Engineer^{IT}
Control Builder F

Engineering Manual
Process Station - Rack System

ABB

Notice

Information provided in this manual is subject to change without prior notice and represents no obligation on the part of ABB Automation Products.

The industrial standards and regulations (e.g. DIN, VDE, VDI, etc.) applicable in the Federal Republic of Germany are used. Outside the Federal Republic of Germany, the relevant national specifications, standards and regulations must be observed.

ABB Automation Products reserves all rights, especially those arising out of BGB, UWG, UrhG as well as out of industrial property rights (patents, utility models, trademarks, service trademarks and flavor samples).

The designations used and the products shown/mentioned in this manual have not been specifically marked regarding existing industrial property rights.

No part of this manual may be reproduced without prior written permission from ABB Automation Products.

Should you find any mistakes in this manual, please make a copy of the appropriate page(s) and send it/them to us with your comments. Any suggestions which may help to improve comprehension or clarity will also be gratefully accepted.

Please send your suggestions to:

Product Management Dept., DEAPR/LMS-Hannover, Fax: +49 (0)511 6782 701

Engineering Manual
Process Station – Rack System

Contents

Notes on this manual

To grant direct access to information, we have used different types of scripts and symbols.

Script	**Meaning**
Italics	Representation for (selectable) menu items or parameters.
SMALL CAPITALS	Inputs to be made via the keyboard, also via virtual keys.
boldface	Highlights important information, also as an orientation hint.

Symbols	**Meaning**
📖	Cross-reference to another Freelance manual.
🖱	Selection with mouse. The various instruction steps are separated by arrows. **Example**: → *Edit* → *Insert Below* → ... . In this example, the menu item *Edit* is to be selected, followed by the menu item *Insert Below*.
🖱	Operating alternative with the mouse.
⌨	Information on operation with the keyboard, inasmuch as it differs from the Windows Standard. Example: Select module → ALT → E → D → ... . Having selected the module via the ARROW KEYS, the keys ALT, E and D must be pressed successively. If two keys are to be pressed simultaneously: ... → SHIFT + INSERT → ... .
⌨	Alternative keyboard operation
☞	Hints
⚠	Warnings and special hints, must be observed!

1 General

Within the hardware structure the resources defined in the project tree are allocated to the hardware actually required. A scalable Freelance 2000 system essentially consists of the process and operator stations. These stations are allocated to resources according to IEC 61131-3 and serve as a structuring level in the project tree for allocation of the application program parts and displays to the hardware actually required.

The D-PS resource stands for a **process station**. This accepts the I/O modules for process control, and is essentially determined by the CPU module. The CPU module processes all programs of this resource, assigns a station number to the process station, and is the communication module to the system and the I/O units. The process station consists of the central unit and up to four I/O units. The central unit is always the station with the CPU module. The I/O units do not need an own CPU module: they can accept other I/O modules so that their numbers can be expanded (from 8 for the central unit) to a maximum of 44 for a process station. All units need a link module for feeding in the power supply and an identical rack for holding the modules.

A single software tool suffices for making all necessary configuration and commissioning actions from a PC. Graphical user interfaces are available for diagnosing.

Efficient graphic editors allow you to configure your automation tasks in accordance with IEC 6-1131-3 and then download them into the process station in online mode.

In order to facilitate the visualization of your process, powerful visualization packages like DigiVis are available for your process stations.

1.1 System overview

The system view shows the graphic view of the hardware structure.
The system bus **(Ethernet)** links the individual stations with each other. It transmits data between the process stations, the operator stations and the engineering station via coaxial or fiber-optic cables.

The station bus **(CAN)** links the main rack with other racks or I/O units.

The **engineering station** is a PC or laptop with MS-Windows NT 4.0. It is used by the operator for system configuration, documentation and commissioning. After this has been done, it can be disconnected and used for other purposes.

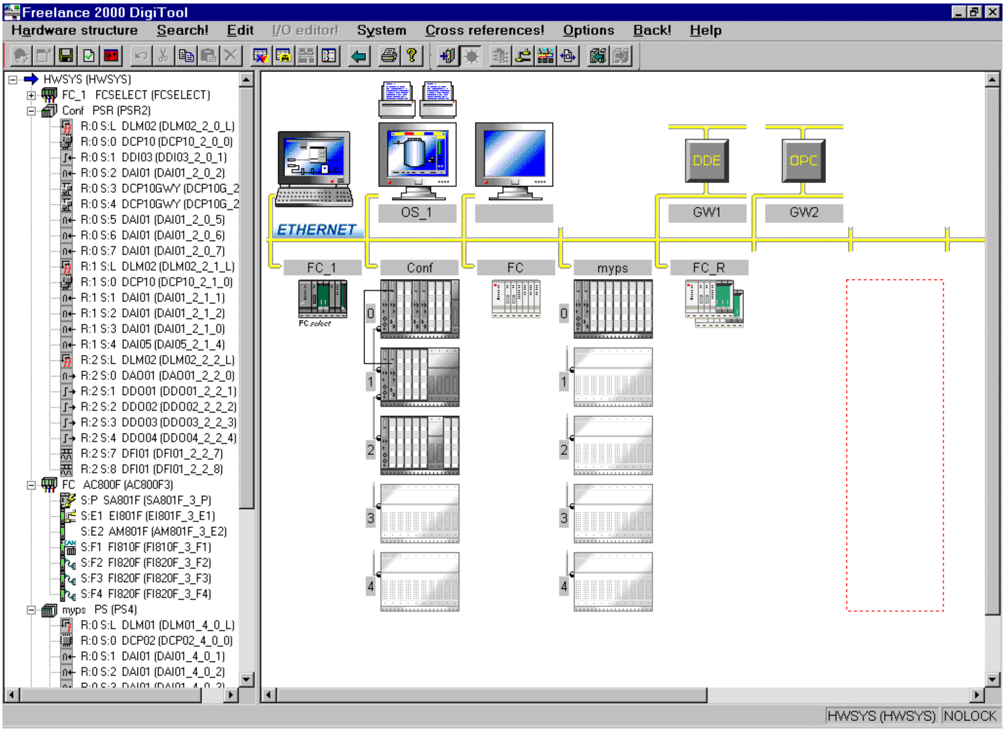
The **operator station** is based on a PC with MS-Windows NT 4.0. It is usable for configuration and commissioning as well, provided that the engineering software has been installed here.

A scalable Freelance system consists of operator stations and process stations/FieldControllers. The process stations consist of a central unit and up to four I/O units.

Up to two printers can be connected to each PC. The engineering station does not need to be activated separately. It is only used for displaying the network address.

Specific positions are assigned to the possible stations, either directly by double-clicking a position in the graphic view, or in a dialog when they are inserted in the tree structure.

→ System → Hardware structure → System object (HWSYS) → Edit → Insert



```
HWSYS (HWSYS)
  FC_1   FCSELECT (FCSELECT)
  Conf   PSR (PSR2)
    R:0 S:L  DLM02 (DLM02_2_0_L)
    R:0 S:0  DCP10 (DCP10_2_0_0)
    R:0 S:1  DDI03 (DDI03_2_0_1)
    R:0 S:2  DAI01 (DAI01_2_0_2)
    R:0 S:3  DCP10GWY (DCP10G_2
    R:0 S:4  DCP10GWY (DCP10G_2
    R:0 S:5  DAI01 (DAI01_2_0_5)
    R:0 S:6  DAI01 (DAI01_2_0_6)
    R:0 S:7  DAI01 (DAI01_2_0_7)
    R:1 S:L  DLM02 (DLM02_2_1_L)
    R:1 S:0  DCP10 (DCP10_2_1_0)
    R:1 S:1  DAI01 (DAI01_2_1_1)
    R:1 S:2  DAI01 (DAI01_2_1_2)
    R:1 S:3  DAI01 (DAI01_2_1_0)
    R:1 S:4  DAI05 (DAI05_2_1_4)
    R:2 S:L  DLM02 (DLM02_2_2_L)
    R:2 S:0  DAO01 (DAO01_2_2_0)
    R:2 S:1  DDO01 (DDO01_2_2_1)
    R:2 S:2  DDO02 (DDO02_2_2_2)
    R:2 S:3  DDO03 (DDO03_2_2_3)
    R:2 S:4  DDO04 (DDO04_2_2_4)
    R:2 S:7  DFI01 (DFI01_2_2_7)
    R:2 S:8  DFI01 (DFI01_2_2_8)
  FC   AC800F (AC800F3)
    S:P  SA801F (SA801F_3_P)
    S:E1  EI801F (EI801F_3_E1)
    S:E2  AM801F (AM801F_3_E2)
    S:F1  FI810F (FI810F_3_F1)
    S:F2  FI820F (FI820F_3_F2)
    S:F3  FI820F (FI820F_3_F3)
    S:F4  FI820F (FI820F_3_F4)
  myps   PS (PS4)
    R:0 S:L  DLM01 (DLM01_4_0_L)
    R:0 S:0  DCP02 (DCP02_4_0_0)
    R:0 S:1  DAI01 (DAI01_4_0_1)
    R:0 S:2  DAI01 (DAI01_4_0_2)
    R:0 S:3  DAI01 (DAI01_4_0_3)
```

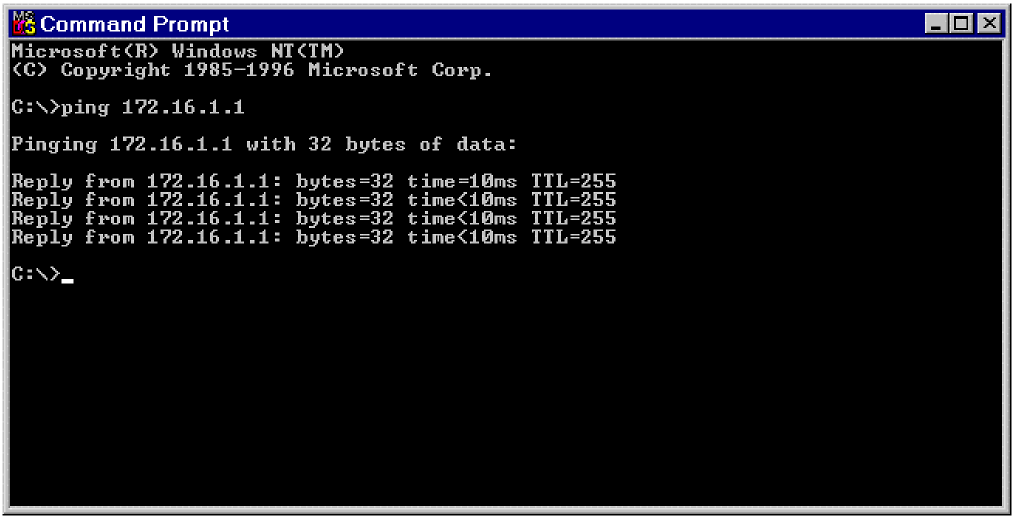
ra015us.bmp

2 Operating System and EPROM Download

2.1 Boot EPROM Download

The Boot EPROM on the DCP02/PCP10 must have the same version number as the engineering software. The Boot EPROM can be updated either in the installation phase from DigiTool, or at a later time from DigiTool Configure. The prerequisite is that the TCP/IP protocol is installed on the engineering station, and that there is an Ethernet connection to the process station.

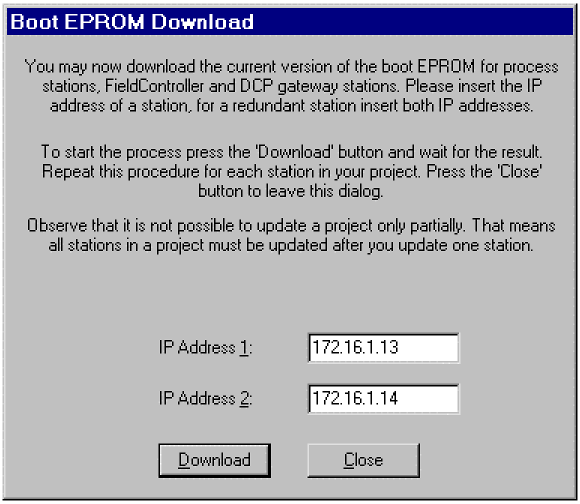
Prior to loading the Boot EPROM to a CPU you should check that the network connection between the engineering PC and the process station works properly. For this purpose, a "ping" is entered in the command prompt window under Windows NT and then transmitted to the process station. A positive response from the CPU (Reply from 172.16.1.5: bytes=32 time=10ms TTL=255) indicates that there is a network connection to the process station.



```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\>ping 172.16.1.1

Pinging 172.16.1.1 with 32 bytes of data:

Reply from 172.16.1.1: bytes=32 time=10ms TTL=255
Reply from 172.16.1.1: bytes=32 time<10ms TTL=255
Reply from 172.16.1.1: bytes=32 time<10ms TTL=255
Reply from 172.16.1.1: bytes=32 time<10ms TTL=255

C:\>_
```

ra002us.bmp

→ *Start* → *Programs* → Command prompt c:\> ping 172.16.1.5



Boot EPROM Download

You may now download the current version of the boot EPROM for process stations, FieldController and DCP gateway stations. Please insert the IP address of a station, for a redundant station insert both IP addresses.

