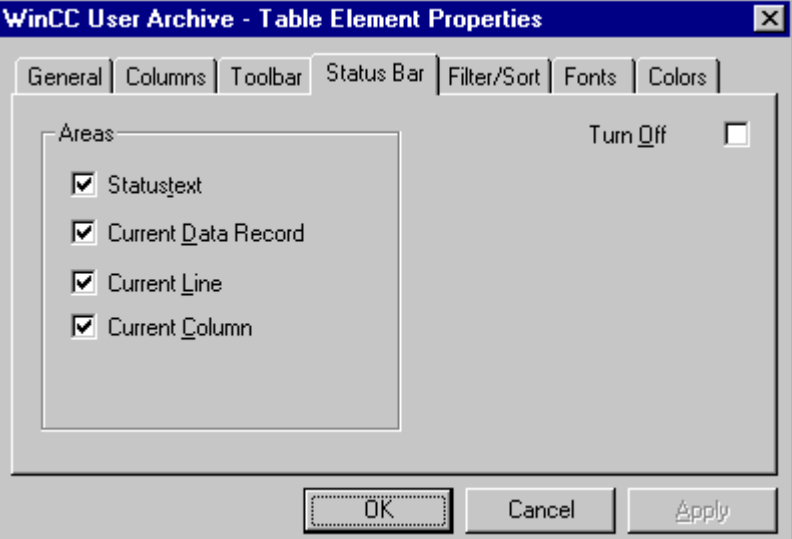
概念的实现

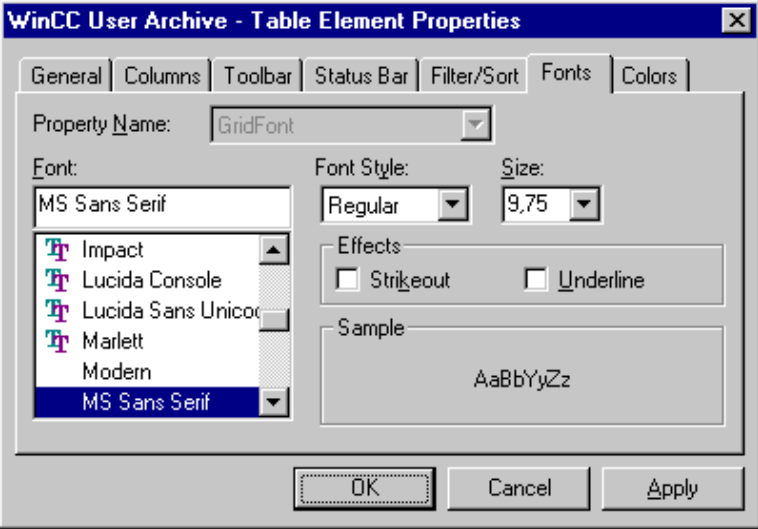
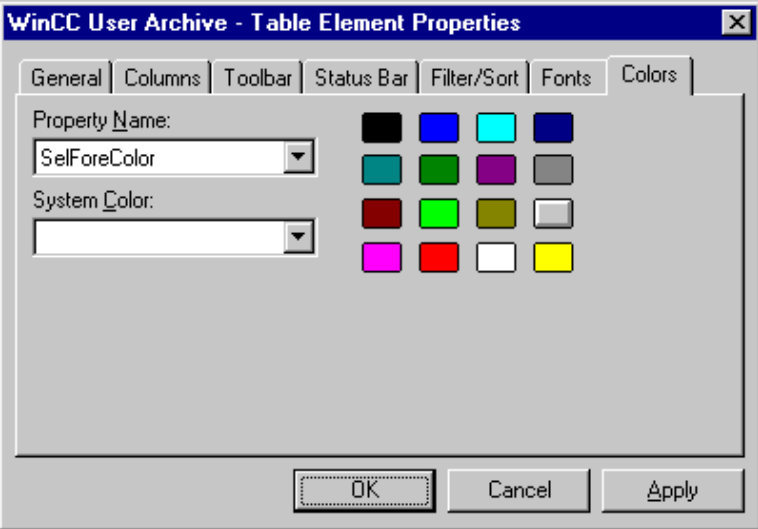
在运行系统中，归档通过特定的控件来显示。该控件以表格形式显示数据。通过该控件，用于所显示归档的各种编辑选项可供用户使用。

在图形编辑器中的实现

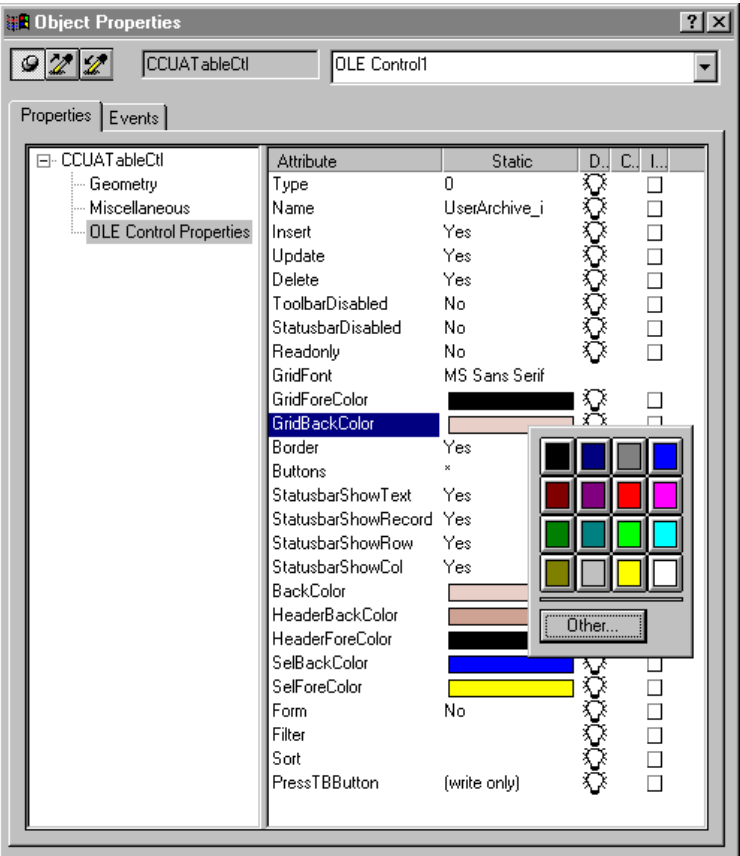

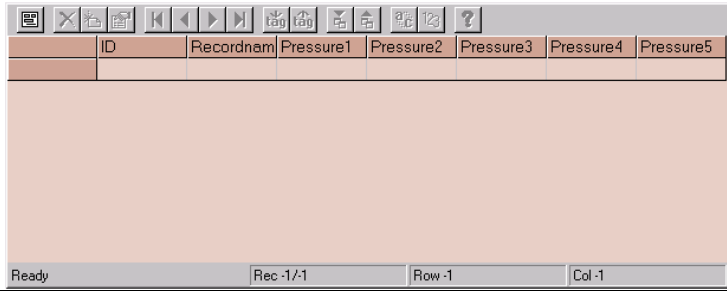
步骤	过程：在图形编辑器中的实现
1	打开 图形编辑器 并创建一个新画面。在本实例中，它是画面 <i>ex_3_chapter_01.pdl</i> 。
2	组态用于显示数据的控件。它是 WinCC 用户归档 - 表格元素。从对象选项板的控件选择菜单中选择它，然后将其置于画面中。 <div></div>
3	通过  刚创建的对象 控件 1，打开其属性对话框。 <div></div>

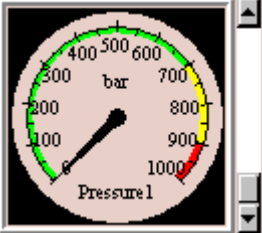
步骤	过程：在图形编辑器中的实现
4	<p>在<i>常规信息</i>标签内，选择源域中的<i>归档</i>选项。在本实例中，不能选择视图，因为还没有对其进行任何组态。至于归档，则选择 <i>UserArchive_i</i>。</p> <p>在<i>编辑</i>域中，可指定用户访问归档的<i>编辑</i>选项。在本实例中，用户可以使用最多的编辑选项。</p> <p>控件可以带<i>边框</i>显示。此外，还可使用窗体视图来显示对象。在本实例中，选择<i>边框</i>复选框。但不选择<i>窗体</i>复选框。</p> 
5	<p>在<i>列</i>标签中，可以指定表格中所显示的数据域。在本实例中，选择所有可用的域。</p> <p>此外，可在属性域中为各数据域定义确定的<i>属性</i>。然而，在本实例中保留所有数据域的缺省设置。</p> 

步骤	过程：在图形编辑器中的实现
6	<p>在工具栏标签中，可以组态工具栏所需的元素。在本实例中，选择所有剩下的功能。可以为每个工具栏按钮分配一个授权等级，这样就只允许经授权的用户对其进行操作。</p> <p>如果选择关闭复选框，则工具栏将不会显示。如果通过左边列表中的复选框撤消选定各按钮，则它们将以不可操作的状态显示。</p> 
7	<p>在状态栏标签中，可定义状态栏的外观。在本实例中，选择所有剩下的元素。</p> <p>如果选择关闭复选框，则状态栏将不会显示。</p> 

步骤	过程：在图形编辑器中的实现
8	<p>在字体标签中，可根据需要更改控件所使用的字体。在本实例中，为了在运行系统中同时显示所有的列，将字体的大小缩小为 10。所有其它设置均保留。</p> 
9	<p>在过滤器/排序标签中，本实例没有进行设置。如果工具栏中已经提供相应的按钮，则过滤器和排序命令也可以在运行期间进行设置。</p>
10	<p>在颜色标签中，可更改各表格元素的颜色。特别是要求 16 色以上时，设置颜色的最方便地点就是 WinCC 提供的控件 1 对象自身的属性对话框。</p> 
11	<p>在 WinCC 用户归档 - 表格元素的属性对话框中所作的设置可通过确定按钮来结束。</p>







步骤	过程：在图形编辑器中的实现
12	<p>设置表格的颜色方案。为此， 然后从弹出式菜单中选择 <i>属性</i> 来打开 <i>控件 1</i> 对象的属性对话框。</p> 

步骤	<p>过程：在图形编辑器中的实现</p> <p>在本实例中，颜色 <i>GridBackColor</i>、<i>BackColor</i> 与 <i>HeaderBackColor</i> 均按照项目中使用的颜色方案进行修改。其余颜色均保留缺省设置。</p> <p>在此处还可以进行 <i>WinCC 用户归档表格控件属性</i> 对话框的所有设置。然而，这对于有些设置可能没用。</p> 
13	<p>所做的某些组态已经可以在图形编辑器中进行检查。在按下 CTRL 键的同时， 控件 1 对象，可激活该对象。</p> 

步骤	过程：在图形编辑器中的实现
14	<p>为了显示由用户归档写入的变量值，实例使用了五个指针工具。</p> <p>使用 <i>WinCC 仪表控件</i> 类型的 <i>控件</i> 作为指针工具。在本实例中，它们是对象 <i>控件 2</i> 至 <i>控件 6</i>。</p> <p>对于其中的每个对象，在 <i>属性</i> → <i>控件属性</i> → <i>值</i> 处创建一个与由归档写入的五个整型变量之一的 <i>变量连接</i>。</p> <p>为了模拟这些变量的数值变化，组态五个 <i>智能对象</i> → <i>滚动条对象</i>。在本实例中，它们是 <i>滚动条对象 1</i> 至 <i>滚动条对象 5</i>。</p> <p>为这些对象，在 <i>属性</i> → <i>其它</i> → <i>过程驱动程序连接</i> 处创建与相应变量之一的 <i>变量连接</i>。而且，在 <i>事件</i> → <i>属性主题</i> → <i>其它</i> → <i>过程驱动程序连接</i> → <i>更改</i> 处各创建一个与相同变量的 <i>直接连接</i>。对于这些 <i>直接连接</i>，将 <i>源本对象</i> → <i>过程驱动程序连接</i> 与相应变量的目标相连接。完成该操作可使得每次滚动条位置变化时，变量值立即改变。</p> 
15	<p>为了显示当前装载到变量中的数据记录，组态一个 <i>标准对象</i> → <i>静态文本</i>。在本实例中，它是 <i>静态文本 2</i> 对象。为该对象，在 <i>属性</i> → <i>字体</i> → <i>文本</i> 处创建一个与包含数据记录名称的变量 <i>T08i_ex_UAi_01</i> 的 <i>变量连接</i>。</p>

常规应用的注意事项

在运行系统中，如下所示的控件的工具栏按钮执行下列任务：

-  通过窗体按钮，可在运行系统中进行表格视图与窗体视图之间的切换。
-  通过编辑按钮，也可在运行系统中修改归档的数据。可以删除、添加或修改数据记录。用户具有何种程度的修改数据权限取决于 *图形编辑器* 中控件的设置。
-  通过浏览按钮，用户可浏览表格并选择不同的数据记录。也可以通过直接用鼠标选择记录来完成数据记录的选择。
-  数据传送按钮用于写数据和读数据。通过 *写变量* 按钮，将表格中选择的数据记录写入 *用户归档* 中所设置的变量。通过 *读变量* 按钮，将 *用户归档* 中所设置变量的当前值读到表格中所选择的数据记录中。
-  通过导出与导入按钮，可将归档数据导出到 CSV 文件或从 CSV 文件中导入新的数据。这些按钮的功能与 *用户归档编辑器* 中相应按钮的功能相同。数据的导入与导出过程参见数据输入(ex_3_chapter_01.PDL)实例中的描述。
-  通过 *过滤器* 与 *排序* 按钮，可定义要显示的数据记录及其显示顺序。控件的这些设置也可在 *图形编辑器* 中进行。

5.1.4 表单视图的组态(ex_3_chapter_011.PDL)

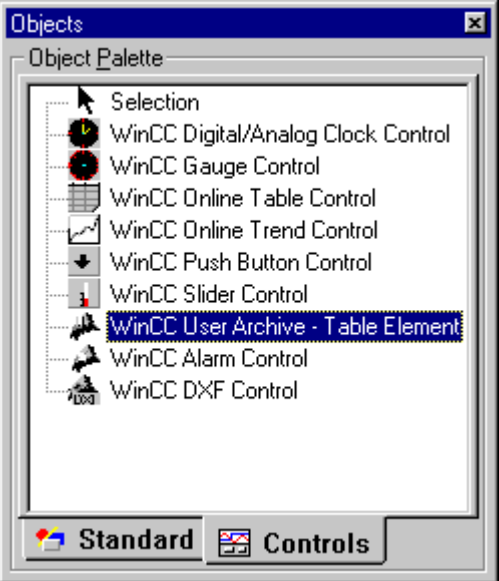

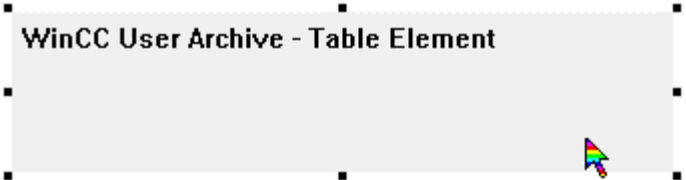
任务定义

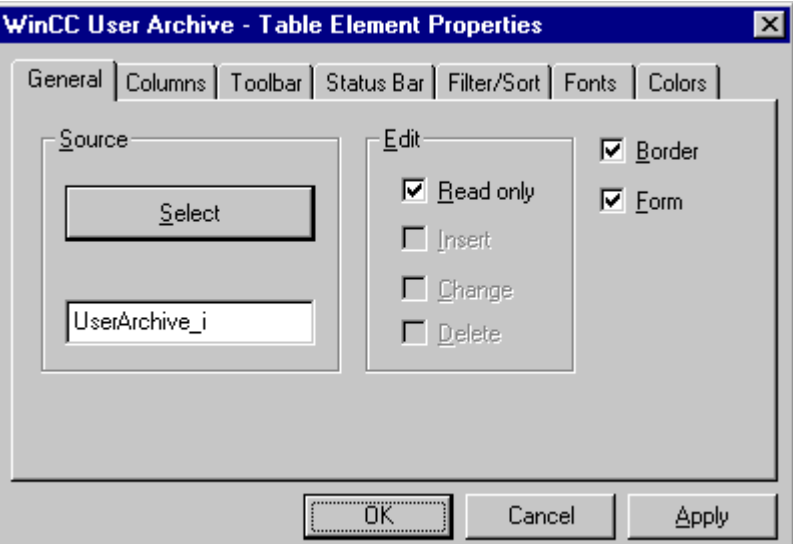
本实例基于先前的实例用户归档的创建(ex_3_chapter_01.PDL)与数据输入(ex_3_chapter_01.PDL)。在这些实例中组态的归档数据可供用户在运行系统中使用。将以表单的形式显示归档数据，每个表单显示一条数据记录。




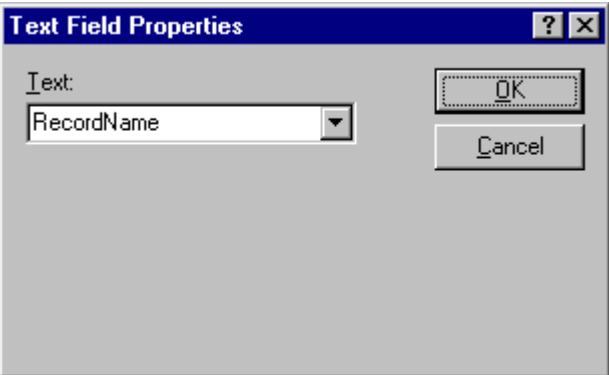
概念的实现


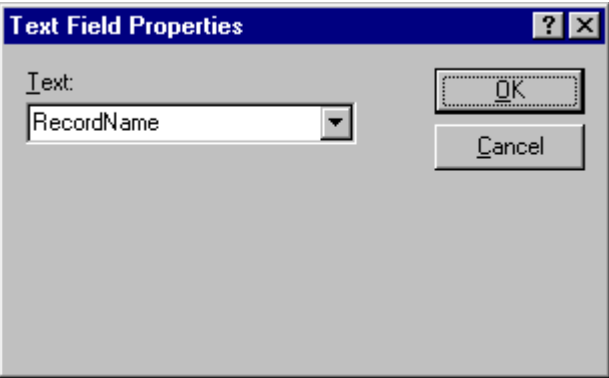
在运行系统中，可通过一个特定的控件来对归档进行显示。数据以表单形式显示。通过多个按钮，用户可使用各种用于归档的编辑选项。


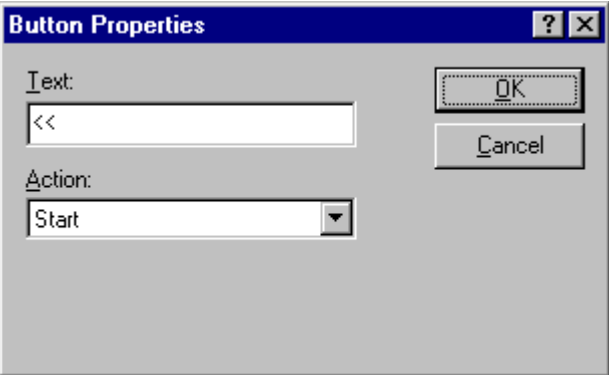


在图形编辑器中的实现

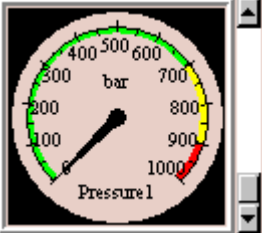
步骤	过程：在图形编辑器中的实现
1	打开 <i>图形编辑器</i> 并创建一个新画面。在本实例中，它就是画面 <i>ex_3_chapter_011.pdl</i> 。
2	组态用于显示数据的控件。它是 <i>WinCC 用户归档 - 表格元素</i> 。从 <i>对象选项板</i> 的 <i>控件</i> 选择菜单中选择它，然后将其置于画面中。 <div></div>
3	通过  刚创建的对象 <i>控件 1</i> ，打开其属性对话框。 <div></div>

步骤	过程：在图形编辑器中的实现
4	<p>在 <i>常规信息</i> 标签中，在 <i>源域</i> 中选择 <i>归档</i> 选项。在本实例中，不能选择 <i>视图</i>，因为还未对其进行组态。至于归档，则选择 <i>UserArchive_i</i>。</p> <p>在 <i>编辑域</i> 中，可指定用户对归档进行访问的编辑选项。在本实例中，选择了复选框 <i>只读</i>，以防止用户修改归档数据。</p> <p>控件可带 <i>边框</i> 显示。此外，可使用 <i>表单视图</i> 的方式来显示对象。在本实例中，选择了 <i>边框与表单</i> 复选框。</p> 
5	<p>在 <i>工具栏</i> 标签中，可组态一个工具栏。但在本实例中，选择的是 <i>关闭</i> 复选框。因此，在运行系统中不会显示任何工具栏。</p> <p>在 <i>状态栏</i> 标签中，可组态一个状态栏。在前面的标签中，选择了复选框 <i>关闭</i>。因此，在运行系统中也不会显示状态栏。</p>
6	<p>在其余标签中，本实例将不进行任何设置。</p> <p>在 <i>WinCC 用户归档 - 表格元素</i> 的属性对话框中所作的设置通过 <i>确定</i> 按钮来结束。</p>

步骤	过程：在图形编辑器中的实现
7	<p>为了对表单视图进行组态，必须在图形编辑器中激活控件。这可通过在按下 CTRL 键的同时用  对象来实现。</p> <p>通过  激活的对象，可访问一个弹出菜单，它允许将文本域、输入域或按钮插入到表单中。通过 <i>插入文本域</i> 菜单，可输入一个文本域。</p> 
8	<p>在表单中显示文本框，同时其属性对话框被打开。可在文本域中输入任何文本。而且，可从下拉列表中选择数据域的名称或别名。</p> <p>在本实例中，从文本域的下拉列表中选择 RecordName 域的别名 @TXT:40。单击 <i>确定</i> 按钮关闭对话框。</p> 

步骤	过程：在图形编辑器中的实现												
9	<p>组态输入域，使其适合文本域。这可通过使用  激活的控件，然后选择 <i>插入编辑域</i> 来完成。</p> <p>在表单中显示输入域，同时其属性对话框被打开。可从下拉列表中选择所期望的域，其数据将进行显示或输入。</p> <p>在本实例中，为 <i>域</i> 选择 <i>RecordName</i> 域。单击 <i>确定</i> 按钮关闭对话框。</p> 												
10	<p>在表单中显示输入域，同时其属性对话框被打开。现在即可从下拉列表中选择所期望的域，其数据将进行显示或输入。</p> <p>在本实例中，为 <i>域</i> 选择 <i>RecordName</i> 域。</p>												
11	<p>按照前面所描述的四个步骤，可为每个归档域组态一个文本和输入域。</p> <table><tr><td>RecordName</td><td>RecordName</td></tr><tr><td>Pressure_a</td><td>Pressure_a</td></tr><tr><td>Pressure_b</td><td>Pressure_b</td></tr><tr><td>Pressure_c</td><td>Pressure_c</td></tr><tr><td>Pressure_d</td><td>Pressure_d</td></tr><tr><td>Pressure_e</td><td>Pressure_e</td></tr></table>	RecordName	RecordName	Pressure_a	Pressure_a	Pressure_b	Pressure_b	Pressure_c	Pressure_c	Pressure_d	Pressure_d	Pressure_e	Pressure_e
RecordName	RecordName												
Pressure_a	Pressure_a												
Pressure_b	Pressure_b												
Pressure_c	Pressure_c												
Pressure_d	Pressure_d												
Pressure_e	Pressure_e												

步骤	过程：在图形编辑器中的实现
12	<p>可在归档域间移动。这可通过组态按钮来实现。在表单中，工具栏可使用的所有功能均可使用按钮来完成。</p> <p>通过  激活的控件并选择 插入按钮，可将一个 按钮 插入到表单中。</p> <p>在表单中显示该按钮，同时其属性对话框被打开。在文本输入域中，可输入任何文本作为按钮标签。在本实例中，使用了字符串作为文本。该按钮可使跳转到归档的第一条数据记录的功能符号化。</p> <p>在 动作 域中，对将由按钮完成的功能进行设置。在下拉列表中，列出所有可用的功能。在本实例中，指定 启动 功能。</p> 
13	<p>另外组态三个按钮。它们可完成功能 向后、向前 与 结束。</p>  <p>而且，组态了两个按钮，用来允许用户将单个数据记录写入变量，并从变量中读出记录。为这些按钮所设置的功能是 读变量 与 写变量。</p> 

步骤	过程：在图形编辑器中的实现
14	<p>为了显示由用户归档写入的变量值，本实例使用了五个指针仪。</p> <p>使用 WinCC 标准控件类型的控件作为指针仪。在本实例中，它们是控件 2 至控件 6 对象。</p> <p>对于这每个对象，在属性 → 控件属性 → 值处创建一个与这五个整型变量中的一个变量(由归档写入)的变量连接。</p> <p>为了对这些变量的数值变化进行模拟，组态了五个智能对象 → 滚动条对象。在本实例中，它们是滚动条对象 1 至滚动条对象 5。</p> <p>对这些对象，可在属性 → 其它 → 过程驱动程序连接处，创建与其中一个相关变量的变量连接。而且，在事件 → 属性主题 → 其它 → 过程驱动程序连接 → 修改处创建与相同变量的直接连接。对这些直接连接，将源此对象 → 过程驱动程序连接与相关变量的目标相连接。完成该操作可使变量值在每次滚动条位置发生变化时，立即进行修改。</p> 
15	<p>为了显示当前装载到变量中的数据记录，组态了一个标准对象 → 静态文本。</p> <p>在本实例中，这是静态文本 2 对象。为该对象，在属性 → 字体 → 文本处创建了一个与包含有数据记录名称的变量 T08i_ex_UAi_01 的变量连接。</p>

