SIEMENS

WinCC

Configuration Manual

Manual Volume 3

This manual is part of the documentation package
with the order number:
6AV6392-1CA05-0AB0
C79000-G8276-C164-01

Release: September 1999

C79000-G8276-C164
Printed in the Federal Republic of Germany

Siemens Aktiengesellschaft

Table of Contents

Preface

Purpose of the Manual

This manual introduces you to the configuration options available with WinCC by by means of the following sections:

- New Function Description
- Multi Client
- Distributed Servers
- Redundancy
- User Archives

This manual is available in printed form as well as an electronic online document.
The table of contents or the index will quickly point you to the information desired. The online document also offers an expanded search function.

Requirements for Using this Manual

Basic knowledge of WinCC, for example from the Getting Started manual or through practical experience in the configuration with WinCC.

Additional Support

For technical questions, please contact your Siemens representative at your local Siemens branch.
In addition, you can contact our Hotline at the following number:

+49 (911) 895-7000 (Fax -7001)

Information about SIMATIC Products

Constantly updated information about SIMATIC products can be found in the CA01 catalog. This catalog can be accessed at the following Internet address:

http://www.ad.siemens.de/ca01online/

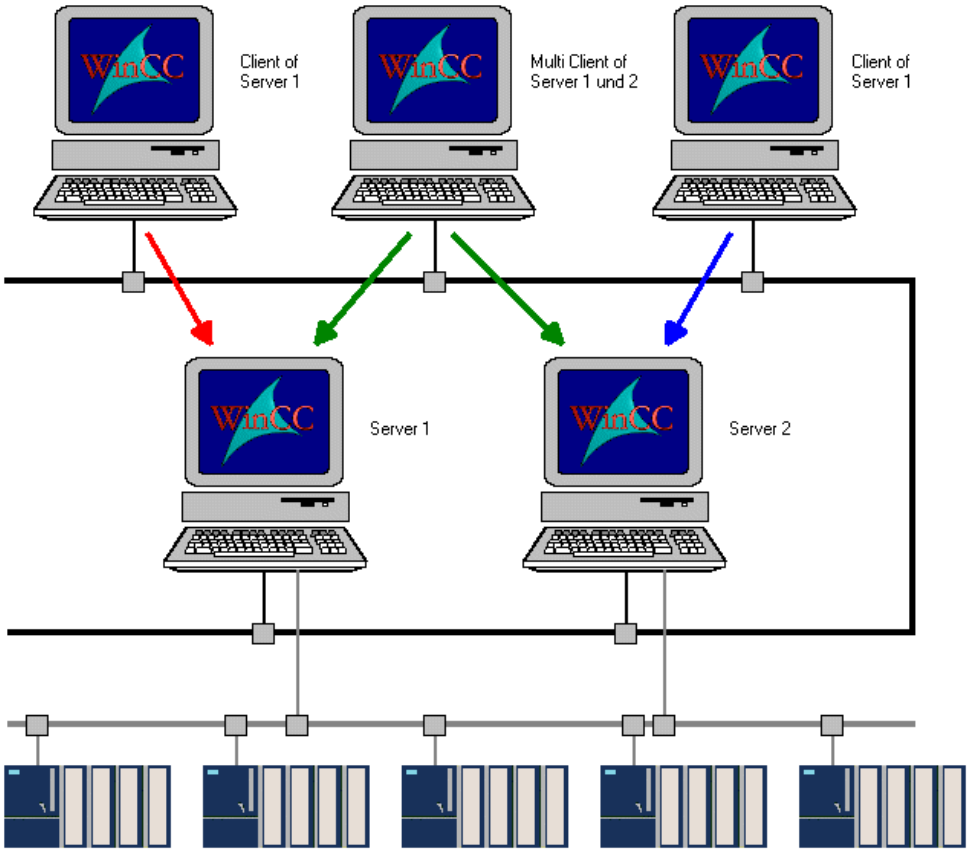
In addition, the Siemens Customer Support provides you with current information and downloads. A compilation of frequently asked questions is available at the following Internet address:

http://www.ad.siemens.de/support/html_00/index.shtml

1 New Function Description

1.1 Distribution of multiple Clients to multiple Servers

SIMATIC WinCC Version 5.0 introduces a new project type: the multi-client project. A multi-client is a client that possesses separate configuration data such as local pictures, scripts and tags. The architecture of SIMATIC WinCC V5.0 enables the multi-client to connect to all available servers on the network. Through these connections, either a functional distribution (separate process data, message and archive servers) or a load distribution (multiple process data, message and archive servers) can be realized.



A server project can be referenced by up to 16 multi-clients, i.e. the sum of the clients and multi-clients that reference one server project must not exceed 16. In this regard, the same limitations as to WinCC V 4.02 apply. A multi-client can access a maximum of 6 server projects. Therefore, the data points that can be displayed by a multi-client are the result of the theoretical limit of 6 * 64k for tags or 6 * 50000 for messages.

1.1.1 Configuration of a Multi-Client Project

A multi-client project can only configure its own data, not the data of a server project. It can, however, reference the data on the servers (provide so-called views to servers).
In the Server Data (Packages) chapter, the mechanism of packages is described. These packages are required to make the relevant data of one or several servers available to a multi-client project during the configuration phase.
The relevant data is:

- Graphics System: Pictures
- Data Manager: Tag Name/Tag Type
- Alarm System: Message Server Yes/No
- Archiving System: Archives with the corresponding Archive Tags
- Text Library: Text IDs
- Group Display: Server Yes/No
- Measurement Points List: Server Yes/No
- PictureTreeManager: Server Yes/No, ID/Text

To connect this configuration data during the configuration of the client project, the existing configuration dialogs are expanded.

Examples:

- The tag dialog is expanded to permit browsing of the tags of the various servers.
- The dialog for connecting a picture of a picture window is expanded to also display the pictures of the servers.

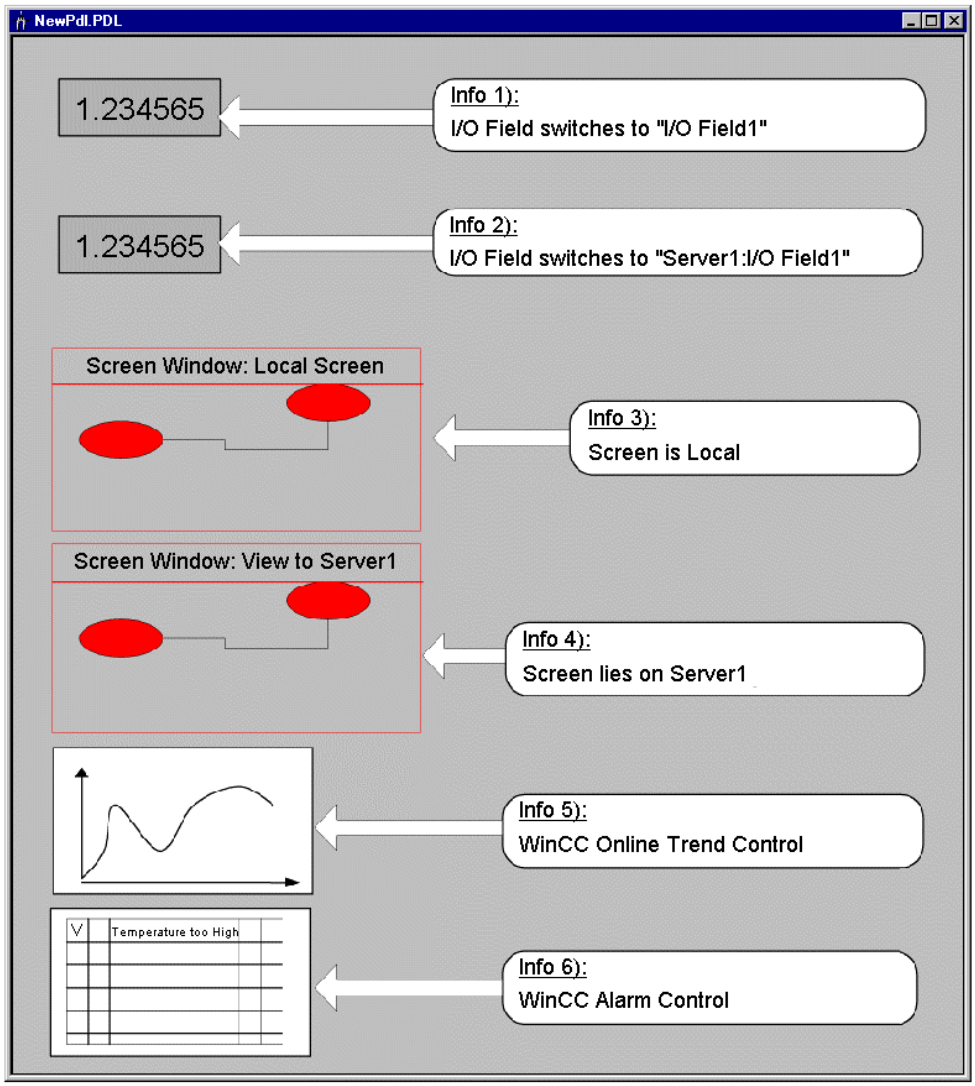
In general, only the editors that are relevant to a client project can be started in the client project.
This information can be entered editor-specific in the mcp.ini file, possibly in the following format:

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

1.1.2 Configuration of a Multi-Client Project

1.1.2.1 Picture Configuration

The client project possesses its own local pictures, but can also use the pictures of one or multiple servers.
The following picture illustrates the possibilities of the multi-client.



- Info 1: I/O Field1 is an internal tag of the client.
- Info 2: Server1:I/O Field1 is a process tag of SERVER1. Here, the SERVER1::I/OField1 tag is addressed via the name service, i.e. the SERVER1 server name is not a physical computer name, but a name that is assigned by the name service.
- The importance is that several servers can be addressed simultaneously in a picture.
- Info 3: In the picture window, a local picture of the multi-client project is displayed.

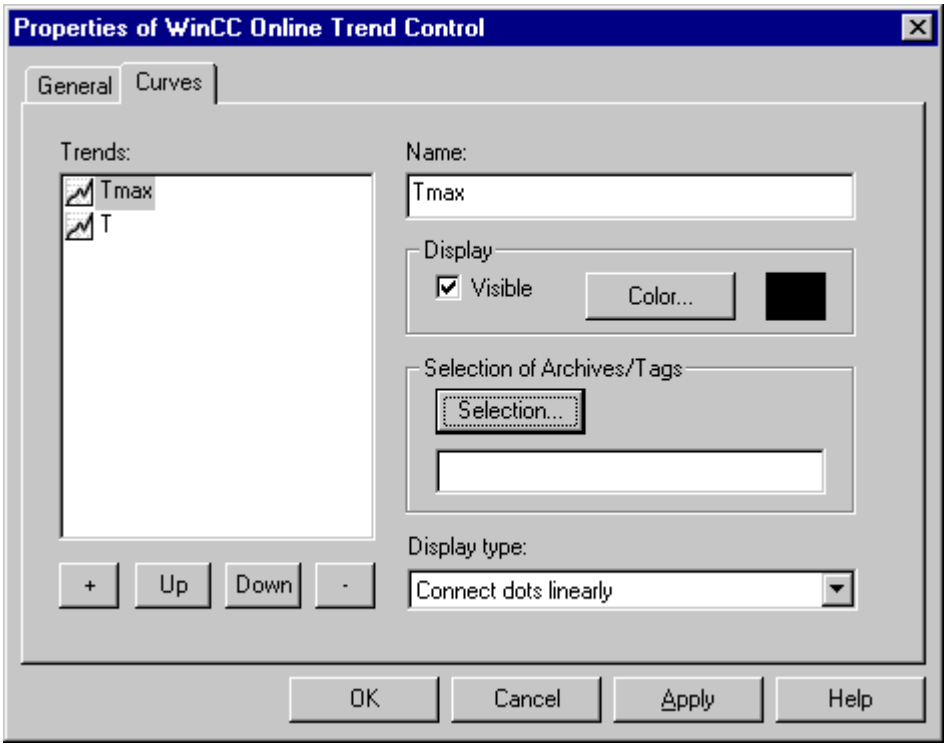
- Info 4: In the picture window, a picture of Server1 is displayed. The connection of this picture window on Server1 is implemented via the server prefix property of the picture window.
- Info 5: Tag Logging ACX (see chapter Configuration of the Tag Logging ActiveX Controls in the Multi-Client Project)
- Info 6: Alarm Logging ACX (see chapter Configuration of the Alarm Logging ActiveX Control in the Multi-Client Project)

Note:

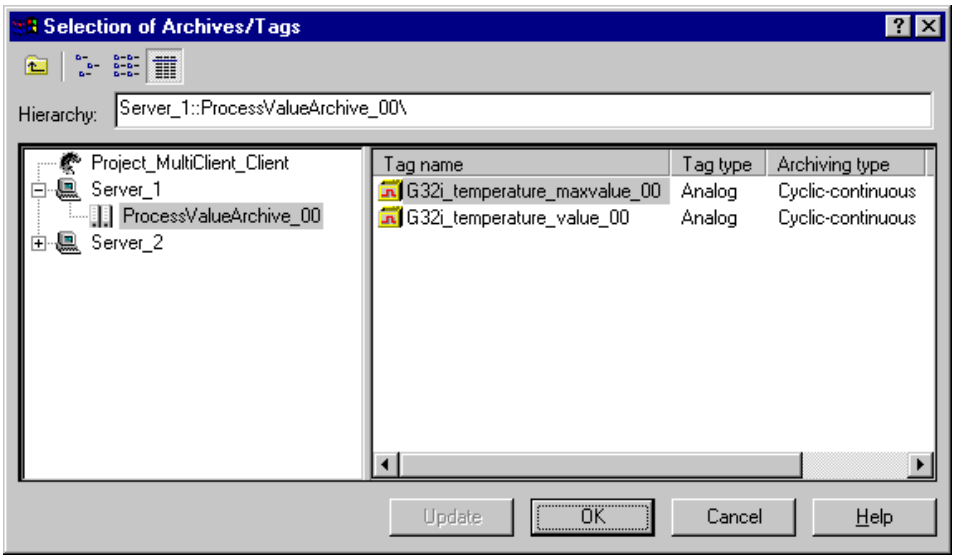
In hierarchical picture window techniques (picture in picture in picture), the server prefix is always passed on to the next subordinate picture. If a picture of a server is used, the tag connections and text library accesses in the text list object also refer to this server. The picture name property of the picture window can be made dynamic in the form of Server1::Picture.pdl. The breakdown into picture name or server prefix is then performed automatically. If the picture contains faceplates, the tag name property is automatically expanded by the server prefix. Therefore, the implementation of faceplates must not be changed for the multi-client.

1.1.2.2 Configuration of the Tag Logging ActiveX Controls in the Multi-Client Project

In order to display or edit data from one or multiple Tag Logging servers in a multi-client project, the new *WinCC Online Trend Control* and *WinCC Online Table Control* must be used.
Both can display data from multiple servers or server projects.
Via the *Graphics Designer*, a picture is configured in a multi-client project. In this picture, a *WinCC Online Trend Control* or a *WinCC Online Table Control* is placed. Via the Control's configuration dialog, a connection of the curve (Trend Control) or column (Table Control) can be made to a server, archive and archive tag or online tag.
In the following picture, the configuration is displayed using a sample of the *WinCC Online Trend Control*.

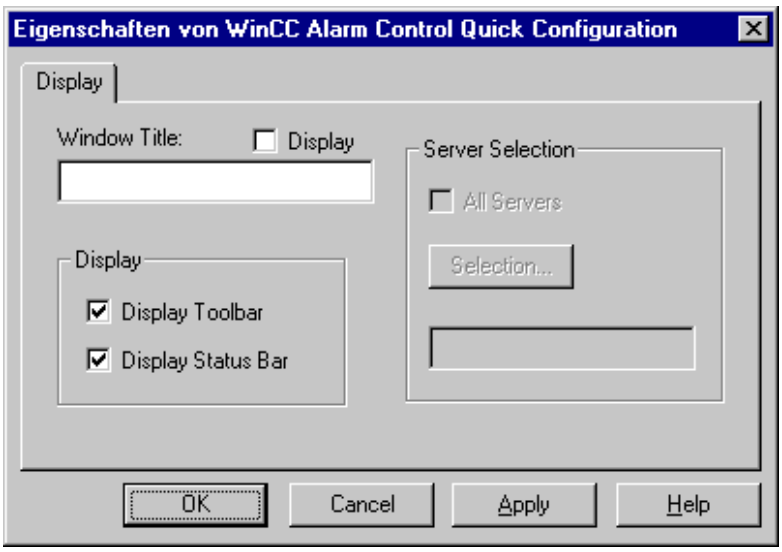


In the *Archive/Tag Selection* dialog, either a text containing a reference to a server/archive/archive tag can be entered or a dialog be opened via the selection button that enables a selection of a server/archive/archive tag from the server information imported via packages.



1.1.2.3 Configuration of the Alarm Logging ActiveX Control in the Multi-Client Project

To use *Alarm Logging* in a multi-client project, the new *WinCC Alarm Control* must be used. It provides the option of displaying messages from multiple servers.
This is configured as follows: Via the *Graphics Designer*, a picture is configured in a multi-client project. In this picture, a *WinCC Alarm Control* is placed. Via the Control's properties dialog, a connection of the *WinCC Alarm Control* to one or multiple servers can be made.



There is also the option to always connect to all servers (that start an alarm server) - this is set via a parameter.

1.1.3 Server Data (Packages)

Packages are used to provide the configurator of a multi-client with data references of one or multiple WinCC servers, and to use these data references in the multi-client project. Additionally, the packages are used to configure the name service (NS).
The packages contain the names of the objects from the server projects sorted by object type (tag names, archive names, picture names, etc.). The storage of the names is server-specific, but is still carried out using a uniform format. These files are exported from the server projects. The packages can then be imported into the multi-client project. The user is responsible for updating the data.

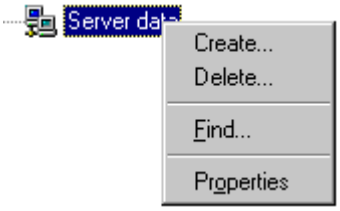
To permit access to the data of a WinCC server while offline, a new Explorer DLL is created, which provides the following functionality:

- Export of server-relevant data from a server.
- Import of server-relevant data to a multi-client project.
- Makes available data in a selection dialog.
- Configures the name service.

1.1.3.1 Packages in the WinCC Server Project

In a WinCC server project, the following menu entries are made available:

- Generate Server Data
- Delete Server Data
- Server Data Properties



Generate Server Data

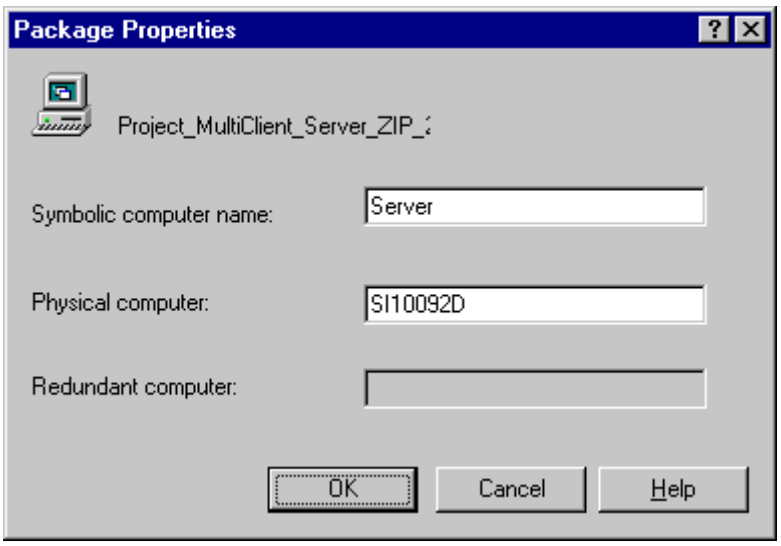
If the *Server Data* → *Generate* menus are selected, then a new CompoundDocument with the file name Server_*physical computer name*.pck is created in the Packages sub-folder of the project folder.
Example: If the computer name of the server is SI10092D, then the package will be named Server_SI10092D.pck.
If a package has been exported previously in this project, i.e. the file Server_*physical computer name*.pck exists already, then this package will be overwritten.

Delete Server Data

If a package is deleted, it is deleted from the *project folder*\Package.

Server Data Properties

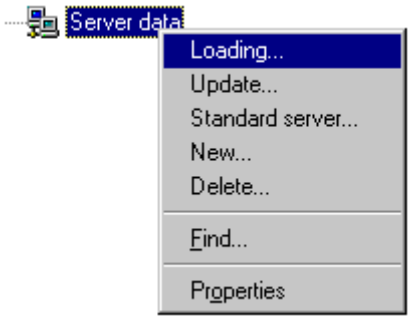
In the properties of the packages on a WinCC server or WinCC client, the name services for the multi-client is preconfigured.



1.1.3.2 Packages in the WinCC Multi-Client

On the multi-client, the following menu entries are made available:

- Import Packages
- Export Packages (not enabled)
- Delete Packages
- Package Properties



Load Server Data

The import is realized via *Server Data* → *Load...* and a File Open dialog, from which the user can select a package. This package is then copied to the project folder under \project folder\Package.
Following that, the name service for this multi-client project is configured.
If no entry with this symbolic computer name exists yet, then this package is recorded new and displayed under the Packages entry with the following information:

- *Symbolic Computer Name* on which the server project is running
- Name of the package (Server_*symbolic computer name*.pck)
- Date of creation

If an entry with the symbolic computer name of the package is already known to the name service, then the following options will be made available during the import of such a package: cancel the import, rename the symbolic computer name or overwrite the existing symbolic computer name in the name service with the new data.

Update Server Data

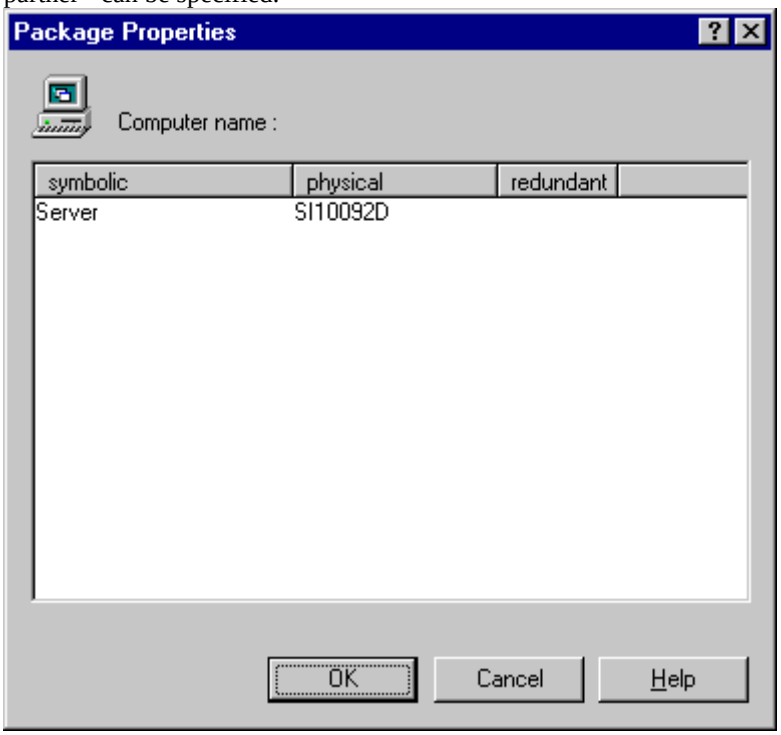
Via this menu entry, all imported packages are updated.

Delete Packages

Via this menu entry, all imported packages are deleted.

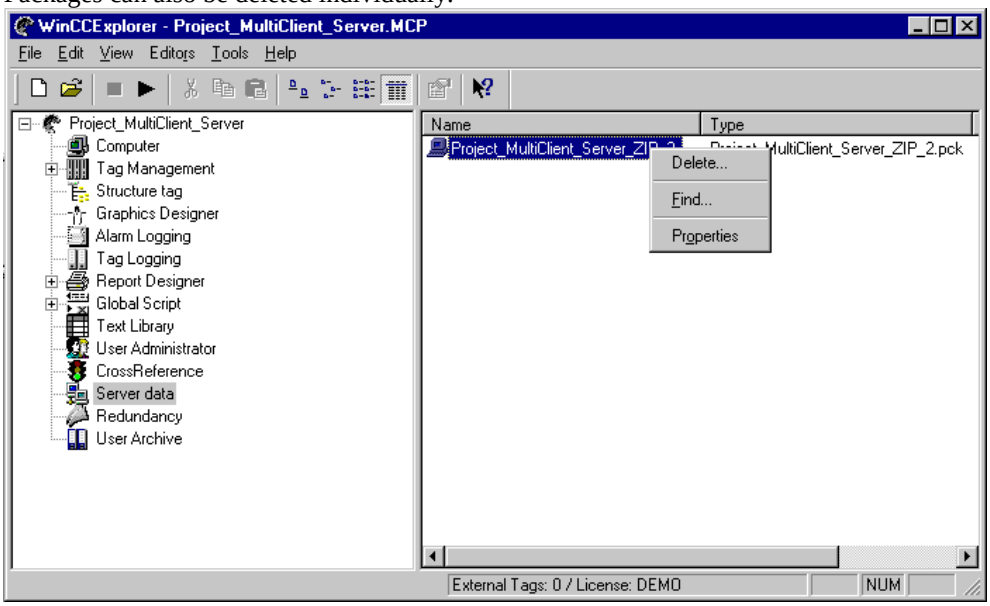
Package Properties

Via this menu entry, the preferred server - or in a redundant system, also its redundant partner - can be specified.



Deleting an individual Package

Packages can also be deleted individually.



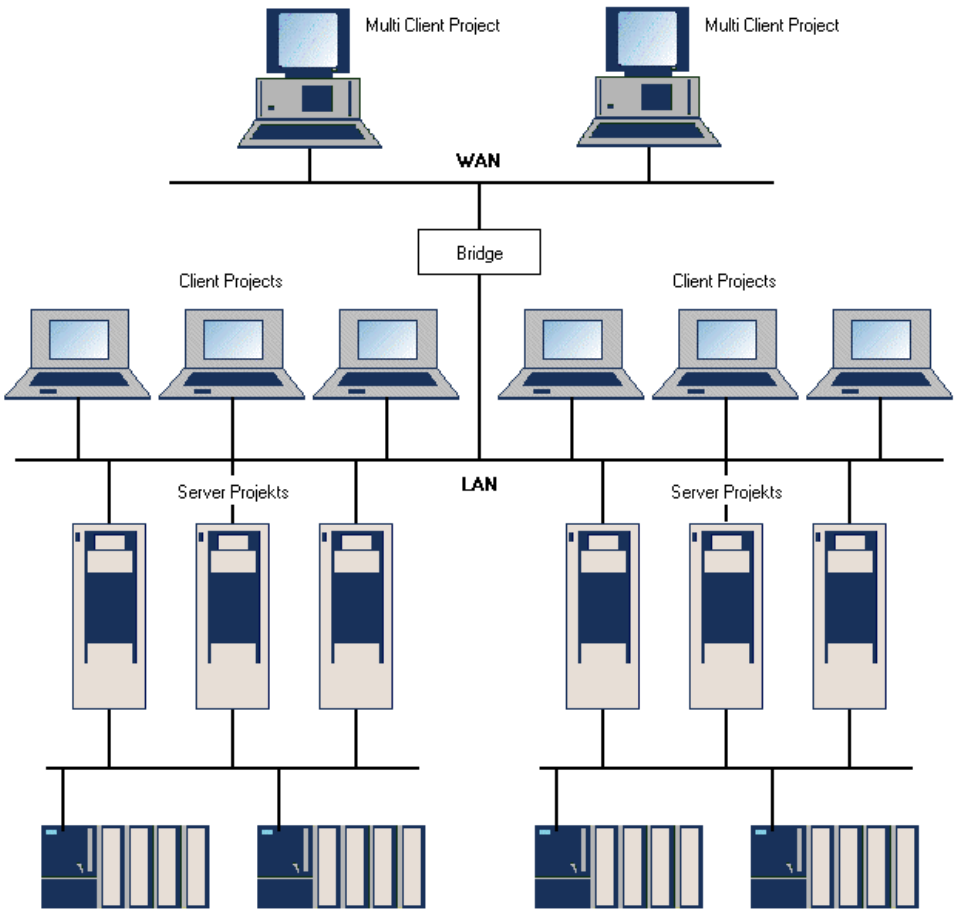
Preferred Server

In the context of the project, the multi-client does not possess its own archive or message server. If the RT APIs (MSRTCLI.DLL, PDERTCLI.DLL) of the servers are called in the context of a multi-client, then the server specified as the preferred server is addressed.In this case, each server type has its own preferred server. The configuration of the preferred servers is project-specific and only applies to one multi-client project. Generally, the following applies: If no prefix is used to specify a server, then the preferred server is used.

1.1.4 Distributed Servers

Distribution using 3 Levels

A distribution with Win CC V5.0 could look as follows:



At the bottom hierarchy level, process computers provide process data to their clients via server projects. In the server projects, the archive servers, message servers and process data servers can be configured in any form. The clients at the second hierarchy level correspond to the clients of the client/server structure in WinCC V4.0. These clients are assigned fixed to a server project and only have access to the data of this server project. At the third hierarchy level, multi-clients enable the view to multiple server projects. This includes access to pictures in the server project and to data of the server project.

2 Multi-Client

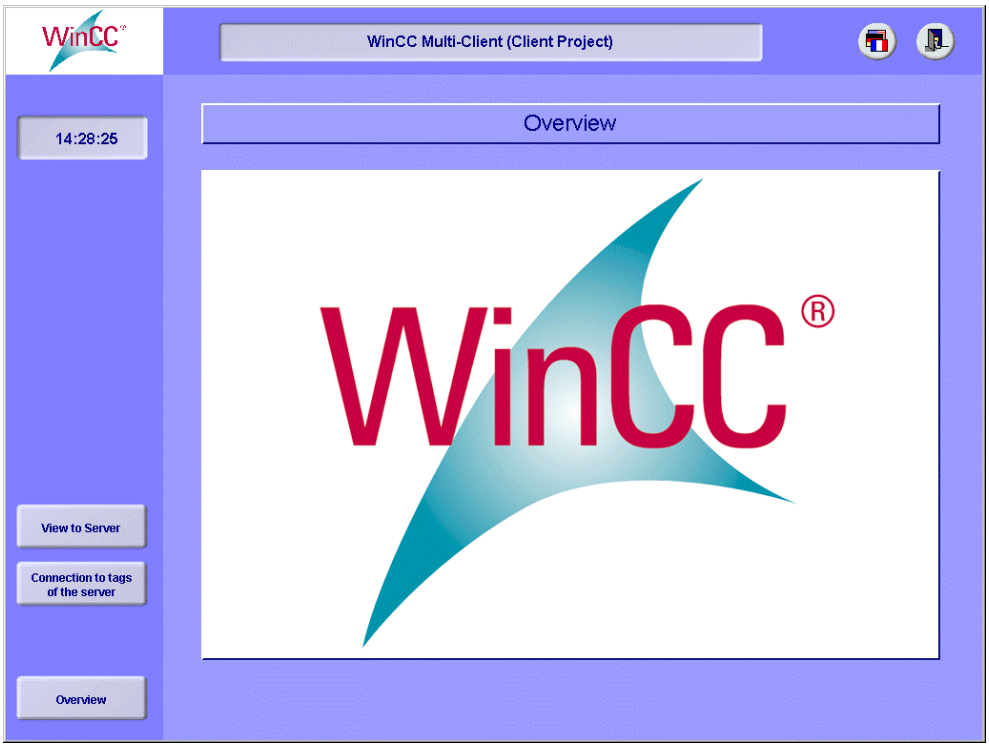
The projects created in this chapter can also be copied directly from the online document to your hard drive. By default, they will be copied to the folder *C:\Communication_Manual*. You have the option to copy the following components to the hard drive:

 Project_MultiClient_Server

The WinCC server project we will create.

 Project_MultiClient_Client

The WinCC client project we will create.
In this section, samples pertaining to the Multi-Client are presented.
The samples pertaining to this topic are configured in the WinCC projects Project_MultiClient_Server and Project_MultiClient_Client.



2.1 Application of the Multi-Client

A multi-client is a WinCC project, from which the data of multiple servers can be accessed. The multi-client has its own project, which is independent from the server. The configuration of the server takes place on the server computer, the configuration of the multi-client on the multi-client computer.
A server can be accessed simultaneously by clients as well as multi-clients. The maximum number of stations, clients and multi-clients that can access a server is limited to 16.
In runtime, the multi-client can access up to 6 servers simultaneously. For example, in a picture of a multi-client, the data from 6 different servers can be visualized. After a picture change, the multi-client can access 6 different servers. In addition, a functional or technological distribution of a project onto multiple servers can be configured.

Server

The server implements the connection to the process, the data storage and the processing of the process data. All project data on the server such as pictures, tags and archives are made available to the clients. Instead of one server, a redundant server pair can also be employed.

Multi-Client

The multi-client can access the data from up to 6 servers. The process can be controlled and monitored from a multi-client.

2.2 Server Data (Packages)

Packages are used to provide the configurator of a multi-client with data references of one or multiple WinCC servers, and to use these data references in the multi-client project. Additionally, the packages are used to configure the name service (NS).
The packages contain the names of the objects from the server projects sorted by object type (tag names, archive names, picture names, etc.). The storage of the names is server-specific, but is still carried out using a uniform format.
These files are exported from the server projects. The packages can then be imported into the multi-client project. The user is responsible for updating the data.

2.2.1 Packages in the WinCC Server Project

In a WinCC server project, the following menu entries are made available:

- Generate Server Data
- Delete Server Data
- Server Data Properties

Generate Server Data

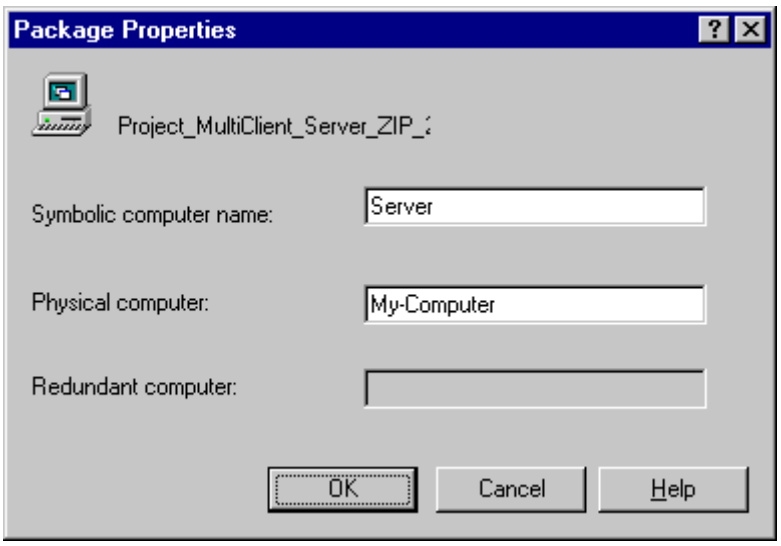
If the *Server Data* → *Generate* menus are selected, then a new CompoundDocument with the file name Server_*physical computer name*.pck is created in the Packages sub-folder of the project folder.
Example: If the computer name of the server is SI10092D, then the package will be named Server_SI10092D.pck.
If a package has been exported previously in this project, i.e. the file Server_*physical computer name*.pck exists already, then this package will be overwritten.

Delete Server Data

If a package is deleted, it is deleted from the \project folder\Package.

Server Data Properties

Via the properties of the packages on a WinCC server, the name service for the multi-client is preconfigured.



2.2.2 Packages in the WinCC Multi-Client Project

In a WinCC multi-client project, the following menu entries are made available:

- Load Server Data
- Update Server Data
- Delete Server Data
- Server Data Properties

Load Server Data

The import is realized via *Server Data* → *Load...* and a File Open dialog, from which the user can select a package. This package is then copied to the project folder under \project folder\Package.
Following that, the name service for this multi-client project is configured.
If no entry with this symbolic computer name exists yet, then this package is recorded new and displayed under the Packages entry with the following information:

- *Symbolic Computer Name* on which the server project is running
- Name of the package (Server_*symbolic computer name*.pck)
- Date of creation

If an entry with the symbolic computer name of the package is already known to the name service, then the following options will be made available during the import of such a package: cancel the import, rename the symbolic computer name or overwrite the existing symbolic computer name in the name service with the new data.

Update Server Data

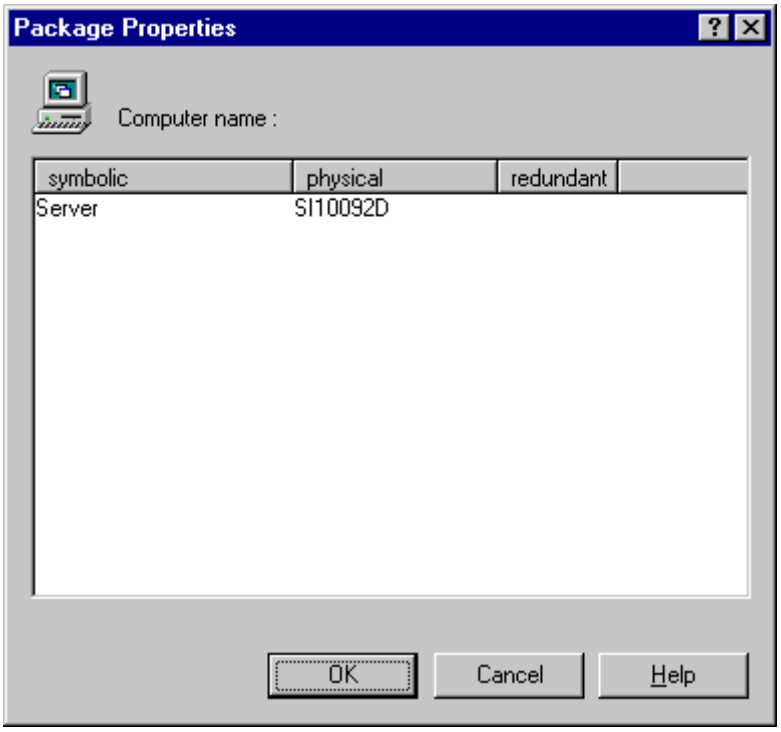
Via this menu entry, all imported packages are updated.

Delete Server Data

Via this menu entry, all imported packages are deleted.

Server Data Properties

Via this menu entry, the preferred server - or in a redundant system, also its redundant partner - can be specified.



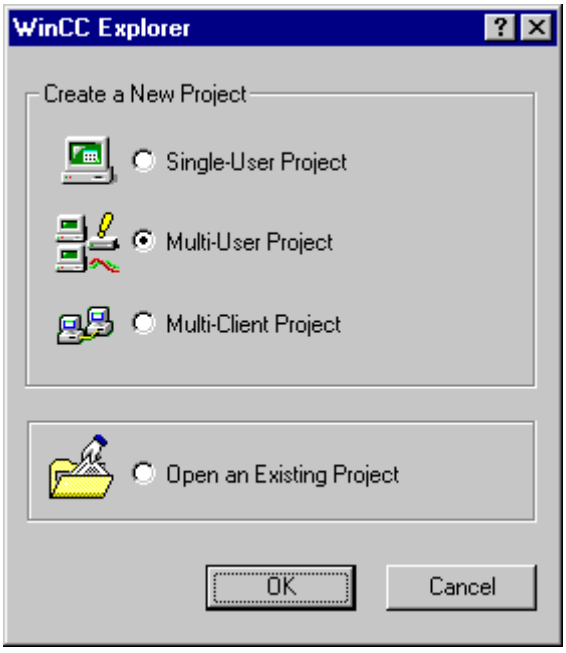
Note:

In order for the sample projects to run without problems, the packages in the server project must first be generated on both server computers. Following that, the symbolic computer name of the first server project is renamed to *Server_1*, or of the second server project to *Server_2*, in the properties dialog of the packages. Only then will the packages of the servers be loaded in the multi-client project.

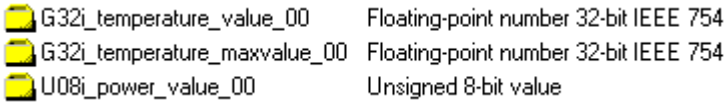
2.3 Creation of the Project_MultiClient_Server Project

The following describes in detail the steps necessary to create the multi-client project *Project_MultiClient_Server*.
The project is based on the simulation of an oven temperature control, which is then run on two server computers. Configurations are made in the Graphics Designer, Tag Logging, Alarm Logging and Global Script editors.

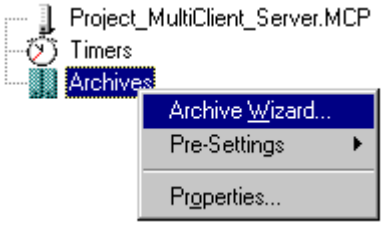
Creating a Server Project

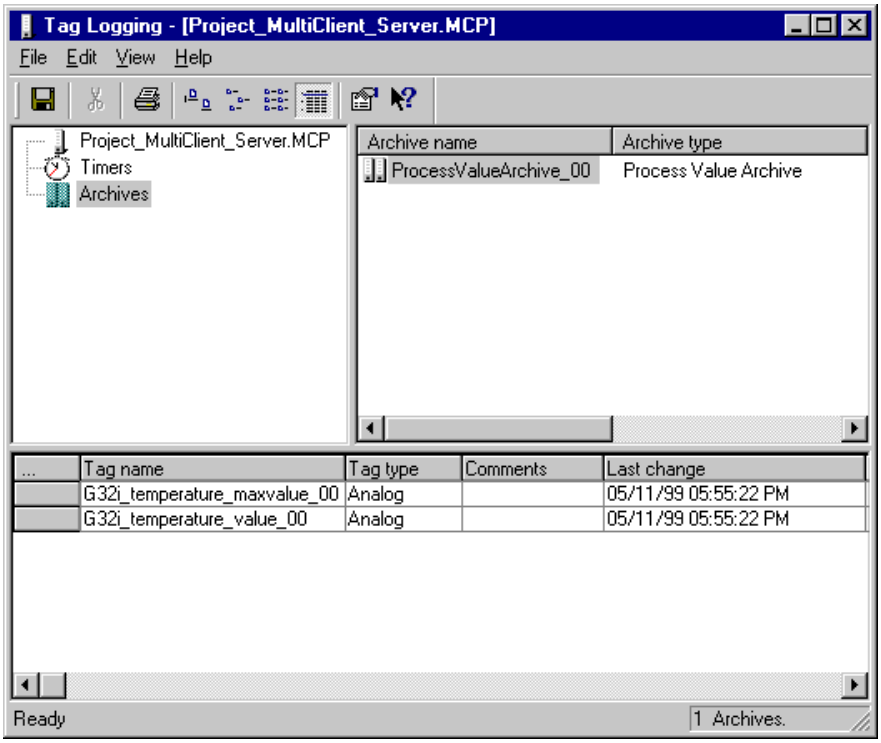
Step	Procedure: Creating a Server Project
1	Creation of a new WinCC project. The WinCC Explorer is started via *Start* → *Simatic* → *WinCC* → *Windows Control Center*.
2	The *WinCC Explorer* will be displayed. Via the menus *File* → *New*, the dialog box for specifying the properties of a new WinCC project will be opened. This sample project is created as a *Multi-User Project*. Exit the dialog box by clicking on *OK*. 

Creating the Tags

Step	Procedure: Creating the Tags
1	In this sample, three internal tags with the following names are created: The tag *G32i_temperature_value_00* corresponds to the actual value of the temperature, the tag *G32i_temperature_maxvalue_00* to the setpoint value of the temperature and the tag *U08i_power_value_00* to the heating capacity. 

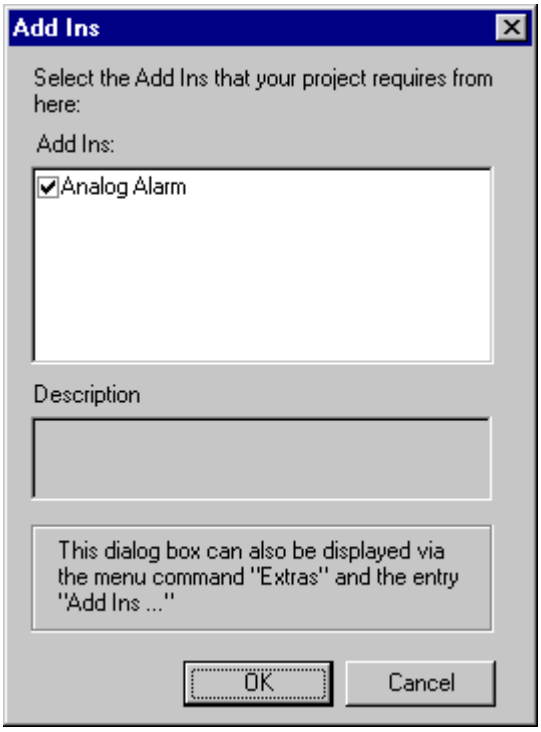
Creating a Process Value Archive

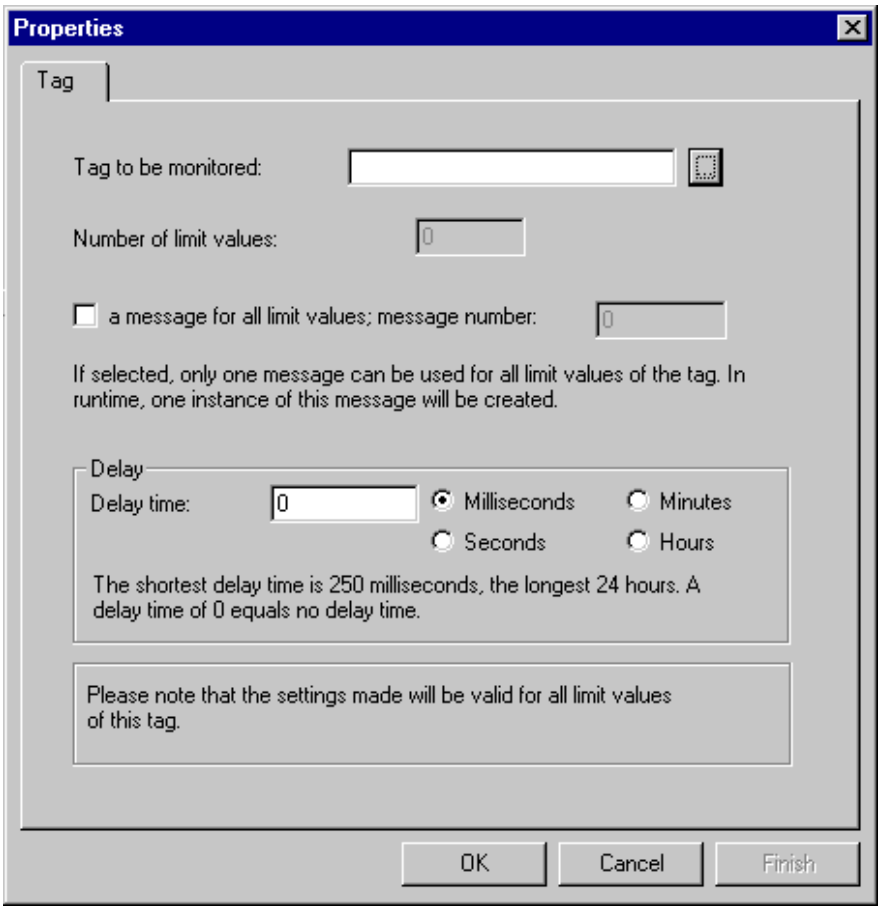
Step	Procedure: Creating a Process Value Archive
1	Open the *Tag Logging* editor.
2	Creation of a process value archive. For this purpose, the Archive Wizard is started via a R on *Archives*. 

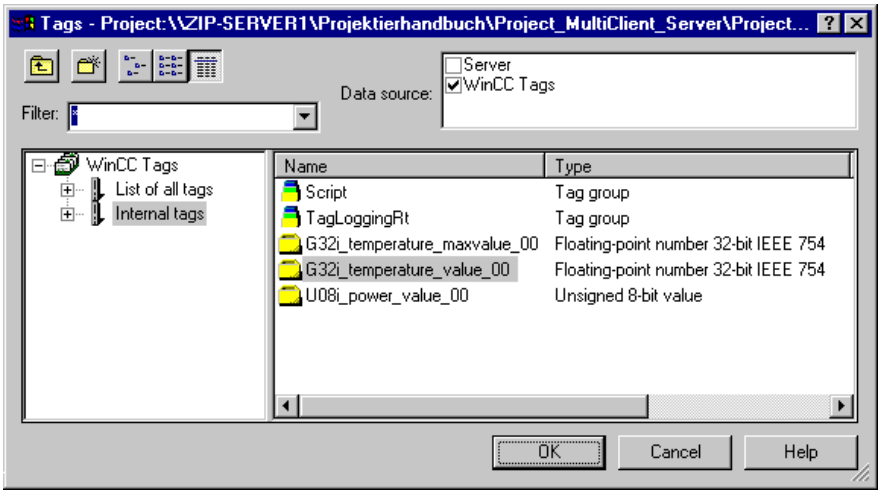
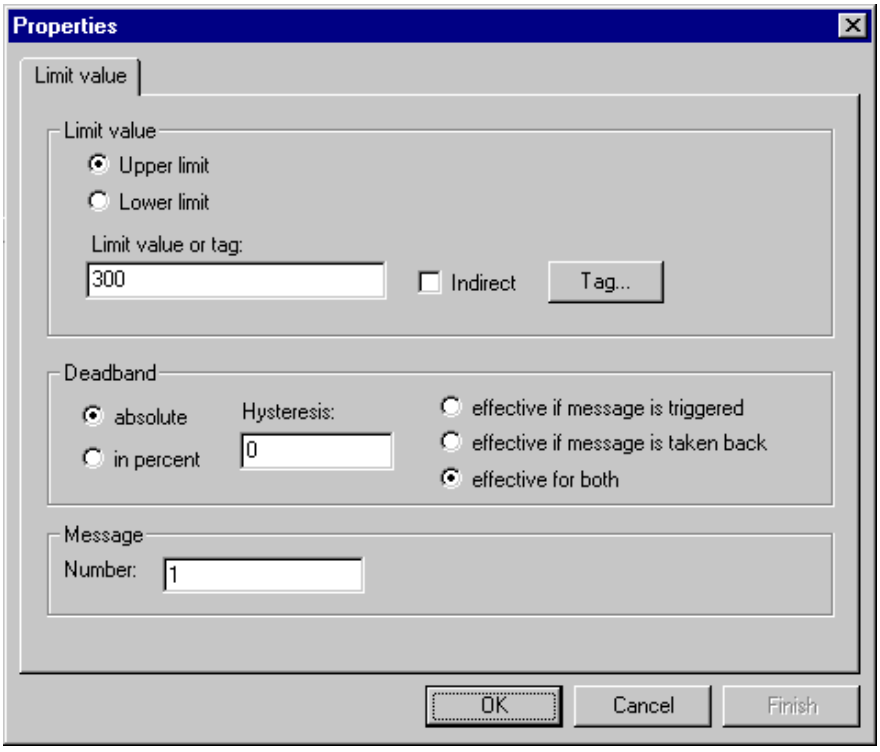
Step	Procedure: Creating a Process Value Archive
3	In this sample, the archive has been named *ProcessValueArchive_00*. The tags *G32i_temperature_value_00* and *G32i_temperature_maxvalue_00* are specified as the archive tags. 
4	Save and then exit the *Tag Logging* editor.

Configuring Alarm Logging

Step	Procedure: Configuring Alarm Logging
1	Open the *Alarm Logging* editor.
2	Creation of single messages. In the lower window of the *Alarm Logging* editor, the already configured messages are displayed. Via a R, a new line can be added. In this sample, two different messages are required. The error type, message text and point of error must be changed correspondingly. ... 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

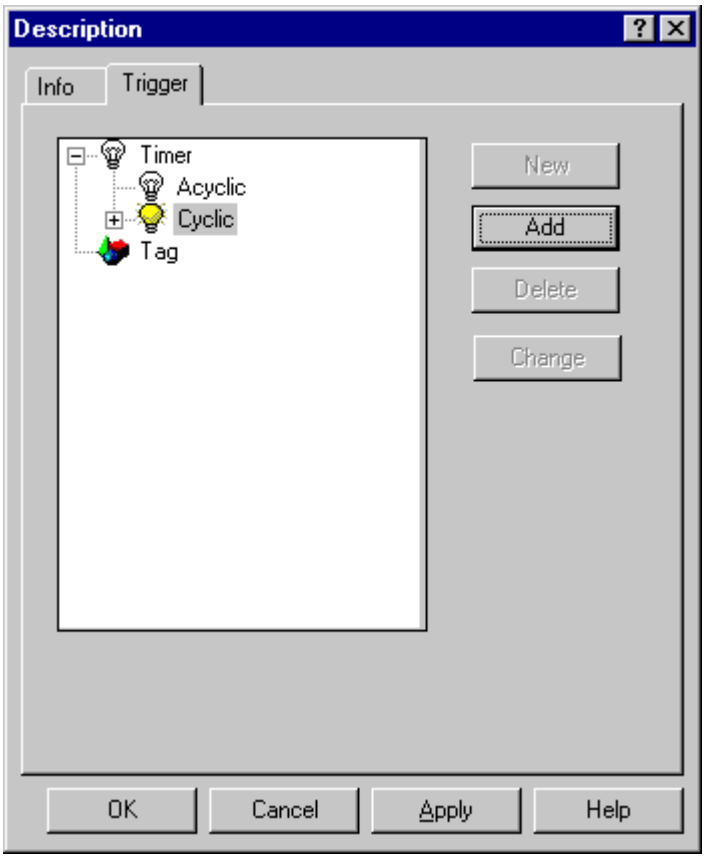
Step	Procedure: Configuring Alarm Logging
3	Configuration of the limit value monitoring. If the *Limit Value Monitoring* (Analog Alarm) entry is not present, it must be loaded first. This is done via the *Options* → *Add Ins* menus in *Alarm Logging*. In the dialog displayed, the check-box for the *Limit Value Monitoring (Analog Alarm)* must be selected. Close the dialog box by clicking on *OK*. 

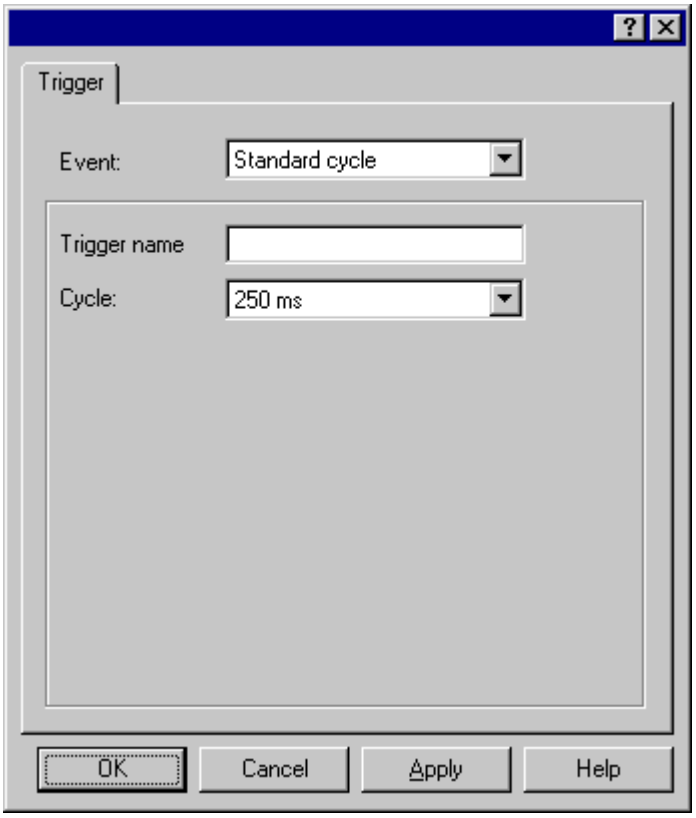
Step	Procedure: Configuring Alarm Logging
4	Via a R on the *Limit Value Monitoring entry and then selecting* → *New...* , the *Properties* dialog of the tag is accessed. In this dialog, a new tag for the limit value monitoring can be set. 
5	Via a on the button displayed below, the *Select Tag* dialog is accessed. 

Step	Procedure: Configuring Alarm Logging
6	In the left window, the entry *Internal Tags* is selected. The right window will then list the corresponding tags. Select the desired tag. In the sample, this is the *G32i_temperature_value_00* tag. The dialog is closed with the *OK* button. 
7	The *Properties* dialog of the tag is closed with *OK* as well. The right window of *Alarm Logging* will then display the icon of the new tag to be monitored. Via a R on *G32i_temperature_value_00* → *New*, the *Properties* dialog of the limit value is accessed. In this dialog, a new limit value can be assigned to the tag. In this sample, the *Upper Limit* is set to *300* and the message number to *1*. The dialog is closed with the *OK* button. 

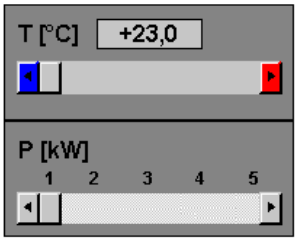
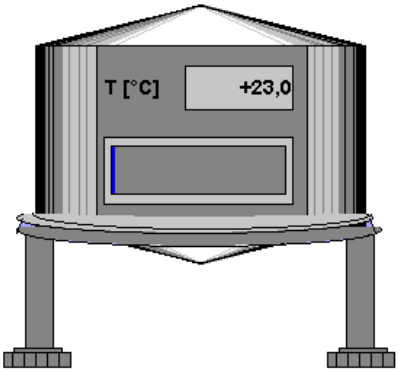
Step	Procedure: Configuring Alarm Logging
8	Following the previously described step, a second limit value is assigned to the tag. The *Upper Limit* is set to *700* and the message number to *2*.
9	Save and then exit the *Alarm Logging* editor.

Creating a Global Action

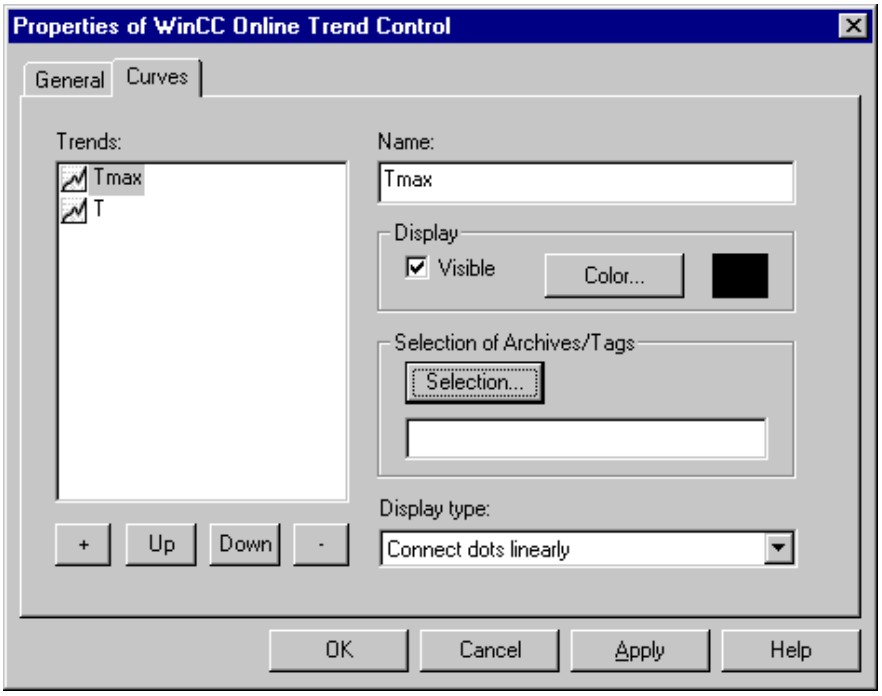
Step	Procedure: Creating a Global Action
1	Open the *Global Script* editor.
2	Creation of a new global action. This is carried out via the *File* → *New Action* menus in the Global Script editor.
3	In this sample, a C-Action has been programmed that simulates an e-function as a trend. The difference *dDelta* between the setpoint temperature *dTemp2* and the actual temperature *dTemp1* is computed. If this difference is positive, the trend increases. If it is negative, the trend drops. The heating capacity *nPower* defines, how fast the temperature reaches the setpoint value.
4	Via *Edit* → *Compile*, the C-Action is compiled.
5	Via *Edit* → *Info*, the *Description* dialog is opened. In the *Trigger tab, a Cyclic Timer* is selected in this sample. Via the *Add* button, the dialog for changing the trigger is displayed. 

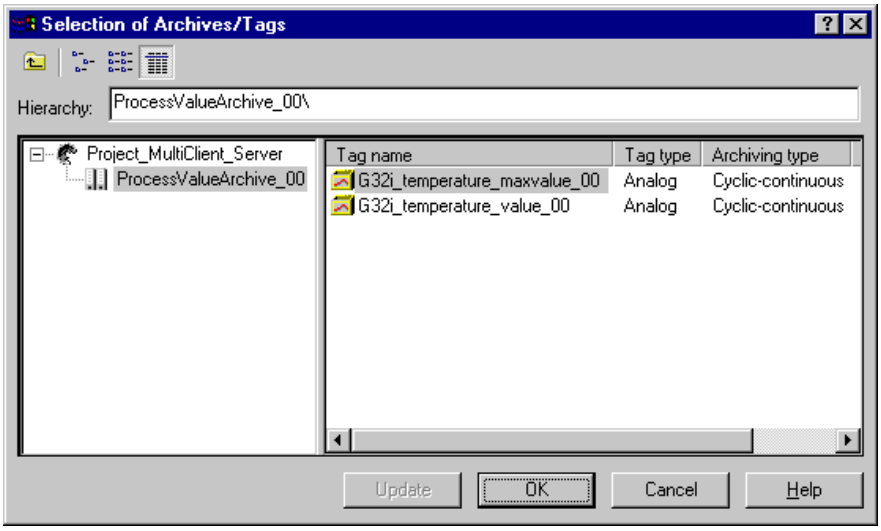
Step	Procedure: Creating a Global Action
6	The cycle time is set to *250 ms*. Both dialogs are closed with *OK*. 
7	Save and then exit the *Global Script* editor.