To start the process press the 'Download' button and wait for the result. Repeat this procedure for each station in your project. Press the 'Close' button to leave this dialog.

Observe that it is not possible to update a project only partially. That means all stations in a project must be updated after you update one station.

IP Address 1: 172.16.1.13

IP Address 2: 172.16.1.14

ra001us.bmp

DOWNLOAD The current version of the Boot EPROM is downloaded to the process station with the given IP address(es).

CLOSE Terminates the EPROM update and completes the installation of the DigiTool version or the download of the Boot EPROMs.

If you have more than one process station in your project, repeat the steps described above for each of them. If you are using a redundant process station, enter both addresses (IP addresses 1 an 2). The Boot EPROM will then be loaded for both of the CPUs.

Taskbar → *Start* → *Programs* → *Freelance 2000 Tools V6.1* → *DigiTool Configure*

2.2 Operating system download

The operating system is downloaded into the station via the system bus by actuating **BOOTSTRAP**.

Prerequisites: Make sure that no operating system resides yet in the process station (RUN-STOP LED off). Otherwise, you have to **initialize all** prior to downloading the system.

The operating system bootstrap is normally executed automatically when the project is loaded for the first time by using the **Load whole station** option. However, it can also be performed upon a separate initialization action, independently of loading the user program. Downloading the operating system into the secondary CPU is not necessary, this is done automatically by the primary CPU.

3 Configuring the D-PS or D-PS/RED Resource in the Project Tree

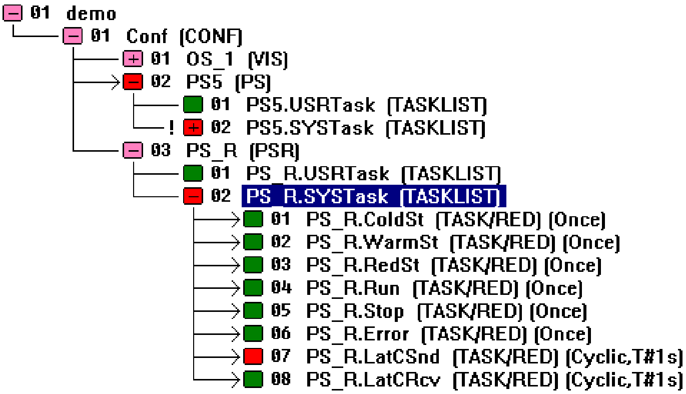
In the project tree the individual software components of a project are shown in a clear overview. The individual elements or objects, usually referred to as the **project objects**, are implemented according to the IEC 61131-3 standard. Data processing of process data actually takes place in the **D-PS resources** (**process stations**). Process operation and monitoring are realized in the **D-LS** resources (**operator stations**) or **D-GS resources** (**gateway stations**). Data from other systems are integrated by the **OPC server** resource.

The resources in the project tree have to be assigned to the hardware objects in the hardware manager, in order to define which program is to be executed by which process, operator or gateway stations.

The process station resource (**D-PS**) stands for a Freelance 2000 process station, on which the **resource** and the tasks and programs assigned to it are processed.

The short name **D-PS** or **D-PS/RED** in the project tree indicates that no allocation to a physical station has been made yet in the hardware manager. After the assignment has been made, the allocated station type PS (or PSR for a redundant process station) is indicated.

The illustration below shows a redundant and a non-redundant PS in the tree.



```
⊟ 01 demo
└──⊟ 01 Conf (CONF)
      ├──⊞ 01 OS_1 (VIS)
      ├─>⊟ 02 PS5 (PS)
      │     ├──🟩 01 PS5.USRTask (TASKLIST)
      │     └─! ⊞ 02 PS5.SYSTask (TASKLIST)
      └──⊟ 03 PS_R (PSR)
            ├──🟩 01 PS_R.USRTask (TASKLIST)
            └──⊟ 02 PS_R.SYSTask (TASKLIST)
                  ├─>🟩 01 PS_R.ColdSt (TASK/RED) (Once)
                  ├─>🟩 02 PS_R.WarmSt (TASK/RED) (Once)
                  ├─>🟩 03 PS_R.RedSt (TASK/RED) (Once)
                  ├─>🟩 04 PS_R.Run (TASK/RED) (Once)
                  ├─>🟩 05 PS_R.Stop (TASK/RED) (Once)
                  ├─>🟩 06 PS_R.Error (TASK/RED) (Once)
                  ├─>🟥 07 PS_R.LatCSnd (TASK/RED) (Cyclic,T#1s)
                  └─>🟩 08 PS_R.LatCRcv (TASK/RED) (Cyclic,T#1s)
```

ra003us.bmp

3.1 Inserting a D-PS or D-PS/RED resource

Select the target position in the project tree

→ *Insert above*
Inserts a new process station **above** the selected object.

→ *Insert below*
Inserts a new process station **below** the selected object

→ *Insert next level*
Inserts a new process station one level lower

The target position can be on the next lower level under the configuration element, or on the configuration level itself (with *Insert next level* only).

→ Position cursor on target position, click with left mouse button → OK

3.2 Inserting a task or redundant task

In the process station, the actual programs are run in the tasks. The programs are processed either with program lists or with the sequential function charts.
A redundant task has so-called redundancy data. These are data of the process image and of the function blocks. The data sets are matched with each other after each cycle. This ensures that a redundancy toggle can take place at any point in time.
So that the data of a redundant task are redundancy-capable, all variables of a redundant task must be written **in process image mode**.
Not only the redundant task (TASK/RED), but also tasks of the previous type (TASK) can be configured under the redundant resources. Thus only those functions of the user program which also have actually to be redundant need to be configured redundantly within a redundant resource.

4 Processing and Error Handling

4.1 Automatic error handling at task level

Error handling at task level is described in the flow chart (**see page 15**).

If an error is detected in a user program, the error task is executed once. The error task has the highest priority (priority 100) in a resource and is used for handling errors in user programs with user programs. If an unrecoverable error is detected, the error-producing task changes over to the unrunnable state. It the error is recoverable, the error-producing task can remain in the running state, provided that automatic error correction has been activated for the D-PS resource. Due to its high priority it cannot be interrupted by other tasks. However, the execution of the error task can be deactivated.

Error handling at task level is switched on by default, but can be deactivated in the resource configuration (see the **Engineering Manual, System Configuration, Project Tree**). When deactivated an error causes the task to change to the unrunnable state, even if the error is recoverable. The error appears in text form in the task header, and the object number of the faulty project object is also displayed.

If automatic error handling is active and a recoverable error is detected, the affected task remains in the running state. In this case, the error condition is ascertainable only be evaluation of the system variables.

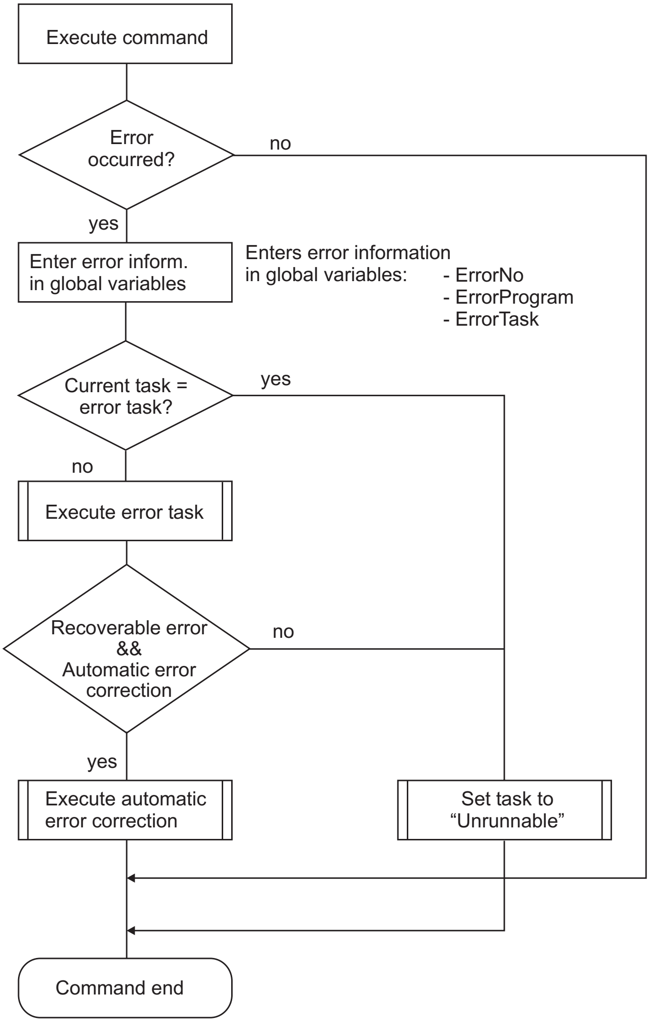
Each runtime error detected during execution of a command results in an entry recording the cause and location of the error in system variables specially defined for error handling. The cause of the error, e.g. 4 for UNIT Div by 0, is saved in the variable "ErrorNo". The variable "ErrorProgra" is written with the object number of the program or function block that caused the error. Furthermore, the object number of the affected task is saved in the variable "ErrorTask", but only displayed in the task header if error handling is deactivated. The affected object can then be localized in the object list with its name, state, type and position in the project tree. If the affected task, the one which triggered the error, is not the error-handling task, the error handling task itself is now triggered. Error handling specific to the cause of the error and/or its location can be undertaken in this task, with the aid of the system variables.
If an error should occur in the error-handling task itself, the error-handling task is signaled as **unrunnable**.

See also **Engineering Manual, IEC 61131-3 Programming, Variables, System Variables**.

4.2 Task error messages

Error #	Error text	Description
1	INT/DINT overflow	Signed integer or time operation caused overflow.
2	INT/DINT underflow	Signed integer or time operation caused underflow.
3	INT/DINT div by 0	Signed integer division by zero.
4	UINT/DINT div by 0	Unsigned integer division by zero
5	INT/DINT save overflow	Signed 16-bit-integer operation caused overflow on storage.
6	REAL overflow	Real operation caused overflow.
7	REAL underflow	Real operation caused underflow.
8	REAL div by 0	Real division by zero.
9	REAL invalid value	Real operation with undefined value
10	DT overflow	Operation with DT and TIME caused overflow.
11	DT underflow	Operation with DT and TIME or with DT and DT caused underflow
12	UINT/UDINT overflow	Unsigned integer operation caused overflow.
13	UINT/UDINT underflow	Unsigned integer operation caused underflow.
128	Program exec. error	Unloaded object (e.g. program, FB, etc.) to be executed.
129	FB error integer	A function block from the function block library has triggered an integer overflow or underflow, the error can only be detected following execution of the block and, thus, no recovery is possible.
130	FB error real	Function block from function block library triggered REAL overflow or underflow, error can only be detected after execution of the block and, thus, no recovery is possible.
131	Process image read error	Unrecoverable error occurred on reading process image.
132	Process image write error	Unrecoverable error occurred on writing process image

4.3 Task error characteristics


Execute command
Error occurred?
no
yes
Enter error inform. in global variables
Enters error information in global variables: - ErrorNo - ErrorProgram - ErrorTask
Current task = error task?
yes
no
Execute error task
Recoverable error && Automatic error correction
no
yes
Execute automatic error correction
Set task to “Unrunnable”
Command end

Error messages

Recoverable errors		Unrecoverable errors
1 INT_OVERFLOW	8 FLOAT_DIV_BY_ZERO	128 PROGRAM_EXECUTION_ERROR
2 INT-UNDERFLOW	9 FLOAT_NAN_VALUE	129 INT_FB_ERROR
3 INT_DIV_BY_ZERO	10 DT_OVERFLOW	130 FLOAT_FB_ERROR
4 UINT_DIV-BY_ZERO	11 DT_UNDERFLOW	131 PROZABB_READ_ERROR
5 INT_STORE_OVERFLOW	12 UINT_OVERLOW	132 PROZABB_WRITE_ERROR
6 FLOAT_OVERFLOW	13 UINT_UNDERFLOW	
7 FLOAT_UNDERFLOW		

di1503us.eps

4.4 Startup and shutdown performance of the modules

A signal for connection monitoring is permanently exchanged between I/O modules and the CPU module of the process station. If this signal does not appear for more than 250 ms (for 500 kbit/s) and/or 1,25 s (for 100 kbit/s) both sides detect the interruption of the connection. The CPU module sends a system message to the operator station and the output modules adopt the safety values. Dependent on the requests of the process the safety values can configured as "Hold the last value" or to a particular value/state.