5.1.5 使用控制变量进行工作(ex_3_chapter_012.PDL)



任务定义

创建一个归档，其数据记录包括三个浮点型数字域以及一个用于记录数据记录名称的文本域。数据记录的数量限制为 3 个。通过应用控制变量来记录可以使归档数据应用于整个项目记录。

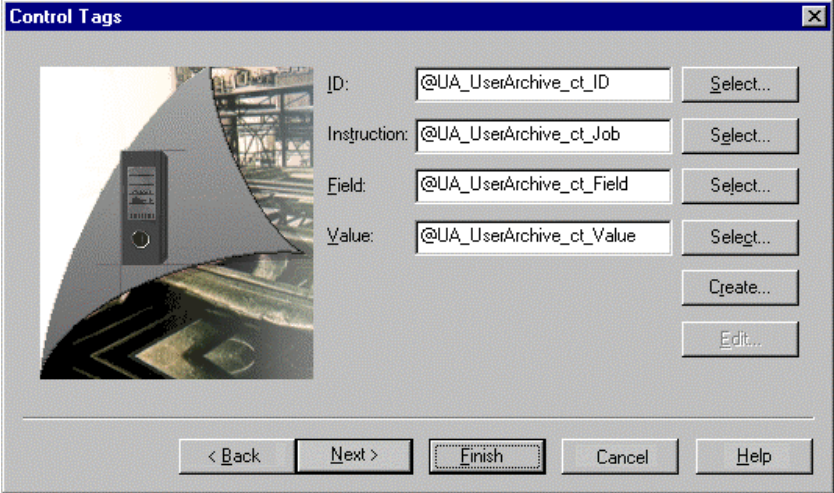
概念的实现

为了对数据进行归档，在用户归档编辑器中创建归档。使用 WinCC 变量，组态归档的通讯。在该归档中，创建 4 个所需域类型的数据域。给每个数据域分配一个内部变量，它可以允许归档与其余项目进行通讯。
将 4 个控制变量分配给归档。目的是控制数据写入 WinCC 变量以及从 WinCC 变量读取数据。在图形编辑器中，组态多个按钮和 I/O 域，用于写入控制变量。控制变量的内容定义要编辑哪个数据记录以及是否读取或写入数据。


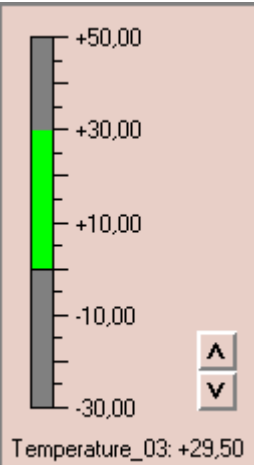
创建归档

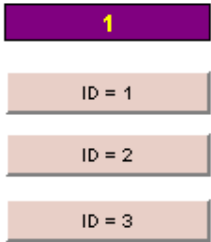
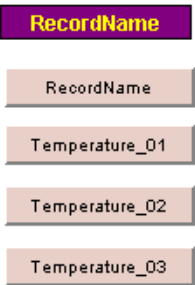

步骤	过程：创建归档
1	为归档的每个数据记录创建一个内部变量。由这些变量来执行归档和其余系统之间的通讯。 在本实例中，使用 IEEE 754 64 位浮点数类型的变量 G64i_ex_UAct_01 至 G64i_ex_UAct_03。此外，创建一个 8 位字符集文本变量类型的变量来存储数据记录名。在本实例中，它是变量 T08i_ex_UAct_01。
2	打开用户归档编辑器。在该编辑器中，创建一个新的归档。为此通过  归档条目来启动向导。 
3	在该向导的第一页上，输入归档名称。在本实例中，将名称 UserArchive_ct 输入归档名称域内。别名域保留空白。 对于归档类型，则选择限制。根据任务定义，将数据记录的最大数量设置为 3。 通过单击下一步继续到下一页。
4	在向导的第二页上，选择通讯类型。在本实例中，选择类型通过 WinCC 变量进行通讯。 通过单击下一步继续到下一页

步骤	过程：创建归档
5	<p>在向导的第三页上，可以设置 4 个控制变量。通过这些变量，可以在运行系统中控制归档。</p> <p>如果预先没有创建变量，则可以由向导自动执行该任务。通过 <i>创建</i> 按钮，可以访问一个对话框，在其中完成与要创建的变量相关的设置。</p> <p>指定是否要创建内部变量或外部变量。如果向导要创建外部变量，则也必须指定为这些变量所创建的 <i>连接</i>。</p> <p>向导在变量组中创建变量。必须指定组的名称以及各个变量的名称。向导也会做出建议。</p> <p>在本实例中，选择变量类型 <i>内部</i>。采纳向导建议的变量组名称和各个变量的名称。通过 <i>确定</i>，关闭 <i>创建控制变量</i> 对话框。</p> 

步骤	过程：创建归档																														
	<p>刚创建的变量自动输入到相应的域中，即 <i>ID</i>、<i>作业</i>、<i>域</i>和<i>数值</i>。 由于在向导的下一页中不必为本实例做任何设置，所以可通过单击<i>完成</i>退出向导。</p> 																														
6	<p>为刚创建的归档，创建列于下表的域。将步骤 1 中所组态的变量设置为要分配的变量。 对于浮点型数字域，将<i>最小值</i>设置为-30 并将<i>最大值</i>设置为 50。将文本域的<i>长度</i>设置为 10。</p> <table> <tr> <th>Name</th><th>Art</th><th>Länge</th><th>Minimalwert</th><th>Maximalwert</th><th>Variablenname</th></tr> <tr> <td> Temperature_03</td><td>Zahl (double)</td><td></td><td>-30</td><td>50</td><td>G64i_ex_UAct_03</td></tr> <tr> <td> Temperature_02</td><td>Zahl (double)</td><td></td><td>-30</td><td>50</td><td>G64i_ex_UAct_02</td></tr> <tr> <td> Temperature_01</td><td>Zahl (double)</td><td></td><td>-30</td><td>50</td><td>G64i_ex_UAct_01</td></tr> <tr> <td> RecordName</td><td>Zeichenkette</td><td>10</td><td></td><td></td><td>T08i_ex_UAct_01</td></tr> </table>	Name	Art	Länge	Minimalwert	Maximalwert	Variablenname	Temperature_03	Zahl (double)		-30	50	G64i_ex_UAct_03	Temperature_02	Zahl (double)		-30	50	G64i_ex_UAct_02	Temperature_01	Zahl (double)		-30	50	G64i_ex_UAct_01	RecordName	Zeichenkette	10			T08i_ex_UAct_01
Name	Art	Länge	Minimalwert	Maximalwert	Variablenname																										
Temperature_03	Zahl (double)		-30	50	G64i_ex_UAct_03																										
Temperature_02	Zahl (double)		-30	50	G64i_ex_UAct_02																										
Temperature_01	Zahl (double)		-30	50	G64i_ex_UAct_01																										
RecordName	Zeichenkette	10			T08i_ex_UAct_01																										
7	<p>在<i>用户归档编辑器</i>的下部表格窗口中，现在可以为归档创建总共 3 个数据记录。</p> <table> <tr> <th></th><th>ID</th><th>RecordName</th><th>Temperature_01</th><th>Temperature_02</th><th>Temperature_03</th></tr> <tr> <td>1</td><td>1</td><td>Record_01</td><td>22,5</td><td>12</td><td>29,5</td></tr> <tr> <td>2</td><td>2</td><td>Record_02</td><td>31</td><td>30,5</td><td>42</td></tr> <tr> <td>3</td><td>3</td><td>Record_03</td><td>-13</td><td>-14,5</td><td>-30</td></tr> <tr> <td>...</td><td></td><td></td><td></td><td></td><td></td></tr> </table>		ID	RecordName	Temperature_01	Temperature_02	Temperature_03	1	1	Record_01	22,5	12	29,5	2	2	Record_02	31	30,5	42	3	3	Record_03	-13	-14,5	-30	...					
	ID	RecordName	Temperature_01	Temperature_02	Temperature_03																										
1	1	Record_01	22,5	12	29,5																										
2	2	Record_02	31	30,5	42																										
3	3	Record_03	-13	-14,5	-30																										
...																															

在图形编辑器中的实现

步骤	过程：在图形编辑器中的实现
1	打开图形编辑器并创建一个新画面。在本实例中，它是画面 <i>ex_3_chapter_012.pdl</i> 。 组态用于显示数据的控件。它是 WinCC 用户归档 - 表格元素。从对象选项板的控件选择菜单中选择它，然后将其置于画面中。
2	通过  刚创建的对象控件 1，打开其属性对话框。 在常规标签中，将预先组态的归档 <i>UserArchive_ct</i> 设置在源域中。在编辑域中，保持只读设置。撤消选择边框复选框。窗体复选框仍保持取消选择状态。 在工具栏标签中，选择关闭复选框。在状态栏标签中，也选择关闭复选框。因此，表格显示时不带工具栏和状态栏。 在 WinCC 用户归档 - 表格元素的属性对话框中所作的设置通过确定按钮来结束。
3	为了显示由用户归档写入的变量，组态 3 个智能对象 → 棒图。在本实例中，它们是对象棒图 1 至棒图 3。 对于其中的每个对象，在属性 → 其它 → 过程驱动程序连接处创建与 3 个由归档写入的浮点型数字变量之一的变量连接。 为了模拟这些变量的数值变化，为每个棒图组态两个 Windows 对象 → 按钮。使用缓动模式，其中一个按钮使相应的变量值递增，另一个则使其递减。 

步骤	过程：在图形编辑器中的实现
4	<p>必须组态控件元素，以便允许用户选择某个数据记录以及执行所选记录的函数。</p> <p>通过控制变量选择数据记录，有两种方法可用。</p> <ul style="list-style-type: none"> • @UA_UserArchive_ct_IDA 可以将数据记录的有效 ID 写入 ID 域内的控制变量，该变量在创建归档时已设置(有关的描述请参见步骤 5)。 • @UA_UserArchive_ct_Field 和 @UA_UserArchive_ct_ValueA 可以将有效的域名写入 域名域内的控制变量，该变量在创建归档时已设置。通过为 数值域中设置的控制变量输入搜索值，选择某个数据记录(有关描述请参见步骤 6)。
5	<p>为了通过其 ID 选择数据记录，在本实例中组态 3 个 Windows 对象 → 按钮。通过 直接连接，它们将相应的数值写入为 ID 设置的变量 @UA_UserArchive_ct_ID 中。通过一个 智能对象 → I/O 域显示变量的当前值。</p> <p>ID</p> 
6	<p>在本实例中为了通过数据记录的域名和搜索值来选择数据记录，组态了 4 个 Windows 对象 → 按钮。通过 直接连接，它们将相应的域名写入为域名设置的变量 @UA_UserArchive_ct_Field 中。通过一个 智能对象 → I/O 域显示变量的当前值。</p> <p>附加的 智能对象 → I/O 域用于将搜索值输入变量 @UA_UserArchive_ct_Value 中。如果将文本用作搜索值，则必须用单引号将它们括起来。</p> <p>Field Name</p>  <p>Search Value</p> 

步骤	过程：在图形编辑器中的实现
7	<p>在本实例中，两个 <i>Windows 对象</i> → <i>按钮</i>用于在两个所描述的数据记录的选择项之间进行切换。缺省情况下，归档通过 ID 来控制。如果要通过域名和搜索值来进行控制，则必须将数值 0 写入变量 @UA_UserArchive_ct_ID 中。</p> <p>通过 二进制变量类型的变量 <i>BINi_ex_UAct_01</i>，一种方式的输入对象切换为不可操作模式。</p> <div><div>Control via ID</div><div>Control via Field</div></div>
8	<p>为了触发当前所选数据记录的作业，组态两个 <i>Windows 对象</i> → <i>按钮</i>。通过 直接连接，它们将相应的数值写入为作业设置的变量 @UA_UserArchive_ct_Job 中。有效数值为：</p> <ul style="list-style-type: none">从变量中读取记录的作业为数值 6将记录写入变量的作业为数值 7从归档中删除记录的作业为数值 8 <p>在本实例中，只使用了读取变量和写入变量的作业。</p> <div><div>Read Record</div><div>Write Record</div></div> <p>通过为作业设置的变量，也可以检查作业的执行是否成功。如果成功完成作业，变量设置为 0。如果作业失败，变量设置为 -1。</p> <p>在本实例中，这通过 智能对象 → <i>I/O 域</i>处的 动态对话框来判断。</p> <p>State</p> <div>OK</div>

5.1.6 通过 WinCC 原始数据变量进行通讯

常规信息

可采用两种不同的方法使用户归档与 PLC 进行通讯。

- *通过 WinCC 变量进行通讯*将归档的每条数据记录分配给 WinCC 变量。数据可写入这些变量或从这些变量中读取。已在前面章节所描述的实例中使用了该通讯类型。
- *通过 WinCC 原始数据变量进行通讯*将 WinCC 原始数据变量分配给归档。通过该变量，PLC 将作业报文发送给归档。归档通过一个确认报文对这些作业报文作出响应。

实例

在该章节后，可找到下面所列出的实例。它们详细说明了在 PLC 和 *用户归档* 之间建立通讯所必需的组态步骤。

- 通过 WinCC 原始数据变量与 SIMATIC S5 进行通讯(ex_3_chapter_01a.pdl)
- 通过 WinCC 原始数据变量与 SIMATIC S7 进行通讯(ex_3_chapter_01b.pdl)

报文

PLC 和 WinCC 之间的通讯是通过交换报文来实现的。在通过 WinCC 原始数据变量与 *用户归档* 进行通讯时，PLC 是主动伙伴。PLC 将作业报文发送给归档，再由归档用确认报文作出应答。

每个作业报文由报文标题和报文主体组成，它包含了实际作业或多个作业。作业自身还包括了可能后跟数据的报文标题。

通过确认报文，通知 PLC 有关作业的执行情况。此外，确认报文还传送请求的数据。

作业报文 - 报文标题

字节编号	描述
0	以字节 LSB 为单位的报文长度(报文长度总共占据了 4 个字节)
1	以字节为单位的报文长度
2	以字节为单位的报文长度
3	以字节 MSB 为单位的报文长度
4	传送类型: 2 对应于 PLC 发送的数据
5	保留
6	作业 LSB 的编号(作业编号总共占据 2 个字节)
7	作业 MSB 的编号
8	归档的 PLCID 的第 1 个字符(PLCID 是 8 个字符长度)
9	归档的 PLCID 的第 2 个字符
10	归档的 PLCID 的第 3 个字符
11	归档的 PLCID 的第 4 个字符
12	归档的 PLCID 的第 5 个字符
13	归档的 PLCID 的第 6 个字符
14	归档的 PLCID 的第 7 个字符
15	归档的 PLCID 的第 8 个字符

在报文标题中所包含的传送类型具有数值 2，其对应于将数据发送至 WinCC。
 由 WinCC 所发送的对作业报文进行响应的确认报文，其传送类型为数值 1。
 报文标题中所包含的 8 个字符长度的归档 PLCID 用于对目标归档进行标识。在创建归档时指定此 PLCID。

作业报文 - 作业

字节编号	描述
0	以字节 LSB 为单位的作业长度(作业长度总共占据了 2 个字节)
1	以字节 MSB 为单位的作业长度
2	作业类型
3	保留
4	域编号 LSB (域编号总共占据 2 个字节)
5	域编号 MSB
6	数据记录编号 LSB (数据记录编号总共占据 4 个字节)
7	数据记录编号
8	数据记录编号
9	数据记录编号 MSB
10	选择标准 LSB (选择标准总共占据 2 个字节)
11	选择标准 MSB
12 至 n	作业数据(PLC 的数据)

在作业中指定的记录和域编号并不是与所有的作业都相关。

记录编号与*用户归档*中的数据记录的 ID 相关。数据记录编号从 1 开始。如果为归档的写作业指定记录编号 0，则新的数据记录将添加至归档。

域编号从 0 开始。

作业类型

在作业中，为作业类型保留一个字节。对于作业类型，可使用下列数值：

作业类型	描述
4	检查归档是否存在
5	从归档中删除所有数据记录
6	读数据记录(从归档)
7	写数据记录(至归档)
8	删除数据记录
9	读数据域(从归档)
10	写数据域(至归档)

作业数据

作业数据与用于作业写数据记录的数据记录的内容相关，或与用于作业写数据域的数据域的内容相关。对其它作业类型，不传送作业数据。
以 Intel 格式显示编号值(如在报文中指定的所有编号值)，即 LSB 最先传送，MSB 最后传送。
在用户归档中组态的整型域具有 4 个字节长度，浮点型数具有 8 个字节长度。文本域不是以 0 终止的。

确认报文

字节编号	描述
0	以字节 LSB 为单位的报文长度(报文长度总共占据了 4 个字节)
1	以字节为单位的报文长度
2	以字节为单位的报文长度
3	以字节 MSB 为单位的报文长度
4	传送类型：1 对应于 WinCC 发送的数据
5	保留
6	错误代码：请参考下列有关错误代码的描述
7	作业类型：请参考前面有关作业类型的描述
8	保留
9	保留
10	域编号 LSB (域编号总共占据 2 个字节)
11	域编号 MSB
12	数据记录编号 LSB (数据记录编号总共占据 4 个字节)
13	数据记录编号
14	数据记录编号
15	数据记录编号 MSB
16	归档的 PLCID 的第 1 个字符(PLCID 是 8 个字符长度)
17	归档的 PLCID 的第 2 个字符
18	归档的 PLCID 的第 3 个字符
19	归档的 PLCID 的第 4 个字符
20	归档的 PLCID 的第 5 个字符
21	归档的 PLCID 的第 6 个字符
22	归档的 PLCID 的第 7 个字符
23	归档的 PLCID 的第 8 个字符
24 至 n	确认数据(归档数据)

在确认报文中所包含的传送类型具有数值 1，其对应于由 WinCC 所发送的数据。由 PLC 发送的作业报文包含了数值 2 作为传送类型。
确认报文中所包含的 8 个字符长度的归档 PLCID 用于对目标归档进行标识。在创建归档时指定此 PLCID。

错误代码

如果产生错误，则 WinCC 将在确认报文中返回一个错误代码给 PLC。已定义了下列错误代码：

错误代码	描述
0	功能已被执行
1	无效数据(归档)
2	数据不可用(归档)
101	无效数据(数据记录)
102	数据不可用(数据记录)
201	无效数据(域)
202	数据不可用(域)
254	功能不可用
255	未定义的错误

确认数据

确认数据与用于作业*读数据记录*的数据记录的内容相关，或与用于作业*读数据域*的数据域的内容相关。对其它作业类型，不传送确认数据。

以 Intel 格式显示编号值(如在报文中指定的所有编号值)，即 LSB 最先传送，MSB 最后传送。

在*用户归档*中组态的整型域具有 4 个字节长度，浮点型数具有 8 个字节长度。文本域不是以 0 终止的。

5.1.7 通过 WinCC 原始数据变量与 SIMATIC S5 进行通讯 (ex_3_chapter_01a.pdl)

本章中所创建的 STEP5 项目也可直接从在线文档复制到用户的硬盘上。缺省情况下，它将存储在 *C:\Configuration_Manual* 文件夹中。



S5_UA_st

任务定义

在 SIMATIC S5 PLC 和 WinCC 站之间建立通讯连接。PLC 从在 WinCC 站上创建的用户归档中读取数据、将数据写入其中以及从中删除数据。
归档的每个数据记录包括两个整数域以及一个用于记录数据记录名的文本域。

概念的实现

在本实例中，使用带有 *CPU 944* 中心模块的 *SIMATIC S5-115U PLC*。使用 *工业以太网* 建立与该 PLC 的通讯。为此，WinCC 站使用 *CP 1413* 通讯处理器，PLC 使用 *CP 1430 TF* 通讯处理器。

在 WinCC 中，使用 *SIMATIC S5 ETHERNET LAYER 4* 通讯驱动程序。其中，该通讯处理器支持从 PLC 中主动发送数据。

为通讯驱动程序创建两个与 PLC 的连接。用于从 PLC 中主动发送数据一个连接。为该连接，创建一个用于与 *用户归档* 进行通讯的 *WinCC 原始数据变量*。此外，创建一个 *用户归档*，其通讯通过该 *WinCC 原始数据变量* 来组态。

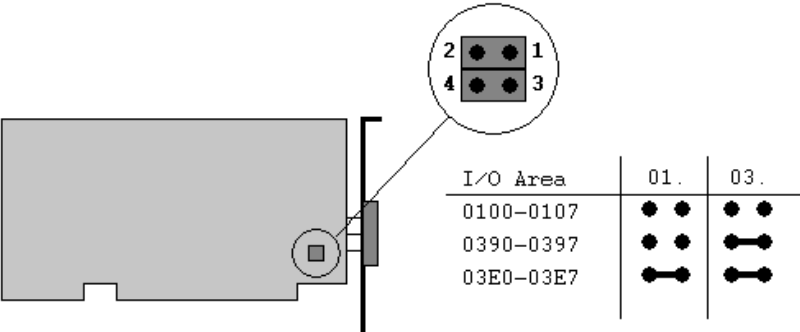
对于第二个连接，WinCC 作为主动伙伴。其中，该连接用于模拟 WinCC 中的通讯操作。通过各种 *按钮*，在 PLC 中设置控制位，用于触发作业报文的发送。另外，PLC 中现有的数据(当前数据记录、当前设置的记录编号和域编号以及作业状态)通过 *I/O 域* 来显示。

在运行系统中，*用户归档* 的数据通过控件以列表形式显示。

5.1.7.1 通讯处理器 CP 1413 的启动

以下详细描述成功启动通讯处理器 CP 1413必需的组态步骤。

在计算机中安装通讯处理器

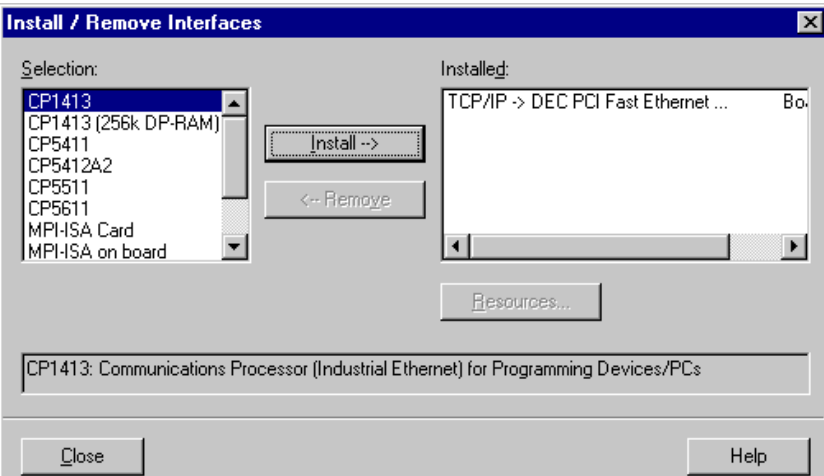
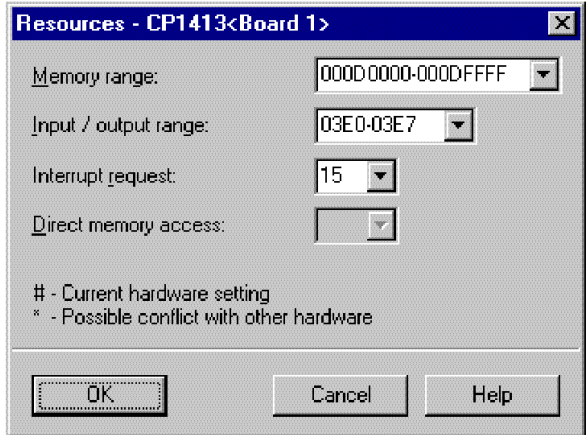
步骤	过程：在计算机中安装通讯处理器												
1	<p>在 CP 1413 处检查所选择的跳线设置。</p> <p>在进行 CP 1413 的软件安装时，必须指定 I/O 范围。该范围通过跳接设置来设定。</p> <p>在缺省状态下，I/O 范围设置为 03E0-03E7。也可设置为 0100-0117 和 0390-0397。下面的图形说明了各种 I/O 范围所需的跳线设置。</p> <div><table><tr><th>I/O Area</th><th>01.</th><th>03.</th></tr><tr><td>0100-0107</td><td>● ●</td><td>● ●</td></tr><tr><td>0390-0397</td><td>● ●</td><td>— ●</td></tr><tr><td>03E0-03E7</td><td>— ●</td><td>— ●</td></tr></table></div>	I/O Area	01.	03.	0100-0107	● ●	● ●	0390-0397	● ●	— ●	03E0-03E7	— ●	— ●
I/O Area	01.	03.											
0100-0107	● ●	● ●											
0390-0397	● ●	— ●											
03E0-03E7	— ●	— ●											
2	<p>根据安装说明安装模块。其中还要遵守处理静电敏感设备(ESD)的步骤。必须在计算机关闭时才能安装模块。</p> <p>对于通讯卡 CP 1413，需要计算机内有空闲的 ISA 插槽。安装 CP 1413 之后，将计算机的机箱盖上然后启动计算机。</p>												