Configuring Objects

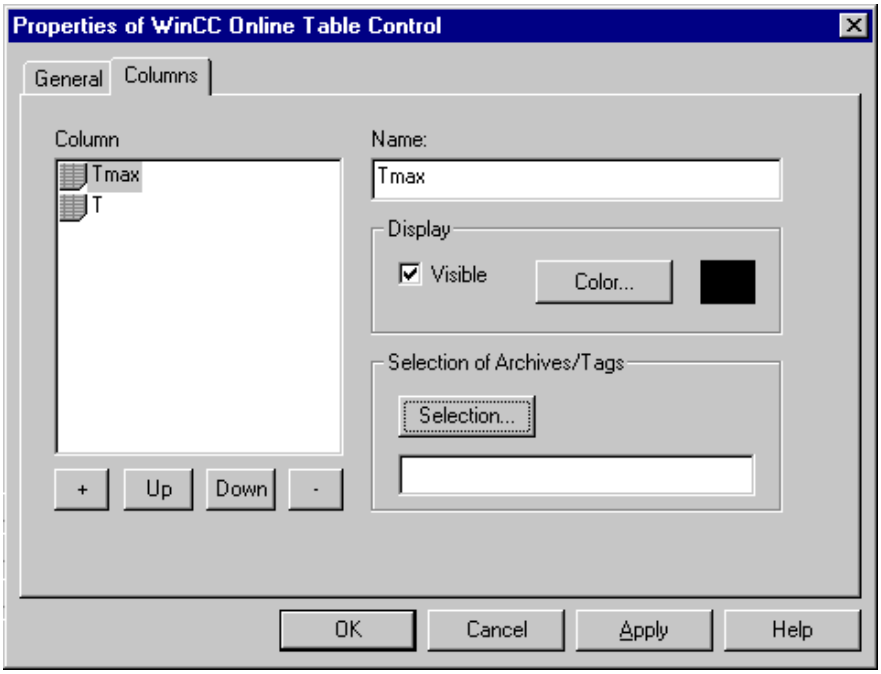
Step	Procedure: Configuring Objects
1	Create a new picture in the *Graphics Designer*. In the sample, this is the *mcs_3_chapter_01.PDL* picture. In this picture, various *Objects* are connected to process tags.
2	The simulation of the input tags is implemented via a *Windows Object* → *Slider Object* each. In this sample, these are the *Slider Object1 (G32i_temperature_maxvalue_00)* and *Slider Object2 (U08i_power_value_00)* that together with the *I/O Field1 (G32i_temperature_maxvalue_00)* represent the temperature control. In *I/O Field1*, the value of the setpoint temperature is displayed and can also be changed there. The output tag *(G32i_temperature_value_00)* is displayed in the oven. It consists of the *I/O Field2* and the *Bar1* objects. The update of these objects is set to *Upon Change*.  

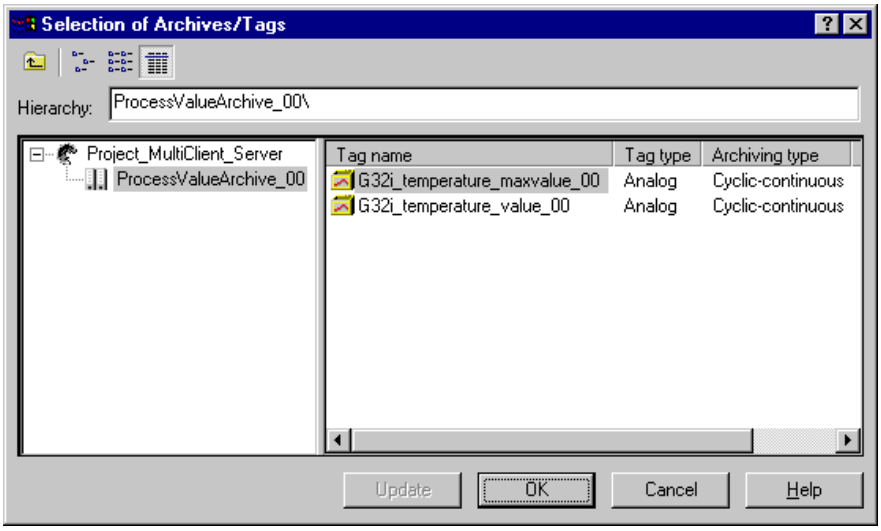
Configuring the Trend Windows

Step	Procedure: Configuring the Trend Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *mcs_3_chapter_02.PDL* picture. In this picture, two temperature values are displayed using trend windows.
2	Creation of a *Trend Control* via *Control* → *WinCC Online Trend Control*. In the sample, this is the *TlgOnlineTrend1* object. The dialog *WinCC Online Trend Control Properties* is displayed. In the *Trends* tab, a new trend is added by clicking on the + button. *Trend 1* is renamed to *Tmax* and *Trend 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

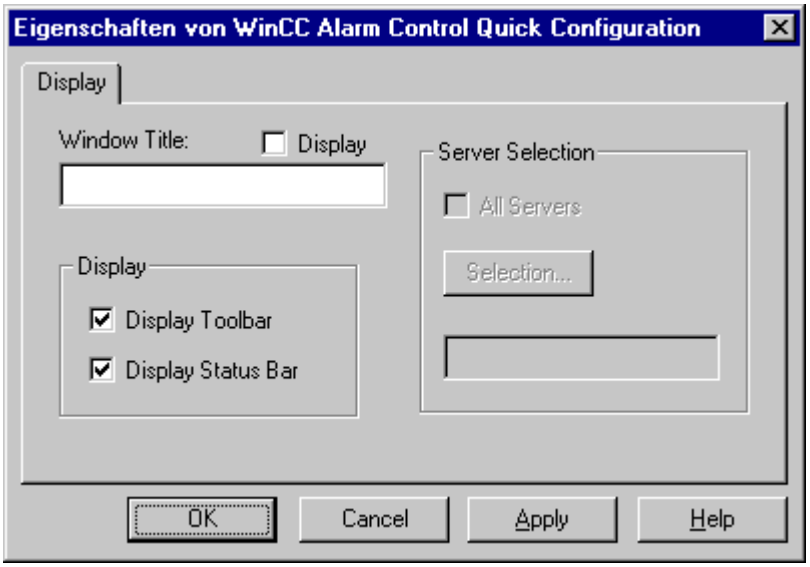
Step	Procedure: Configuring the Trend Windows
3	The *Archive/Tag Selection* dialog is displayed. This dialog allows the selection of archives/archive tags. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag. 

Configuring the Table Windows

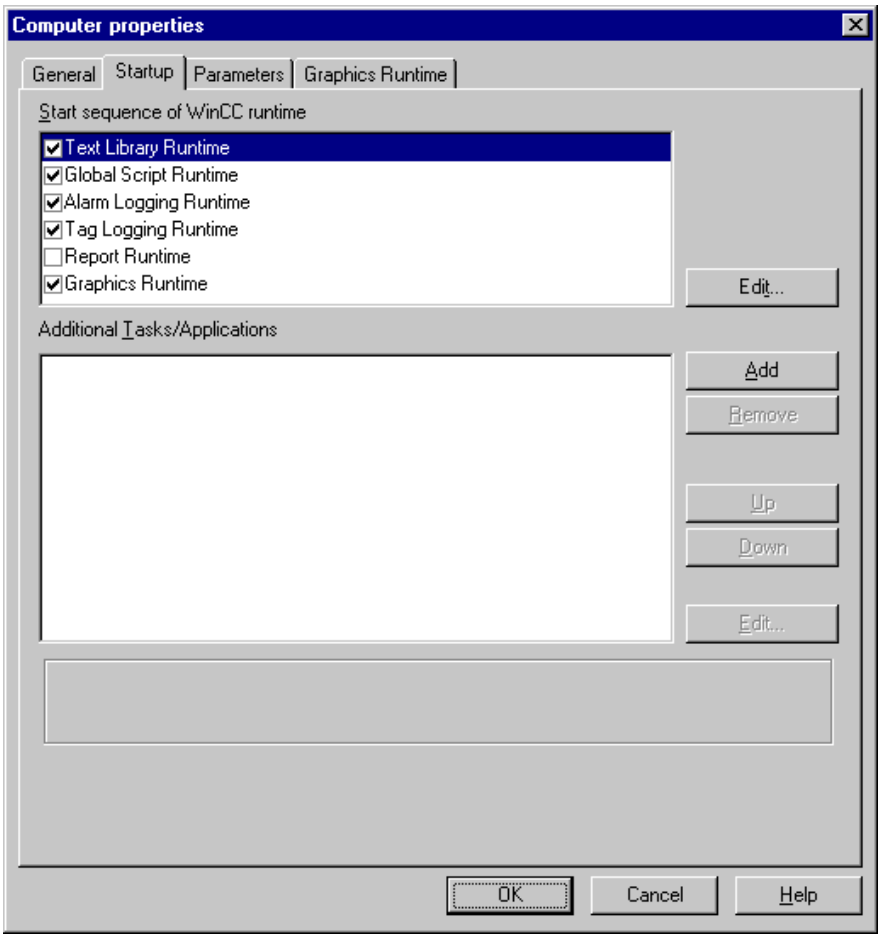
Step	Procedure: Configuring the Table Windows
1	In the same picture (*mcs_3_chapter_02.PDL*), two temperature values are displayed using table windows.
2	Creation of a *Table Control* via *Control* → *WinCC Online Table Control*. In the sample, this is the TlgOnlineTable1 object. The dialog *WinCC Online Table Control Properties* is displayed. In the *Columns* tab, a new column is added by clicking on the + button. *Column 1* is renamed to *Tmax* and *Column 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

Step	Procedure: Configuring the Table Windows
3	The *Archive/Tag Selection* dialog is displayed. This dialog allows the selection of archives/archive tags. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag. 

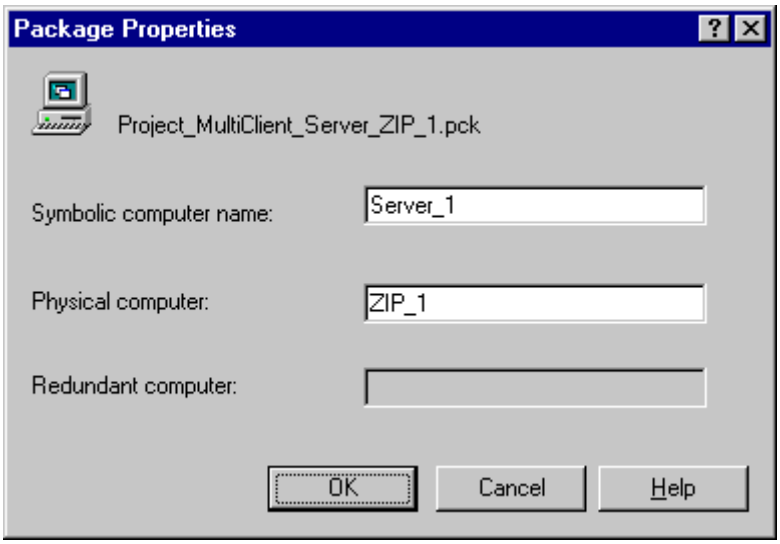
Configuring the Message Windows

Step	Procedure: Configuring the Message Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *mcs_3_chapter_03.PDL* picture. In this picture, the configured messages are output using message windows.
2	Creation of an *Alarm Control* via *Control* → *WinCC Alarm Control*. In the sample, this is the CCAlgWinCtrl1 object. The dialog *WinCC Alarm Control Properties - Quick Configuration* is displayed. This dialog is closed with *OK*. 

Setting the WinCC Runtime Startup Properties

Step	Procedure: Setting the WinCC Runtime Startup Properties
1	Via a on the *Computer* entry on the left side of the *WinCC Explorer*, the computer name will be displayed on the right.
2	*Via a* R on *Computer Name* → *Properties*, the *Computer Properties* dialog is displayed. In the *Startup* tab, the following settings are made. Close the dialog box by clicking on *OK*. 

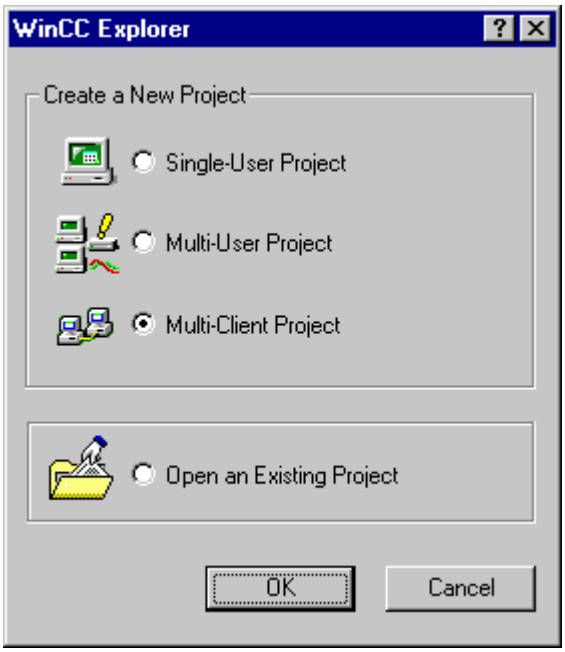
Generating the Server Data

Step	Procedure: Generating the Server Data
1	The server data is generated via a R on *Server Data* → *Generate* on the left side of the WinCC Explorer. A message stating that the server data has been generated successfully will be displayed. This dialog is acknowledged with OK. Following that, the generated package is displayed on the right side of the WinCC Explorer. 
2	*Via a* R on the newly generated package and *Properties*, the dialog *Package Properties* is displayed. In this sample, the *Symbolic Computer Name* is renamed to *Server_1* . The dialog is closed with the *OK* button. 
3	Following the previously described steps, generate the server data on the second server and rename its *Symbolic Computer Name* to *Server_2*.

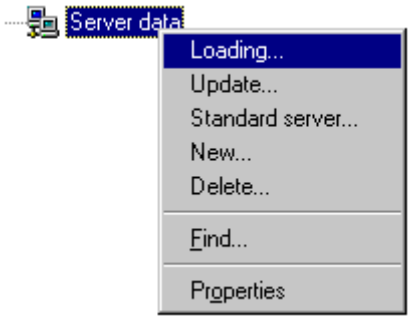
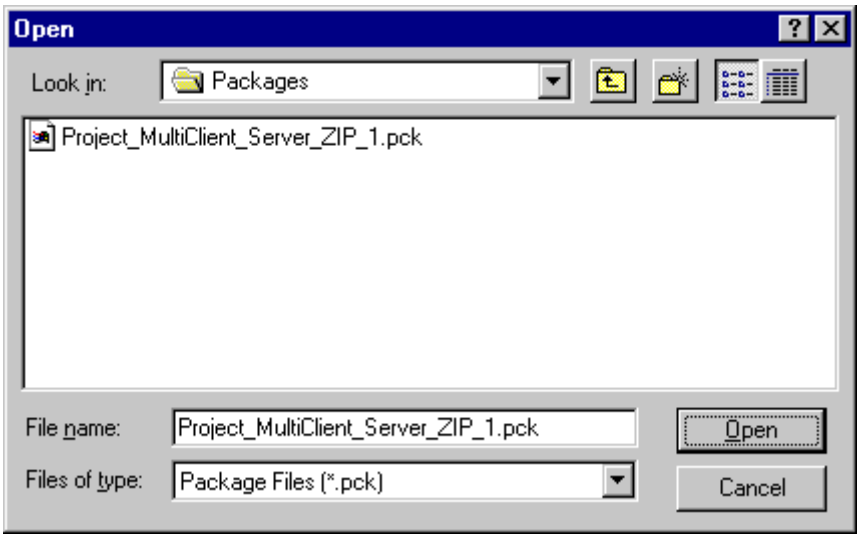
2.4 Creation of the Project_MultiClient_Client Project

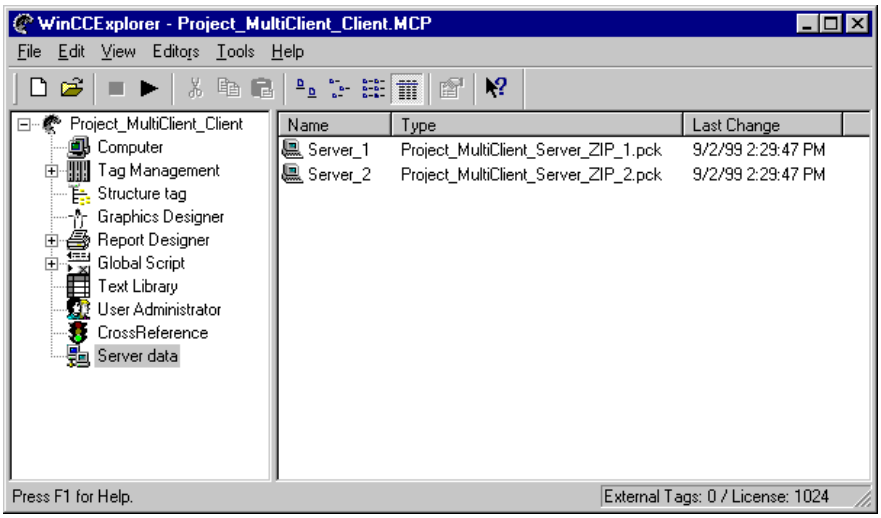
The following describes in detail the steps necessary to create the multi-client project *Project_MultiClient_Client*.
This project references the data of the two previously configured servers.
A multi-client project can only configure its own data, not the data of a server project. It can, however, reference the data on the server project (provide so-called views to servers).
The server data (package) is required to make the relevant data of one or several servers available to the multi-client project during the configuration phase.
The multi-client project possesses its own pictures, but can also display the pictures of one or multiple servers.

Creating a Multi-Client Project

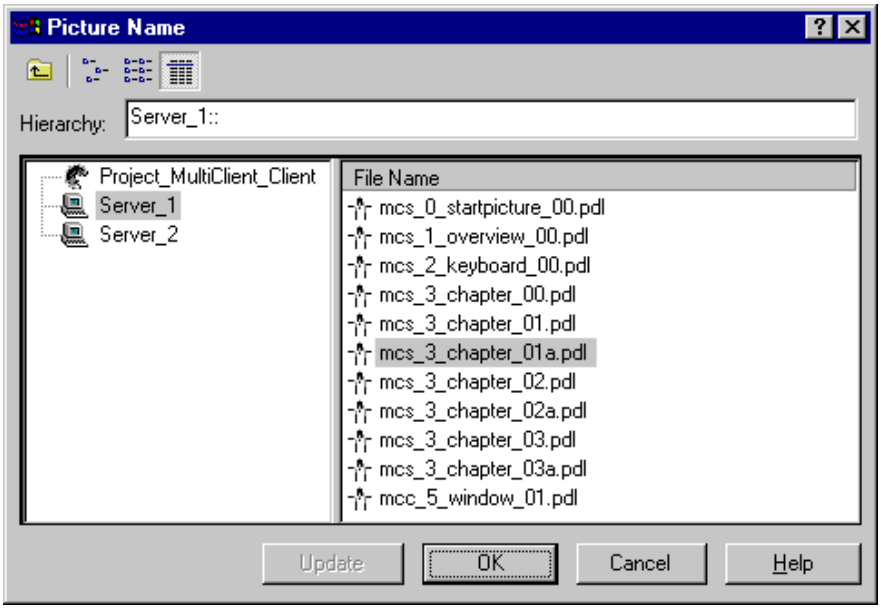
Step	Procedure: Creating a Multi-Client Project
1	Creation of a new WinCC project. The WinCC Explorer is started via *Start* → *Simatic* → *WinCC* → *Windows Control Center*.
2	The *WinCC Explorer* will be displayed. Via the menus *File* → *New*, the dialog box for specifying the properties of a new WinCC project will be opened. This sample project is created as a *Multi-Client Project*. Exit the dialog box by clicking on *OK*. 

Loading the Server Data

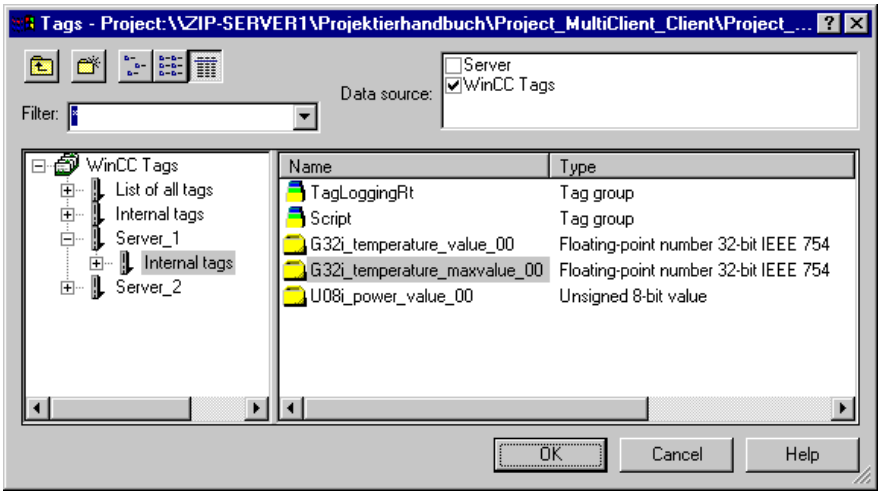
Step	Procedure: Loading the Server Data
1	*Via a* R on *Server Data* → *Load* on the *left* side of the *WinCC Explorer*, the dialog *Open* is displayed. 
2	Via the *Network Neighborhood* entry, the server computer is selected. The package file is located on the server in the folder *Project_MultiClient_Server* → *Server Name* → *Packages*. This file is selected and loaded via the *Open* button. 
3	A dialog confirming the successful loading of the server data will be displayed. This dialog is closed with *OK*. 

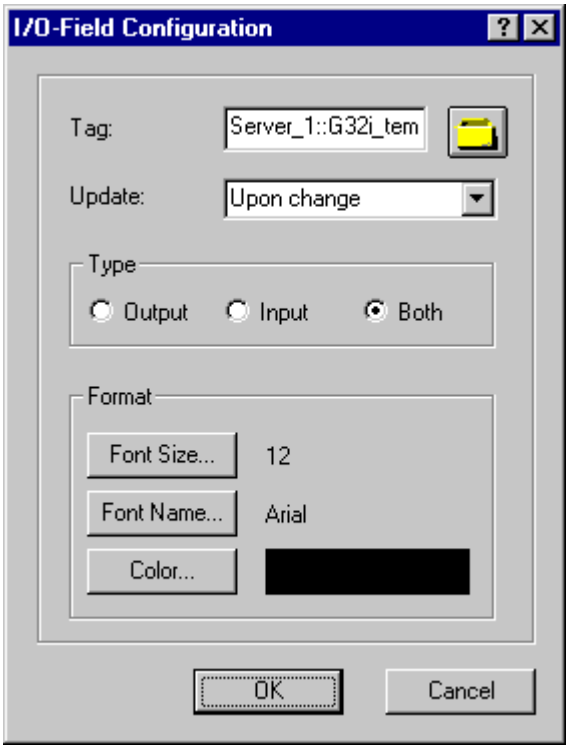
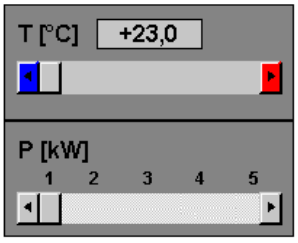
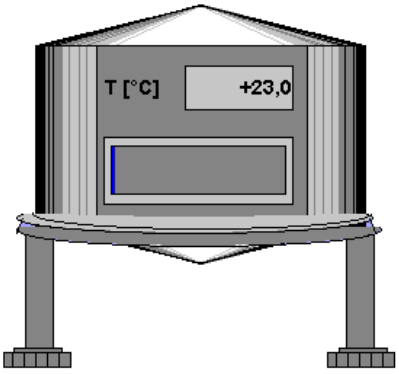
Step	Procedure: Loading the Server Data
4	Following the steps just described, the package file of the second server is loaded. Both loaded packages will be displayed in the right window of the WinCC Explorer. 

Configuring Views to Servers

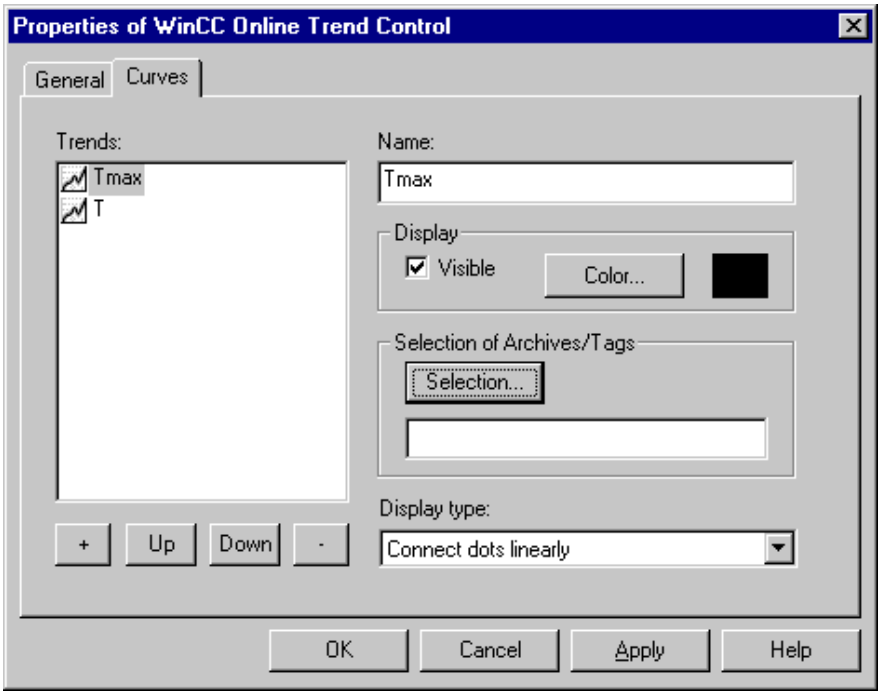
Step	Procedure: Configuring Views to Servers
1	Create a new picture in the *Graphics Designer*. In the sample, this is the *mcc_3_chapter_01.PDL* picture. In this picture, pictures configured on the servers are displayed using *Picture Windows*.
2	Configuration of a *Smart Object* → *Picture Window*. In its *Object Properties* dialog, the *Picture Name* dialog is opened via a D on *Properties* → *Miscellaneous* → *Picture Name*. In this dialog, the picture to be displayed in the *Picture Window* can be specified. To select a server picture, the desired server must first be selected in the left window. The picture files associated with this server will then be displayed in the right window. Select the desired picture. In the sample, this is the *mcs_3_chapter_01a.PDL* picture of *Server_1*. Close the dialog box by clicking on *OK*. 
3	Configuration of another *Smart Object* → *Picture Window*. In this picture window, the *mcs_3_chapter_01a.PDL* picture of *Server_2* is displayed.
4	Following steps 1 to 3, two additional pictures are configured. In these pictures, the *mcs_3_chapter_02a.PDL* and *mcs_3_chapter_03a.PDL* pictures of both servers are displayed.

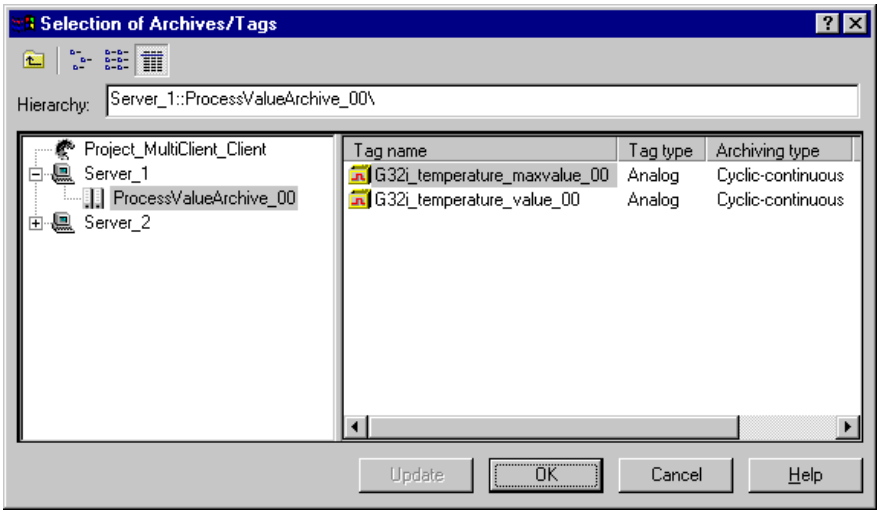
Configuring Objects

Step	Procedure: Configuring Objects
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *mcc_3_chapter_11.PDL* picture. In it, various *Objects* are connected with process tags of *Server_1*.
2	Configure a *Smart Object* → *I/O Field*. In the sample, this is the *I/O Field1* object. Its *Configuration Dialog will be displayed.* Via a on the button displayed below, the *Select Tag* dialog is accessed. 
3	In the left window, the *Internal Tags* entry of the desired server is selected. The right window will then list the corresponding tags. Select the desired tag. In the sample, this is the *G32i_temperature_maxvalue_00* tag of *Server_1*. The dialog is closed with the *OK* button. 

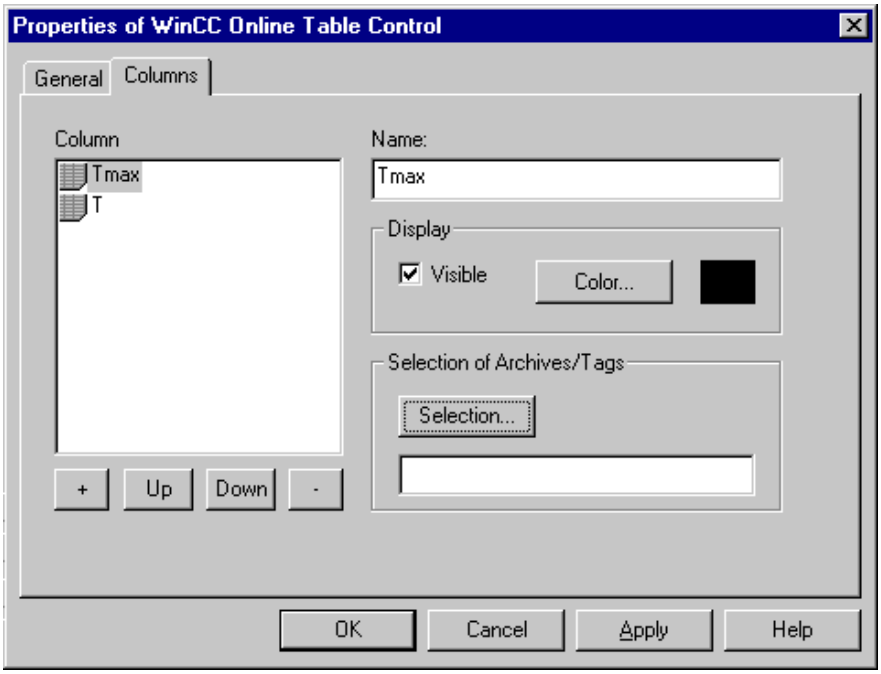
Step	Procedure: Configuring Objects
4	The *Update* is set to *Upon Change*. The *Configuration Dialog* can be exited with *OK*. 
5	Configuration of additional objects (*I/O Fields*, *Slider Objects*, *Bar Graphs*) to display the remaining tags of the servers.  

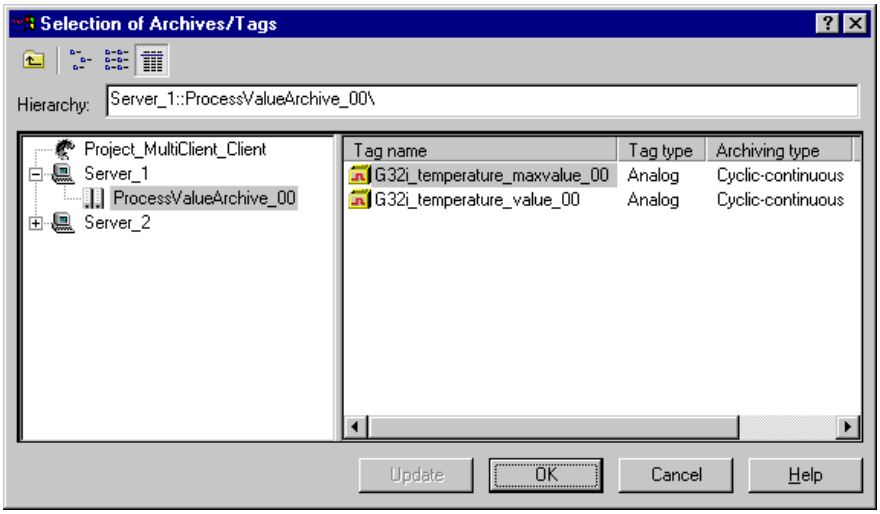
Configuring the Trend Windows

Step	Procedure: Configuring the Trend Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *mcc_3_chapter_12.PDL* picture. In this picture, the two temperature values of *Sever_1* are displayed using trend windows.
2	Creation of a *Trend Control* via *Control* → *WinCC Online Trend Control*. In the sample, this is the *TlgOnlineTrend1* object. The dialog *WinCC Online Trend Control Properties* is displayed. In the *Trends* tab, a new trend is added by clicking on the + button. *Trend 1* is renamed to *Tmax* and *Trend 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

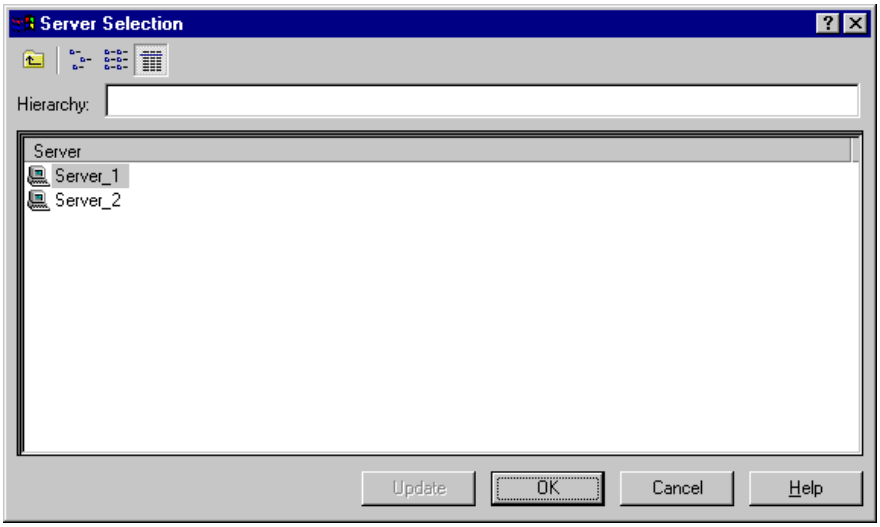
Step	Procedure: Configuring the Trend Windows
3	The *Archive/Tag Selection* dialog is displayed. From this dialog, the servers/archives/archive tags can be selected from the server data imported by the packages. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag of *Server_1*. 
4	Following the steps just described, an additional *WinCC Online Trend Control* is configured. This Control is connected to the tags of *Server_2*.

Configuring the Table Windows

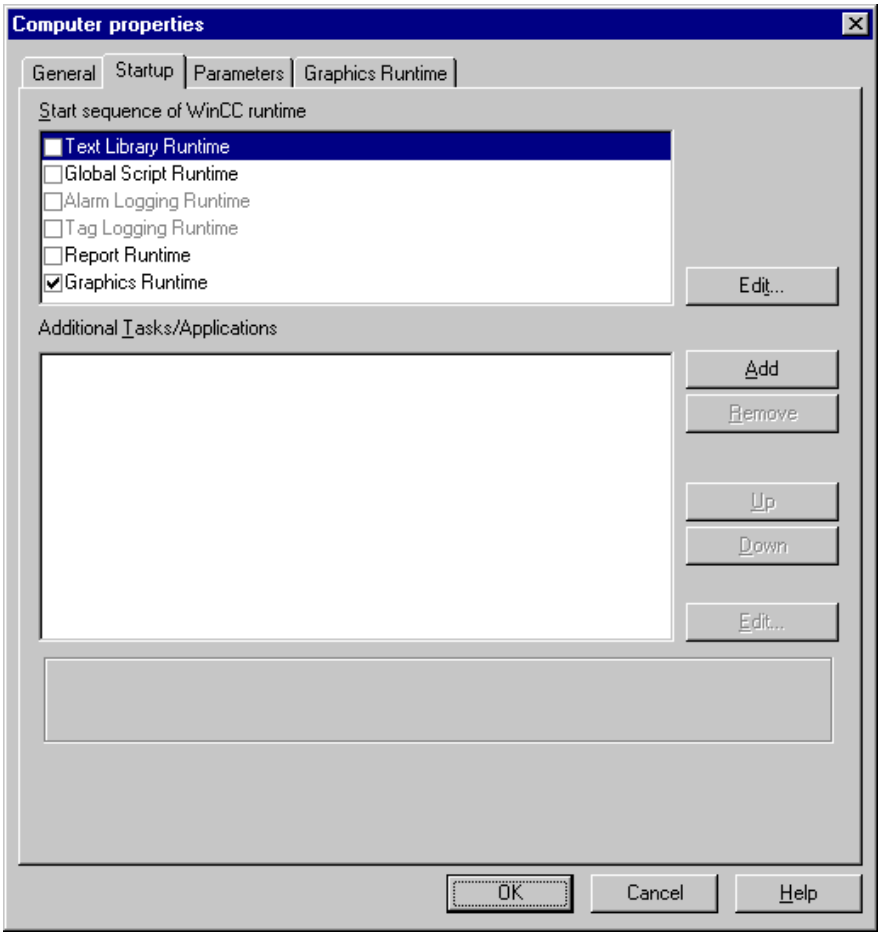
Step	Procedure: Configuring the Table Windows
1	In the same picture (*mcc_3_chapter_12.PDL*), the two temperature values of *Server_1* are displayed using table windows.
2	Creation of a *Table Control* via *Control* → *WinCC Online Table Control*. In the sample, this is the TlgOnlineTable1 object. The dialog *WinCC Online Table Control Properties* is displayed. In the *Columns* tab, a new column is added by clicking on the + button. *Column 1* is renamed to *Tmax* and *Column 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

Step	Procedure: Configuring the Table Windows
3	The *Archive/Tag Selection* dialog is displayed. From this dialog, the servers/archives/archive tags can be selected from the server data imported by the packages. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag of *Server_1*. 
4	Following the steps just described, an additional *WinCC Online Table Control* is configured. This Control is connected to the tags of *Server_2*.

Configuring the Message Windows

Step	Procedure: Configuring the Message Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *mcc_3_chapter_13.PDL* picture. In this picture, the messages configured on *Server_1* are output using message windows.
2	Creation of an *Alarm Control* via *Control* → *WinCC Alarm Control*. In the sample, this is the CCAlgWinCtrl1 object. The dialog *WinCC Alarm Control Properties - Quick Configuration* is displayed. Via the *Select* button, the *Server Selection* dialog is accessed.
3	In this sample, *Server_1* is selected and the dialog closed with *OK*. 
4	Following the steps just described, a *WinCC Alarm Control* is configured. This Control is connected to *Server_2*.

Setting the WinCC Runtime Startup Properties

Step	Procedure: Setting the WinCC Runtime Startup Properties
1	Via a on the *Computer* entry on the left side of the *WinCC Explorer*, the computer name will be displayed on the right. Through a R on *Computer Name* → *Properties*, the *Computer Properties* dialog is displayed. In the *Startup* tab, the following settings are made.
2	In the multi-client project, the *Graphics Runtime* properties are set. The *Alarm Logging Runtime* and *Tag Logging Runtime* properties cannot be selected. Exit the dialog box by clicking on *OK*. 

2.5 Description of the WinCC Projects

Activate runtime on both servers. Following that, runtime can also be activated on the multi-client project.
If runtime is activated on the multi-client project before the server project, communication problems would arise, since the multi-client project references the data of the servers.

2.5.1 Server Project

After the appearance of the overview picture, the plant picture can be accessed via the button displayed above.

Via the button displayed above, you can switch among the individual pictures.

Via this button, you can go back to the overview.

Plant Picture

In the plant picture, an oven with a temperature control is displayed. With this temperature control, a temperature can be preset. The temperature in the oven rises, until the preset value has been reached. With the power control, the heating capacity can be specified. This value influences the speed with which the oven temperature rises.

Trend and Table Windows

In the next picture, the trend and table windows are displayed. The trend window depicts the progress of the preset temperature (setpoint value) and the oven temperature (actual value). Both of these values are also displayed in the table window.

Message Window

The next picture displays the message window. If the oven temperature exceeds the value of 300, a warning is generated and displayed in the message window. If the value of 700 is exceeded, an alarm is generated and displayed in the message window.

2.5.2 Multi-Client Project

View to the Server

After the appearance of the overview picture, the pictures providing the view to both servers can be accessed via the button displayed above.
In the first picture, the plant pictures of both servers are displayed. In the following pictures, the trend, table and message windows of both servers are displayed.

Connection to the Server

Via the button displayed above, the pictures in which the multi-client objects have been configured can be accessed. These objects have been connected to various server process tags.
In the first picture, ovens with temperature controls are displayed. The process tags of the servers are displayed. They can also be changed. In the following pictures, trend, table and message windows are configured, which again are linked to the data of the respective servers.

3 Distributed Servers

The projects created in this chapter can also be copied directly from the online document to your hard drive. By default, they will be stored to the *C:\Configuration_Manual* folder. You have the option to copy the following components to the hard drive:

Project_DisServer_Server

The WinCC server project we will create.

Project_DisServer_Client

The WinCC client project we will create.
In this section, a sample pertaining to the *Distributed Servers* is presented. The samples pertaining to this topic are configured in the WinCC projects *Project_DisServer_Server* and *Project_DisServer_Client*.

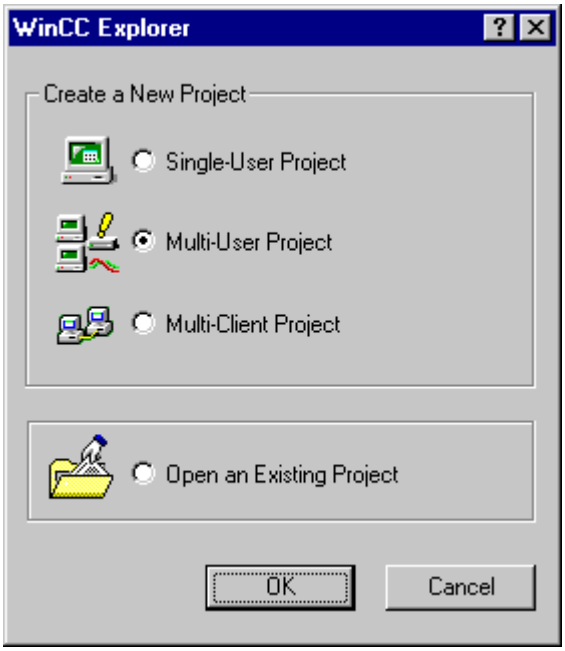
3.1 General Information

In WinCC distributed systems can be configured, i.e. 2 to 6 servers can be controlled and operated by a so-called multi-client.
The distribution through the multi-client is achieved in this way: the pictures of the multi-client contain references to objects on the WinCC servers. These objects can be tags, messages, pictures or archives.
The advantage of distributed systems is that the server computer loads are lightened.
In the sample, a server project and a multi-client project are displayed. The server project is then started on three separate computers, each performing a different function. The multi-client retrieves the data from the corresponding server.

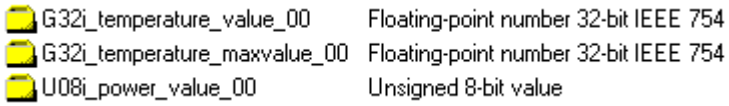
3.2 Creation of the Project_DisServer_Server Project

The following describes in detail the steps necessary to create the multi-client project *Project_DisServer_Server*.
The project is based on the simulation of an oven temperature control, which is then run on three server computers. Each server has different startup properties in runtime and therefore performs different tasks. Configurations are made in the Graphics Designer, Tag Logging, Alarm Logging and Global Script editors.

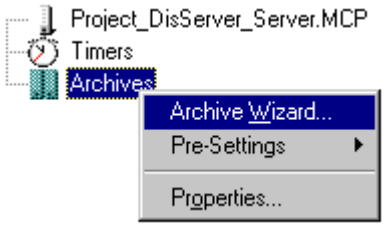
Creating a Server Project

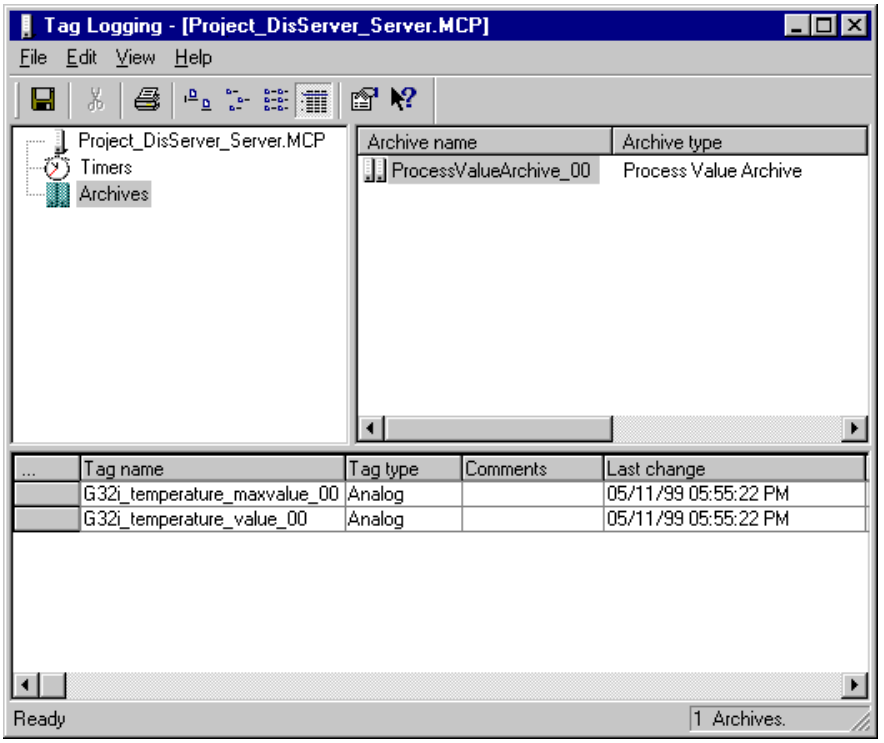
Step	Procedure: Creating a Server Project
1	Creation of a new WinCC project. The WinCC Explorer is started via *Start* → *Simatic* → *WinCC* → *Windows Control Center*.
2	The *WinCC Explorer* will be displayed. Via the menus *File* → *New*, the dialog box for specifying the properties of a new WinCC project will be opened. This sample project is created as a *Multi-User Project*. Exit the dialog box by clicking on *OK*. 

Creating the Tags

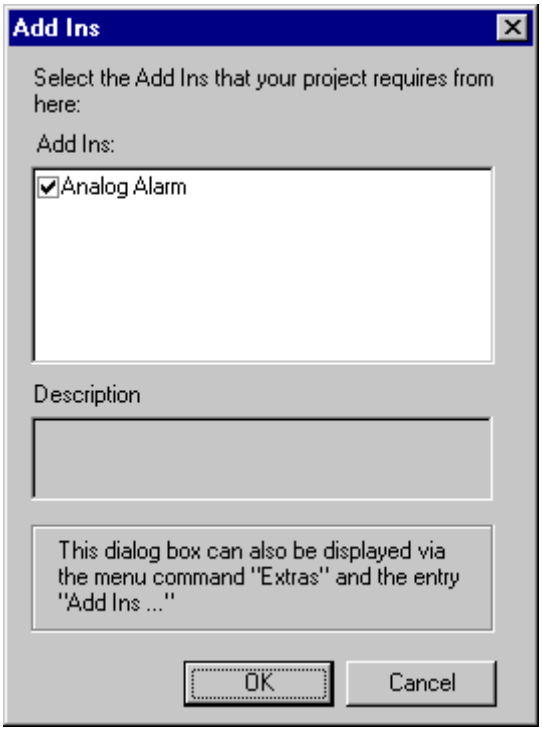
Step	Procedure: Creating the Tags
1	In this sample, three internal tags with the following names are created: They are needed for the simulation of an oven temperature control. The tag *G32i_temperature_value_00* corresponds to the actual value of the temperature, the tag *G32i_temperature_maxvalue_00* to the setpoint value of the temperature and the tag *U08i_power_value_00* to the heating capacity. 

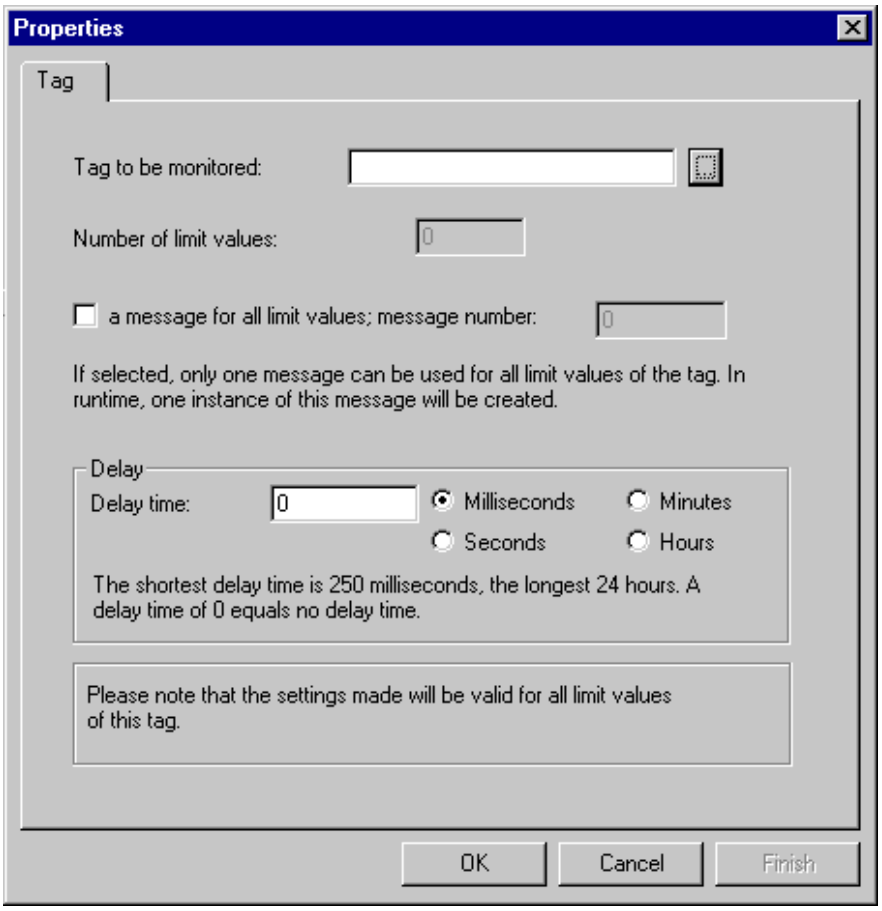
Creating a Process Value Archive

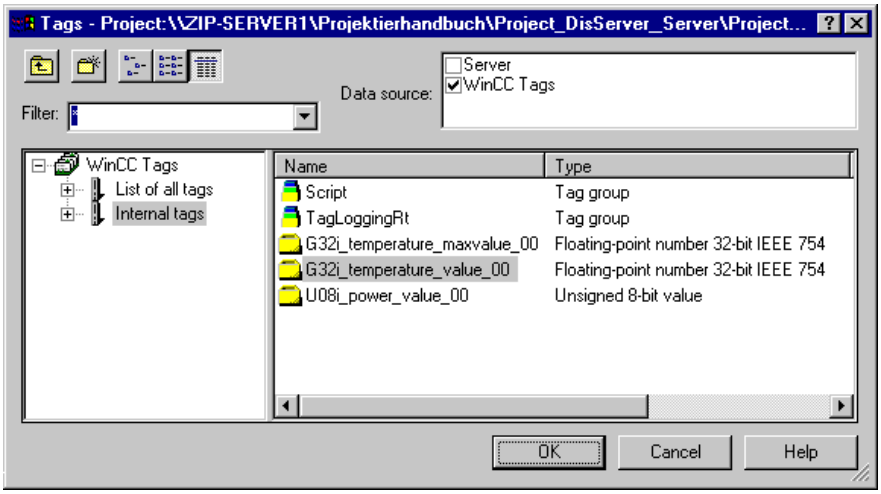
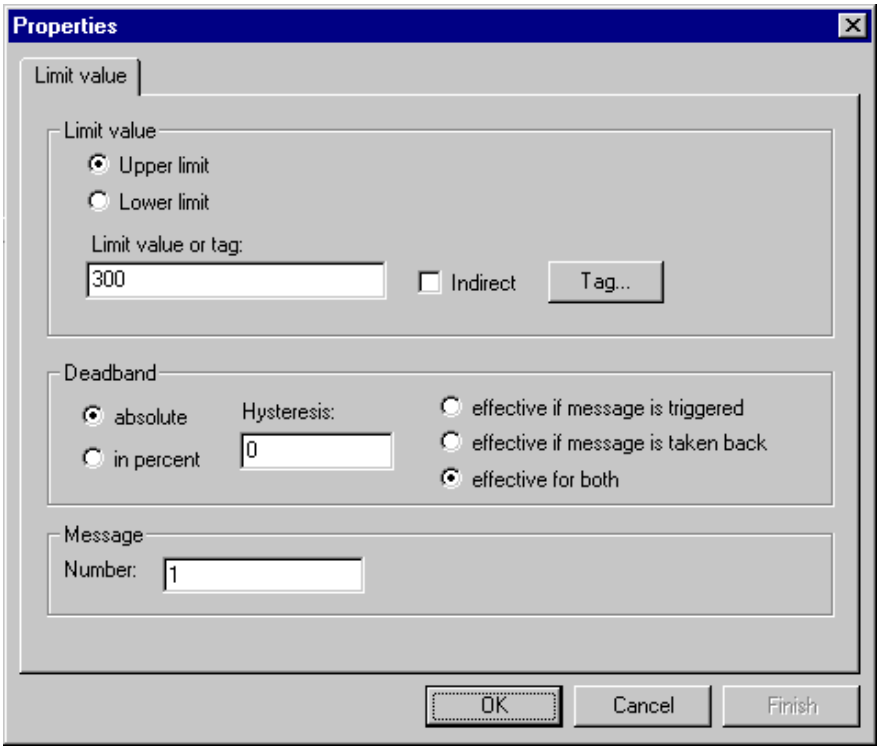
Step	Procedure: Creating a Process Value Archive
1	Open the *Tag Logging* editor.
2	Creation of a process value archive. For this purpose, the Archive Wizard is started via a R on *Archives.* 

Step	Procedure: Creating a Process Value Archive
3	In this sample, the archive has been named *ProcessValueArchive_00*. The tags *G32i_temperature_value_00* and *G32i_temperature_maxvalue_00* are specified as the archive tags. 
4	Save and then exit the *Tag Logging* editor.

Configuring Alarm Logging

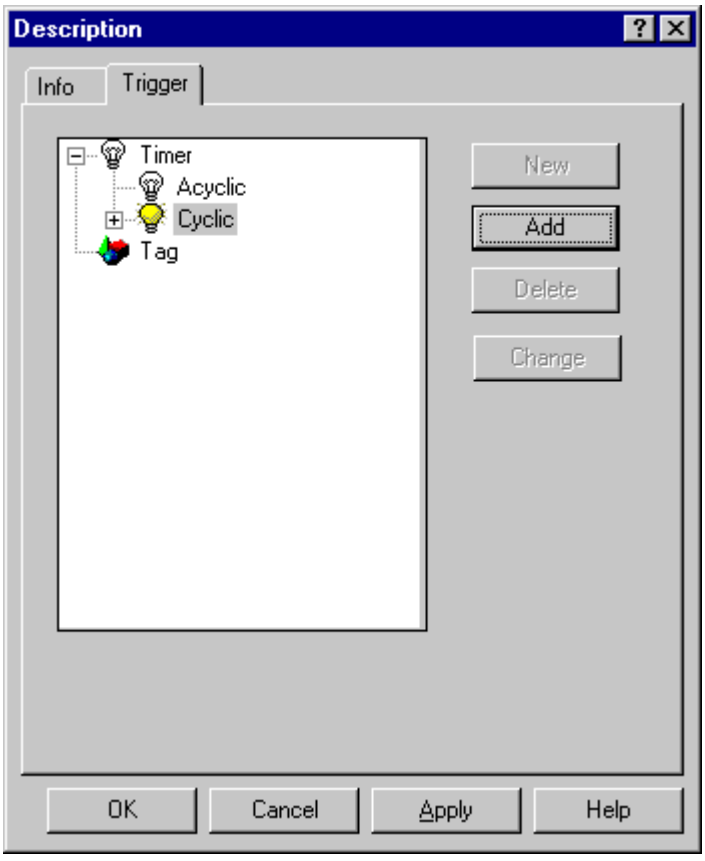
Step	Procedure: Configuring Alarm Logging
1	Open the *Alarm Logging* editor.
2	Creation of single messages. In the lower window of the *Alarm Logging* editor, the already configured messages are displayed. Via a R, a new line can be added. In this sample, two different messages are required. The error type, message text and point of error must be changed correspondingly. ... 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	Configuration of the limit value monitoring. If the *Limit Value Monitoring* (Analog Alarm) entry is not present, it must be loaded first. This is done via the *Options* → *Add Ins* menus in *Alarm Logging*. In the dialog displayed, the check-box for the *Limit Value Monitoring (Analog Alarm)* must be selected. Close the dialog box by clicking on *OK*. 

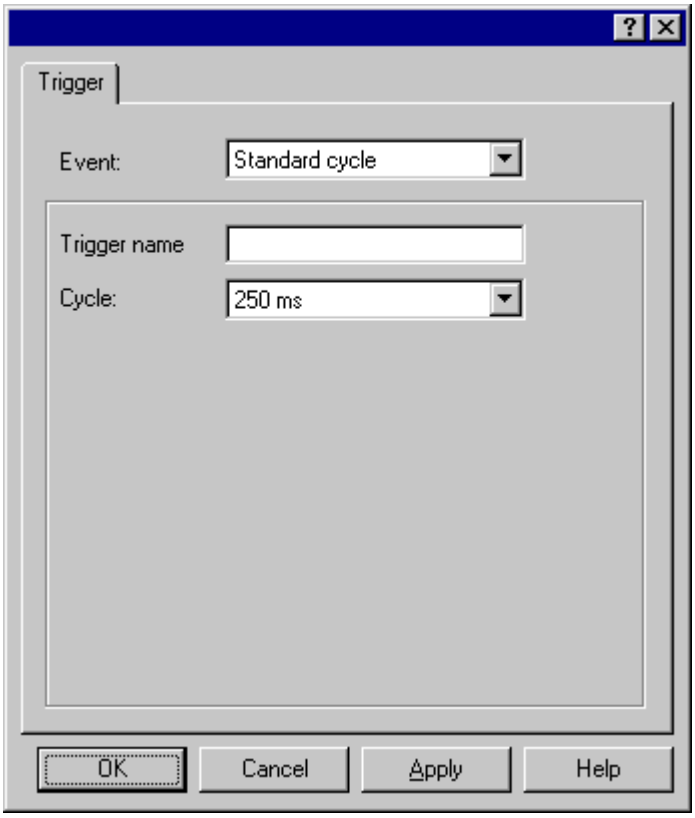
Step	Procedure: Configuring Alarm Logging
4	Via a R on the *Limit Value Monitoring entry and then selecting* → *New...* , the *Properties* dialog of the tag is accessed. In this dialog, a new tag for the limit value monitoring can be set. 
5	Via the button displayed below, the *Select Tag* dialog is accessed. 

Step	Procedure: Configuring Alarm Logging
6	In the left window, the entry *Internal Tags* is selected. The right window will then list the corresponding tags. Select the desired tag. In the sample, this is the *G32i_temperature_value_00* tag. The dialog is closed with the *OK* button. 
7	The *Properties* dialog of the tag is closed with *OK* as well. The right window of *Alarm Logging* will then display the icon of the new tag to be monitored. Via a R on *G32i_temperature_value_00* → *New*, the *Properties* dialog of the limit value is accessed. In this dialog, a new limit value can be assigned to the tag. In this sample, the *Upper Limit* is set to *300* and the message number to *1*. The dialog is closed with the *OK* button. 