A cold process station start always leads to output of the safety values.

For "Load whole station" and at resource Initialization/Overall reset, the output levels of the output modules go to zero current.

4.5 Power fail signal (PF) and no power fail

PF < 15 ms While PF signal is active, the CPU module of the process station does not communicate actual values; I/O modules maintain the last values.

PF > 15 ms I/O modules maintain the last values until they have recognized the interrupted connection with the CPU module; they then adopt the configured safety values. Safety values are maintained until the CPU module has executed a warm start and sends new values to the I/O modules.

4.6 Power fail signal (PF) and power fail

I/O modules maintain the last values until they have recognized the connection interruption and then adopt safety values until the CPU module sends actual values after restart. In the case of power fail of the I/O module, output levels adopt zero voltage and/or zero current.

After voltage return I/O modules only change their output level when the CPU module communicates actual values. Analog output modules are an exception. They hold the last values as long as external voltage supply does not fail. Outputs are only deenergized when the CPU module is powered again.

5 Configuring the Process Station in the Hardware Structure

Within the hardware structure the resources defined in the project tree are allocated to the hardware actually required. The D-PS resource stands for a process station in the hardware structure.

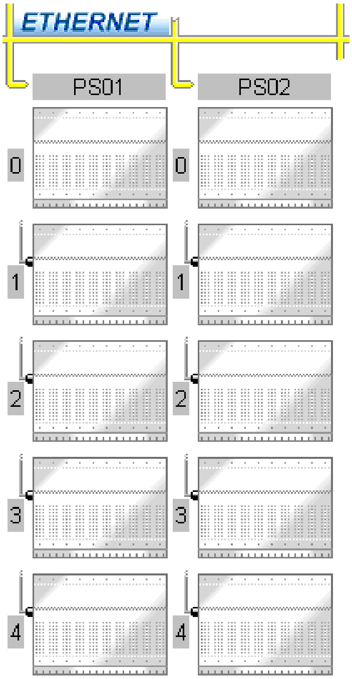
The rack-based process station consists of a basic unit (main rack) and the extension units (extension racks). Each rack accommodates various modules. A CPU module is required for every process station, and a link module DLM is needed for every rack. The modules must be plugged in special slots.
The I/O modules can be arranged in all racks as required. They can be plugged in any slots, except for those reserved for the DCP and DLM modules.

The allocation to the physical stations is made in the hardware manager. You can select here the process stations (PS) for rack-based systems and FieldControllers for connection of fieldbus modules. A distinction is made between redundant and non-redundant process stations. Already when inserting a resource you can thus define whether or not the tasks and programs assigned to it are to be processed in redundant or non-redundant mode.

A redundant process station of type D-PS/RED consists of two DCP10s linked via a redundancy link. In the project tree, they appear as one process station resource. When loading the process station, one of the two process stations becomes a primary and one the secondary station. Note that only the primary stations is shown in the project tree.

5.1 Inserting process stations

Process stations can be inserted in the system in either the tree view or system view. After *Insert* the object will appear in both the tree view and the system view. The object is shown in the respective station position, in graphical form. Double-click on the gray field in the header of the inserted process station to make the assignment of the resource.



ra005us.bmp

→ Select system object (HWSYS) in the tree view → *Edit* → *Insert*
→ Select station type → Select position

→ Select station position in system view → Double-click

5.2 Allocating a resource

When a station is allocated, it is automatically activated or a search operation is carried out in the DigiTool "Commissioning" mode to find the station on the system bus.

In this way, the parts of the user program configured in the project tree are allocated to the stations.

→ Left mouse click on the gray text field of the station
→ *Edit* → *Resource allocation*
→ Select the corresponding resource from the list

→ Double-click on the gray text field of the station
→ Select the corresponding resource from the list

5.3 Inserting modules

Modules can be inserted in the process station in either the tree view or the station view. After *Insert*, the object appears in both the tree view and the station view. The object is shown in the corresponding slot. In the station view, the modules can be equipped by double-clicking.

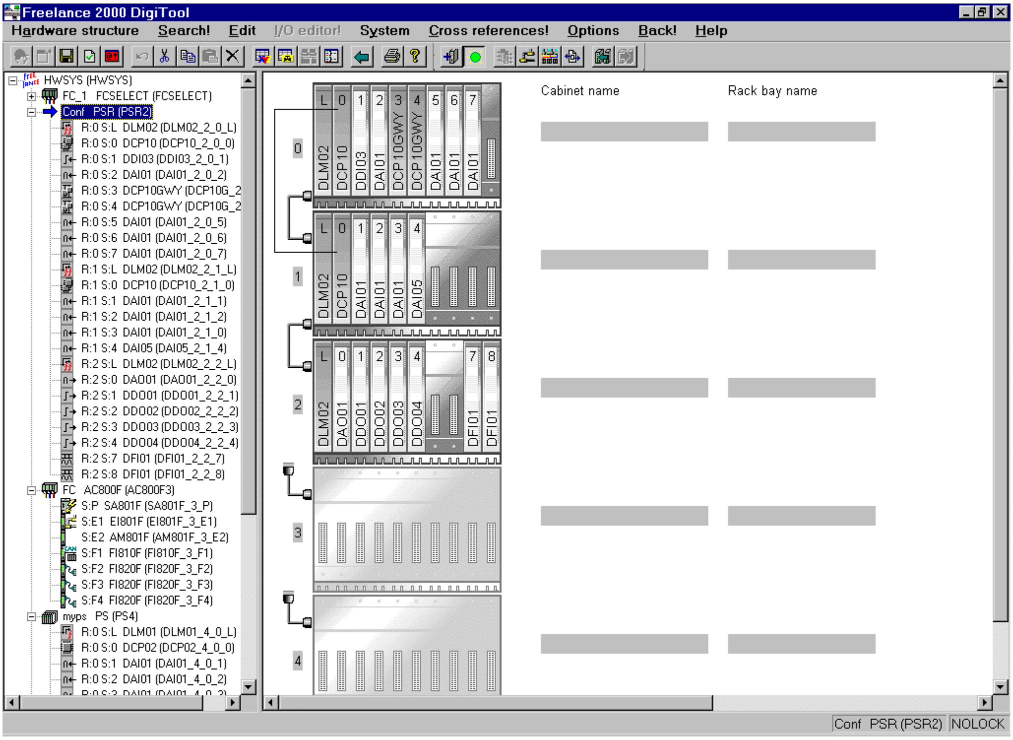
→ Select process station in the tree view → *Edit* → *Insert*
→ Select module type → Select a free slot

→ Select slot (L, 0 - 8) in the station view → Double click

The following modules are available:

DAI01	Analog input module 16 x 0/4...20mA, Ri=50 ohms
DAI02	Analog input module 16 x 0...10V DC
DAI03	Analog input module 16 x 0/4...20mA, Ri=250 ohms
DAI04	Analog input module 8 x Pt100 / mV
DAI05	Analog input module 16 x 0/4...20mA, MU supply
DAO01	Analog output module 16 x 0/4...20mA
DCO01	Communication module 4 x RS 485/RS 422 or RS 232
DCP02	CPU module
DCP02GWY	DCP gateway DCP for the Maestro UX gateway
DCP10	CPU module (capable of redundant operation)
DCP10GWY	DCP gateway DCP for the Maestro UX gateway
DDI01	Digital input module 32 x 24V DC
DDI02	Digital input module 16 x 24...60V AC/DC
DDI03	Digital input module 16 x 90...230V AC
DDI04	Digital input module 28 x 24 V DC NAMUR initiators
DDI05	Digital input module 32 x115/230 V AC
DDO01	Digital output module 32 x 24 V DC, 0.5 A
DDO02	Digital output module 16 x 24...230 V AC/DC
DDO03	Digital output module 16 x 24...60 V AC/DC, read back
DDO04	Digital output module 16 x 115...230 V AC, read back
DFI01	Frequency input module 4 x f <=45kHz
DLM01	Link module for power supply
DLM02	Link module with redundant power supply

When an object is inserted, a default name is assigned to it which basically represents the mounting position. When the object is copied, the object name is adapted to the new position. When the object is moved, the name remains unchanged.



```
HWSYS (HWSYS)
  FC_1  FCSELECT (FCSELECT)
  Conf  PSR (PSR2)
    R:0 S:L  DLM02 (DLM02_2_0_L)
    R:0 S:0  DCP10 (DCP10_2_0_0)
    R:0 S:1  DDI03 (DDI03_2_0_1)
    R:0 S:2  DAI01 (DAI01_2_0_2)
    R:0 S:3  DCP10GWY (DCP10G_2
    R:0 S:4  DCP10GWY (DCP10G_2
    R:0 S:5  DAI01 (DAI01_2_0_5)
    R:0 S:6  DAI01 (DAI01_2_0_6)
    R:0 S:7  DAI01 (DAI01_2_0_7)
    R:1 S:L  DLM02 (DLM02_2_1_L)
    R:1 S:0  DCP10 (DCP10_2_1_0)
    R:1 S:1  DAI01 (DAI01_2_1_1)
    R:1 S:2  DAI01 (DAI01_2_1_2)
    R:1 S:3  DAI01 (DAI01_2_1_0)
    R:1 S:4  DAI05 (DAI05_2_1_4)
    R:2 S:L  DLM02 (DLM02_2_2_L)
    R:2 S:0  DAO01 (DAO01_2_2_0)
    R:2 S:1  DDO01 (DDO01_2_2_1)
    R:2 S:2  DDO02 (DDO02_2_2_2)
    R:2 S:3  DDO03 (DDO03_2_2_3)
    R:2 S:4  DDO04 (DDO04_2_2_4)
    R:2 S:7  DFI01 (DFI01_2_2_7)
    R:2 S:8  DFI01 (DFI01_2_2_8)
  FC  AC800F (AC800F3)
    S:P  SA801F (SA801F_3_P)
    S:E1  EI801F (EI801F_3_E1)
    S:E2  AM801F (AM801F_3_E2)
    S:F1  FI810F (FI810F_3_F1)
    S:F2  FI820F (FI820F_3_F2)
    S:F3  FI820F (FI820F_3_F3)
    S:F4  FI820F (FI820F_3_F4)
  myps  PS (PS4)
    R:0 S:L  DLM01 (DLM01_4_0_L)
    R:0 S:0  DCP02 (DCP02_4_0_0)
    R:0 S:1  DAI01 (DAI01_4_0_1)
    R:0 S:2  DAI01 (DAI01_4_0_2)
    R:0 S:3  DAI01 (DAI01_4_0_3)
```

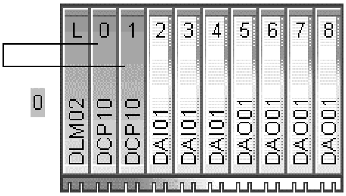
ra009us.bmp

Example: DCP10 in the first extension rack
R:1 S:0 DCP10 (DCP10_2_1_0)

R:	Rack)
S:	Slot
DCP10	Default name of the module
DCP10_2_1_0	Default name with resource 2,rack 1, slot 0

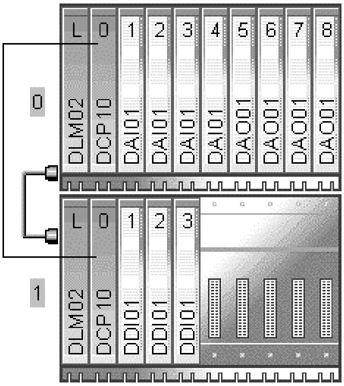
5.4 Redundant process station

A redundant process station always consists of two DCP10 module mounted in a resource. Both DCP10 modules can be integrated in the station view of the redundant process station by double-clicking with the mouse. Note that these modules can be mounted in slot 0 or 1 of the basic unit, only.



ra008us.bmp

It is also possible to mount one redundant DCP10 module in the basic unit, and the second one in the extension rack.