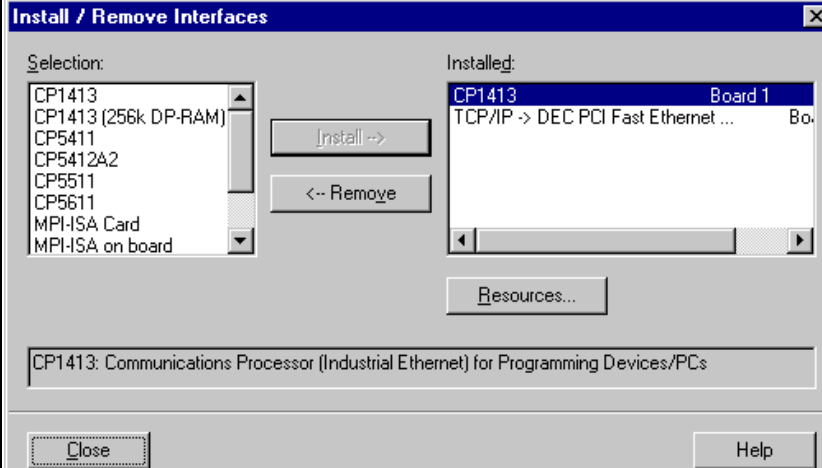
安装通讯驱动程序

步骤	过程：安装通讯驱动程序												
1	<p>从 <i>SIMATIC NET</i> 光盘上安装通讯驱动程序 <i>IE TF-1413</i>。</p> <p>在插入 <i>SIMATIC NET</i> 光盘后，将自动启动安装程序。如果没有自动启动，则打开 <i>Windows NT</i> 资源管理器，然后启动光盘上的 <i>setup.exe</i> 程序。</p> <p>通过如下显示的按钮启动软件的安装。</p> <div data-bbox="548 504 774 585"><p>SIMATIC NET Software Installieren</p></div> <p>按照安装程序的说明。在 <i>组件</i> 页上，必须选中要安装的驱动程序 <i>IE TF-1413</i> 的复选框。完成安装。</p> <div data-bbox="527 665 1349 1262"><p>SIMATIC NET Software: Components</p><p>Please select the programs to be installed.</p><table><tr><td><input type="checkbox"/> AuthorsW V2.0.1</td><td>2 MB</td></tr><tr><td><input type="checkbox"/> IE PG-1413 V5.1</td><td>17 MB</td></tr><tr><td><input type="checkbox"/> IE S7-1413 V5.1</td><td>17 MB</td></tr><tr><td><input checked="" type="checkbox"/> IE TF-1413 V5.1</td><td>17 MB</td></tr><tr><td><input type="checkbox"/> IE SOFTNET PG V3.1</td><td>5 MB</td></tr><tr><td><input type="checkbox"/> IE SOFTNET-S7 BASIC V3.1</td><td>5 MB</td></tr></table><p>Description: SIMATIC NET Industrial Ethernet TF-1413/Windows NT Version 5.1.</p><p>Destination directory: c:\siemens\simatic.net</p><p>Required on c: 17 MByte Available on c: 513 MByte</p><p>< Back Next > Cancel</p></div>	<input type="checkbox"/> AuthorsW V2.0.1	2 MB	<input type="checkbox"/> IE PG-1413 V5.1	17 MB	<input type="checkbox"/> IE S7-1413 V5.1	17 MB	<input checked="" type="checkbox"/> IE TF-1413 V5.1	17 MB	<input type="checkbox"/> IE SOFTNET PG V3.1	5 MB	<input type="checkbox"/> IE SOFTNET-S7 BASIC V3.1	5 MB
<input type="checkbox"/> AuthorsW V2.0.1	2 MB												
<input type="checkbox"/> IE PG-1413 V5.1	17 MB												
<input type="checkbox"/> IE S7-1413 V5.1	17 MB												
<input checked="" type="checkbox"/> IE TF-1413 V5.1	17 MB												
<input type="checkbox"/> IE SOFTNET PG V3.1	5 MB												
<input type="checkbox"/> IE SOFTNET-S7 BASIC V3.1	5 MB												

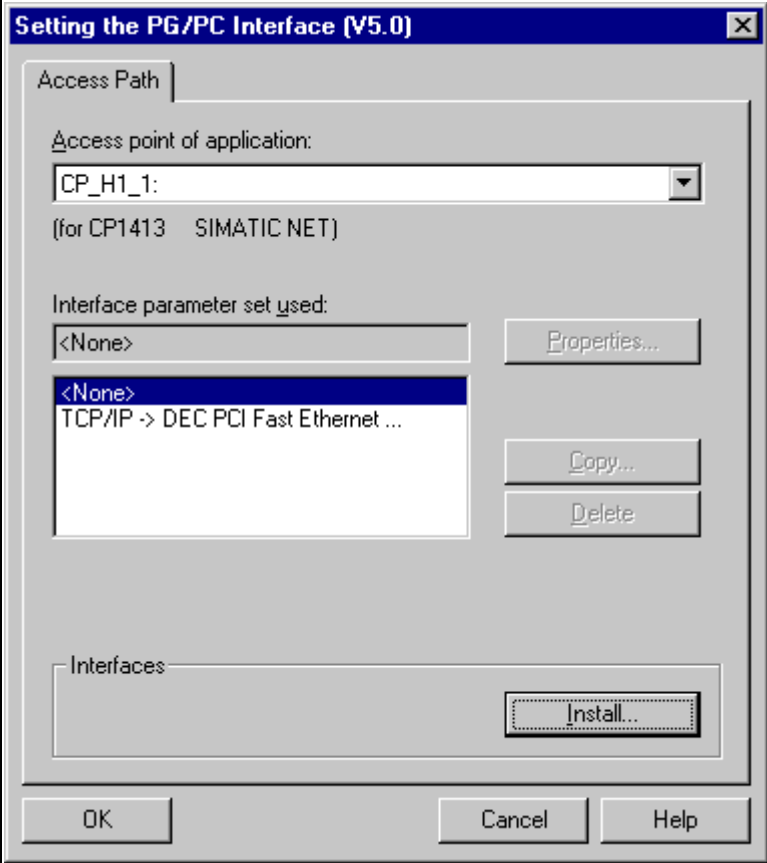
安装通讯处理器

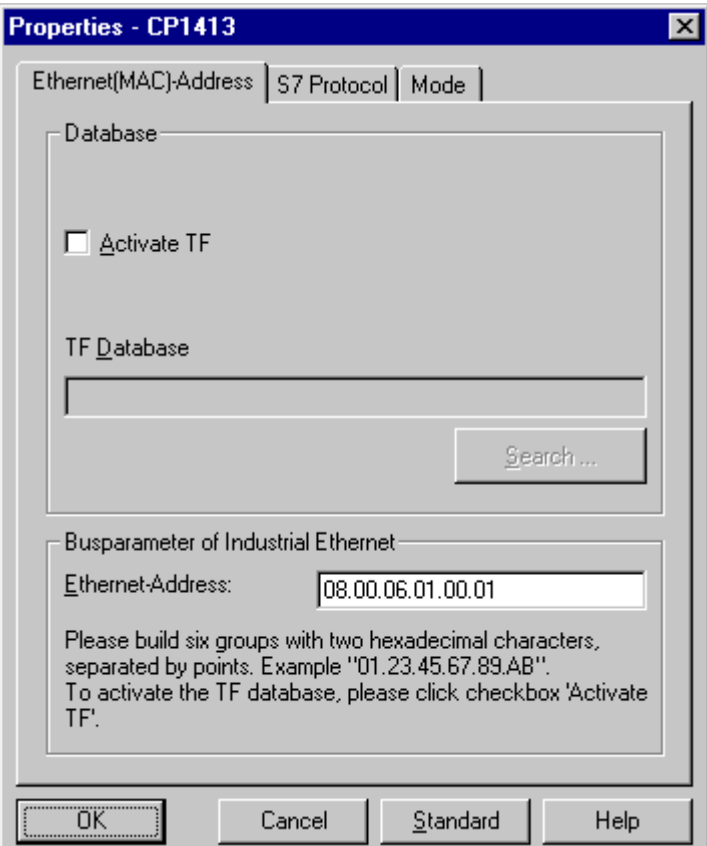
步骤	过程：安装通讯处理器
1	<p>通过 设置 PG/PC 接口程序来安装通讯处理器 CP 1413。</p> <p>通过 开始 → 设置 → 控制面板 → 设置 PG/PC 接口访问该程序。</p> <div data-bbox="586 443 651 506" data-label="Image"> </div> <p>Setting the PG/PC Interface</p>
2	<p>将显示程序 设置 PG/PC 接口。</p> <p>通过 安装按钮打开安装新接口的对话框。</p> <div data-bbox="529 632 1284 1476" data-label="Image"> </div>

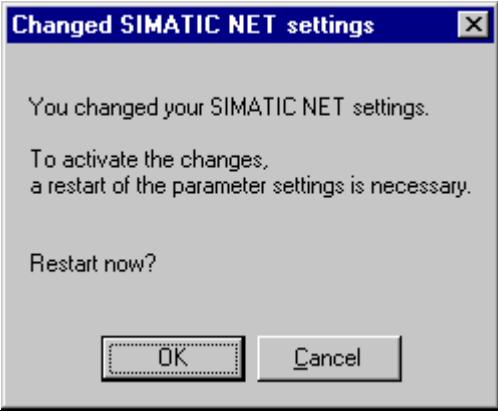
步骤	过程：安装通讯处理器
3	<p>将显示对话框 <i>安装/删除模块</i>。选择域列出了所有可以安装的接口。如果先前已经按步骤 B 中描述的方式安装了通讯驱动程序，则在其中将会有条目 <i>CP 1413</i>。</p> <p>从选择域中选择条目 <i>CP 1413</i>。单击 <i>安装-></i>按钮来启动通讯处理器的安装。</p> 
4	<p>将显示 <i>资源 - CP 1413</i>对话框。</p> <p>必须指定 <i>存储器范围</i>、<i>I/O 范围</i>和 <i>中断</i>的设置。</p> <p><i>I/O 范围</i>通过 <i>CP 1413</i>处的跳线设置已经确定。</p> <p>确保所分配的资源还未被计算机中的其它模块占用。有关已占用的系统资源的信息可以通过 <i>开始 → 程序 → 管理工具(公用) → Windows NT 诊断器</i>来访问资源标签而获得。</p> <p>单击 <i>确定</i>关闭 <i>资源</i>标签。</p> 

步骤	过程：安装通讯处理器
5	<p data-bbox="521 302 1193 359">在安装/删除模块对话框中，已安装域现在将包含 <i>CP 1413</i> 条目。 通过 关闭按钮，退出 安装/删除模块对话框。</p> 

分配通讯处理器

步骤	过程：分配通讯处理器
1	<p>在设置 PG/PC 接口程序中，将访问点 CP_H1_1:分配给刚安装的接口。</p> <p>访问点 CP_H1_1:是由 WinCC 通过工业以太网进行通讯所使用的缺省访问点。它在安装通讯驱动程序 IE TF-1413 时已自动创建。</p> <p>在应用程序的访问点域内，设置条目 CP_H1_1:。在下面的域内，选择条目 CP1413。这样就完成了访问点和通讯处理器之间的分配。</p> 

步骤	过程：分配通讯处理器
2	<p>设置通讯处理器 CP 1413 的属性。</p> <p>通过 设置 PG/PC 接口程序的 属性按钮可打开设置属性的对话框。</p> <p>将显示通讯处理器 CP 1413 的属性对话框。</p> <p>在以太网(MAC)地址标签内，输入 CP 1413 的以太网地址。在我们的实例中是 08.00.06.01.00.01。</p> 

步骤	过程：分配通讯处理器
3	<p>通过 确定按钮，退出 <i>设置 PG/PC 接口程序</i>。</p> <p>将显示一个请求重新启动 CP 1413 的对话框。单击确定确认该对话框，以重新启动通讯处理器 <i>CP 1413</i>。</p> <p>这样就完成了通讯处理器的安装。</p> <div></div>
4	<p>如果重新启动通讯处理器未成功，则必须确定出错原因并予以纠正。有关该主题的帮助信息可参见<i>通讯手册</i>。</p>

5.1.7.2 PLC 的启动

以下详细说明了创建和启动 STEP5 项目 *S5_UA_st*所需的组态步骤。

安装硬件

步骤	过程：安装硬件和软件
1	<p>在机架上安装使用的模块。</p> <p>在本实例中，要安装的模块是电源 <i>PS 951</i>、CPU 模块 <i>CPU 944</i> 和通讯处理器 <i>CP 1430 TF</i>。</p> <p>建立从编程设备至 CPU 模块的编程接口的连接。</p> <p>建立从计算机中通讯处理器 <i>CP 1413</i> 至 PLC 中通讯处理器 <i>CP 1430 TF</i> 的连接。</p>

创建 STEP5 程序

步骤	过程: 创建 STEP5 程序
1	<p>用 STEP5 软件创建新项目。</p> <p>启动 STEP5 软件。从对象 → 项目 → 设置 → 第 1 页和第 2 页菜单中, 定义新项目的设置。在程序文件域内, 指定要创建的新程序文件的名称。在本实例中, 使用名称 UA_S5_ST.S5D。只有文件名的前 6 个字符可以由用户更改。</p>
2	<p>编制启动块。对于本实例中所使用的 SIMATIC S5 115U PLC, 这些块为组织块 OB21 和 OB22。</p> <p>在启动 PLC 时, 必须使通讯处理器 CP 1430 TF 同步。这可由数据处理块 SYNCHRON 来完成。对于本实例中所使用的 SIMATIC S5 115U PLC, 它是 FB249 块。将通讯处理器的接口编号、所期望的块大小以及参数化的错误字节作为参数传送给这个块。</p> <p>下面描述了完整的 OB21 组织块。</p> <pre>OB 21 NETZWERK 1 von 1 Synchronize CP 1430 TF :SPA FB 249 NAME :SYNCHRON SSNR : KY 000.000 BLGR : KY 000.000 PAFE : MB 110 :BE</pre>
3	<p>创建用于 PLC 的被动通讯的程序块。</p> <p>在 STEP5 中, 新程序块的创建通过编辑器 → STEP5 块 → 程序文件的菜单来执行。本实例中使用 PB100 作为程序块的名称。</p> <p>通过调用处理块 SEND 和 RECEIVE 来执行被动通讯。对于本实例中所使用的 SIMATIC S5 115U PLC, 这些块为 FB244 和 FB245。对于作业号 A-NR, 将 0 赋值给这些块以允许执行 SEND ALL 和 RECEIVE ALL 功能。</p> <p>必须在周期性的程序(在 OB1 中)中调用 PB100 程序块。下面描述了完整的 PB100。</p> <pre>PB 100 NETZWERK 1 von 1 Passive PLC Communication :SPA FB 244 SEND ALL NAME :SEND SSNR : KY 000.000 A-NR : KY 000.000 ANZW : MW 100 QIYP : KC NM DBNR : KY 000.000 QANF : KF +0 QLAE : KF +0 PAFE : MB 104 :SPA FB 245 RECEIVE ALL NAME :RECEIVE SSNR : KY 000.000 A-NR : KY 000.000 ANZW : MW 105 ZIYP : KC NM DBNR : KY 000.000 ZANF : KF +0 ZLAE : KF +0 PAFE : MB 109 :BE</pre>

步骤	过程：创建 STEP5 程序
4	<p>创建用于报文数据的数据块。</p> <p>在 STEP5 中，这可通过 编辑器 → 数据块 → 程序文件 的菜单来完成。本实例使用 <i>DB25</i> 至 <i>DB30</i> 作为数据块的名称。</p> <p>这些数据块包含各种作业的报文数据。为每个完成的作业，本实例使用独立的数据块。记录编号和域编号的数据范围在初始阶段为空。仅在报文发送前才用当前数值来填充。同样，只有在报文发送前才输入可能需要的作业数据。</p> <p>下面描述了完整的 <i>DB26</i> 数据块。该数据块包含了用于读取<i>用户归档</i>中数据记录的作业的报文数据。</p> <div><div>DB 26</div><div>Telegram: Read Record</div><div><div>0:KY = 028.000</div><div>1:KY = 000.000</div><div>2:KY = 002.000</div><div>3:KY = 001.000</div><div>4:KC = 'USRARCRD'</div><div>8:KY = 012.000</div><div>9:KY = 006.000</div><div>10:KY = 000.000</div><div>11:KY = 000.000</div><div>12:KY = 000.000</div><div>13:KY = 000.000</div><div>14:KH = 0000</div><div>15:KH = 0000</div><div>Telegramlength low</div><div>Telegramlength high</div><div>Transfertyp. Reserved</div><div>Number of Jobs</div><div>PLCID</div><div>Joblength</div><div>Jobtyp. Reserved</div><div>Fieldnumber</div><div>Recordnumber low</div><div>Recordnumber high</div><div>Selection</div></div></div>
5	<p>创建与<i>用户归档</i>进行主动通讯的数据块。</p> <p>本实例中使用 <i>DB23</i> 作为数据块的名称。</p> <p>在该数据块中，访问为<i>用户归档</i>所创建的 <i>WinCC</i> 原始数据变量。由该数据块发送作业报文，WinCC 发送确认报文给该数据块。</p> <p>数据块的大小取决于最大报文长度。在本实例中，其长度为 46 个字节，将在把数据记录的作业写入归档时产生该报文。该长度由 16 字节的报文头、12 字节的作业和 18 字节的作业数据组成。</p>
6	<p>创建用于删除 <i>DB23</i> 的程序块。</p> <p>本实例中使用 <i>PB111</i> 作为程序块的名称。</p> <p>在将新的作业报文发送给<i>用户归档</i>之前，调用本程序块。</p>

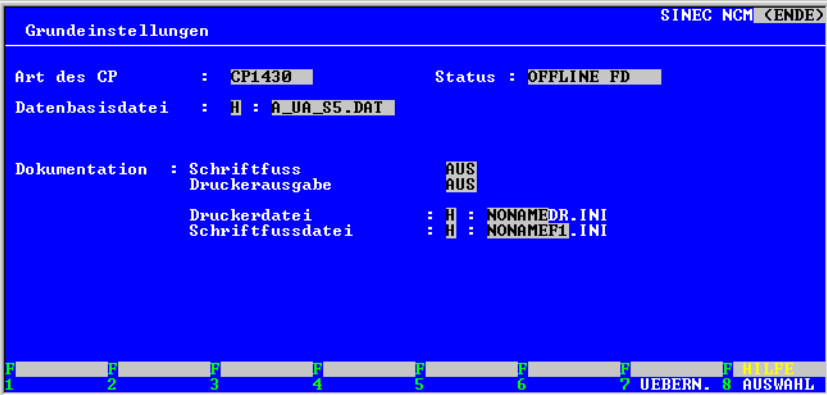
步骤	过程：创建 STEP5 程序
7	<p>另外创建两个数据块。</p> <p>本实例使用 <i>DB5</i> 和 <i>DB21</i> 作为数据块的名称。</p> <p><i>DB21</i> 包含数据记录的数据。该数据与 <i>用户归档</i> 当前所请求的数据记录相关。可以使用 WinCC 变量使该数据可视化，并在运行系统中对其修改以进行模拟。</p> <p><i>DB5</i> 包含用于控制变量的数据范围、域编号以及记录编号。可以使用 WinCC 变量使所有三个数据可视化，并在运行系统中对其修改以进行模拟。</p> <p>此外，<i>DB5</i> 包含了用于 <i>SEND</i> 处理块的发送数据，其执行作业报文至 <i>用户归档</i> 的发送。在定义了作业类型后，发送报文前，输入报文数据的出处(在下面的图例中是输入 <i>From DB Number</i> 这一行)。</p> <p>下面描述了完整的 <i>DB5</i> 数据块。</p> <div><pre>DB 5 0:KM = 00000000 00000000 1:KY = 000.000 2:KY = 000.000 3:KY = 000.000 4:KH = 0000 5:KH = 0000 6:KH = 0000 7:KH = 0000 8:KH = 0000 9:KH = 0000 10:KC = 'DB' 11:KF = +000000 12:KF = +000000 13:KF = +00046 14:KC = 'DB' 15:KF = +00023 16:KF = +000000 17:KF = +00046 18:KH = 0000 19:KH = 0000 Job Record Number Record Number Field Number From Typ DB From DB Number From DW 0 From Length To Typ DB To DB 23 To DW 0 To Length</pre></div>

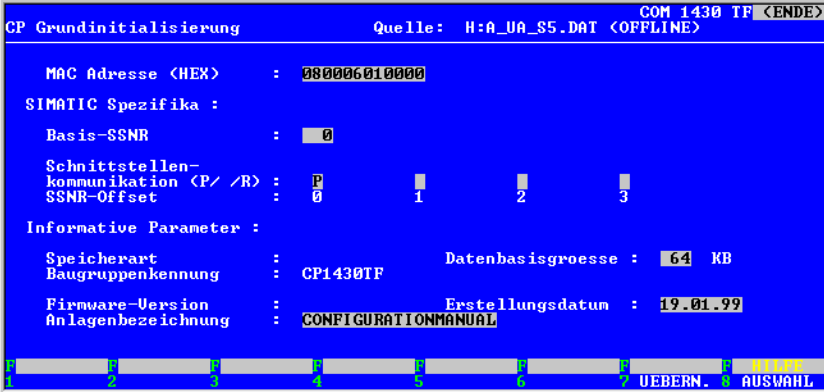
步骤	过程: 创建 STEP5 程序
8	<p>创建功能块用来执行作业报文至<i>用户归档</i>的发送。</p> <p>在 STEP5 中, 新功能块的创建通过<i>编辑器</i> → <i>STEP5 块</i> → <i>程序文件的菜单</i>来执行。本实例中使用 <i>FB111</i> 作为程序块的名称。将它命名为 <i>SENDMSG</i>。在该功能块中, 将存储在 <i>DB5</i> 中的控制变量传送给 <i>MW30</i> 然后将其重新设定为 0。已采用这种方式定义控制值来触发作业, 设置标识位 <i>M31.0</i> 至 <i>M31.5</i>。如果没有一个标识位被设置, 则终止块。</p> <p>如果设置了其中一个标识位, 则实现至作业指定点的跳转。此处, 将包含各作业报文的数据块编号输入 <i>DB5</i>。此外, 将作业报文初始化。如果需要, 输入记录编号、域编号和要发送的作业数据。在这种情况下, 必须将数值的字节顺序反置。</p> <div><pre>FB 111 NETZWERK 1 von 1 NAME :SENDMSG : A DB 5 Load Job in MW30 : L DW 0 : T MW 30 : L KF +0 Clear Job : T DW 0 : U M 31.0 Delete All : SPB -DELA Read Record : U M 31.1 Write Record : SPB -REAR Delete Record : U M 31.2 Read Field : SPB -WRIR : U M 31.3 : SPB -DELA : U M 31.4 : SPB -REAR : U M 31.5 : SPB -WIMP No Job to execute : BE0 DELA : A DB 5 : L KF +25 : T DW 11 : SPA -STMP</pre></div> <p>在完成了作业指定的程序部分后, 跳转至另一个点。此处, 调用用于删除通讯数据块 <i>DB23</i> 的程序块。随后, 通过 <i>SEND</i> 处理块将作业报文发送给 <i>DB23</i>。在发送了作业报文之后, 设置标识位 <i>M32.0</i>。只要确认报文仍未完成, 就会阻止一个附加作业报文的发送。</p> <div><pre>SEND :SPA PB 111 Clear Telegram DB :SPA FB 244 Send Message NAME :SEND SSNR : KY 000.000 A-NR : KY 000.003 ANZW : MW 111 QIVP : KC RW DBNR : KY 000.005 QANF : KF +10 QLAE : KF +0 PAFE : MB 115 : S M 32.0 Set Busy : BE</pre></div> <p>在这个步骤中所采用的图例对应于所描述的 <i>FB111</i> 功能块部分。由于这个块的长度, 所以不显示整个程序。</p>