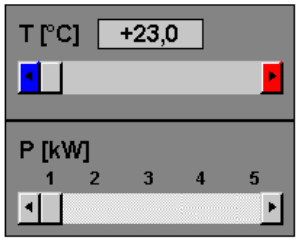
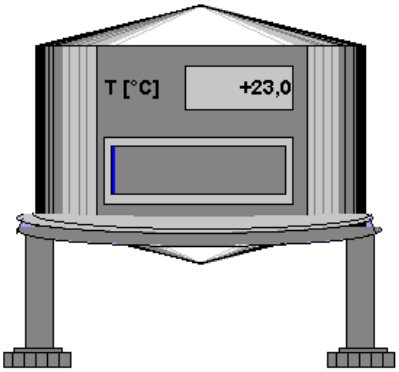
Step	Procedure: Configuring Alarm Logging
8	Following the previously described step, a second limit value is assigned to the tag. The *Upper Limit* is set to *700* and the message number to *2*.
9	Save and then exit the *Alarm Logging* editor.

Creating a Global Action

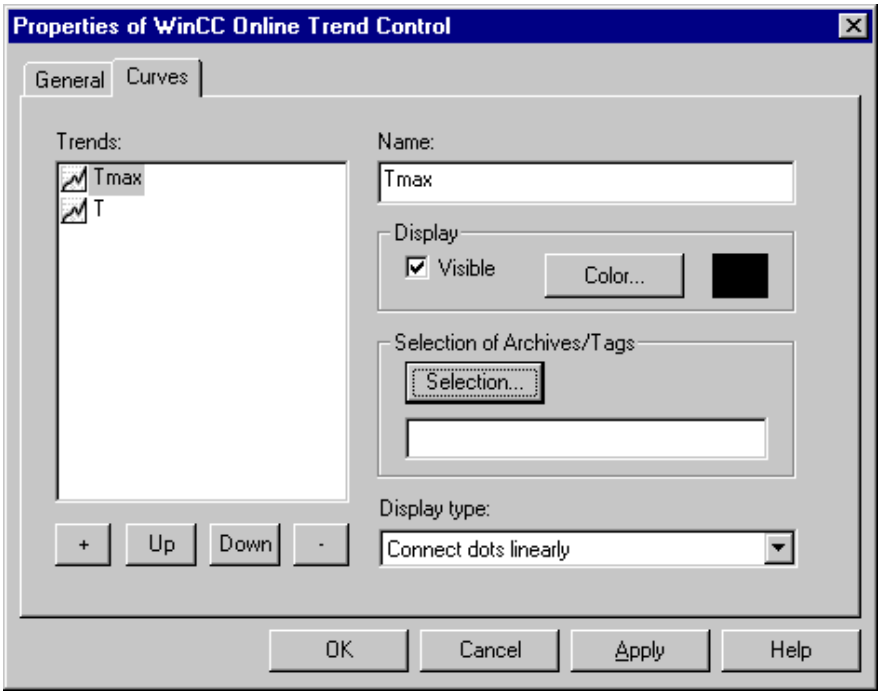
Step	Procedure: Creating a Global Action
1	Open the *Global Script* editor.
2	Creation of a new global action. This is carried out via the *File* → *New Action* menus in the Global Script editor.
3	In this sample, a C-Action has been programmed that simulates an e-function as a trend. The difference *dDelta* between the setpoint temperature *dTemp2* and the actual temperature *dTemp1* is computed. If this difference is positive, the trend increases. If it is negative, the trend drops. The heating capacity *nPower* defines, how fast the temperature reaches the setpoint value.
4	Via *Edit* → *Compile*, the following C-Action is compiled.
5	Via *Edit* → *Info*, the *Description* dialog is opened. The *Trigger* tab is selected. In this sample, a *Cyclic Timer* is selected. Via the *Add* button, the dialog for changing the trigger is displayed. 

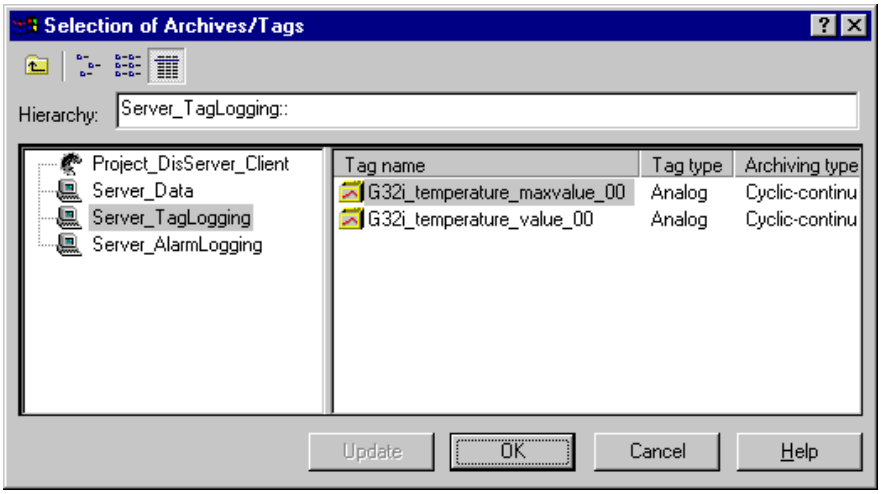
Step	Procedure: Creating a Global Action
6	The cycle time is set to *250 ms*. Both dialogs are closed with *OK*. 
7	Save and then exit the *Global Script* editor.

Graphics Designer

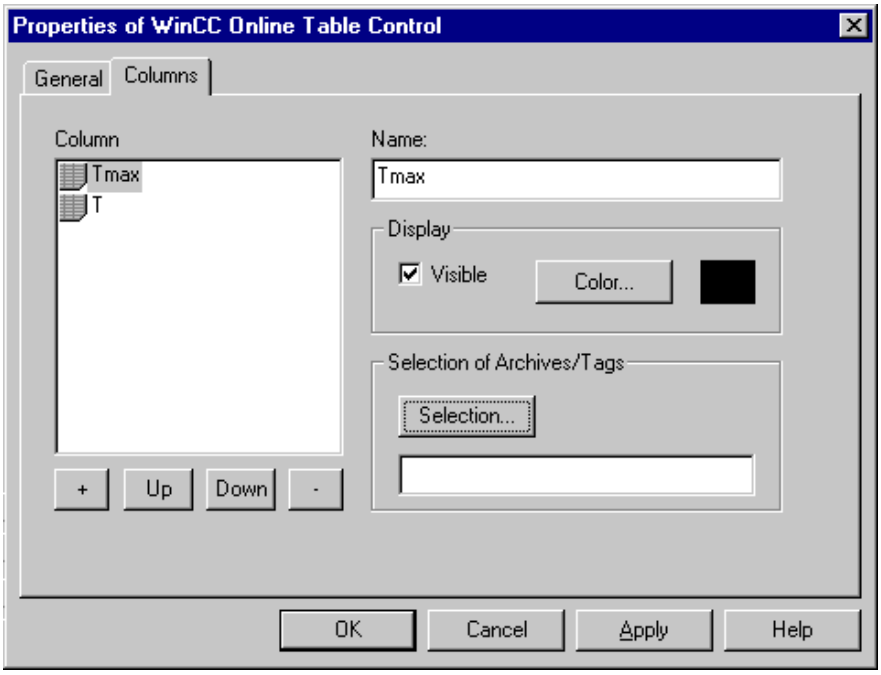
Step	Procedure: Graphics Designer
1	Create a new picture in the *Graphics Designer*. In the sample, this is the *dss_3_chapter_01.PDL* picture. In this picture, various objects are connected to process tags.
2	The simulation of the input tags is implemented via a *Windows Object* → *Slider Object* each. In this sample, these are the *Slider Object1 (G32i_temperature_maxvalue_00)* and *Slider Object2(U08i_power_value_00)* that together with the *I/O Field1 (G32i_temperature_maxvalue_00)* represent the temperature control. In *I/O Field1*, the value of the setpoint temperature is displayed and can also be changed there. The output tag *(G32i_temperature_value_00)* is displayed in the oven. It consists of the *I/O Field2* and the *Bar1* objects. The update of these objects is set to *Upon Change*.  

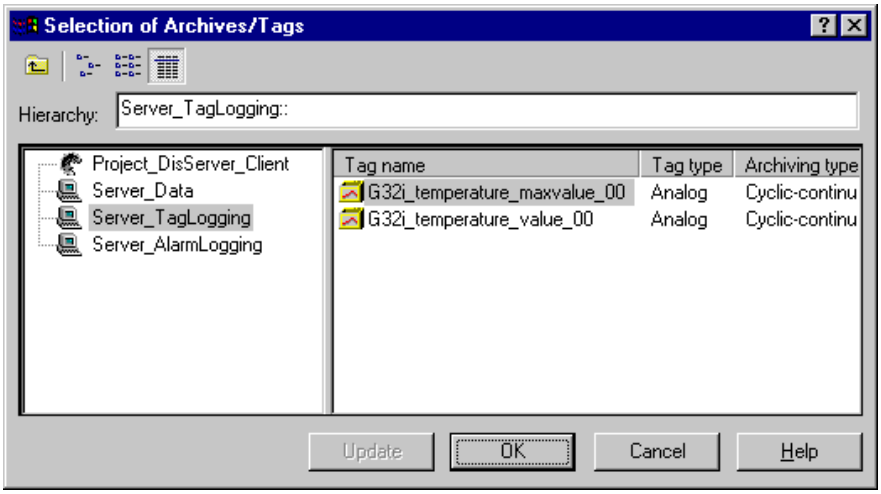
Configuring the Trend Windows

Step	Procedure: Configuring the Trend Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *dss_3_chapter_02.PDL* picture. In this picture, two temperature values are displayed using trend windows.
2	Creation of a *Trend Control* via *Control* → *WinCC Online Trend Control*. In the sample, this is the *TlgOnlineTrend1* object. The dialog *WinCC Online Trend Control Properties* is displayed. In the *Trends* tab, a new trend is added by clicking on the + button. *Trend 1* is renamed to *Tmax* and *Trend 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

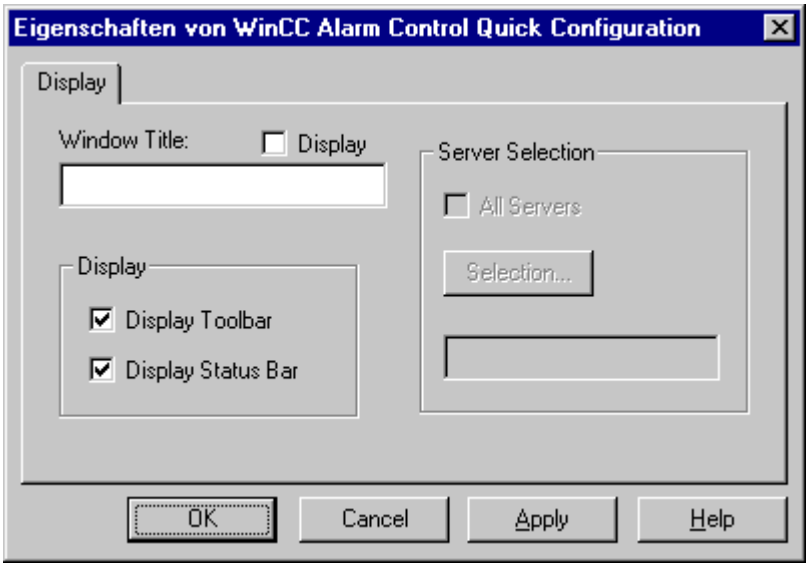
Step	Procedure: Configuring the Trend Windows
3	The *Archive/Tag Selection* dialog is displayed. This dialog allows the selection of archives/archive tags. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag. 

Configuring the Table Windows

Step	Procedure: Configuring the Table Windows
1	In the same picture (*dss_3_chapter_02.PDL*), two temperature values are displayed using table windows.
2	Creation of a *Table Control* via *Control* → *WinCC Online Table Control*. In the sample, this is the TlgOnlineTable1 object. The dialog *WinCC Online Table Control Properties* is displayed. In the *Columns* tab, a new column is added by clicking on the + button. *Column 1* is renamed to *Tmax* and *Column 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

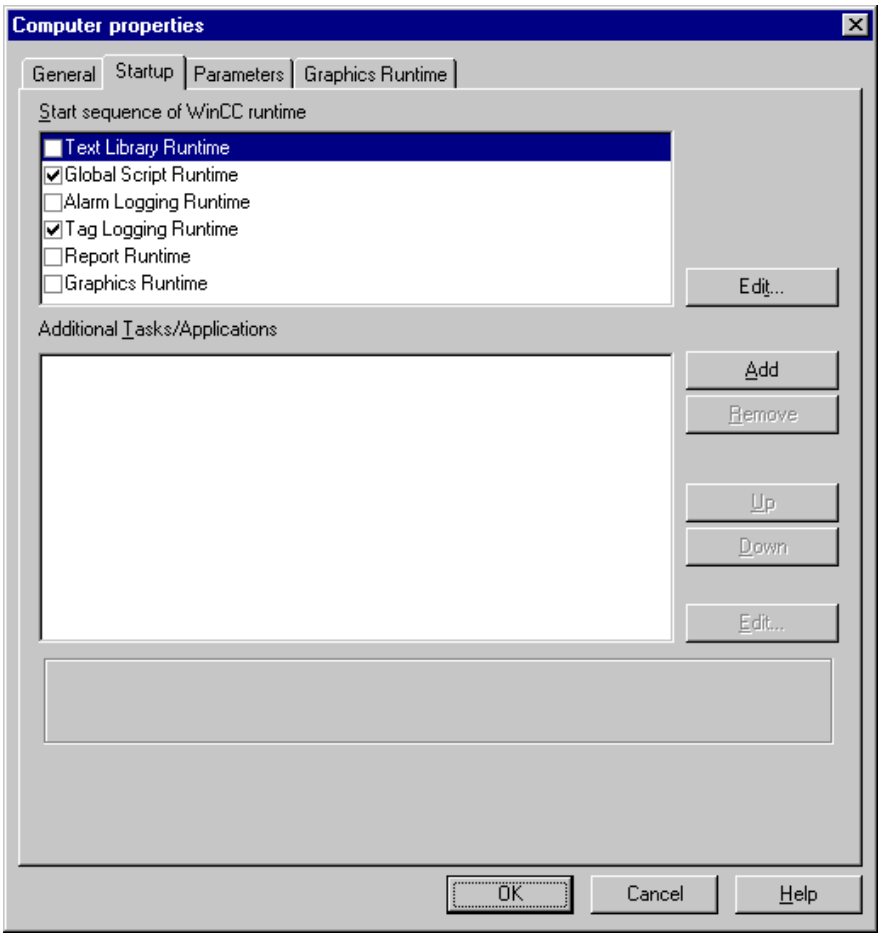
Step	Procedure: Configuring the Table Windows
3	The *Archive/Tag Selection* dialog is displayed. This dialog allows the selection of archives/archive tags. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag. 

Configuring the Message Windows

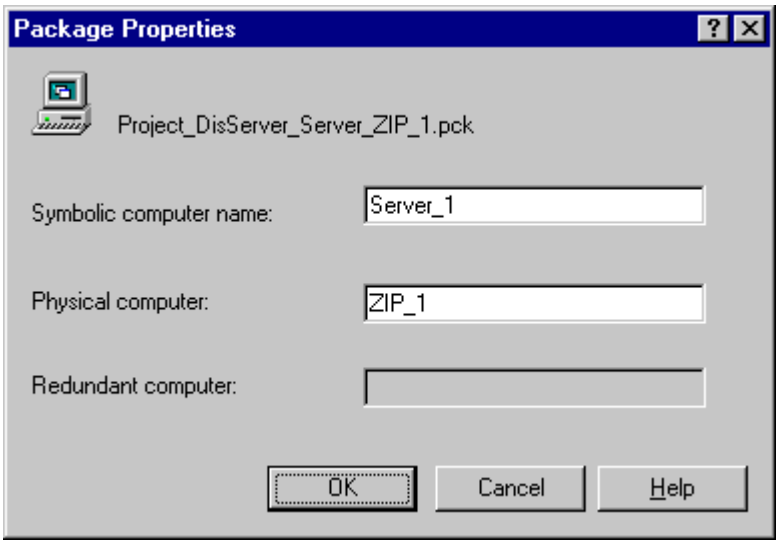
Step	Procedure: Configuring the Message Windows
1	Creation of an additional picture in the *Graphics Designer*. In this picture, the configured messages are output using message windows. In the sample, this is the *dss_3_chapter_03.PDL* picture.
2	Creation of an *Alarm Control* via *Control* → *WinCC Alarm Control*. In the sample, this is the CCAlgWinCtrl1 object. The dialog *WinCC Alarm Control Properties - Quick Configuration* is displayed. This dialog is closed with *OK*. 

Setting the WinCC Runtime Startup Properties

As already mentioned, this server project runs on three computers - with each computer only performing its assigned function - which lowers the computer loads. In order to achieve this, the runtime properties must be changed accordingly.
For the server that keeps the archives for the trends and tables (Tag Logging), the properties for Tag Logging Runtime and Global Script Runtime are set:

Step	Procedure: Setting the WinCC Runtime Startup Properties for the Tag Logging Server
1	Via a on the *Computer* entry on the left side of the *WinCC Explorer*, the computer name will be displayed on the right.
2	*Via a* R on *Computer Name* → *Properties*, the *Computer Properties* dialog is displayed. In the *Startup* tab, the following settings are made. Close the dialog box by clicking on *OK*. 
3	Following the above steps, the *Alarm Logging Runtime* and *Global Script Runtime* properties are set for the Alarm Logging server and the *Global Script Runtime* and *Graphics Runtime* properties for the data server.

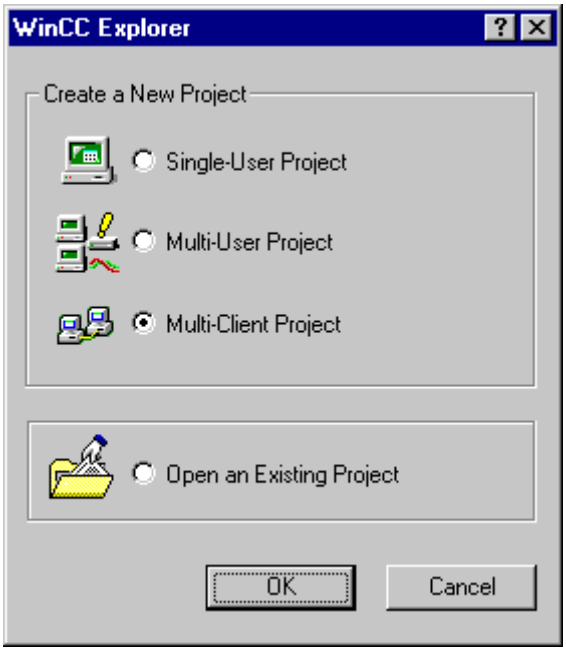
Generating the Server Data

Step	Procedure: Generating the Server Data for the Tag Logging Server
1	The server data is generated via a R on *Server Data* → *Generate* on the left side of the WinCC Explorer. A message stating that the server data has been generated successfully will be displayed. This dialog is acknowledged with OK. Following that, the generated package is displayed on the right side of the WinCC Explorer. 
2	*Via a* R on the newly generated package and *Properties*, the dialog *Package Properties* is displayed. In this sample, the *Symbolic Computer Name* is renamed to *Server_TagLogging*. The dialog is closed with the *OK* button. 
3	Following the above steps, the server data on the two other servers are generated. On the Alarm Logging server, the *Symbolic Computer Name* is renamed to *Server_AlarmLogging* and on the *data server* to *Server_Data*.

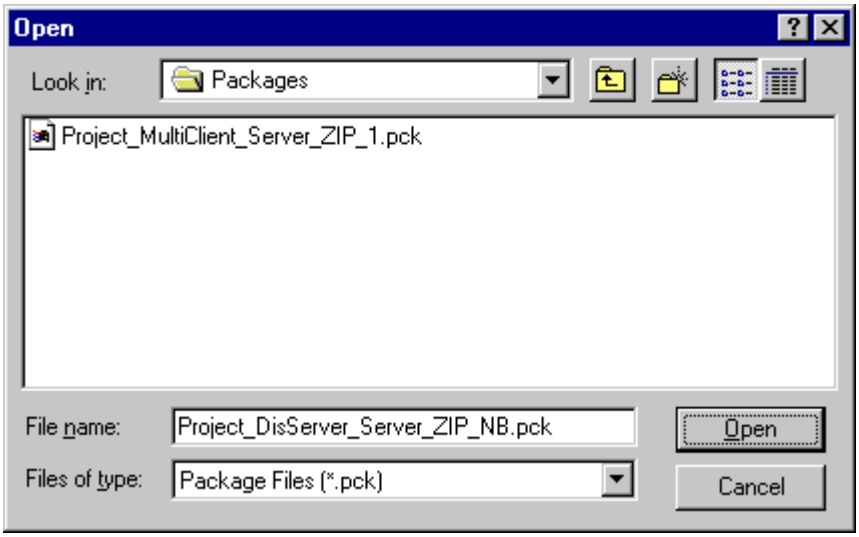
3.3 Creation of the Project_DisServer_Client Project

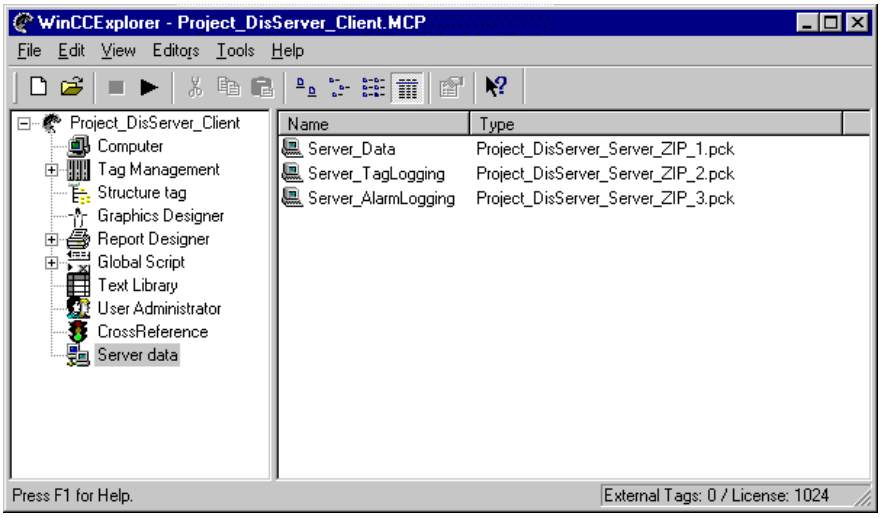
The following describes in detail the steps necessary to create the multi-client project *Project_DisServer_Client.*
This project references the data of the three previously configured servers *Server_Data, Server_TagLogging* and *Server_AlarmLogging.*

Creating a Multi-Client Project

Step	Procedure: Creating a Multi-Client Project
1	Creation of a new WinCC project. The WinCC Explorer is started via *Start* → *Simatic* → *WinCC* → *Windows Control Center*.
2	The *WinCC Explorer* will be displayed. Via the menus *File* → *New*, the dialog box for specifying the properties of a new WinCC project will be opened. This sample project is created as a *Multi-Client Project.* Exit the dialog box by clicking on *OK*. 

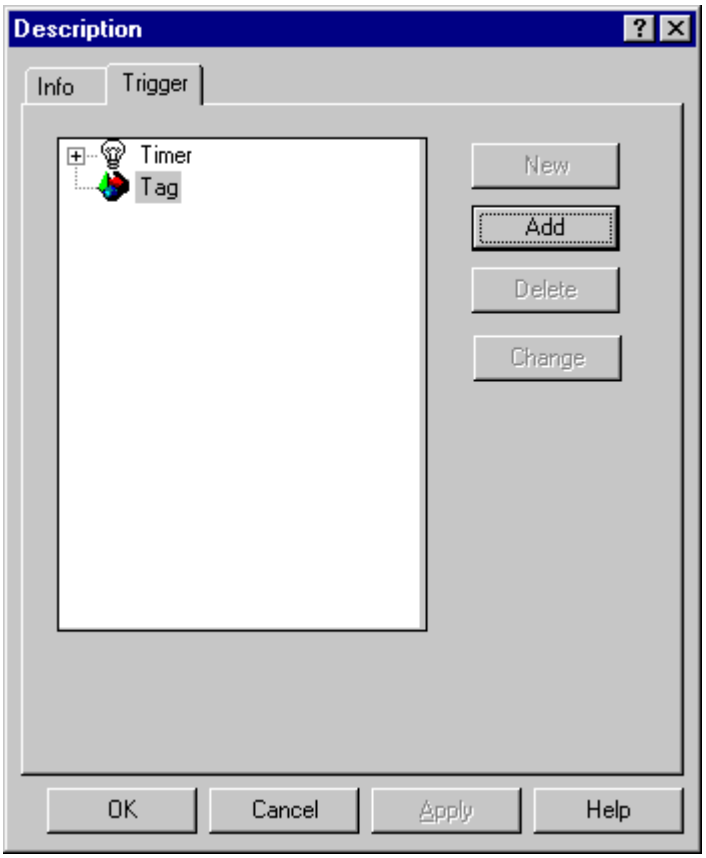
Loading the Server Data

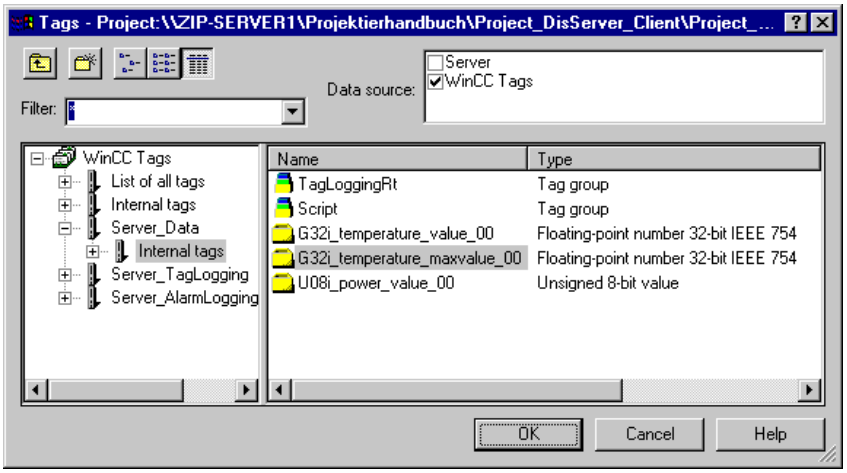
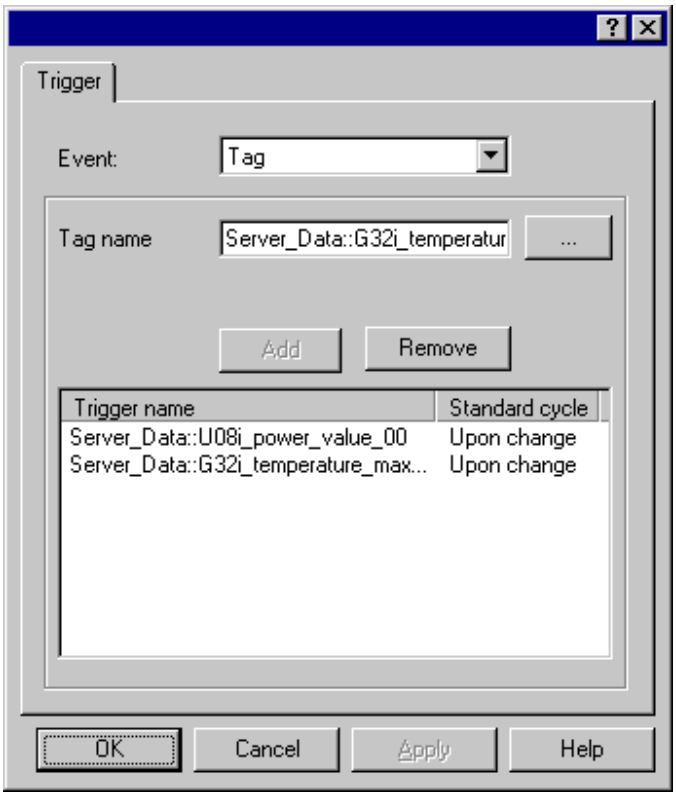
Step	Procedure: Loading the Server Data
1	*Via a* R on *Server Data* → *Load* on the left side of the *WinCC Explorer*, the dialog Open is displayed.
2	From the *Network Neighborhood*, the server computer is selected. The package file is located on the server in the following folder: *Project Name* → *Name of the Tag Logging Servers* → *Packages* This file is selected and loaded via the *Open* button. 
3	A dialog confirming the successful loading of the server data will be displayed. *This dialog is acknowledged with OK.* 

Step	Procedure: Loading the Server Data
4	Following the steps just described, the package files of the other two servers are loaded. The loaded packages will be displayed in the right window of the *WinCC Explorer*. 

Creating a Global Action

Step	Procedure: Creating a Global Action
1	Open the *Global Script* editor.
2	Creation of a new global action. This is carried out via the *File* → *New Action* menus in the *Global Script* editor.
3	In the sample, the following C-Action has been programmed. This action transfers the input values (setpoint temperature and heating capacity) to all three servers upon a change.
4	Via *Edit* → *Compile*, the C-Action is compiled.

Step	Procedure: Creating a Global Action
5	Via *Edit* → *Info*, the *Description* dialog is opened. The *Trigger* tab is selected. In this sample, the trigger is set depending on the change of the two input values. Via the *Add* button, the dialog is accessed in which the tags responsible for this can be configured. 
6	Via a on the button displayed below, the *Select Tag* dialog is accessed. 

Step	Procedure: Creating a Global Action
7	A window from which the tags can be selected is displayed. In this sample, the *G32i_temperature_maxvalue_00* tag from the *Internal Tags* of the *Server_Data* is selected. Close the dialog box by clicking on *OK*. 
8	Following steps 6 and 7, the *U08i_power_value_00* tag from the *Internal Tags* of the *Server_Data* is added.
9	Following that, the dialog in which the standard cycle can be selected is displayed again. In this sample, the standard cycle is set to *Upon Change* for both tags. 
10	The dialogs are closed with *OK*. Save and then exit the *Global Script* editor.

C-Action

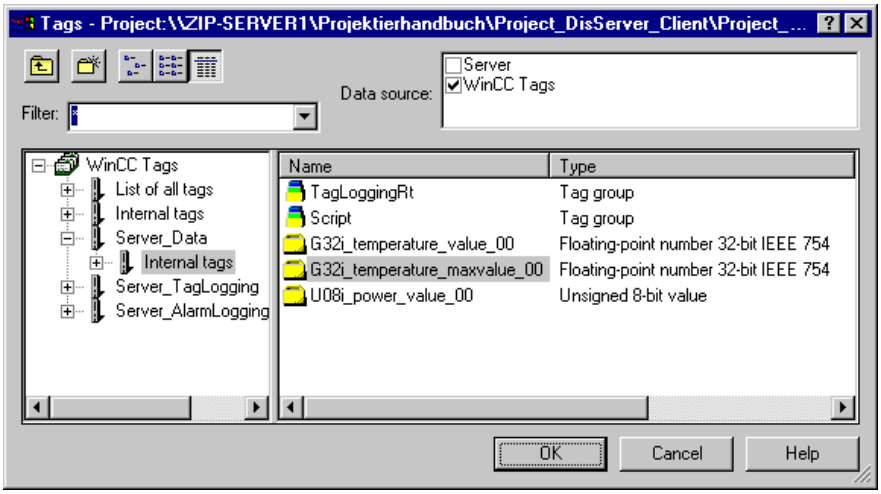
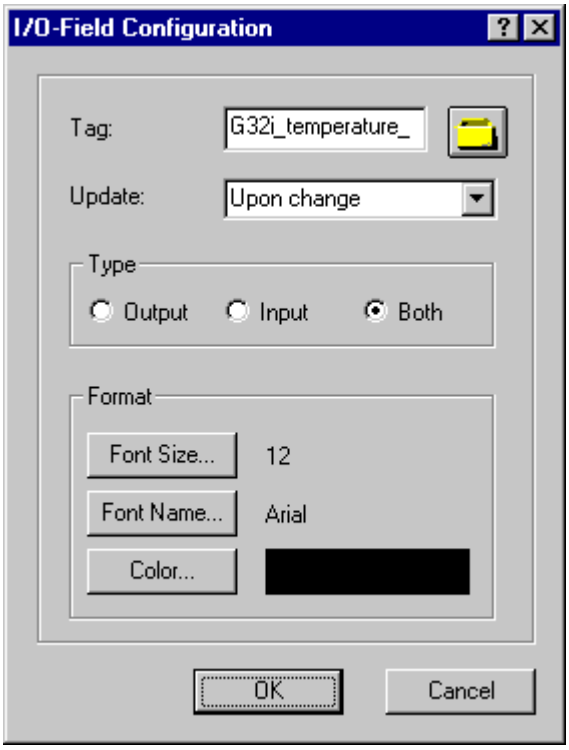
```
#include "apdefap.h"

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;
}
```

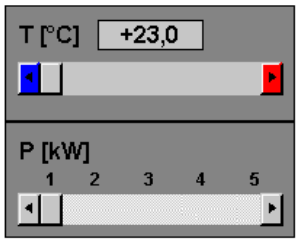
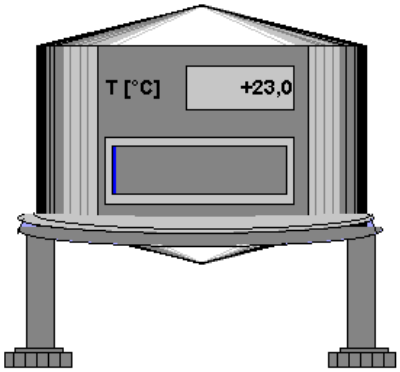
Graphics Designer

In the multi-client project, no tags have been created, i.e. it works with the tags of the servers. The trend and table windows are connected to the archive tags on the Tag Logging server, the message window works with the tag on the Alarm Logging server. The remaining objects (I/O fields, slider objects, etc.) are connected to the tags on the data server.

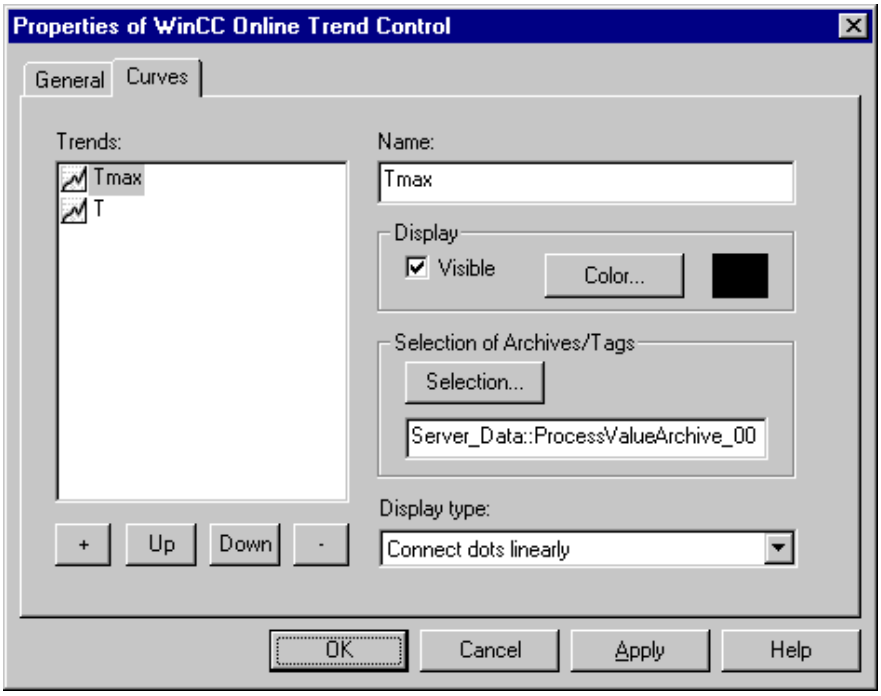
Configuring the Objects

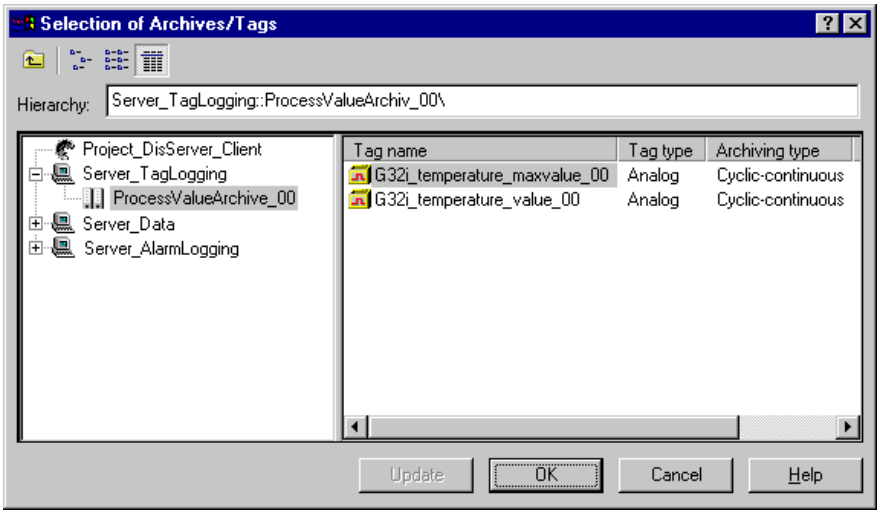
Step	Procedure: Configuring the Objects
1	Create a new picture in the *Graphics Designer*. In the sample, this is the *dsc_3_chapter_01.PDL* picture. In this picture, various objects are connected with the process tags of the *Server_Data*.
2	Configure a *Smart Object* → *I/O Field*.In the sample, this is the *I/O Field1* object. Its *Configuration Dialog will be displayed.* Via a on the button displayed below, the *Select Tag* dialog is accessed. 

Step	Procedure: Configuring the Objects
3	In the left window, the *Internal Tags* entry of the desired server is selected. The right window will then list the corresponding tags. Select the desired tag. In the sample, this is the *G32i_temperature_maxvalue_00* tag of the *Server_Data*. The dialog is closed with the *OK* button. 
4	The *Update* is set to *Upon Change*. The *configuration dialog* can be exited by clicking on *OK*. 

Step	Procedure: Configuring the Objects
5	Configuration of additional objects (*I/O Fields, Slider Objects, Bar Graphs*) to display the remaining tags of the servers.  

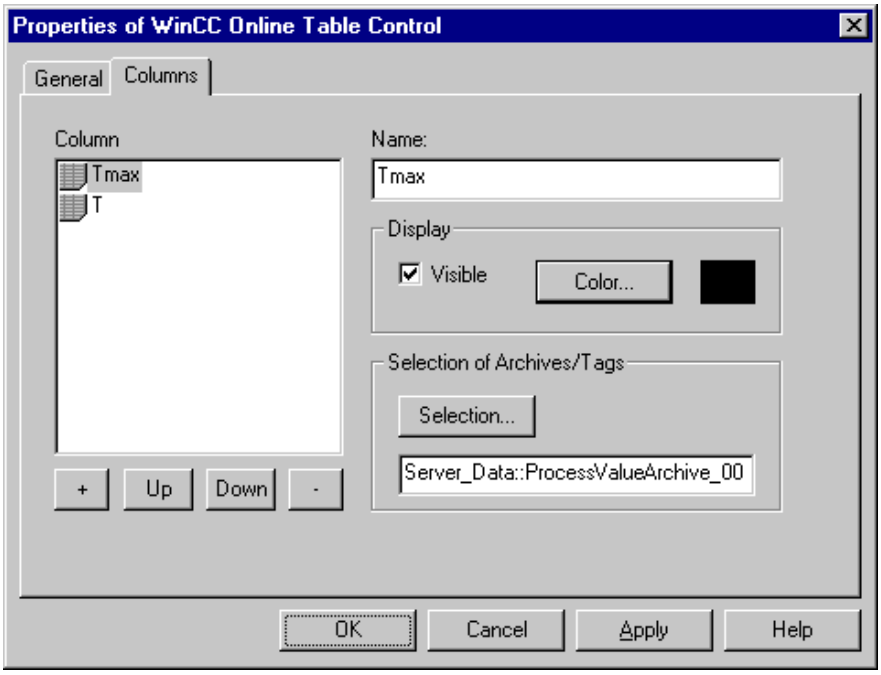
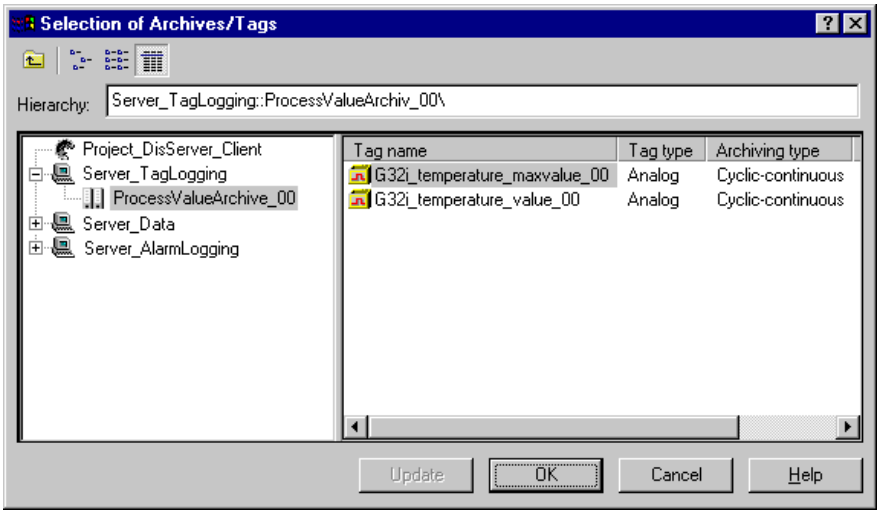
Configuring the Trend Windows

Step	Procedure: Configuring the Trend Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *dsc_3_chapter_02.PDL* picture. In this picture, the two temperature values of *Sever_TagLogging* are displayed using trend windows.
2	Creation of a *Trend Control* via *Control* → *WinCC Online Trend Control*. In the sample, this is the *TlgOnlineTrend1* object. The dialog *WinCC Online Trend Control Properties* is displayed. In the *Trends* tab, a new trend is added by clicking on the + button. *Trend 1* is renamed to *Tmax* and *Trend 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

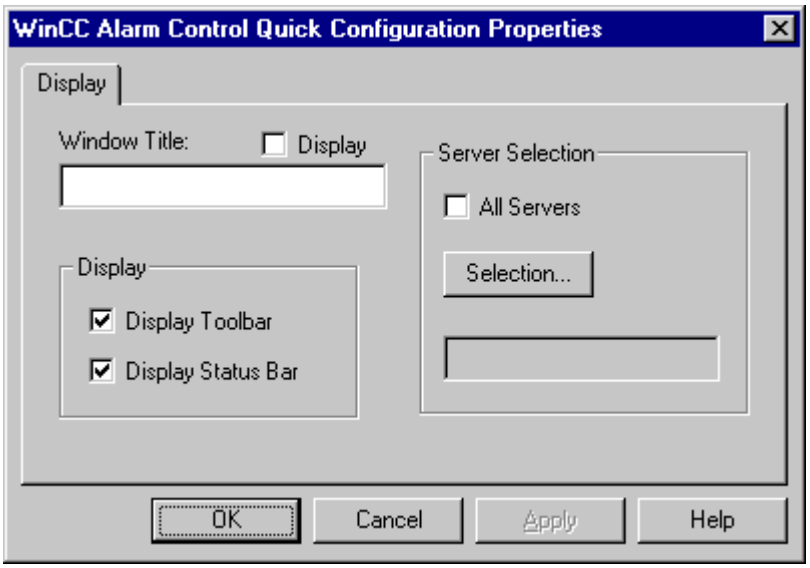
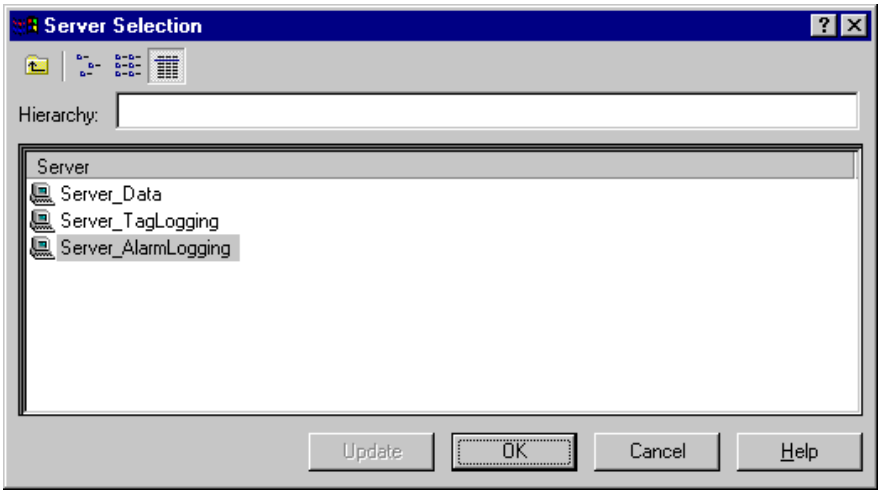
Step	Procedure: Configuring the Trend Windows
3	The *Archive/Tag Selection* dialog is displayed. From this dialog, the servers/archives/archive tags can be selected from the server data imported by the packages. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag of *Server_TagLogging*. 

Configuring the Table Windows

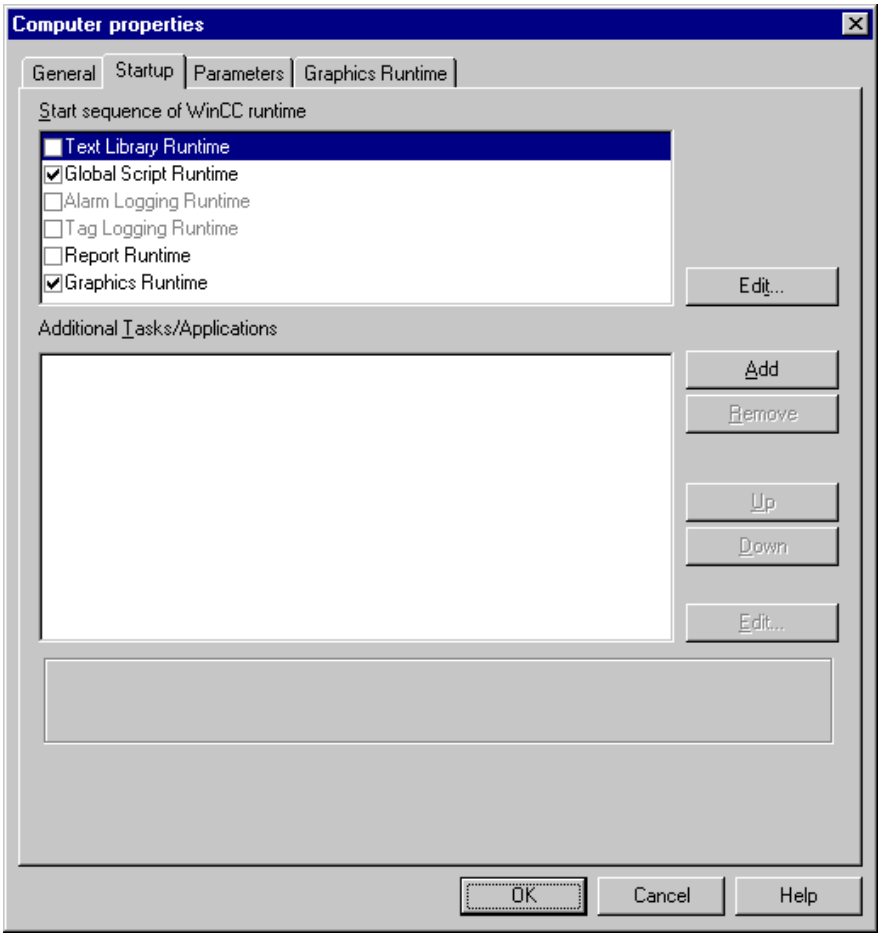
Step	Procedure: Configuring the Table Windows
1	In the same picture (*dsc_3_chapter_02.PDL*), the two temperature values of *Server_TagLogging* are displayed using table windows.

Step	Procedure: Configuring the Table Windows
2	Creation of a *Table Control* via *Control* → *WinCC Online Table Control*. In the sample, this is the TlgOnlineTable1 object. The dialog *WinCC Online Table Control Properties* is displayed. In the *Columns* tab, a new column is added by clicking on the + button. *Column 1* is renamed to *Tmax* and *Column 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 
3	The *Archive/Tag Selection* dialog is displayed. From this dialog, the servers/archives/archive tags can be selected from the server data imported by the packages. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag of *Server_TagLogging*. 

Configuring the Message Windows

Step	Procedure: Configuring the Message Windows
1	Creation of an additional picture in the *Graphics Designer*. In this picture, the messages configured on *Server_AlarmLogging* are output using message windows. In the sample, this is the *dsc_3_chapter_03.PDL* picture.
2	Creation of an *Alarm Control* via *Control* → *WinCC Alarm Control*. In the sample, this is the CCAlgWinCtrl1 object. The dialog *WinCC Alarm Control Properties - Quick Configuration* is displayed. Via the *Select* button, the *Server Selection* dialog is accessed. 
3	In this sample, *Server_AlarmLogging* is selected. The dialog is then closed with the *OK* button. 

Setting the WinCC Runtime Startup Properties

Step	Procedure: Setting the WinCC Runtime Startup Properties
1	Via a on the *Computer* entry on the left side of the *WinCC Explorer*, the computer name will be displayed on the right. Through a R on *Computer Name* → *Properties*, the *Computer Properties* dialog is displayed. In the *Startup* tab, the following settings are made.
2	In the multi-client project, the *Global Script Runtime* and *Graphics Runtime* properties are set. The properties for *Alarm Logging Runtime* and *Tag Logging Runtime* are grayed out and cannot be set. Exit the dialog box by clicking on *OK*. 

3.4 Description of the WinCC Projects

Activate runtime on all three servers. Following that, runtime can also be activated on the multi-client project.
If runtime is activated on the multi-client project before the server project, communication problems would arise, since the multi-client project references the data of the servers.
The overview pictures are displayed on the data server and the multi-client. On the other two servers (*Server_TagLogging* and *Server_AlarmLogging*), the properties for *Graphics Runtime* have not been set.

3.4.1 Server Project

After the appearance of the overview picture, the plant picture can be accessed via the button displayed above.

Via the button displayed above, you can switch among the individual pictures.

Via this button, you can go back to the overview.

Plant Picture

In the plant picture, an oven with a temperature control is displayed. With this temperature control, a temperature can be preset. The temperature in the oven rises, until the preset value has been reached. With the power control, the heating capacity can be specified. This value influences the speed with which the oven temperature rises.

Trend and Table Windows

In the next picture, the trend and table windows are displayed. The trend window depicts the progress of the preset temperature (setpoint value) and the oven temperature (actual value). Both of these values are also displayed in the table window.

Message Window

The next picture displays the message window. If the oven temperature exceeds the value of 300, a warning is generated and displayed in the message window. If the value of 700 is exceeded, an alarm is generated and displayed in the message window.

3.4.2 Client Project

Connection to the three Servers

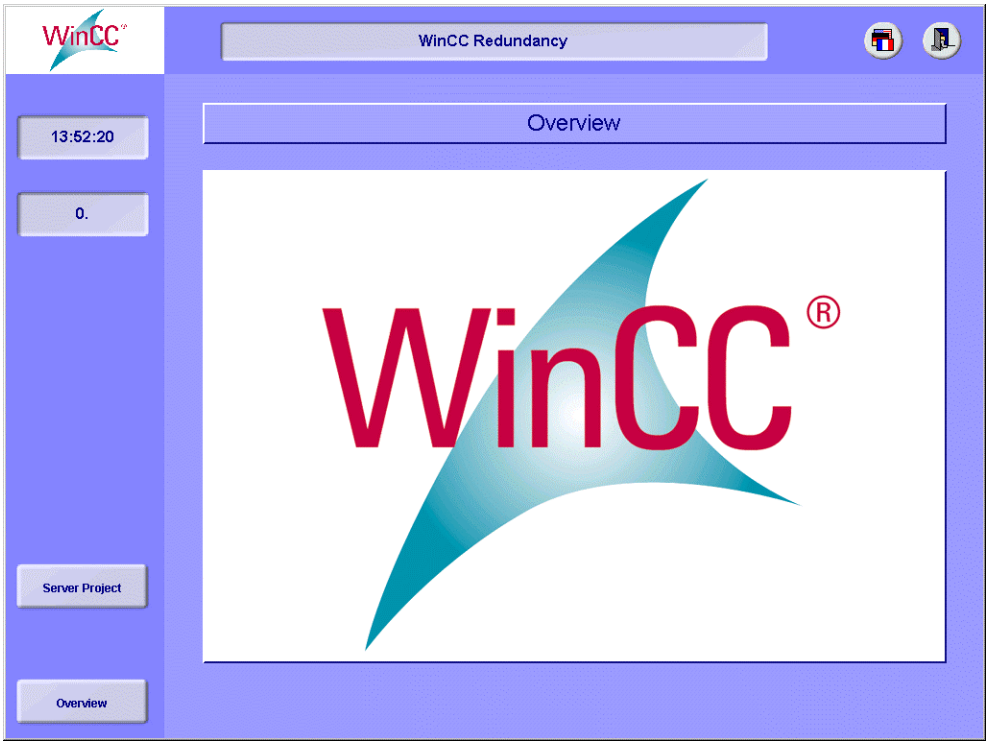
After the appearance of the overview picture, the pictures in which the own objects on the multi-client have been configured can be accessed via the button displayed above. These objects have been connected to various server process tags.
In the first picture, the plant picture is displayed. In the following pictures, the trend, table and message windows are displayed.

4 Redundancy

The project created in this chapter can also be copied directly from the online document to your hard drive. By default, it will be stored to the *C:\Configuration_Manual* folder. You have the option to copy the following components to the hard drive:

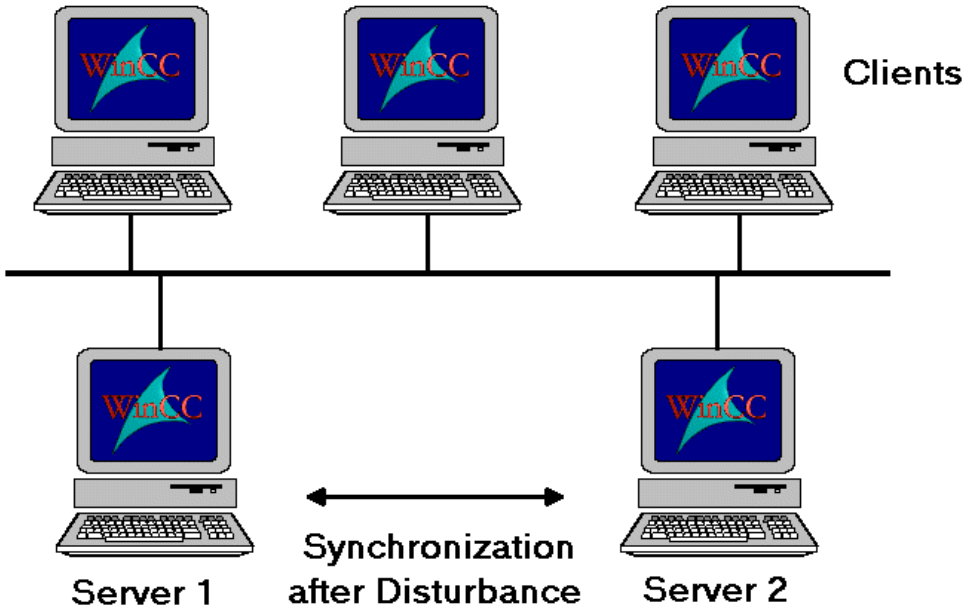

Project_Redundancy_Server

The WinCC project we will create.
In this section, a sample pertaining to the Redundancy is presented.
The samples for this topic are configured in the Project_Redundancy_Server WinCC project.



4.1 General Information

The WinCC Redundancy significantly increases the availability of WinCC and the plant altogether by operating two server PCs connected to each other in parallel.



In order to recognize the failure of a partner early, the servers monitor each other in runtime .
If one of the server computers fails, the clients are automatically switched from the failed server to the still active server. As a result, all clients always remain available for the control and monitoring of the process.
During the failure, the still running server continues to archive all messages and process data of the WinCC project. After the failed server comes back online, the contents of all message, process value and user archives are automatically copied to the returned server. This fills the data gaps in the archives of the failed server. This process is also called the synchronization.
The WinCC Redundancy option offers:

- The automatic synchronization of message, process value and user archives after the return of a failed server.
- The automatic synchronization of message and process value archives after a process connection error has been corrected.
- The online synchronization of message archives in a certain number range (server-local messages).
- The online synchronization of user archives.
- A Project Switcher for the automatic or manual switch of the clients between the redundant servers.
- A Project Duplicator for copying a project to the redundant partner server.

4.1.1 Operation of Redundancy

WinCC Archiving during normal Operation

During normal operation, the process data servers run completely in parallel in runtime. Each server station has a separate process connection and its own data archives. The process data and messages of the PLCs are sent to both redundant servers and are processed by them accordingly.
The servers monitor each other in runtime in order to recognize the failure of a partner early and to issue a process control message.
User archives and messages in certain number ranges can continuously be synchronized online (online synchronization).
Both servers have equal rights, work independently of each other and are available to the user. Should one of the servers fail, an equal redundant server is always available.
The communication between the redundant server stations for lifebeat monitoring and archive synchronization purposes is carried out via the terminal bus. For the network, a PC LAN with TCP/IP or NetBEUI protocols is used.

Failure of a Server

If one of the servers fails, the still functioning server continues to receive and archive the process values and messages from the PLCs. This ensures the complete data integrity.
The clients are switched automatically from the failed server to the redundant partner server. After a brief transfer time, all operating stations will be available again.

Factors triggering the Client Switch

The switch of the clients from the standard server to the partner server is performed automatically by the system during a server failure. The following factors trigger the client switch:

- Network connection error to the server.
- Server Failure

Factors triggering the Archive Synchronization after the Return

The synchronization of the archives between the servers is triggered upon the correction of the following errors:

- Process connection error. The process connection monitoring can be turned off.
- Network connection error to the partner server.
- Server Failure
- Project is not active.
- Project is not open.

Synchronization after the Return

After the failed server comes back online, Redundancy performs an archive synchronization for the failure period. The gap in the archives caused by the failure is filled by transferring the missing data to the failed server. As a result, two equal servers are available again.
A synchronization of the message archive, process value archive and user archive is performed. The failed server will receive its data after a time delay caused by the failure. The different archive types are synchronized in the following order:

- Message Archives
- Process Value Archives
- User Archives
- The archive synchronization is implemented as a background function which runs parallel to the process control and archiving of WinCC. This ensures the continuous control and monitoring of the plant.

Note:

The application of the Storage option in conjunction with Redundancy can cause the following problems:

If, during a server failure, Storage exports and deletes data from the second server, this data cannot be synchronized anymore.

If Storage exports data of a failure period that has not been synchronized yet, this gap in the exported data cannot be filled by the archive synchronization.

To avoid data losses, the Storage option is deactivated during the synchronization of the archives. Nach Abschluß der Archivierung wird Storage" automatisch reaktiviert.

Synchronization after a Process Connection Error

If a network error occurs between a server and one or multiple PLCs during runtime, the synchronization will be started automatically - if so configured - after the error has been corrected.

Online Synchronization (optional)

A direct server to server synchronization (online synchronization) can be performed for user archives and Alarm Logging messages in certain number ranges.

Note:

For the online synchronization of Alarm Logging, the short-term archive must be configured in the Alarm Logging system.

Process Data and Message Archives

Tag Logging and Alarm Logging must be configured functionally identical on the redundant servers. Functionally identical configuration means:
Identical archives, where expansions in the form of additional measurement points and archives are permissible. These expansions will not be synchronized, but must be updated manually on the partner server.
The following archives are synchronized by WinCC:

- Archives located on hard drives, i.e. process value, compressed and message archives. Short-term as well as sequential archives are synchronized.
- However, no synchronization of main memory archives is performed.

For the online synchronization of message archives, the short-term archive must be configured in the Alarm Logging system.

User Archives

For the user archives, the same structure on both servers is required.

- The configuration of the user archives to be synchronized must be identical with regard to the field/record structure and their properties.

Note:

To avoid problems with Redundancy, the clocks on the computers used should be synchronized.

4.1.2 Redundant User Archives

User Archives can be edited by operations, separate programs, PLCs or other functions.

Editing User Archives in Parallel

While adding records in parallel to User Archives that are redundant to each other, the following must be noted:

- Due to runtime reasons, the order in which the records are added can vary.
- Even before the synchronization after the server return is complete, additional records can be added to the previously failed server.
- Even during the online synchronization, some time will pass until the record has been synchronized in the redundant archive.

The configuration of the archives must be identical on both computers. Therefore, the Project Duplicator should be used.
Should the archives not be identical, the following system message will be displayed: Synchronization not ready for all User Archives.

Unique Key

For the clear assignment of the records from one archive to the records of the redundant archive, a unique key field is required. Records with the same content in this field are synchronized with each other. This field must have the property of a unique value, in order to avoid having 2 records with the same content in one archive. This is possible via:

- The record number, which is always part of a record and must not be configured additionally (the record number is always unique). If the record number is used, no other field must have the unique value property.
- An archive field, to which the unique value property has been assigned. If a field other than the record number is used, then this field must be the only one with the unique value property.
- For example: Recipe Name (Text Type)
- Recipe Number (Integer Type)
- Insertion Date/Creation Date (Date Type)

Last Accessed Field

This field must be selected during the configuration of the archive properties, since the time stamp is used as the synchronization criterion.
A data record with a newer time stamp overwrites the older record during the synchronization, resulting in the most current data record being kept. This must be noted while editing in parallel or while making changes during a synchronization.
The time of the last change is automatically entered by the system. During the import of records, the date of change contained in the csv file is accepted unchanged.