ra007us.bmp

5.5 Defining parameters of the process station

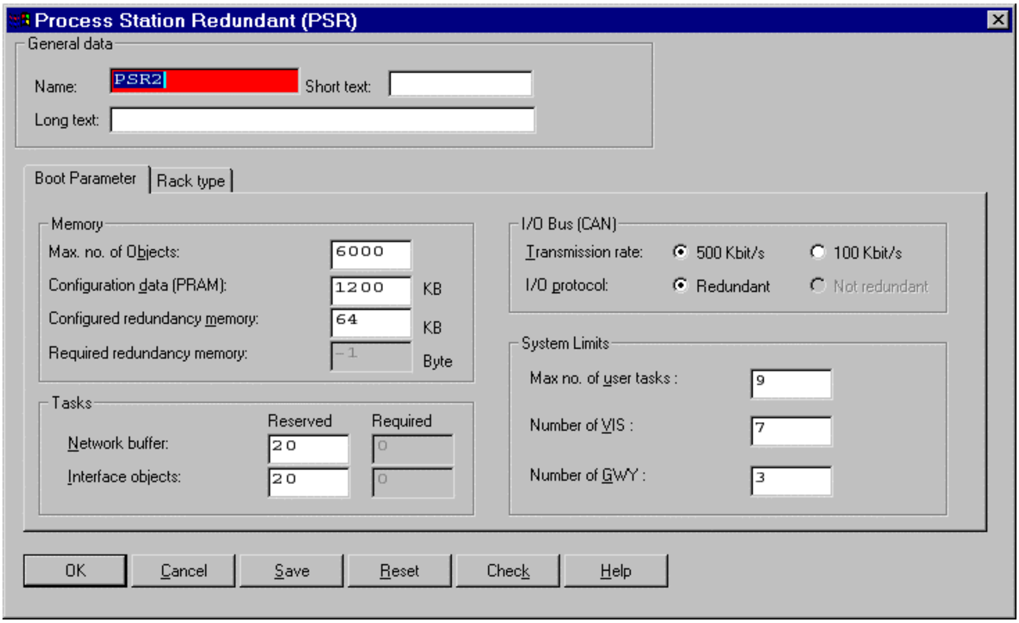
The station must be allocated to a resource (D-PS) to define which part of the user program is to be run on which process station.
You can access the PS parameters by selecting the PS in the tree view or graphics view.

Select PS in the tree view → Click with right mouse button → *Parameters*

Select PS in the tree view → *Edit* → *Parameter Setting*

The boot parameters and the rack data can be defined.

5.5.1 Boot parameters



Process Station Redundant (PSR)

General data

Name: PSR2

Short text:

Long text:

Boot Parameter | Rack type

Memory

Max. no. of Objects:	6000	
Configuration data (PRAM):	1200	KB
Configured redundancy memory:	64	KB
Required redundancy memory:	-1	Byte

Tasks

	Reserved	Required
Network buffer:	20	0
Interface objects:	20	0

I/O Bus (CAN)

Transmission rate: ● 500 Kbit/s ○ 100 Kbit/s

I/O protocol: ● Redundant ○ Not redundant

System Limits

Max no. of user tasks :	9
Number of VIS :	7
Number of GWY :	3

ra013us.bmp

Prior to changing the boot parameters always initialize the resource. Otherwise, all changes to configurable boot parameters in a resource will cause a project version error.

Memory These parameters have an effect on the allocation of memory within the resource. When there are an extremely large number of objects in the process station, errors may occur on loading the objects; these settings may be altered to try to make the project loadable.

Attention should also be paid to the system variables PRAM_Free and RAM_Free after loading. If the variable PRAM_Free has a value of 0, then the value of PRAM_Size must be increased; if the variable RAM_Free is set to 0, then both PRAM_Size and the maximum number of objects must be reduced.

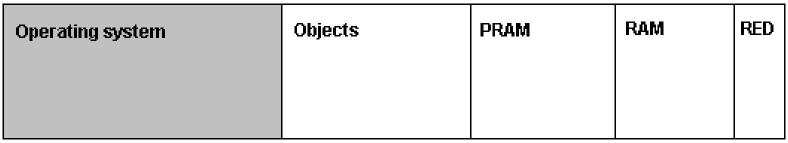
The memory for the Freelance 2000 CPU modules (DCP 02, DCP 10) is limited to 8 Mbytes. The memory can be subdivided as listed below:

Memory for operating system
+ Memory for the object directory
+ Memory for PRAM
+ Redundancy memory
+ Memory for RAM

Total memory = 8 Mbytes

The memory for the operating system is separated into two parts: One part of fixed size and another one which size depends on the number of communication connections. The memory for the objects, memory for PRAM and, if available, memory for redundancy can be adjusted in the boot parameters of the resource.

Any surplus memory is made available in RAM. There is no absolute limit for the memory ranges that can be set. Any time that a limit is reached, it can be changed. This results in a reduction of free RAM.



ra011us.bmp

To ascertain a suitable value for PRAM, current values can be read in commissioning mode. These values are shown in the following **system variables**:

PS.PRAM_SIZE	The PRAM size set in the boot parameters
PS.PRAM_FREE	The proportion of free PRAM
PS.RAM_SIZE	The amount of RAM in use
PS.RAM_FREE	The proportion of free RAM

Max. no. of objects The maximum number of objects that can be loaded on the resource

Configuration date (PRAM) Memory size in Kbytes that is reserved for the configuration data. This memory area is cold-start resistant.

Configured redundancy memory Memory size in Kbytes that is reserved for the transfer of redundancy data.

Max. redundancy memory required Maximum size of memory required by the project for transfer of redundancy data.

☞ The redundancy memory is only shown in redundant and plausible process stations.

Tasks

Information about the tasks that are run in the background. Every communication link occupies internal system resources like additional memory or system objects.

Network buffer

Area of memory in Kbytes that is reserved for the communication links.

Interface objects

Maximum number of objects that can be reserved for interfaces.

I/O-Bus (CAN)

Transmission rate

Select a transmission rate of 100 or 500 Kbit/s of the station bus DigiNet P (default 500 Kbit/s). The transmission rate is closely linked to the max. distance between two stations of one process station.

Transmission rate	Distance to last I/O unit
500 Kbit/s	80m
100 Kbit/s	400m

Also refer to the **Mounting and Installation Instructions**, **Setting Link Module**, and to the **Engineering Manual, System Configuration, Hardware structure**.

If an I/O module with a version-2-compatible EPROM version (not suitable for redundant operation) is plugged into a later version of process station, and if the DigiNet-P station bus transmission rate (rack ID) that is set on the link module does not match the transmission rate configured for the resource, this will lead to a communication error in the I/O modules! This can lead to the modules' outputs dropping out!
I/O modules later than version 2 no longer check the setting of the coding switch on the link module. In the interest of safety, the coding switches on the link modules should be set to match the configured transmission rates. See also manual **Mounting and Installation Instructions, Adjusting the link module**.

I/O protocol

Select the I/O protocol

Redundancy

● Default, - I/O EPROM’s capable of redundant operation must be plugged on all I/O modules.

Not redundant

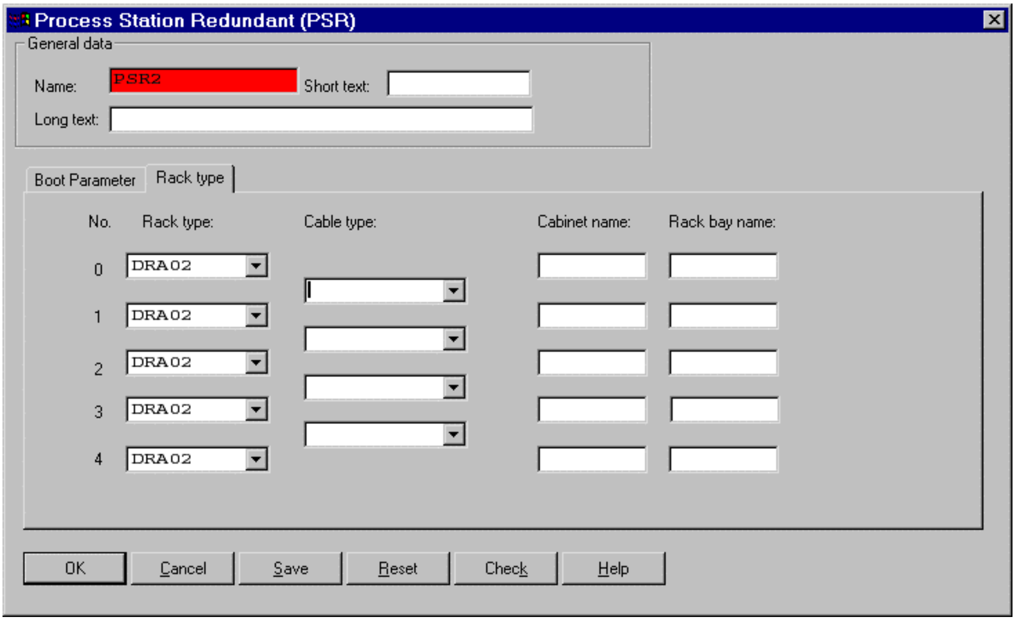
● Under this protocol redundancy is not supported.
Designed for operation with a mixture of not redundant I/O modules (until version 2), if redundancy is not required. This mode can only be selected in non-redundant resources.

Max. no. of User tasks: Max. number of tasks to be configured for this resource.

No. of VIS links: Number of permissible communication links to the operator stations (DigiVis).
See also **Engineering Manual, System Configuration, Project Tree, Links**

No. of GWY links: Number of permissible communication links between the resource and the gateway stations (e.g. DDE, OPC).
See also **Engineering Manual, System Configuration, Project Tree, Configuration of Communication Links**

5.5.2 Rack



Name: PSR2

Short text:

Long text:

No.	Rack type:	Cable type:	Cabinet name:	Rack bay name:
0	DRA02			
1	DRA02			
2	DRA02			
3	DRA02			
4	DRA02			

ra014us.bmp

I/O Racks

Rack type	Rack selection DRA 01 (delivery until end of 1997) DRA 02 with 10 module slots DRA 03 with 3 module slots DRA 04 with 5 module slots
Cable type	CAN cable for station bus DigiNet P

Location name

Cabinet name	freely selectable
Rack bay name	freely selectable

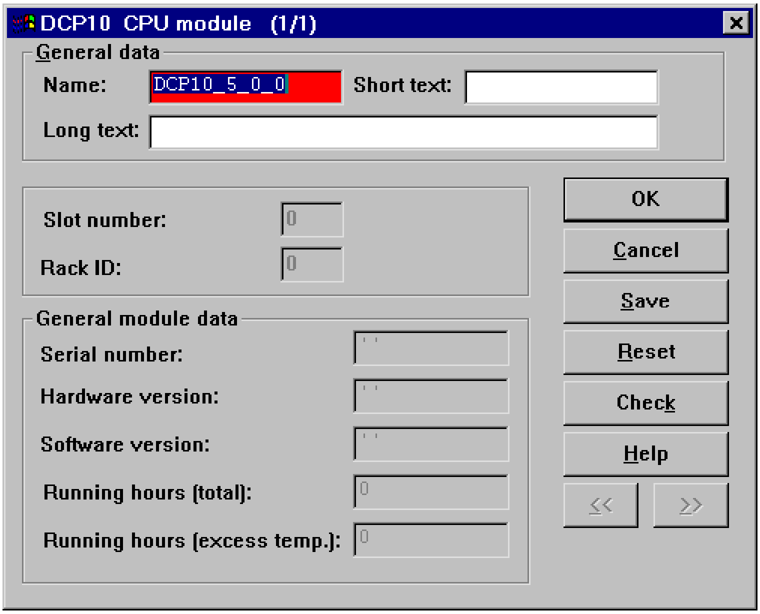
For cable length see **Mounting and Installation Instructions**, **Mounting the modules.**

☞ Use the standard setting if the total distance between central unit and I/O units is less than 80 m. With this setting the process station bus operates with normal data rate, i.e. 500 kbit/s, in V1 and V2 systems.
For distances between central unit and I/O units of more than 80 m the long distance setting has to be used. With this setting, the process station bus operates with a reduced data rate of 100 kbit/s. Thus, data transmission to the I/O modules is five times slower. In this case, the distance between central unit and I/O units is limited to max. 400 m.
With version 3 or higher the transmission rate of the DigiNet P is set using the DigiTool configuration software. See also **Engineering Manual**, **System Configuration**, **Project Tree**, **Process station boot parameters**.

☞ For racks DRA 03 and DRA 04, the number of module slots is reduced. Otherwise the behavior is equivalent to rack DRA 02

5.6 Defining module parameters

The module parameters can be called up for display via the parameter setting masks. The current values can also be read in the commissioning mode.