步骤	过程：创建 STEP5 程序
9	<div><p>创建功能块用来执行<i>用户归档</i>的确认报文的接收。</p><p>本实例中使用 <i>FB112</i> 作为功能块的名称。将它命名为 <i>RECVMMSG</i>。</p><p>功能块检查在 <i>DB23</i> 中输入的传送类型。只要传送类型为数值 1，则就由<i>用户归档</i>接收确认报文。如果情况如此，则可以对 <i>M32.0</i> 标识位进行复位，用来加锁以便防止新作业的发送。随后，读取包含在确认报文中的错误代码。如果错误代码为数值 0，则表示已成功地执行了作业。在成功地执行了从<i>用户归档</i>中读取数据的作业后，将所收到的数据存储在 <i>DB5</i> 中。在这种情况下，必须将数值的字节顺序反置。</p></div> <div><pre>FB 112 NETZWERK 1 von 1 NAME :RECVMMSG : A DB 23 : L DL 2 : L KF +1 : I=F : R M 32.0 : SPB =REC0 : BEA REC0 : L DL 3 : T MB 35 : L KF +0 : I=F : SPB =NOER : BEA NOER : U M 31.1 : SPB =REAR : U M 31.4 : SPB =REAF : BEA If Transfer Type == 1 Telegram arrived Load Error in MB35 If Error Number == 0 No Error Load arrived Record Load arrived Field</pre></div> <p>在这个步骤中所使用的图例对应于所描述的 <i>FB112</i> 功能块部分。由于这个块的长度，所以不显示整个程序。</p>
10	<div><p><i>OB1</i> 中循环程序的编写。</p><p>通过 <i>PB100</i> 程序块，执行与 WinCC 的被动通讯。</p><p>通过 <i>FB111</i> 和 <i>FB112</i> 功能块，执行与<i>用户归档</i>的主动通讯。如果不处理任何作业(<i>M32.0</i> 没有设置)，则使用 <i>FB111</i> 来检查是否必须发送确认报文。如果在处理一个作业(<i>M32.0</i> 进行设置)，则使用 <i>FB112</i> 来检查是否必须接收确认报文。</p><p>下面描述了完整的 <i>OB1</i> 组织块。</p></div> <div><pre>OB 1 NETZWERK 1 von 1 : SPA PB 100 : UN M 32.0 : SPB FB 111 NAME :SENDMSG : U M 32.0 : SPB FB 112 NAME :RECVMMSG : BE Communicate with User Archive Send All/Receive All If not busy Send Message If busy Receive Message</pre></div>

步骤	过程：创建 STEP5 程序
11	将 STEP5 程序装入 PLC 中。 在 STEP5 中，这可通过对象 → 块 → 传送 → PLC 文件菜单来完成。在选择域中，必须选择所有块选项来将所有先前创建的块装入 PLC。

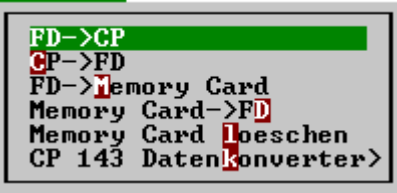
组态通讯处理器

步骤	过程：组态通讯处理器
1	启动通讯程序包 <i>SINEC NCM for COMs</i> 来组态通讯处理器 <i>CP 1430 TF</i> 。 从 STEP5 中，通过更改 → 其它 → <i>SINEC NCM for COMs</i> 菜单启动通讯程序包。
2	这样将打开通讯程序包 <i>SINEC NCM for COMs</i> 。 如果没有设置数据库文件，则在开始时将显示基本设置输入屏。该输入屏也可以通过文件 → 选择或初始化 → 编辑)菜单来打开。 在 <i>CP Type</i> 域中，指明所使用的通讯处理器类型。通过 F8 功能键，可以设置其中一个可用的通讯处理器。选择 <i>CP 1430</i> 。用 F8 功能键将 <i>Status</i> 域设置为 <i>OFFLINE FD</i> 。这样就将程序中所进行的组态存储到数据库文件中。在 <i>Database File</i> 域内，指定该数据库文件的名称。其名称必须以字母 <i>A</i> 开始。在本实例中，数据库文件的名称为 <i>A_UA_S5.DAT</i> 。 基本设置输入屏中所作的设置通过 F7 功能键来实现。 

步骤	过程：组态通讯处理器
3	<p>必须对通讯处理器的基本初始化进行设置。</p> <p>将它们输入基本初始化输入屏。该输入屏通过 编辑 → CP 初始化 菜单来打开。</p> <p>在 MAC Address (HEX) 域内，指定通讯处理器的 以太网地址。在本实例中，已输入了地址 080006010000。该地址是在创建 WinCC 中的连接时必须设置的参数之一。</p> <p>在下图中可以看到其它设置。基本初始化 输入屏中所作的设置通过 F7 功能键来实现。</p> 
5	<p>创建用于与 WinCC 进行被动通讯的传输连接。</p> <p>这在 传输连接 输入屏中完成。该输入屏可以通过 编辑 → 连接 → 传输连接 菜单来打开。</p> <p>对于被动通讯，必须创建两个连接。一个处理 WinCC 的写作业，而另一个则处理 WinCC 的读作业。</p>

步骤	过程：组态通讯处理器
	<p>通过在 <i>Active/Passive</i> 域内输入 <i>P</i>，将 PLC 的两个连接设置为被动。</p> <p>对于用于处理来自 WinCC 的读作业的连接，本实例将 1 作为作业号 <i>ANR</i> 的值。在 <i>Job Type</i> 域中，指定 <i>FETCH</i>。在 <i>Transport Addresses</i> 区域内，以 ASCII 码的形式将 <i>Local Parameter</i> 的 <i>TSAP</i> 设置为 <i>AG_PAS_S</i>，将 <i>Remote Parameter</i> 的 <i>TSAP</i> 设置为 <i>CC_PAS_S</i>。远程参数还需要在 <i>MAC Address</i> 域内指定以太网地址，该 <i>MAC 地址</i> 域已在计算机中为通讯处理器 <i>CP 1413</i> 输入过。本实例中，在安装通讯处理器 <i>CP 1413</i> 时已经设置了地址 <i>080006010001</i>。</p> <div><div>Transportverbindung</div><div>Quelle: H:A_UA_S5.DAT <OFFLINE> COM 1430 TF <ENDE></div><div><div>SSNR-Offset : 0</div><div>ANR : 1</div><div>Auftragsart : FETCH</div><div>Aktiv/Passiv <A/P> : P</div><div>Prioritaet : 2</div><div>Read/Write <J/N> : J</div><div>Anzahl der Auftraege pro TSAP : 1 von 1</div></div><div><div>Transportadressen : lokale Parameter :</div><div>TSAP <ASC> : AG_PAS_S TSAP <HEX> : 41 47 5F 50 41 53 5F 53 TSAP-Laenge: 8</div><div>ferne Parameter : MAC-Adresse <HEX> : 080006010001 TSAP <ASC> : CC_PAS_S TSAP <HEX> : 43 43 5F 50 41 53 5F 53 TSAP-Laenge: 8</div></div><div><div>+ 1</div><div>- 1</div><div>EINGABE</div><div>LOESCHEN</div><div>END-DRG</div><div>FE-PRG</div><div>FE-PRG</div><div>UEBERN.</div><div>AUSWAHL</div></div></div> <p>通过按下 <i>F3</i> 功能键，可以为下一个传输连接输入参数。这个传输连接将处理 WinCC 的写作业。</p> <p>在本实例中，将数值 2 作为作业号 <i>ANR</i>。在 <i>Job Type</i> 域内，指定 <i>RECEIVE</i>。在 <i>Transport Addresses</i> 区域内，以 ASCII 码的形式将 <i>Local Parameter</i> 的 <i>TSAP</i> 设置为 <i>AG_PAS_R</i>，将 <i>Remote Parameter</i> 的 <i>TSAP</i> 设置为 <i>CC_PAS_R</i>。对于远程参数，还要从计算机输入通讯处理器 <i>CP 1413</i> 的以太网地址。</p> <div><div>Transportverbindung</div><div>Quelle: H:A_UA_S5.DAT <OFFLINE> COM 1430 TF <ENDE></div><div><div>SSNR-Offset : 0</div><div>ANR : 2</div><div>Auftragsart : RECEIVE</div><div>Aktiv/Passiv <A/P> : P</div><div>Prioritaet : 2</div><div>Read/Write <J/N> : J</div><div>Anzahl der Auftraege pro TSAP : 1 von 1</div></div><div><div>Transportadressen : lokale Parameter :</div><div>TSAP <ASC> : AG_PAS_R TSAP <HEX> : 41 47 5F 50 41 53 5F 52 TSAP-Laenge: 8</div><div>ferne Parameter : MAC-Adresse <HEX> : 080006010001 TSAP <ASC> : CC_PAS_R TSAP <HEX> : 43 43 5F 50 41 53 5F 52 TSAP-Laenge: 8</div></div><div><div>+ 1</div><div>- 1</div><div>EINGABE</div><div>LOESCHEN</div><div>END-DRG</div><div>FE-PRG</div><div>FE-PRG</div><div>UEBERN.</div><div>AUSWAHL</div></div></div> <p>在创建连接 <i>S5-115U-WinCC-Active</i> 时，必须在 WinCC 项目中设置刚定义的连接参数。对于 <i>TSAP</i> 数值的设置，注意：输入空格与没有输入字符之间有区别。始终都要检查十六进制代码。</p>

步骤	过程：组态通讯处理器
6	<p>创建用于与 WinCC 进行主动通讯的传输连接。</p> <p>对于 PLC 的主动通讯，必须创建两个连接。一个处理 PLC 的写作业，而另一个则处理 WinCC 的写作业。</p> <p>通过按下 F3 功能键，可以为下一个传输连接输入参数。</p> <p>对于用于处理 PLC 的写作业的连接，本实例将 3 作为作业号 ANR 的值。在 <i>Job Type</i> 域内，指定 <i>SEND</i>。通过在 <i>Active/Passive</i> 域中输入 <i>A</i> 将 PLC 设置为主动。在 <i>Transport Addresses</i> 区域内，以 ASCII 码的形式将 <i>Local Parameter</i> 的 <i>TSAP</i> 设置为 <i>AG_AKT_S</i>，将 <i>Remote Parameter</i> 的 <i>TSAP</i> 设置为 <i>CC_AKT_S</i>。</p> <p>对于远程参数，还要从计算机输入通讯处理器 <i>CP 1413</i> 的以太网地址。</p> <div><div>Transportverbindung</div><div>Quelle: H:A_UA_S5.DAT <OFFLINE> COM 1430 TF <ENDE></div><div><div>SSNR-Offset : 0</div><div>ANR : 3</div></div><div><div>Auftragsart : SEND</div><div>Aktiv/Passiv <A/P> : A</div></div><div><div>Prioritaet : 2</div><div>Read/Write <J/N> : J</div></div><div><div>Anzahl der Auftraege pro TSAP : 1 von 1</div></div><div><div>Transportadressen : lokale Parameter :</div><div>TSAP <ASC> : AG_AKT_S TSAP <HEX> : 41 47 5F 41 4B 54 5F 53 TSAP-Laenge: 8</div></div><div><div>ferne Parameter :</div><div>MAC-Adresse <HEX> : 000006010001 TSAP <ASC> : CC_AKT_S TSAP <HEX> : 43 43 5F 41 4B 54 5F 53 TSAP-Laenge: 8</div></div><div><div>+ 1 - 1 EINGABE LOESCHEN</div><div>UEBERN. AUSWAHL</div></div></div> <p>通过按下 F3 功能键，可以为下一个传输连接输入参数。</p> <p>对于用于处理 WinCC 的写作业的连接，本实例将 4 作为作业号 <i>ANR</i> 的值。在 <i>Job Type</i> 域内，指定 <i>RECEIVE</i>。通过在 <i>Active/Passive</i> 域中输入 <i>P</i> 将 PLC 设置为被动。在 <i>Transport Addresses</i> 区域内，以 ASCII 码的形式将 <i>Local Parameter</i> 的 <i>TSAP</i> 设置为 <i>AG_AKT_R</i>，将 <i>Remote Parameter</i> 的 <i>TSAP</i> 设置为 <i>CC_AKT_R</i>。对于远程参数，还要从计算机输入通讯处理器 <i>CP 1413</i> 的以太网地址。</p> <div><div>Transportverbindung</div><div>Quelle: H:A_UA_S5.DAT <OFFLINE> COM 1430 TF <ENDE></div><div><div>SSNR-Offset : 0</div><div>ANR : 4</div></div><div><div>Auftragsart : RECEIVE</div><div>Aktiv/Passiv <A/P> : P</div></div><div><div>Prioritaet : 2</div><div>Read/Write <J/N> : J</div></div><div><div>Anzahl der Auftraege pro TSAP : 1 von 1</div></div><div><div>Transportadressen : lokale Parameter :</div><div>TSAP <ASC> : AG_AKT_R TSAP <HEX> : 41 47 5F 41 4B 54 5F 52 TSAP-Laenge: 8</div></div><div><div>ferne Parameter :</div><div>MAC-Adresse <HEX> : 000006010001 TSAP <ASC> : CC_AKT_R TSAP <HEX> : 43 43 5F 41 4B 54 5F 52 TSAP-Laenge: 8</div></div><div><div>+ 1 - 1 EINGABE LOESCHEN</div><div>UEBERN. AUSWAHL</div></div></div> <p>在创建连接 <i>S5-115U-WinCC-Passive</i> 时，必须在 WinCC 项目中设置刚定义的连接参数。</p> <p>传输连接输入屏中所作的设置通过 F7 功能键来实现。</p>

步骤	过程：组态通讯处理器
7	<p>将数据库文件的组态数据装入通讯处理器 <i>CP 1430 TF</i> 中。</p> <p>可以通过 <i>传送</i> → <i>FD->CP</i> 菜单来完成。只有当通讯处理器处于 <i>STOP</i> 运行模式时，组态数据才能上传。</p> <p>Transfer</p> 

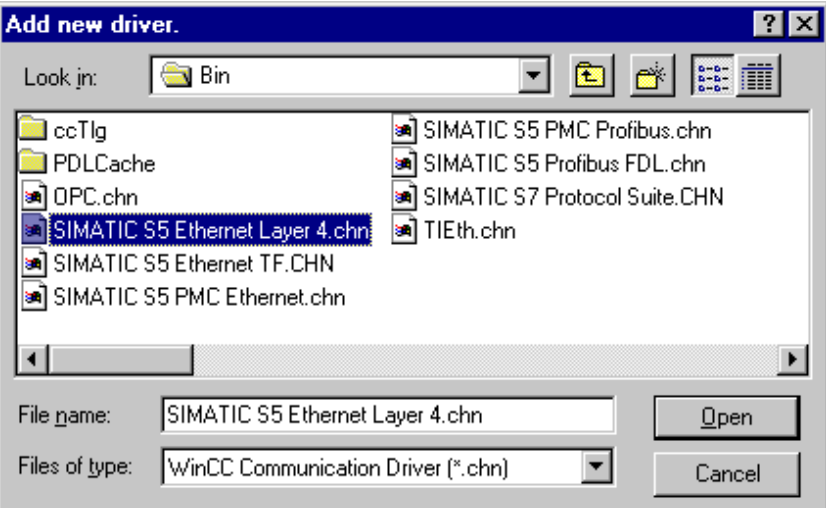

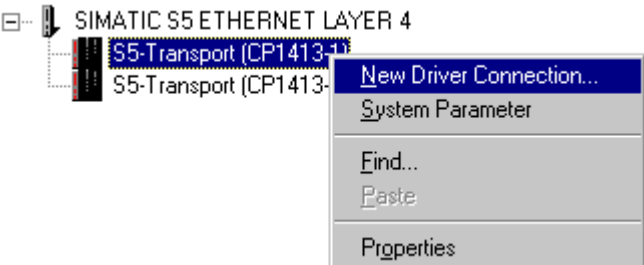
启动 PLC

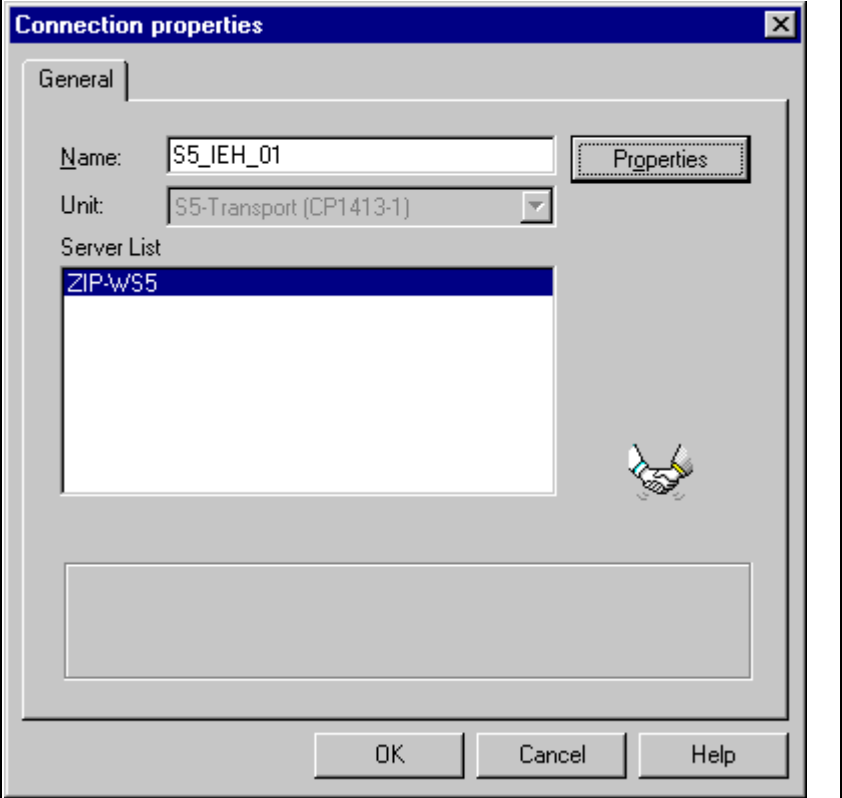
步骤	过程：启动 PLC
1	<p>启动 PLC 的各个模块。</p> <p>通讯处理器 <i>CP 1430 TF</i> 的 STEP5 程序和数据库文件必须预先装载到 PLC 中。</p> <p>首先，将通讯处理器的运行模式开关设置在 <i>RUN</i> 位置上。在通讯处理器上的 <i>RUN</i> 和 <i>STOP</i> 状态 LED 将亮起，表示模块尚未同步。</p> <p>接着，将 CPU 模块的运行模式开关设置在 <i>RN</i> 位置。在启动 CPU 模块时，由启动块来使通讯处理器同步。通讯处理器的 <i>STOP</i> 状态 LED 熄灭。在 CPU 模块处，将只有 <i>RN</i> 状态 LED 点亮。</p>

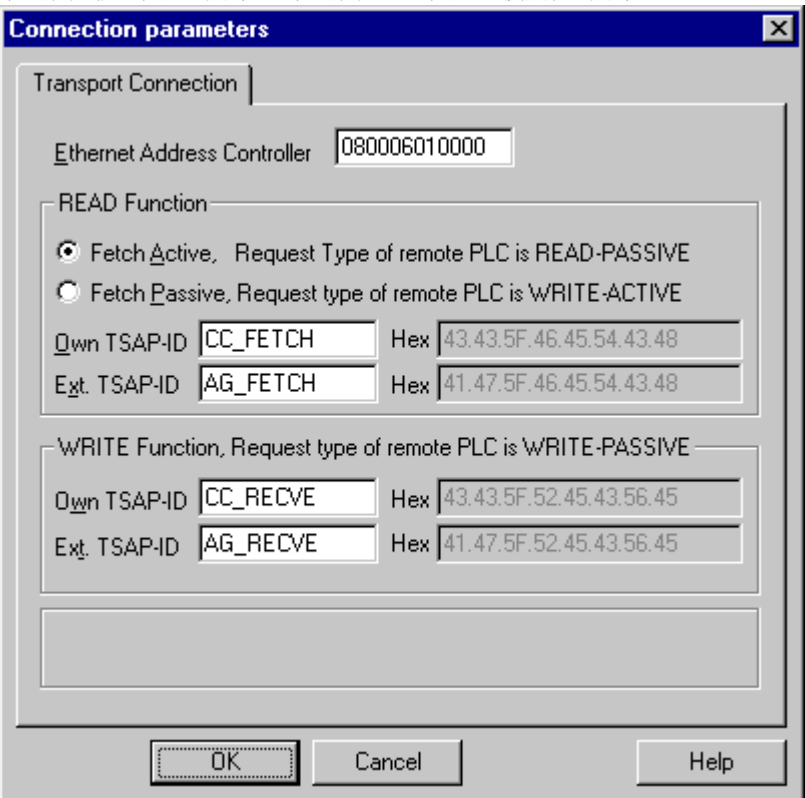
5.1.7.3 在 WinCC 中组态


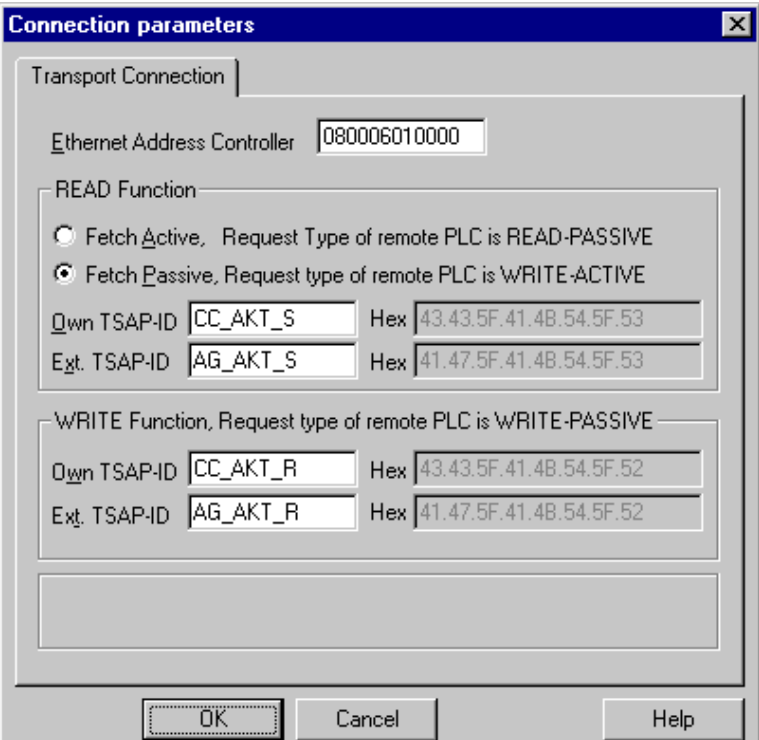
创建连接

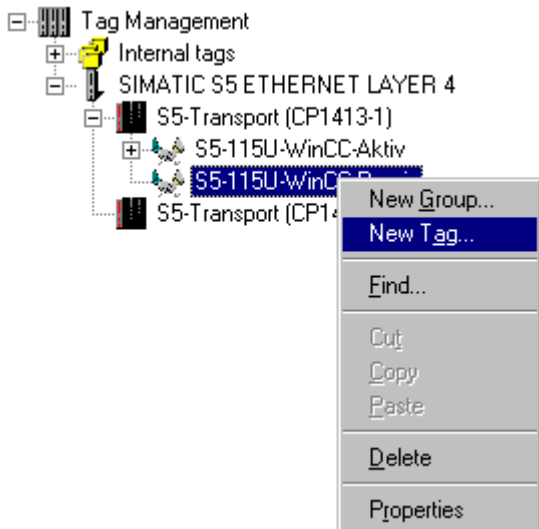
步骤	过程：创建连接
1	<p>所需通讯驱动程序的安装。这可以通过  <i>变量管理器</i>，然后从弹出式菜单中选择 <i>添加新驱动程序</i> 来完成。</p> 


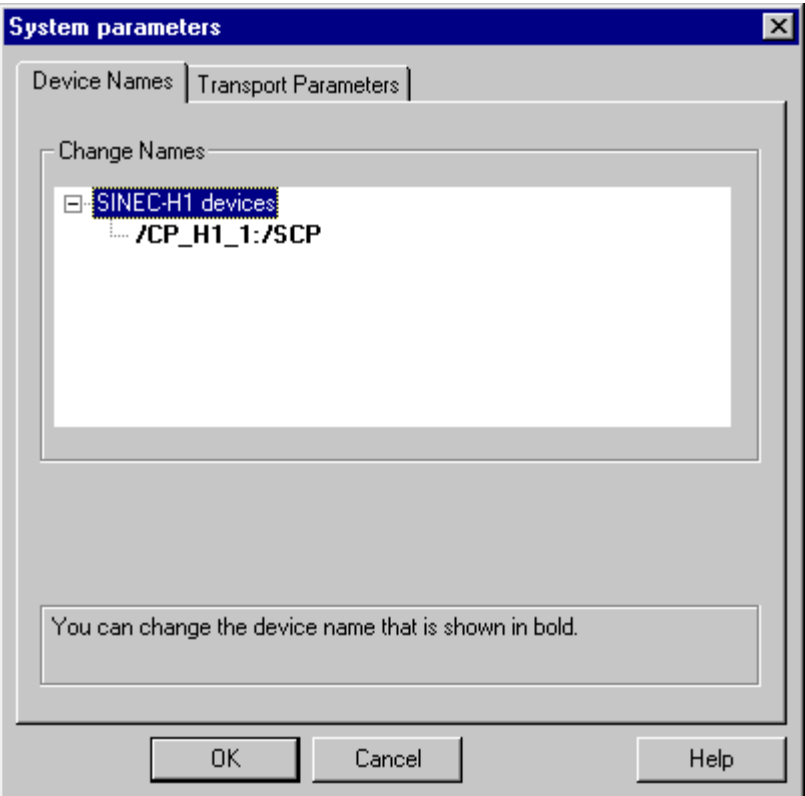
步骤	过程：创建连接
2	<p>将显示 添加新驱动程序 对话框。</p> <p>该对话框列出所有可以安装的通讯驱动程序。对于通过工业以太网与 <i>SIMATIC S5</i> 进行的通讯，安装驱动程序 <i>SIMATIC S5 Ethernet Layer 4</i>。从对话框中选择该驱动程序。通过单击 打开 退出对话框。</p> 
3	<p>新添加的驱动程序 <i>SIMATIC S5 Ethernet Layer 4</i> 将作为 变量管理器 的子条目来显示。</p> <p>该驱动程序包含两个通道单元。如果在计算机中有两个 <i>CP 1413</i> 通讯处理器在运行，则需要第二个通道单元。用于 <i>S5-Transport (CP1413-1)</i> 通道单元的新连接通过  <i>S5-Transport (CP1413-1)</i>，然后从弹出式菜单中选择 新建驱动程序连接 来创建。该连接用于从 PLC 中主动发送数据。在这种情况下，WinCC 为主动伙伴。</p> 