	Synchronization via the Record Number	Synchronization via a Unique Value
Deleting a record during a server failure.	Does not take place.	Does not take place
Editing/adding a record during a server failure.	During the offline synchronization, all data records will be synchronized that have been edited or added during the failure period.	During the offline synchronization, all data records will be synchronized that have been edited or added during the failure period.
Online synchronization if a record is edited.	OK	OK The field content of the unique key must not be changed - even though the new record content is still added to the redundant archive, the old record cannot be identified and will also be kept.
Online synchronization if a record is added.	OK If a record is added separately in both redundant archives before a synchronization has completed after the return of a server, then the automatically assigned record number might already exist in the redundant archive. This results in the older of the two records being overwritten.	OK
Online synchronization if a record is deleted.	OK An online synchronization of deleted records is only performed if the synchronization takes place via the record number and the changes are made within an OLE Control Element (OCX) or via API functions of the User Archives. Only those records will be deleted that have an older time in the last accessed field than the time of the deletion.	Not available.

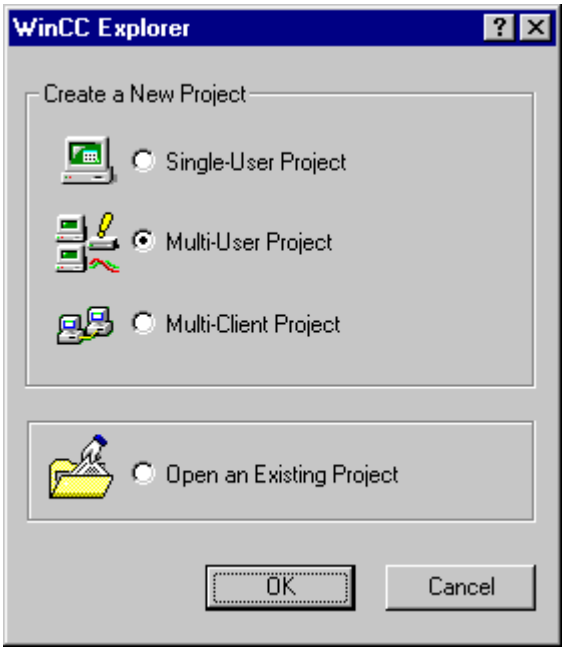
Note:
Records that have a value in the last accessed field which is above the value of the current system time should not be added to the archive (e.g. via an import): A synchronization is only performed up to the current system time.

If the data issuing server is terminated or has a failure before all records have been synchronized online, then only the last 50 records per archive will be synchronized at the next start of runtime.
If WinCC runtime is exited and restarted within 10 seconds - which is generally only possible with small projects - then this is not recognized as a failure and no synchronization is performed upon the return.
The online synchronization memorizes up to 10 records to the redundant server in the case of a connection error and synchronizes them immediately after the connection to the partner has been reestablished, before the actual synchronization is activated.

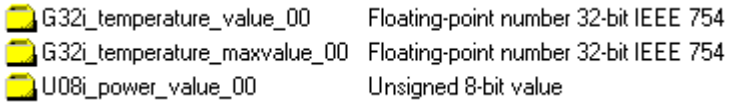
4.2 Creation of the Project_Redundancy_Server Project

The following describes in detail the steps necessary to create *Project_Redundancy_Server* project.
The project is based on the simulation of an oven temperature control, which is then run on two server computers. Configurations are made in the Graphics Designer, Tag Logging, Alarm Logging and Global Script editors.

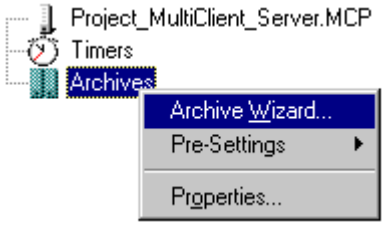
Creating a Server Project

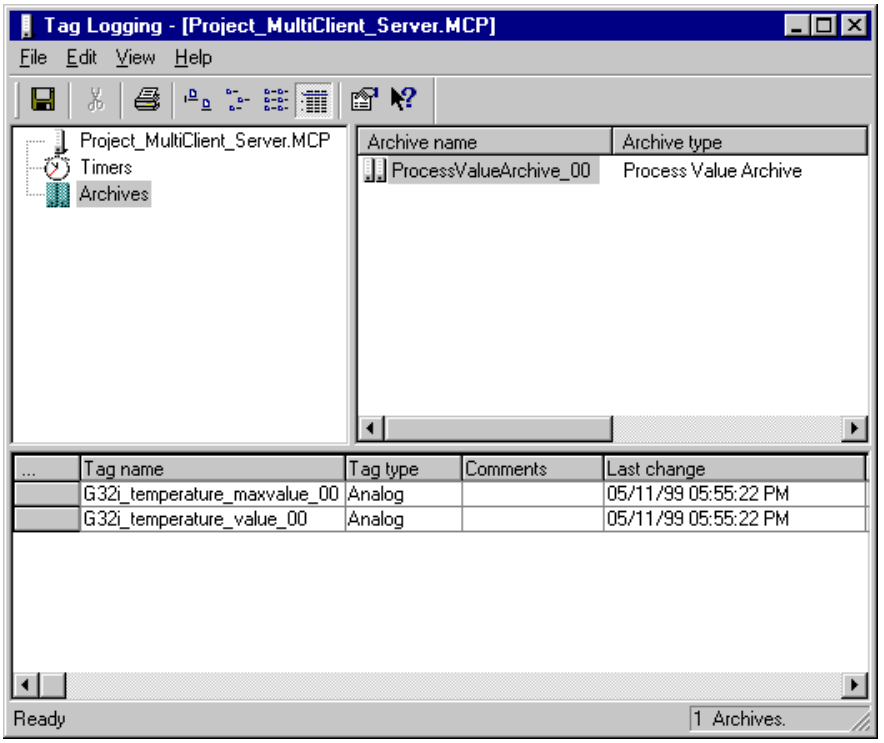
Step	Procedure: Creating a Server Project
1	Creation of a new WinCC project. The WinCC Explorer is started via *Start* → *Simatic* → *WinCC* → *Windows Control Center*.
2	The *WinCC Explorer* will be displayed. Via the menus *File* → *New*, the dialog for specifying the properties of a new WinCC project will be opened. This sample project is created as a *Multi-User Project*. Exit the dialog by clicking on *OK*. 

Creating the Tags

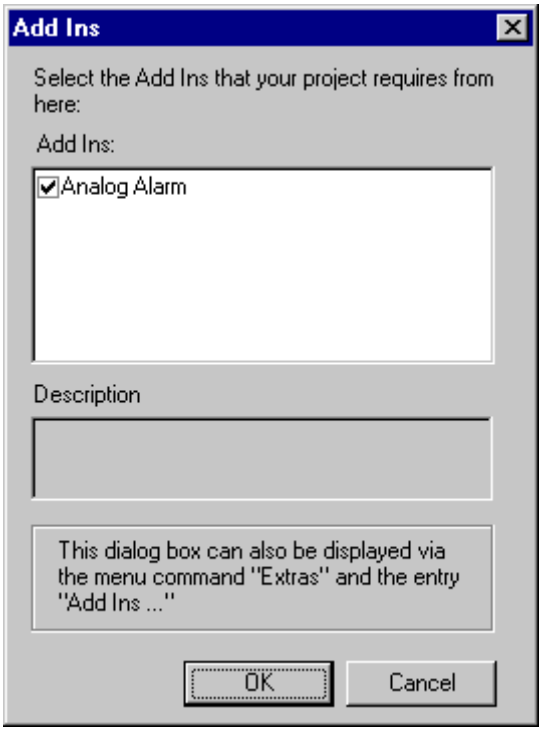
Step	Procedure: Creating the Tags
1	In this sample, three internal tags with the following names are created: The tag *G32i_temperature_value_00* corresponds to the actual value of the temperature, the tag *G32i_temperature_maxvalue_00* to the setpoint value of the temperature and the tag *U08i_power_value_00* to the heating capacity. 

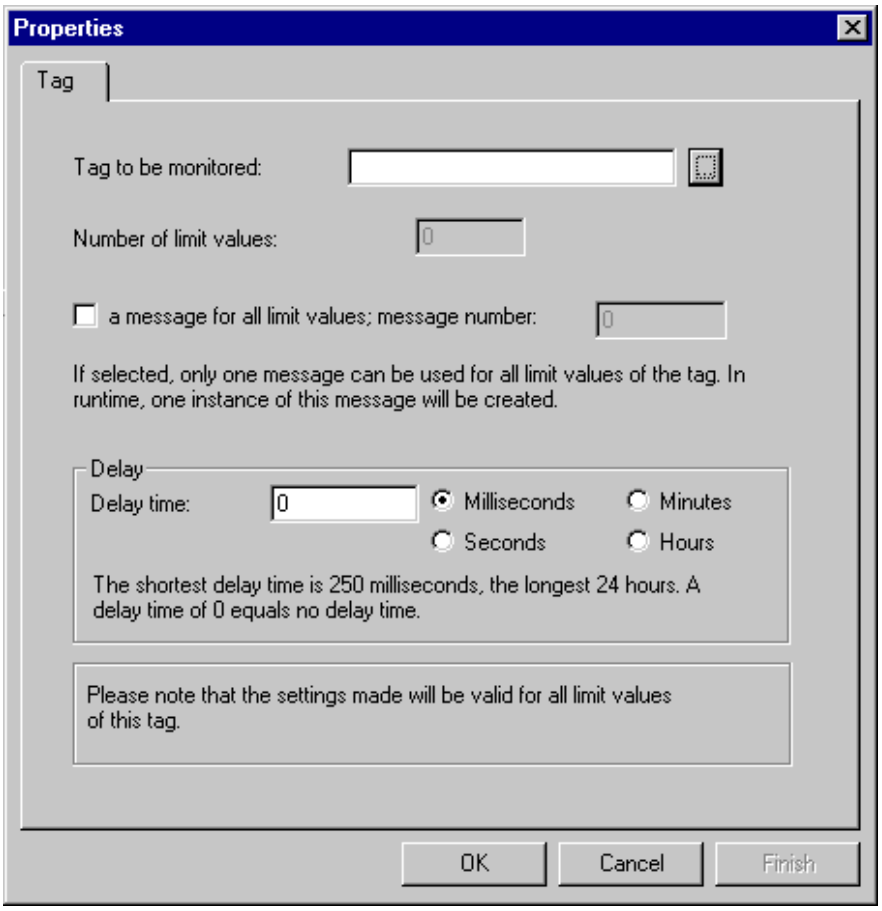
Creating a Process Value Archive

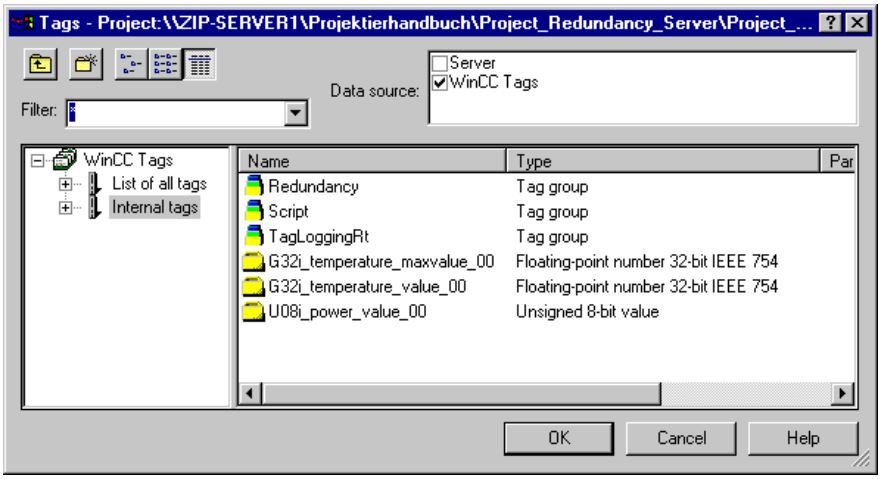
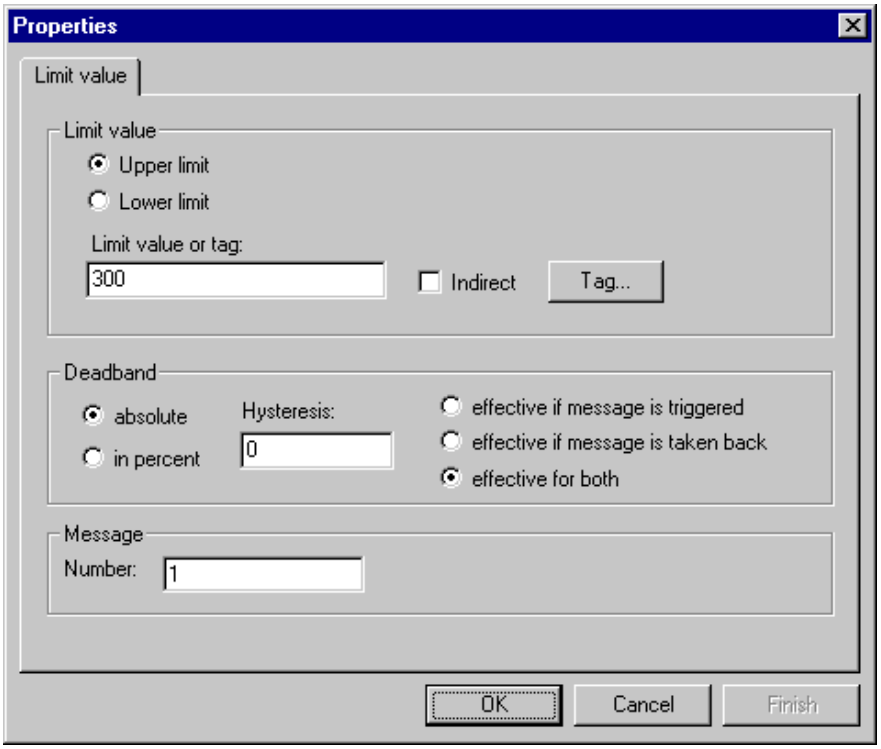
Step	Procedure: Creating a Process Value Archive
1	Open the *Tag Logging* editor.
2	Creation of a process value archive. For this purpose, the Archive Wizard is started via a R on *Archives*. 

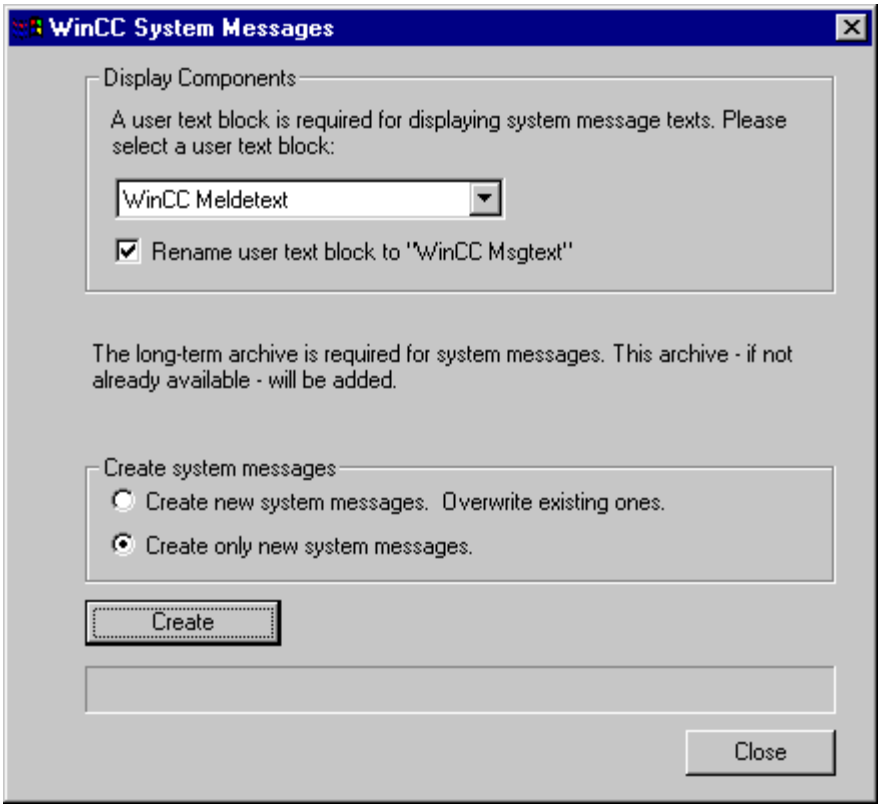
Step	Procedure: Creating a Process Value Archive
3	In this sample, the archive has been named *ProcessValueArchive_00*. The tags *G32i_temperature_value_00* and *G32i_temperature_maxvalue_00* are specified as the archive tags. 
4	Save and then exit the *Tag Logging* editor.

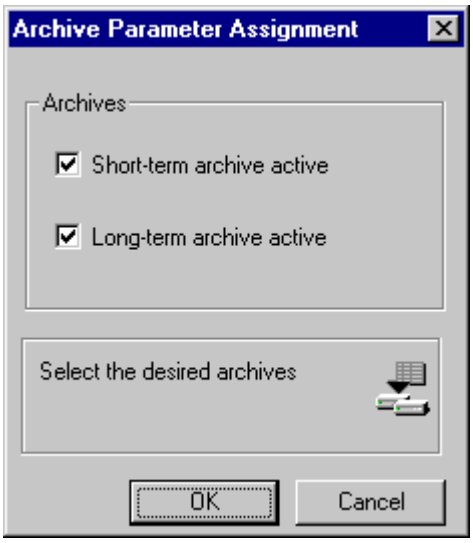
Configuring Alarm Logging

Step	Procedure: Configuring Alarm Logging
1	Open the *Alarm Logging* editor.
2	Creation of single messages. In the lower window of the *Alarm Logging* editor, the already configured messages are displayed. Via a  R, a new line can be added. In this sample, two different messages are required. The error type, message text and point of error must be changed correspondingly. ... | 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	Configuration of the limit value monitoring. If the *Limit Value Monitoring* (Analog Alarm) entry is not present, it must be loaded first. This is done via the *Options* → *Add Ins* menus in *Alarm Logging*. In the dialog displayed, the check-box for the *Limit Value Monitoring (Analog Alarm)* is selected. Close the dialog by clicking on *OK*. 

Step	Procedure: Configuring Alarm Logging
4	Via a R on the *Limit Value Monitoring entry and then selecting* → *New...* , the *Properties* dialog of the tag is accessed. In this dialog, a new tag for the limit value monitoring can be set. 
5	Via a on the button displayed below, the *Select Tag* dialog is opened. 

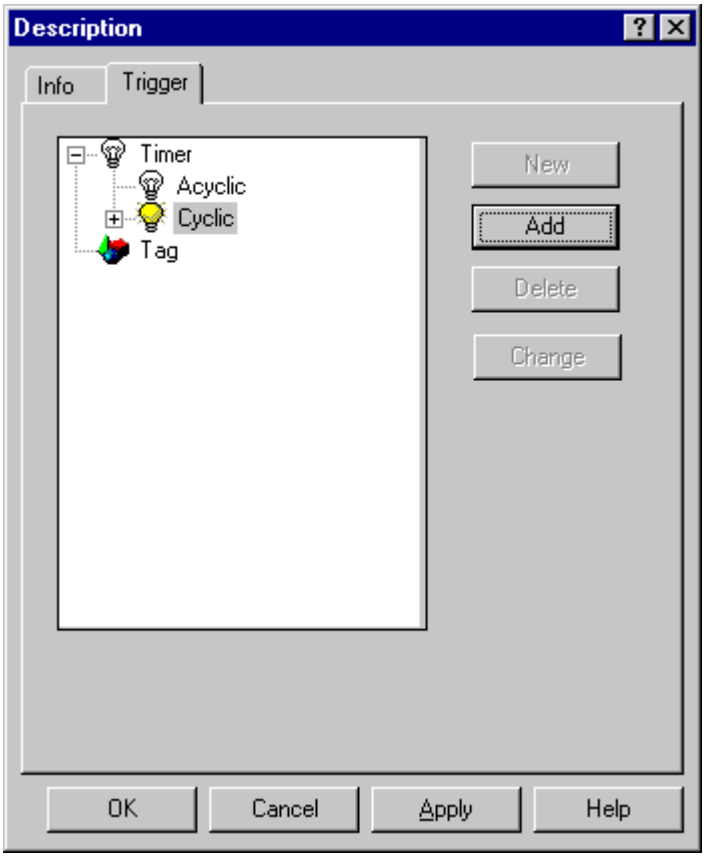
Step	Procedure: Configuring Alarm Logging
6	In the left window, the entry *Internal Tags* is selected. The right window will then list the corresponding tags. Select the desired tag. In the sample, this is the *G32i_temperature_value_00* tag. The dialog is closed with the *OK* button. 
7	The *Properties* dialog of the tag is closed with *OK* as well. The right window of *Alarm Logging* will then display the icon of the new tag to be monitored. Via a R on *G32i_temperature_value_00* → *New*, the *Properties* dialog of the limit value is accessed. In this dialog, a new limit value can be assigned to the tag. In this sample, the *Upper Limit* is set to *300* and the message number to *1*. The dialog is closed with the *OK* button. 
8	Following the previously described step, a second limit value is assigned to the tag. The *Upper Limit* is set to *700* and the message number to *2*.

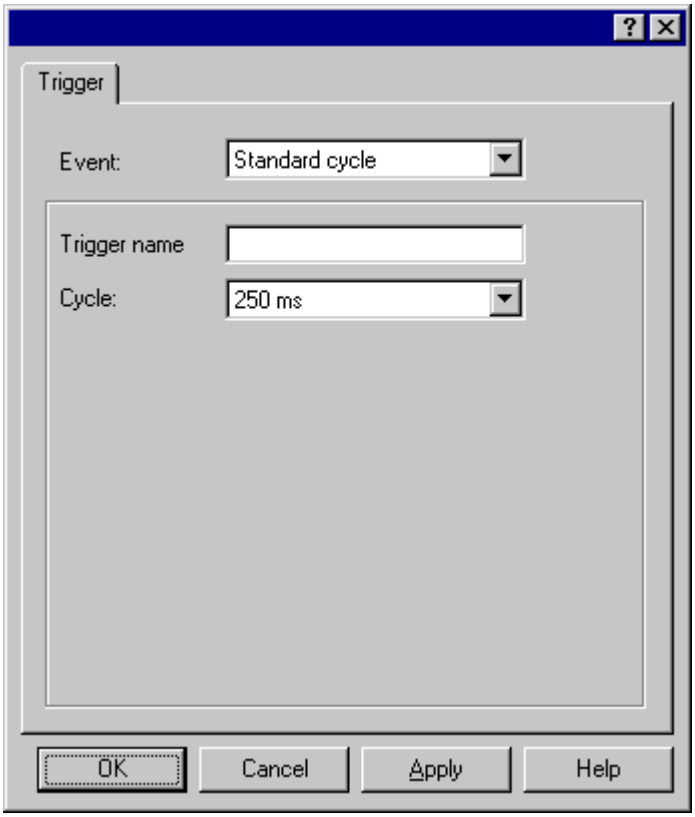
Step	Procedure: Configuring Alarm Logging
9	Creation of the WinCC system messages. Via the *Options* → *WinCC System Messages* menus, the *WinCC System Messages* dialog is accessed. Via the *Create* button, these WinCC system messages will be generated. The dialog is closed with *OK*. 

Step	Procedure: Configuring Alarm Logging
10	Activation of the short-term and long-term archives. Via a R on *Archives* → *Add/Remove*, the *Archive Parameter Assignment* dialog is opened. In this dialog, the *Short-Term Archive* and the *Long-Term Archive (Sequence Archive)* are activated. The dialog is closed with the *OK* button. 
11	Save and then exit the *Alarm Logging* editor.

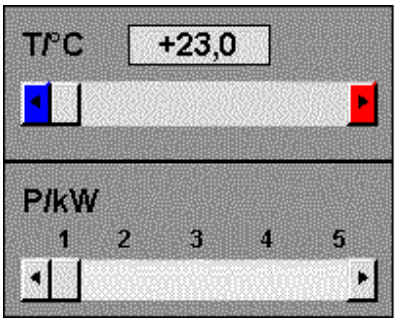
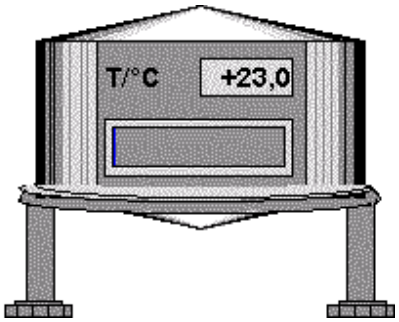
Creating a Global Action

Step	Procedure: Creating a Global Action
1	Open the *Global Script* editor.
2	Creation of a new global action. This is carried out via the *File* → *New Action* menus in the Global Script editor.
3	In this sample, a C-Action has been programmed that simulates an e-function as a trend. The difference *dDelta* between the setpoint temperature *dTemp2* and the actual temperature *dTemp1* is computed. If this difference is positive, the trend increases. If it is negative, the trend drops. The heating capacity *nPower* defines, how fast the temperature reaches the setpoint value.
4	Via *Edit* → *Compile*, the C-Action is compiled.

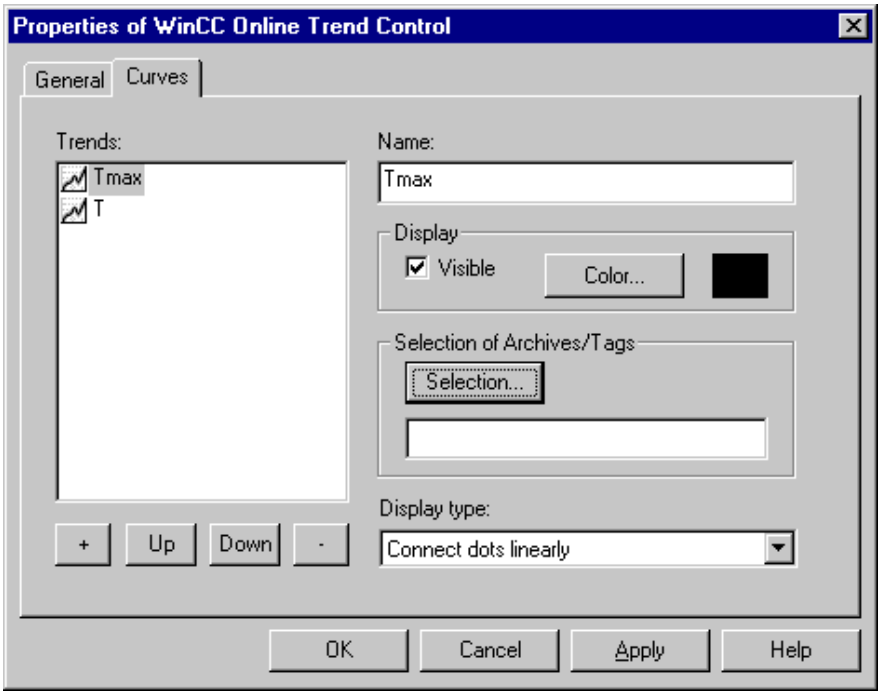
Step	Procedure: Creating a Global Action
5	Via *Edit* → *Info*, the *Description* dialog is opened. In the *Trigger* tab, a *Cyclic Timer* is selected in this sample. Via the *Add* button, the dialog for changing the trigger is displayed. 

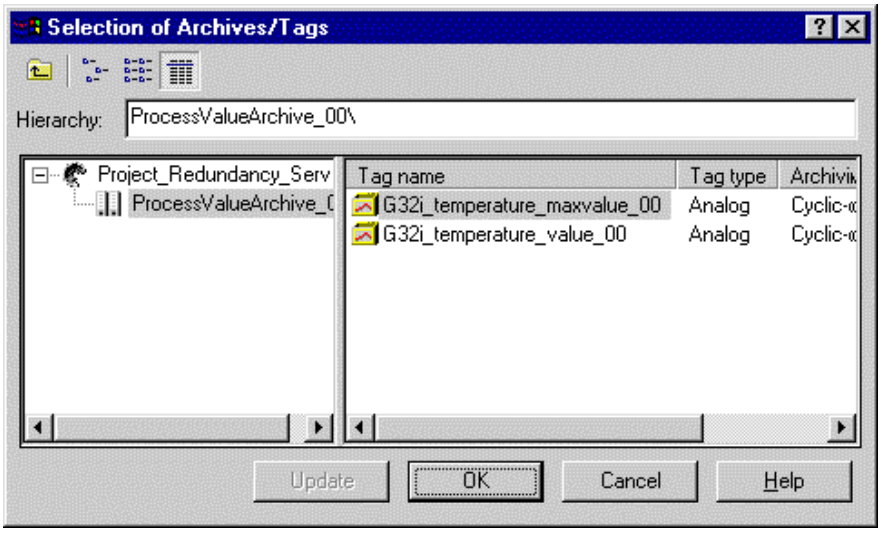
Step	Procedure: Creating a Global Action
6	The cycle time is set to *250 ms*. Both dialogs are closed with *OK*. 
7	Save and then exit the *Global Script* editor.

Configuring Objects

Step	Procedure: Configuring Objects
1	Create a new picture in the *Graphics Designer*. In the sample, this is the *red_3_chapter_01.PDL* picture. In this picture, various objects are connected to process tags.
2	The simulation of the input tags is implemented via a *Windows Object* → *Slider Object* each. In this sample, these are the *Slider Object1 (G32i_temperature_maxvalue_00)* and *Slider Object2(U08i_power_value_00)* that together with the *I/O Field1 (G32i_temperature_maxvalue_00)* represent the temperature control. In *I/O Field1*, the value of the setpoint temperature is displayed and can also be changed there. The output tag *(G32i_temperature_value_00)* is displayed in the oven. It consists of the *I/O Field2* and the *Bar1* objects. The update of these objects is set to *Upon Change*.  

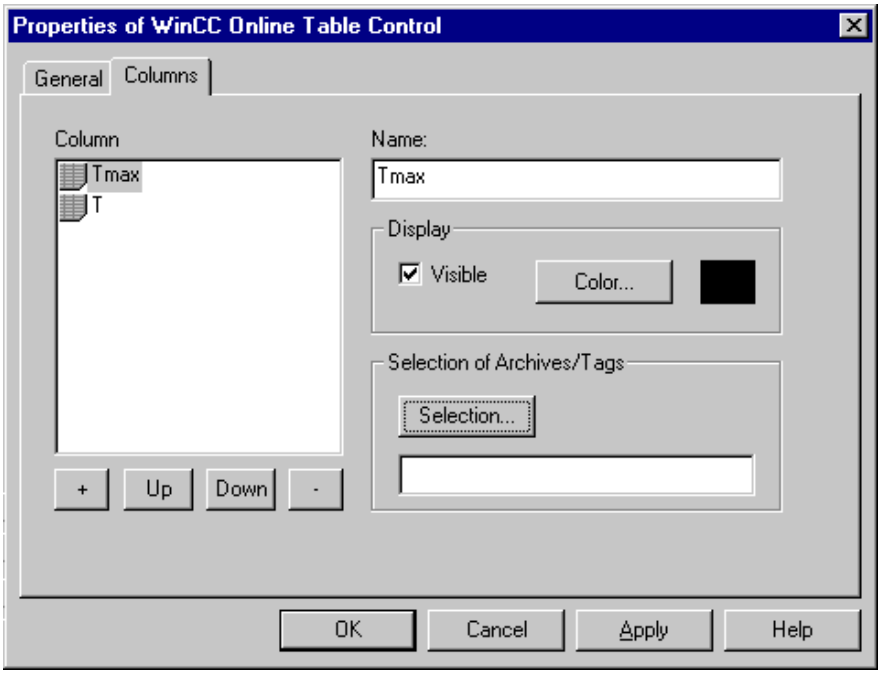
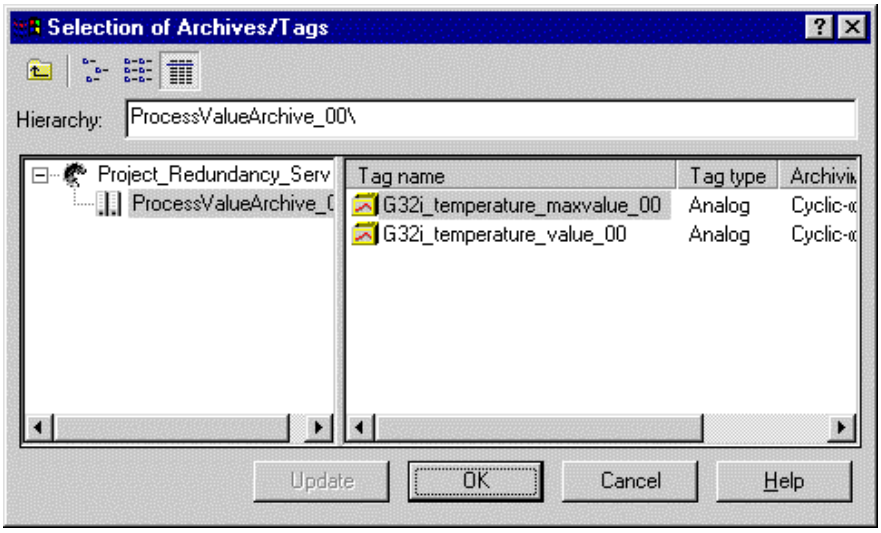
Configuring the Trend Windows

Step	Procedure: Configuring the Trend Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *mcs_3_chapter_02.PDL* picture. In this picture, two temperature values are displayed using trend windows.
2	Creation of a *Trend Control* via *Control* → *WinCC Online Trend Control*. In the sample, this is the *TlgOnlineTrend1* object. The dialog *WinCC Online Trend Control Properties* is displayed. In the *Trends* tab, a new trend is added by clicking on the + button. *Trend 1* is renamed to *Tmax* and *Trend 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 

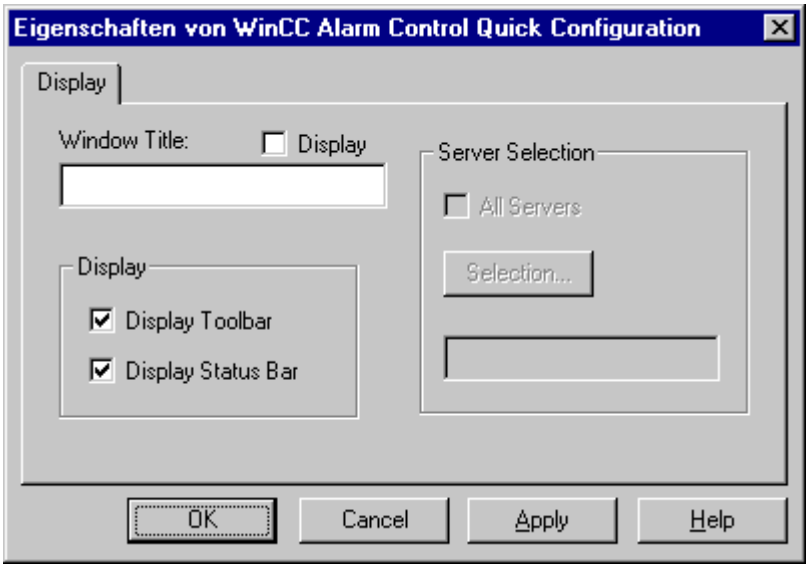
Step	Procedure: Configuring the Trend Windows
3	The *Archive/Tag Selection* dialog is displayed. This dialog allows the selection of archives/archive tags. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag. 

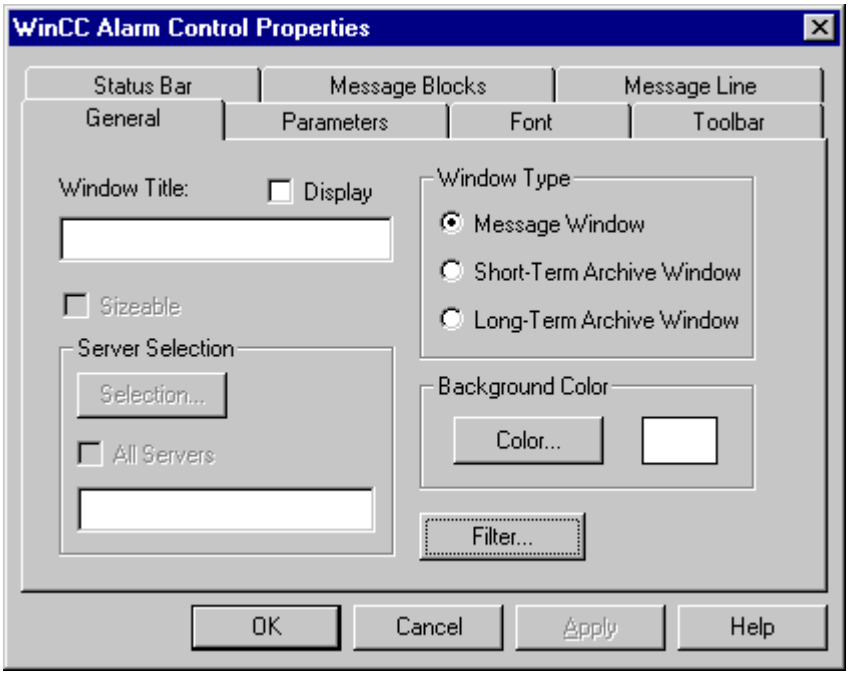
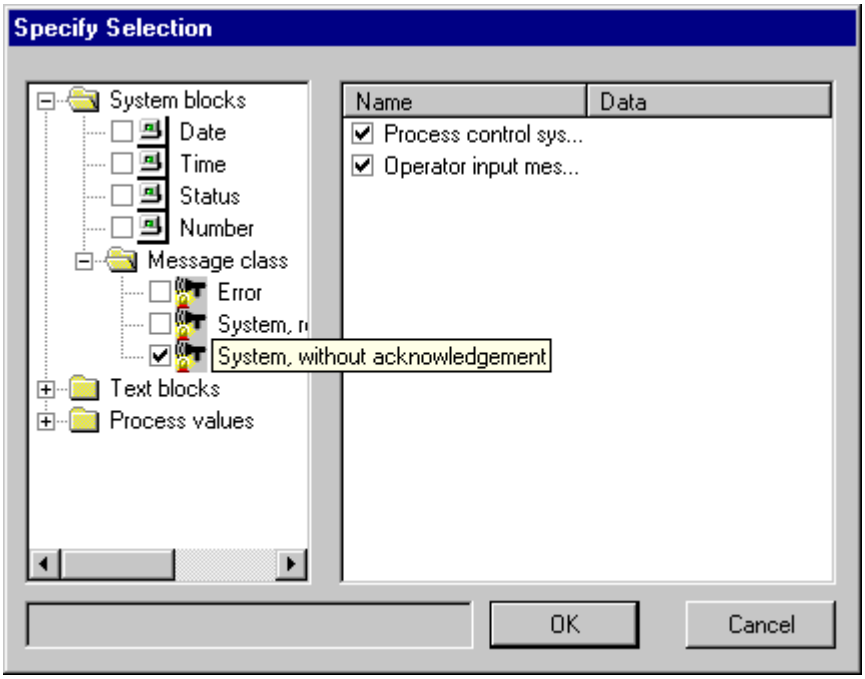
Configuring the Table Windows

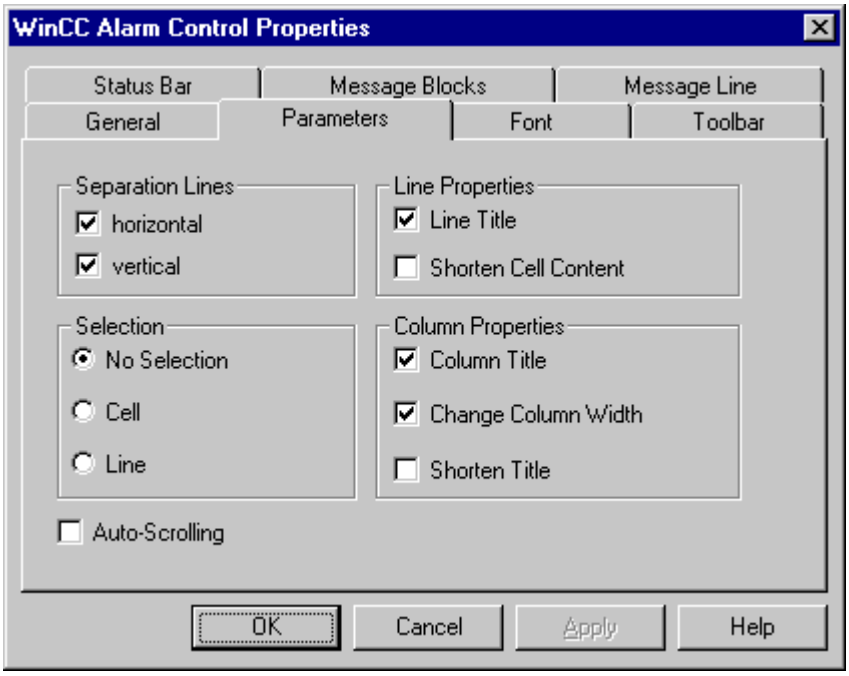
Step	Procedure: Configuring the Table Windows
1	In the same picture (*red_3_chapter_02.PDL*), two temperature values are displayed using table windows.

Step	Procedure: Configuring the Table Windows
2	Creation of a *Table Control* via *Control* → *WinCC Online Table Control*. In the sample, this is the TlgOnlineTable1 object. The dialog *WinCC Online Table Control Properties* is displayed. In the *Columns* tab, a new column is added by clicking on the + button. *Column 1* is renamed to *Tmax* and *Column 2* to *T*. In the *Archive/Tag Selection* field, a dialog for the selection of the desired archive tag is opened via the *Select* button. 
3	The *Archive/Tag Selection* dialog is displayed. This dialog allows the selection of archives/archive tags. In this sample, *Tmax* is connected with the *G32i_temperature_value_00* tag and *T* with the *G32i_temperature_maxvalue* tag. 

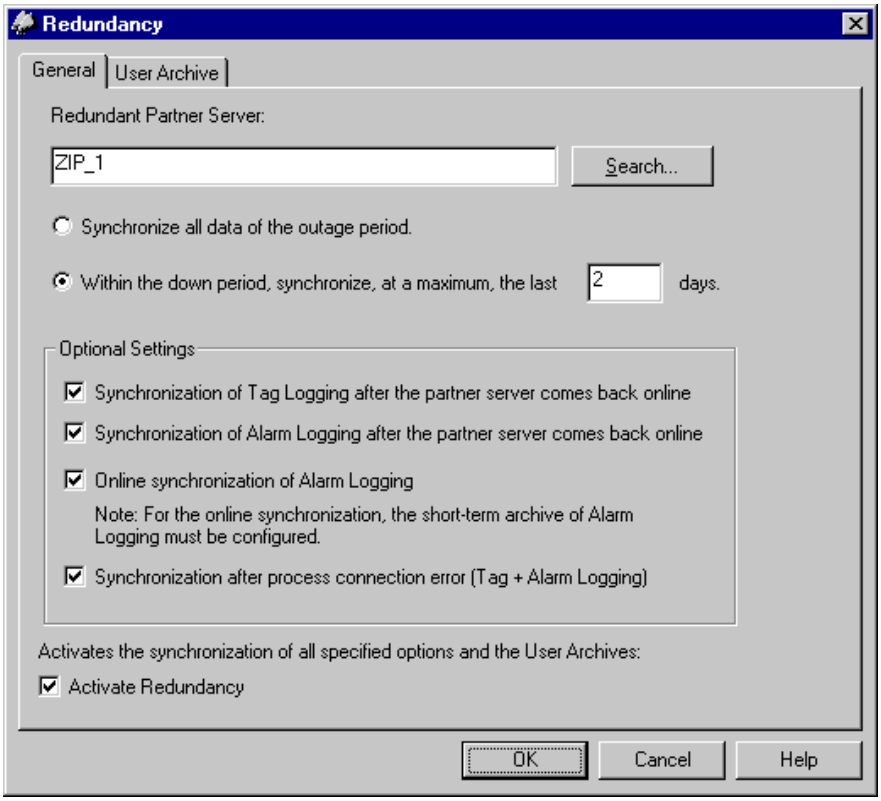
Configuring the Message Windows

Step	Procedure: Configuring the Message Windows
1	Creation of an additional picture in the *Graphics Designer*. In the sample, this is the *red_3_chapter_03.PDL* picture. In this picture, the configured messages are output using message windows.
2	Creation of a *WinCC Alarm Control* via *Control* → *WinCC Alarm Control*. In the sample, this is the *CCAlgWinCtrl1* object. The dialog *WinCC Alarm Control Properties - Quick Configuration* is displayed. This dialog is closed with *OK*. 
3	Following the steps just described, an additional *WinCC Alarm Control* is configured. In the sample, this is the *CCAlgWinCtrl2* object.

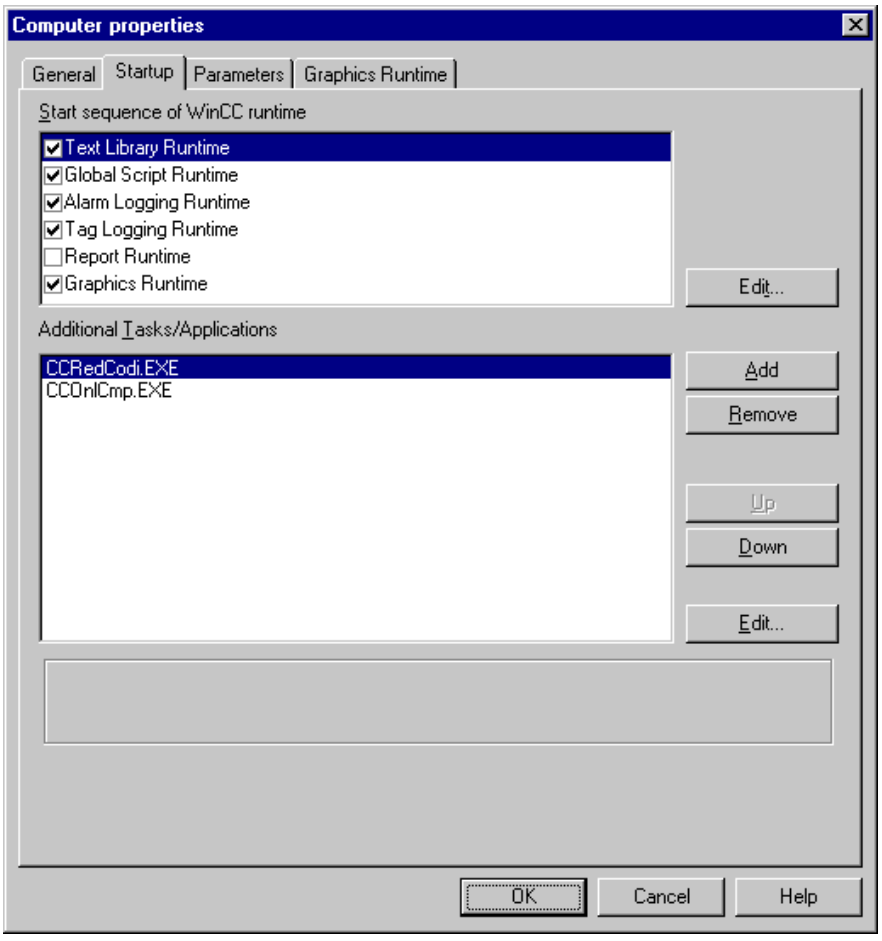
Step	Procedure: Configuring the Message Windows
4	Via a D on the *WinCC Alarm Control* just created, the *WinCC Alarm Control Properties* dialog will be displayed. In the *General Information* tab under *Window Type*, the *Short-Term Archive Window* is selected. Via the *Select* button, the *Define Selection* dialog is accessed. 
5	In the dialog displayed, the messages can be specified which are to be displayed. The *System, without Acknowledgment* check-box is selected. Close the dialog by clicking on *OK*. 

Step	Procedure: Configuring the Message Windows
6	In the Parameters tab, *Auto-Scrolling* is activated. Exit the dialog by clicking on *OK*. 

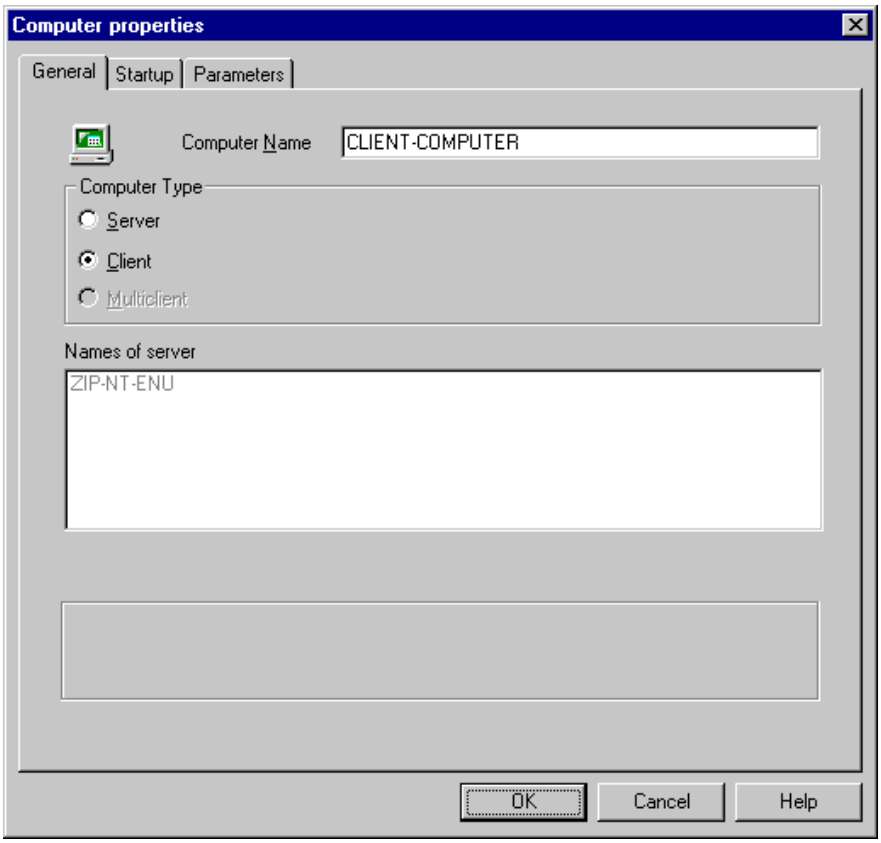
Creating the Redundancy

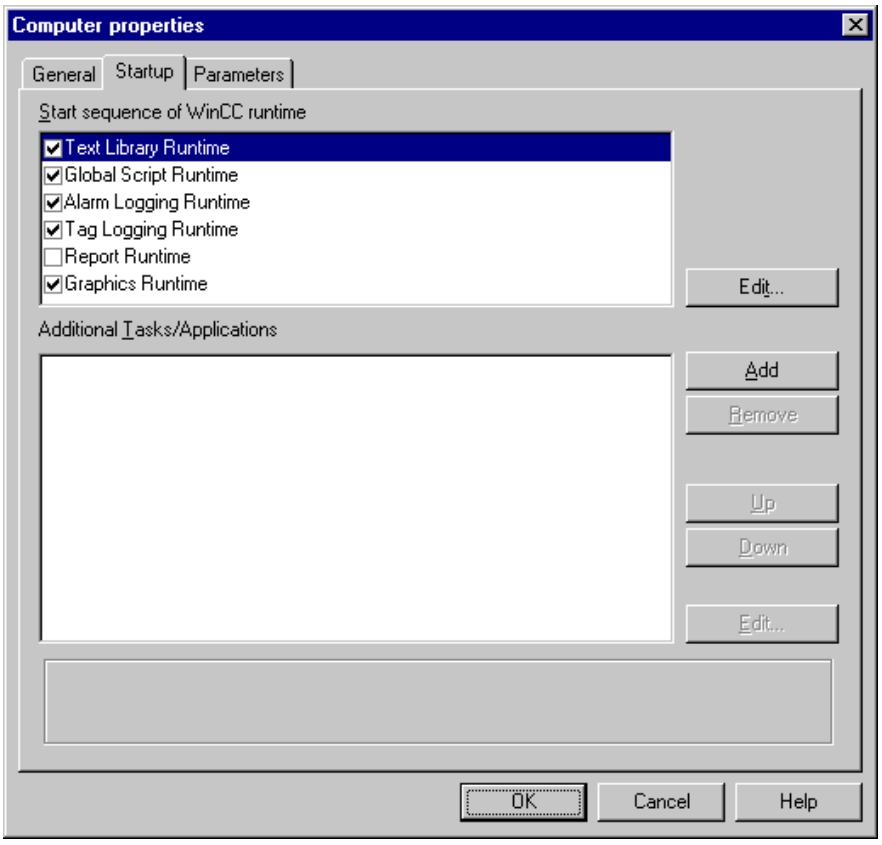
Step	Procedure: Creating the Redundancy
1	*Via a* RR on *Redundancy* → *Open* on the *left side of the WinCC Explorer, the Redundancy dialog is opened.*
2	In the *General* tab, the name of the redundant partner server is entered in the *Redundant Partner Server* field. Via the *Search* button, the *Select Redundancy Partner* dialog is displayed which facilitates the search for the corresponding computer. The *Activate Redundancy* check-box is selected. In the *Optional Settings* field, all check-boxes are selected. Exit the dialog by clicking on *OK*. 

Setting the WinCC Runtime Startup Properties of the Server

Step	Procedure: Setting the WinCC Runtime Startup Properties
1	Via a on the *Computer* entry on the left side of the *WinCC Explorer*, the computer name will be displayed on the right.
2	Via a R on *Computer Name* → *Properties*, the *Computer Properties* dialog is displayed. In the *Startup* tab, the following settings are made. Exit the dialog by clicking on *OK*. 

Adding the Client

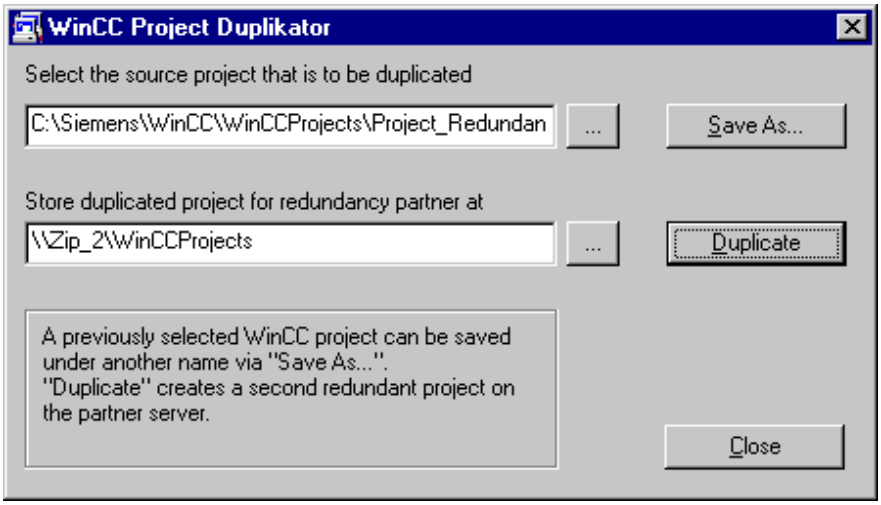
Step	Procedure: Adding the Client
1	Adding a new computer. Via a R on *Computer* → *New Computer* on the left side of the *WinCC Explorer*, the *Computer Properties* dialog is displayed.
2	In the *General Information* tab under *Computer Name*, the name of the corresponding client computer is specified. As the *Computer Type*, *Client* is specified. 

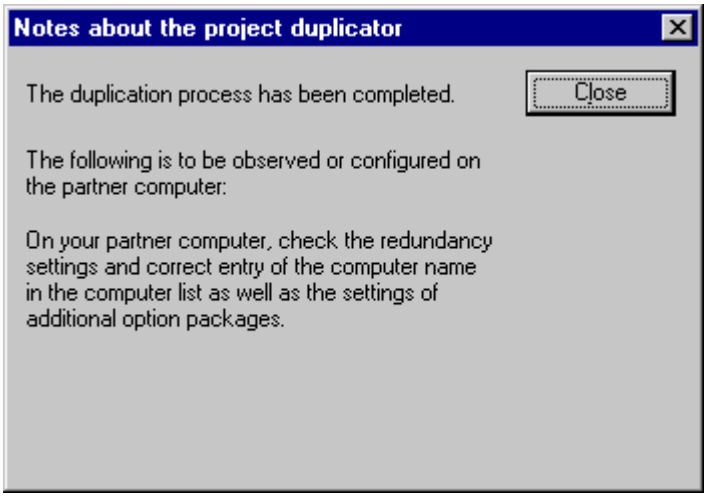
Step	Procedure: Adding the Client
3	In the *Startup* tab, the following settings are made. Exit the dialog by clicking on *OK*. 

Duplicating the Project

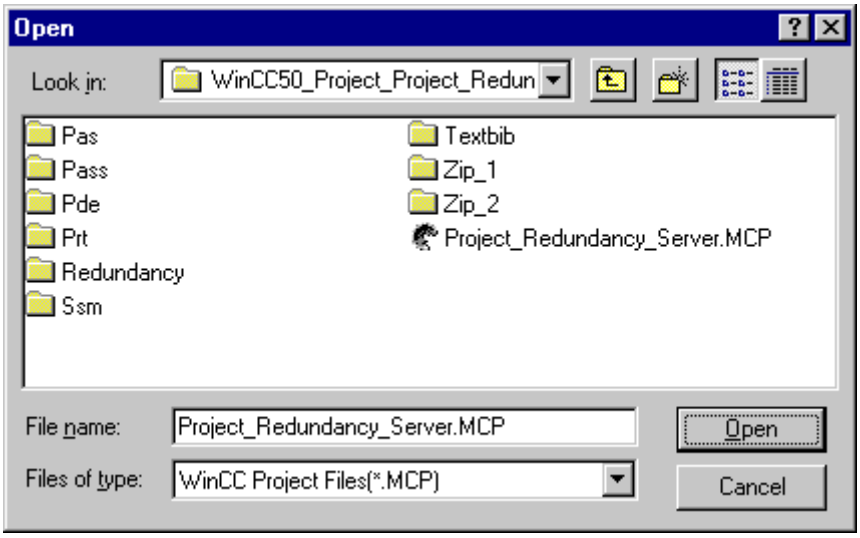
On both servers, functionally identical projects must be configured. The WinCC Project Duplicator makes is possible to copy all data pertaining to a project to the redundant partner.

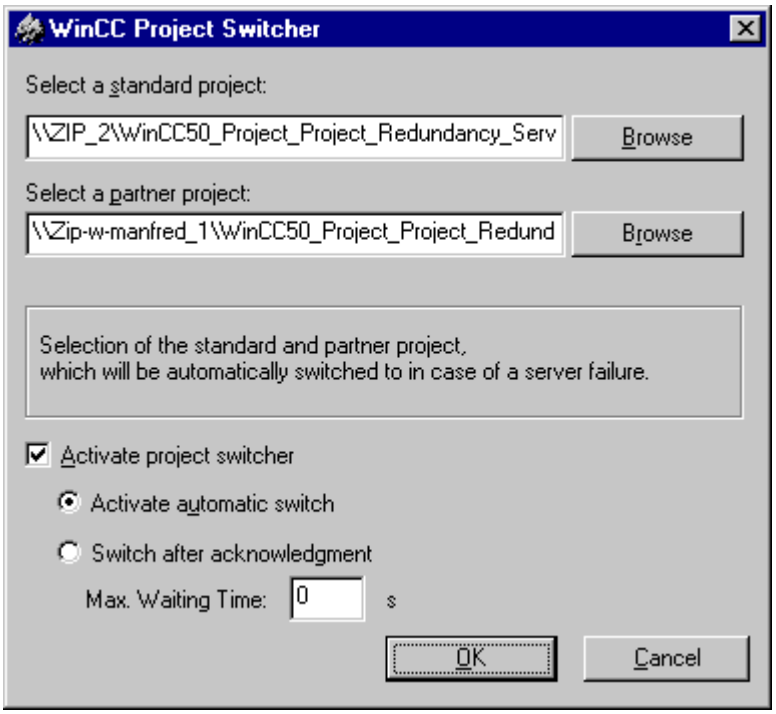
The WinCC Project Duplicator automatically creates the redundant partner project. All associated project data (pictures, scripts, archives, etc.) is copied to the target computer and all settings are made, to make the target computer ready for Redundancy.

Step	Procedure: Duplicating the Project
1	The WinCC Project Duplicator is opened. It is started via *Start* → *Simatic* → *WinCC* → *Tools* → *Project Duplicator*.
2	The *WinCC Redundancy Project Duplicator* dialog will be displayed. In the *Select the source project that is to be duplicated* input field, the source project is selected. In the *Store duplicated project for redundancy partner at* input field, the target computer including the folder of the target project is specified. Via the *Duplicate* button, the duplication process is started. 

Step	Procedure: Duplicating the Project
3	After the duplication process has completed, the *Notes about the Project Duplicator* dialog is displayed. This dialog is closed with *OK*. 