DCP10 CPU module (1/1)

General data

Name: DCP10_5_0_0

Short text:

Long text:

Slot number: 0

Rack ID: 0

General module data

Serial number: ' '

Hardware version: ' '

Software version: ' '

Running hours (total): 0

Running hours (excess temp.): 0

ra010us.bmp

→ Select module in tree view → Click with right mouse button → *Parameters*

→ Select module in tree view → Double-click on module in graphic view

Example: CPU module DCP10

Serial number	Indicates the sequential serial number from manufacturing.
Hardware version	Indicates the hardware version.
Software version	Indicates the version of the operating system
Running hours (total)	Indicates the total number of running hours.
Running hours (excess temp.)	Indicates the number of running hours at excess temp. (Excess temperature in the module: more than 70 °C).

These current values are only available in the commissioning mode.

5.6.1 Link modules DLM 01/02

Link module with power supply input of the process station and external battery connector. One link module must be mounted in each central unit and in each I/O unit. Link modules must be plugged in slot L (first slot on the left hand side of the rack).

Two link module models are available: the standard link module DLM 01 with single power supply, and link module DLM 02 with a redundant power supply meeting high power reliability requirements.

Object data: Link module DLM 01

Name	Data type	Comment
Diagnostic data		
BTL	BOOL	Battery low
ERR	BOOL	Module defective
STA	UDINT	Bit-coded error status

Object data: Link module DLM 02

Name	Data type	Comment
Diagnostic data		
TMP	BOOL	Excess temperature of the I/O module
PF1	BOOL	Power-fail channel 1
PF2	BOOL	Power-fail channel 2
BTL	BOOL	Battery low
ERR	BOOL	Module defective
STA	UDINT	Bit-coded error status

☞ The BTL signal is only set if a CPU module is present in this rack.

5.6.2 CPU modules DCP 02/10

Object data: CPU module DCP02

Name	Data type	Comment
Parameter setting mask		
RSState	INT	Position of the run/stop switch. 0 - Run, 1 - Stop
Diagnostic data		
TMP	BOOL	Excess temperature
BTL	BOOL	Battery low
ERR	BOOL	Module defective
STA	UDINT	Bit-coded error status

Object data: CPU module DCP10

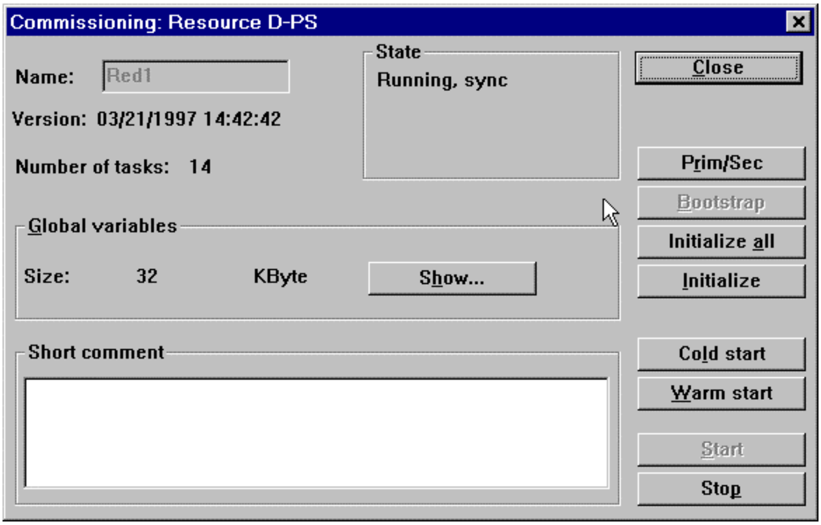
Name	Data type	Comment
Parameter definition mask		
RSState	INT	Position of the run/stop switch. 0 - Run, 1 - Stop
RedState	INT	Redundancy state. 0 - Primary, 1 - Secondary
Diagnostic data		
TMP	BOOL	Excess temperature
BTL	BOOL	Battery low
PNE	BOOL	Physical network down
ERR	BOOL	Module defective
STA	UDINT	Bit-coded error status

6 Commissioning the Process Station

A redundant resource is loaded like a non-redundant resource in commissioning. The process station can be loaded with the selection of the resource. One of the two CPU modules automatically becomes Primary CPU and the other Secondary. It is not possible to influence which of the two CPU modules becomes Primary.
All I/O modules retain their previous value in a redundancy toggle.

☞ A redundancy toggle can also be activated with the system function „Force redundancy toggle (PRIM/SEC)“.

6.1 Operating via resource in the project tree



Commissioning: Resource D-PS

Name: Red1

Version: 03/21/1997 14:42:42

Number of tasks: 14

State
Running, sync

Global variables
Size: 32 KByte

Short comment

di7161us.bmp

PRIM/SEC A redundancy toggle is initiated (toggle between Primary and Secondary CPU). A toggle is possible only in the SYNC status

BOOTSTRAP The Primary CPU is loaded with the operating system and is initialized.

INITIALIZE ALL Both CPU modules are initialized. The Primary CPU again becomes first active and can then be bootstrapped. After INITIALIZE ALL the redundancy is no longer active.

INITIALIZE In the initialization, the process station performs a cold start. Any user program already present is also deleted. The redundancy is now no longer active. The Secondary CPU moves to the "no OS” status.

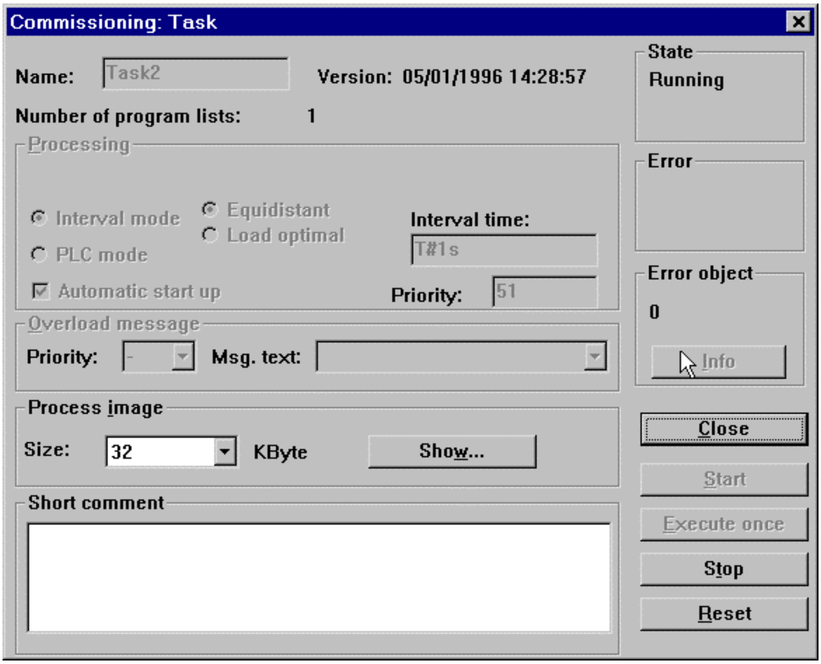
COLD START Both CPU modules perform a cold start.

WARM START Both CPU modules perform a warm start. The CPU module previously active again becomes Primary.

RUN/STOP Both CPU modules perform the command (Run or Stop). The LED at the Primary CPU indicates the current status. After a redundancy toggle, the last status is retained, irrespective of the RUN/STOP switch position.

6.2 Task

Select project object task → *Project* → *Header*



Commissioning: Task

Name: Task2

Version: 05/01/1996 14:28:57

Number of program lists: 1

Processing

● Interval mode

● Equidistant

○ Load optimal

○ PLC mode

☑ Automatic start up

Interval time: T#1s

Priority: 51

Overload message

Priority: -

Msg. text:

Process image

Size: 32 KByte

Short comment

State: Running

Error

Error object: 0

di1526uk.bmp

If configured for autostart, the task starts automatically when the resource is started. All lower-level program lists and programs start along with the task as long as the program list parameter has been set **ON**.

Task states are explained on **page 38**. For further information on tasks **see Engineering Manual, System Configuration, Project Tree**.

START — The selected task is started. It is processed either in the cycle or as soon as possible (SPS mode), depending on configuration.

EXECUTE ONCE — The selected task is executed precisely once. Task subsequently reverts to the stopped state.

STOP — The selected task is stopped.
Task can also be stopped by using the RUN/STOP switch on the CPU module.

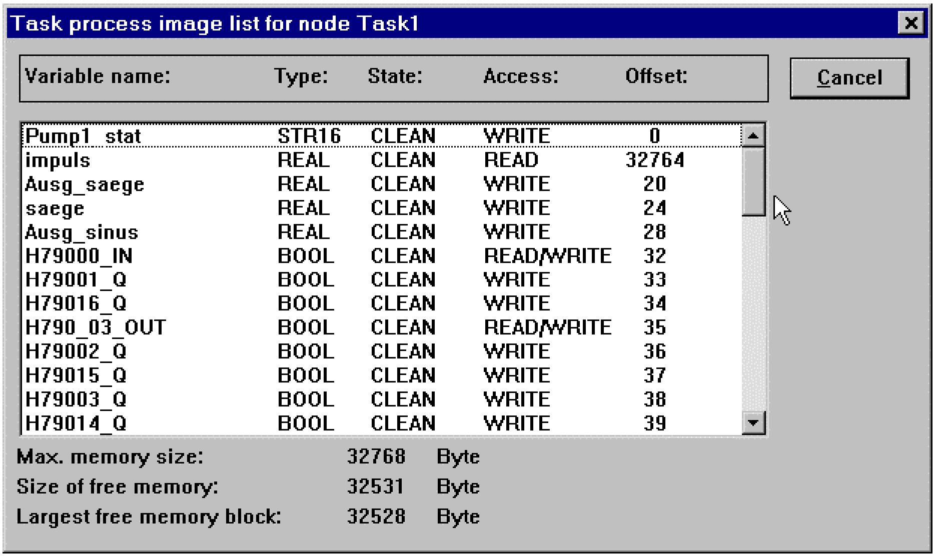
RESET — Current task reverts to the configured state.

CLOSE — OK exits the dialog box and returns to project tree.

SHOW... — Selected task's process image is displayed. Process image size can be configured. Process image is displayed automatically when the operating system is loaded.

6.2.1 Process image display

Select project object task → *Project* → *Header*
→ SHOW...



Task process image list for node Task1

Variable name:	Type:	State:	Access:	Offset:
Pump1_stat	STR16	CLEAN	WRITE	0
impuls	REAL	CLEAN	READ	32764
Ausg_saege	REAL	CLEAN	WRITE	20
saege	REAL	CLEAN	WRITE	24
Ausg_sinus	REAL	CLEAN	WRITE	28
H79000_IN	BOOL	CLEAN	READ/WRITE	32
H79001_Q	BOOL	CLEAN	WRITE	33
H79016_Q	BOOL	CLEAN	WRITE	34
H790_03_OUT	BOOL	CLEAN	READ/WRITE	35
H79002_Q	BOOL	CLEAN	WRITE	36
H79015_Q	BOOL	CLEAN	WRITE	37
H79003_Q	BOOL	CLEAN	WRITE	38
H79014_Q	BOOL	CLEAN	WRITE	39

Max. memory size:	32768	Byte
Size of free memory:	32531	Byte
Largest free memory block:	32528	Byte

di1529uk.bmp

Variable name — Variable name, max. 16 characters

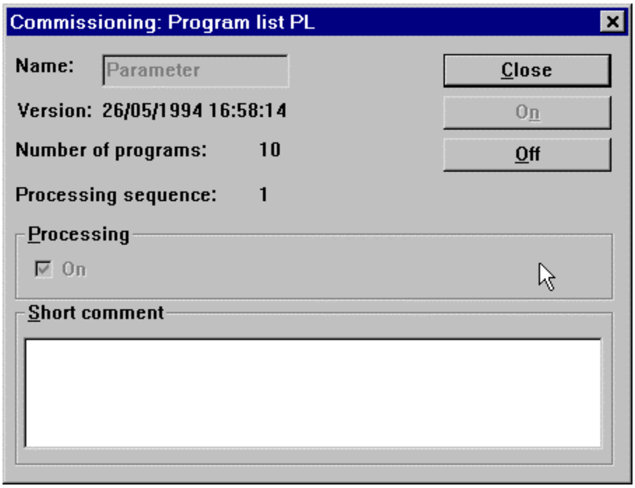
Type — Data type, e.g. REAL, BOOL, UINT etc.