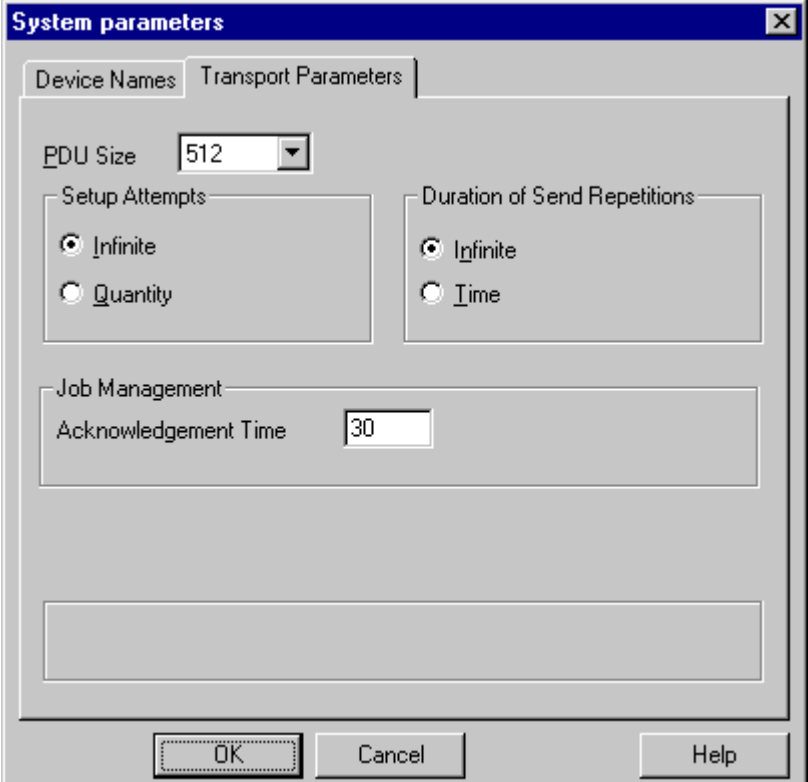
步骤	过程：创建连接
4	<p data-bbox="521 296 1359 422">将显示连接的属性对话框。 在常规标签中，输入新建连接的名称。在本实例中，它就是 <i>S5-115U-WinCC-Active</i>。 单击 <i>属性</i> 按钮来定义连接属性。</p> <div data-bbox="521 422 1359 1215"></div>

步骤	过程：创建连接
5	<p>将显示连接属性对话框。</p> <p>在传输连接标签内，定义期望的通讯连接参数。</p> <p>在 PLC 以太网地址域内，输入为 PLC 指定的以太网地址。本实例中，在组态通讯处理器 CP 1430 TF 时，已定义了地址 080006010000。</p> <p>在读功能区域中，为从 PLC 中读取数据进行连接设置。为了让 WinCC 主动请求数据，必须选择主动获取选项钮。</p> <p>对于本地和远程 TSAP，在为通讯处理器 CP 1430 TF 创建传输连接时，输入定义的数值。在本实例中，将数值 CC_PAS_S 输入本地 TSAP 域，将数值 AG_PAS_S 输入远程 TSAP 域。</p> <p>在写功能区域，为将数据写入 PLC 进行连接设置。在本实例中，将数值 CC_PAS_R 输入本地 TSAP 域，将数值 AG_PAS_R 输入远程 TSAP 域。</p> <p>单击确定按钮关闭对话框。单击确定也可关闭连接属性对话框。</p> 


























步骤	过程：创建连接
6	<p>为通道单元 <i>S5-Transport (CP1413-1)</i> 创建第二个连接</p> <p>通过  <i>S5-Transport (CP1413-1)</i>，然后从弹出式菜单中选择 <i>新建驱动程序连接</i>。PLC 使用该连接主动写数据。在这种情况下，WinCC 为被动伙伴。</p> <p>将显示连接的属性对话框。</p> <p>在 <i>常规</i> 标签中，输入新连接的名称。在本实例中，它就是 <i>S5-115U-WinCC-Passive</i>。通过 <i>属性</i> 按钮来定义连接属性。</p> <p>将显示 <i>连接属性</i> 对话框。</p> <p>在 <i>传输连接</i> 标签内，定义期望的通讯连接参数。</p> <p>在 <i>PLC 以太网地址</i> 域内，输入为 PLC 指定的以太网地址。本实例中，在组态通讯处理器 <i>CP 1430 TF</i> 时，已定义了地址 <i>080006010000</i>。</p> <p>在 <i>读功能</i> 区域内，选择 <i>被动获取</i> 项选钮。这就使 WinCC 成为被动伙伴。由 PLC 启动数据的发送。</p> <p>对于本地和远程 TSAP，在为通讯处理器 <i>CP 1430 TF</i> 创建传输连接时，输入定义的数值。在本实例中，将数值 <i>CC_AKT_S</i> 输入 <i>本地 TSAP</i> 域，将数值 <i>AG_AKT_S</i> 输入 <i>远程 TSAP</i> 域。</p> <p>在 <i>写功能</i> 区域中，为将数据写入 PLC 进行连接设置。在本实例中，将数值 <i>CC_AKT_R</i> 输入 <i>本地 TSAP</i> 域，将数值 <i>AG_AKT_R</i> 输入 <i>远程 TSAP</i> 域。</p> <p>单击 <i>确定</i> 按钮关闭对话框。单击 <i>确定</i> 也可关闭 <i>连接属性</i> 对话框。</p> 


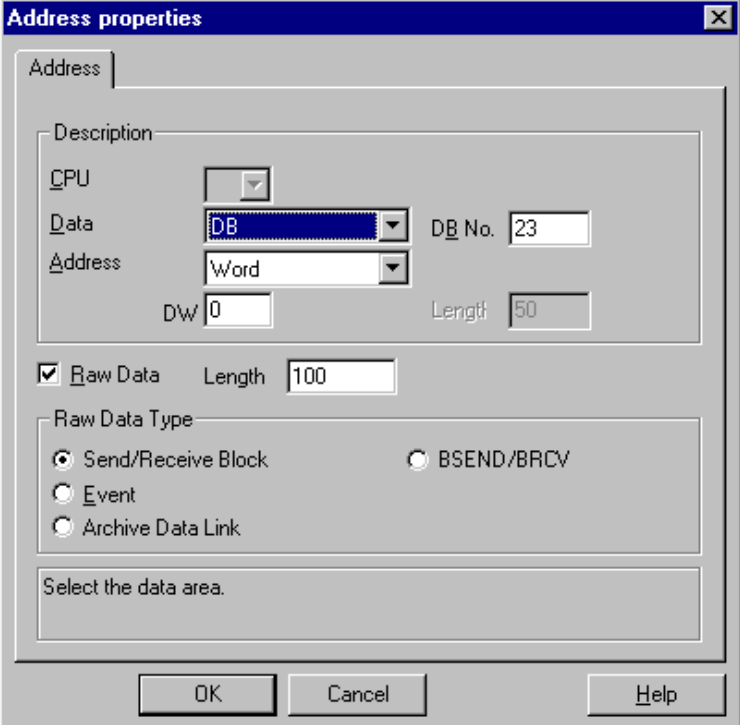
步骤	过程：创建连接
7	<div>刚创建的连接现在将显示在 <i>WinCC 资源管理器</i> 中。</div> <div></div>

步骤	过程：创建连接
8	<p>设置通道单元的系统参数。</p> <p>在系统参数对话框中进行这些设置，该对话框可通过  <i>S5-Transport</i> (<i>CP1413-1</i>) 条目，然后从弹出式菜单中选择 <i>系统参数</i> 来访问。</p> <p>在显示的对话框中，可以改变由 WinCC 用来访问 PLC 的访问点的名称。缺省情况下，设置的访问点是 <i>CP_H1_1</i>。先前在计算机中安装通讯处理器时，已将 <i>CP 1413</i> 分配给了访问点 <i>CP_H1_1</i>。</p> 



步骤	过程：创建连接
9	<p data-bbox="524 306 1354 390">在传输参数标签内，可以编辑影响通讯的各种设置。在本实例中，保留缺省设置。</p> <p data-bbox="524 365 797 390">单击确定按钮关闭对话框。</p> <div data-bbox="524 394 1328 1171"><p>The screenshot shows a 'System parameters' dialog box with a blue title bar and a close button (X). It has two tabs: 'Device Names' and 'Transport Parameters', with the latter selected. The 'Transport Parameters' section contains several settings: 'PDU Size' is a dropdown menu set to '512'; 'Setup Attempts' has two radio buttons, 'Infinite' (selected) and 'Quantity'; 'Duration of Send Repetitions' has two radio buttons, 'Infinite' (selected) and 'Time'; 'Job Management' is a section header; 'Acknowledgement Time' is a text box containing the value '30'. At the bottom, there are three buttons: 'OK' (highlighted with a dashed border), 'Cancel', and 'Help'.</p></div>

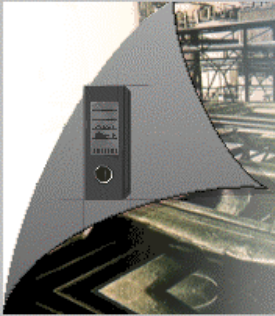
创建变量

步骤	过程：创建变量																											
1	<p>创建用于 <i>S5-115U-WinCC-Active</i> 连接的变量。这可从 <i>WinCC 资源管理器</i> 中通过  相应的连接条目，然后从弹出式菜单中选择 <i>新建变量</i> 来完成。</p> <p>各个变量的名称、数据类型和地址在下图中列出。</p> <p>变 量 <i>T08w_ex_UArd_RecordName</i> 、 <i>S32w_ex_UArd_Pressure</i> 和 <i>S32w_ex_UArd_Temperature</i> 代表 PLC 中当前的数据记录，它存储在 <i>DB21</i> 中。</p> <p>变量 <i>U32w_ex_UArd_Record</i>、 <i>U16w_ex_UArd_Field</i> 和 <i>U16w_ex_UArd_Job</i> 代表要处理的记录编号和域编号以及控制变量，它们存储在 <i>DB5</i> 中。</p> <p>变量 <i>U08w_ex_UArd_Busy</i> 和 <i>U08w_ex_UArd_Error</i> 用于显示当前作业状态。</p> <table><tr><th>Name</th><th>Type</th><th>Parameters</th></tr><tr><td> <i>S32w_ex_UArd_Temperature</i></td><td>Signed 32-bit value</td><td>DB21,DD5</td></tr><tr><td> <i>S32w_ex_UArd_Pressure</i></td><td>Signed 32-bit value</td><td>DB21,DD7</td></tr><tr><td> <i>T08w_ex_UArd_RecordName</i></td><td>Text tag 8-bit character set</td><td>DB21,DW0</td></tr><tr><td> <i>U16w_ex_UArd_Job</i></td><td>Unsigned 16-bit value</td><td>DB5,DW0</td></tr><tr><td> <i>U16w_ex_UArd_Field</i></td><td>Unsigned 16-bit value</td><td>DB5,DW3</td></tr><tr><td> <i>U32w_ex_UArd_Record</i></td><td>Unsigned 32-bit value</td><td>DB5,DD1</td></tr><tr><td> <i>U08w_ex_UArd_Busy</i></td><td>Unsigned 8-bit value</td><td>MB32</td></tr><tr><td> <i>U08w_ex_UArd_Error</i></td><td>Unsigned 8-bit value</td><td>MB35</td></tr></table>	Name	Type	Parameters	 <i>S32w_ex_UArd_Temperature</i>	Signed 32-bit value	DB21,DD5	 <i>S32w_ex_UArd_Pressure</i>	Signed 32-bit value	DB21,DD7	 <i>T08w_ex_UArd_RecordName</i>	Text tag 8-bit character set	DB21,DW0	 <i>U16w_ex_UArd_Job</i>	Unsigned 16-bit value	DB5,DW0	 <i>U16w_ex_UArd_Field</i>	Unsigned 16-bit value	DB5,DW3	 <i>U32w_ex_UArd_Record</i>	Unsigned 32-bit value	DB5,DD1	 <i>U08w_ex_UArd_Busy</i>	Unsigned 8-bit value	MB32	 <i>U08w_ex_UArd_Error</i>	Unsigned 8-bit value	MB35
Name	Type	Parameters																										
 <i>S32w_ex_UArd_Temperature</i>	Signed 32-bit value	DB21,DD5																										
 <i>S32w_ex_UArd_Pressure</i>	Signed 32-bit value	DB21,DD7																										
 <i>T08w_ex_UArd_RecordName</i>	Text tag 8-bit character set	DB21,DW0																										
 <i>U16w_ex_UArd_Job</i>	Unsigned 16-bit value	DB5,DW0																										
 <i>U16w_ex_UArd_Field</i>	Unsigned 16-bit value	DB5,DW3																										
 <i>U32w_ex_UArd_Record</i>	Unsigned 32-bit value	DB5,DD1																										
 <i>U08w_ex_UArd_Busy</i>	Unsigned 8-bit value	MB32																										
 <i>U08w_ex_UArd_Error</i>	Unsigned 8-bit value	MB35																										


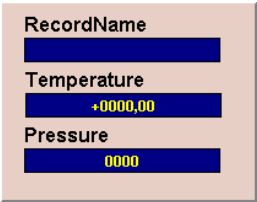
步骤	过程：创建变量
2	<p>创建用于 <i>S5-115U-WinCC-Passive</i> 连接的变量。这可从 <i>WinCC 资源管理器</i> 中通过  相应的连接条目，然后从弹出式菜单中选择 <i>新建变量</i> 来完成。</p> <p>将显示该变量的属性对话框。在本实例中，该变量的名称为 <i>RAW_ex_UArd_01</i>。将该变量的数据类型设置为 <i>原始数据类型</i>。</p> <p>通过 <i>选择</i> 按钮，打开对变量寻址的对话框。</p> <p>在 <i>地址描述</i> 域内，为本实例设置 <i>数据范围 DB</i> 和 <i>DB 编号 23</i>。对于 <i>地址</i>，输入 <i>字</i>，而且为 <i>DW, 0</i>。选择 <i>原始数据</i> 复选框，并将变量的长度设置为 <i>46</i>。在 <i>原始数据类型</i> 域内，选择选项按钮 <i>发送/接收块</i>。</p> <p>通过 <i>确定</i> 按钮关闭 <i>地址属性</i> 对话框。也可通过 <i>确定</i> 按钮关闭 <i>变量属性</i> 对话框。</p> 

创建用户归档

步骤	过程：创建用户归档
1	<p>打开 <i>用户归档编辑器</i>。在该编辑器中，通过  <i>归档</i> 条目，然后从弹出式菜单中选择 <i>新建归档</i> 来创建新的归档。这样可启动用于创建归档的向导。</p> 

步骤	过程：创建用户归档																									
2	<p>在该向导的第一页上，输入归档名称。在本实例中，将名称 <i>UserArchive_rd</i> 输入归档名称域内。别名域保留空白。</p> <p>对于归档类型，则选择无限制。</p> <p>通过单击下一步继续到下一页。</p>																									
3	<p>在向导的第二页，选择选项钮通过 <i>WinCC 原始数据变量进行通讯</i>。</p> <p>对于 <i>PLCID</i>，本实例使用名称 <i>USRARCRD</i>。该 ID 不能超过 8 个字符。通过该 ID，由 PLC 发送的作业报文可以分配给某个归档。</p> <p>通过选择按钮，设置为归档通讯所组态的 <i>WinCC 原始数据变量</i>。在本实例中，这就是变量 <i>RAW_ex_UArd_01</i>。</p> <p>由于在下一页中不必为本实例做任何设置，所以可通过单击完成退出向导。</p> <div><div>Communication</div><div><div>Type: <input type="radio"/> No communication <input checked="" type="radio"/> Communication via a WinCC Raw Data Tag <input type="radio"/> Communication via a WinCC Tag</div><div>PLCID: USRARCR</div><div>WinCC Tag: RAW_ex_UArd_01 Select... Create... Edit...</div></div><div>< Back Next > Finish Cancel Help</div></div>																									
4	<p>为刚创建的归档，创建在下表中列出的域。</p> <p>保存所作的设置。</p> <table><tr><th>Name</th><th>Alias</th><th>Type</th><th>Length</th></tr><tr><td> Pressure</td><td></td><td>Number (integer)</td><td></td></tr><tr><td> RecordName</td><td></td><td>String</td><td>10</td></tr><tr><td> Temperature</td><td></td><td>Number (double)</td><td></td></tr></table>	Name	Alias	Type	Length	Pressure		Number (integer)		RecordName		String	10	Temperature		Number (double)										
Name	Alias	Type	Length																							
Pressure		Number (integer)																								
RecordName		String	10																							
Temperature		Number (double)																								
5	<p>在用户归档编辑器的底部表格窗口中，现在可以为归档创建多个数据记录。</p> <table><tr><th></th><th>ID</th><th>RecordName</th><th>Temperature</th><th>Pressure</th></tr><tr><td>1</td><td>1</td><td>Record_01</td><td>236,23</td><td>563</td></tr><tr><td>2</td><td>2</td><td>Record_02</td><td>302,78</td><td>399</td></tr><tr><td>3</td><td>3</td><td>Record_03</td><td>278,92</td><td>456</td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr></table>		ID	RecordName	Temperature	Pressure	1	1	Record_01	236,23	563	2	2	Record_02	302,78	399	3	3	Record_03	278,92	456	...				
	ID	RecordName	Temperature	Pressure																						
1	1	Record_01	236,23	563																						
2	2	Record_02	302,78	399																						
3	3	Record_03	278,92	456																						
...																										

在图形编辑器中的实现

步骤	过程：在图形编辑器中的实现
1	打开图形编辑器并创建一个新画面。在本实例中，它是画面 <i>ex_3_chapter_01a.pdl</i> 。 组态用于显示数据的控件。它是 WinCC 用户归档 - 表格元素。从对象选项板的控件选择菜单中选择它，然后将其置于画面中。
2	 通过 D 刚创建的对象控件 1，打开其属性对话框。 在常规标签中，将先前组态的归档 <i>UserArchive_rd</i> 设置在源域中。在编辑域内，选择插入、改变和删除复选框，以使所有的编辑选项能为用户所使用。此外，选择边框复选框。 在工具栏标签中，撤消选择按钮写变量和读变量这两个复选框。在本实例中，由 PLC 执行该控件。还将使用工具栏的所有其它按钮。 为了在运行系统中同时显示所有的列，在字体标签中，将字体的大小减小到 10。 在 WinCC 用户归档 - 表格元素的属性对话框中所作的设置通过确定按钮来结束。 在本实例中，通过对象控件 1 的属性对话框使表格的颜色方案与项目的颜色方案相匹配。
3	为了显示当前在 PLC 中装载的数据记录，组态三个智能对象 → I/O 域。在本实例中，它们是对象 I/O 域 1、I/O 域 2 和 I/O 域 3。 对于 I/O 域，在属性 → 输出/输入 → 输出值处为三个变量 <i>T08w_ex_UArd_RecordName</i> 、 <i>S32w_ex_UArd_Temperature</i> 和 <i>S32w_ex_UArd_Pressure</i> 中的每个变量分别创建一个变量连接。 

步骤	过程：在图形编辑器中的实现
4	<p>为了指定要处理的记录编号和域编号，另外组态两个智能对象 → I/O 域。在本实例中，它们是对象 I/O 域 4 和 I/O 域 5。</p> <p>对于对象 I/O 域 4，在属性 → 输出/输入 → 输出值处创建至变量 <i>U32w_ex_UArd_Record</i> 的变量连接。该 I/O 域显示记录编号。为对象 I/O 域 5，在属性 → 输出/输入 → 输出值处创建一个至变量 <i>U16w_ex_UArd_Field</i> 的变量连接。该 I/O 域显示域编号。</p> <p>为了触发 PLC 中的动作，组态了五个 <i>Windows</i> 对象 → 按钮。在本实例中，它们是对象按钮 9 至按钮 13。</p> <p>如果按下一个按钮，则将不同的数值写入了变量 <i>U16w_ex_UArd_Job</i> 中。在 STEP5 程序中，周期性查询该变量。根据该变量的内容，触发相应的作业。</p> <p>对于 PLC，每个数值对应于不同的作业类型。下面列出了各个数值所对应的作业。对于 STEP5 程序中执行的删除所有记录作业，还未组态任何按钮。</p> <ul style="list-style-type: none">• 1 = 删除所有数据记录• 2 = 从归档中读取记录• 4 = 将记录写入归档• 8 = 从归档中删除记录• 16 = 从归档中读取域• 32 = 将域写入归档 <p>通过重新设定按钮，可以重新设定变量 <i>U08w_ex_UArd_Busy</i>。这样可进行加锁，以防在没有自动撤消将要删除的新作业报文时将其发送(仅与出错有关)。</p> <p>在状态域中，使用 <i>C</i> 动作显示通讯连接的当前状态以及归档通讯。该 <i>C</i> 动作计算变量 <i>U08w_ex_UArd_Error</i>。在本步骤之后描述 <i>C</i> 动作本身。</p> <div><div><div>Record Number</div><div>001</div><div>Field Number</div><div>000</div></div><div><div>Write record to PLC</div><div>Read record from PLC</div><div>Delete Record</div><div>Delete All Records</div></div><div><div>Write field to PLC</div><div>Read field from PLC</div></div></div>

用于显示状态的 C 动作

```

#include "apdefap.h"
char* _main(char* lpszPictureName, char* lpszObjectName, char* lpszProperty)
{
    BYTE byError = 0; //communication state archive
    DWORD dwState = 0; //communication state connection

    //activate communication check led
    SetBackColor(lpszPictureName, "LED", CO_RED);

    //communication check
    byError = GetTagByteStateWait("U08w_ex_UArd_Error",&dwState);

    //deactivate communication check led
    SetBackColor(lpszPictureName, "LED", CO_DKGRAY);

    //if connection error
    if (dwState > 0)
    {
        return "No Connection";
    }

    //switch archive state
    switch (byError)
    {
        case 0:      return "Ready";
        case 1:
        case 2:      return "Error Archive";
        case 101:
        case 102:    return "Error Record";
        case 201:
        case 202:    return "Error Field";
        case 254:    return "Not Supported";
        default:     return "Unknown Error";
    }
}

```

- 在对象 *StaticText3* 的属性 → 字体 → 文本处已经创建了 C 动作。将 C 动作的触发设置为 2 秒。这样将每 2 秒对连接和归档通讯进行状态检查。
- 由函数 *GetTagByteStateWait* 进行状态检查。将当前的连接状态写入变量 *dwState*，将归档通讯状态写入变量 *byError* (来自 WinCC 的确认报文的错误代码)。
- 在执行该动作前，将对象 *LED* 的背景色设置为红色。一旦函数结束，则对象 *LED* 的背景色重新设定为灰色。
- 计算变量 *dwState* 和 *byError* 并将相应的状态文本返回。

常规应用的注意事项

在进行常规应用之前，必须完成下述修改：

- 如果使用的硬件相同，可以将所进行的通讯组态直接应用于用户自己的应用程序。否则，必须做修改。
- 必须根据需要修改归档结构。
- 可以重新使用 STEP5 程序的组织类型。然而，可以使用记录作业报文的单个数据块。在这种情况下，通过 STEP5 程序，除了输入记录编号、域编号和作业数据之外，还要将其余的作业参数输入该数据块。

5.1.8 通过 WinCC 原始数据变量与 SIMATIC S7 进行通讯 (ex_3_chapter_01a.pdl)

本章中所创建的 STEP7 项目也可直接从在线文档复制到用户的硬盘上。缺省情况下，它将存储在 *C:\Configuration_Manual* 文件夹中。



S7_UserArchive

任务定义

在 SIMATIC S7 PLC 与 WinCC 站之间建立通讯连接。PLC 从在 WinCC 站上创建的用户归档中读取数据、将数据写入其中以及从中删除数据。

归档的每个数据记录包括两个整型域以及一个用于记录数据记录名的文本域。

概念的实现

在本实例中，使用带有 *CPU 416-1* 中心模块的 *SIMATIC S7-400* PLC。使用 MPI 网络建立与该 PLC 的通讯。为此在 WinCC 站中使用通讯处理器 *CP 5412 A2*，在 PLC 中用中心模块的编程接口来进行通讯。