Setting the Project Switcher on the Client

Step	Setting the Project Switcher on the Client
1	WinCC is opened on the client computer. The WinCC Explorer is started via *Start* → *Simatic* → *WinCC* → *Windows Control Center*.
2	The *WinCC Explorer* will be displayed. Via the *File* → *Open* menus, the *Open* dialog for selecting a WinCC project is displayed. Under Network Neighborhood, the computer is selected on which the server project *Project_Redundancy_Server* has been configured. The *WinCC50_Project_Project_Redundancy_Server* folder contains the file *Project_Redundancy_Server.mcp*. This file is selected and opened via the *Open* button. 

Step	Setting the Project Switcher on the Client
3	The WinCC Project Switcher is opened. It is started via *Start* → *Simatic* → *WinCC* → *Tools* → *Project Switcher*. In the *Select Default Project* input field, the default project of the of the associated server is specified to which the client is connected by default. The *Search* button aids in the project selection. In the *Select Partner Project* input field, the default project of the of the associated server is specified to which the switch is made in the case of a failure. The *Search* button aids in the project selection. The *Activate Project Switcher* check-box and then the *Switch Automatically* entry are selected. Exit the dialog by clicking on *OK*. 

4.3 Description of the WinCC Projects

Activate runtime on both server projects. Following that, runtime is automatically activated on the client project.
During the startup of the server computers, the Redundancy component determines whether the partner server has already been activated. If the partner server has already been activated, then a slave identification will be set on the server computer. If the partner server has not yet been activated during startup, then a master identification will be set on the server computer. If a network connection error occurs between the servers of the partner server is deactivated, the master identification will be reset. To mark the server computer as the master, the internal WinCC tag @RM_MASTER is set. If the server computer is the slave, the tag @RM_MASTER is reset. The @RM_MASTER_NAME tag contains the name of the server computer, e.g. Server 1. These tags can be evaluated and changed by other applications or scripts.
The Redundancy component only sets the tags mentioned above. Both server are always completely equal.

4.3.1 Server Project

After the appearance of the overview picture, the plant picture can be accessed via the button displayed above.

Via the button displayed above, you can switch among the individual pictures.

Via this button, you can go back to the overview.

Plant Picture

In the plant picture, an oven with a temperature control is displayed. With this temperature control, a temperature can be preset. The temperature in the oven rises, until the preset value has been reached. With the power control, the heating capacity can be specified. This value influences the speed with which the oven temperature rises.

Trend and Table Windows

In the next picture, the trend and table windows are displayed. The trend window depicts the progress of the preset temperature (setpoint value) and the oven temperature (actual value). Both of these values are also displayed in the table window.

Message Window

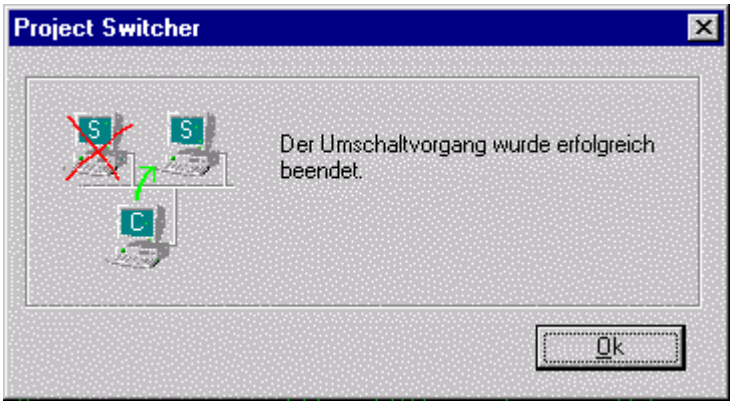
The next picture displays both message windows. If the oven temperature exceeds the value of 300, a warning is generated and displayed in the top message window. If the value of 700 is exceeded, an alarm is generated and displayed in the bottom message window.
In the bottom message window, system messages are displayed that are output by Redundancy.
These can be at:

- Process Connection Error
- Synchronization of Alarm Logging
- Synchronization of Tag Logging

Switch Process during the Failure of the Master Server

If the network connection to the master server is interrupted, the following dialog will be displayed:

The client project is then closed automatically and switched to the partner server. Following that, runtime is again activated on the client. After the switch process is complete, the following dialog is displayed:

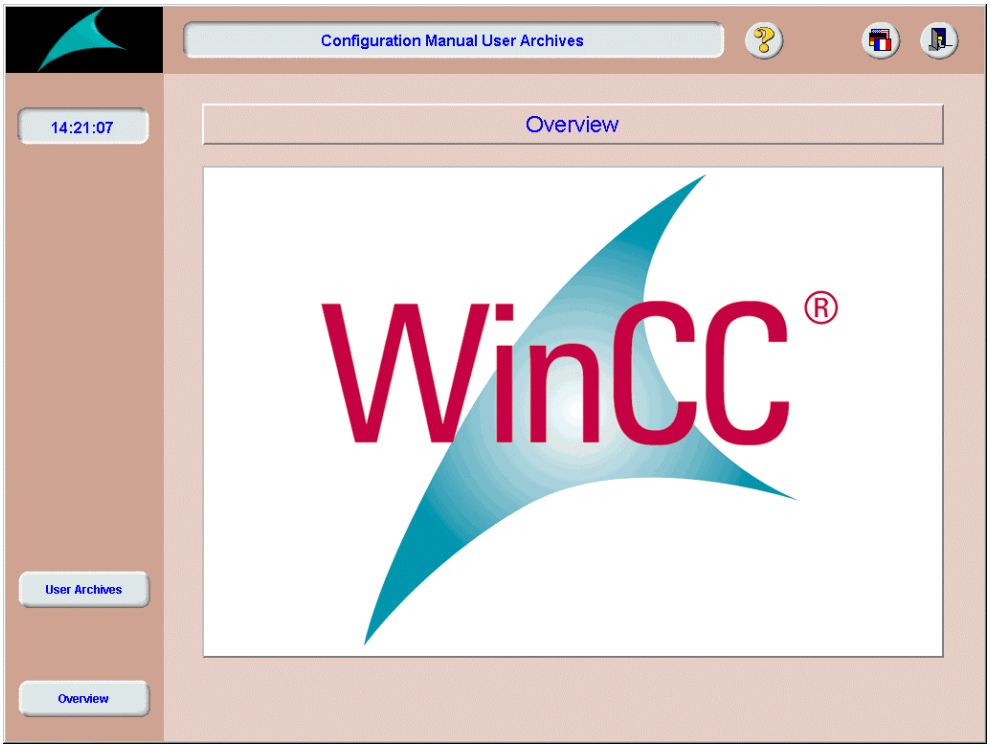


5 User Archives (Project_UserArchive)

The project created in this chapter can also be copied directly from the online document to your hard drive. By default, it will be stored to the *C:\Configuration_Manual* folder.


Project_UserArchive

This WinCC project is used to illustrate the operation of the User Archives editor. The configuration procedure and application of the User Archives is explained.
The following displays the start page of the Project_UserArchive project.



5.1 User Archives

In runtime, the samples pertaining to this topic are accessed in the *Project_UserArchive* project by selecting the button displayed above using the . The samples are configured in the *ex_3_chapter_01.pdl* to *ex_3_chapter_01c.pdl* pictures.

General Information

The User Archives make available a database for storing data in a user-defined table structure.
This data can be transferred to a lower-level PLC using various methods. For the communication with the PLC, WinCC tags as well as WinCC raw data tags can be used.
The User Archives can also be configured without communication.

- At this time, four data types for the individual data fields are available. These are:ž XE "Data:Transfer to the PLC"
- Integer
- Double Integer
- Character String
- Date/Time

Using an OLE Control, the data of the User Archives can be displayed in runtime in tabular form. Via this OLE Control, data of the User Archives can be edited, written to the PLC, read from the PLC, exported to a CSV file and imported from a CSV file. The data displayed can be filtered and sorted according to certain criteria and conditions.
The OLE Control comes with a configurable toolbar and status bar. The individual table columns can be assigned various display formats.

Via the application of views, data from various archives can be displayed combined.

5.1.1 Creation of User Archives (ex_3_chapter_01.PDL)

Task Definition

An archive is to be created, whose data records consist of five integer fields as well as a text field for recording the data record name. The data of the archive is to be made available to the entire project record by record.

Implementation Concept

To archive the data, an archive is created in the *User Archives Editor*. The communication of the archive is configured using WinCC tags. In this archive, six data fields of the required field types are created. A tag (in this sample, an *internal tag*) is assigned to each data field, which allows the archive to communicate with the remaining project.

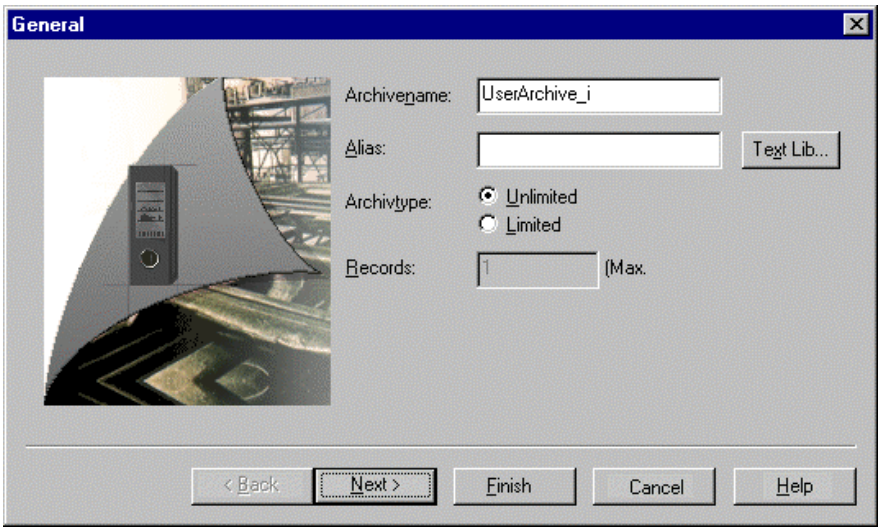
Note:

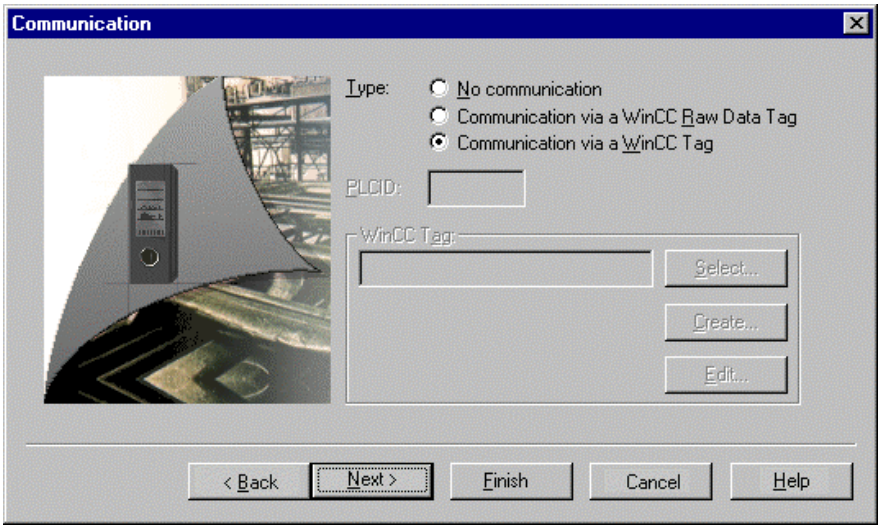
The following sample is a general description of the procedure performed to configure a User Archive. No connection to a lower-level PLC is established, instead internal WinCC tags are used to demonstrate the operation.

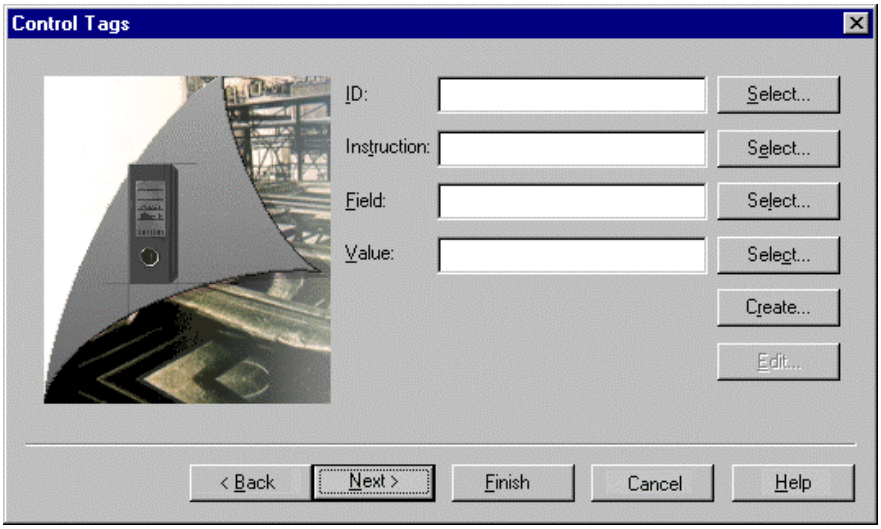
Samples with completely configured connections can be found in the following chapters.

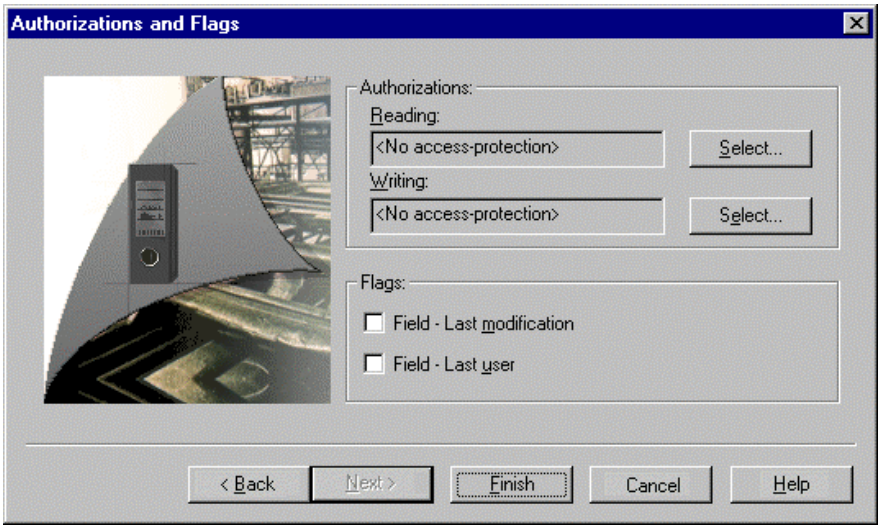
Creation of an Archive

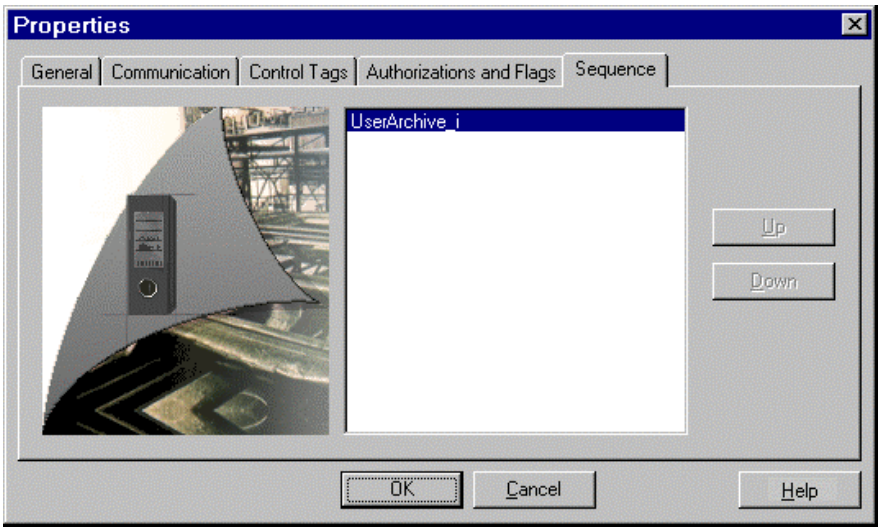
Step	Procedure: Creation of an Archive
1	Creation of one internal tag for each data record of the archive. The communication between the archive and the remaining system is carried out via these tags. In the sample, the tags *S32i_ex_UAi_01* to *S32i_ex_UAi_05* of the *Unsigned 32-Bit Value* type were used. In addition, a tag of the *Text Tag 8-Bit Character Set* type is created for storing the data record name. In the sample, this is the *T08i_ex_UAi_01* tag.
2	Open the User Archives Editor. This is done from the WinCC Explorer via a R on the User Archives entry and then selecting Open from the pop-up menu. 
3	Creating a new archive. Via a R on the *Archives* entry and then selecting *New Archives* from the pop-up menu, a Wizard is started. This Wizard guides the user through the creation of a new archive. 

Step	Procedure: Creation of an Archive
4	Filling out the *General Information* page of the Wizard. In here, general information about the archive to be created is provided. The *Archive Name* must be entered. The archive name must not exceed a length of 20 characters. No keywords of the ANSI-C programming language or the SQL database query language must be used. In the sample, the name *UserArchive_i* is entered in the *Archive Name* field. In the Alias field, a text can be entered describing the archive in greater detail. This field accepts all characters and numbers. Filling out this field is optional. In the sample, the Alias field is left blank. Furthermore, the Archive Type must be selected. If Limited is selected, then the maximum number of Data Records, which the archive can contain, must be specified. In this sample, Unlimited is set as the Archive Type. Continue to the next page by clicking on *Next*. 

Step	Procedure: Creation of an Archive
5	Filling out the Communication page of the Wizard. In here it is specified, how the archive to be created communicates with the rest of the project. If the *Communication via WinCC Raw Data Tag* is selected, a WinCC raw data tag must be specified on this page. With this type of communication, a complete data record is transferred using the set tag. Furthermore, a *PLCID* must be entered, which can contain a maximum of eight characters. Via this ID, a raw data block sent by the PLC can be assigned to a certain archive. If the *Communication via WinCC Tag(s)* is selected, a WinCC tag must be assigned to each archive data field. This is carried out during the creation of the corresponding data fields. Furthermore, there is the possibility to configure an archive with the option *No Communication*. In this sample, the *Type Communication via WinCC Tag(s)* is selected. Continue to the next page by clicking on *Next*. 

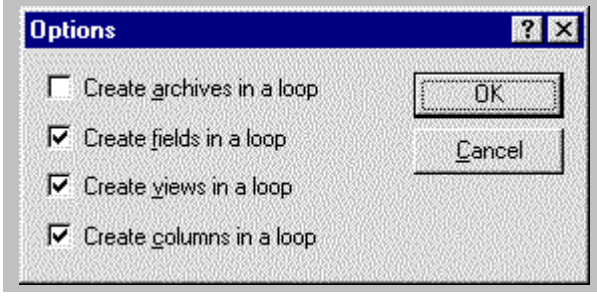
Step	Procedure: Creation of an Archive
6	Filling out the *Control Tags* page of the Wizard. In here, four internal or external WinCC tags can be set. Via these tags, the archive can be controlled in runtime. In this sample, the control tags option is not used. The sample Working with Control Tags (ex_3_chapter_012.PDL) contains a detailed description about this topic. Continue to the next page by clicking on *Next*. 

Step	Procedure: Creation of an Archive
7	Filling out the *Rights and Flags* page of the Wizard. In the Rights field, there is the option of setting authorization levels for the access types Read and Write. Via the button displayed below, a dialog containing a list of all authorization levels created with the User Administrator editor is opened. From this list, a desired authorization level can be selected. The chapter *pictu_3_chapter_02.pd* contains a detailed description about the configuration of users and authorization levels with the *User Administrator* editor. In the *Flags* field, predefined columns can be inserted into the archive. A column storing the date of the last access of the data record is available. Furthermore, a column storing the user name - who last accessed the data record - is available. In this sample, no settings are made on this page. This page of the Wizard is completed by clicking on *Finish*. 

Step	Procedure: Creation of an Archive
8	The just created archive will now be displayed in the tree structure under the *Archives* entry.  Via a R on this entry of the new archive *UserArchive_i* and then selecting *Properties* from the pop-up menu or by clicking on the properties button on the tool bar while the entry is selected, the properties dialog of the archive is opened. This dialog displays all pages of the Wizard and the entries made in the form of tabs. In addition, a tab for setting the archive order is provided. At this phase of the configuration, however, only a single archive is present. 

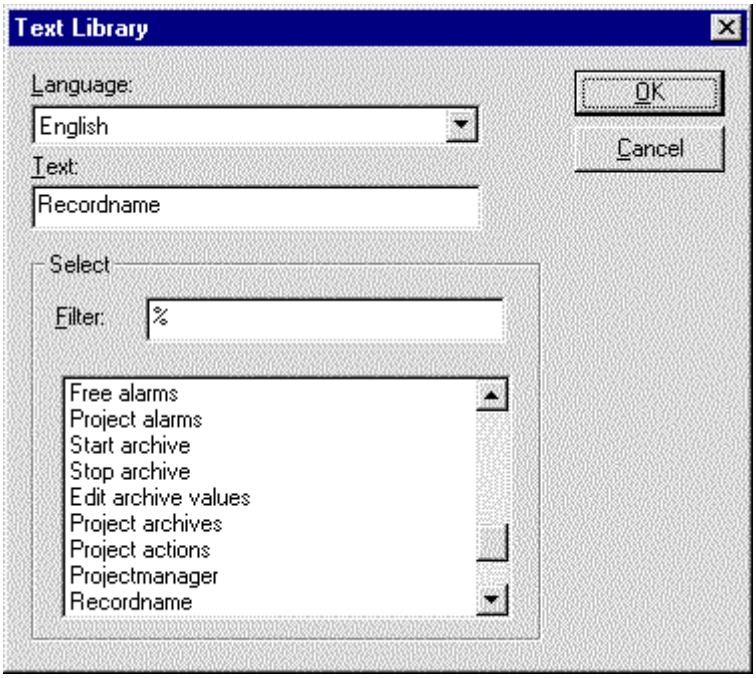
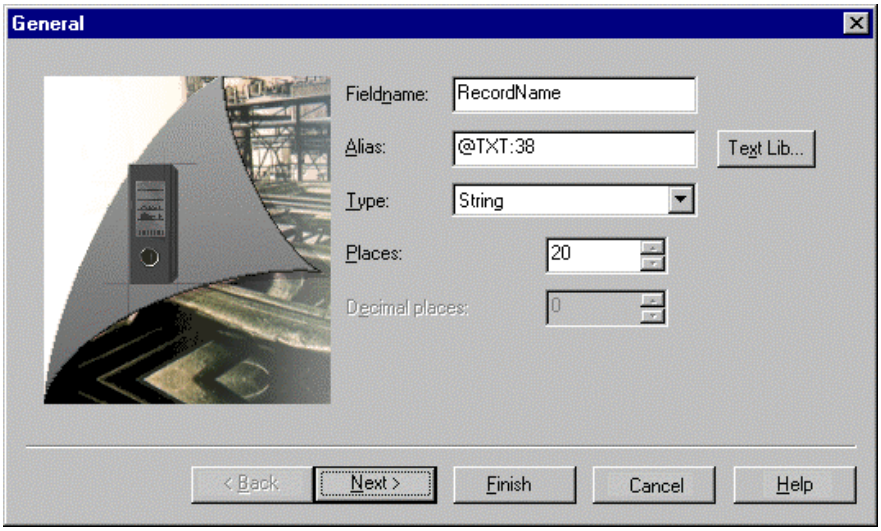
Note:

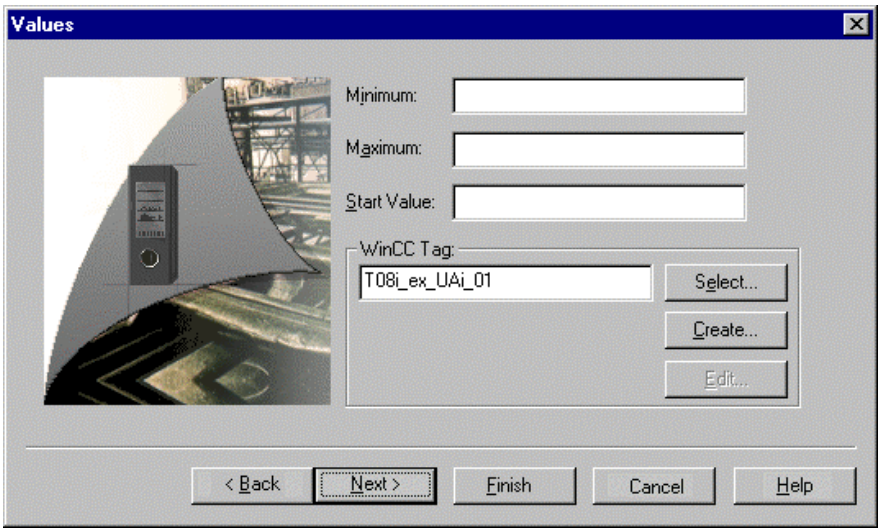
By default, the User Archives editor has been set to ask, whether a new archive is to be created after the completion of an archive. This behavior can be changed via the *Edit -> Options* menus by deselecting the corresponding check-box. The same applies to the fields, views and columns.

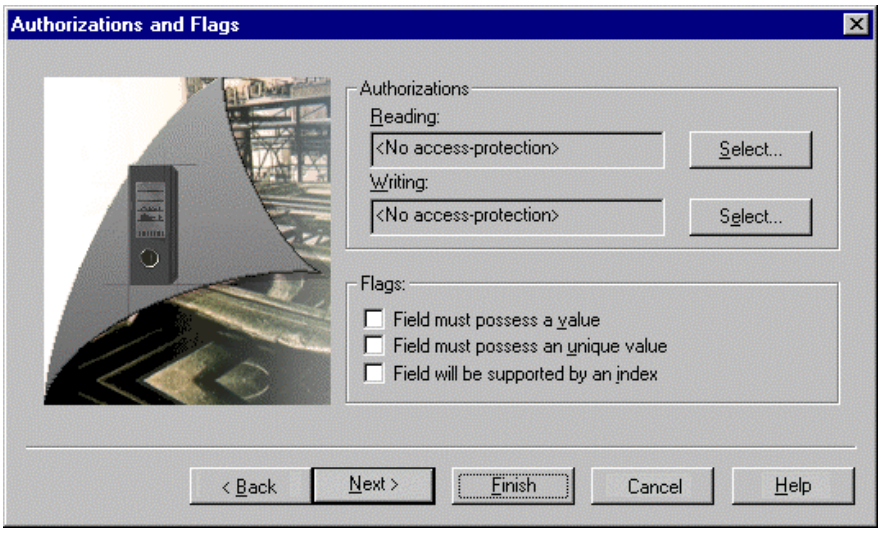


Configuration of Data Fields

Step	Procedure: Configuration of Data Fields
1	Creation of the individual fields of the archive. Via a R on the entry of the new archive UserArchive_i and then selecting New Field from the pop-up menu, a Wizard is started. This Wizard guides the user through the creation of a new field. 
2	Filling out the *General Information* page of the Wizard. In here, general information about the field to be created is provided. A *Field Name* must be entered. This name is subject to the same limitations as apply to the archive name. In this sample, the *Field Name RecordName* is used. In the Alias field, a name can be entered not subject to any limitations. The name entered in the Alias field, is used as the column title in runtime. Otherwise, the Field Name is sued as the column title. The Alias can be entered directly or a reference to a text in the Text Library can be made. The second option is useful, if projects are created in multiple languages. Via the button displayed below, a dialog for entering texts in the Text Library or for selecting texts from the Text Library is opened. Texts entered in the *Text* field are entered into the *Text Library* after pressing the *OK* button. As the *Alias*, a reference to this text will be used.

Step	Procedure: Configuration of Data Fields
	In the *Select* field, already existing texts can be searched for. In this case, a word or text segment, which is contained in the text searched for, is specified as the *Filter*. The *%* character can be used a placeholder. If only the *%* character is entered, the entire content of the *Text Library* is displayed in the list field below. Via a on the list field, its content is updated. In this sample, the *Text Field Name* is entered. The *Text Library* dialog is closed with *OK*.  Furthermore, the *Type* of the field must be selected. At this time, the types *integer*, *double integer*, *character string* and *date/time* are available. In this sample, the *Type Character String* is set. For this type, the *Length* of the character string must be set as well. In this sample, the value is left at *10*. Continue to the next page by clicking on *Next*. 

Step	Procedure: Configuration of Data Fields
3	Filling out the *Values* page of the Wizard. For numerical field types, a *Maximum*, *Minimum* and *Start Value* can be set. For text fields, a *Start Value* can be entered as the default. If a *Communication via WinCC Tag(s)* has been configured, a corresponding tag must be set for each field. In this sample, the previously created text tag *T08i_ex_UAi_01* is selected. Continue to the next page by clicking on *Next*. 

Step	Procedure: Configuration of Data Fields
4	Filling out the *Rights and Flags* page of the Wizard. In the *Rights* field, there is the option of setting authorization levels for the access types *Read* and *Write*. The authorization levels set here, refer to a single field. This allows the data of individual fields to be protected against unauthorized access. In the *Flags* field, various properties can be assigned to the field. The following properties can be assigned: • The *Field must contain a value*, which must not be zero. • The *Field must contain a unique value*, the values of this column must differ from each other. • The *Field is supported by an index*. In this sample, no settings are made on this page. This page of the Wizard is completed by clicking on *Finish*. 
5	In this sample, 5 additional fields are created. They are of the *Integer* type. The fields are named from *Pressure_01* to *Pressure_05*, the aliases are from *Pressure1* to *Pressure5*. As the tag, one of the previously created *Unsigned 32-Bit Value* tags is set for each field. The remaining field settings correspond to the first field. The newly created fields are displayed in the list view, if the archive *UserArchive_i* is selected in the left tree structure. 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

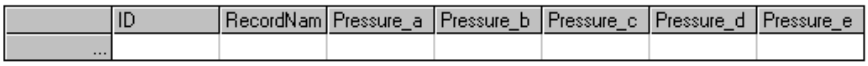
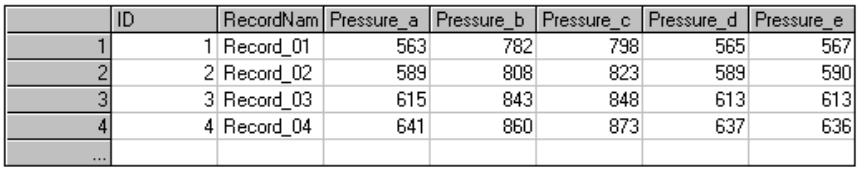
Step	Procedure: Configuration of Data Fields
6	Via a R on the entry of a field and then selecting *Properties* from the pop-up menu, the properties dialog of the field can be opened. This dialog displays all pages of the Wizard and the entries made in the form of tabs. In addition, a tab for setting the field order of the archive is provided. 
7	Save the configurations made. This can be done via the toolbar button displayed below or the *Project* → *Save* menus.  Make sure that the *User Archives Editor* is not in the mode for entering archive data, i.e. the button displayed below must not be pressed. Only then, can the configuration data be transferred to the database. 

5.1.2 Data Entry (ex_3_chapter_01.PDL)

Task Definition

This sample is based on the previous sample Creation of User Archives (ex_3_chapter_01.PDL). The previously created archive is to be supplied with data. In this case, the options of directly entering data in the User Archives editor as well as the option of importing data from an external file are to be utilized.

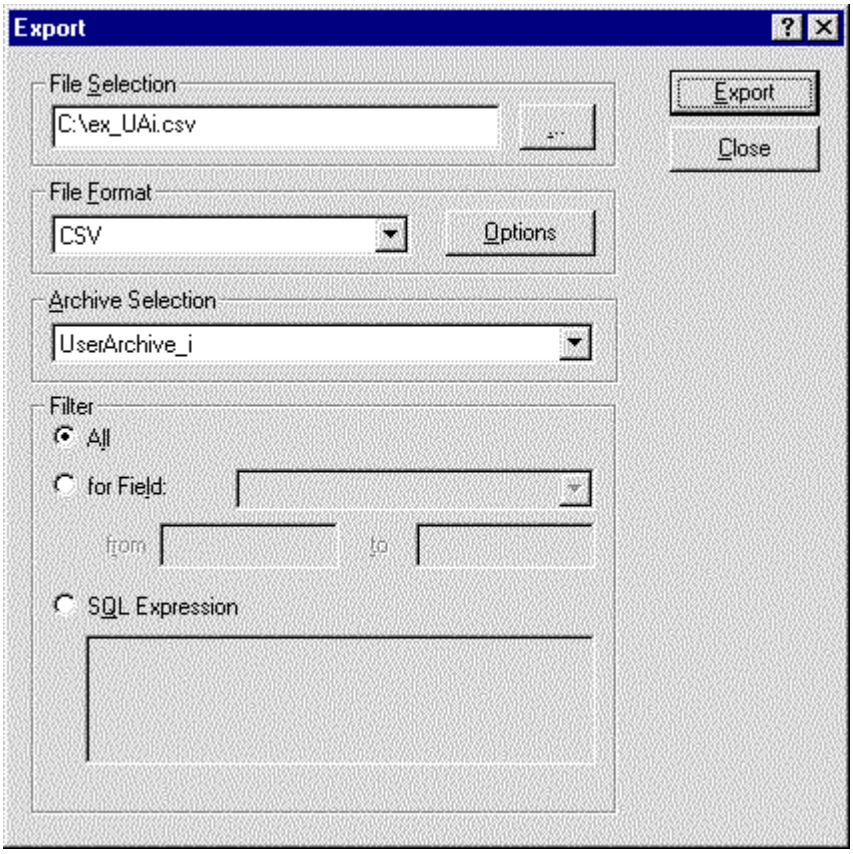
Data Entry in the User Archives Editor

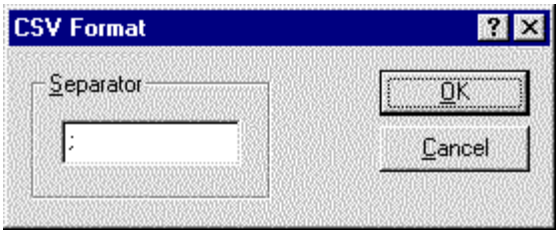
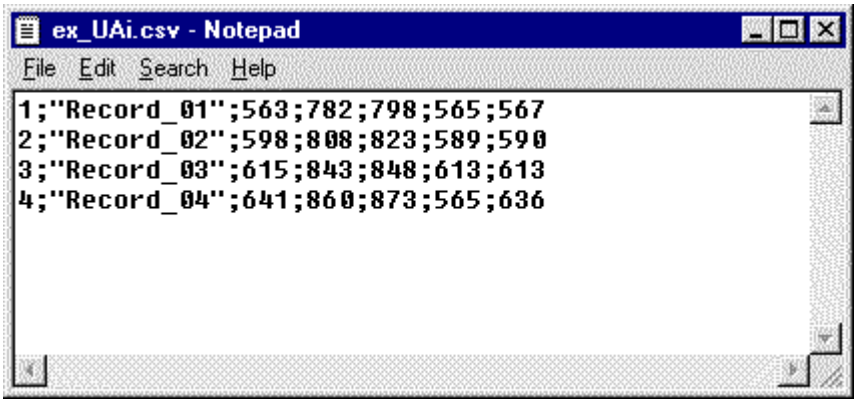
Step	Procedure: Data Entry in the User Archives Editor
1	Open the *User Archives Editor*. In order to enter data records into an archive, the archive must be selected in the left tree structure with the  . 
2	The archive must be switched into the mode for entering data. This is done via the *Edit* → *Runtime Data* menus. Only in this mode can data of an archive be changed. The switch into this mode can only be performed after saving all changes made. In the lower table window, a blank table is displayed whose columns correspond to the data structure of the configured archive. 
3	Entering of the data into the table. To enter data into a field, it must be selected with a  D. If the table window is the active window, a new data record can also be entered by clicking on the toolbar button displayed below.  The entry of a value is completed with the ENTER key. 
4	If the table window is the active window and the toolbar button displayed below is pressed, the table window is switched to the mode for the simplified change and editing of data.  Now the selection of a field with the  makes the entry possible. If the table window is the active window and the toolbar button displayed below is pressed, the currently selected data record is deleted. 

	ID	RecordNam	Pressure_a	Pressure_b	Pressure_c	Pressure_d	Pressure_e
...							

	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
...							

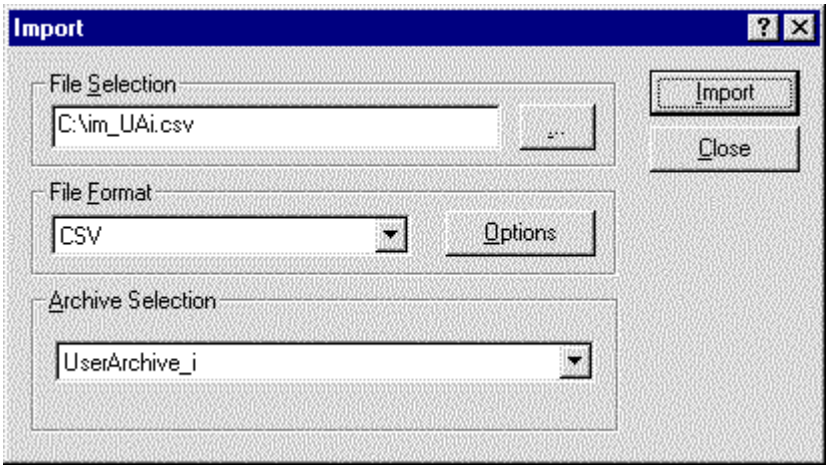
Import of Data from CSV Files

Step	Procedure: Import of Data from CSV Files
1	Determination of the structure of an import file. For this purpose, the previously entered data records are exported to a CSV file. This is done via the toolbar button displayed below.  Make sure that the *User Archives Editor* is not in the mode for entering archive data, i.e. the button displayed below must not be pressed. Only then, is the export as well as the import of data possible.  The *Export* dialog will be displayed.  In the *File Selection* field, the name of the file is specified to which the data records of the archive are to be exported. Via the button displayed below, an already existing file can also be selected. 

Step	Procedure: Import of Data from CSV Files
	In this sample, the *ex_UAi.csv* file is set in the *File Selection* field. The CSV (Comma Separated Values) file format is kept. Via the Options button, the separator used for the file can be changed. In this sample, the default semicolon is kept.  In the *Archive Selection* field, the archive *UserArchive_i* is specified. For the *Filter*, *All* is kept - thus all the data of the archive is exported. It is also possible to set the filter to the values of a certain field or to make a selection using an SQL expression. Via the *Export* button, the export process of the data to the selected file is started.
2	The just created file can now be opened using any text editor. In this sample, the file has been opened using the Microsoft Editor. The file structure required for the import is obtained.  While creating a file, which is to be the base for the import of data, make sure that the single quotation marks around text fields are removed, otherwise errors will occur during the import. Already existing data records will not be overwritten during the import. Data records are identified using the ID's displayed in the first column. Import files should only contain data records with ID's that have not been assigned yet.

Step	Procedure: Import of Data from CSV Files
3	You now have the option to open the file obtained in Excel and for further processing. In the Open dialog of Excel, specify the file type Text Files instead of Microsoft Excel Files. The individual fields of the archive will then be displayed separated into table columns. New data records can now be conveniently added, as shown below. In this sample, the file is only saved via the *File* → *Save As... menus and* renamed to *im_UAi.csv*. Excel will then automatically remove the single quotation marks around text fields. Close Excel. The final query to save can be answered with *No*, since the data has already been saved in a new file.
4	Importing the data into the *User Archives Editor*. This is done via the toolbar button displayed below. The *Import* dialog will be opened. In the *File Selection* field, the file containing the data is specified via the button displayed below. In this sample, the previously created file *im_UAi.csv* is selected in the *File Selection* field. In the *File Format* field, *CSV* is kept. Via the *Options* button, the separator must be specified, which is used in the file. In this sample, the default semicolon is kept. In the *Archive Selection* field, the target archive is set, in the sample, this is the *UserArchive_i*. Via the *Import* button, the data import is started.

	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								



Step	Procedure: Import of Data from CSV Files
5	The data imported is written directly into the database. It is therefore not necessary to save the data again.

Note:

While importing, make sure that the file is no longer open in Excel. Otherwise an error is reported. The same error message will also be reported, if the file contains existing data records or the data structure of the file is not identical to the archive's structure.

5.1.3 Configuration of a Table View (ex_3_chapter_01.PDL)

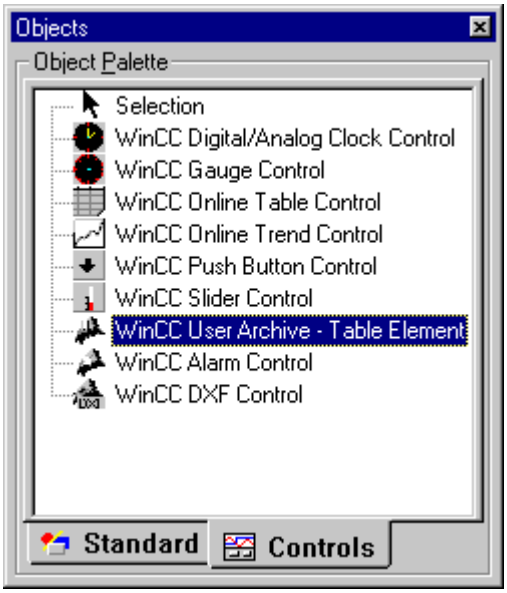
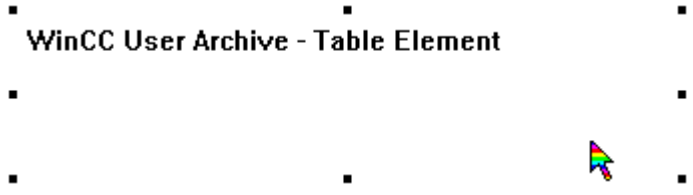
Task Definition

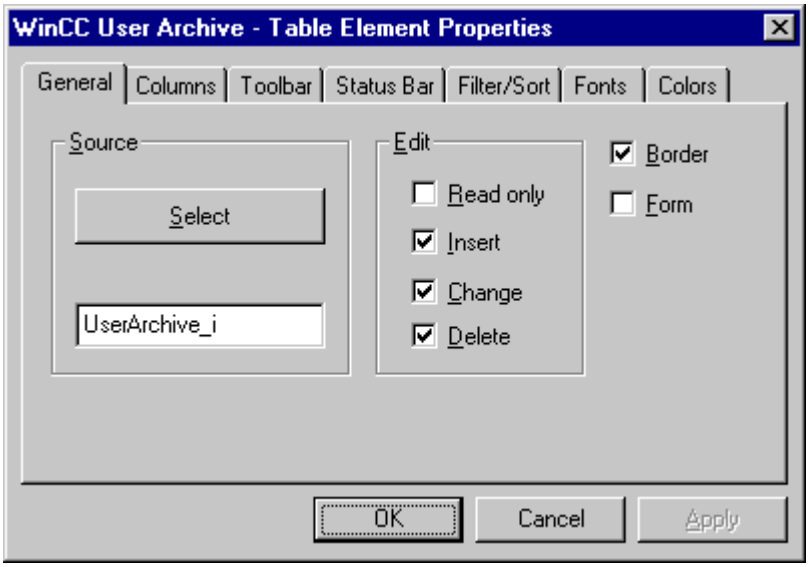
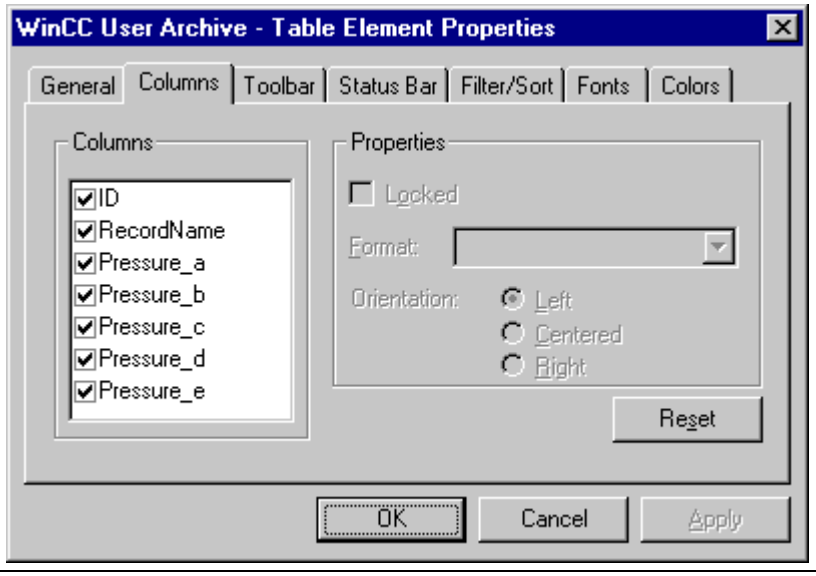
This sample is based on the two previous samples Creation of User Archives (ex_3_chapter_01.PDL) and Data Entry (ex_3_chapter_01.PDL). The data of the archives configured in those samples is to be made available to the user in runtime. The display of the archive data is to be made in the form of a table.

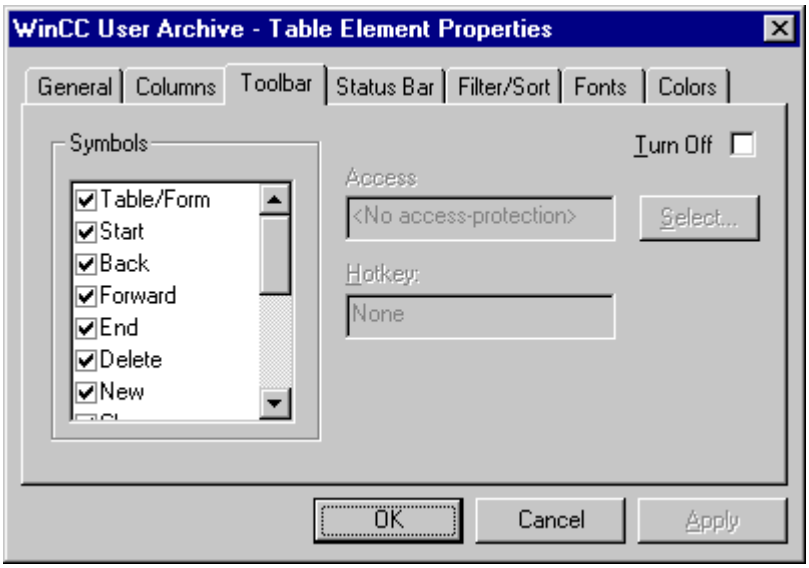
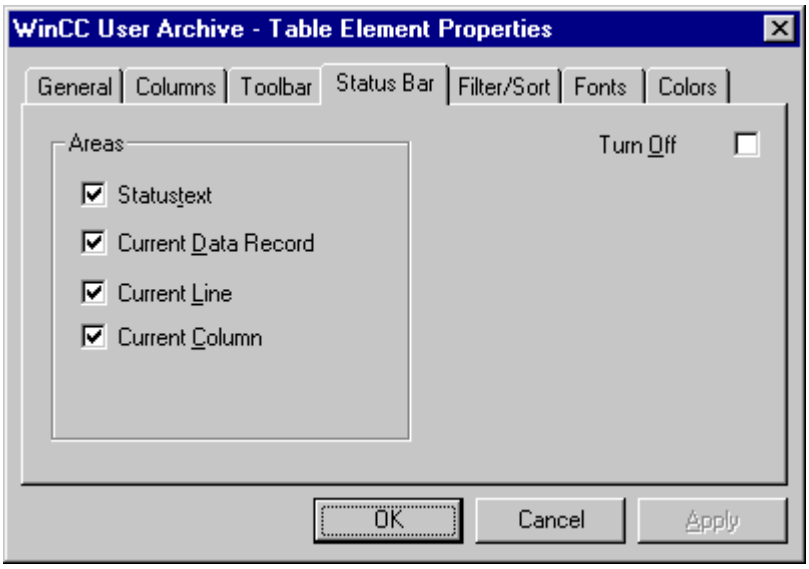
Implementation Concept

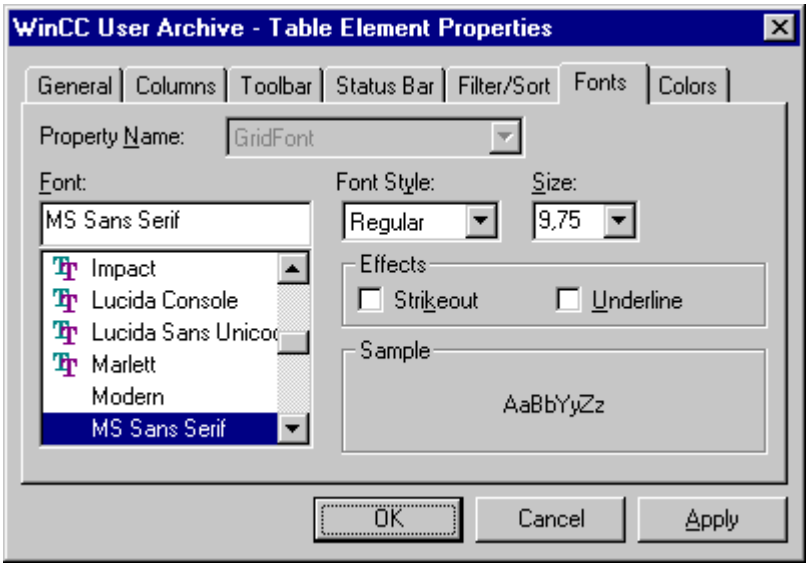
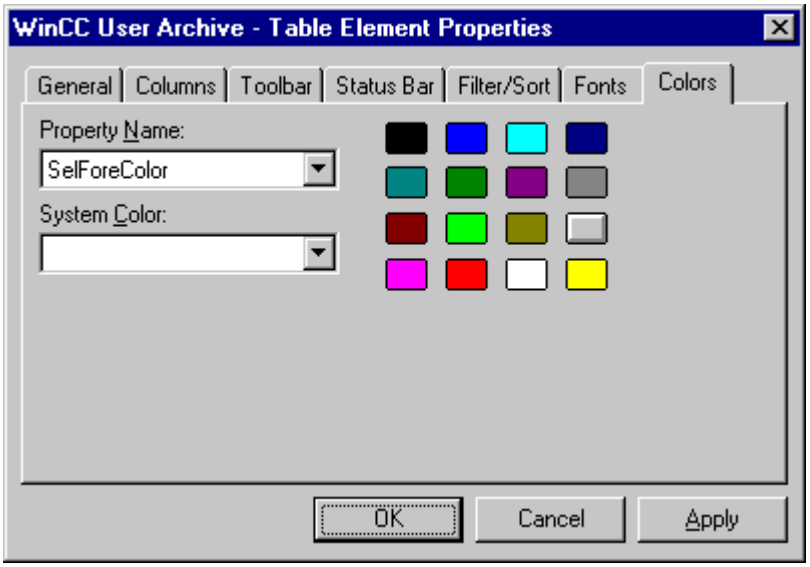
In runtime, the archive is displayed via a special Control. This Control displays the data in table form. Via this Control, various editing options for the archive displayed can be made available to the user.

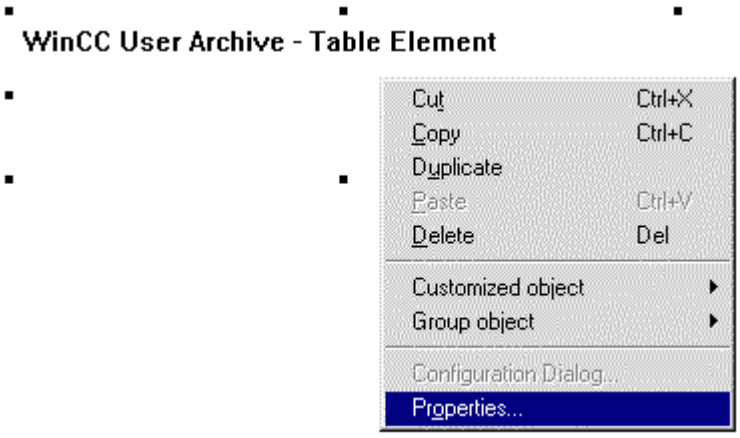
Implementation in the Graphics Designer

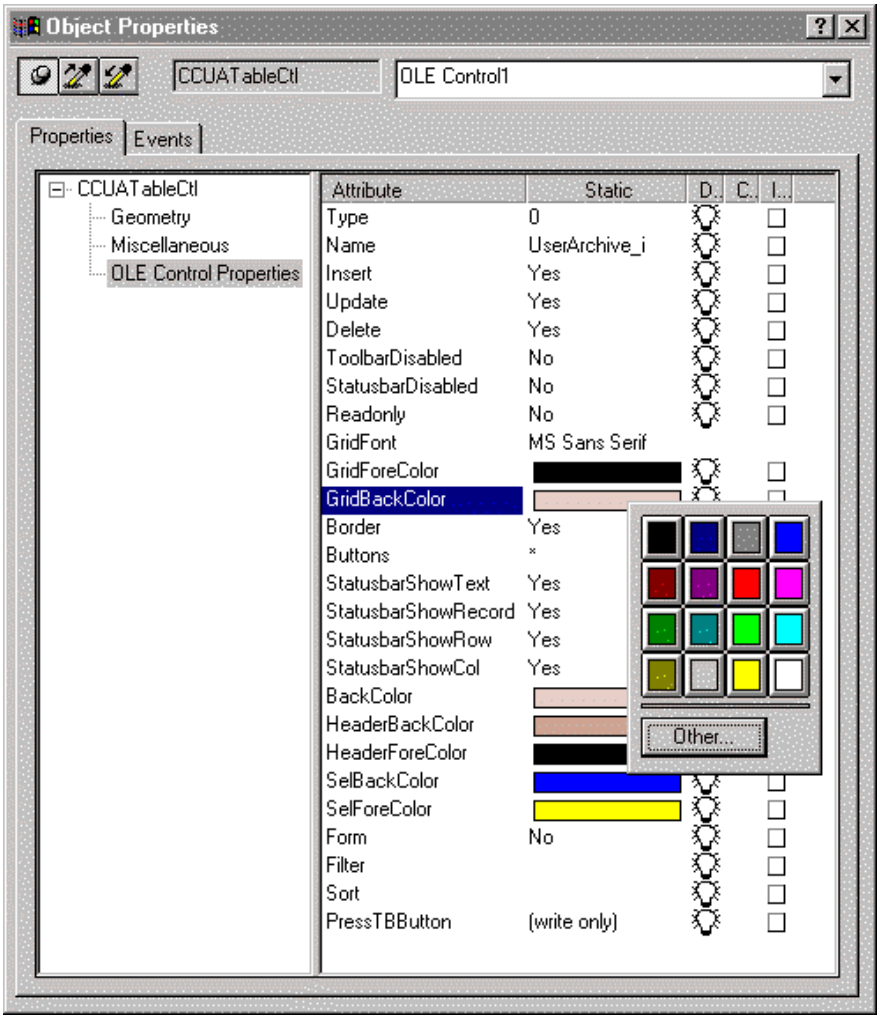
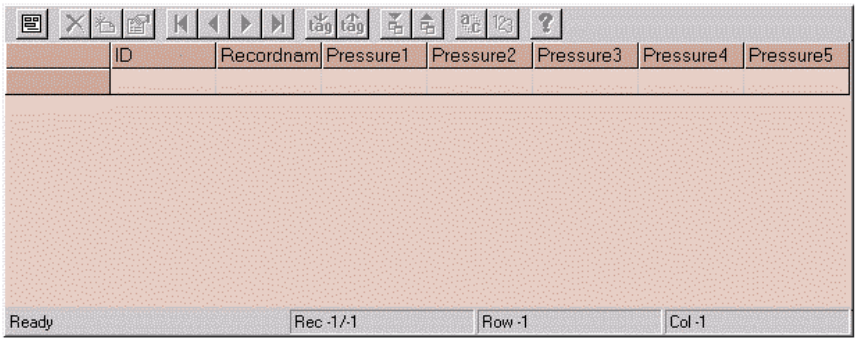
Step	Procedure: Implementation in the Graphics Designer
1	Open the *Graphics Designer* and create a new picture. In this sample, this is the *ex_3_chapter_01.pdl* picture.
2	Configuration of the Control used to display the data. This is the WinCC User Archives - Table Element. It is selected from the Object Palette's Control selection menu and then placed in the picture. 
3	Via a D on the just created *Control1* object, its properties dialog is opened. 

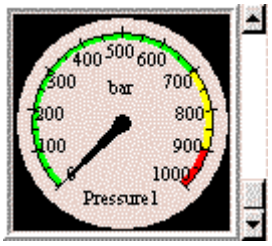
Step	Procedure: Implementation in the Graphics Designer
4	In the *General Information* tab, the *Archive* option is selected in the *Source* field. In this sample, a *View* cannot be selected, since none has been configured. As the archive, *UserArchive_i* is selected. In the *Edit* field, the editing options for the archive accessible to the user can be specified. In this sample, the maximum editing options are made available to the user. The Control can be displayed with a *Border*. In addition, it is possible to display the object using the form view. In this sample, the *Border* check-box is selected. The *Form* check-box, however, is not selected. 
5	In the *Columns* tab, the data fields displayed in the table can be specified. In this sample, all available fields are selected. In addition, certain properties can be defined for each individual data field in the *Properties* field. In this sample, however, the default settings for all data fields are kept. 

Step	Procedure: Implementation in the Graphics Designer
6	In the *Toolbar* tab, the elements needed for the toolbar can be configured. In this sample, all functions remain selected. It is possible to assign an authorization level to each toolbar button, which permits operation only by authorized users. If the *Turn Off* check-box is selected, the toolbar will not be displayed. If individual buttons are deselected via the check-boxes in the list to the left, they will be displayed in an inoperational state. 
7	In the *Status Bar* tab, the appearance of the status bar can be defined. In this sample, all elements remain selected. If the *Turn Off* check-box is selected, the status bar will not be displayed. 

Step	Procedure: Implementation in the Graphics Designer
8	In the *Fonts* tab, the font used by the Control can be changed to meet your requirements. In this sample, the *Size* of the font is reduced to *10* in order to display all columns simultaneously in runtime. All other settings are kept. 
9	In the *Filter/Sort* tab, not settings are made in this sample. Filters and sort commands can also be set during runtime, if the corresponding buttons have been provided in the toolbar.
10	In the *Colors* tab, the colors of the individual table elements can be changed. A more convenient place for setting colors, especially if more than 16 colors are needed, is the properties dialog of the *Control1* object itself, made available by WinCC. 
11	The settings made in the properties dialog of the *WinCC User Archives - Table Element* are concluded via the *OK* button.

Step	Procedure: Implementation in the Graphics Designer
12	Setting the color scheme of the table. For this purpose, R and then select *Properties* from the pop-up menu to open the properties dialog of the *Control1* object.  In this sample, the colors *GridBackColor*, *BackColor* and *HeaderBackColor* are adapted to the color scheme used in the project. For the remaining colors, the default settings are kept.

Step	Procedure: Implementation in the Graphics Designer
	You can also make all the settings of the *WinCC User Archives Table Control Properties* dialog in here. For some settings, however, this is not useful. 
13	Some of the configurations made can already be checked in the *Graphics Designer*. With a D on the *Control1* object while the CTRL key is pressed, the object is activated. 

Step	Procedure: Implementation in the Graphics Designer
14	To display the tag values, which are written by the User Archives, the sample uses five pointer instruments. As the pointer instruments, *Controls* of the *WinCC Gauge Control* type are used. In this sample, these are the *Control2* to *Control6* objects. For each of these objects, a *Tag Connection* to one of the five integer tags written to by the archive is created at *Properties* → *Control Properties* → *Value*. To simulate a value change of these tags, five *Smart Objects* → *Slider Objects* are configured. In the sample, these are the *SliderObject1* to *SliderObject5*. For these objects, a *Tag Connection* to one of the corresponding tags is created at *Properties* → *Miscellaneous* → *Process Driver Connection*. Furthermore, a *Direct Connection* each to the same tag is created at *Events* → *Property Topics* → *Miscellaneous* → *Process Driver Connection* → *Change*. For these *Direct Connections*, the *Source This Object* → *Process Driver Connection* is connected with the *Target* of the corresponding *Tag*. This is done to achieve an immediate change of the tag value, every time the slider position is changed. 
15	To display the data record currently loaded in the tag, a *Standard Object* → *Static Text* is configured. In the sample, this is the *StaticText2* object. For this object, a *Tag Connection* to the tag *T08i_ex_UAi_01* containing the name of the data record is created at *Properties* → *Font* → *Text*.

Note for the General Application

- The toolbar buttons of the Control displayed below perform the following tasks in runtime:
-  Via the Form button, a switch between the table view and the form view can be performed in runtime.
-  Via the Edit button, the data of the archive can also be changed in runtime. Data records can be deleted, added or changed. To what extent the user has the right to change data, depends on the settings of the Control in the *Graphics Designer*.
-  Via the Navigation buttons, the user can move around in the table and select different data records. The selection of data records can also be made by directly selecting a record with the mouse.
-  The Data Transfer buttons are used for writing and reading data. Via the *Write Tags* button, the data record selected in the table is written to the tag set in the *User Archives*. Via the *Read Tags* button, the current values of the tag set in the *User Archives* are read into the data record selected in the table.
-  Via the Export and Import buttons, the archive data can be exported to CSV files or new data from CSV files be imported. These buttons perform the same functions as the corresponding buttons in the *User Archives Editor*. The procedure for the import and export of data is described in the Data Entry (ex_3_chapter_01.PDL) sample.
-  Via the *Filter* and *Sort* buttons, the data records to be displayed and their display order can be defined. These settings for the Control can also be made in the *Graphics Designer*.