State	CLEAN	Object is correct and loaded.
	DIRTY	Object version in engineering station does not match that in the station.
	CREATE	Object not yet loaded into station.
	DELETE	Object deleted in project database, but still present in station.
Access	READ	Read-only,
	WRITE	Write-only,
	READ/WRITE	read and write.
Offset	Memory address offset	

See also **Engineering Manual, System Configuration, Project Tree, Process Image**

6.3 Program list (PL)

🖱 Select project object program list → *Project* → *Header*



Commissioning: Program list PL

Name: Parameter

Version: 26/05/1994 16:58:14

Number of programs: 10

Processing sequence: 1

Processing

☑ On

Short comment

di1533uk.bmp

Processing

☑ On — Selected program list is switched on, and with it all lower-level programs. All function blocks with configured processing state set to **ON** are evaluated with each task cycle.

□ On — Selected program list is switched off, and with it all lower-level programs. Processing of all function blocks is terminated immediately.

OK — OK exits the dialog box and returns to the project tree.

On — Processing of program list is switched on.

Off — Processing of program list is switched off.

6.4 State displays in the project tree

In contrast to configuration, state information is maintained on project objects during commissioning. This state information appears after each project object in brackets (like the object types) and is updated as it changes. The state information reflects the state of this object in the process station. If arrows should appear before the nodes, these project objects must first be loaded or reloaded into the station because of a configuration change. An exclamation point indicates that other objects at levels below the one so marked need to be updated because of changes in their configuration.
The color of the node in the display provides information about the effects of its configuration change on other objects. Higher-level information may also need to be updated in the station.
In commissioning, **nodes** are displayed in the same formats as in configuration.
See **Engineering Manual, System Configuration, Project Tree, Display of project object states**

6.4.1 Resource state displays

Resource state information reflects the state of the process stations. The following details may appear after the object name and type.
All states except **Running** are displayed in red.

No connection — The system bus connection has been interrupted, or **initialize all** is currently being executed, or no connection is possible, e.g. because of an invalid network address or an incorrect DigiVis station number.

☞ Any interruption in the DigiNet S system bus connection will be signaled after approx. 15 seconds.

No operating system Seen when system is first switched on, or after **initialize all** of resource

Stopping Resource in transition from **Running** to **stopped.**

☞ This state is only displayed if task processing takes a relatively long time.

Stopped [HW] Resource has been **Stopped** by utilizing the RUN/STOP switch on the CPU module

Stopped [SW] Resource → **Stop** was selected in commissioning

Cold start stopped The command to cold start the resource had been initiated by the person responsible for commissioning, or from the system function "Force cold start". Indicates self-test of CPU module was OK, but, the RUN/STOP switch on the CPU module was in the STOP position or the resource was stopped by the person responsible for commissioning before achieving **running** state.

☞ The cold start task, *Name*.ColdSt (where *Name* = resource name) is executed only when the resource is started (STOP to RUN). When cold starting, all the resource data is initialized. The operating system is not affected and the output module channels assume their safety values.

Warm start stopped The command to warm start the resource was initiated by the person responsible for commissioning or a power failure (power-fail signal from the power supply unit), the RUN/STOP switch on the CPU module was set to STOP or the resource was stopped before achieving **running** state by the person responsible for commissioning.

☞ The warm start task, *Name*.WarmSt (where *Name* = resource name) is executed only when the resource is started (STOP to RUN). When warm starting, all data and variable values of the resource are unaffected and the output module channels assume their safety values.

Starting Resource in transition from **stopped** to **running**

☞ This state is only displayed if task processing takes a relatively long time.

Running Resource started and RUN/STOP switch in RUN position

Running partially Not all resource program objects are processing, e.g. task in stop or program list OFF

Version error, running There are user program version disparities between the active station and the assigned resource. Such disparities are displayed independently of the run state of the resource, and are therefore also displayed when stopped.

[del2] Resource objects have been deleted, in this case, two objects

Wrong station number The station number in the process station (CPU module) does not match the station number in the hardware structure (network).

Incompatible EPROM version The EPROM version of the CPU module does not match that of DigiTool.

☞ The boot EPROM must be updated. See **Engineering Manual, System Configuration, Installation**.

Additional states of a redundant resource

Not redundant The operating system is loaded but the user program has not yet been completely loaded.

No secondary CPU Operating system and user program are completely loaded. Secondary CPU module has not yet been announced to the Primary CPU module.

Not sync Secondary CPU module is present, operating system and user program are loaded but processes are not yet updated and synchronization is thus not yet concluded.

Sync Normal state of a redundant process station. Secondary CPU module is synchronized and ready for redundancy switching. Cyclic data mirroring between the CPU modules continues during program processing.

Redundancy error Redundancy is not possible.

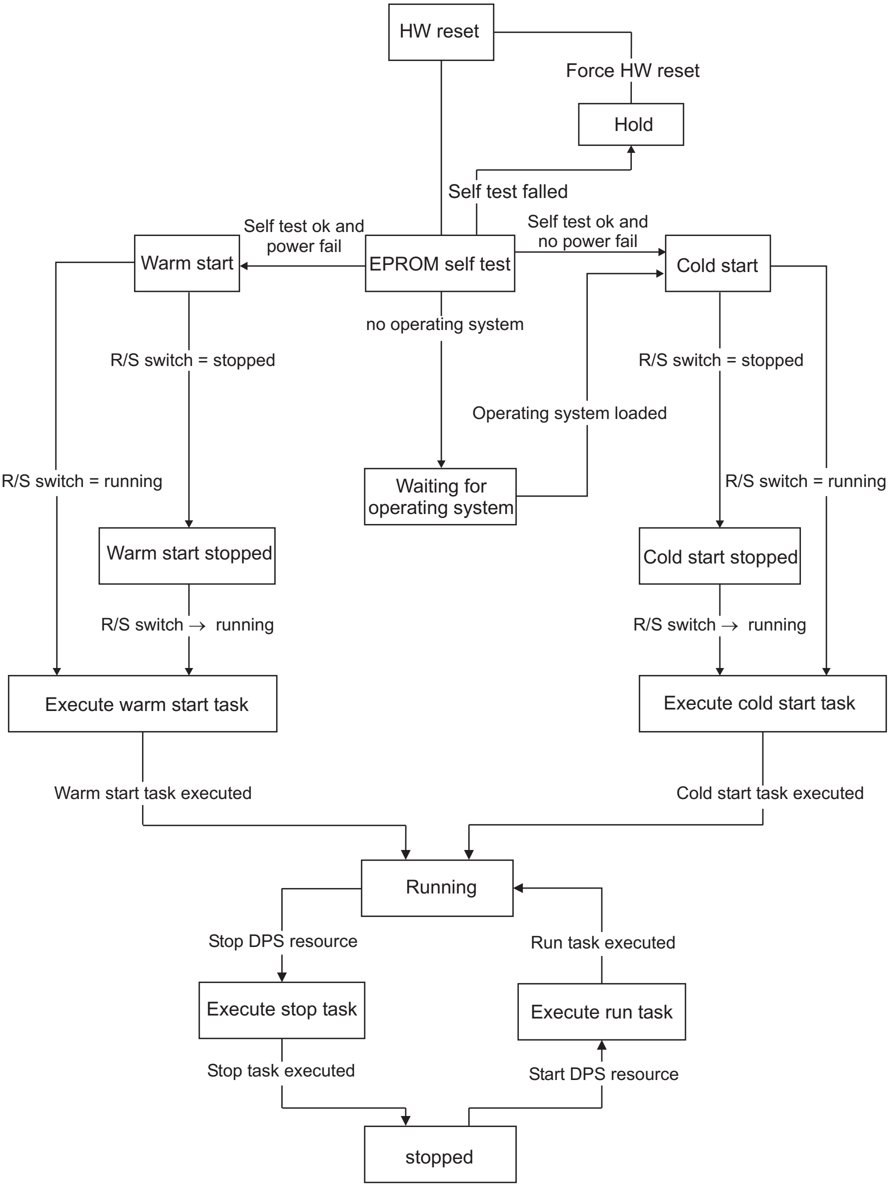
☞ CPU module can not achieve Sync state. It has tried to reach “sync” state repeatedly with resource stop and start operations. Possible reason may be unsuccessful loading of user program.

☞ System workload is too heavy; the redundant connection can not be maintained. Task cycle times are not configured correctly.

☞ The quantity of redundant data to be transferred for a task is too high. System variable xxxx.RedBufLow displays 0 memory remaining for redundant data. Spread user program over more tasks.

6.4.2 Resource state diagram

State diagrams of the redundant CPU modules are **see page 48, Redundancy status.**



di1501us.eps

6.4.3 Task state displays

Ready Task already loaded and manual start possible.

☞ If no **autostart** has been configured for a task, it reverts to the **ready** state after each cold start and has to be restarted.

Non-existent Task is loaded but cannot be started. A software version error exists.

Not loaded A change to a task, or to a program within the task, which has a side effect on the task, has not yet been loaded.

Unrunnable During execution of the task either an unrecoverable error occurred or a recoverable error occurred while automatic error correction was switched off.

☞ Automatic error correction for a task is switched on in the default setting. If automatic error correction is switched off, the task changes to the **unrunnable** state on any error, even if the error is recoverable. The error appears in text form in the task header and the object number of the faulty project object is also displayed.

☞ An exception is constituted by tasks which, although loadable, have no lower-level user program. Such a task is thus unable to initiate execution of a program, and if started nevertheless, it goes into the unrunnable state.

Running After manual **start** of the task or when the task has been set up with **Autostart**.

☞ When a higher-level resource is stopped and re-started, the lower-level task reverts to its old state. A task with **Autostart** changes automatically to **running**. A task's state may also be changed when the resource is stopped, and the changed state will be reassumed when the resource is restarted.

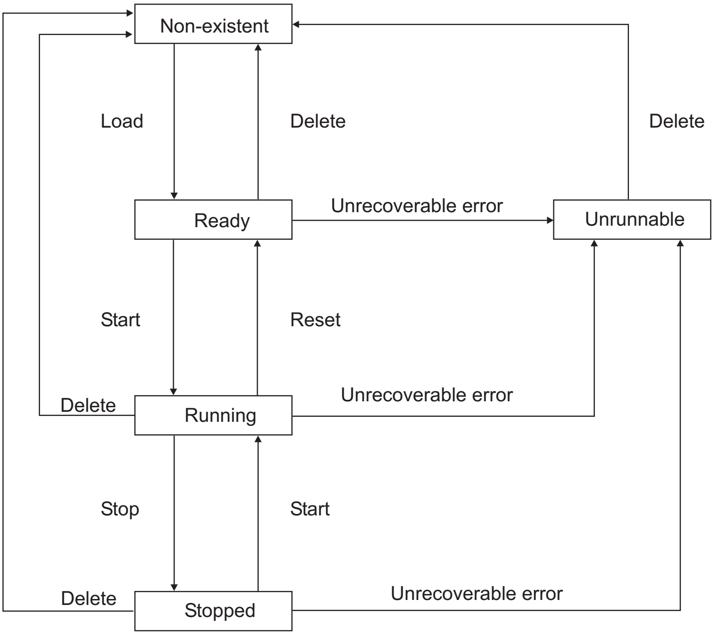
Stopped After execution of **Stop** task, **Stop** station (HW) or **Stop** higher-level resource (SW).

☞ If a stopped task is reset to the ready state (not visible) with **Reset**, no change to its state of **stopped** is displayed to the user. When **Stop** is changed to **Start**, the RUN task is executed.

⚠ A task starts automatically if the switch on CPU module is switched to **RUN** and previous state of the task was **Running**.