在 WinCC 中，使用 *SIMATIC S7 PROTOCOL SUITE* 通讯驱动程序的 *MPI* 通道单元。其中，该通讯处理器支持从 PLC 中主动发送数据。

为 *MPI 通道* 单元创建 2 个与 PLC 的连接。一个连接用于从 PLC 中主动发送数据。为该连接创建一个用于与用户归档进行通讯的 *WinCC 原始数据变量*。此外，创建一个用户归档，其通讯通过该 *WinCC 原始数据变量* 来组态。

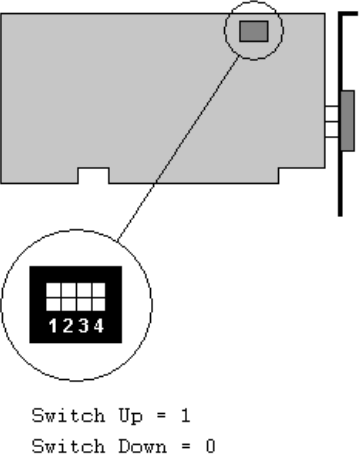
对于第二个连接，WinCC 作为主动伙伴。其中，该连接用于模拟 WinCC 中的通讯操作。通过各种按钮，在 PLC 中设置控制位，用于触发作业报文的发送。另外，PLC 中现有的数据(当前数据记录、当前设置的记录编号和域编号以及作业状态)通过 *I/O 域* 来显示。

在运行系统中，用户归档的数据由控件以列表形式显示。

5.1.8.1 通讯处理器 CP 5412 A2 的启动

以下详细描述成功启动通讯处理器 CP 5412 A2 必要的组态步骤。


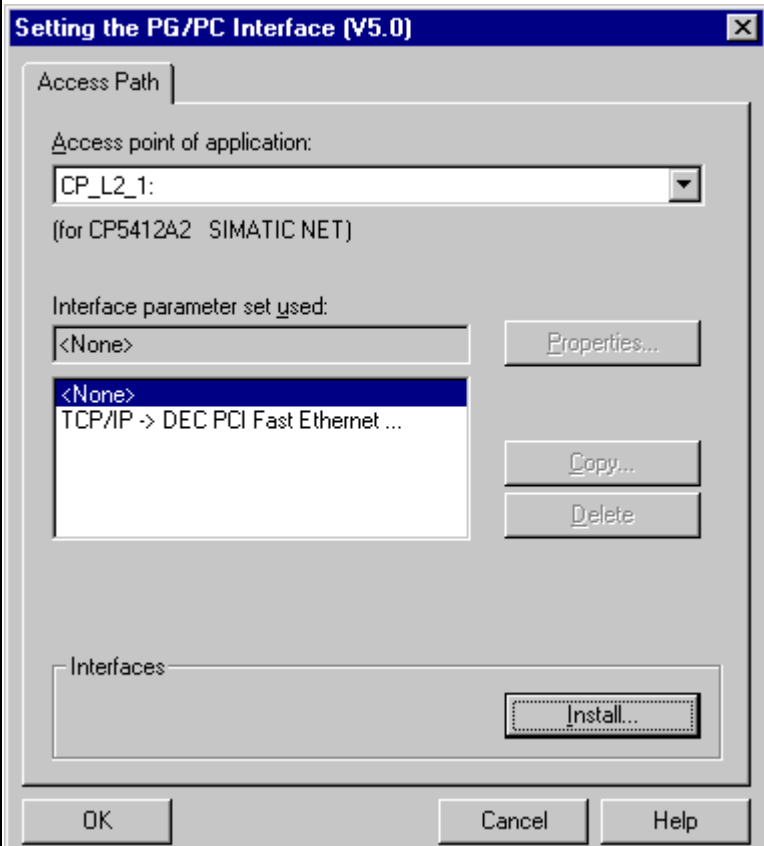
在计算机中安装通讯处理器

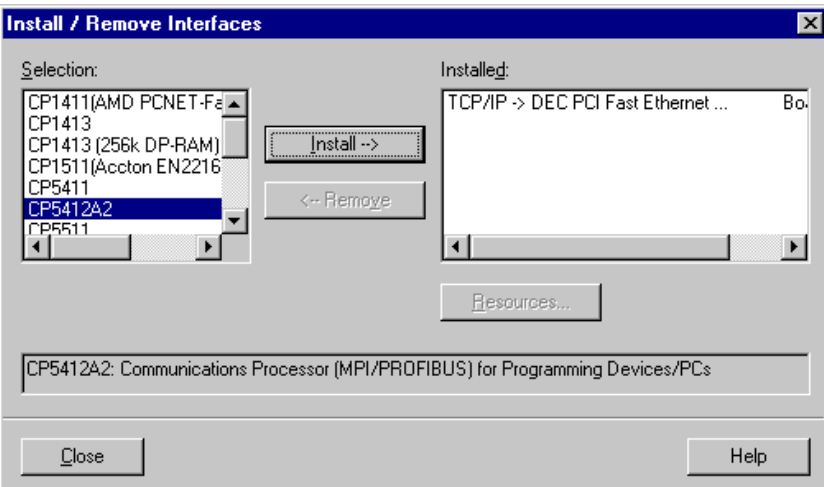
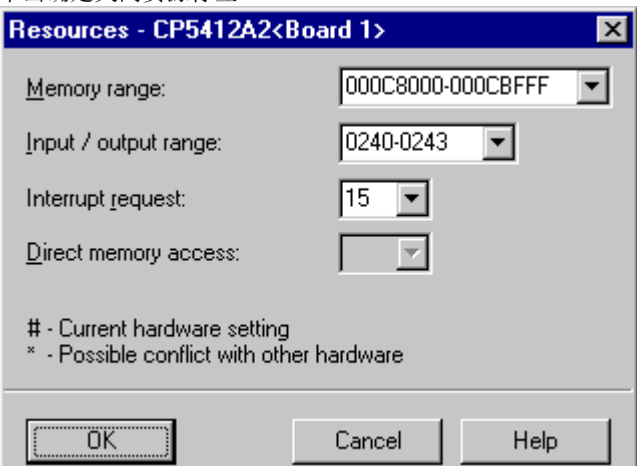
步骤	过程：在计算机中安装通讯处理器																																		
1	<div><p>检查 CP 5412 A2 处所选择的跳线设置。</p><p>在安装 CP 5412 A2 时，必须指定 I/O 范围。通过跳线设置来设置 I/O 范围。缺省情况下，I/O 范围设置为 0240-0243。但是也可以用其它设置。下面的图形说明了各种 I/O 范围所需的跳线设置。</p><div><table><thead><tr><th>I/O Area</th><th>1-2-3-4</th></tr></thead><tbody><tr><td>0240-0243</td><td>0 0 0 0</td></tr><tr><td>0244-0247</td><td>0 0 0 1</td></tr><tr><td>0248-024B</td><td>0 0 1 0</td></tr><tr><td>024C-024F</td><td>0 0 1 1</td></tr><tr><td>0280-0283</td><td>0 1 0 0</td></tr><tr><td>0284-0287</td><td>0 1 0 1</td></tr><tr><td>0288-028B</td><td>0 1 1 0</td></tr><tr><td>028C-028F</td><td>0 1 1 1</td></tr><tr><td>0300-0303</td><td>1 0 0 0</td></tr><tr><td>0304-0307</td><td>1 0 0 1</td></tr><tr><td>0308-030B</td><td>1 0 1 0</td></tr><tr><td>030C-030F</td><td>1 0 1 1</td></tr><tr><td>0390-0393</td><td>1 1 0 0</td></tr><tr><td>0394-0397</td><td>1 1 0 1</td></tr><tr><td>0398-039B</td><td>1 1 1 0</td></tr><tr><td>039C-039F</td><td>1 1 1 1</td></tr></tbody></table></div><p>Switch Up = 1 Switch Down = 0</p></div>	I/O Area	1-2-3-4	0240-0243	0 0 0 0	0244-0247	0 0 0 1	0248-024B	0 0 1 0	024C-024F	0 0 1 1	0280-0283	0 1 0 0	0284-0287	0 1 0 1	0288-028B	0 1 1 0	028C-028F	0 1 1 1	0300-0303	1 0 0 0	0304-0307	1 0 0 1	0308-030B	1 0 1 0	030C-030F	1 0 1 1	0390-0393	1 1 0 0	0394-0397	1 1 0 1	0398-039B	1 1 1 0	039C-039F	1 1 1 1
I/O Area	1-2-3-4																																		
0240-0243	0 0 0 0																																		
0244-0247	0 0 0 1																																		
0248-024B	0 0 1 0																																		
024C-024F	0 0 1 1																																		
0280-0283	0 1 0 0																																		
0284-0287	0 1 0 1																																		
0288-028B	0 1 1 0																																		
028C-028F	0 1 1 1																																		
0300-0303	1 0 0 0																																		
0304-0307	1 0 0 1																																		
0308-030B	1 0 1 0																																		
030C-030F	1 0 1 1																																		
0390-0393	1 1 0 0																																		
0394-0397	1 1 0 1																																		
0398-039B	1 1 1 0																																		
039C-039F	1 1 1 1																																		
2	<p>根据安装说明安装模块。其中还要遵守处理静电敏感设备(ESD)的步骤。必须在计算机关闭时才能安装模块。</p> <p>对于通讯卡 CP 5412 A2，要求计算机内有一个空 ISA 插槽。安装 CP 5412 A2 之后，盖上计算机外壳并启动计算机。</p>																																		

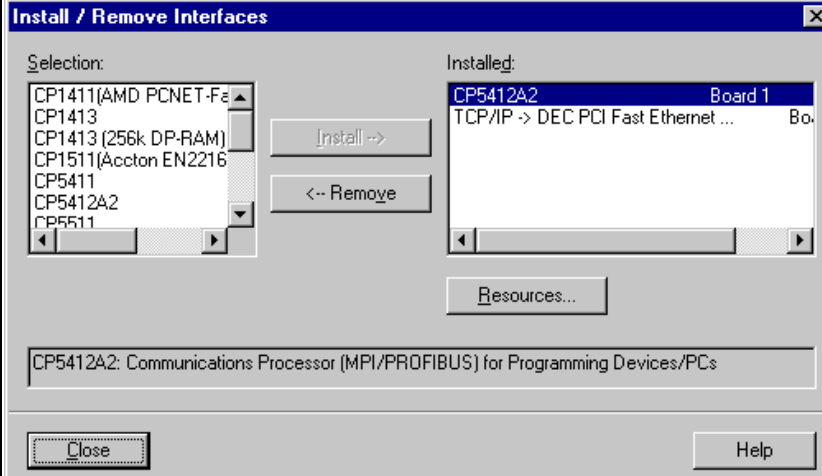
安装通讯驱动程序

步骤	过程：安装通讯驱动程序
1	<p>从 <i>SIMATIC NET</i> 光盘上安装通讯驱动程序 <i>PB S7-5412</i>。</p> <p>在插入 <i>SIMATIC NET</i> 光盘后，将自动启动安装程序。如果没有自动启动，则打开 <i>Windows NT</i> 资源管理器，然后启动光盘上的 <i>setup.exe</i> 程序。</p> <p>通过如下显示的按钮启动软件的安装。</p> <div></div> <p>按照安装程序的说明。在 <i>组件</i> 页面上，必须选择要安装的驱动程序 <i>PB S7-5412</i> 的复选框。完成安装。</p> <div></div>

安装通讯处理器

步骤	过程：安装通讯处理器
1	<p>通过 设置 PG/PC 接口程序安装通讯处理器 CP 5412 A2。</p> <p>通过 开始 → 设置 → 控制面板 → 设置 PG/PC 接口访问该程序。</p>  <p>Setting the PG/PC Interface</p>
2	<p>将显示程序设置 PG/PC 接口。</p> <p>通过 安装按钮打开安装新接口的对话框。</p> 

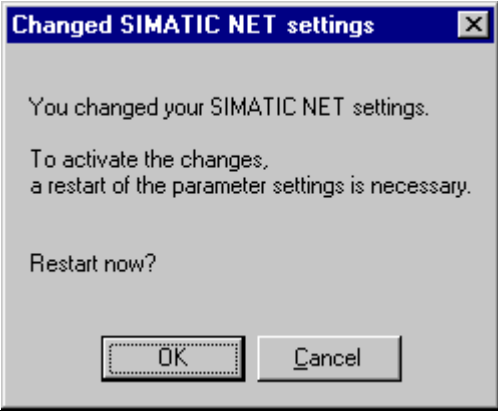
步骤	过程：安装通讯处理器
3	<p>将显示对话框 安装/删除模块。</p> <p>选择域列出了所有可以安装的接口。如果先前已按步骤 B 中所述的方法安装了通讯驱动程序，则在其中将会有 CP 5412 A2 条目。</p> <p>从选择域中，选择 CP 5412 A2 条目。单击 安装> 按钮来启动通讯处理器的安装。</p> 
4	<p>将显示 资源 - CP 5412 A2 对话框。</p> <p>必须指定 存储器范围、I/O 范围和 中断的设置。</p> <p>I/O 范围通过 CP 5412 A2处的跳线设置已经确定。</p> <p>确保所分配的资源还未被计算机中的其它模块占用。有关已占用的系统资源的信息可以通过 开始 → 程序 → 管理工具(公用) → Windows NT 诊断器来访问资源标签而获得。</p> <p>单击确定关闭资源标签。</p> 

步骤	过程：安装通讯处理器
5	<p data-bbox="521 302 1339 359">在安装/删除模块对话框中，已安装域现在将包含 CP 5412 A2 条目。 通过 关闭按钮，退出 安装/删除模块对话框。</p> 

分配通讯处理器

步骤	过程：分配通讯处理器
1	<p>在设置 PG/PC 接口程序中，将访问点 MPI (WinCC)分配给刚安装的接口。访问点 <i>MPI (WinCC)</i>是由 WinCC 所使用的通过 <i>MPI</i>进行通讯的缺省访问点。在应用程序的访问点域内，设置 <i>MPI (WinCC)</i>条目。在下面的域内，选择 <i>CP 5412 A2 (MPI)</i>条目。这样就完成了访问点和通讯处理器之间的分配。</p> <div><div>Setting the PG/PC Interface [V5.0]</div><div><div>Access Path</div><div>Access point of application: MPI (WinCC) --> CP5412A2(MPI) (for CP5412A2 SIMATIC NET)</div><div>Interface parameter set used: CP5412A2(PROFIBUS) <keine> CP5412A2(MPI) <Aktiv> CP5412A2(PROFIBUS) TCP/IP -> DEC PCI Fast Ethernet ...</div><div>Properties... Diagnostics... Copy... Delete</div><div>(Configuration of your Communications Processor CP 5412 [A2] for a PROFIBUS-Network)</div><div>Interfaces Install...</div><div>OKCancelHelp</div></div></div>

步骤	过程：分配通讯处理器
2	<p>设置通讯处理器 CP 5412 A2 的属性。</p> <p>通过 设置 PG/PC 接口程序的属性按钮可打开设置属性的对话框。</p> <p>将显示属性 - CP 5412 (MPI)对话框。</p> <p>在 MPI 标签内，设置与站和网络相关的参数。</p> <p>在本实例中，将通讯处理器的地址设置为 0。</p> <p>对于 MPI 网络，本实例所使用的传输率为 187.5 Kbit/s。最高站地址设置为最大值 31。</p> <p>通过单击确定来关闭 CP 5412 A2 的属性对话框。</p> 

步骤	过程：分配通讯处理器
3	<p>通过 确定 按钮，退出 <i>设置 PG/PC 接口程序</i>。</p> <p>将显示一个对话框，请求重新启动 <i>CP 5412 A2</i>。通过单击 确定 确认该对话框，将导致重新启动通讯处理器 <i>CP 5412 A2</i>。</p> <p>这样就完成了通讯处理器的安装。</p> 
4	<p>如果重新启动通讯处理器没有成功，则必须确定出错原因并予以纠正。有关该主题的帮助信息可参见 <i>通讯手册</i>。</p>

5.1.8.2 PLC 的启动

以下详细说明创建与启动 STEP7 项目 S7_UserArchive 所需的步骤。










安装硬件


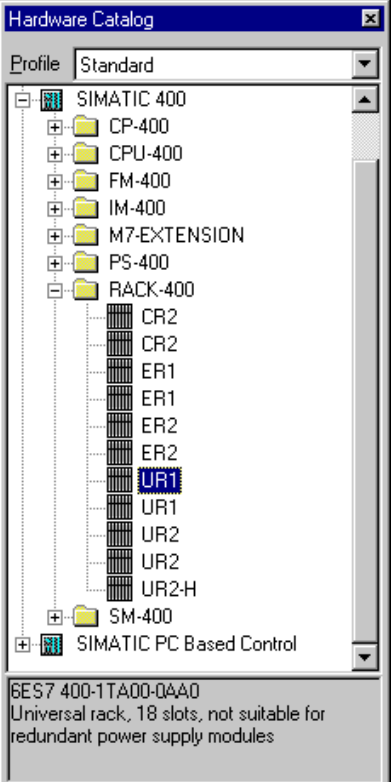
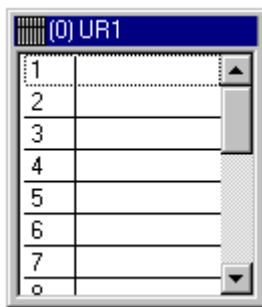
步骤	过程：安装硬件
1	<p>在机架上安装使用的模块。</p> <p>在本实例中，要安装的模块是电源 <i>PS 407 10A</i> 和 CPU 模块 <i>CPU 416-1</i>。</p> <p>建立从编程设备至 CPU 模块的编程接口的连接。</p>


创建 STEP7 项目

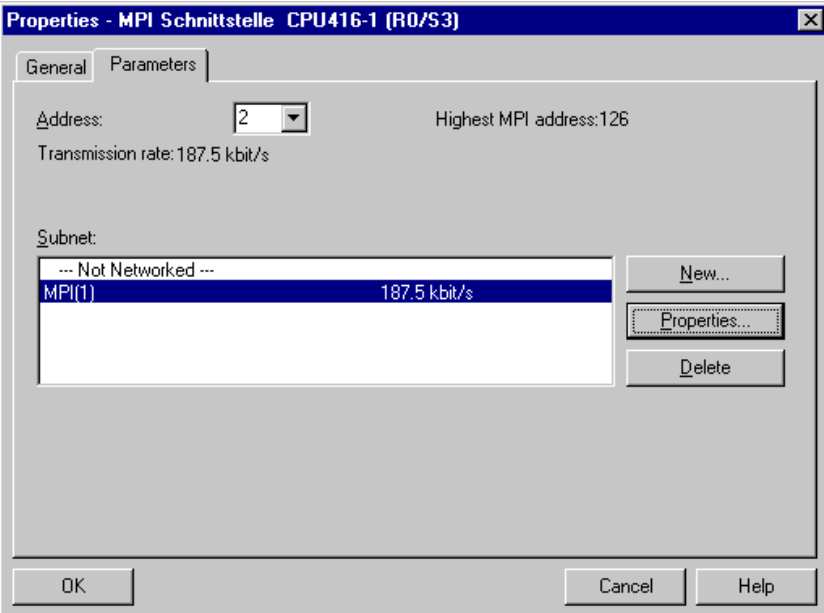

步骤	过程：创建 STEP7 项目
1	<p>在 <i>SIMATIC 管理器</i> 中创建新的 STEP7 项目。</p> <p>它通过 <i>开始</i> → <i>Simatic</i> → <i>SIMATIC 管理器</i> 来启动。</p> <div data-bbox="587 438 652 504"></div> <p>SIMATIC Manager</p>
2	<p>这样就会显示 <i>SIMATIC 管理器</i>。</p> <p>通过菜单 <i>文件</i> → <i>新建</i>，将打开用于指定新 STEP7 项目参数的对话框。</p> <p>将显示 <i>新建</i> 对话框。</p> <p>必须选择 <i>新建项目</i> 选项钮。在 <i>名称</i> 域中，输入要创建的新项目名。本实例的项目名为 <i>S7_UserArchive</i>。</p> <p>在缺省情况下，项目存储在 <i>C:\SIEMENS\STEP7\S7proj</i> 文件夹中。这可以通过 <i>浏览</i> 按钮随时进行改变。</p> <p>通过 <i>确定</i> 按钮关闭 <i>新建</i> 对话框。</p> <div data-bbox="526 798 1102 1562"></div>

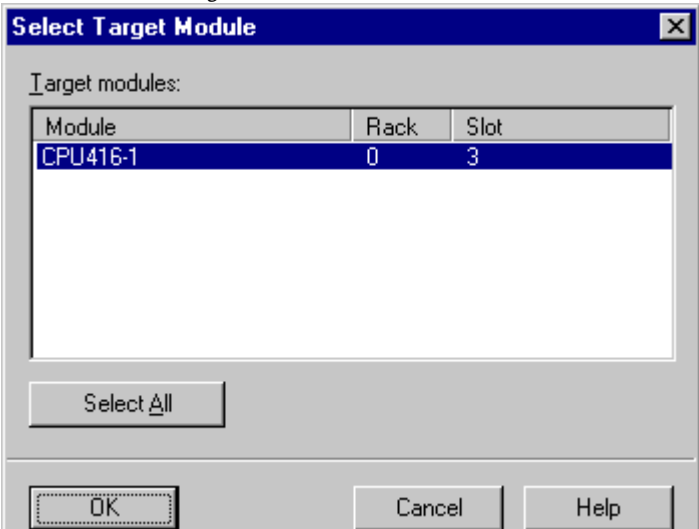
组态硬件

步骤	过程：组态硬件
1	<p>新的项目将在 <i>SIMATIC 管理器</i> 中显示。 必须为该项目组态硬件。需要一个 <i>SIMATIC 400 站</i> 组件。该组件用以下方法添加至 <i>SIMATIC 管理器</i> 中，即  项目名称 <i>S7_UserArchive</i>，然后从弹出式菜单中选择插入 <i>新对象</i> → <i>SIMATIC 400 站</i>。 刚添加的组件将显示在 <i>SIMATIC 管理器</i> 的右边窗口中。</p> <p> <i>SIMATIC 400(1)</i>  <i>MPI(1)</i></p>
2	<p>通过  右边窗口中的组件 <i>SIMATIC 400(1)</i>，将会显示 <i>硬件点</i>。通过  <i>硬件点</i> 或  它，然后从弹出式菜单中选择 <i>打开对象</i>，将启动程序 <i>HW Config</i>。该程序用于明确定义 PLC 中使用的硬件并组态其属性。</p> <p> HW Konfig</p>
3	<p>通过单击如下显示的 <i>HW Config</i> 程序的工具栏按钮，打开 <i>硬件目录</i>。此目录用于选择所需的硬件组件。</p> <p> </p>




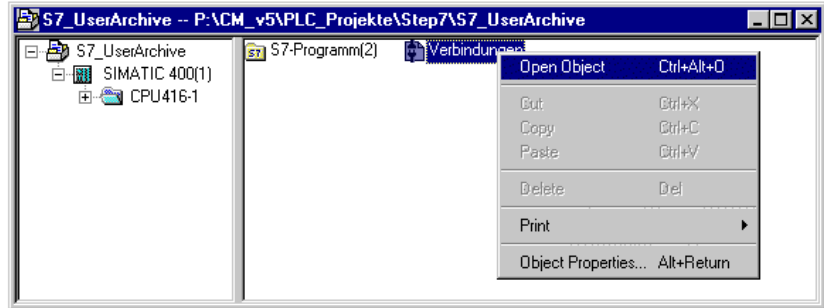

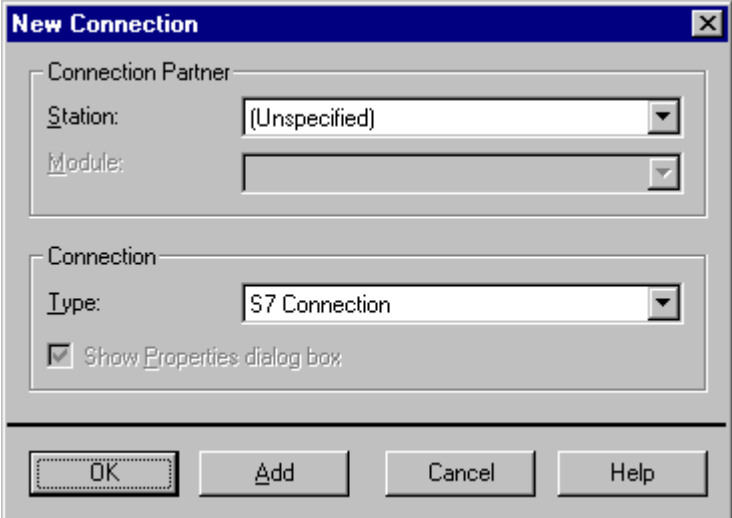
步骤	过程：组态硬件
4	<p>硬件目录将会显示。</p> <p>所选择的第一个组件是机架。在此机架上将安装所有其它组件。通过  或者拖放，将机架插入项目中。在本实例中使用的机架类型是 UR1。</p> 
5	<p>HW Config 程序将会显示当前的空机架。它接收到机架号 0。在组态 WinCC 项目中的连接时，机架号是必须设置的参数之一。</p> 