5.1.4 Configuration of a Form View (ex_3_chapter_011.PDL)

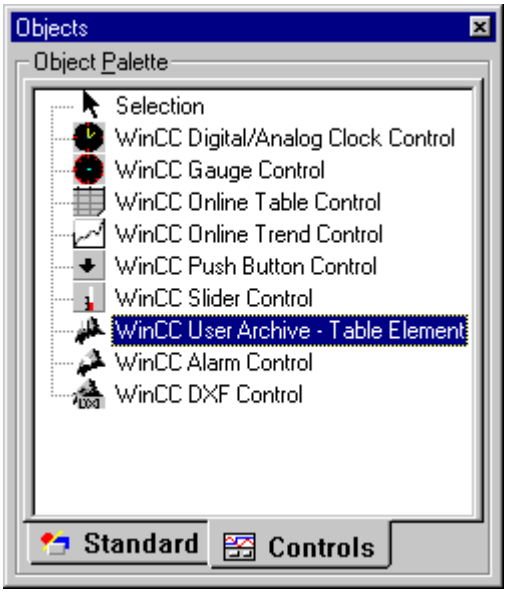
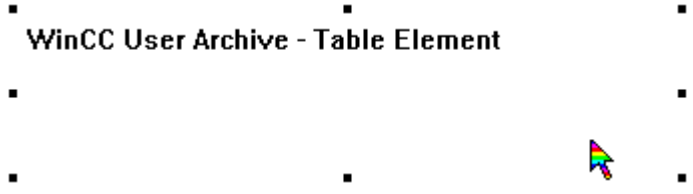
Task Definition

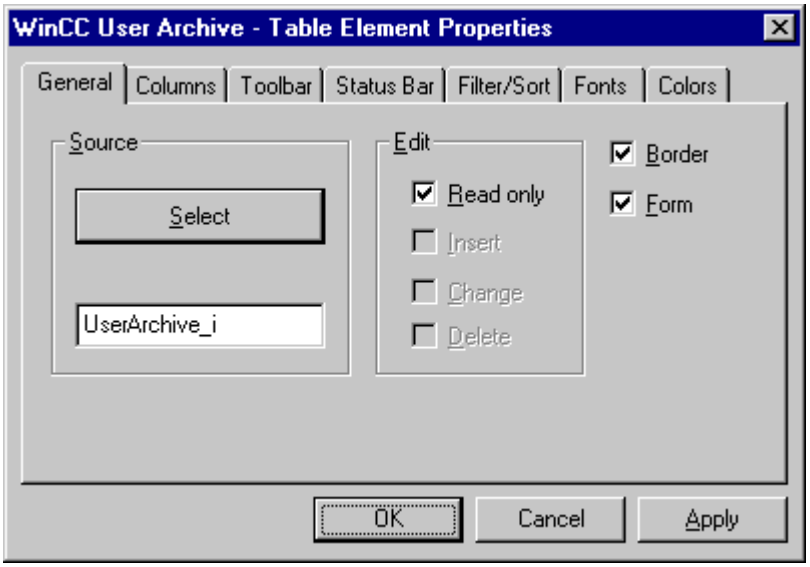
This sample is based on the previous samples Creation of User Archives (ex_3_chapter_01.PDL) and Data Entry (ex_3_chapter_01.PDL). The data of the archives configured in those samples is to be made available to the user in runtime. The display of the archive data is to be in a form, which displays one data record each.

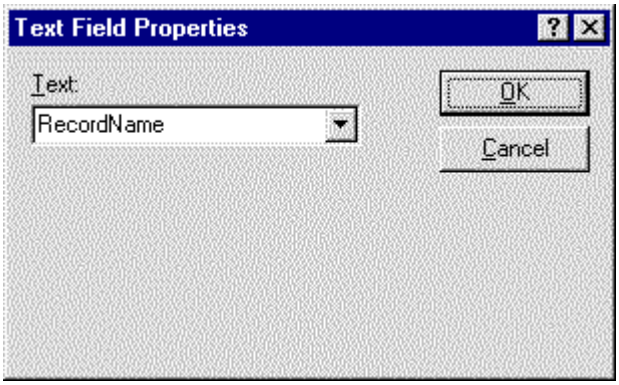
Implementation Concept

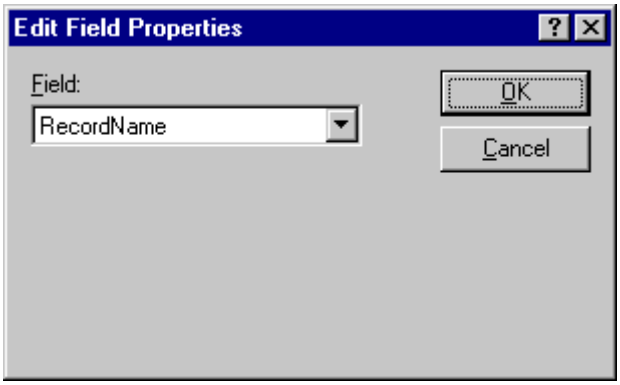
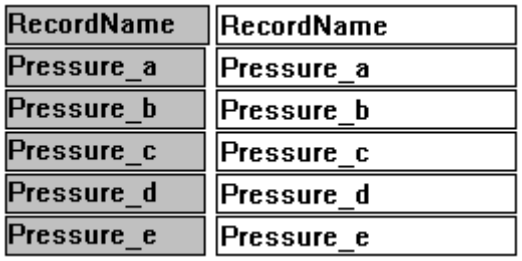
In runtime, the archive is displayed via a special Control. The data is displayed in a form. Via multiple buttons, various editing options for the archive are made available to the user.

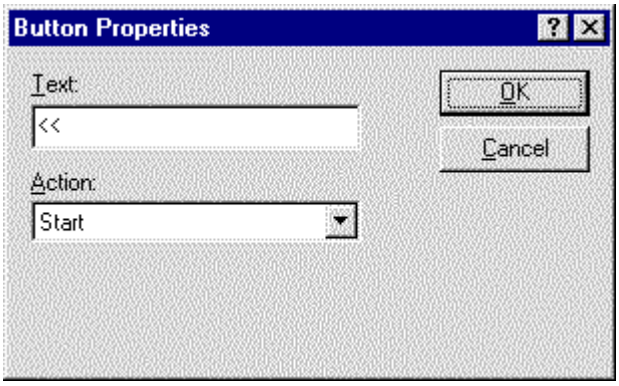
Implementation in the Graphics Designer

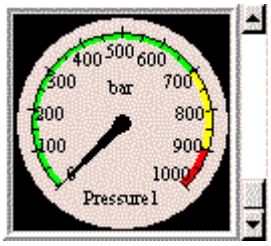
Step	Procedure: Implementation in the Graphics Designer
1	Open the *Graphics Designer* and create a new picture. In this sample, this is the *ex_3_chapter_011.pdl* picture.
2	Configuration of the Control used to display the data. This is the *WinCC User Archives - Table Element*. It is selected from the *Object Palette's Control* selection menu and then placed in the picture. 
3	Via a D on the just created *Control1* object, its properties dialog is opened. 

Step	Procedure: Implementation in the Graphics Designer
4	In the *General Information* tab, the *Archive* option is selected in the *Source* field. In this sample, a *View* cannot be selected, since none has been configured. As the archive, *UserArchive_i* is selected. In the *Edit* field, the editing options for the archive accessible to the user can be specified. In this sample, the check-box *Read Only* is selected, preventing the user from changing the archive data. The Control can be displayed with a *Border*. In addition, it is possible to display the object using the form view. In this sample, the *Border* and *Form* check-boxes are selected. 
5	In the *Toolbar* tab, a toolbar can be configured. In this sample, however, the *Turn Off* check-box is selected. Therefore, no toolbar is displayed in runtime. In the *Status Bar* tab, a status bar can be configured. As in the previous tab, the check-box *Turn Off* is selected. Therefore, no status bar is displayed in runtime as well.
6	In the remaining tabs, no settings are made in this sample. The settings made in the properties dialog of the *WinCC User Archives - Table Element* are concluded via the *OK* button.

Step	Procedure: Implementation in the Graphics Designer
7	To configure the form view, the Control must be activated in the *Graphics Designer*. This is done via a D on the object while keeping the CTRL key pressed. Via a R on the activated object, a pop-up menu is accessed which allows text fields, input fields or buttons to be inserted into the form. Via the *Insert Text Field* menu, a text field is entered. 
8	The text field is displayed in the form and its properties dialog is opened at the same time. Any text can be entered in the *Text* field. Furthermore, a name or alias of a data field can be selected from the drop-down list. In this sample, the alias *@TXT:40* of the RecordName field is selected from the drop-down list for the *Text* field. Close the dialog box by clicking on *OK*. 

Step	Procedure: Implementation in the Graphics Designer
9	Configuration of the input field suitable to the text field. This is done via a R on the activated Control and then selecting *Insert Edit Field*. The input field is displayed in the form and its properties dialog is opened at the same time. The desired field, whose data is to be displayed or entered, can be selected from the drop-down list. In this sample, the *RecordName* field is selected for the *Field* field. Close the dialog box by clicking on *OK*. 
10	The input field is displayed in the form and its properties dialog is opened at the same time. The desired field, whose data is to be displayed, can now be selected from the drop-down list. In this sample, the *RecordName* field is selected for the *Field* field.
11	Following the previous four steps described, a text and input field is configured for each archive field. 

Step	Procedure: Implementation in the Graphics Designer
12	It must be possible to move through the fields of the archive. This is accomplished by configuring buttons. In the form, all functions that are available from the toolbar can be performed using buttons. Via a R on the activated Control and selecting *Insert Button*, a *Button* is inserted into the form. The button is displayed in the form and its properties dialog is opened at the same time. In the *Text* input field, any text can be entered as the button label. In this sample, the character string is used as the *Text*. This button symbolizes the function for jumping to the first data record of the archive. In the *Action* field, the function to be performed by the button is set. In a drop-down list, all functions available are listed. In this sample, the *Start* function is specified. 
13	Three additional buttons are configured. They perform the functions *Back*, *Forward* and *End*.  Furthermore, two buttons are configured, which allow the user to write individual data records into tags and to read records from tags. The functions to be set for these buttons are *Read Tags* and *Write Tags*. 

Step	Procedure: Implementation in the Graphics Designer
14	To display the tag values, which are written by the *User Archives*, the sample uses five pointer instruments. As the pointer instruments, *Controls* of the *WinCC Gauge Control* type are used. In this sample, these are the *Control2* to *Control6* objects. For each of these objects, a *Tag Connection* to one of the five integer tags written to by the archive is created at *Properties* → *Control Properties* → *Value*. To simulate a value change of these tags, five *Smart Objects* → *Slider Objects* are configured. In the sample, these are the *SliderObject1* to *SliderObject5*. For these objects, a *Tag Connection* to one of the corresponding tags is created at *Properties* → *Miscellaneous* → *Process Driver Connection*. Furthermore, a *Direct Connection* each to the same tag is created at *Events* → *Property Topics* → *Miscellaneous* → *Process Driver Connection* → *Change*. For these *Direct Connections*, the *Source This Object* → *Process Driver Connection* is connected with the *Target* of the corresponding *Tag*. This is done to achieve an immediate change of the tag value, every time the slider position is changed. 
15	To display the data record currently loaded in the tag, a *Standard Object* → *Static Text* is configured. In the sample, this is the *StaticText2* object. For this object, a *Tag Connection* to the tag *T08i_ex_UAi_01* containing the name of the data record is created at *Properties* → *Font* → *Text*.

5.1.5 Working with Control Tags (ex_3_chapter_012.PDL)

Task Definition

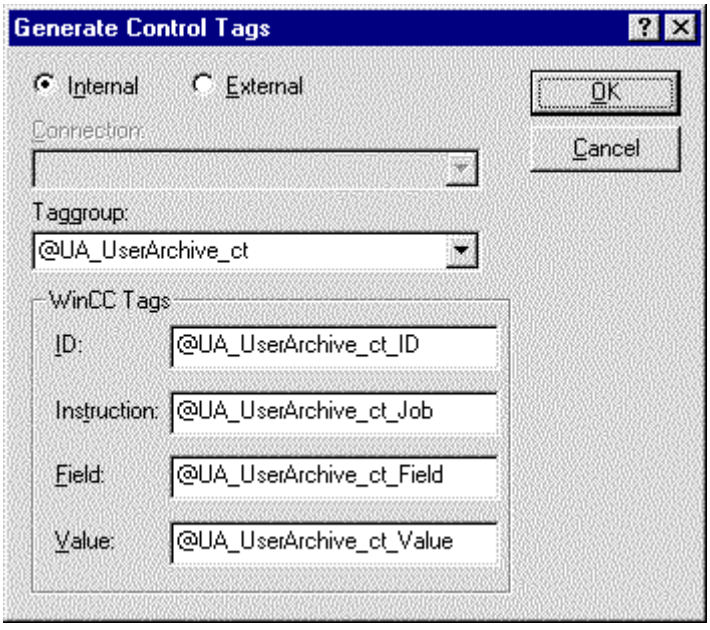
An archive is to be created, whose data records consist of three floating-point number fields as well as a text field for recording the data record name. The number of data records is to be limited to three. The archive data is to be made available to the entire project record by record through the application of control tags.

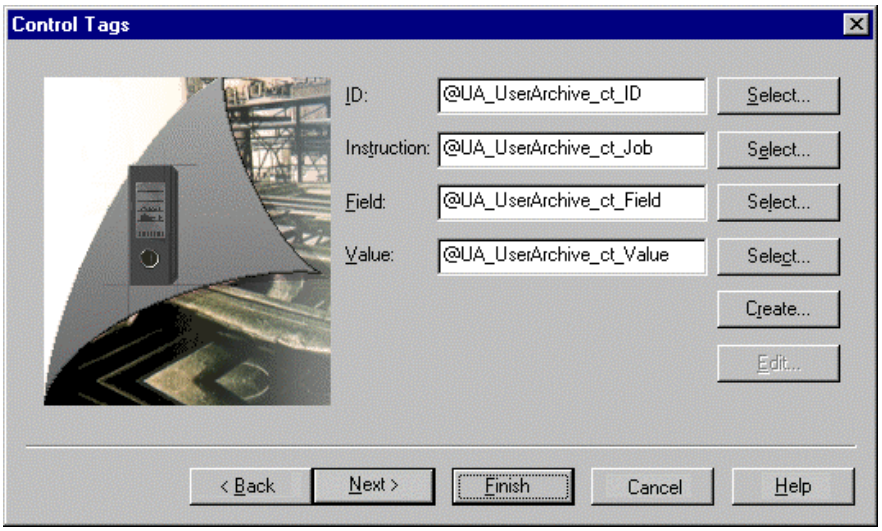
Implementation Concept

To archive the data, an archive is created in the User Archives Editor. The communication of the archive is configured using WinCC tags. In this archive, four data fields of the required field types are created. An internal tag is assigned to each data field, which allows the archive to communicate with the remaining project.
Four control tags are assigned to the archive. They control the writing of the data into the WinCC tags as well as the reading of the data from the WinCC tags. In the Graphics Designer, multiple buttons and I/O fields are configured, which are used to write to the control tags. The contents of the control tags define, which data record is to be edited and whether data is to be read or written.

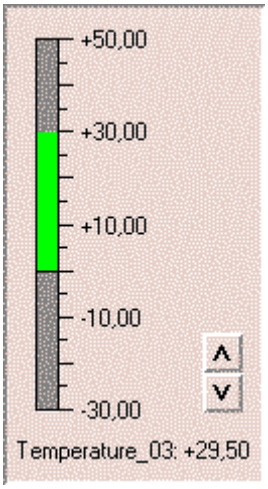
Creation of an Archive

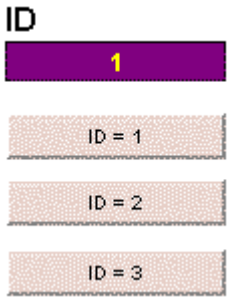
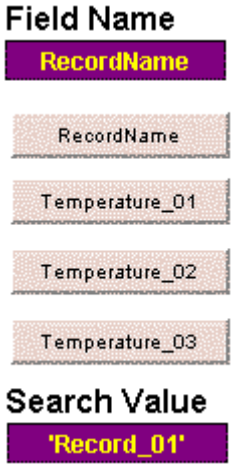
Step	Procedure: Creation of an Archive
1	Creation of one internal tag for each data record of the archive. The communication between the archive and the remaining system is carried out via these tags. In this sample, the tags *G64i_ex_UAct_01* to *G64i_ex_UAct_03* of the *Floating-Point Number 64-Bit IEEE 754* type are used. In addition, a tag of the *Text Tag 8-Bit Character Set* type is created for storing the data record name. In the sample, this is the *T08i_ex_UAct_01* tag.
2	Open the *User Archives Editor*. In this editor, a new archive is created. Via a R on the *Archives* entry, a Wizard is started for this purpose. 
3	On the first page of this Wizard, the *Archive Name* is entered. In the sample, the name *UserArchive_ct* is entered in the *Archive Name* field. The *Alias* field is left blank. As the *Archive Type*, *Limited* is selected. In accordance with the task definition, the maximum number of *Data Records* is set to the value *3*. Continue to the next page by clicking on *Next*.
4	On the second Wizard page, the communication type is selected. In this sample, the *Type Communication via WinCC Tag(s)* is selected. Continue to the next page by clicking on *Next*

Step	Procedure: Creation of an Archive
5	On the third Wizard page, four control tags can be set. Via these tags, the archive can be controlled in runtime. If the tags have not been created previously, this task can be performed automatically by the Wizard. Via the *Create* button, a dialog is accessed in which settings pertaining to the tags to be created are made. Specify, whether internal or external tags are to be created. If the Wizard is to create external tags, the *Connection* to be created for these tags must also be specified. The Wizard creates the tags in a tag group. The name of the group as well as the names of the individual tags must be specified. The Wizard also makes suggestions. In this sample, the tag type *Internal* is selected. The names suggested by the Wizard for the tag group as well as the individual tags are accepted. Via *OK*, the *Create Control Tags* dialog is closed. 

Step	Procedure: Creation of an Archive
	The just created tags were automatically entered into the corresponding fields for *ID*, *Job*, *Field* and *Value*. Since no settings are made for this sample on the next Wizard page, the Wizard can be exited by clicking on *Finish*. 
6	For the just created archive, the fields listed in the following table are created. As the tags to be assigned, the tags configured is step 1 are set. For the floating-point number fields, the *Minimum Value* of *-30* and the *Maximum Value* of *50* is set. The *Length* of the text field is set to *10*. 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	In the lower table window of the *User Archives Editor*, a total of three data records can be now be created for the archive. 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 ...

Implementation in the Graphics Designer

Step	Procedure: Implementation in the Graphics Designer
1	Open the *Graphics Designer* and create a new picture. In this sample, this is the *ex_3_chapter_012.pdl* picture. Configuration of the Control used to display the data. This is the *WinCC User Archives - Table Element*. It is selected from the *Object Palette's Control* selection menu and then placed in the picture.
2	Via a D on the just created *Control1* object, its properties dialog is opened. In the *General* tab, the previously configured archive *UserArchive_ct* is set in the *Source* field. In the *Edit* field, the setting *Read Only* is kept. The *Border* check-box is deselected. The *Form* check-box also remains deselected. In the *Toolbar* tab, the *Turn Off* check-box is selected. In the *Status Bar* tab, the *Turn Off* check-box is selected as well. Therefore, the table is displayed without a toolbar and status bar. The settings made in the properties dialog of the *WinCC User Archives - Table Element* are concluded via the *OK* button.
3	To display the tags written to by the *User Archives*, three *Smart Objects* → *Bar Graphs* are configured. In the sample, these are the *Bar1* to *Bar3* objects. For each of these objects, a *Tag Connection* to one of the three floating-point number tags written to by the archive is created at *Properties* → *Miscellaneous* → *Process Driver Connection*. To simulate a value change of these tags, two *Windows Objects* → *Buttons* are configured for each bar graph. Using the jog mode, one of these buttons increments the corresponding tag value, the other decrements it. 

Step	Procedure: Implementation in the Graphics Designer
4	Control elements must be configured, which allow the user to select a certain data record as well as to perform a function for the record selected. The selection of a data record via the control tags can be performed in two ways. • *@UA_UserArchive_ct_ID*A valid ID of a data record can be written to the control tag - that has been set during the creation of the archive - in the *ID* field (for a description, see step 5). • *@UA_UserArchive_ct_Field* and *@UA_UserArchive_ct_Value*A valid field name can be written to the control tag - that has been set during the creation of the archive - in the *Field Name* field. Via a search value entered for control tag set in the *Value* field, a certain data record is selected (for a description, see step 6).
5	For the selection of a data record via its ID, three *Windows Objects* → *Buttons* are configured in this sample. Via *Direct Connections*, they write a corresponding value to the *@UA_UserArchive_ct_ID* tag set for the ID. The current value of the tag is displayed via a *Smart Object* → *I/O Field*. 
6	For the selection of the data record via its field name and a search value, four *Windows Objects* → *Buttons* are configured in this sample. Via *Direct Connections*, they write a corresponding field name to the *@UA_UserArchive_ct_Field* tag set for the field name. The current value of the tag is displayed via a *Smart Object* → *I/O Field*. An additional *Smart Object* → *I/O Field* is used to input the search value into the *@UA_UserArchive_ct_Value* tag. If texts are used as search values, they must be enclosed in single quotation marks. 

Step	Procedure: Implementation in the Graphics Designer
7	In this sample, two *Windows Objects* → *Buttons* are used to switch between the two described selection options for data records. By default, the archive uses the control via the ID. If the control is to be performed via the field name and a search value, *0* must be written to the *@UA_UserArchive_ct_ID* tag. Via a *BINi_ex_UAct_01* tag of the *Binary Tag* type, the input objects of one method are switched into an inoperational mode. 
8	To trigger jobs for the currently selected data record, two *Windows Objects* → *Buttons* are configured. Via *Direct Connections*, they write a corresponding value to the *@UA_UserArchive_ct_Job* tag set for the job. Valid values are: • the value 6 for the job of reading the record from the tag • the value 7 for the job of writing the record to the tag • the value 8 for the job of deleting the record from the archive In this sample, only the jobs for reading tags and writing to tags are used.  Via the tag set for job, the successful execution of a job can also be checked. If the job has been executed successfully, the tag is set to zero. If the job failed, the tag is set to *-1*. In the sample, this is evaluated via a *Dynamic Dialog* at a *Smart-Object* → *I/O Field*. 

5.1.6 Communication via WinCC Raw Data Tags

General Information

The *User Archives* can communicate with a PLC using two different methods.

- *Communication via WinCC Tags*A WinCC tag is assigned to each data record of the archive. Data can be written to and read from these tags. This type of communication has been used in the samples described in the previous chapters.
- *Communication via WinCC Raw Data Tags*A WinCC raw data tag is assigned to the archive. Via this tag, the PLC send job telegrams to the archive. The archive responds to these job telegrams with an acknowledgment telegram.

Samples

Following this chapter, the samples listed below can be found. They show in detail the configuration steps necessary to establish the communication between a PLC and the *User Archives*.

- Communication to the SIMATIC S5 via WinCC Raw Data Tags (ex_3_chapter_01a.pdl)
- Communication to the SIMATIC S7 via WinCC Raw Data Tags (ex_3_chapter_01b.pdl)

Telegrams

The communication between a PLC and WinCC is carried out through the exchange of telegrams. In the case of the communication to the *User Archives* via WinCC raw data tags, the PLC is the active partner. The PLC sends job telegrams to the archive, which are answered by the archive with an acknowledgment telegram.
Each job telegram consists of a telegram header and a telegram body, which contains the actual job or even multiple jobs. The jobs itself, also consist of a telegram header followed possibly by data.
Via the acknowledgment telegram, the PLC is informed about the execution of the job. Additionally, the acknowledgment telegram also transmits data requested.

Job Telegram - Telegram Header

Byte Number	Description
0	Telegram length in Byte LSB (telegram length occupies a total of 4 Bytes)
1	Telegram length in Byte
2	Telegram length in Byte
3	Telegram length in Byte MSB
4	Transfer type: 2 corresponds to data sent by the PLC
5	Reserved
6	Number of Jobs LSB (number of jobs occupies a total of 2 Bytes)
7	Number of jobs MSB
8	1st character of the archive's PLCID (PLCID is eight characters long)
9	2nd character of the archive's PLCID
10	3rd character of the archive's PLCID
11	4th character of the archive's PLCID
12	5th character of the archive's PLCID
13	6th character of the archive's PLCID
14	7th character of the archive's PLCID
15	8th character of the archive's PLCID

The transfer type contained in the telegram header has the value 2 and corresponds to sending data to WinCC. The acknowledgment telegram sent by WinCC as the response to the job telegram has as the transfer type the value 1.
The 8 character long archive PLCID contained in the telegram header is used for the identification of the target archive. This PLCID is specified while creating the archive.

Job Telegram - Job

Byte Number	Description
0	Job length in Byte LSB (job length occupies a total of 2 Bytes)
1	Job length in Byte MSB
2	Job type
3	Reserved
4	Field number LSB (field number occupies a total of 2 Bytes)
5	Field number MSB
6	Data record number LSB (data record number occupies a total of 4 Bytes)
7	Data record number
8	Data record number
9	Data record number MSB
10	Selection criterion LSB (selection criterion occupies a total of 2 Bytes)
11	Selection criterion MSB
12 to n	Job data (data of the PLC)

The record and field numbers specified in the job are not relevant for all jobs.
The record number corresponds to the ID of a data record in the *User Archives*. The numbering of the data records starts at 1. If the record number 0 is specified for a write job to the archive, a new data record will be added to the archive.
The numbering of the fields starts at 0.

Job Types

In the job, one Byte is reserved for the job type. For the job types, the following values can be used:

Job Type	Description
4	Check archive for presence
5	Delete all data records from the archive
6	Read data record (from the archive)
7	Write data record (to the archive)
8	Delete data record
9	Read data field (from the archive)
10	Write data field (to the archive)

Job Data

The job data corresponds to the content of a data record for the job *Write data record* or to the content of a data field for the job *Write data field*. For the other job types, no job data is transmitted.
Number values - just like all number values specified in the telegram - are displayed in the Intel format, i.e. the LSB is transmitted first and the MSB last.
Integer fields configured in the *User Archives* have a length of 4 Bytes, floating-point numbers a length of 8 Bytes. Text fields are not 0-terminated.

Acknowledgment Telegram

Byte Number	Description
0	Telegram length in Byte LSB (telegram length occupies a total of 4 Bytes)
1	Telegram length in Byte
2	Telegram length in Byte
3	Telegram length in Byte MSB
4	Transfer type: 1 correspond to data sent by WinCC
5	Reserved
6	Error code: refer to following description about error codes
7	Job type: refer to previous description about job types
8	Reserved
9	Reserved
10	Field number LSB (field number occupies a total of 2 Bytes)
11	Field number MSB
12	Data record number LSB (data record number occupies a total of 4 Bytes)
13	Data record number
14	Data record number
15	Data record number MSB
16	1st character of the archive's PLCID (PLCID is eight characters long)
17	2nd character of the archive's PLCID
18	3rd character of the archive's PLCID
19	4th character of the archive's PLCID
20	5th character of the archive's PLCID
21	6th character of the archive's PLCID
22	7th character of the archive's PLCID
23	8th character of the archive's PLCID
24 to n	Acknowledgment data (data of the archive)

The transfer type contained in the acknowledgment telegram has the value 1 and corresponds to data sent by WinCC. The job telegram sent by the PLC contains as the transfer type the value 2.
The 8 character long archive PLCID contained in the acknowledgment telegram is used for the identification of the target archive. This PLCID is specified while creating the archive.

Error Codes

If an error occurs, WinCC will return an error code to the PLC in the acknowledgment telegram. The following error codes have been defined:

Error Code	Description
0	Function has been executed
1	Invalid data (archive)
2	Data not available (archive)
101	Invalid data (data record)
102	Data not available (data record)
201	Invalid data (field)
202	Data not available (field)
254	Function not available
255	Undefined error

Acknowledgment Data

The acknowledgment data corresponds to the content of a data record for the job *Read data record* or to the content of a data field for the job *Read data field*. For the other job types, no acknowledgment data is transmitted.
Number values - just like all number values specified in the telegram - are displayed in the Intel format, i.e. the LSB is transmitted first and the MSB last.
Integer fields configured in the *User Archives* have a length of 4 Bytes, floating-point numbers a length of 8 Bytes. Text fields are not 0-terminated.

5.1.7 Communication to the SIMATIC S5 via WinCC Raw Data Tags (ex_3_chapter_01a.pdl)

The STEP5 project created in this chapter can also be copied directly from the online document to your hard drive. By default, it will be stored to the *C:\Configuration_Manual* folder.


S5_UA_st

Task Definition

A communication connection between a SIMATIC S5 PLC and a WinCC station is to be established. The PLC is to read data from a User Archive created on the WinCC station, write data to and delete data from it.
The data records of the archive each consist of two integer fields as well as a text field for recording the data record name.

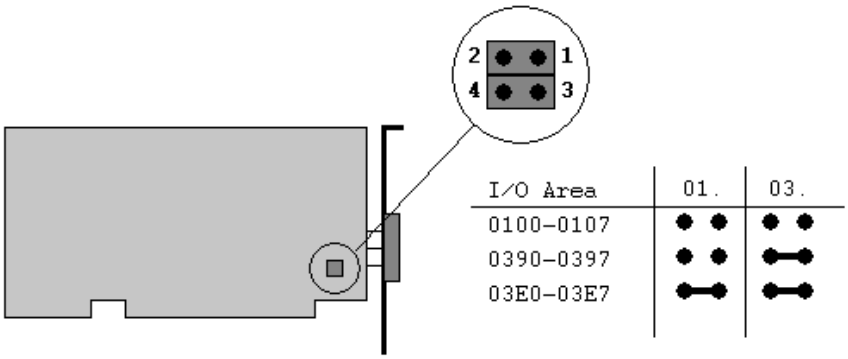
Implementation Concept

In this sample, a *SIMATIC S5-115U* PLC with a *CPU 944* central module is used. The communication to this PLC is established using *Industrial Ethernet*. For this purpose, the WinCC station uses the *CP 1413* communication processor and the PLC, the *CP 1430 TF* communication processor.
In WinCC, the *SIMATIC S5 ETHERNET LAYER 4* communication driver is used. This communication processor supports, among other things, the active sending of data from the PLC.
For the communication driver, two connections to the PLC are created. One connection is used for the active sending of data from the PLC. For this connection, a *WinCC Raw Data Tag* for the communication to the *User Archive* is created. In addition, a *User Archive* is created, whose communication is configured via this *WinCC Raw Data Tag*.
For the second connection, WinCC is the active partner. Among other things, this connection is used to simulate the operation of the communication from WinCC. Via various *Buttons*, control bits are set in the PLC, which trigger the sending of job telegrams. Additionally, the present data (the current data record, currently set record number and field number as well as the job status) in the PLC is displayed via *I/O Fields*.
In runtime, the data of the *User Archive* is displayed by a Control in tabular form.

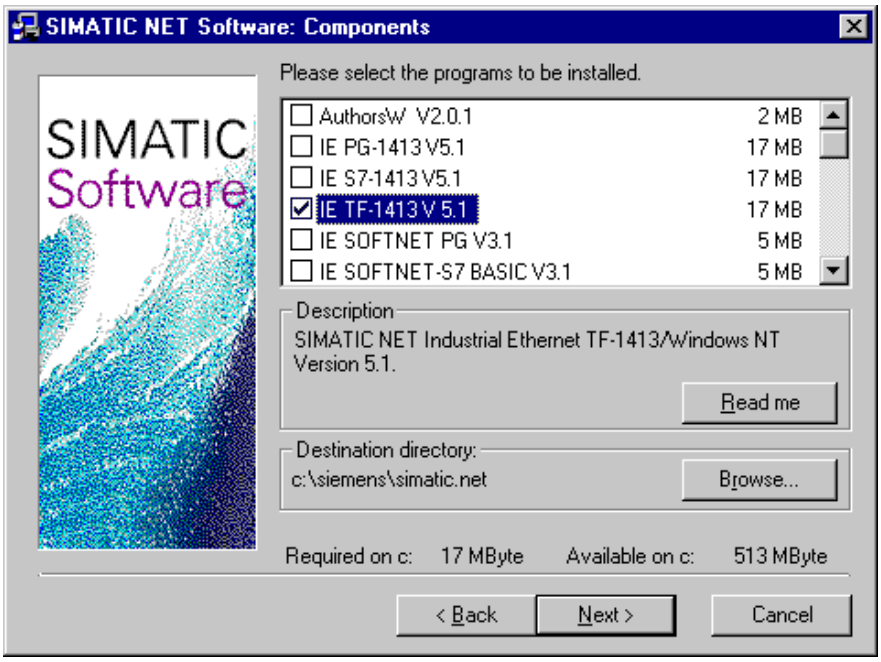
5.1.7.1 Startup of the Communication Processor CP 1413

The following description details the configuration steps necessary to successfully start up the communication processor *CP 1413*.

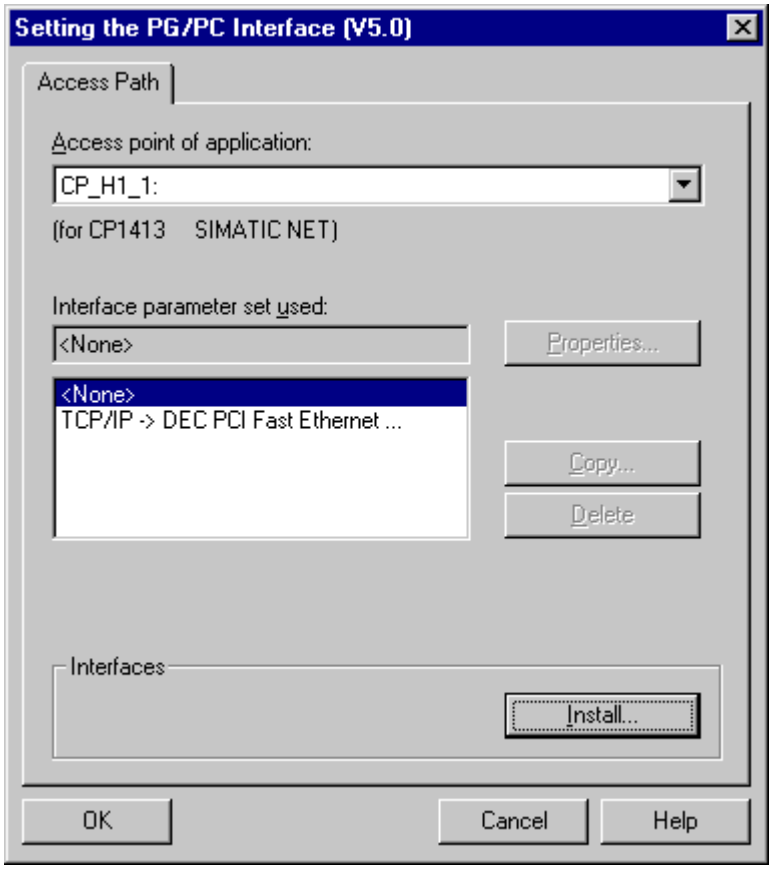
Mounting the Communication Processor in the Computer

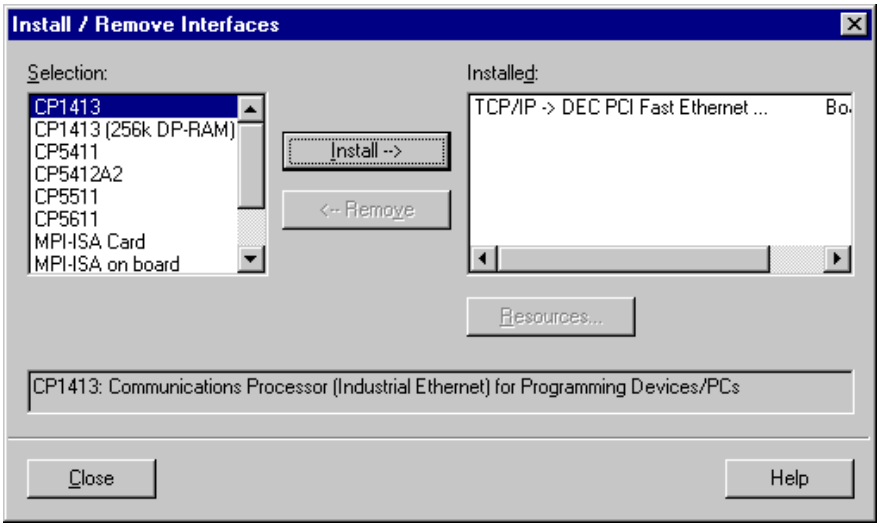
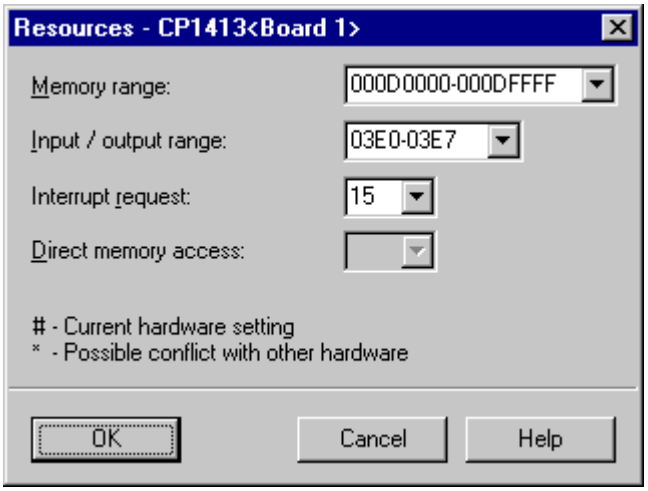
Step	Procedure: Mounting the Communication Processor in the Computer
1	Check the selected jumper settings at the *CP 1413*. During the software installation of the *CP 1413*, the *I/O Range* must be specified. This range is set via *Jumper Settings*. By default, the *I/O Range* is set to *03E0-03E7*. The settings *0100-0117* and *0390-0397* are also possible. The following graphic illustrates the jumper settings necessary for the various *I/O Ranges*. 
2	Mount the module according to the installation instructions. Among other things, follow the steps for handling electrostatic sensitive devices (ESD). The module must only be installed while the computer is off. For the communication card *CP 1413*, a free ISA slot in the computer is required. After the installation of the *CP 1413*, close the computer's case and start the computer.

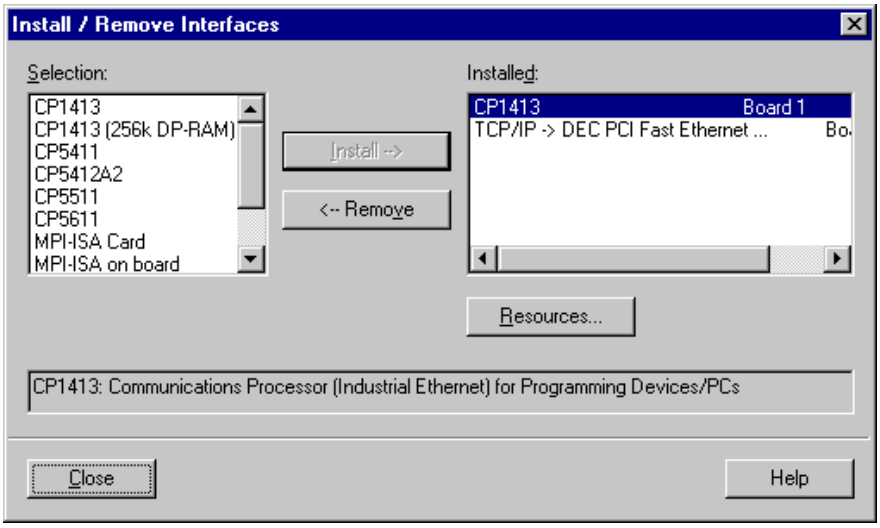
Installing the Communication Driver

Step	Procedure: Installing the Communication Driver
1	Installation of the communication driver *IE TF-1413* from the *SIMATIC NET* CD-ROM. After inserting the *SIMATIC NET* CD-ROM, the installation program is automatically started. If this is not the case, open the *Windows NT Explorer* and start the *setup.exe* program located on the CD-ROM. The installation of the software is started via the button displayed below.  Follow the instructions of the installation program. On the *Components* page, the check-box of the driver *IE TF-1413* to be installed must be selected. Finish the installation. 

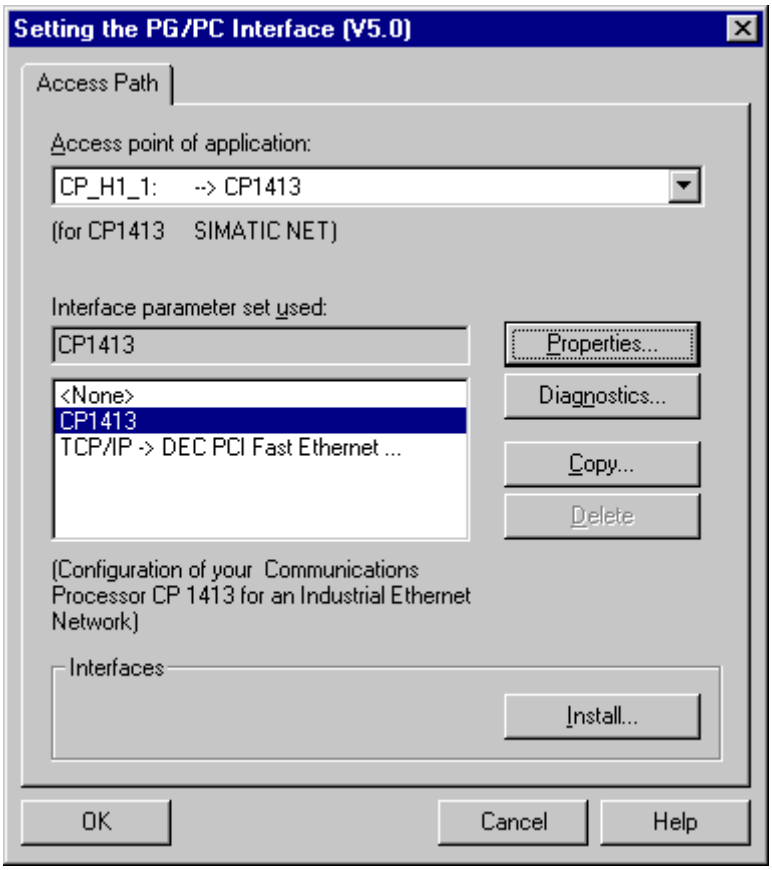
Installing the Communication Processor

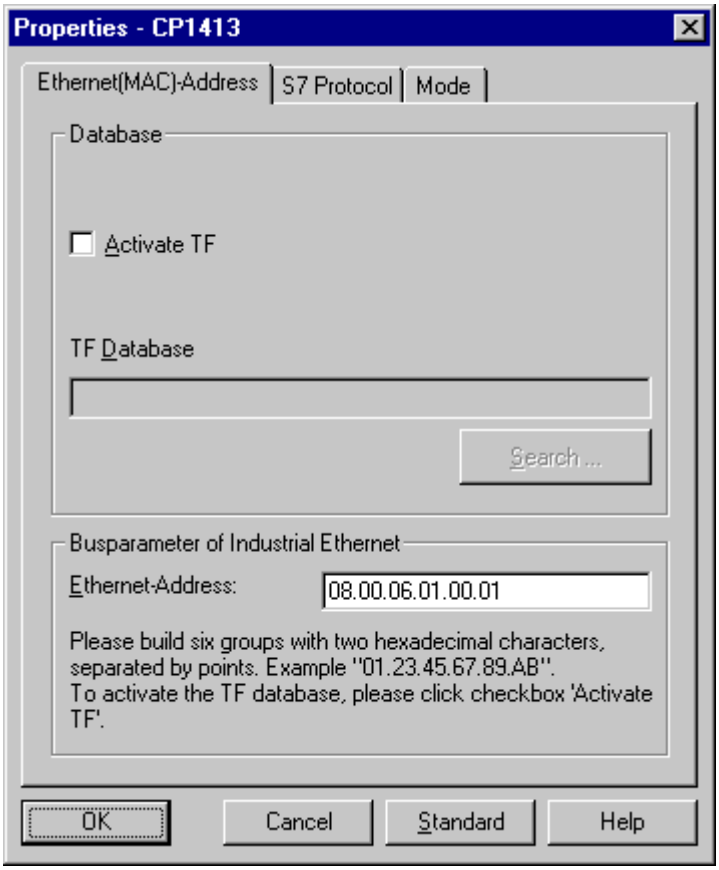
Step	Procedure: Installing the Communication Processor
1	Install the communication processor *CP 1413* via the program *Setting the PG/PC Interface*. This program is accessed via *Start* → *Settings* → *Control Panel* → *Setting the PG/PC Interface*.  Setting the PG/PC Interface
2	The program *Setting the PG/PC Interface* will be displayed. The dialog for installing a new interface is opened via the *Install* button. 

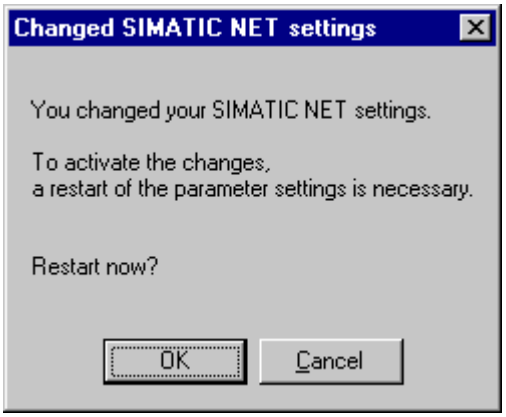
Step	Procedure: Installing the Communication Processor
3	The dialog *Install/Remove Modules* will be displayed. The *Selection* field lists all interfaces that can be installed. Among them will be the entry *CP 1413*, if the communication driver has been installed previously as outlined in step B. From the *Selection* field, select the entry *CP 1413*. The installation of the communication processor is started by clicking on the *Install* -> button. 
4	The dialog *Resources - CP 1413* will be displayed. The settings for the *Memory Range*, *I/O Range* and *Interrupt* have to be specified. The *I/O Range* has already been determined via the *Jumper Settings* at the *CP 1413*. Make sure that the assigned resources have not already been taken by other modules in the computer. Information about already taken system resources can be obtained from the *Resources* tab accessed via *Start* → *Programs* → *Administrative Tools (Common)* → *Windows NT Diagnostics*. Close the *Resources* tab by clicking on *OK*. 

Step	Procedure: Installing the Communication Processor
5	In the dialog *Install/Remove Modules*, the *Installed* field will now contain the entry for the *CP 1413*. Exit the dialog *Install/Remove Modules* via the *Close* button. 

Assigning the Communication Processor

Step	Procedure: Assigning the Communication Processor
1	In the program *Setting the PG/PC Interface*, assign the access point *CP_H1_1:* to the just installed interface. The access point *CP_H1_1:* is the default access point used by WinCC for the communication via the *Industrial Ethernet*. It has been created automatically during the installation of the communication driver *IE TF-1413*. In the field *Access Point of the Application*, set the entry *CP_H1_1:*. In the field below, select the entry *CP1413*. This completes the assignment between the access point and the communication processor. 

Step	Procedure: Assigning the Communication Processor
2	Setting the properties of the communication processor *CP 1413*. The dialog for setting the properties is opened via the *Properties* button of the *Setting the PG/PC Interface* program. The properties dialog of the communication processor *CP 1413* will be displayed. In the *Ethernet (MAC) Address* tab, enter the *Ethernet Address* of the *CP 1413*. In our sample, this is *08.00.06.01.00.01*. 

Step	Procedure: Assigning the Communication Processor
3	Exit the program *Setting the PG/PC Interface* via the *OK* button. A dialog will be displayed requesting the restart of the *CP 1413*. Acknowledge this dialog by clicking on *OK*, which will result in the restart of the communication processor *CP 1413*. This completes the installation of the communication processor. 
4	If the restart of the communication processor is not successful, the error cause must be determined and corrected. Help regarding this topic can be found in the *Communication Manual*.

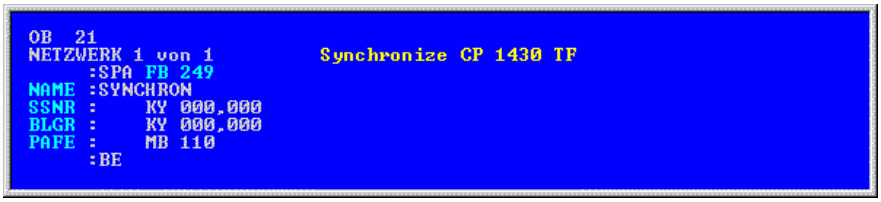
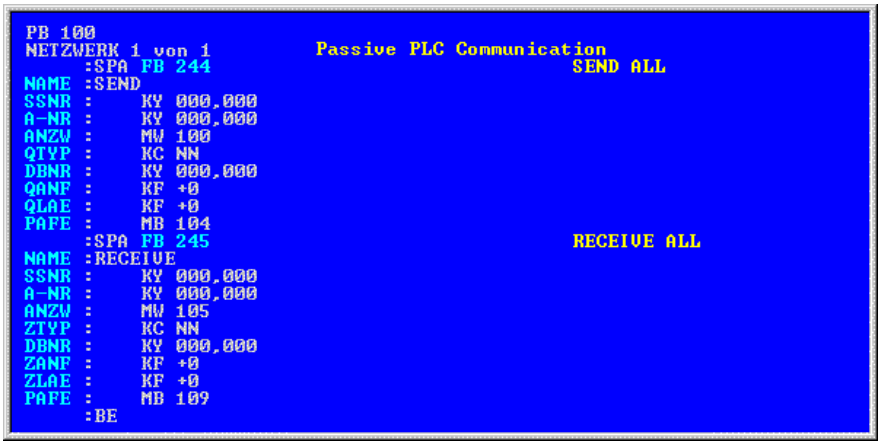
5.1.7.2 Startup of the PLC

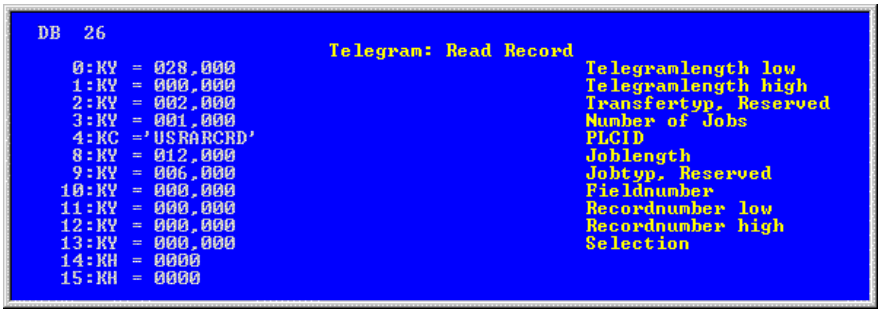
The following description details the configuration steps necessary to create and start up the STEP5 project *S5_UA_st*.

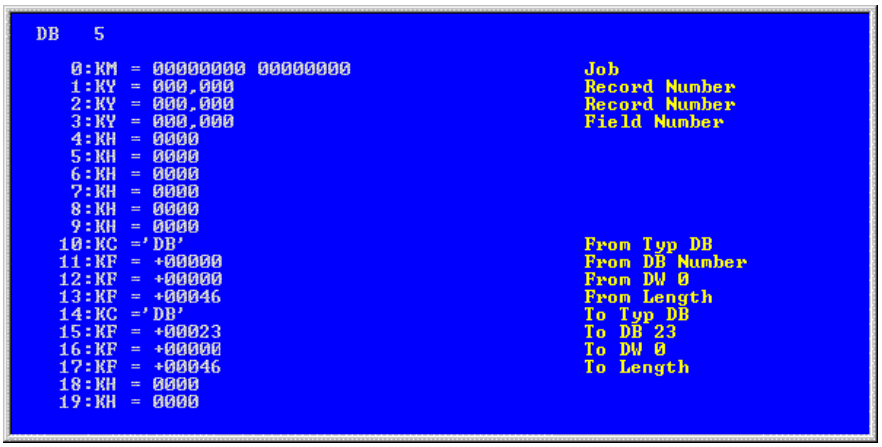
Installing the Hardware

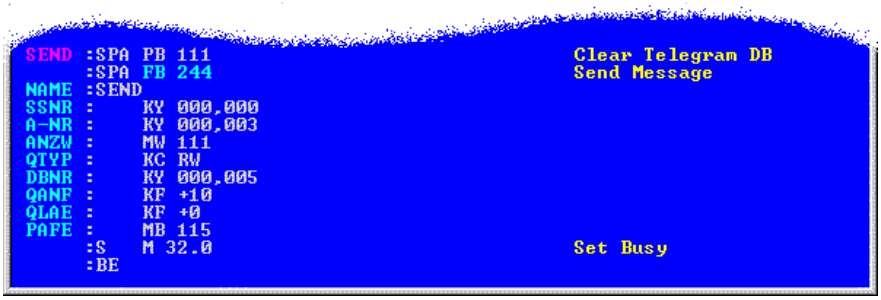
Step	Procedure: Installing the Hardware and Software
1	Rack-mounting of the modules used. In this sample, the modules to be installed are the power supply *PS 951*, the CPU module *CPU 944* and the communication processor *CP 1430 TF*. Establishing the connection from the programming device to the programming interface of the CPU module. Establishing the connection from the communication processor *CP 1413* in the computer to the communication processor *CP 1430 TF* in the PLC.

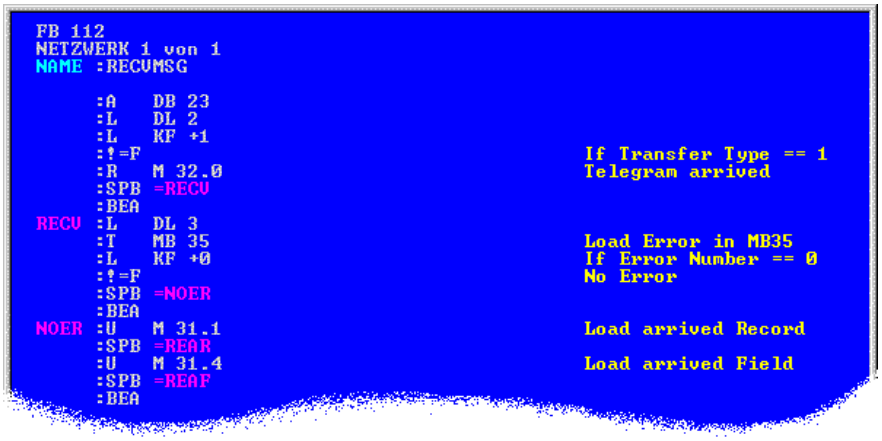
Creating the STEP5 Program

Step	Procedure: Creating the STEP5 Program
1	Creation of a new project with the STEP5 software. Start the STEP5 software. From the *Object* → *Project* → *Settings* → *Page1* and *Page2* menus, define the settings for the new project. In the *Program File* field, specify the name of the new program file to be created. In this sample, the name *UA_S5_ST.S5D* is used. Only the first six characters of the file name can be changed by the user.
2	Programming of the startup blocks. For the *SIMATIC S5 115U* PLC used in this sample, these are the organization blocks *OB21* and *OB22*. During the startup of the PLC, the communication processor *CP 1430 TF* must be synchronized. This is done by the data handling block *SYNCHRON*. For the *SIMATIC S5 115U* PLC used in this sample, this is the block *FB249*. As parameters, the interface number of the communication processor, the desired block size as well as a parameterization error byte are transferred to this block. The following depicts the completed *OB21* organization block. 
3	Creation of a program block for the passive communication of the PLC. In STEP5, the creation of a new program block is carried out via the *Editor* → *STEP5 Block* → *menus of the program file*. As the name of the program block, this sample uses *PB100*. The passive communication is carried out by calling the handling blocks *SEND* and *RECEIVE*. For the *SIMATIC S5 115U* PLC used in this sample, these are the blocks *FB244* and *FB245*. As the job number *A-NR*, *0* is assigned to these blocks to allow execution of the *SEND ALL* and *RECEIVE ALL* functions. The *PB100* program block must be called in the cyclic program (in the *OB1*). The following depicts the completed *PB100*. 

Step	Procedure: Creating the STEP5 Program
4	Creation of data blocks for the telegram data. In STEP5, this is accomplished via the *Editor* → *Data Block* → *menus of the program file*. As the names of the data blocks, this sample uses *DB25* to *DB30*. These data blocks contain the telegram data of the various jobs. For each job implemented, this sample uses a separate data block. The data ranges for the record and field numbers initially remain empty. They are only filled with current values just before the telegram is sent. Likewise, possibly required job data is only entered before the telegram is sent. The following depicts the completed *DB26* data block. This block contains the telegram data for the job of reading a data record from the *User Archives*. 
5	Creation of the data block for the active communication with the *User Archives*. As the name of the data block, this sample uses *DB23*. In this data block, the *WinCC Raw Data Tag* created for the *User Archives* is addressed. The job telegrams are send by this data block, WinCC sends the acknowledgment telegrams to this data block. The size of the data block depends on the maximum telegram length. In this sample, this is a length of 46 Bytes, which occurs during a write job of a data record into the archive. This length consists of a 16 Byte telegram header, a 12 Byte job and 18 Bytes of job data.
6	Creation of a program block for deleting the *DB23*. As the name of the program block, this sample uses *PB111*. This program block is called, before a new job telegram is sent to the *User Archives*.

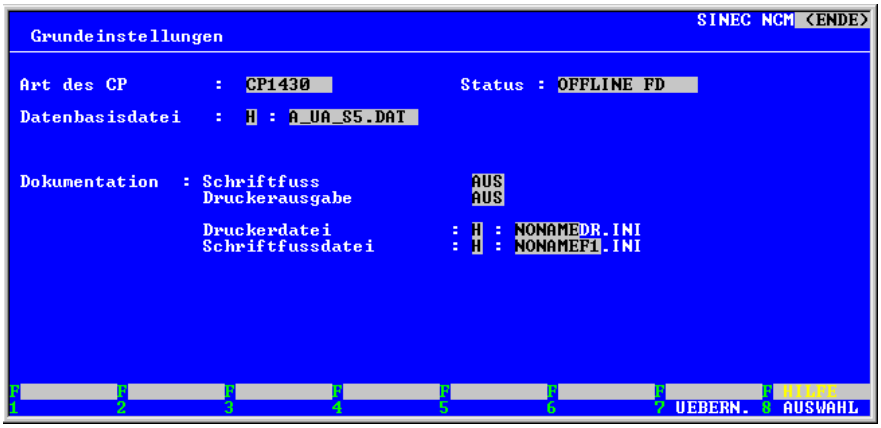
Step	Procedure: Creating the STEP5 Program
7	Creation of two additional data blocks. As the names of the data blocks, this sample uses *DB5* and *DB21*. The *DB21* contains the data of a data record. This data corresponds to the data record currently requested by the *User Archives*. The data is visualized using WinCC tags and can be changed in runtime for simulation purposes. The *DB5* contains data ranges for a control tag, the field number as well as the record number. All three values are visualized using WinCC tags and be changed in runtime for simulation purposes. In addition, the *DB5* contains the send data for the *SEND* handling block, which carries out the sending of job telegrams to the *User Archives*. From where the telegram data originates (in the illustration below, the line *From DB Number*) is entered just before the telegram is sent - once the job type has been defined. The following depicts the completed *DB5* data block. 

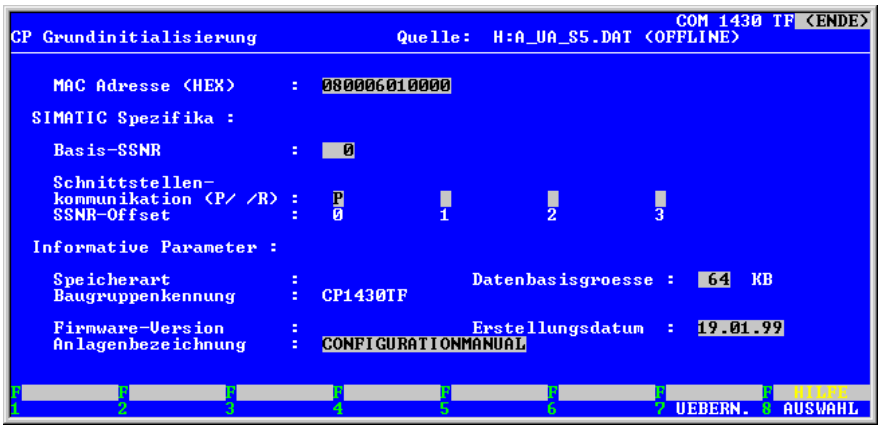
Step	Procedure: Creating the STEP5 Program
8	Creation of a function block, which carries out the sending of the job telegrams to the *User Archives*. In STEP5, the creation of a new function block is carried out via the *Editor* → *STEP5 Block* → *menus of the program file*. As the name of the function block, this sample uses *FB111*. It is named *SENDMSG*. In the function block, the control tag stored in the *DB5* is transferred to the *MW30* and is then reset to 0. The control values have been defined in such a way that to trigger a job, one of the flags *M31.0* to *M31.5* is set. If none of these flags are set, the block is terminated. If one of the flags is set, a jump to a job-specific point is made. Here, the data block number containing the respective job telegram is entered in the *DB5*. In addition, the job telegram is initialized. If necessary, the record number, field number as well as the job data to be sent is entered. In this case, the byte order must be reversed for number values.  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 :U M 31.1 — Read Record :SPB =REAR :U M 31.2 — Write Record :SPB =WRIR :U M 31.3 — Delete Record :SPB =DELR :U M 31.4 — Read Field :SPB =REAF :U M 31.5 :SPB =WTMP :BEA — No Job to execute DELA :A DB 5 :L KF +25 :T DW 11 :SPA =STMP Once the job-specific program part has been completed, a jump to another point is made. Here, the program block for deleting the communication data block *DB23* is called. Subsequently, the job telegram is sent to the *DB23* via the SEND handling block. After sending the job telegram, the flag *M32.0* is set. It prevents the sending of an additional job telegram for as long as the acknowledgment telegram from WinCC is still outstanding.  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 QTYP : KC RW DBNR : KY 000,005 QANF : KF +10 QLAE : KF +0 PAFE : MB 115 :S M 32.0 — Set Busy :BE The illustrations used in this step correspond to sections from the described *FB111* function block. Due to the length of this block, the entire program is not displayed.