6.4.4 Task state diagram

☞ Unrecoverable errors are errors in configuration, for example division by zero. In this case the task, and in particular the program, should be checked.



di1502us.eps

6.4.5 Program list state displays

versions error Program list was loaded but the versions do not match

not loaded Program list is not yet loaded

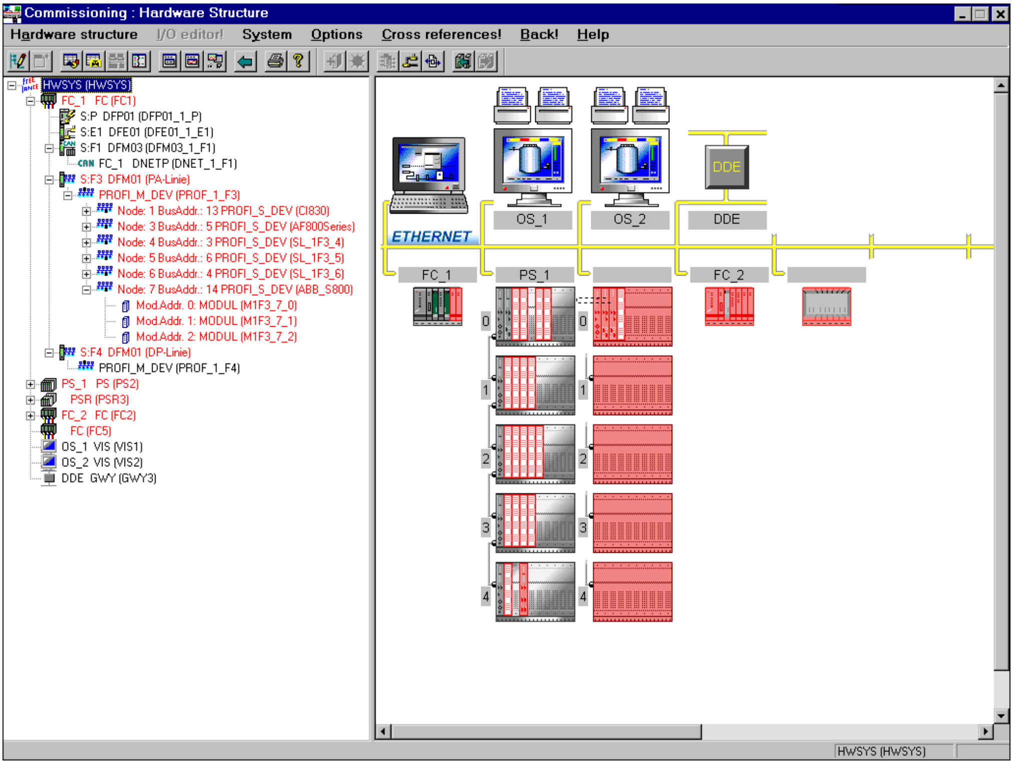
On Program list is switched on

Off Program list is switched off

6.5 Status displays in the hardware structure

After loading the resource in the station, the system or station knows which modules or field devices are configured in the station. The components (modules, devices) themselves may not have been installed yet, or the position of the installed component may not match the positions (slots) configured., Such errors are signaled by colored state information.

System → Hardware structure



Commissioning : Hardware Structure

```
HWSYS (HWSYS)
  FC_1  FC (FC1)
    S:P  DFP01 (DFP01_1_P)
    S:E1  DFE01 (DFE01_1_E1)
    S:F1  DFM03 (DFM03_1_F1)
      CAN FC_1  DNETP (DNET_1_F1)
    S:F3  DFM01 (PA-Linie)
      PROFI_M_DEV (PROF_1_F3)
        Node: 1 BusAddr.: 13 PROFI_S_DEV (CI830)
        Node: 3 BusAddr.: 5 PROFI_S_DEV (AF800Series)
        Node: 4 BusAddr.: 3 PROFI_S_DEV (SL_1F3_4)
        Node: 5 BusAddr.: 6 PROFI_S_DEV (SL_1F3_5)
        Node: 6 BusAddr.: 4 PROFI_S_DEV (SL_1F3_6)
        Node: 7 BusAddr.: 14 PROFI_S_DEV (ABB_S800)
          Mod.Addr. 0: MODUL (M1F3_7_0)
          Mod.Addr. 1: MODUL (M1F3_7_1)
          Mod.Addr. 2: MODUL (M1F3_7_2)
    S:F4  DFM01 (DP-Linie)
      PROFI_M_DEV (PROF_1_F4)
  PS_1  PS (PS2)
    PSR (PSR3)
  FC_2  FC (FC2)
    FC (FC5)
  OS_1  VIS (VIS1)
  OS_2  VIS (VIS2)
  DDE  GWY (GWY3)
```

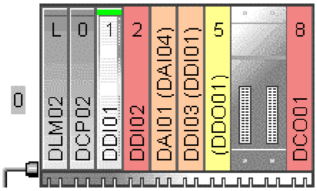
di1540uk.bmp

6.5.1 State in the tree view

In the tree view, state information is shown thus: active (black), faulty (red) and inactive (gray). The color coding is applied to the object information (text) rather than the icon.

6.5.2 Module state in the system and station views

The modules in the rack or the FieldController modules are displayed in various different colors according to their current state.



di1576.bmp

Color and state

transparent
(Rear view) There is no module configured in this slot, neither is any module installed in the station.

gray The module is configured and the correct module type installed.

red A module is configured in this slot, but not present in the station.

yellow A module (e.g. type DDO01) has been detected as installed in this slot, but no configuration is available.

orange The module type configured in the hardware structure (e.g. DAI01) does not match the installed module type (e.g. DAI04).

Force state

green A green rectangle is displayed towards the top of the module (only for I/O modules). At least one channel on the module has been forced.

6.5.3 State of the racks

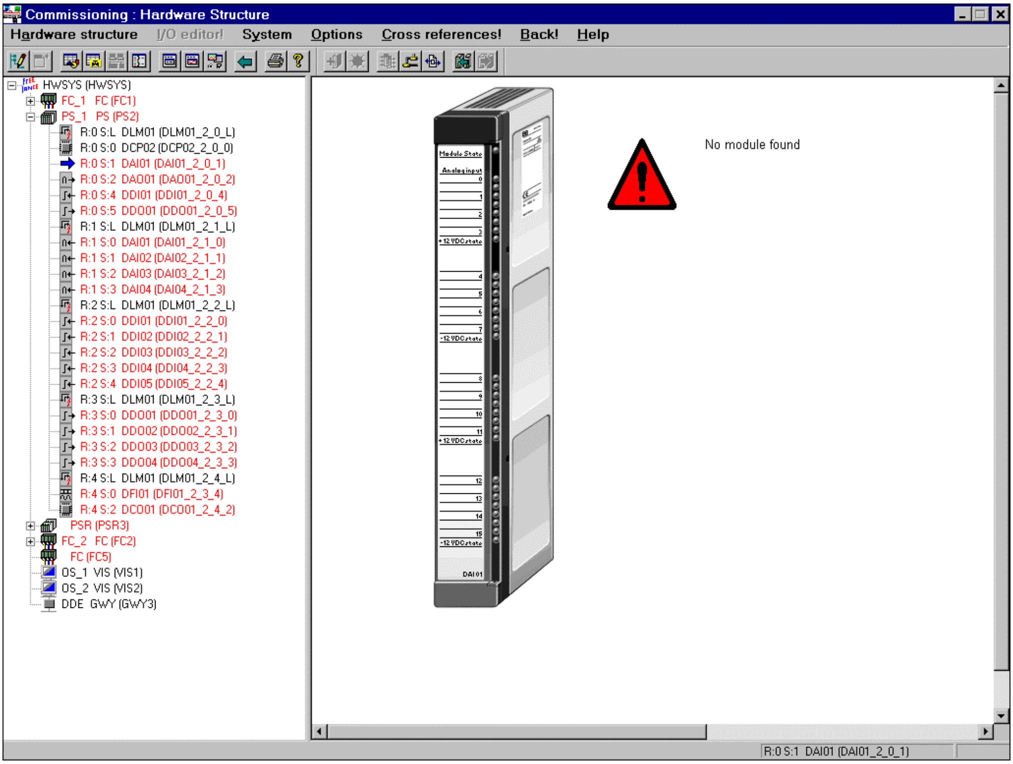
When they are added to the hardware structure, for a process station the max. number of racks is shown. The rack indicates whether the unit is active, inactive or faulty. Where a fault is indicated, this is generally due to the unit not being detected at all, i.e. not being installed or being impossible to address over the system bus.

dark gray	The rack is activated.
light gray	The rack is not activated
red	The rack cannot be addressed

6.5.4 State of the modules

The d**etail view** provides some detailed information about the I/O modules of the process station, thus enabling the user to easily identify the modules.

The state of modules is indicated by a colored triangle.



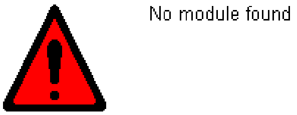
HWSYS (HWSYS)
FC_1 FC (FC1)
PS_1 PS (PS2)
R:0 S:L DLM01 (DLM01_2_0_L)
R:0 S:0 DCP02 (DCP02_2_0_0)
R:0 S:1 DAI01 (DAI01_2_0_1)
R:0 S:2 DAO01 (DAO01_2_0_2)
R:0 S:4 DDI01 (DDI01_2_0_4)
R:0 S:5 DDO01 (DDO01_2_0_5)
R:1 S:L DLM01 (DLM01_2_1_L)
R:1 S:0 DAI01 (DAI01_2_1_0)
R:1 S:1 DAI02 (DAI02_2_1_1)
R:1 S:2 DAI03 (DAI03_2_1_2)
R:1 S:3 DAI04 (DAI04_2_1_3)
R:2 S:L DLM01 (DLM01_2_2_L)
R:2 S:0 DDI01 (DDI01_2_2_0)
R:2 S:1 DDI02 (DDI02_2_2_1)
R:2 S:2 DDI03 (DDI03_2_2_2)
R:2 S:3 DDI04 (DDI04_2_2_3)
R:2 S:4 DDI05 (DDI05_2_2_4)
R:3 S:L DLM01 (DLM01_2_3_L)
R:3 S:0 DDO01 (DDO01_2_3_0)
R:3 S:1 DDO02 (DDO02_2_3_1)
R:3 S:2 DDO03 (DDO03_2_3_2)
R:3 S:3 DDO04 (DDO04_2_3_3)
R:4 S:L DLM01 (DLM01_2_4_L)
R:4 S:0 DFI01 (DFI01_2_3_4)
R:4 S:2 DCO01 (DCO01_2_4_2)
PSR (PSR3)
FC_2 FC (FC2)
FC (FC5)
OS_1 VIS (VIS1)
OS_2 VIS (VIS2)
DDE GWY (GWY3)

No module found

R:0 S:1 DAI01 (DAI01_2_0_1)

ra045gr.bmp

No module found (red warning sign)

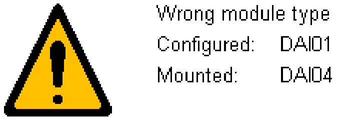


No module found

ra043gr.bmp

A module is configured in this slot, but not present in the station.

Wrong module type (orange warning sign)

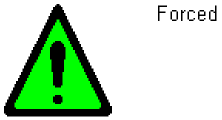


Wrong module type
Configured: DAI01
Mounted: DAI04

ra042gr.bmp

The module type configured in the hardware structure does not match the installed module type (e.g. DAI04).

Forced channels (green warning sign)



Forced

ra044us.bmp

At least one channel of the module has been forced. See **Engineering Manual**, **System Configuration**, **Commissioning, Forcing inputs and outputs.**

7 Redundancy

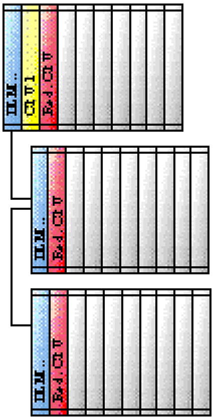
7.1 General description of CPU redundancy

It is possible to increase the availability of the process station D-PS by means of CPU redundancy.

☞ A redundant supply voltage for a sub-rack can be provided by the link module DLM 02. A bus redundancy (Ethernet redundancy) can be implemented between two star couplers.

The CPU redundancy is implemented in such a way that the user of DigiTool and of DigiVis sees only one process station. The fact that a redundant process station has two CPU modules is only noticeable initially in the configuration and later in the status evaluation within commissioning or operation. A redundant process station otherwise behaves like a non-redundant process station with only one CPU module (single-machine model).

7.1.1 Hardware structure of a redundant process station



di7173us.bmp

In a redundant process station, two CPU modules type DCP 10 must be plugged in and connected correctly via the redundancy link. The redundancy link (Coax 2/AUI 2) is an Ethernet connection between the two CPU modules and is used for the synchronization of the redundant CPU modules.

The possible slots of the CPU modules for the sub-rack DRA 02 are slots 0 and 1. If the CPU modules are installed in different sub-racks, then slot 0 should be used in each case.
The sub-rack DRA 01, as of hardware index 4, is corresponding to DRA 02 and is also usable for redundancy.

☞ A CPU module must always be plugged into slot 0 of a central processing unit.
See also **Mounting and Installation Instructions**.
If power supply redundancy is also desired, the link module DLM 01 must be exchanged for the DLM 02. The DLM 02 enables two DPW 01/02 power supplies to be connected. A system message is sent if a power supply fails.