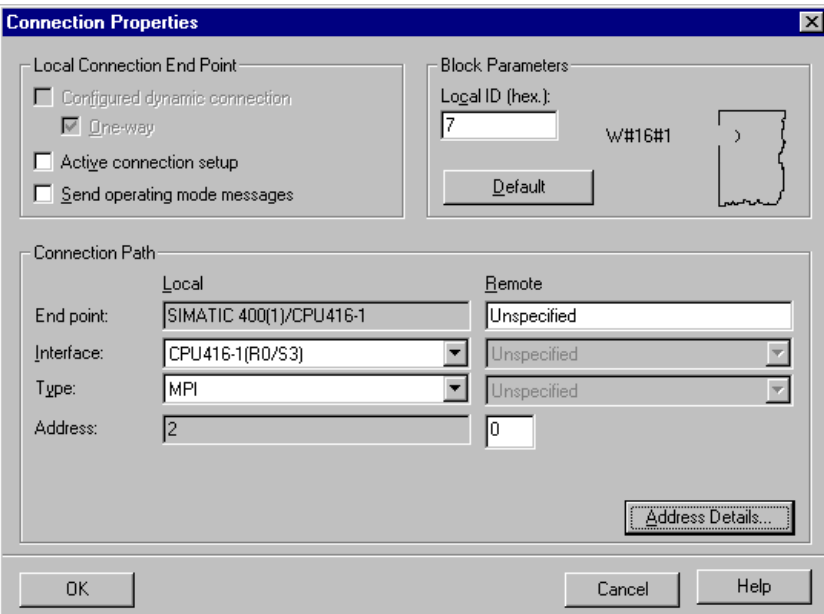
步骤	过程：组态硬件
6	<p>排列机架中的其它硬件组件。这通过将期望的组件从 <i>硬件目录</i> 拖放至机架中的相应插槽来完成。</p> <p>本实例使用的电源为 <i>PS 407 10A</i>。它插在插槽 <i>1</i> 中。这种类型的电源占用两个插槽。</p> <p>本实例使用的 CPU 模块为 <i>CPU 416-1</i>。此模块插在插槽 <i>3</i> 中。在组态 WinCC 项目中的连接时，另一个要设置的参数是 CPU 模块的插槽号。</p>
7	<p>设置 CPU 模块的属性。为此，通过  机架图标来打开 CPU 模块的属性对话框。</p> <p>在 <i>常规信息</i> 标签中，初步通讯设置在 <i>接口</i> 域内显示。CPU 模块的接口已定义为未进行网络连接。这通过 <i>属性</i> 按钮来更改。</p> 

步骤	过程：组态硬件
8	<p>将显示 MPI 接口的属性对话框。</p> <p>在本实例中，地址仍然为 2。在子网域内，选择 <i>MPI(1)</i> 条目。这将 CPU 模块分配给 <i>MPI 网络</i>。</p> <p>如果需要，可以通过 <i>属性</i> 按钮来更改 <i>MPI 网络</i> 的属性。在这种情况下，也必须修改安装通讯处理器 <i>CP 5412 A2</i> 时所作的设置。然而在本实例中，传输率保持为 <i>187.5 Kbit/s</i>，最高站地址保持为 <i>31</i>。</p> <p>对于最新版本的 CPU 模块，传输率可以大于 <i>187.5 Kbit/s</i>。</p> <p>用 <i>确定</i> 按钮关闭 MPI 接口的属性对话框。CPU 模块的属性对话框也用 <i>确定</i> 按钮来关闭。</p> 
9	<p>保存所做的设置。</p> <p>此外，将创建的硬件组态传送至 PLC。这通过如下所示的工具栏按钮来完成。</p> 

步骤	过程：组态硬件						
10	<p>将显示一个对话框，从中可以选择要装载的组件。</p> <p>在本实例中，只有 CPU 模块的条目可用。注意：只有当运行模式开关设置为 <i>STOP</i>或 <i>RUN-P</i>时，才可以装载到 CPU 模块。单击<i>确定</i>按钮关闭对话框。</p> <p>现在将显示<i>选择站地址</i>对话框。</p> <p>在此对话框中指定 STEP7 软件用于与 CPU 模块进行通讯的站地址。在本实例中，通过 MPI 接口进行通讯。CPU 模块的<i>地址</i>是 2。单击<i>确定</i>按钮关闭对话框。</p> <p>组态数据现在将传送至 PLC。为此，将 CPU 模块设置为 <i>STOP</i>状态。</p> <p>可以退出 <i>HW Config</i> 程序。</p>  <table><tr><th>Module</th><th>Rack</th><th>Slot</th></tr><tr><td>CPU416-1</td><td>0</td><td>3</td></tr></table>	Module	Rack	Slot	CPU416-1	0	3
Module	Rack	Slot					
CPU416-1	0	3					

创建 S7 连接

步骤	过程: 创建 S7 连接
1	<p>通过  右边窗口中的 CPU 模块 <i>CPU 416-1</i> 的条目, 将会显示连接条目。通过  连接条目或者  然后从弹出式菜单中选择 <i>打开对象</i> 来启动 <i>NetPro</i> 程序。</p> 
2	<p>当前的网络组态通过 <i>NetPro</i> 程序来显示。 会显示 <i>SIMATIC S7-400</i> 站。其 CPU 模块已经连接至 <i>MPI</i> 网络。 选择 CPU 模块。这会激活插入  连接菜单。该菜单用于创建新连接。在连接伙伴域中, 为站设置条目(未指定)。在连接域中, 为类型设置条目 <i>S7 连接</i>。 通过 <i>确定</i> 按钮打开用于更详细地定义连接的对话框。</p> 

步骤	过程：创建 S7 连接
3	<p>将显示连接的属性对话框。</p> <p>撤消选定 <i>建立主动连接</i> 复选框。将本地 ID 分配给连接。本实例中，在本地 ID (十六进制) 域内输入数值 7。对于伙伴的地址，则输入通讯处理器 CP 5412 A2 的数值 0。</p> <p>通过 <i>地址细节</i> 按钮来设置连接的附加参数。</p> <div></div>

步骤

过程：创建 S7 连接

4

将显示地址细节对话框。

对于 伙伴 的机架号和插槽号，都使用 0。对于连接资源，本地站与伙伴必须使用相同的数值。在本实例中，使用数值 10。

可以通过 确定按钮关闭地址细节对话框。连接的属性对话框也可以通过 确定按钮来关闭。

Address Details

LocalRemote

End point:

SIMATIC 400(1)/CPU416-1Unspecified

Rack / slot:

03

00

Connection resource (hex.):

1010

TSAP:

10.0310.00

OK

Cancel

Help

5

在下表中将会显示新创建的连接。选择该连接。通过 目标系统 → 装载 → 所选择的连接菜单，将连接装载到 PLC 中。

保存在 NetPro 程序中所做的组态。可以退出该程序。

Local ID	Remote ID	Partner	Type	Active Connection Setup	Send Operating Mode Messages
7		Unspecified	S7 Connection	No	No

创建 STEP7 程序

步骤

过程：创建 STEP7 程序

1

创建用于报文数据的数据块。

这可以在 *SIMATIC 管理器* 中通过以下方法来完成，即  CPU 模块的 *S7 程序(1)*条目的 *模块子*条目，然后从弹出式菜单中选择*插入新对象* → *数据块*。本实例使用 *DB25* 至 *DB30* 作为数据块的名称。

通过  数据块，或者  然后从弹出式菜单中选择*打开对象*，可以编制块的内容。这将启动程序 *LAD/STL/SFC*。

这些数据块包含各种作业的报文数据。为每个完成的作业，本实例使用独立的数据块。记录编号和域编号的数据范围在初始阶段为空。仅在报文发送前才用当前数值来填充。同样，只有在报文发送前才输入可能需要的作业数据。

下面描述了完整的 *DB26* 数据块。该数据块包含了用于读取*用户归档*中数据记录的作业的报文数据。



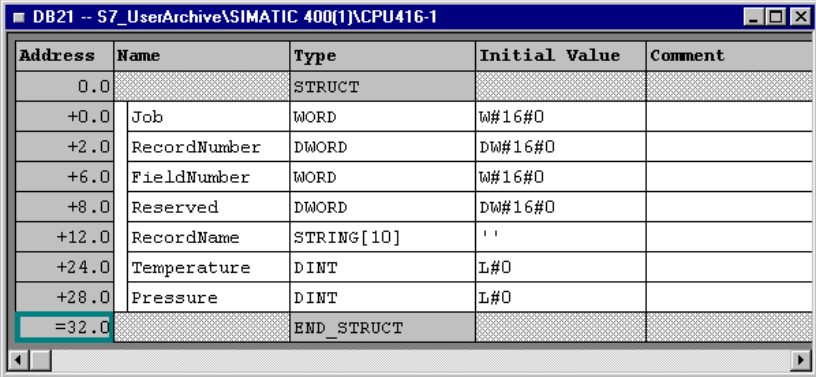
Address	Name	Type	Initial Value	Comment
0.0		STRUCT		
+0.0	TelegramLength_01	BYTE	B#16#1C	Telegramlength: 28
+1.0	TelegramLength_02	BYTE	B#16#0	
+2.0	TelegramLength_03	BYTE	B#16#0	
+3.0	TelegramLength_04	BYTE	B#16#0	
+4.0	TransferType	BYTE	B#16#2	Transfertype: 2
+5.0	Reserved_01	BYTE	B#16#0	
+6.0	NumberOfJobs_01	BYTE	B#16#1	Number of Jobs: 1
+7.0	NumberOfJobs_02	BYTE	B#16#0	
+8.0	PLCID	ARRAY[1..8]	'U', 'S', 'R', 'A',	PLCID: USRARCS7
+9.0		CHAR		
+16.0	JobLength_01	BYTE	B#16#C	Joblength: 12
+17.0	JobLength_02	BYTE	B#16#0	
+18.0	JobType	BYTE	B#16#6	Jobtype: 6
+19.0	Reserved_02	BYTE	B#16#0	
+20.0	FieldNumber	WORD	W#16#0	
+22.0	Recordnumber	DWORD	DW#16#0	
+26.0	Selection	WORD	W#16#0	
+28.0		END_STRUCT		

2

创建与*用户归档*进行通讯的数据块。

本实例中使用 *DB23* 作为数据块的名称。由 WinCC 发送的确认报文存储在该数据块中。

数据块的大小取决于最大报文长度。在本实例中报文长度为 42 个字节，它在进行数据记录的读作业时从归档中产生。其长度由 24 字节的确认信息与 18 字节的作业数据组成。

步骤	过程：创建 STEP7 程序
3	<p>创建一个附加的数据块。</p> <p>本实例用 <i>DB21</i> 作为此数据块的名称。</p> <p>其中，<i>DB21</i> 包含控制变量的数据范围、域编号以及记录编号。可以使用 WinCC 变量使三个数据全部可视化，并在运行系统中对其修改以进行模拟。</p> <p>此外，该数据块还包含数据记录的数据。该数据与<i>用户归档</i>当前所请求的数据记录相关。用 WinCC 变量可以使数据可视化，并且为了进行模拟可以在运行系统中对其进行修改。</p> <p>下面描述了完成的 <i>DB21</i> 数据块。</p> <div></div>

步骤

过程：创建 STEP7 程序

4

创建一个功能块，它将执行与*用户归档*的通讯。在本实例中，它是 *FB100*。该功能块通过 *OB1* 周期性地调用。












创建用于处理某种作业类型的作业的功能块。这些功能块在 *FB100* 中顺序调用。在本实例中，使用功能块 *FB106* 至 *FB110*。

根据分配给每个作业类型的标志的状态，在这些功能块中通过系统功能块 *SFB12 BSEND* 将作业报文发送给用户归档。对于 *SFB12*，必须提供一个情景块。在本实例中，它是 *DB12*。

创建一个功能块，用于接收由*用户归档*发送的确认报文。此功能块也通过 *FB100* 来调用。在本实例中，它是 *FB120*。


根据作业报文是否已经发送给用户归档，*FB120* 通过系统功能块 *SFB13 BRCV* 接受*用户归档*的确认报文。对于 *SFB13*，必须提供一个情景块。在本实例中，它是 *DB13*。

在此表之后，STEP7 程序的操作通过功能块 *FB100*、*FB106* 与 *FB120* 描述得更详细。

Object Name	Type	Language	Author	Comment
 OB1	Organization Block	STL	zip	PROGRAM_CYCLE
 FB100	Function Block	STL	zip	ARCHIVE_COMMUNICATION
 FB105	Function Block	FBD	zip	DELETE_ARCHIVE
 FB106	Function Block	FBD	zip	READ_RECORD
 FB107	Function Block	FBD	zip	WRITE_RECORD
 FB108	Function Block	FBD	zip	DELETE_RECORD
 FB109	Function Block	FBD	zip	READ_FIELD
 FB110	Function Block	FBD	zip	WRITE_FIELD
 FB120	Function Block	FBD	zip	RECEIVE_RESPONSE
 SFB12	System Function Block	STL	SIMATIC	
 SFB13	System Function Block	STL	SIMATIC	

5

保存创建的块并将其装载到 PLC 中。这通过如下所示的工具栏按钮来完成。注意：只有当运行模式开关设置为 *STOP* 或 *RUN-P* 时，才可以装载到 CPU 模块。



Download

FB100 (ARCHIVE_COMMUNICATION)

```

//Dont load a new Job if the previous Job is still running
    A    M    32.0
    JC    BUSY
//Load new Job
    L    DB21.DBW    0
    T    MW    30
//Exit if no Job to do
    L    0
    ==I
    BEC
//Delete Job
    SET
    T    DB21.DBW    0
//Lock following Jobs
    S    M    32.0
//Execute Job
BUSY: UC    FB    105           //DELETE_ARCHIVE
      UC    FB    106           //READ_RECORD
      UC    FB    107           //WRITE_RECORD
      UC    FB    108           //DELETE_RECORD
      UC    FB    109           //READ_FIELD
      UC    FB    110           //WRITE_FIELD
//Receive Response Telegram
      UC    FB    120           //RECEIVE_RESPONSE
//End
    BE

```

- *M32.0* 标志表示作业当前正在运行。如果设置了 *M32.0* 标志，则不会接受新作业并且跳转到 *BUSY*位置。
- *DB21.DBW0* 数据字的各个位请求来自*用户归档*的某种作业类型的作业。在本实例中，WinCC 写入数据字。在 WinCC 中实现使得一次只能设置数据字的一个位。
- *MW30* 标志字存储 *DB21.DBW0* 数据字的内容。此标志字的各个标志位触发来自*用户归档*的某种作业类型的作业。
- 如果没有设置 *M32.0* 标志，则将 *DB21.DBW0* 数据字写入 *MW30* 标志字。
- 如果不需要执行任何作业，则终止该块。
- 如果需要执行一个作业，则删除 *DB21.DBW0* 数据字的内容。将会设置 *M32.0*标志。
- 按顺序调用处理各种作业类型的功能块。
- 调用负责接受确认报文的功能块。

FB106 (READ_RECORD)

```

//Set Telegram Length
L      28
T      MW      103
//Fill Send Telegram if Job to do
AN      M      31.1
JC      SEND
//Fill Record Number
L      DB21.DBB      2
T      DB26.DBB      25
L      DB21.DBB      3
T      DB26.DBB      24
L      DB21.DBB      4
T      DB26.DBB      23
L      DB21.DBB      5
T      DB26.DBB      22
//Record to Receive
SET
S      M      33.0
//Send Delete Telegram
SEND: CALL "BSEND" , DB12
      REQ      :=M31.1
      R      :=M104.0
      ID      :=W#16#7
      R_ID     :=DW#16#5
      DONE     :=M104.1
      ERROR    :=M104.2
      STATUS   :=MW105
      SD_1     :=F#DB26.DBX 0.0 BYTE 46
      LEN      :=MW103
//Stop Sending of Data when ready
A      M      104.1
R      M      31.1
BE

```

- *MW103* 标志字包含要传送数据的长度。从*用户归档*进行读作业的报文长度为 28 字节。
- *M31.1* 标志触发从*用户归档*读取数据的作业。如果它没有设置，则跳转到 *SEND* 位置。
- 如果设置了 *M31.1* 标志，则把来自 *DB21* 的要处理的记录编号写入包含读作业的报文数据的 *DB26*。在这种情况下，各字节的顺序必须反置。
- 对用于接收确认报文的 *FB120* 块，*M31.1* 标志表示是否已经接收到一个完整的数据记录。它被设置。
- 通过系统功能块 *SFB12 BSEND* 发送作业报文。它的调用参数描述如下。
 - **REQ:** 激活数据交换。在本实例中，如果设置了 *M31.1* 标志，则发送报文。
 - **R:** 取消正在进行的数据交换。在本实例中没有使用。
 - **ID:** 注明本地连接描述。在本实例中，它在用十六进制数值 7 创建 S7 连接时已进行定义。

- **R_ID**: 注明通讯伙伴。此处所设置的数值必须在创建用于处理通讯的原始数据变量时指定。
- **DONE**: 表示作业处理无错。在本实例中, 数值存储在 *M104.1* 标志内。如果它被设置, 则触发作业的 *M31.1* 标志将会重新设置。
- **ERROR**: 表示作业处理有错。在本实例中没有使用。
- **STATUS**: 提供有关产生的错误类型的详细信息。在本实例中没有使用。
- **SD_1**: 指向要发送的数据的指针。在本实例中, 数据要从 *DB26* 发送。
- **LEN**: 要发送的数据的长度。在本实例中, 它存储在 *MW103* 内。
- 如果作业已经无错执行(设置了 *M104.1* 标志), 则重新设置 *M31.1* 标志。

FB120 (RECEIVE_RESPONSE)

```

//Set Telegram Length
L      42
T      MW      203
//Receive Response Telegram
CALL   "BRCV" , DB13
EN_R   :=M32.0
ID      :=W#16#7
R_ID    :=DW#16#5
NDR     :=M200.0
ERROR   :=M200.1
STATUS:=MW201
RD_1    :=P#DB23.DBX 0.0 BYTE 42
LEN     :=MW203
//Exit if not ready
AN      M      200.0
BEC
//Cleanup
SET
R      M      200.0
R      M      32.0
//Check Response
L      DB23.DBB      6
T      MB      35
L      0
<>I
BEC
//Get received Record
AN      M      33.0
JC      FLD
//...code to get received record not displayed
//...
//Get received Field
FLD: AN      M      33.1
JC      END
//...code to get received record not displayed
//...
END: BE

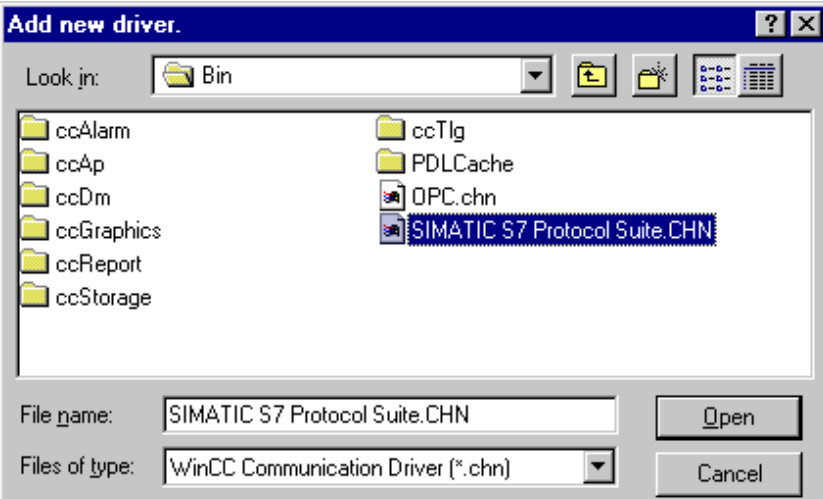

```

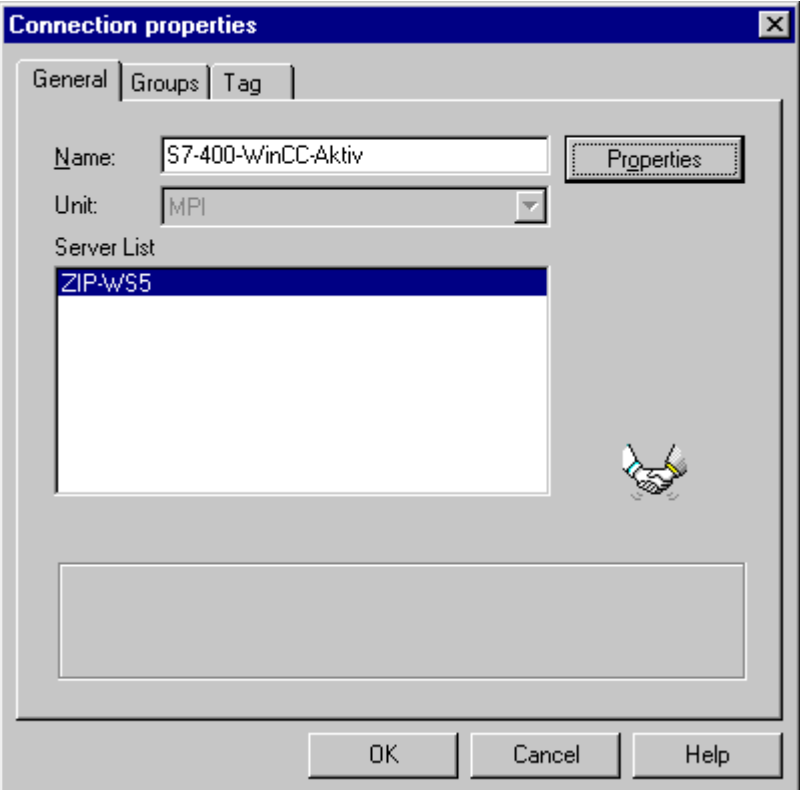
- *MW103* 标志字包含要接收的数据的长度。确认报文的最大长度为 42 字节。它在进行从 *用户归档* 中读取数据记录的作业时产生。
- 通过系统功能块 *SFB13 BRCV* 接收确认报文。它的调用参数描述如下。
 - **EN_R**: 允许接收数据。在本实例中, *M32.0* 标志控制块的准备接收状态。因此, 它当仍在处理作业时也准备接收数据。
 - **ID**: 注明本地连接描述。在本实例中, 它用十六进制数值 7 创建 S7 连接时已进行定义。
 - **R_ID**: 注明通讯伙伴。此处所设置的数值必须在创建用于处理通讯的原始数据变量时指定。
 - **NDR**: 表明数据传送无错。在本实例中, 数值存储在 *M200.0* 标志内。如果它被设置, 则标明作业正在进行的 *M32.0* 标志将会重新设置。
 - **ERROR**: 表明数据传送有错。在本实例中没有使用。
 - **STATUS**: 提供有关产生的错误类型的详细信息。在本实例中没有使用。
 - **RD_1**: 指向存储所接收数据的数据范围的指针。在本实例中, 它是 DB23。
 - **LEN**: 要接收的数据的长度。在本实例中, 它存储在 *MW103* 内。
- 如果数据传送还没有完成, 则终止该块。
- 如果数据传送已完成, 则通过重新设置 *M32.0* 标志将作业标记为完成。
- 从确认报文载入错误代码。它存储在 *MW35* 标志字中。*MW35* 标志字在 WinCC 中进行判断。如果错误代码的值不等于 0, 则表示产生了一个错误。在这种情况下, 终止块。
- 如果没有产生错误, 则必须检查是否已获得归档数据, 如果已获得, 其类型是什么。*M33.0* 标志表明到达的数据记录, *M33.1* 标志表明到达的数据域。在上述的 *FB120* 功能块中, 因为考虑到其长度, 所以没有显示用于数据传送的段。它们已经用注释进行替代。

5.1.8.3 在 WinCC 中组态


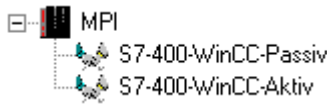
创建连接


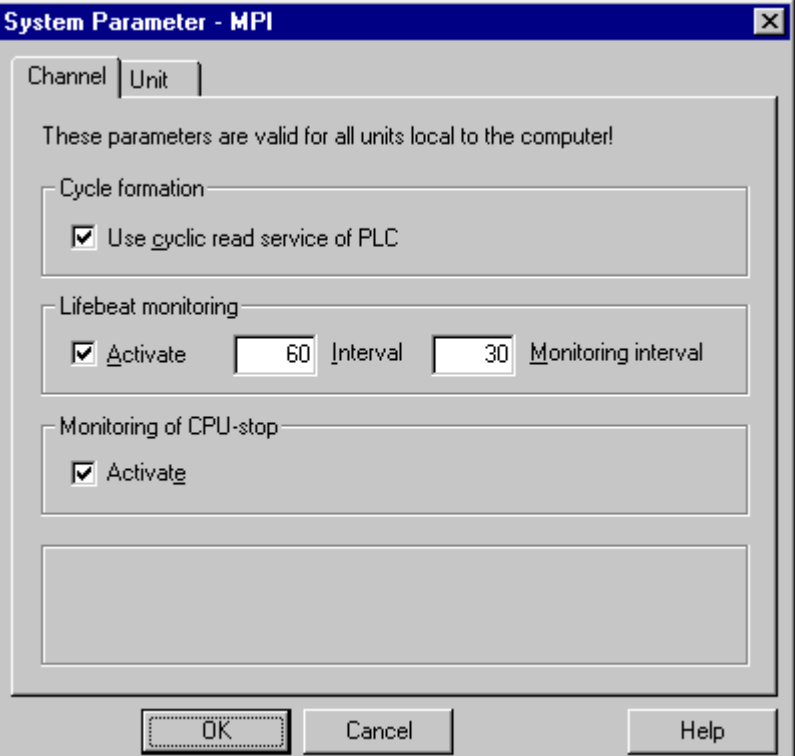
步骤	过程: 创建连接
1	<p>所需通讯驱动程序的安装。这可以通过  <i>变量管理器</i>, 然后从弹出式菜单中选择 <i>添加新驱动程序</i> 来完成。</p> 