Step	Procedure: Creating the STEP5 Program
9	Creation of a function block, which carries out the receiving of the acknowledgment telegrams of the *User Archives*. As the name of the function block, this sample uses *FB112*. It is named *RECVMSG*. The function block checks the transfer type entered in the *DB23*. As soon as the transfer type has the value 1, an acknowledgment telegram has been received by the *User Archives*. If this is the case, the *M32.0* flag can be reset, which lifts the lock preventing the sending of new jobs. Following that, the error code contained in the acknowledgment telegram is read. If the error code has the value 0, the job has been executed successfully. In the case of a successfully performed job for reading data from the *User Archives*, the data received is stored in the *DB5*. In this case, the byte order must be reversed for number values.  The illustration used in this step corresponds to a section from the described *FB112* function block. Due to the length of this block, the entire program is not displayed.
10	Programming of the cyclic program in the *OB1*. Via the *PB100* program block, the passive communication with WinCC is performed. Via the *FB111* and *FB112* function blocks, the active communication with the *User Archives* is performed. If no job is processed (M32.0 is not set), the *FB111* is used to check whether an acknowledgment telegram has to be sent. If a job is processed (M32.0 is set), the *FB112* is used to check whether an acknowledgment telegram has been received. The following depicts the completed *OB1* organization block. 

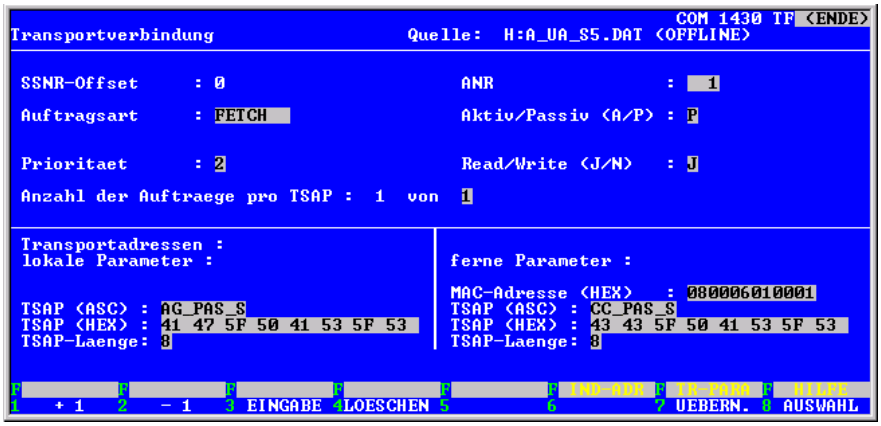
Step	Procedure: Creating the STEP5 Program
11	Loading the STEP5 program into the PLC. In STEP5, this is done via the *Object* → *Blocks* → *Transfer* → *PLC File* menus. In the Selection field, the option *All Blocks* must be selected to load all previously created blocks to the PLC.

Configuring the Communication Processor

Step	Procedure: Configuring the Communication Processor
1	Start the communication package *SINEC NCM for COMs* to configure the communication processor *CP 1430 TF*. From STEP5, start the communication package via the *Change* → *Additional* → *SINEC NCM for COMs* menus.
2	This will open the communication package *SINEC NCM for COMs*. If no database file is set, the *Basic Settings* entry mask will initially be displayed. This entry mask can also be opened via the *File* → *Select* (or *Init.* → *Edit*) menus. In the *CP Type* field, indicate the type of communication processor used. Via the *F8* function key, one of the available communication processors can be set. Select the *CP 1430*. Set the *Status* field to *OFFLINE FD* via the *F8* function key. This stores the configuration made in the program to a database file. In the *Database File* field, specify the name of this database file. This name has to start with the letter *A*. For this sample, the name *A_UA_S5.DAT* is used for the database file. The settings made in the *Basic Settings* entry mask are applied via the *F7* function key. 

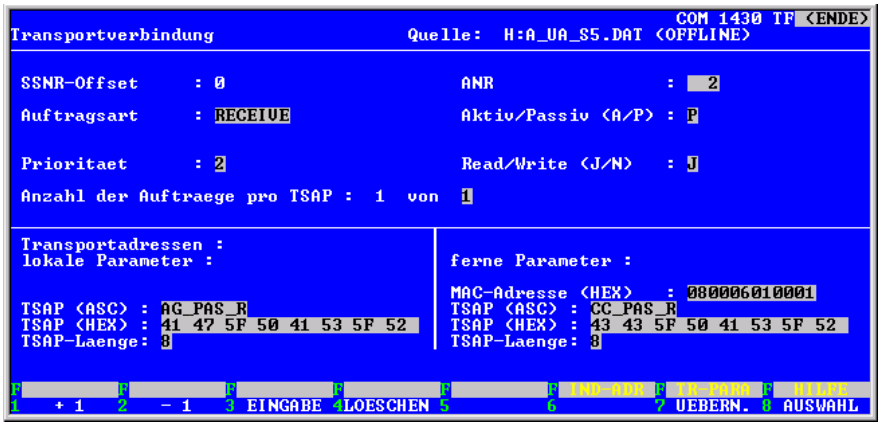
Step	Procedure: Configuring the Communication Processor
3	The settings for the basic initialization of the communication processor must be made. They are entered in the *Basic Initialization* entry mask. This entry mask is opened via the *Edit* → *CP Init.* menus. In the *MAC Address (HEX)* field, the *Ethernet Address* of the communication processor is specified. In this sample, the address *080006010000* has been entered. This address is one of the parameters that has to be set during the creation of the connections in WinCC. The remaining settings can be seen in the following graphic. The settings made in the *Basic Initialization* entry mask are applied via the *F7* function key. 
5	Creation of the transport connections for the passive communication with WinCC. This is done in the *Transport Connection* entry mask. This entry mask is opened via the *Edit* → *Connections* → *Transport Connections* menus. For the passive communication, two connections must be created. One processes the write jobs of WinCC, the other one the read jobs of WinCC. The PLC is set to passive for both connections by entering *P* in the *Active/Passive* field.

Step	Procedure: Configuring the Communication Processor
	For the connection used to process the read jobs from WinCC, this sample keeps *1* as the value of the job number *ANR*. In the *Job Type* field, *FETCH* is specified. In the *Transport Addresses* area, *TSAP* with *AG_PAS_S* is set for the *Local Parameter* and *TSAP* with *CC_PAS_S* for the *Remote Parameter* in ASCII-Code. The remote parameter also requires the specification of the Ethernet address in the *MAC Address* field that has been entered for the communication processor *CP 1413* in the computer. In this sample, the address *080006010001* has been set during the installation of the communication processor *CP 1413*.

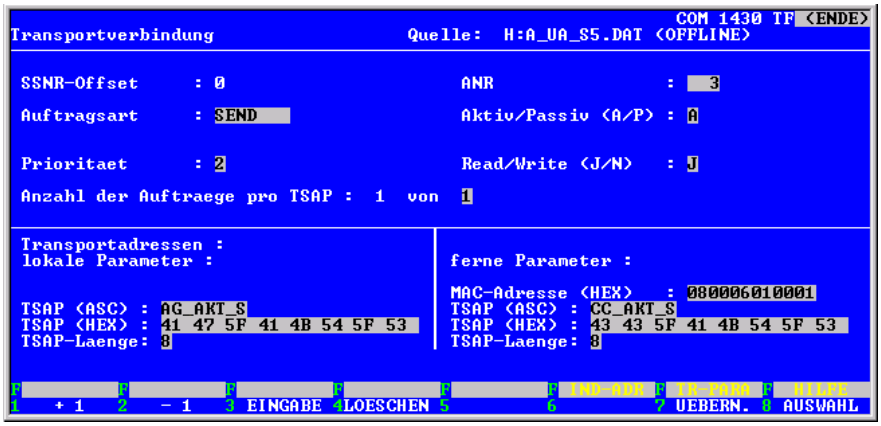
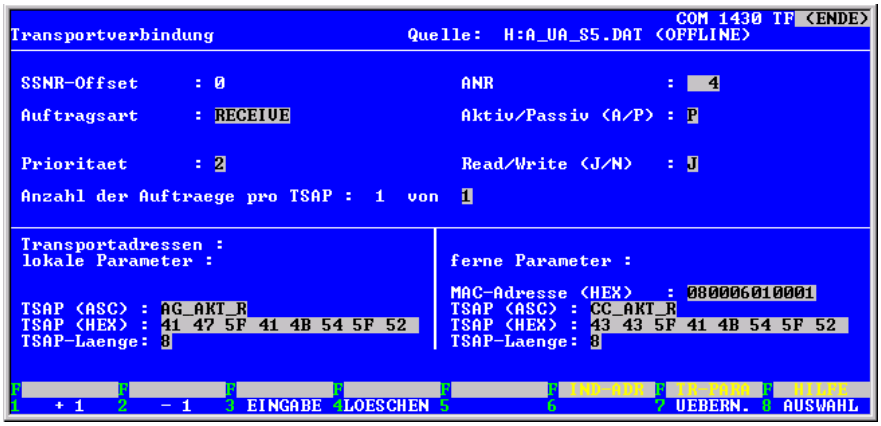


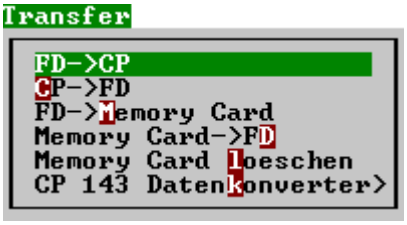
By hitting the *F3* function key, the parameters for the next transport connection can be entered. This transport connection will process the write jobs of WinCC.

In this sample, the value *2* for the job number *ANR* is kept. In the *Job Type* field, *RECEIVE* is specified. In the *Transport Addresses* area, *TSAP* with *AG_PAS_R* is set for the *Local Parameter* and *TSAP* with *CC_PAS_R* for the *Remote Parameter* in ASCII-Code. For the remote parameter, also enter the Ethernet address of the communication processor *CP 1413* from the computer.



The connection parameters just defined must be set in the WinCC project while creating the connection *S5-115U-WinCC-Active*. For the TSAP values set, note that there is a difference between an entered blank space and no entered character. Always check the hexadecimal code.

Step	Procedure: Configuring the Communication Processor
6	Creation of the transport connections for the active communication with WinCC. For the active communication of the PLC, two connections must be created. One processes the write jobs of the PLC, the other one the write jobs of WinCC. By hitting the *F3* function key, the parameters for the next transport connection can be entered. For the connection used to process the write jobs of the PLC, this sample keeps *3* as the value of the job number *ANR*. In the *Job Type* field, *SEND* is specified. The PLC is set to active by entering *A* in the *Active/Passive* field. In the *Transport Addresses* area, *TSAP* with *AG_AKT_S* is set for the *Local Parameter* and *TSAP* with *CC_AKT_S* for the *Remote Parameter* in ASCII-Code. For the remote parameter, also enter the Ethernet address of the communication processor *CP 1413* from the computer.  By hitting the *F3* function key, the parameters for the next transport connection can be entered. For the connection used to process the write jobs of WinCC, this sample keeps *4* as the value of the job number *ANR*. In the *Job Type* field, *RECEIVE* is specified. The PLC is set to passive by entering *P* in the *Active/Passive* field. In the *Transport Addresses* area, *TSAP* with *AG_AKT_R* is set for the *Local Parameter* and *TSAP* with *CC_AKT_R* for the *Remote Parameter* in ASCII-Code. For the remote parameter, also enter the Ethernet address of the communication processor *CP 1413* from the computer.  The connection parameters just defined must be set in the WinCC project while creating the connection *S5-115U-WinCC-Passive*. The settings made in the *Transport Connection* entry mask are applied via the *F7* function key.

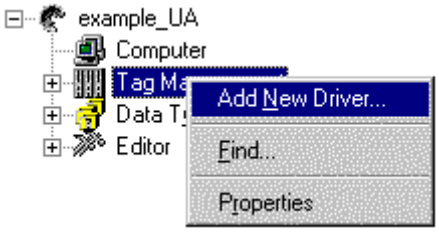
Step	Procedure: Configuring the Communication Processor
7	Loading the configuration data of the database file to the communication processor *CP 1430 TF*. This is done via the *Transfer* → *FD->CP* menus. The configuration data can only be uploaded while the communication processor is in the *STOP* operating mode. 

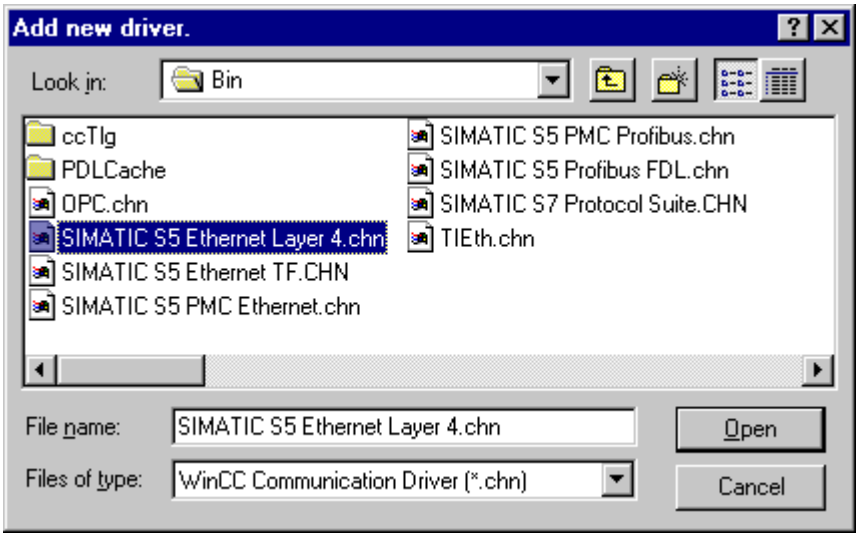
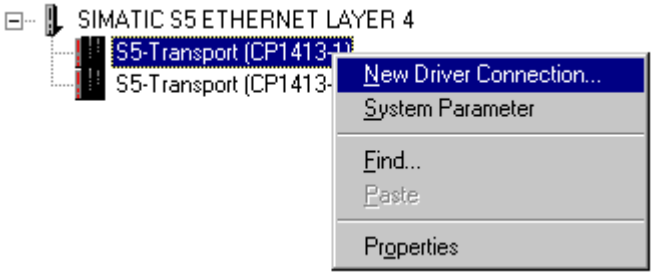
Starting up the PLC

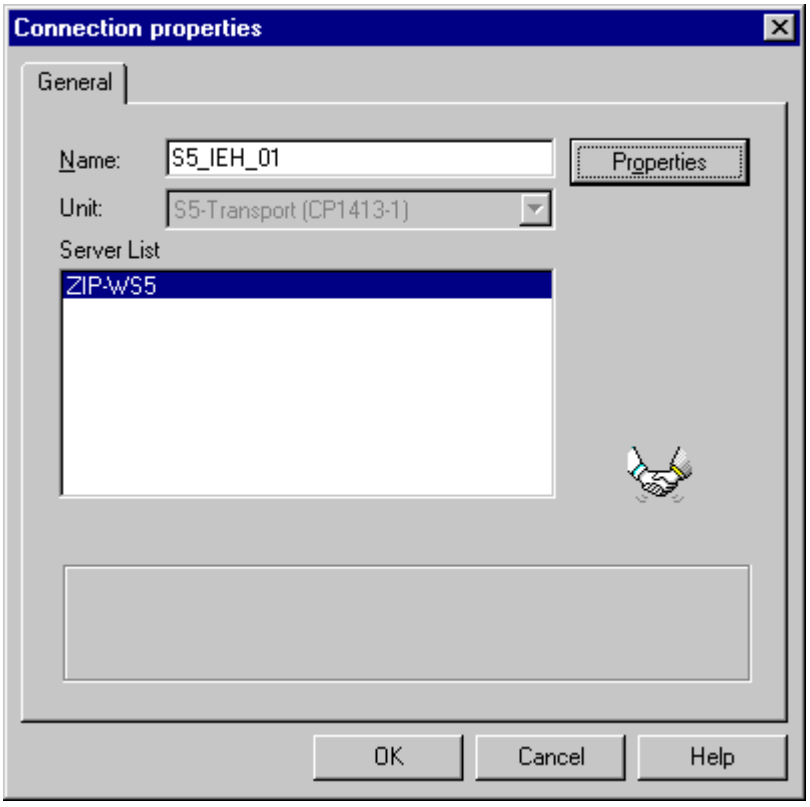
Step	Procedure: Starting up the PLC
1	Starting the individual modules of the PLC. Previously, the STEP5 program and the database file of the communication processor *CP 1430 TF* must have been loaded to the PLC. First, the operating mode switch of the communication processor is set to the RUN position. The status LEDs *RUN* and *STOP* will light up at the communication processor, indicating that the module has not be synchronized. Next, the operating mode switch of the CPU module is set to the *RN* position. During the startup of the CPU module, the communication processor is synchronized by the startup block. The communication processor's status LED *STOP* go out. At the CPU module, only the status LED *RN* will be illuminated.

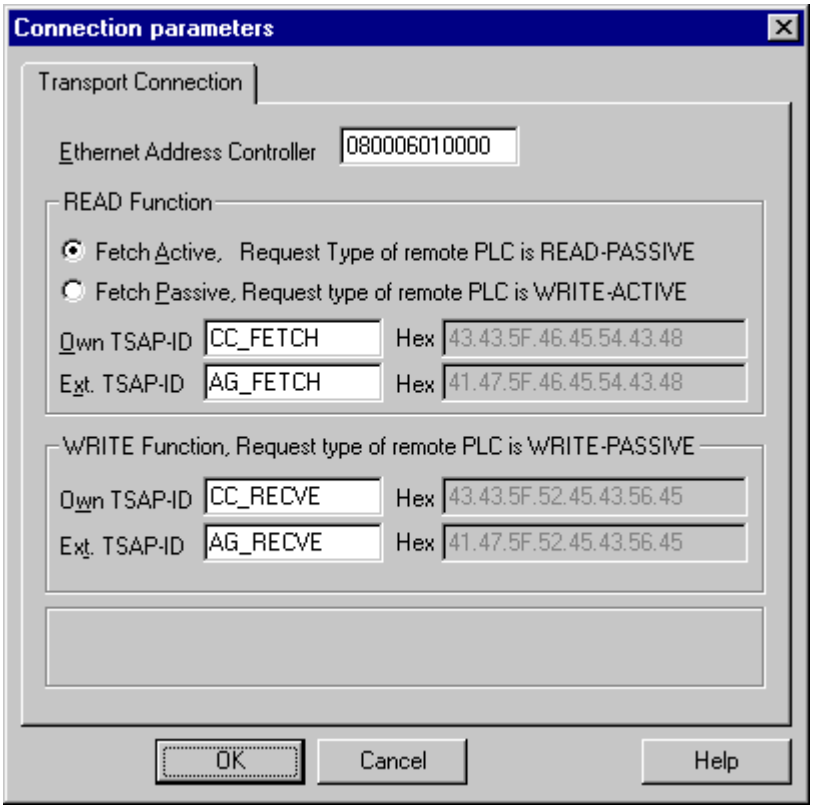
5.1.7.3 Configuration in WinCC

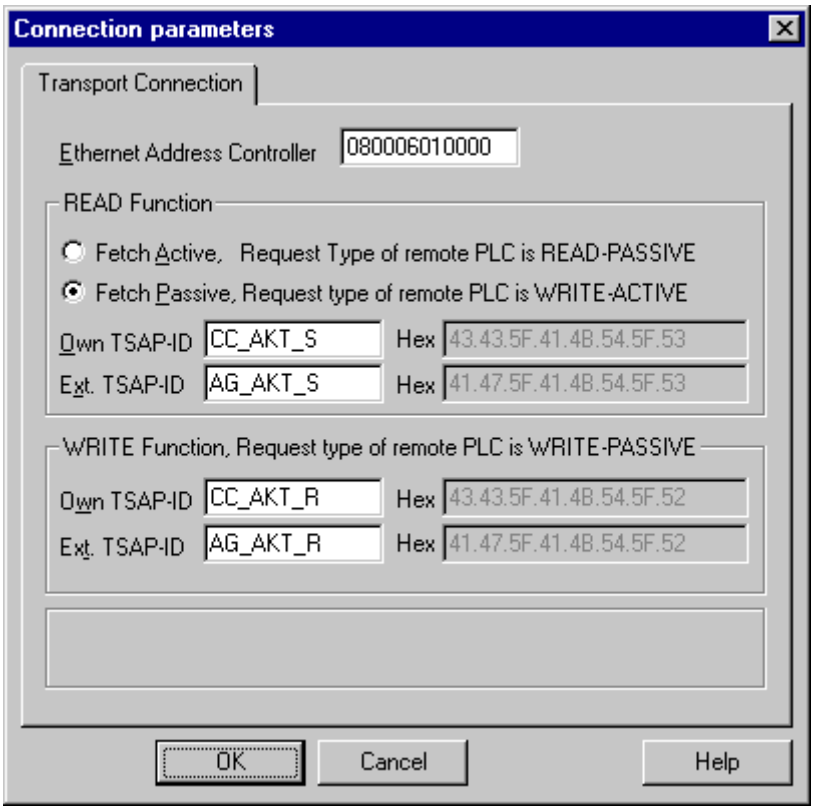
Creating the Connections

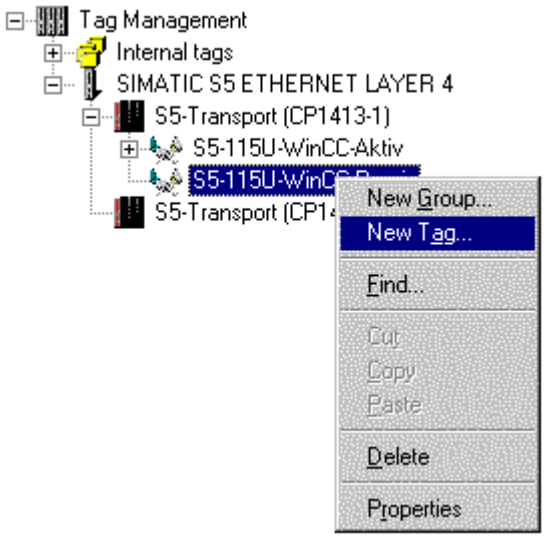
Step	Procedure: Creating the Connections
1	Installation of the required communication driver. This is performed via a R on *Tag Management* and selecting *Add New Driver* from the pop-up menu. 

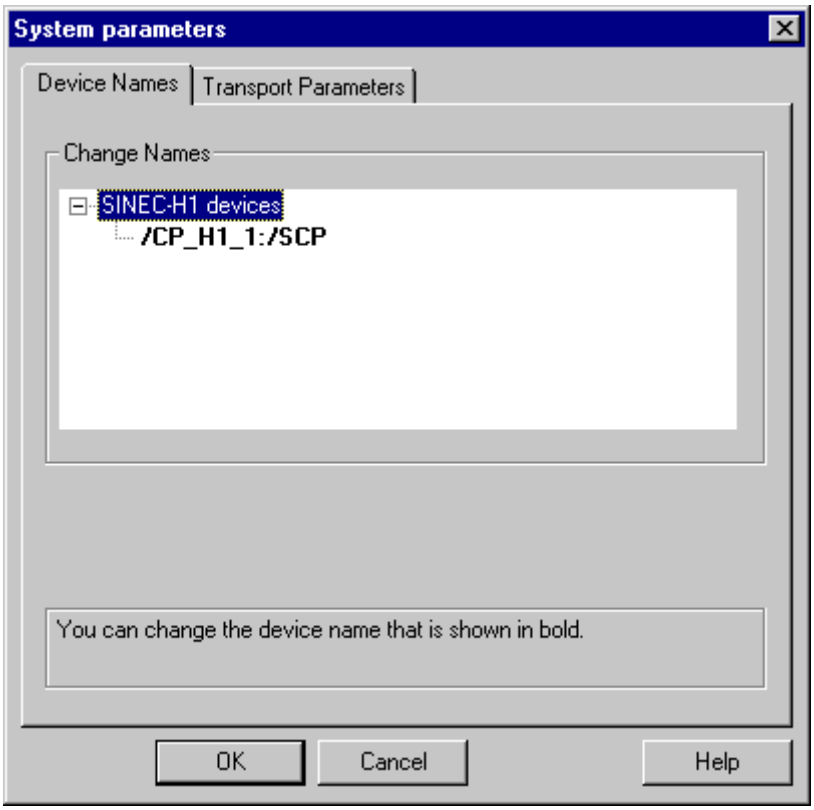
Step	Procedure: Creating the Connections
2	The dialog *Add New Driver* will be displayed. This dialog lists all communication drivers that can be installed. For the communication to the *SIMATIC S5* via *Industrial Ethernet*, the driver *SIMATIC S5 Ethernet Layer 4* is installed. Select this driver from the dialog. Exit the dialog by clicking on *Open*. 
3	The newly added driver *SIMATIC S5 Ethernet Layer 4* will be displayed as a sub-entry to *Tag Management*. The driver contains two channel units. The second channel unit is needed if two *CP 1413* communication processors are operated in the computer. A new connection for the *S5-Transport (CP1413-1)* channel unit is created by R on *S5-Transport (CP1413-1)* and then selecting *New Driver Connection* from the pop-up menu. This connection is used for the active sending of data from the PLC. In this case, WinCC is the active partner. 

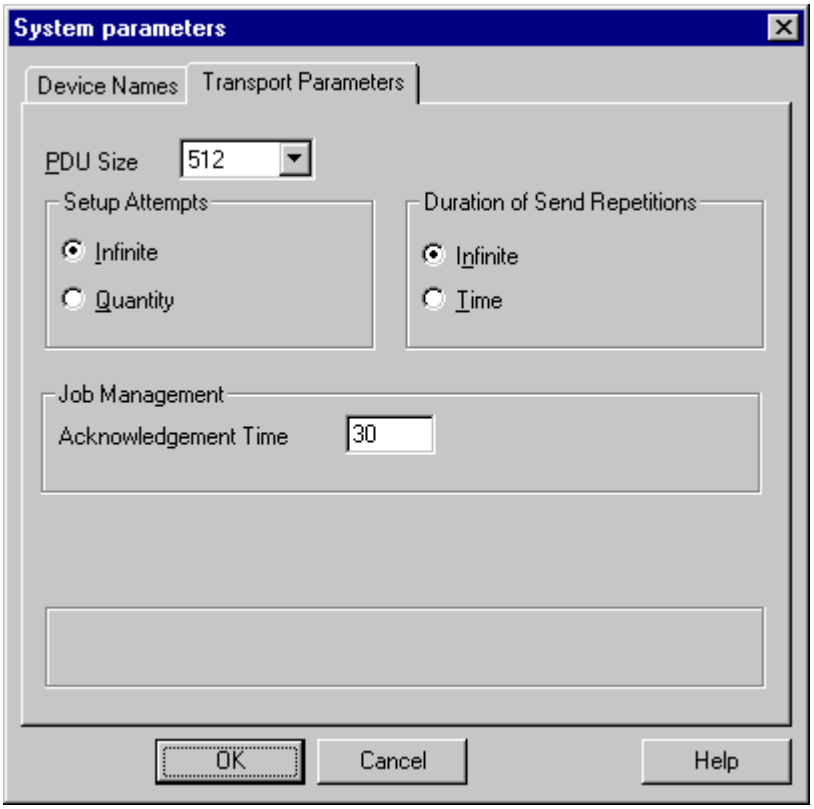
Step	Procedure: Creating the Connections
4	The properties dialog of the connection will be displayed. In the *General* tab, the *Name* of the new connection is entered. In the sample, this is *S5-115U-WinCC-Active*. Click on the *Properties* button to define the connection properties. 

Step	Procedure: Creating the Connections
5	The dialog *Connection Properties* will be displayed. In the *Transport Connection* tab, the parameters of the desired communication connection are defined. In the *PLC Ethernet Address* field, the Ethernet address specified for the PLC is entered. In this sample, the address *080006010000* has been defined during the configuration of the communication processor *CP 1430 TF*. In the *READ Function* area, the connection settings for reading data from the PLC are made. In order for WinCC to request the data actively, the radio-button *Fetch Actively* must be selected. For the local and remote TSAPs, the values defined during the creation of the transport connections for the communication processor *CP 1430 TF* are entered. In this sample, the value *CC_PAS_S* is entered in the *Local TSAP* field and the value *AG_PAS_S* in the *Remote TSAP* field. In the *WRITE Function* area, the connection settings for writing data to the PLC are made. In this sample, the value *CC_PAS_R* is entered in the *Local TSAP* field and the value *AG_PAS_R* in the *Remote TSAP* field. Close the dialog by clicking on *OK*. Also close the *Connection Properties* dialog by clicking on *OK*. 

Step	Procedure: Creating the Connections
6	Creation of a second connection for the *S5-Transport (CP1413-1)* channel unit by R on *S5-Transport (CP1413-1)* and then selecting *New Driver Connection* from the pop-up menu. This connection is used by the PLC for actively writing data. In this case, WinCC is the passive partner. The properties dialog of the connection will be displayed. In the *General* tab, the *Name* of the new connection is entered. In the sample, this is *S5-115U-WinCC-Passive*. Via the *Properties* button, the connection properties are defined. The dialog *Connection Properties* will be displayed. In the *Transport Connection* tab, the parameters of the desired communication connection are defined. In the *PLC Ethernet Address* field, the Ethernet address specified for the PLC is entered. In this sample, the address *080006010000* has been defined during the configuration of the communication processor *CP 1430 TF*. In the *READ Function* area, the *Fetch Passive* radio-button is selected. This makes WinCC the passive partner. The sending of data is initiated by the PLC. For the local and remote TSAPs, the values defined during the creation of the transport connections for the communication processor *CP 1430 TF* are entered. In this sample, the value *CC_AKT_S* is entered in the *Local TSAP* field and the value *AG_AKT_S* in the *Remote TSAP* field. In the *WRITE Function* area, the connection settings for writing data to the PLC are made. In this sample, the value *CC_AKT_R* is entered in the *Local TSAP* field and the value *AG_AKT_R* in the *Remote TSAP* field. Close the dialog by clicking on *OK*. Also close the *Connection Properties* dialog by clicking on *OK*. 

Step	Procedure: Creating the Connections
7	The just created connections will now be displayed in the *WinCC Explorer*. 

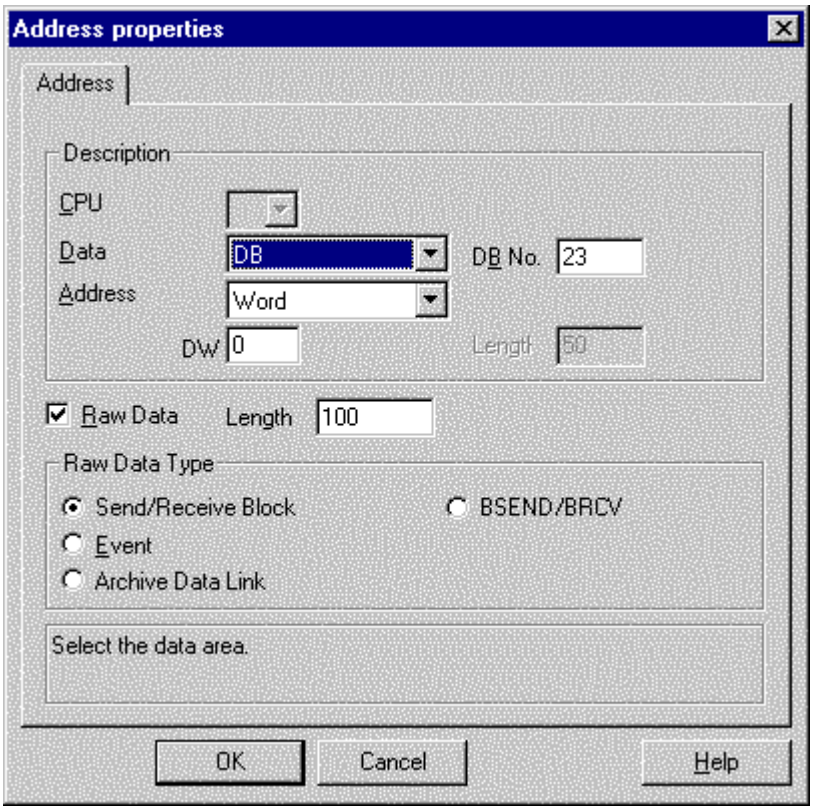
Step	Procedure: Creating the Connections
8	Setting the system parameters of the channel unit. These settings are made in the System Parameters dialog, which is accessed via a R on the *S5-Transport (CP1413-1)* entry and then selecting *System Parameters* from the pop-up menu. In the displayed dialog, the name of the access point, which is used by WinCC to access the PLC, can be changed. By default, the access point *CP_H1_1:* is set. Previously, during the installation of the communication processor in the computer, the *CP 1413* has been assigned to the access point *CP_H1_1:*. 

Step	Procedure: Creating the Connections
9	In the *Transport Parameters* tab, various settings affecting the communication can be edited. In this sample, the default settings are kept. Close the dialog by clicking on *OK*. 

Creating the Tags

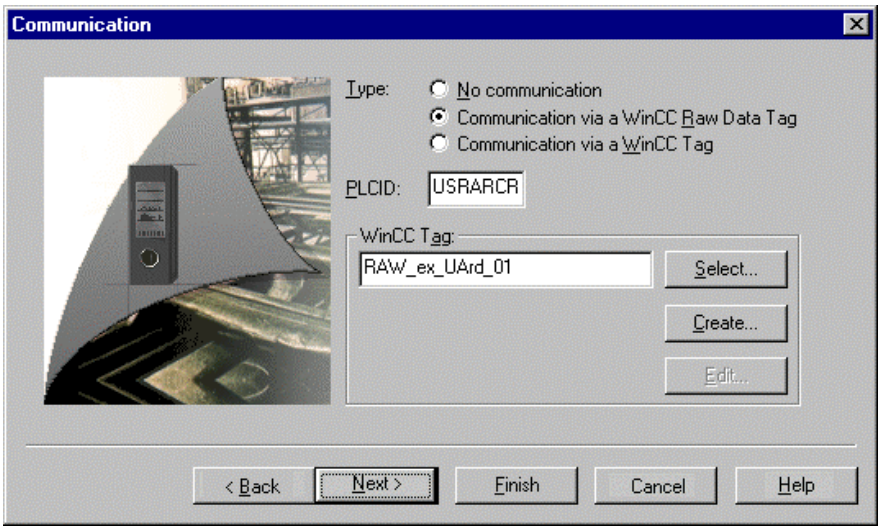
Step	Procedure: Creating the Tags
1	Creation of the tags for the *S5-115U-WinCC-Active* connection. This is done in the *WinCC Explorer* via a R on the corresponding connection entry and then selecting *New Tag* from the pop-up menu. The names, data types and addresses of the individual tags are listed in the following illustration. The *T08w_ex_UArd_RecordName*, *S32w_ex_UArd_Pressure* and *S32w_ex_UArd_Temperature* tags represent the data record currently in the PLC, which is stored in *DB21*. The *U32w_ex_UArd_Record, U16w_ex_UArd_Field* and *U16w_ex_UArd_Job* tags represent the record number and field number to be processed as well as the control tag, which are stored in *DB5*. The *U08w_ex_UArd_Busy* and *U08w_ex_UArd_Error* tags are used to display the current job status.

Name	Type	Parameters
S32w_ex_UArd_Temperature	Signed 32-bit value	DB21,DD5
S32w_ex_UArd_Pressure	Signed 32-bit value	DB21,DD7
T08w_ex_UArd_RecordName	Text tag 8-bit character set	DB21,DW0
U16w_ex_UArd_Job	Unsigned 16-bit value	DB5,DW0
U16w_ex_UArd_Field	Unsigned 16-bit value	DB5,DW3
U32w_ex_UArd_Record	Unsigned 32-bit value	DB5,DD1
U08w_ex_UArd_Busy	Unsigned 8-bit value	MB32
U08w_ex_UArd_Error	Unsigned 8-bit value	MB35

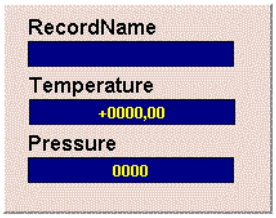
Step	Procedure: Creating the Tags
2	Creation of a tag for the *S5-115U-WinCC-Passive* connection. This is done in the *WinCC Explorer* via a R on the corresponding connection entry and then selecting *New Tag* from the pop-up menu. The properties dialog of the tag will be displayed. In the sample, the *Name* of this tag is *RAW_ex_UArd_01*. As the *Data Type* of this tag, *Raw Data Type* is set. Via the *Select* button, the dialog for addressing the tag is opened. In the *Address Description* field, the *Data Range DB* and the *DB No. 23* are set for this sample. As the *Address*, *Word* is entered and as the *DW, 0*. The *Raw Data* check-box is selected and the *Length* of the tag is set to *46*. In the *Raw Data Type* field, the radio-button *Send/Receive Block* is selected. The *Address Properties* dialog is closed with *OK*. The *Tag Properties* dialog is closed with *OK* as well. 

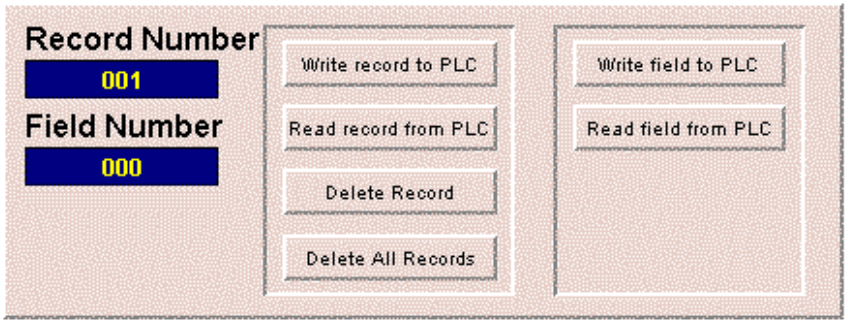
Creating the User Archives

Step	Procedure: Creating the User Archives
1	Open the *User Archives Editor*. In this editor, a new archive is created via a R on the *Archives* entry and then selecting *New Archive* from the pop-up menu. This starts a Wizard for the archive creation. 

Step	Procedure: Creating the User Archives
2	On the first page of this Wizard, the *Archive Name* is entered. In the sample, the name *UserArchive_rd* is entered in the *Archive Name* field. The *Alias* field is left blank. As the *Archive Type*, *Unlimited* is selected. Continue to the next page by clicking on *Next*.
3	On the second Wizard page, the radio-button *Communication via WinCC Raw Data Tag* is selected. As the *PLCID*, this sample uses the name *USRARCRD*. This ID must not exceed 8 characters. Via this ID, a job telegram sent by the PLC can be assigned to a certain archive. Via the *Select* button, the *WinCC Raw Data Tag* configured for the archive communication is set. In this sample, this is the *RAW_ex_UArd_01* tag. Since no settings are made for this sample on the next page, the Wizard can be exited by clicking on *Finish*. 
4	For the just created archive, the fields listed in the following table are created. Save the settings made. Name / Alias / Type / Length Pressure — Number (integer) RecordName — String — 10 Temperature — Number (double)
5	In the lower table window of the *User Archives Editor*, multiple data records can be now be created for the archive. 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 ...

Implementation in the Graphics Designer

Step	Procedure: Implementation in the Graphics Designer
1	Open the *Graphics Designer* and create a new picture. In this sample, this is the *ex_3_chapter_01a.pdl* picture. Configuration of the Control used to display the data. This is the *WinCC User Archives - Table Element*. It is selected from the *Object Palette's Control* selection menu and then placed in the picture.
2	Via a D on the just created *Control1* object, its properties dialog is opened. In the *General* tab, the previously configured archive *UserArchive_rd* is set in the *Source* field. In the *Edit* field, the *Insert*, *Change* and *Delete* check-boxes are selected to make all editing options available to the user. In addition, the *Border* check-box is selected. In the *Toolbar* tab, the two check-boxes for the buttons *Write Tags* and *Read Tags* are deselected. In this sample, the control is carried out by the PLC. All other buttons of the toolbar are used. In the *Fonts* tab, the *Size* of the font is reduced to *10* in order to display all columns simultaneously in runtime. The settings made in the properties dialog of the *WinCC User Archives - Table Element* are concluded via the *OK* button. In this sample, the color scheme of the table was matched to the project's color scheme via the properties dialog of the *Control1* object.
3	To display the data record currently loaded in the PLC, three *Smart Objects* → *I/O Fields* are configured. In this sample, these are the *I/OField1*, *I/OField2* and *I/OField3* objects. For the *I/O Fields*, a *Tag Connection* each at *Properties* → *Output/Input* → *Output Value* to one of the three tags *T08w_ex_UArd_RecordName*, *S32w_ex_UArd_Temperature* and *S32w_ex_UArd_Pressure* is created. 

Step	Procedure: Implementation in the Graphics Designer
4	To specify the record number and field number to be processed, two additional *Smart Objects* → *I/O Fields* are configured. In this sample, these are the *I/OField4* and *I/OField5* objects. For the *I/OField4* object, a *Tag Connection* at *Properties* → *Output/Input* → *Output Value* to the *U32w_ex_UArd_Record* tag is created. This I/O field displays the record number. For the *I/OField5* object, a *Tag Connection* at *Properties* → *Output/Input* → *Output Value* to the *U16w_ex_UArd_Field* tag is created. This I/O field displays the field number. To trigger actions in the PLC, five *Windows Objects* → *Buttons* are configured. In this sample, these are the *Button9* to *Button13* objects. If one button is pressed, a different value is written to the *U16w_ex_UArd_Job* tag. In the STEP5 program, this tag is queried cyclically. Depending on the content of this tag, a corresponding job is triggered. Each value corresponds to a different job type for the PLC. The assignment of the individual values to the corresponding jobs is listed below. For the *Delete All Records* job implemented in the STEP5 program, no button has been configured. • 1 = Delete All Records • 2 = Read Record from the Archive • 4 = Write Record to the Archive • 8 = Delete Record from the Archive • 16 = Read Field from the Archive • 32 = Write Field to the Archive Via the *Reset* button, the tag *U08w_ex_UArd_Busy* can be reset. This permits a lock preventing the sending of a new job telegram to be removed if it has not been lifted automatically (only relevant in the case of errors). In the *Status* field, the current status of the communication connection as well as the archive communication is displayed using a *C-Action*. This *C-Action* evaluates the *U08w_ex_UArd_Error* tag. The *C-Action* itself is described following this step. 

C-Action for Displaying the Status

```
#include "apdefap.h"
 char* _main(char* lpszPictureName, char* lpszObjectName, char* lpszPropert
{

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";
}

}
```

- The *C-Action* has been created at *Properties* → *Font* → *Text* of the *StaticText3* object. The *Trigger* of the *C-Action* is set to *2 s*. This results in a status check of the connection and the archive communication every 2 seconds.
- The status check is performed by the *GetTagByteStateWait* function. The current connection status is written to the *dwState* tag, the archive communication status to the *byError* tag (error code of the acknowledgment telegram from WinCC).
- Before this action is performed, the background color of the *LED* object is set to red. Once the function has concluded, the background color of the *LED* object is reset to gray.
- Evaluation of the *dwState* and *byError* tags and return of a corresponding status text.

Note for the General Application

The following adaptations must be made before the general application:

- The communication configurations made can be applied directly to your own applications, provided that the same hardware is used. Otherwise, adaptations must be made.
- The structure of the archive must be adapted to meet your own requirements.
- The organization type of the STEP5 program can be reused. However, a single data block for recording the job telegram can be used. In this case, in addition to the record number, field number and job data, the remaining job parameters are entered in this data block via the STEP5 program.

5.1.8 Communication to the SIMATIC S7 via WinCC Raw Data Tags (ex_3_chapter_01b.pdl)

The STEP7 project created in this chapter can also be copied directly from the online document to your hard drive. By default, it will be stored to the *C:\Configuration_Manual* folder.


S7_UserArchive

Task Definition

A communication connection between a SIMATIC S7 PLC and a WinCC station is to be established. The PLC is to read data from a User Archive created on the WinCC station, write data to and delete data from it.
The data records of the archive each consist of two integer fields as well as a text field for recording the data record name.

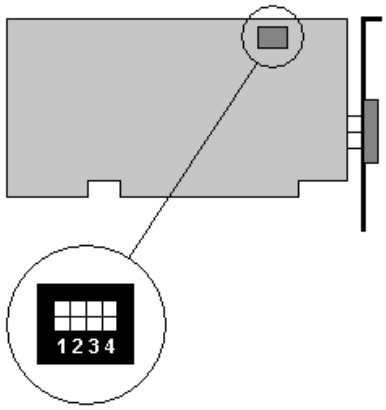
Implementation Concept

In this sample, a *SIMATIC S7-400* PLC with a *CPU 416-1* central module is used. The communication to this PLC is established using the *MPI Network*. In the WinCC station, the *CP 5412 A2* communication processor is used for this purpose, in the PLC, the programming interface of the central module is used for the communication.
In WinCC, the *MPI* channel unit of the *SIMATIC S7 PROTOCOL SUITE* communication driver is used. This communication processor supports, among other things, the active sending of data from the PLC.
For the *MPI* channel unit, two connections to the PLC are created. One connection is used for the active sending of data from the PLC. For this connection, a *WinCC Raw Data Tag* for the communication to the *User Archive* is created. In addition, a *User Archive* is created, whose communication is configured via this *WinCC Raw Data Tag*.
For the second connection, WinCC is the active partner. Among other things, this connection is used to simulate the operation of the communication from WinCC. Via various *Buttons*, control bits are set in the PLC, which trigger the sending of job telegrams. Additionally, the present data (the current data record, currently set record number and field number as well as the job status) in the PLC is displayed via *I/O Fields*.
In runtime, the data of the *User Archive* is displayed by a Control in tabular form.

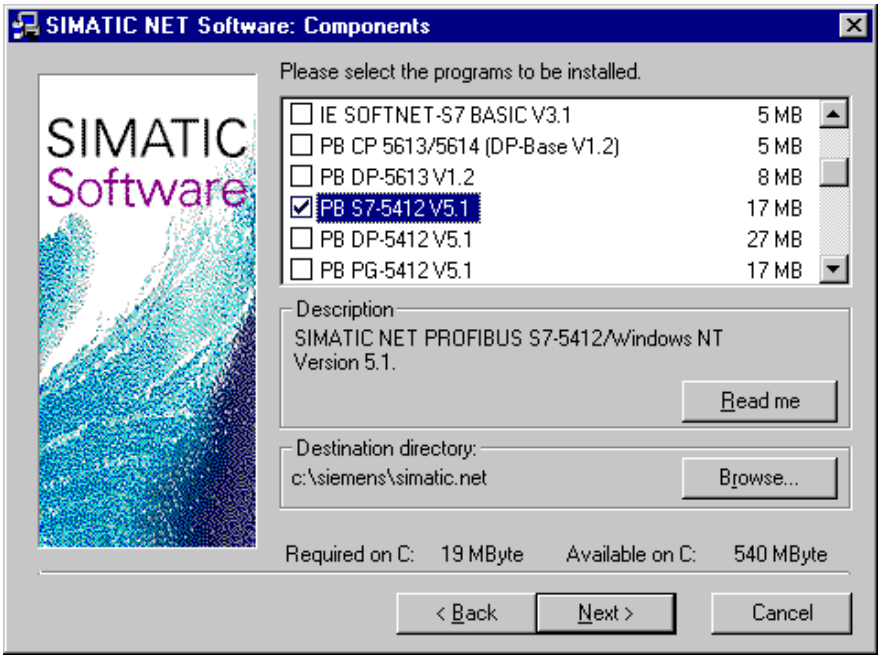
5.1.8.1 Startup of the Communication Processor CP 5412 A2

The following description details the configuration steps necessary to successfully start up the communication processor *CP 5412 A2*.

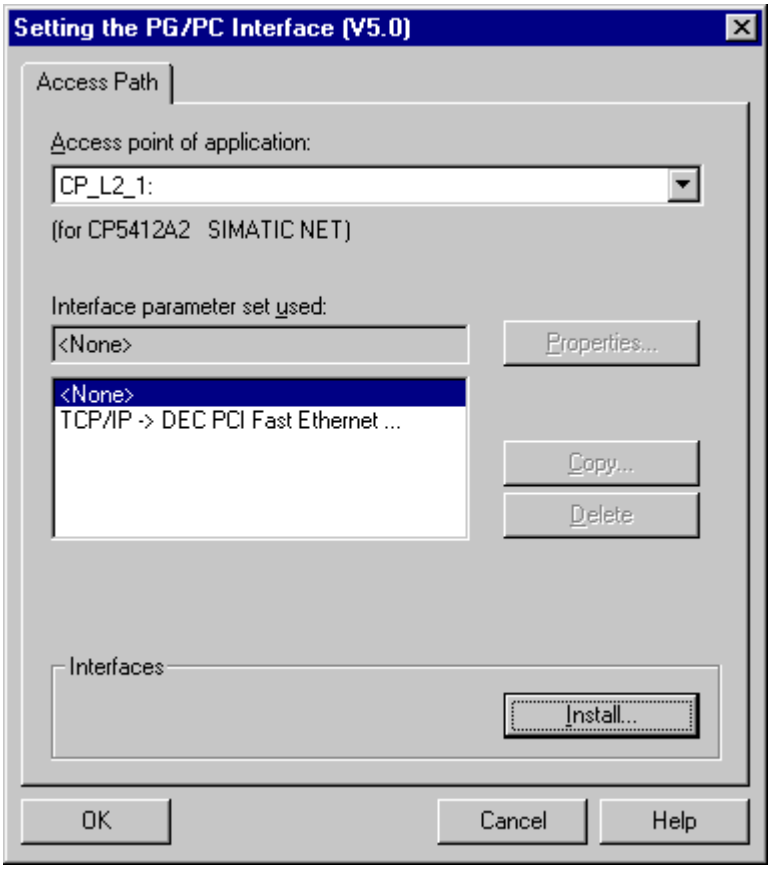
Mounting the Communication Processor in the Computer

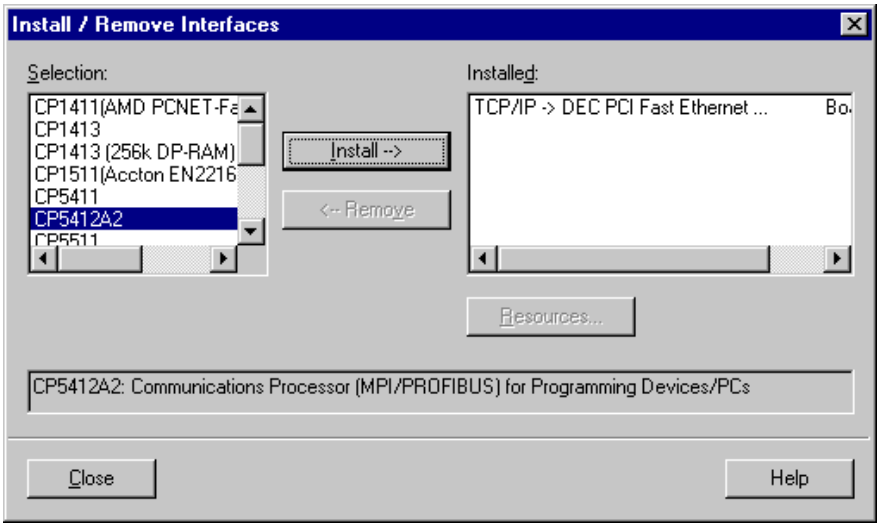
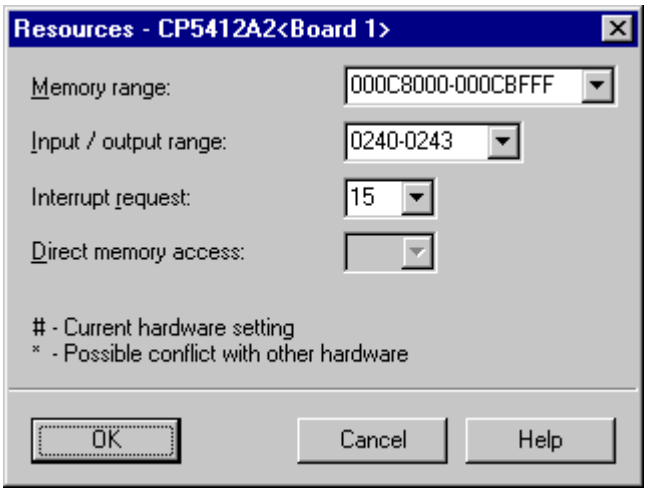
Step	Procedure: Mounting the Communication Processor in the Computer
1	Check the selected jumper settings at the *CP 5412 A2*. During the installation of the *CP 5412 A2*, the *I/O Range* must be specified. The I/O range is set via *Jumper Settings*. By default, the *I/O Range* is set to *0240-0243*. However, other settings are also possible. The following graphic illustrates the jumper settings necessary for the various *I/O Ranges*.  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	Mount the module according to the installation instructions. Among other things, follow the steps for handling electrostatic sensitive devices (ESD). The module must only be installed while the computer is off. For the communication card *CP 5412 A2*, a free ISA slot in the computer is required. After the installation of the *CP 5412 A2*, close the computer's case and start the computer.

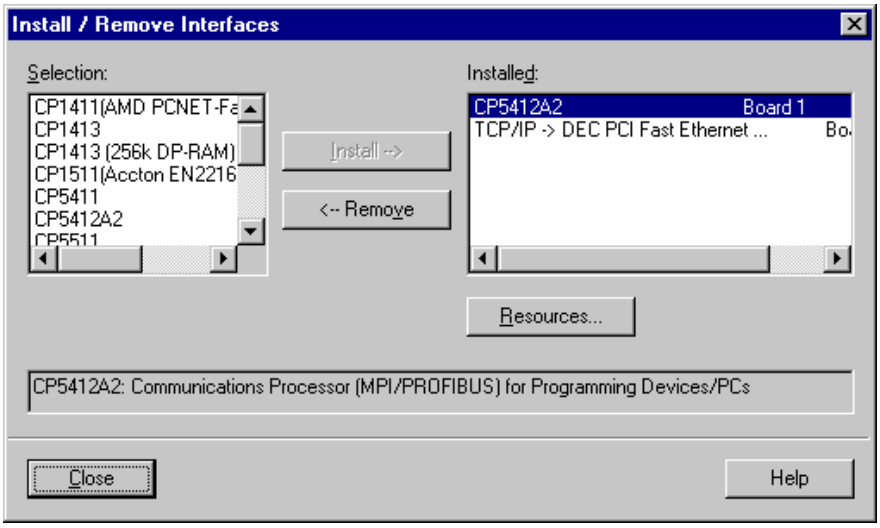
Installing the Communication Driver

Step	Procedure: Installing the Communication Driver
1	Install the communication driver *PB S7-5412* from the *SIMATIC NET* CD-ROM. After inserting the *SIMATIC NET* CD-ROM, the installation program is automatically started. If this is not the case, open the *Windows NT Explorer* and start the *setup.exe* program located on the CD-ROM. The installation of the software is started via the button displayed below.  Follow the instructions of the installation program. On the *Components* page, the check-box of the driver *PB S7-5412* to be installed must be selected. Finish the installation. 

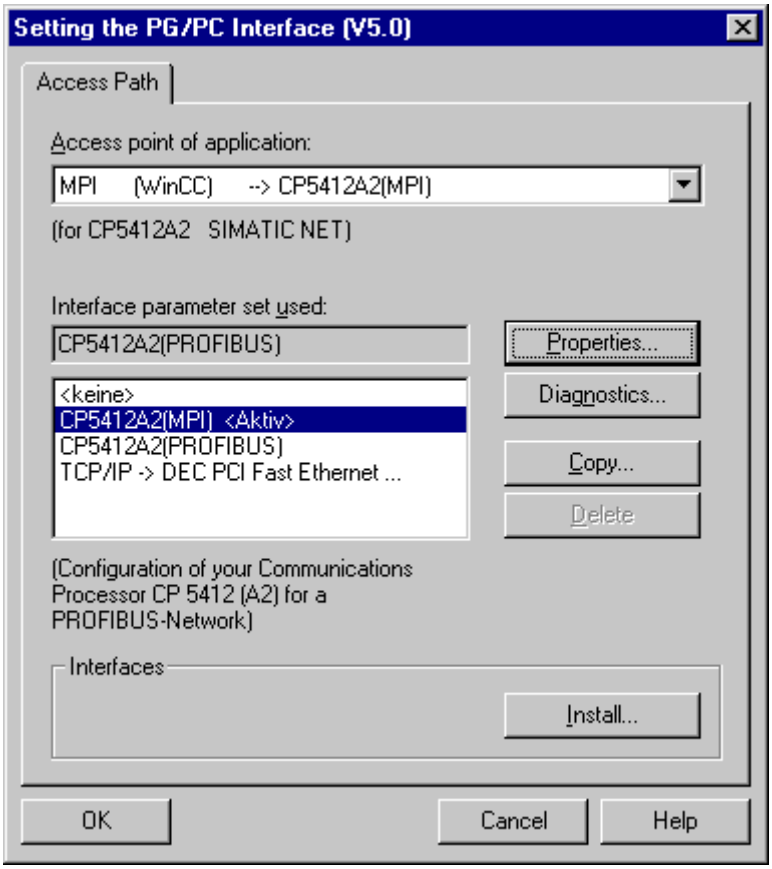
Installing the Communication Processor

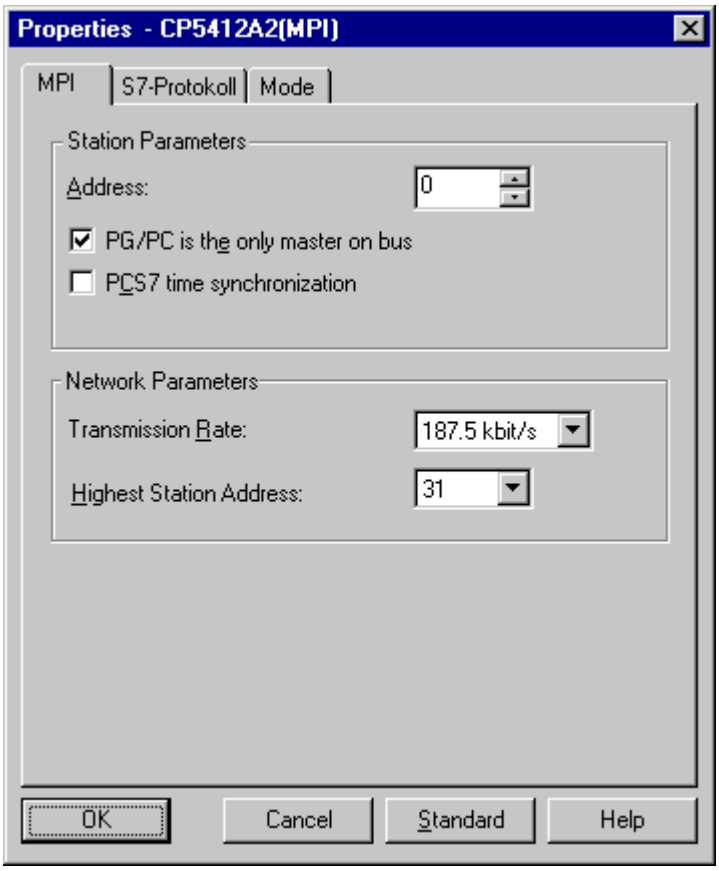
Step	Procedure: Installing the Communication Processor
1	Install the communication processor *CP 5412 A2* via the program *Setting the PG/PC Interface*. This program is accessed via *Start* → *Settings* → *Control Panel* → *Setting the PG/PC Interface*.  Setting the PG/PC Interface
2	The program *Setting the PG/PC Interface* will be displayed. The dialog for installing a new interface is opened via the *Install* button. 

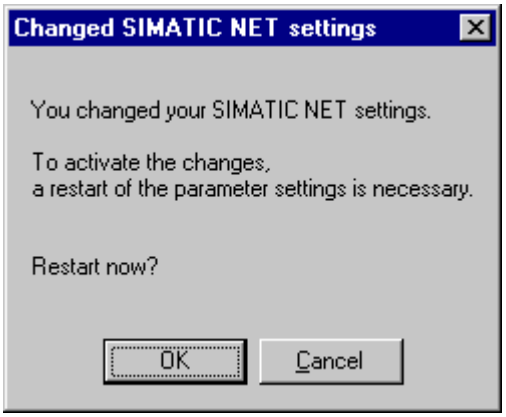
Step	Procedure: Installing the Communication Processor
3	The dialog *Install/Remove Modules* will be displayed. The *Selection* field lists all interfaces that can be installed. Among them will be the entry for the *CP 5412 A2*, if the communication driver has been installed previously as outlined in step B. From the *Selection* field, select the entry *CP 5412 A2*. The installation of the communication processor is started by clicking on the *Install* -> button. 
4	The dialog *Resources - CP 5412 A2* will be displayed. The settings for the *Memory Range*, *I/O Range* and *Interrupt* have to be specified. The *I/O Range* has already been determined via the *Jumper Settings* at the *CP 5412 A2*. Make sure that the assigned resources have not already been taken by other modules in the computer. Information about already taken system resources can be obtained from the *Resources* tab accessed via *Start* → *Programs* → *Administrative Tools (Common)* → *Windows NT Diagnostics.* Close the *Resources* tab by clicking on *OK*. 