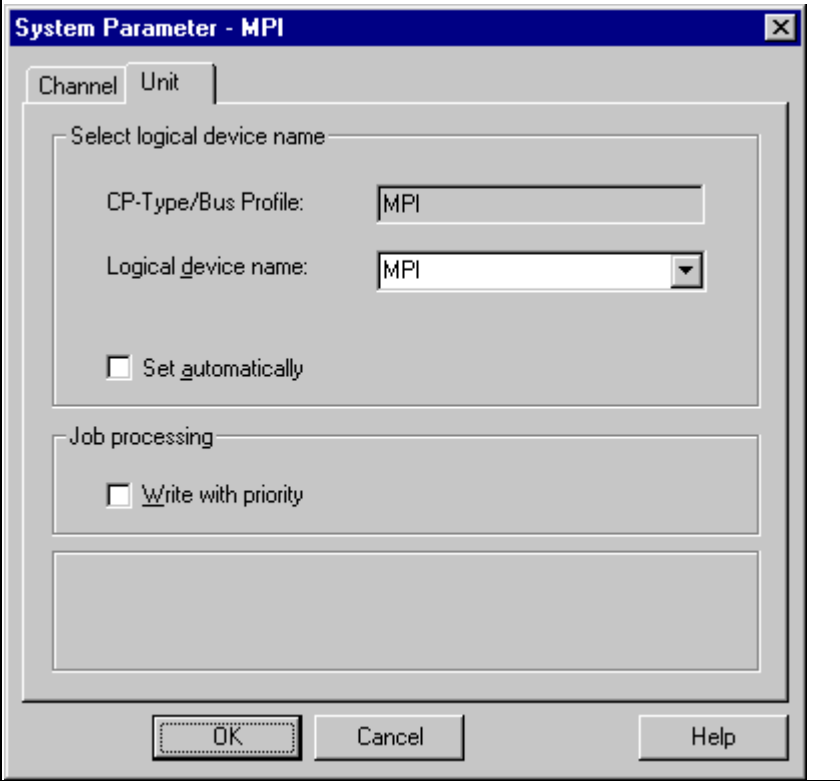
步骤	过程：创建连接
2	<p>将显示添加新驱动程序对话框。</p> <p>该对话框列出所有可以安装的通讯驱动程序。为与 <i>SIMATIC S7</i> 进行通讯，必须安装驱动程序 <i>SIMATIC S7 Protocol Suite</i>。从对话框中选择该驱动程序。通过单击打开退出对话框。</p> 
3	<p>新添加的驱动程序 <i>SIMATIC S7 Protocol Suite</i> 将作为变量管理器的子条目显示。</p> <p>该驱动程序包含 8 个不同的通道单元。对于通过 <i>MPI 网络</i>进行通讯，可使用 <i>MPI</i> 通道单元。</p> <p>对于 <i>MPI</i> 通道单元，通过  <i>MPI</i> 条目然后从弹出式菜单中选择新建驱动程序连接来创建新连接。该连接用于从 <i>PLC</i> 中主动发送数据。在这种情况下，<i>WinCC</i> 为主动伙伴。</p> 

步骤	过程：创建连接
4	<p>将显示连接的属性对话框。</p> <p>在 <i>常规</i> 标签中，输入新连接的 <i>名称</i>。在本实例中，它是 <i>S7-400-WinCC-Active</i>。</p> <p>单击 <i>属性</i> 按钮来定义连接属性。</p> 


























步骤	过程：创建连接
5	<p data-bbox="524 306 1351 451">将显示连接属性对话框。 在站地址域中，输入为 CPU 模块的 MPI 接口所设置的地址。在本实例中，它是地址 2。 另外，必须输入要访问的 CPU 模块的机架号和插槽号。 单击确定按钮关闭对话框。同样单击确定关闭连接属性对话框。</p> <div data-bbox="524 457 1321 1236"></div>


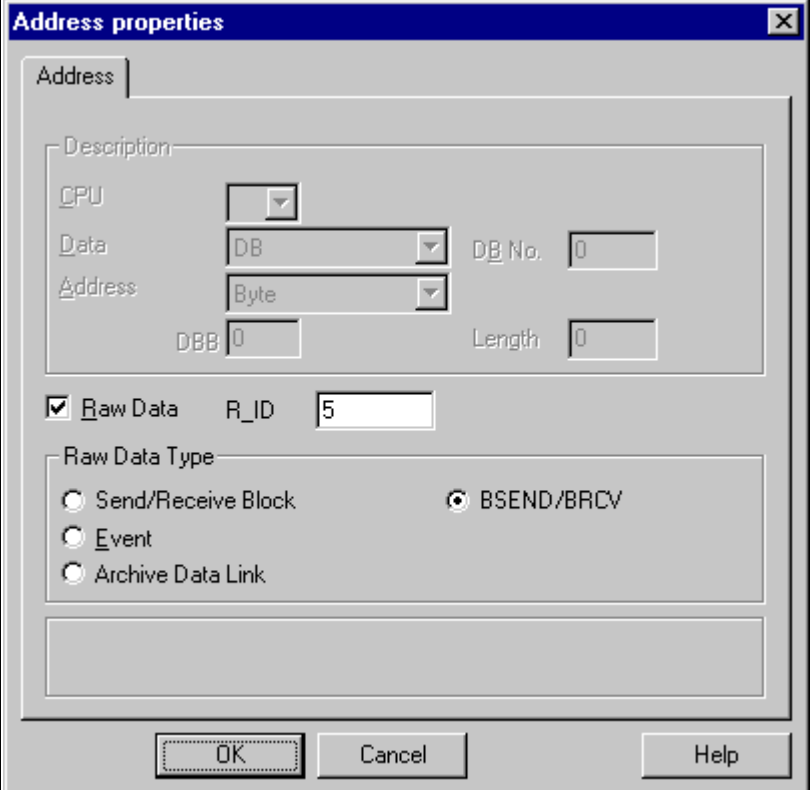
步骤	过程: 创建连接
6	<p>为 <i>MPI</i>通道单元创建第二个连接</p> <p>可通过  <i>OR MPI</i> 条目, 然后从弹出式菜单中选择 <i>新建驱动程序连接</i>来完成。<i>PLC</i> 使用该连接主动写数据。在这种情况下, <i>WinCC</i> 为被动伙伴。</p> <p>将显示连接的属性对话框。</p> <p>在 <i>常规</i> 标签中, 输入新连接的 <i>名称</i>。在本实例中, 它是 <i>S7-400-WinCC-Passive</i>。通过 <i>属性</i>按钮来定义连接属性。</p> <p>将显示 <i>连接属性</i>对话框。</p> <p>对于 <i>站地址</i>、<i>机架号</i>和 <i>插槽号</i>, 必须使用与先前创建的 <i>S7-400-WinCC-Active</i> 连接中所指定值相同的数值。</p> <p>选择 <i>发送/接收原始数据块</i>复选框。在 <i>连接资源域</i>中, 必须输入当创建 <i>S7</i> 连接时在 <i>PLC</i> 中指定的连接资源。在本实例中, 使用数值 <i>10</i>。</p> <p>单击 <i>确定</i>按钮关闭对话框。同样单击 <i>确定</i>关闭 <i>连接属性</i>对话框。</p> 
7	<p>刚创建的连接现在将显示在 <i>WinCC 资源管理器</i>中。</p> 

步骤	过程: 创建连接
8	<p>设置 MPI 通道单元的系统参数。</p> <p>在系统参数对话框中进行这些设置，该对话框可通过  R MPI 条目，然后从弹出式菜单中选择系统参数来访问。</p> <p>在通道标签内，可以指定与通讯以及监控通讯有关的各种设置。这些设置将应用于通讯驱动程序的所有通道单元。</p> 



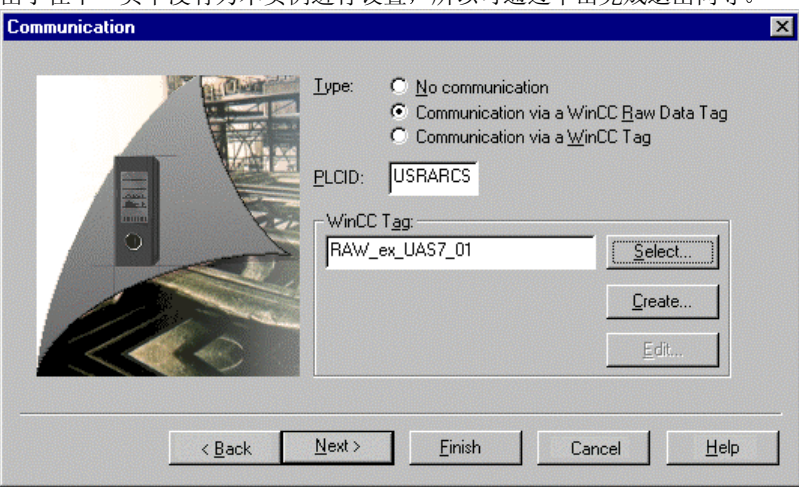









步骤	过程：创建连接
9	<p data-bbox="521 296 1357 451">在 设备标签内，指定访问 PLC 时连接所使用的访问点。 设置 <i>MPI</i> 访问点。先前已将通讯处理器 <i>CP 5412 A2</i> 分配给了程序设置 <i>PG/PC 接口</i>中的该访问点。如果要自动设置访问点，应确保使用正确的通讯处理器，尤其是在使用多个通讯处理器时。 单击 <i>确定</i>按钮关闭对话框。</p> <div data-bbox="521 451 1357 1226"></div>

创建变量

步骤	过程：创建变量																											
1	<p>创建用于 <i>S7-400-WinCC-Active</i> 连接的变量。这可在 <i>WinCC 资源管理器</i> 中通过  相应的连接条目，然后从弹出式菜单中选择新建变量来完成。</p> <p>各变量的名称、数据类型和地址在下图中列出。</p> <p>变 量 <i>T08w_ex_UAS7_RecordName</i> 、 <i>S32w_ex_UAS7_Pressure</i> 和 <i>S32w_ex_UAS7_Temperature</i> 代表 PLC 中当前的数据记录，它存储在 <i>DB21</i> 中。</p> <p>变 量 <i>U32w_ex_UAS7_Record</i> 、 <i>U16w_ex_UAS7_Field</i> 和 <i>U16w_ex_UAS7_Job</i> 代表要处理的记录编号和域编号以及控制变量，它们也存储在 <i>DB21</i> 中。</p> <p>变量 <i>U08w_ex_UAS7_Busy</i> 和 <i>U08w_ex_UAS7_Error</i> 用于显示当前作业状态。</p> <table><tr><th>Name</th><th>Type</th><th>Parameters</th></tr><tr><td> <i>U08w_ex_UAS7_Busy</i></td><td>Unsigned 8-bit value</td><td>MB32</td></tr><tr><td> <i>T08w_ex_UAS7_RecordName</i></td><td>Text tag 8-bit character set</td><td>DB21,DBB12</td></tr><tr><td> <i>U16w_ex_UAS7_Job</i></td><td>Unsigned 16-bit value</td><td>DB21,DW0</td></tr><tr><td> <i>S32w_ex_UAS7_Pressure</i></td><td>Signed 32-bit value</td><td>DB21,DD28</td></tr><tr><td> <i>S32w_ex_UAS7_Temperature</i></td><td>Signed 32-bit value</td><td>DB21,DD24</td></tr><tr><td> <i>U32w_ex_UAS7_Record</i></td><td>Unsigned 32-bit value</td><td>DB21,DD2</td></tr><tr><td> <i>U16w_ex_UAS7_Field</i></td><td>Unsigned 16-bit value</td><td>DB21,DW6</td></tr><tr><td> <i>U08w_ex_UAS7_Error</i></td><td>Unsigned 8-bit value</td><td>MB35</td></tr></table>	Name	Type	Parameters	 <i>U08w_ex_UAS7_Busy</i>	Unsigned 8-bit value	MB32	 <i>T08w_ex_UAS7_RecordName</i>	Text tag 8-bit character set	DB21,DBB12	 <i>U16w_ex_UAS7_Job</i>	Unsigned 16-bit value	DB21,DW0	 <i>S32w_ex_UAS7_Pressure</i>	Signed 32-bit value	DB21,DD28	 <i>S32w_ex_UAS7_Temperature</i>	Signed 32-bit value	DB21,DD24	 <i>U32w_ex_UAS7_Record</i>	Unsigned 32-bit value	DB21,DD2	 <i>U16w_ex_UAS7_Field</i>	Unsigned 16-bit value	DB21,DW6	 <i>U08w_ex_UAS7_Error</i>	Unsigned 8-bit value	MB35
Name	Type	Parameters																										
 <i>U08w_ex_UAS7_Busy</i>	Unsigned 8-bit value	MB32																										
 <i>T08w_ex_UAS7_RecordName</i>	Text tag 8-bit character set	DB21,DBB12																										
 <i>U16w_ex_UAS7_Job</i>	Unsigned 16-bit value	DB21,DW0																										
 <i>S32w_ex_UAS7_Pressure</i>	Signed 32-bit value	DB21,DD28																										
 <i>S32w_ex_UAS7_Temperature</i>	Signed 32-bit value	DB21,DD24																										
 <i>U32w_ex_UAS7_Record</i>	Unsigned 32-bit value	DB21,DD2																										
 <i>U16w_ex_UAS7_Field</i>	Unsigned 16-bit value	DB21,DW6																										
 <i>U08w_ex_UAS7_Error</i>	Unsigned 8-bit value	MB35																										


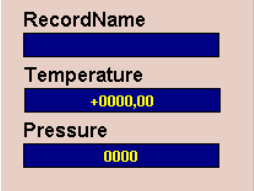
步骤	过程: 创建变量
2	<p>创建用于 <i>S7-400-WinCC-Passive</i> 连接的变量。这可在 <i>WinCC 资源管理器</i>中通过  相应的连接条目, 然后从弹出式菜单中选择新建变量来完成。</p> <p>将显示该变量的属性对话框。在本实例中, 该变量的名称为 <i>RAW_ex_UAS7_01</i>。将原始数据类型设置为该变量的数据类型。</p> <p>通过 <i>选择</i>按钮, 打开对变量编址的对话框。</p> <p>选择 <i>原始数据</i>复选框。在 <i>原始数据类型</i>域内, 选择选项钮 <i>BSEND/BRCV</i>。在 <i>R_ID</i> 域中, 输入在 STEP7 程序中指定的 <i>R_ID</i> 参数的值, 以调用 <i>BSEND</i> 和 <i>BRCV</i>系统功能块。在本实例中, 它是数值 5。</p> <p>通过 <i>确定</i>按钮关闭 <i>地址属性</i>对话框。同样通过 <i>确定</i>按钮关闭 <i>变量属性</i>对话框。</p>  <p>The image shows the 'Address properties' dialog box. It has a title bar with 'Address properties' and a close button. Inside, there's a 'Description' section with dropdowns for 'CPU', 'Data' (set to 'DB'), and 'Address' (set to 'Byte'). There are input fields for 'DB No.' (0) and 'Length' (0). Below this is a 'Raw Data' section with a checked checkbox and an 'R_ID' input field containing the value '5'. Underneath is a 'Raw Data Type' section with four radio buttons: 'Send/Receive Block', 'Event', 'Archive Data Link', and 'BSEND/BRCV' (which is selected). At the bottom are 'OK', 'Cancel', and 'Help' buttons.</p>


创建用户归档

步骤	过程：创建用户归档																
1	<p>打开用户归档编辑器。在该编辑器中，通过  归档条目，然后从弹出式菜单中选择新建归档来创建新的归档。这样可启动用于创建归档的向导。</p> 																
2	<p>在该向导的第一页上，输入归档名称。在本实例中，将名称 <i>UserArchive_S7</i> 输入归档名称域内。别名域保留空白。</p> <p>对于归档类型，则选择无限制。</p> <p>通过单击 下一步 继续到下一页。</p>																
3	<p>在向导的第二页上，选择选项钮通过 <i>WinCC 原始数据变量进行通讯</i>。</p> <p>对于 <i>PLCID</i>，本实例使用名称 <i>USRARCS7</i>。该 ID 不能超过 8 个字符。通过该 ID，由 PLC 发送的作业报文可以分配给某个归档。</p> <p>通过 选择 按钮，设置为归档通讯所组态的 <i>WinCC 原始数据变量</i>。在本实例中，它就是变量 <i>RAW_ex_UAS7_01</i>。</p> <p>由于在下一页中没有为本实例进行设置，所以可通过单击 完成 退出向导。</p> 																
4	<p>为刚创建的归档，创建列于下表的域。</p> <p>保存所作的设置。</p> <table><tr><th>Name</th><th>Alias</th><th>Type</th><th>Length</th></tr><tr><td> Pressure</td><td></td><td>Number (integer)</td><td></td></tr><tr><td> RecordName</td><td></td><td>String</td><td>10</td></tr><tr><td> Temperature</td><td></td><td>Number (double)</td><td></td></tr></table>	Name	Alias	Type	Length	 Pressure		Number (integer)		 RecordName		String	10	 Temperature		Number (double)	
Name	Alias	Type	Length														
 Pressure		Number (integer)															
 RecordName		String	10														
 Temperature		Number (double)															

步骤	过程：创建用户归档				
5	在用户归档编辑器的下部表格窗口中，现在可以为归档创建多个数据记录。				
		ID	RecordName	Temperature	Pressure
	1	1	Record_01	236,23	563
	2	2	Record_02	302,78	399
	3	3	Record_03	278,92	456
	...				

在图形编辑器中的实现

步骤	过程：在图形编辑器中的实现
1	打开图形编辑器并创建一个新画面。在本实例中，它是画面 <i>ex_3_chapter_01b.pdl</i> 。 组态用于显示数据的控件。它是 WinCC 用户归档 - 表格元素。从对象选项板的控件选择菜单中选择它，然后将其置于画面中。
2	通过  刚创建的对象控件 1，打开其属性对话框。 在常规标签中，将先前组态的归档 <i>UserArchive_S7</i> 设置在源域中。在编辑域内，选择插入、更改和删除复选框，以使所有的编辑选项都能为用户所使用。此外，选择边框复选框。 在工具栏标签中，不选择用于按钮写变量和读变量的两个复选框。在本实例中，由 PLC 执行控制。使用工具栏的所有其它按钮。 为了在运行系统中同时显示所有的列，在字体标签中，将字体的大小缩小到 10。 在 WinCC 用户归档 - 表格元素的属性对话框中所作的设置通过确定按钮来结束。 在本实例中，通过对象控件 1 的属性对话框来使表格的颜色方案与项目的颜色方案相匹配。
3	为了显示当前在 PLC 中装载的数据记录，组态三个智能对象 → I/O 域。在本实例中，它们是对象 I/O 域 1、I/O 域 2 和 I/O 域 3。 对于 I/O 域，在属性 → 输出/输入 → 输出值处为三个变量 <i>T08w_ex_UAS7_RecordName</i> 、 <i>S32w_ex_UAS7_Temperature</i> 和 <i>S32w_ex_UAS7_Pressure</i> 各创建一个变量连接。 

步骤	过程：在图形编辑器中的实现
4	<p>为了指定要处理的记录编号和域编号，另外组态两个智能对象 → I/O 域。在本实例中，它们是对象 I/O 域 4 和 I/O 域 5。</p> <p>对于对象 I/O 域 4，在属性 → 输出/输入 → 输出值处创建与变量 <code>U32w_ex_UAS7_Record</code> 的变量连接。该 I/O 域显示记录编号。对于对象 I/O 域 5，在属性 → 输出/输入 → 输出值处创建一个与变量 <code>U16w_ex_UAS7_Field</code> 的变量连接。该 I/O 域显示域编号。</p> <p>为了触发 PLC 中的动作，组态了 5 个 Windows 对象 → 按钮。在本实例中，它们是对象按钮 9 至按钮 13。</p> <p>如果按下一个按钮，则将一个不同的数值写入到变量 <code>U16w_ex_UAS7_Job</code> 中。在 STEP7 程序中，周期性查询该变量。根据该变量的内容，触发相应的作业。</p> <p>每个数值对应 PLC 的不同作业类型。下面列出了各数值所对应的作业。对于 STEP7 程序中执行的删除所有记录作业，还未组态任何按钮。</p> <ul style="list-style-type: none">• 1 = 删除所有记录• 2 = 从归档中读取记录• 4 = 将记录写入归档• 8 = 从归档中删除记录• 16 = 从归档中读取域• 32 = 将域写入归档 <p>通过重新设定按钮，可以重新设定变量 <code>U08w_ex_UAS7_Busy</code>。这样可进行加锁，以防在没有自动撤消将要删除的新作业报文时将其发送(仅与出错有关)。</p> <p>在状态域中，使用 C 动作显示通讯连接的当前状态以及归档通讯。该 C 动作对变量 <code>U08w_ex_UAS7_Error</code> 进行判断。在本步骤之后描述 C 动作本身。</p> 

用于显示状态的 C 动作

```

#include "apdefap.h"
char* _main(char* lpszPictureName, char* lpszObjectName, char* lpszProperty)
{
    BYTE byError = 0; //communication state archive
    DWORD dwState = 0; //communication state connection

    //activate communication check led
    SetBackColor(lpszPictureName, "LED", CO_RED);

    //communication check
    byError = GetTagByteStateWait("U08w_ex_UAS7_Error", &dwState);

    //deactivate communication check led
    SetBackColor(lpszPictureName, "LED", CO_DKGRAY);

    //if connection error
    if (dwState > 0)
    {
        return "No Connection";
    }

    //switch archive state
    switch (byError)
    {
        case 0:      return "Ready";
        case 1:      return "Error Archive";
        case 2:      return "Error Record";
        case 101:    return "Error Field";
        case 102:    return "Not Supported";
        case 201:    return "Unknown Error";
        case 202:    return "Unknown Error";
        case 254:    return "Unknown Error";
        default:     return "Unknown Error";
    }
}

```

- 在对象 *静态文本 3* 的属性 → 字体 → 文本处已经创建了 C 动作。将 C 动作的触发设置为 2 s。这样将会每 2 秒对连接和归档通讯进行状态检查。
- 由函数 *GetTagByteStateWait* 执行状态检查。将当前的连接状态写入变量 *dwState* 中，将归档通讯状态写入变量 *byError* 中(来自 WinCC 的确认报文的错误代码)。
- 在执行该动作前，将对象 *LED* 的背景色设置为红色。一旦函数结束，则对象 *LED* 的背景色重新设定为灰色。
- 判断变量 *dwState* 和 *byError* 并返回相应的状态文本。

常规应用的注意事项

在进行常规应用之前，必须完成下述修改：

- 如果使用的硬件相同，则可以将所进行的通讯组态直接应用于用户自己的应用程序。否则，必须进行修改。
- 必须根据需要修改归档的结构。

索引

字母

CSV
 用户归档, 5-14
MS Excel
 用户归档, 5-14
OLE, 5-19
 插入, 5-19
WinCC 项目的描述, 2-35, 3-31
WinCC 项目的描述, 4-34

B

编辑器, 5-3
 用户归档编辑器, 5-3
表单视图, 5-27
表格
 组态, 5-19
别名, 5-3
 归档, 5-3
 数据域, 5-3

C

测量器控件, 5-19
常规信息, 4-2
创建
 数据域, 5-3
 用户归档, 5-3
错误
 错误代码, 5-39

D

多客户机的应用, 2-2
多客户机项目的组态, 1-3

F

分布式服务器, 3-1
服务器数据(数据包), 1-8
服务器数据(数据包), 2-3

G

工具栏
 用户归档, 5-19

J

集成
 来自 OCX, 5-19
将多个客户机分配给多个服务器, 1-2

K

可操作, 5-3
 归档, 5-3
控制变量, 5-33

L

连接
 原始数据变量, 5-44

R

冗余, 4-1

S

输入, 5-14
 数据输入, 5-14
数据
 数据域, 5-3
数据导入, 5-14
 用户归档, 5-14

T

通过 WinCC 原始数据变量与 SIMATIC S5
 进行通讯(ex_3_chapter_01a.pdl), 5-44
通过 WinCC 原始数据变量与 SIMATIC S7
 进行通讯(ex_3_chapter_01a.pdl), 5-76
通讯
 用户归档, 5-3

W

文本
 文本库, 5-3

Y

用户归档, 5-2
 Create, 5-3
 编辑器, 5-3
原始数据变量, 5-44
 用户归档, 5-44