Step	Procedure: Installing the Communication Processor
5	In the dialog *Install/Remove Modules*, the *Installed* field will now contain the entry for the *CP 5412 A2*. Exit the dialog *Install/Remove Modules* via the *Close* button. 

Assigning the Communication Processor

Step	Procedure: Assigning the Communication Processor
1	In the program *Setting the PG/PC Interface*, assign the access point *MPI (WinCC)* to the just installed interface. The access point *MPI (WinCC)* is the default access point used by WinCC for the communication via *MPI*. In the *Access Point of the Application* field, the *MPI (WinCC)* entry is set. In the field below, select the *CP 5412 A2 (MPI)* entry. This completes the assignment between the access point and the communication processor. 

Step	Procedure: Assigning the Communication Processor
2	Setting the properties of the communication processor *CP 5412 A2*. The dialog for setting the properties is opened via the *Properties* button of the *Setting the PG/PC Interface* program. The *Properties - CP 5412 (MPI)* dialog will be displayed. In the *MPI* tab, station and network related parameters are set. In this sample, the *Address* of the communication processor is set to *0*. For the *MPI Network*, this sample uses a *Transfer Rate* of *187.5 Kbit/s*. The *Highest Station Address* is set to the maximum value of *31*. Close the properties dialog of the *CP 5412 A2* by clicking on *OK*. 

Step	Procedure: Assigning the Communication Processor
3	Exit the program *Setting the PG/PC Interface* via the *OK* button. A dialog will be displayed requesting the restart of the *CP 5412 A2*. Acknowledge this dialog by clicking on *OK*, which will result in the restart of the communication processor *CP 5412 A2*. This completes the installation of the communication processor. 
4	If the restart of the communication processor is not successful, the error cause must be determined and corrected. Help regarding this topic can be found in the *Communication Manual*.

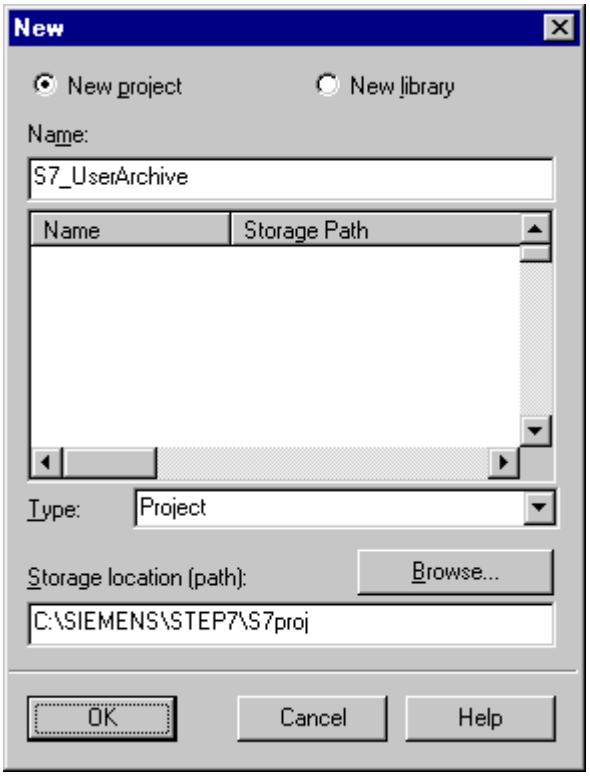
5.1.8.2 Startup of the PLC

The following description details the configuration steps necessary to create and start up the STEP7 project *S7_UserArchive*.

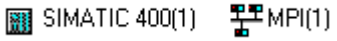
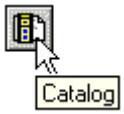
Installing the Hardware

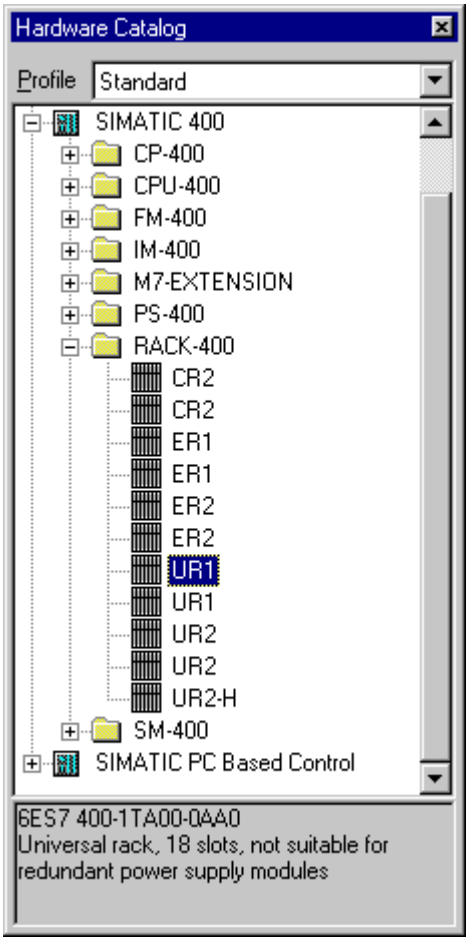
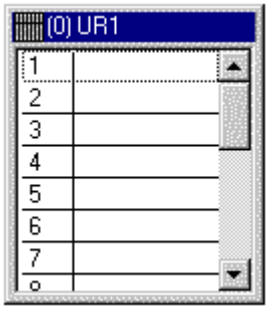
Step	Procedure: Installing the Hardware
1	Rack-mounting of the modules used. In this sample, the modules to be installed are the power supply *PS 407 10A* and the CPU module *CPU 416-1*. Establishing the connection from the programming device to the programming interface of the CPU module.

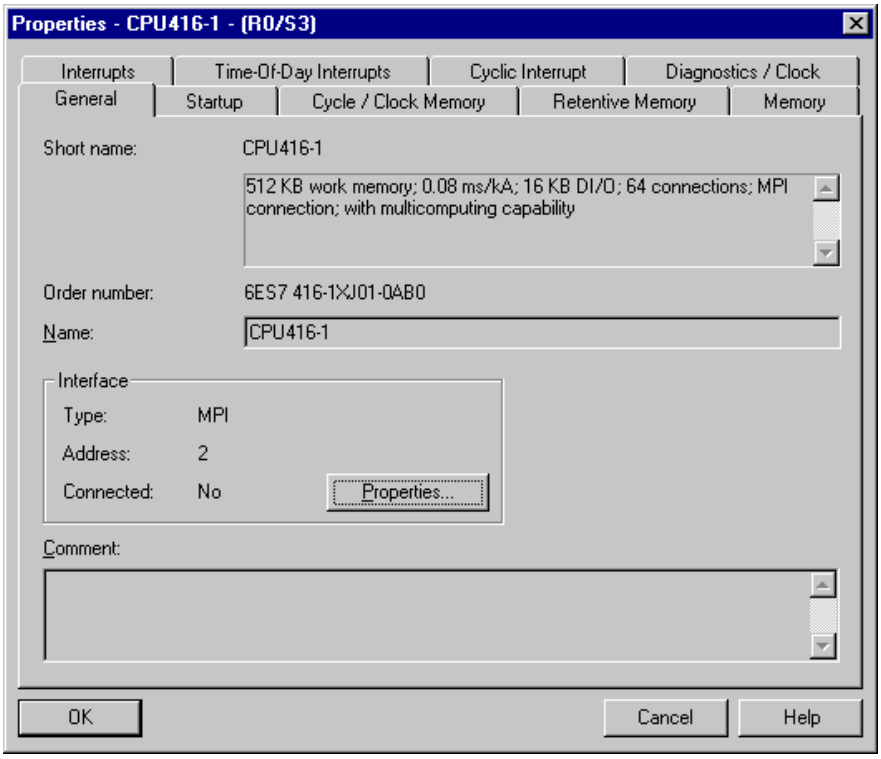
Creating the STEP7 Project

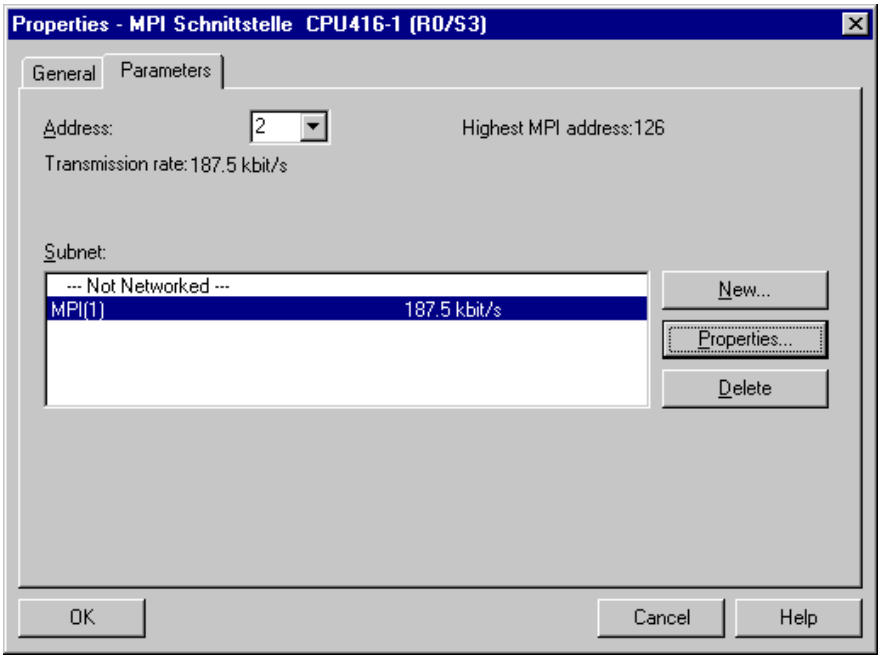
Step	Procedure: Creating the STEP7 Project
1	Create a new STEP7 project in the *SIMATIC Manager*. It is started via *Start* → *Simatic* → *SIMATIC Manager*.  SIMATIC Manager
2	This displays the *SIMATIC Manager*. Via the menus *File* → *New*, the dialog for specifying the parameters of a new STEP7 project will be opened. The *New* dialog will be displayed. The radio-button *New Project* must be selected. In the *Name* field, the name of the new project to be created is entered. The project of this sample has the name *S7_UserArchive*. By default, projects are stored in the *C:\SIEMENS\STEP7\S7proj* folder. This can be changed at any time via the *Browse* button. The *New* dialog is closed via the *OK* button. 

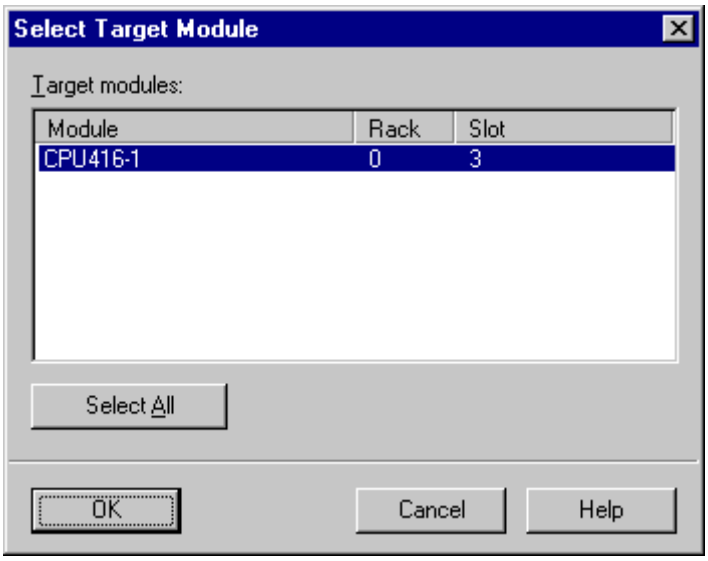
Configuring the Hardware

Step	Procedure: Configuring the Hardware
1	The new project will be displayed in the *SIMATIC Manager*. The hardware for this project must be configured. One *SIMATIC 400-Station* component is required. This component is added in the *SIMATIC Manager* via a R on the project name *S7_UserArchive* and then selecting *Insert New Object* → *SIMATIC 400-Station* from the pop-up menu. The component just added will be displayed in the right window of the *SIMATIC Manager*. 
2	By D on the component *SIMATIC 400(1)* in the right window, the point *Hardware* will be displayed. By D on the point *Hardware* or R on it and then selecting *Open Object* from the pop-up menu, the program *HW Config* will be started. This program is used to precisely define the hardware used in the PLC and to configure their properties. 
3	By clicking on the toolbar button of the program *HW Config* displayed below, the *Hardware Catalog* is opened. This catalog is used to select the required hardware components. 

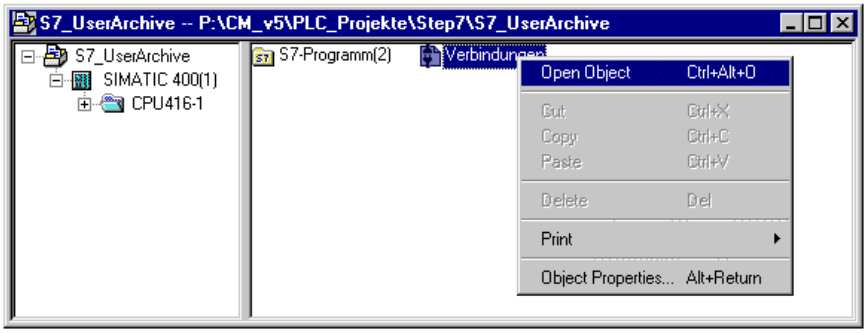
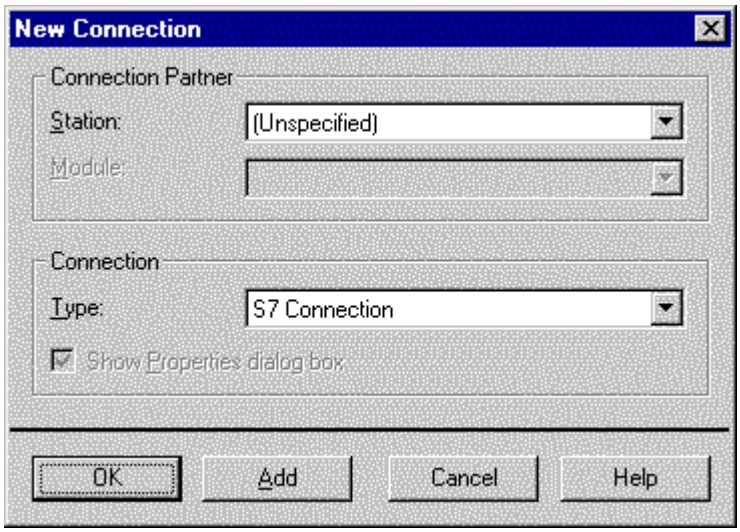
Step	Procedure: Configuring the Hardware
4	The *Hardware Catalog* will be displayed. The first component selected is the rack. On this rack, all other components will be installed. The rack is inserted into the project via a D or by Dragging Dropping. In this sample, the rack type *UR1* is used. 
5	The program *HW Config* will display the currently still empty rack. It received the Rack Number *0*. During the configuration of the connection in the WinCC project, the rack number is one of the parameters that must be set. 

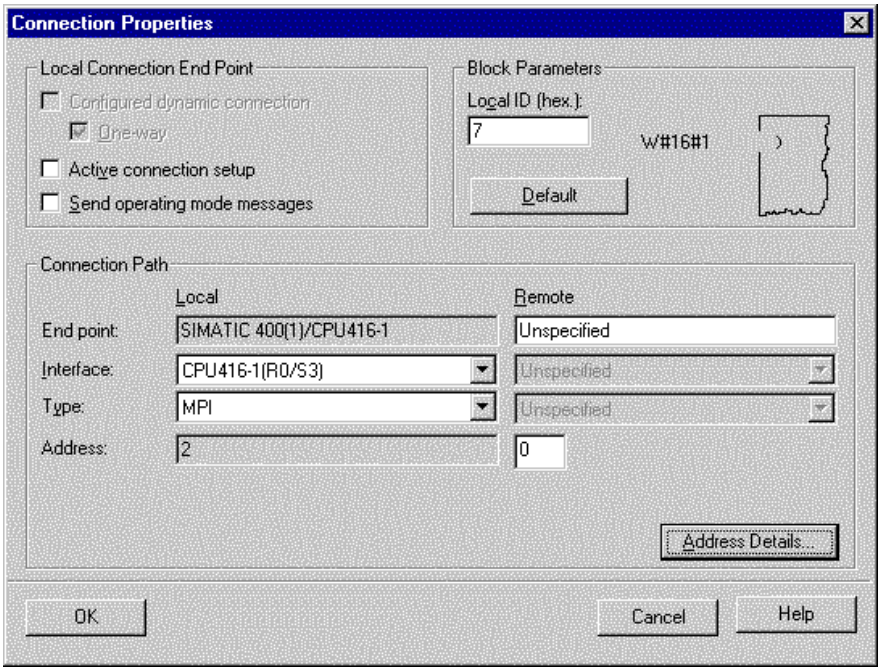
Step	Procedure: Configuring the Hardware
6	Arrange the other hardware components in the rack. This is done by Dragging Dropping the desired components from the *Hardware Catalog* to the corresponding slot in the rack. This sample uses the power supply *PS 407 10A*. It is inserted into slot *1*. A power supply of this type occupies two slots. As the CPU module, this sample uses a *CPU 416-1*. This module is inserted into slot *3*. Another parameter to be set during the configuration of the connection in the WinCC project is the slot number of the CPU module.
7	Setting the properties of the CPU module. For this purpose, the properties dialog of the of the CPU module is opened via a D on the rack icon. In the *General Information* tab, the preliminary communication settings are displayed in the *Interface* field. The interface of the CPU module has been defined as not networked. This is changed via the *Properties* button. 

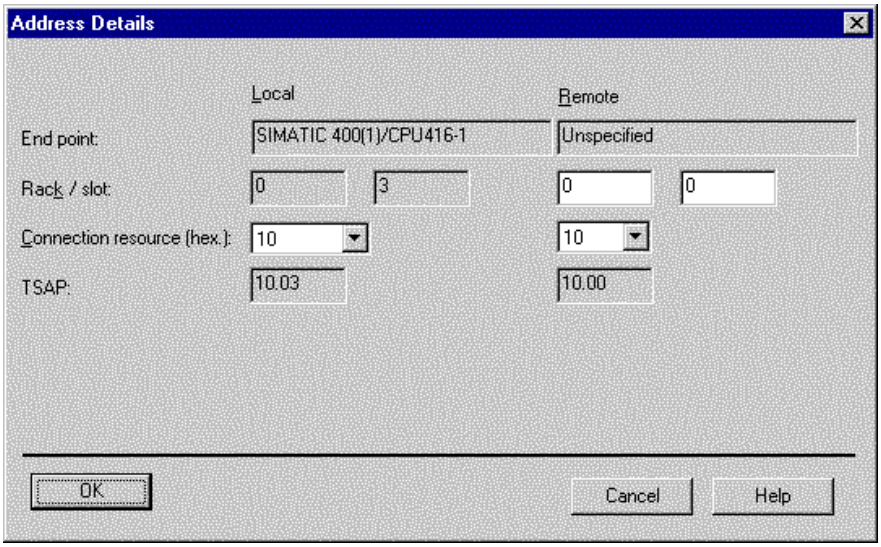
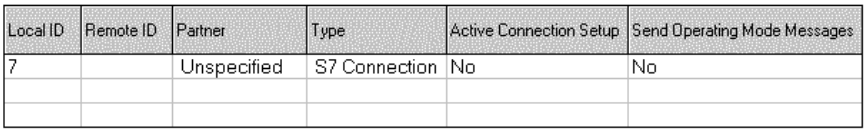
Step	Procedure: Configuring the Hardware
8	The properties dialog of the MPI interface will be displayed. In this sample, the *Address* of *2* is kept. In the *Subnet* field, the *MPI(1)* entry is selected. This assigns the CPU module to the *MPI Network*. Via the *Properties* button, the properties of the *MPI Network* can be changed if necessary. In this case, the settings made while installing the *CP 5412 A2* communication processor must be adapted as well. In this sample however, the *Transfer Rate* of *187.5 Kbit/s* and the *Highest Station Address* of *31* are kept. For the latest editions of the CPU module, a transfer rate of greater than 187.5 Kbit/s is possible. The properties dialog of the MPI interface is closed with *OK*. The properties dialog of the CPU module is closed with *OK* as well. 
9	The settings made are saved. Furthermore, the hardware configuration created is transferred to the PLC. This is done via the toolbar button displayed below. 

Step	Procedure: Configuring the Hardware
10	A dialog will be displayed from which the components to be loaded can be selected. In this sample, only the entry of the CPU module is available. Note that loading to the CPU module is only possible while the operating mode switch is set to *STOP* or *RUN-P*. Close the dialog by clicking on *OK*. Now the dialog *Select Station Address* will be displayed. In this dialog, specify which station address is used by the STEP7 software to communicate with the CPU module. In this sample, the communication is carried out via the MPI interface. The *Address* of the CPU module is *2*. Close the dialog by clicking on *OK*. The configuration data will now be transferred to the PLC. For this purpose, the CPU module is set to the *STOP* status. The program *HW Config* can be exited. 

Creating the S7 Connection

Step	Procedure: Creating the S7 Connection
1	Via a on the entry of the *CPU 416-1* CPU module in the right window, the *Connections* entry will be displayed. Via a D on the *Connections* entry or a R and then selecting *Open Object* from the pop-up menu, the *NetPro* program is started. 
2	The current network configuration is displayed by the *NetPro* program. The *SIMATIC S7-400 Station* is displayed. Its CPU module has already been connected to the *MPI Network*. The CPU module is selected. This enables the *Insert* → *Connection* menu. This menu is used to create a new connection. In the *Connection Partner* field, the entry *(unspecified)* is set as the *Station*. In the *Connection* field, the entry *S7 Connection* is set as the *Type*. Via the *OK* button, a dialog for defining the connection in greater detail is opened. 

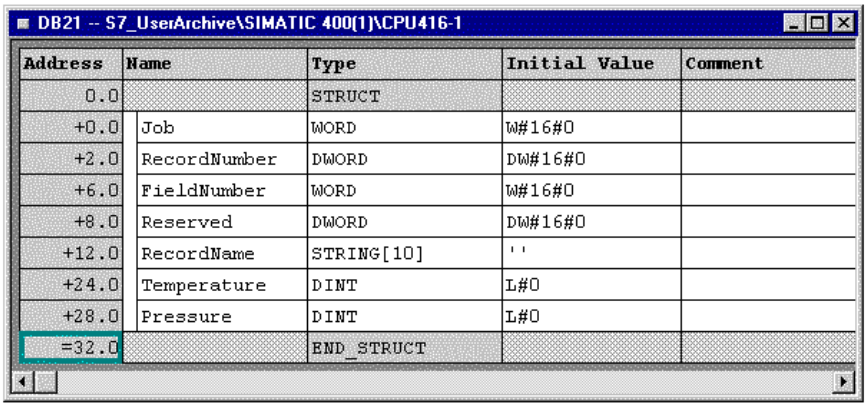
Step	Procedure: Creating the S7 Connection
4	The properties dialog of the connection will be displayed. The check-box *Active Connection Setup* is deselected. A local ID is assigned to the connection. In this sample, the value *7* is entered in the *Local ID (Hex)* field. As the *Address* of the *Partner*, the value *0* of the communication processor *CP 5412 A2* is entered. Via the *Address Details* button, additional parameters of the connection are set. 

Step	Procedure: Creating the S7 Connection
5	The *Address Details* dialog will be displayed. For both the rack and slot number of the *Partner*, *0* is used. As the *Connection Resource*, the same value must be used for the local station and the *Partner*. In this sample, the value *10* is used. The *Address Details* dialog can be closed via the *OK* button. The properties dialog of the connection can be closed as well via the *OK* button. 
6	In the table below, the newly created connection will be displayed. Select this connection. Via the *Target System* → *Load* → *Selected Connections* menus, the connection is loaded into the PLC. The configurations made in the NetPro program are saved. The program can be exited. 

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

Creating the STEP7 Program

Step	Procedure: Creating the STEP7 Program
1	Creation of data blocks for the telegram data. This is done in the *SIMATIC Manager* via a R on the sub-entry *Modules* of the entry *S7-Program(1)* of the CPU module and then selecting *Insert New Object* → *Data Block* from the pop-up menu. As the names of the data blocks, this sample uses *DB25* to *DB30*. Via a D on the data block or a R and then selecting *Open Object* from the pop-up menu, the content of the block can be programmed. This starts the program *LAD/STL/SFC*. These data blocks contain the telegram data of the various jobs. For each job implemented, this sample uses a separate data block. The data ranges for the record and field numbers initially remain empty. They are only filled with current values just before the telegram is sent. Likewise, possibly required job data is only entered before the telegram is sent. The following depicts the completed *DB26* data block. This block contains the telegram data for the job of reading a data record from the *User Archives*. DB26 -- S7_UserArchive\SIMATIC 400(1)\CPU416-1 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 *1.0 / / 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	Creation of the data block for the communication with the *User Archives*. As the name of the data block, this sample uses *DB23*. The acknowledgment telegrams sent by WinCC are stored in this data block. The size of the data block depends on the maximum telegram length. In this sample, this is a length of 42 Bytes, which occurs during a read job of a data record from the archive. This length consists of a 24 Byte acknowledgment and 18 Bytes for the job data.

Step	Procedure: Creating the STEP7 Program
3	Creation of an additional data block. As the name of this data block, this sample uses *DB21*. Among other things, the *DB21* contains data ranges for a control tag, the field number as well as the record number. All three values are visualized using WinCC tags and be changed in runtime for simulation purposes. In addition, the data block contains the data of a data record. This data corresponds to the data record currently requested by the *User Archives*. The data is visualized using WinCC tags and can also be changed in runtime for simulation purposes. The following depicts the completed *DB21* data block. 

DB21 -- S7_UserArchive\SIMATIC 400(1)\CPU416-1

Address	Name	Type	Initial Value	Comment
0.0		STRUCT		
+0.0	Job	WORD	W#16#0	
+2.0	RecordNumber	DWORD	DW#16#0	
+6.0	FieldNumber	WORD	W#16#0	
+8.0	Reserved	DWORD	DW#16#0	
+12.0	RecordName	STRING[10]	''	
+24.0	Temperature	DINT	L#0	
+28.0	Pressure	DINT	L#0	
=32.0		END_STRUCT		

Step	Procedure: Creating the STEP7 Program
4	Creation of a function block, which carries out the communication with the *User Archives*. In the sample, this is the *FB100*. This function block is called cyclically via the *OB1*. Creation of function blocks that are intended to process the jobs of a certain job type. These function blocks are called sequentially in the *FB100*. In this sample, the function blocks *FB106* to *FB110* are used. Depending on the status of the flag assigned to each job type, job telegrams are sent to the *User Archives* in these function blocks via the system function block *SFB12 BSEND*. For the *SFB12*, an instance block must be provided. In the sample, this is the *DB12*. Creation of a function block, which receives acknowledgment telegrams sent by the *User Archives*. This function block is also called by the *FB100*. In the sample, this is the *FB120*. Depending on whether a job telegram has been sent to the User Archives or not, the *FB120* accepts the acknowledgment telegrams of the *User Archives* via the system function block *SFB13 BRCV*. For the *SFB13*, an instance block must be provided. In the sample, this is the *DB13*. Following this table, the operation of the STEP7 program is described in greater detail by means of the function blocks *FB100*, *FB106* and *FB120*.

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	The blocks created are saved and loaded into the PLC. This is done via the toolbar button displayed below. Note that loading to the CPU module is only possible while the operating mode switch is set to *STOP* or *RUN-P*. 

FB100 (ARCHIVE_COMMUNICATION)

```
//Dont load a new Job if the previouse 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
```

- The *M32.0* flag identifies a job currently running. If the *M32.0* flag is set, no new job is accepted and a jump to the *BUSY* position is made.
- The individual Bits of the *DB21.DBW0* data word request jobs of a certain job type from the *User Archives*. In this sample, WinCC writes to the data word. The implementation in WinCC achieves that only one Bit of the data word can be set at a time.
- The *MW30* flag word stores the content of the *DB21.DBW0* data word. The individual flags of this flag word trigger jobs of a certain job type from the *User Archives*.
- If the *M32.0* flag is not set, the *DB21.DBW0* data word is written to the *MW30* flag word.
- If no job needs to be executed, the block is terminated.
- If a job needs to be executed, the content of the *DB21.DBW0* data word is deleted. The *M32.0* flag will be set.
- The function blocks processing the various job types are called sequentially.
- The function block responsible for accepting the acknowledgment telegrams is called.

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  :=P#DB26.DBX 0.0 BYTE 46
       LEN   :=MW103
//Stop Sending of Data when ready
      A     M    104.1
      R     M     31.1
      BE
```

- The *MW103* flag word contains the length of the data to be sent. The telegram length of the job to read from the *User Archives* is 28 Bytes.
- The *M31.1* flag triggers the job for reading a from the *User Archive*. If it is not set, a jump to the *SEND* position is made.
- If the *M31.1* flag is set, the record number to be processed from the *DB21* is written to the *DB26*, which contains the telegram data for the read job. In this case, the order of the individual Bytes must be reversed.
- For the *FB120* block receiving the acknowledgment telegram, the *M31.1* flag identifies whether a complete data record has been received. It is set.
- Sending of the job telegram via the *SFB12 BSEND* system function block. Its call parameters are described below.
  - **REQ:** Activates the data exchange. In this sample, a telegram is sent, if the *M31.1* flag is set.
  - **R:** Cancels a running data exchange. Is not used in this sample.
  - **ID:** The reference to the local connection description. In this sample, it has been defined while creating the S7 connection with the hexadecimal value 7.

- **R_ID:** The reference to the communication partner. The value set here must be specified during the creation of the raw data tag, which is used to process the communication.
- **DONE:** Identifies the error-free processing of a job. In this sample, the value is stored in the *M104.1* flag. If it is set, the *M31.1* flag triggering the job will be reset.
- **ERROR:** Identifies the non-error-free processing of a job. Is not used in this sample.
- **STATUS:** Supplies detailed information regarding the type of error occurred. Is not used in this sample.
- **SD_1:** The pointer pointing to the data to be sent. In this sample, the data is to be sent from the *DB26*.
- **LEN:** The length of the data to be sent. In this sample, it is stored in the *MW103*.
- Resetting of the *M31.1* flag if the job has been executed error-free (the *M104.1* flag is set).

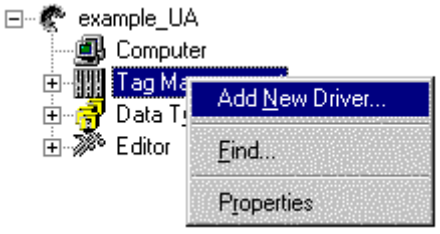
FB120 (RECEIVE_RESPONSE)

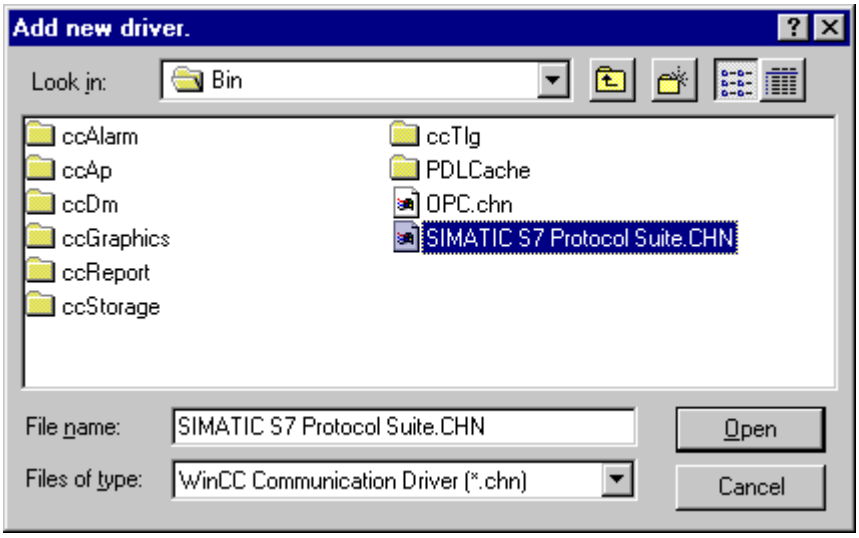
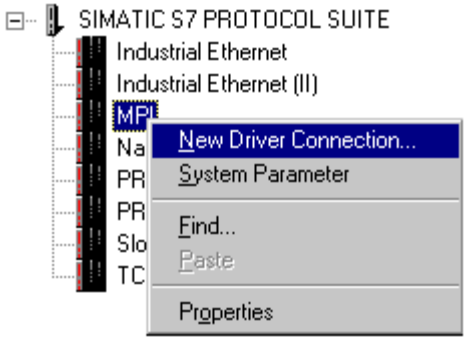
```
//Set Telegram Length
      L     42
      T     MW   203
//Receive Response Telegram
      CALL  "BRCV" , DB13
       EN_R  :=M32.0
       ID    :=W#16#7
       R_ID  :=DW#16#5
       NDR   :=M200.0
       ERROR :=M200.1
       STATUS:=MW201
       RD_1  :=P#DB23.DBX 0.0 BYTE 42
       LEN   :=MW203
//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
```

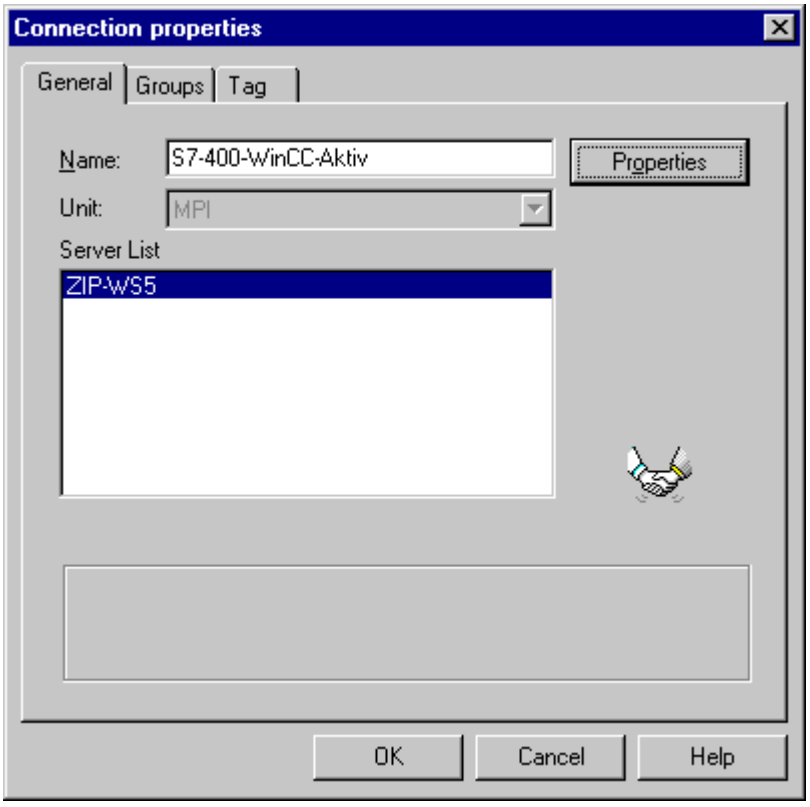
- The *MW103* flag word contains the length of the data to be received. The maximum length of an acknowledgment telegram is 42 Bytes. This occurs for the job of reading a data record from the *User Archives*.
- Receiving of the acknowledgment telegram via the *SFB13 BRCV* system function block. Its call parameters are described below.
  - **EN_R:** Enables the receiving of data. In this sample, the *M32.0* flag controls the ready-to-receive status of the block. Consequently, it is ready to receive while still processing a job.
  - **ID:** The reference to the local connection description. In this sample, it has been defined while creating the S7 connection with the hexadecimal value 7.
  - **R_ID:** The reference to the communication partner. The value set here must be specified during the creation of the raw data tag, which is used to process the communication.
  - **NDR:** Identifies the error-free transfer of data. In this sample, the value is stored in the *M200.0* flag. If it is set, the *M32.0* flag identifying a running job will be reset.
  - **ERROR:** Identifies the non-error-free transfer of data. Is not used in this sample.
  - **STATUS:** Supplies detailed information regarding the type of error occurred. Is not used in this sample.
  - **RD_1:** The pointer pointing to the data range, where the data received is to be stored. In the sample, this is the *DB23*.
  - **LEN:** The length of the data to be received. In this sample, it is stored in the *MW103*.
- If the data transfer is not complete yet, the block is terminated.
- If the data transfer is complete, the job is marked as finished by resetting the *M32.0* flag.
- Loading of the error code from the acknowledgment telegram. It is stored in the *MW35* flag word. The *MW35* flag word is evaluated in WinCC. If the error code has a value unequal to *0*, an error occurred. In this case, the block is terminated.
- If no error occurred, it must be checked whether archive data came in and if yes, of what type it is. The *M33.0* flag identifies a data record that came in, the *M33.1* flag a data field that came in. In the *FB120* function block depicted above, the sections for the data transfer are not shown due to their length. They have been replaced by comments.

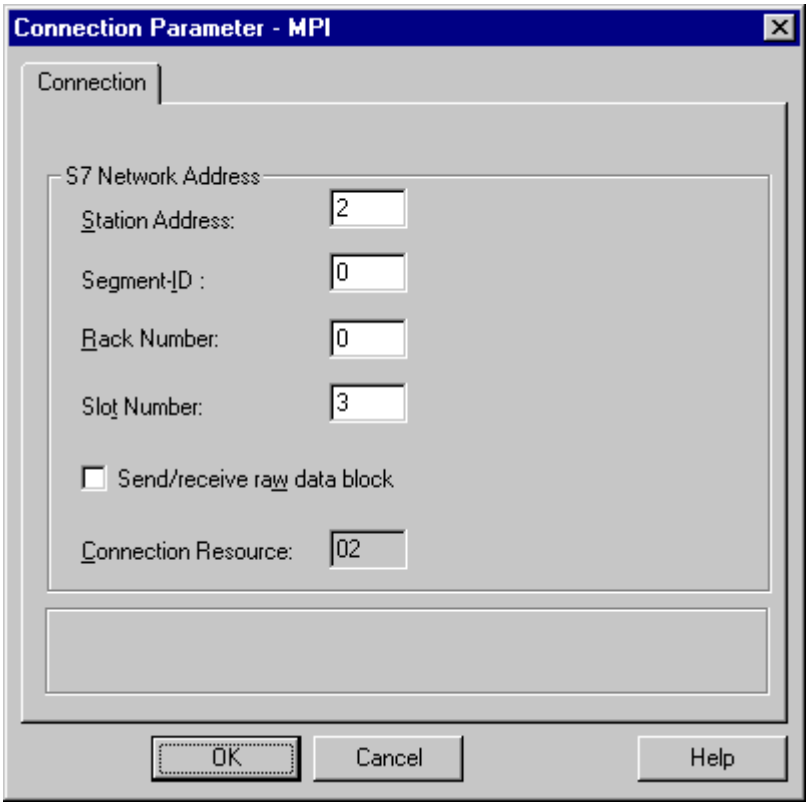
5.1.8.3 Configuration in WinCC

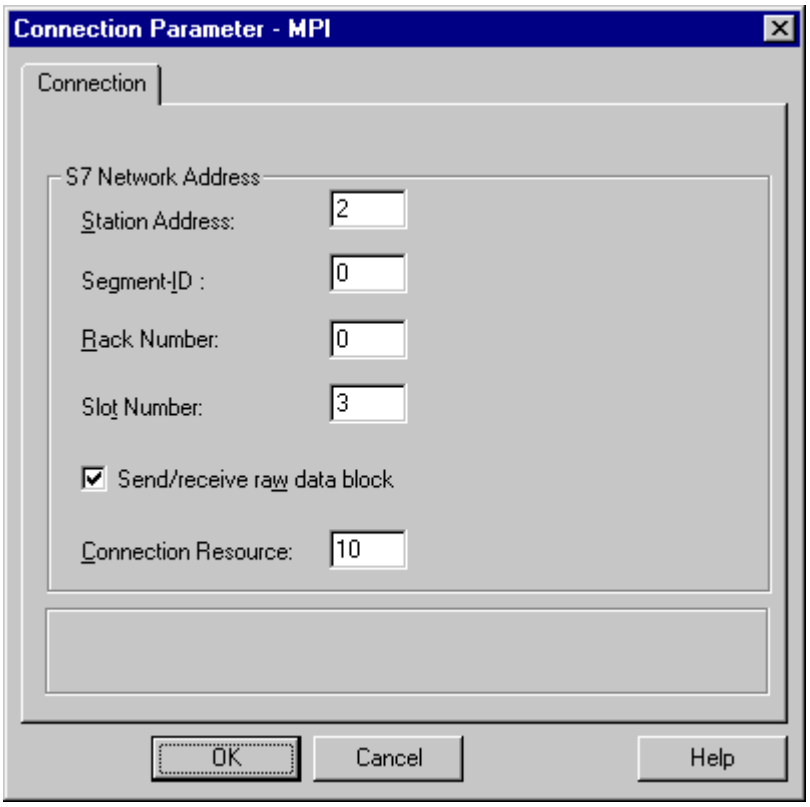
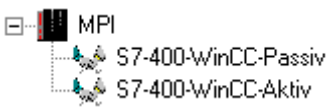
Creating the Connections

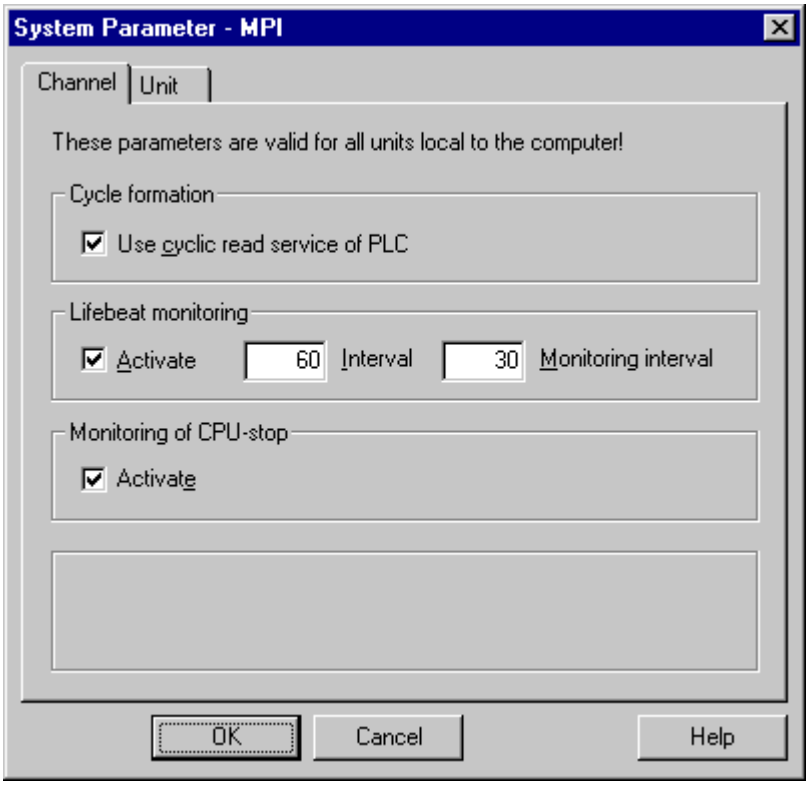
Step	Procedure: Creating the Connections
1	Installation of the required communication driver. This is performed via a R on *Tag Management* and selecting *Add New Driver* from the pop-up menu. 

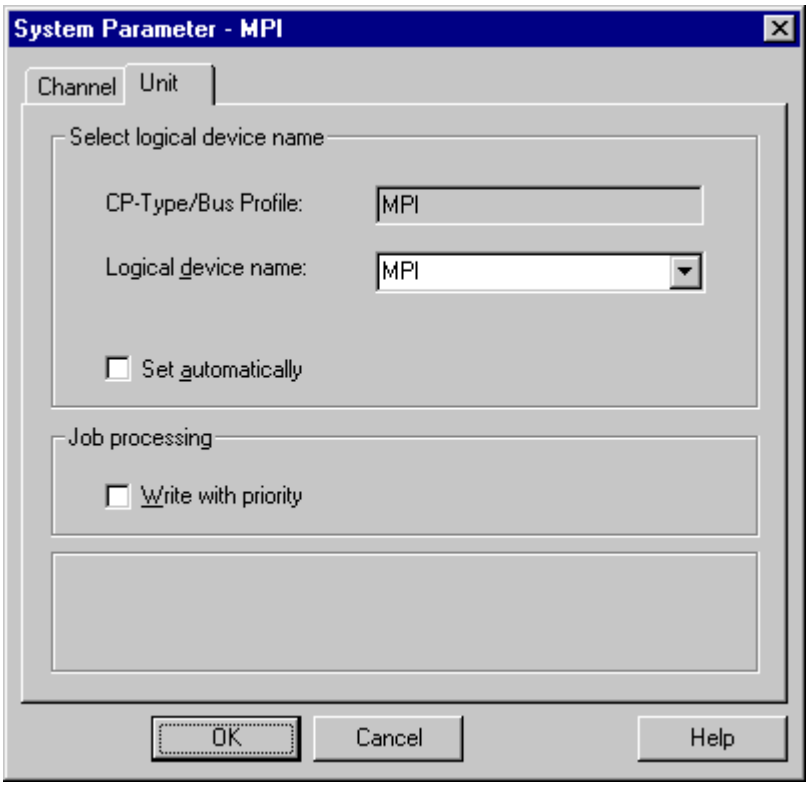
Step	Procedure: Creating the Connections
2	The dialog *Add New Driver* will be displayed. This dialog lists all communication drivers that can be installed. For the communication with the *SIMATIC S7*, the driver *SIMATIC S7 Protocol Suite* must be installed. Select this driver from the dialog. Exit the dialog by clicking on *Open*. 
3	The newly added driver *SIMATIC S7 Protocol Suite* will be displayed as a sub-entry to *Tag Management*. The driver contains eight different channel units. For the communication via the *MPI Network*, the *MPI* channel unit is available. For the *MPI* channel unit, a new connection is created by RR on the *MPI* entry and then selecting *New Driver Connection* from the pop-up menu. This connection is used for the active sending of data from the PLC. In this case, WinCC is the active partner. 

Step	Procedure: Creating the Connections
4	The properties dialog of the connection will be displayed. In the *General* tab, the *Name* of the new connection is entered. In the sample, this is *S7-400-WinCC-Active*. Click on the *Properties* button to define the connection properties. 

Step	Procedure: Creating the Connections
5	The dialog *Connection Properties* will be displayed. In the *Station Address* field, the address set for the MPI interface of the CPU module is entered. In the sample, this is the address*2*. Additionally, the *Rack Number* and *Slot Number* of the CPU module to be accessed must be entered. Close the dialog by clicking on *OK*. Also close the *Connection Properties* dialog by clicking on *OK*. 

Step	Procedure: Creating the Connections
6	Creation of a second connection for the *MPI* channel unit by R on the *MPI* entry and then selecting *New Driver Connection* from the pop-up menu. This connection is used by the PLC for actively writing data. In this case, WinCC is the passive partner. The properties dialog of the connection will be displayed. In the *General* tab, the *Name* of the new connection is entered. In the sample, this is *S7-400-WinCC-Passive*. Via the *Properties* button, the connection properties are defined. The dialog *Connection Properties* will be displayed. For the *Station Address*, *Rack Number* and *Slot Number*, the same values as specified in the previously created *S7-400-WinCC-Active* connection must used. The *Send/Receive Raw Data Block* check-box is selected. In the *Connection Resource* field, the connection resource specified in the PLC during the creation of the S7 connection must be entered. In this sample, the value *10* is used. Close the dialog by clicking on *OK*. Also close the *Connection Properties* dialog by clicking on *OK*. 
7	The just created connections will now be displayed in the *WinCC Explorer*. 

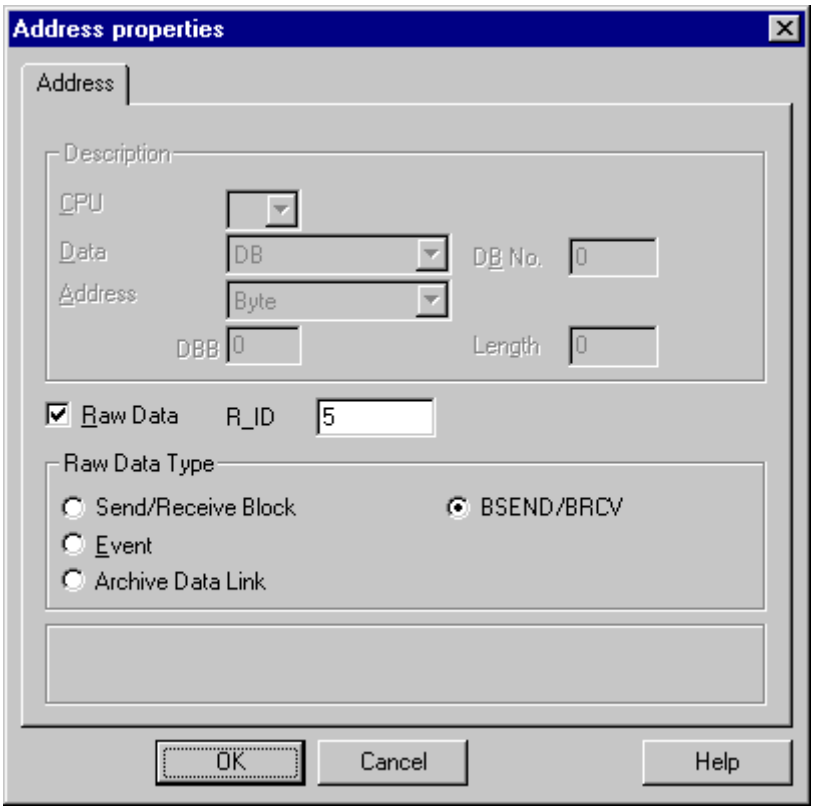
Step	Procedure: Creating the Connections
8	Setting the system parameters of the *MPI* channel unit. These settings are made in the *System Parameters* dialog, which is accessed via a R on the *MPI* entry and then selecting *System Parameters* from the pop-up menu. In the *Channel* tab, various settings pertaining to the communication and monitoring a communication can be made. These settings will apply to all channel units of the communication driver. 

Step	Procedure: Creating the Connections
9	In the *Device* tab, the access point used by the connection to access the PLC is specified. The *MPI* access point is set. Previously, the communication processor *CP 5412 A2* has been assigned to this access point in the program *Setting the PG/PC Interface*. If you want the access point to be set automatically, make sure that the correct one is being used, especially if multiple communication processors are used. Close the dialog by clicking on *OK*. 

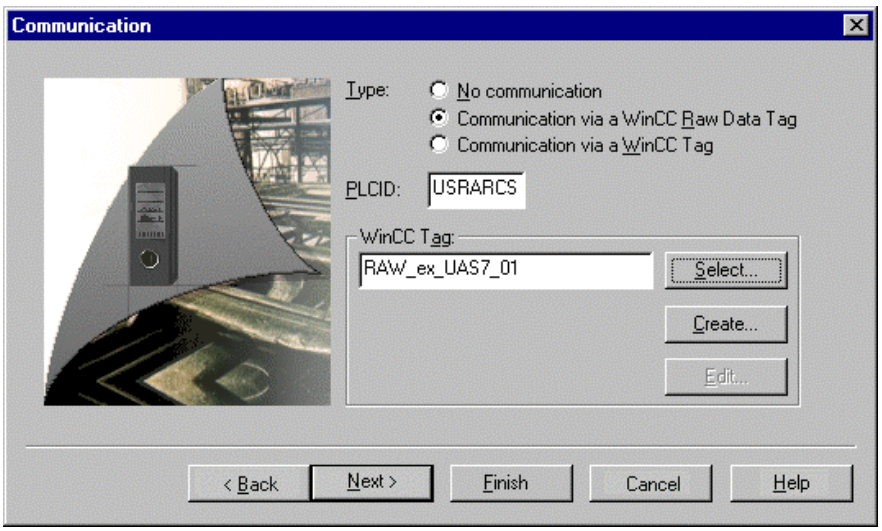
Creating the Tags

Step	Procedure: Creating the Tags
1	Creation of the tags for the *S7-400-WinCC-Active* connection. This is done in the *WinCC Explorer* via a R on the corresponding connection entry and then selecting *New Tag* from the pop-up menu. The names, data types and addresses of the individual tags are listed in the following illustration. The *T08w_ex_UAS7_RecordName*, *S32w_ex_UAS7_Pressure* and *S32w_ex_UAS7_Temperature* tags represent the data record currently in the PLC, which is stored in *DB21*. The *U32w_ex_UAS7_Record*, *U16w_ex_UAS7_Field* and *U16w_ex_UAS7_Job* tags represent the record number and field number to be processed as well as the control tag, which are also stored in *DB21*. The *U08w_ex_UAS7_Busy* and *U08w_ex_UAS7_Error* tags are used to display the current job status.

Name	Type	Parameters
U08w_ex_UAS7_Busy	Unsigned 8-bit value	MB32
T08w_ex_UAS7_RecordName	Text tag 8-bit character set	DB21,DBB12
U16w_ex_UAS7_Job	Unsigned 16-bit value	DB21,DW0
S32w_ex_UAS7_Pressure	Signed 32-bit value	DB21,DD28
S32w_ex_UAS7_Temperature	Signed 32-bit value	DB21,DD24
U32w_ex_UAS7_Record	Unsigned 32-bit value	DB21,DD2
U16w_ex_UAS7_Field	Unsigned 16-bit value	DB21,DW6
U08w_ex_UAS7_Error	Unsigned 8-bit value	MB35

Step	Procedure: Creating the Tags
2	Creation of a tag for the *S7-400-WinCC-Passive* connection. This is done in the *WinCC Explorer* via a R on the corresponding connection entry and then selecting *New Tag* from the pop-up menu. The properties dialog of the tag will be displayed. In this sample, the *Name* of this tag is *RAW_ex_UAS7_01*. As the *Data Type* of this tag, *Raw Data Type* is set. Via the *Select* button, the dialog for addressing the tag is opened. The *Raw Data* check-box is selected. In the *Raw Data Type* field, the *BSEND/BRCV* radio-button is selected. In the *R_ID* field, the value of the *R_ID* parameter is entered that has been specified in the STEP7 program for the call of the *BSEND* and *BRCV* system function blocks. In the sample, this was the value *5*. The *Address Properties* dialog is closed with *OK*. The *Tag Properties* dialog is closed with *OK* as well. 

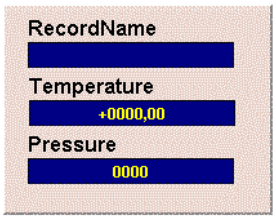
Creating the User Archives

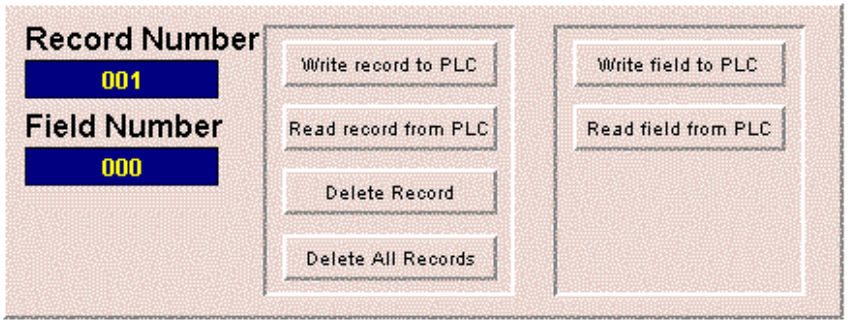
Step	Procedure: Creating the User Archives
1	Open the *User Archives Editor*. In this editor, a new archive is created via a RR on the *Archives* entry and then selecting *New Archive* from the pop-up menu. This starts a Wizard for the archive creation. 
2	On the first page of this Wizard, the *Archive Name* is entered. In the sample, the name *UserArchive_S7* is entered in the *Archive Name* field. The *Alias* field is left blank. As the *Archive Type*, *Unlimited* is selected. Continue to the next page by clicking on *Next*.
3	On the second Wizard page, the radio-button *Communication via WinCC Raw Data Tag* is selected. As the *PLCID*, this sample uses the name *USRARCS7*. This ID must not exceed 8 characters. Via this ID, a job telegram sent by the PLC can be assigned to a certain archive. Via the *Select* button, the *WinCC Raw Data Tag* configured for the archive communication is set. In the sample, this is the *RAW_ex_UAS7_01* tag. Since no settings are made for this sample on the next page, the Wizard can be exited by clicking on *Finish*. 
4	For the just created archive, the fields listed in the following table are created. Save the settings made. Name / Alias / Type / Length Pressure — Number (integer) RecordName — String — 10 Temperature — Number (double)

Step	Procedure: Creating the User Archives
5	In the lower table window of the *User Archives Editor*, multiple data records can be now be created for the archive.

	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
...				

Implementation in the Graphics Designer

Step	Procedure: Implementation in the Graphics Designer
1	Open the *Graphics Designer* and create a new picture. In this sample, this is the *ex_3_chapter_01b.pdl* picture. Configuration of the Control used to display the data. This is the *WinCC User Archives - Table Element*. It is selected from the *Object Palette's Control* selection menu and then placed in the picture.
2	Via a D on the just created *Control1* object, its properties dialog is opened. In the *General* tab, the previously configured archive *UserArchive_S7* is set in the *Source* field. In the *Edit* field, the *Insert*, *Change* and *Delete* check-boxes are selected to make all editing options available to the user. In addition, the *Border* check-box is selected. In the *Toolbar* tab, the two check-boxes for the buttons *Write Tags* and *Read Tags* are deselected. In this sample, the control is carried out by the PLC. All other buttons of the toolbar are used. In the *Fonts* tab, the *Size* of the font is reduced to *10* in order to display all columns simultaneously in runtime. The settings made in the properties dialog of the *WinCC User Archives - Table Element* are concluded via the *OK* button. In this sample, the color scheme of the table was matched to the project's color scheme via the properties dialog of the *Control1* object.
3	To display the data record currently loaded in the PLC, three *Smart Objects* → *I/O Fields* are configured. In this sample, these are the *I/OField1*, *I/OField2* and *I/OField3* objects. For the *I/O Fields*, a *Tag Connection* each at *Properties* → *Output/Input* → *Output Value* to one of the three tags *T08w_ex_UAS7_RecordName*, *S32w_ex_UAS7_Temperature* and *S32w_ex_UAS7_Pressure* is created. 

Step	Procedure: Implementation in the Graphics Designer
4	To specify the record number and field number to be processed, two additional *Smart Objects* → *I/O Fields* are configured. In this sample, these are the *I/OField4* and *I/OField5* objects. For the *I/OField4* object, a *Tag Connection* at *Properties* → *Output/Input* → *Output Value* to the *U32w_ex_UAS7_Record* tag is created. This I/O field displays the record number. For the *I/OField5* object, a *Tag Connection* at *Properties* → *Output/Input* → *Output Value* to the *U16w_ex_UAS7_Field* tag is created. This I/O field displays the field number. To trigger actions in the PLC, five *Windows Objects* → *Buttons* are configured. In this sample, these are the *Button9* to *Button13* objects. If one button is pressed, a different value is written to the *U16w_ex_UAS7_Job* tag. In the STEP7 program, this tag is queried cyclically. Depending on the content of this tag, a corresponding job is triggered. Each value corresponds to a different job type for the PLC. The assignment of the individual values to the corresponding jobs is listed below. For the *Delete All Records* job implemented in the STEP7 program, no button has been configured. • 1 = Delete All Records • 2 = Read Record from the Archive • 4 = Write Record to the Archive • 8 = Delete Record from the Archive • 16 = Read Field from the Archive • 32 = Write Field to the Archive Via the *Reset* button, the *U08w_ex_UAS7_Busy* tag can be reset. This permits a lock preventing the sending of a new job telegram to be removed if it has not been lifted automatically (only relevant in the case of errors). In the *Status* field, the current status of the communication connection as well as the archive communication is displayed using a *C-Action*. This *C-Action* evaluates the *U08w_ex_UAS7_Error* tag. The *C-Action* itself is described following this step. 

C-Action for Displaying the Status

```
#include "apdefap.h"
 char* _main(char* lpszPictureName, char* lpszObjectName, char* lpszProper
{

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:
      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";
}

}
```

- The *C-Action* has been created at *Properties* → *Font* → *Text* of the *StaticText3* object. The *Trigger* of the *C-Action* is set to *2 s*. This results in a status check of the connection and the archive communication every 2 seconds.
- The status check is performed by the *GetTagByteStateWait* function. The current connection status is written to the *dwState* tag, the archive communication status to the *byError* tag (error code of the acknowledgment telegram from WinCC).
- Before this action is performed, the background color of the *LED* object is set to red. Once the function has concluded, the background color of the *LED* object is reset to gray.
- Evaluation of the *dwState* and *byError* tags and return of a corresponding status text.

Note for the General Application

The following adaptations must be made before the general application:

- The communication configurations made can be applied directly to your own applications, provided that the same hardware is used. Otherwise, adaptations must be made.

The structure of the archive must be adapted to meet your own requirements.

Index