☞ A possible power fail can be evaluated with the help of a system variable and used in the applications program.

7.1.2 Important restrictions with an upgrade

- For an upgrade with redundancy mode, it is necessary to check the I/O EPROMs of all I/O modules of this process station.
- A not redundant operation with V2 I/O modules is possible for projects which are to be operated without redundancy, so that it is not necessary to change the I/O EPROMs for such non-redundant applications. V2 I/O modules (DEA3.56, D2A1.04, DPT2.04, DIO1.06, DLM1.03 or less) which are plugged into a redundant system will generate a system message (Slot empty). The status LED of this modules will be flashing orange.

⚠ I/O modules with I/O EPROMs up to Version 3 as well as the new CPU module DCP 10 must not be operated in process stations with the software status of the V2!

- All variables of a redundant resource must be written in process image mode. If the variable cannot be configured this way, the content of these variables is not redundancy-capable, i.e. the variable value is not current in a redundancy toggle.
- Programs and function blocks of a non-redundant task will perform a cold start after a redundancy toggle.
- For projects and recipe programs (DigiBatch), the sequences appertaining to a PLI modules (sequential programs) together with the PLI module in the redundant task must be configured. It is necessary to position the tasks with the PLI modules in the project tree above the SFC programs.
- No structural changes of the configuration in routine operation can be performed in the commissioning without the associated resource being stopped.
- It is necessary to initialize the resource if boot parameters such as number of objects, PRAM, I/O protocol etc. are changed in the header of the resource.

7.1.3 Hardware and software requirements for CPU redundancy

- Two CPU modules type DCP 10 per redundant process station.
- All I/O EPROMs (firmware) of the I/O modules of the redundant process station must be redundancy-capable.
- The PC hardware and software is installed with Windows NT 4.0.
- DigiTool and DigiVis status Freelance 2000, Version 3.2 or above.
- The PC is at least a Pentium with 133 MHz, with 32 Mbytes RAM and
- 1 Gbyte hard disk for an operator panel and 2 Gbyte for an engineering station.
- SCSI hard disk and CD-ROM are recommended.

7.1.4 Synchronization

The active CPU module (Primary CPU) automatically ensures that the passive CPU module (Secondary CPU) can take over in the event of redundancy. The adjustment of operating system, user programs and redundancy data necessary for this is performed via a redundancy connection, the redundancy link. The connection is produced at the front of the CPU module and is implemented over an Ethernet cable, optionally 10BASE2 (Coax 2) or AUI cable (AUI 2).

One of the two CPU modules automatically becomes Primary CPU with bootstrap of a process station. The second CPU module establishes that it detects a Primary CPU on the station bus and logs on at the Primary CPU as Secondary CPU.

☞ A newly plugged-in CPU module recognizes automatically that another module is already active on the stations bus and moves automatically into the passive status. Exception: if the second CPU module (which is intended to operate as redundant CPU module) is loaded with the operating system of a Gateway CPU.

In the adjustment between the active CPU module and a passive CPU module, a connection is established via the redundancy link. The operating system and the user program are first transmitted from the Primary CPU to the Secondary CPU.

The Secondary CPU performs a cold start with these data and then logs on at the Primary CPU. The current process data are then transmitted to the Secondary CPU. After successful transmission, both CPU modules move to “sync“ status. From now on, the Secondary CPU is ready for a bumpless redundancy toggle. The relevant process data are balanced cyclically.

The synchronization runs in parallel with the processing of the applications programs in the process station.

7.1.5 Redundancy toggle

With redundancy, a bumpless toggle is achieved between the active CPU module (Primary CPU) and the passive CPU module (Secondary CPU), so that:

- the outputs of the I/O modules are retained under CPU failure,
- the internal statuses (module data, variable values) are retained,
- operating system, configuration and working data are balanced automatically between the active and the passive CPU modules.

If any of the following occur at the Primary CPU, a redundancy toggle is initiated **deliberately** between Primary CPU and Secondary CPU:

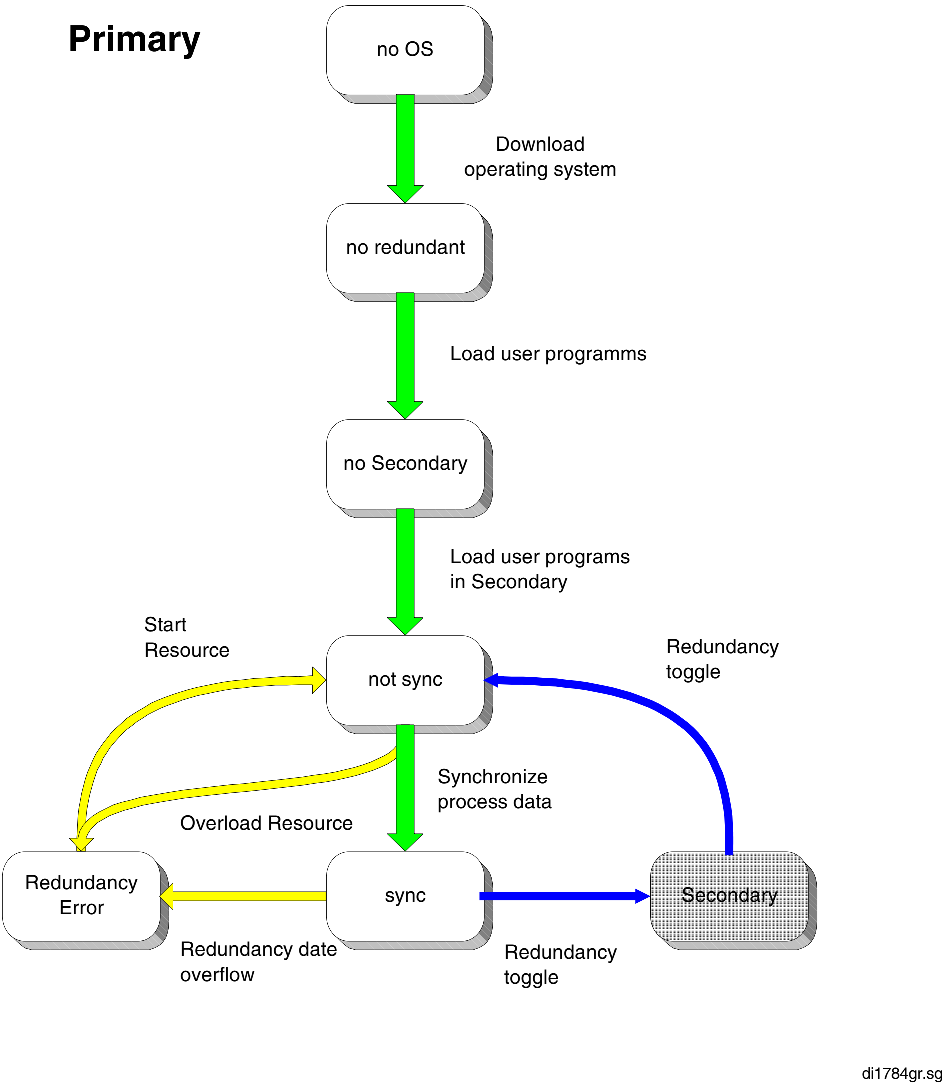
- the Prim/Sec toggle switch of the CPU module is actuated,
- the Prim/Sec button of the resource is actuated in commissioning mode,
- the reset switch is actuated > 5 s, resulting in a cold start of the Primary CPU,
- the system function "Force redundancy toggle" is executed.

If the following have been established by the self-test diagnosis, a redundancy toggle between Primary CPU and Secondary CPU **is forced:**

- the watchdog has responded: a hardware fault has occurred on the CPU module,
- a power fail of the power supply at the CPU module is reported,
- a network fault (Ethernet) has occurred,
- a “fatal software error” has occurred.

7.1.6 Redundancy statuses

7.1.6.1 Status diagram of the active CPU module (Primary)

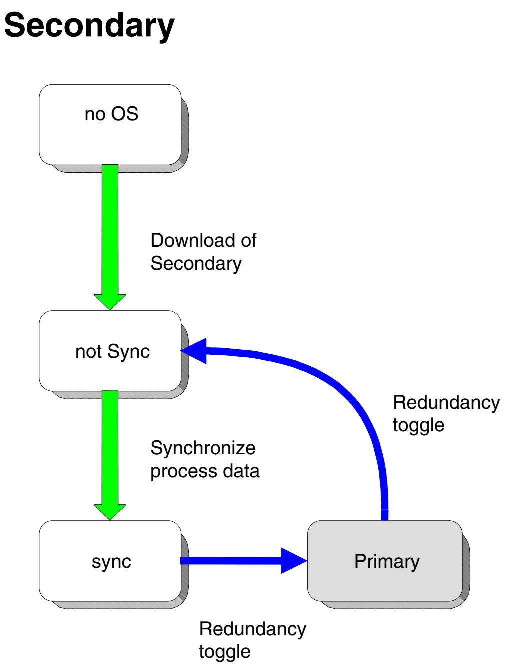


7.1.6.2 Statuses of the redundant resource

The following redundancy Status are reported for the redundant resource and displayed in the header of the resource (project tree).

Status of Primary CPU	Description
no operating system	No operating system loaded.
not redundant	The process station has been configured as redundant resource. The operating system is loaded but the user programs is not yet completely loaded.
no secondary	Operating system and user programs are completely loaded. The Secondary CPU has not yet logged on at the Primary CPU.
not sync	The Secondary CPU is present, the operating system and the user programs have been loaded but the process statuses have not yet been adjusted and the synchronization not yet therefore concluded.
sync	Normal status of the redundant process station. The Secondary CPU is synchronized and ready for a redundancy toggle. The cyclic data adjustment between the two CPU modules runs during the program processing.
redundancy error	• Redundancy not possible. The CPU module cannot achieve the status “sync”. By restopping and then restarting the resource, it is again attempted to reach the status “sync”. The cause of this may be that it is not possible to complete the loading of the user programs successfully. • The system loading is too high, the redundancy link cannot be maintained. The task cycle times have been configured too short. • The volume of redundancy data to be transmitted for a task is too large. The system variable xxxx.RedBufLow indicates the remaining memory location for the redundancy data with 0. The user programs has been distributed to too many tasks.

7.1.6.3 Status diagram of the passive CPU module (Secondary CPU)



di1783us.sg

Status of Secondary CPU	Description
no operating system	No operating system loaded. The Secondary CPU is waiting for the transmission of the operating system and the user programs.
not sync	The Secondary CPU has logged on at the Primary CPU: the operating system and the user programs are loaded but the process statuses have not yet been adjusted and the synchronization not yet therefore concluded.
sync	The Secondary CPU is synchronized and ready for a redundancy toggle. The cyclic data adjustment between the two CPU modules runs during the program processing.

☞ The redundancy Status of a resource can be evaluated using the system variables xxxx.RedState.

See also **Engineering Manual**, **System Configuration**, **Commissioning**, **Status displays of the D-PS resource** and **Engineering Manual**, **IEC 61131-3 Programming**, **Variables, System Variables**.

7.2 Configuration

To configure a redundancy, the redundant resource with the tasks must be configured in the project tree and the CPU modules in the hardware structure.

- There is a new resource type for redundant process stations in the project tree. As well as the redundant process stations, it is also possible to configure non-redundant process stations in a system. Both redundant and non-redundant tasks can be configured within the redundant resource.
- In the hardware structure, two CPU modules are mounted in the process station, and 2 IP addresses for a process station are allocated in the network configuration.

7.2.1 Redundant resources and tasks in the project tree

The following new project objects can be configured in the project tree:

- Redundant process station
- Redundant task (cyclic)
- Redundant default task (PLC mode)

A redundant task has so-called redundancy data. These are data of the process image and of the function blocks for which a data adjustment is performed after each cycle. This ensures that a redundancy toggle can take place at any point in time.
So that the data of a redundant task are redundancy-capable, all variables of a redundant task must be written in process image mode.

Not only the redundant task (TASK/RED), but also tasks of the previous type (TASK) can be configured under the redundant resources. Thus only those functions of the user program which also have actually to be redundant need to be configured redundantly within a redundant resource.

A new system task can be configured under the redundant resource, a redundancy start task. The **RedSt-Task** is activated at a redundancy toggle, and implemented once irrespective of the cause the toggle before the user tasks are calculated. Applications programs can be configured here specially for the redundancy toggle.

☞ It is possible by using a block import of part projects to make a resource redundant automatically by means of the new menu *Edit → Import block as redundant.* This means that the project objects are created and all variables (only with appropriate option) are written in process image mode.

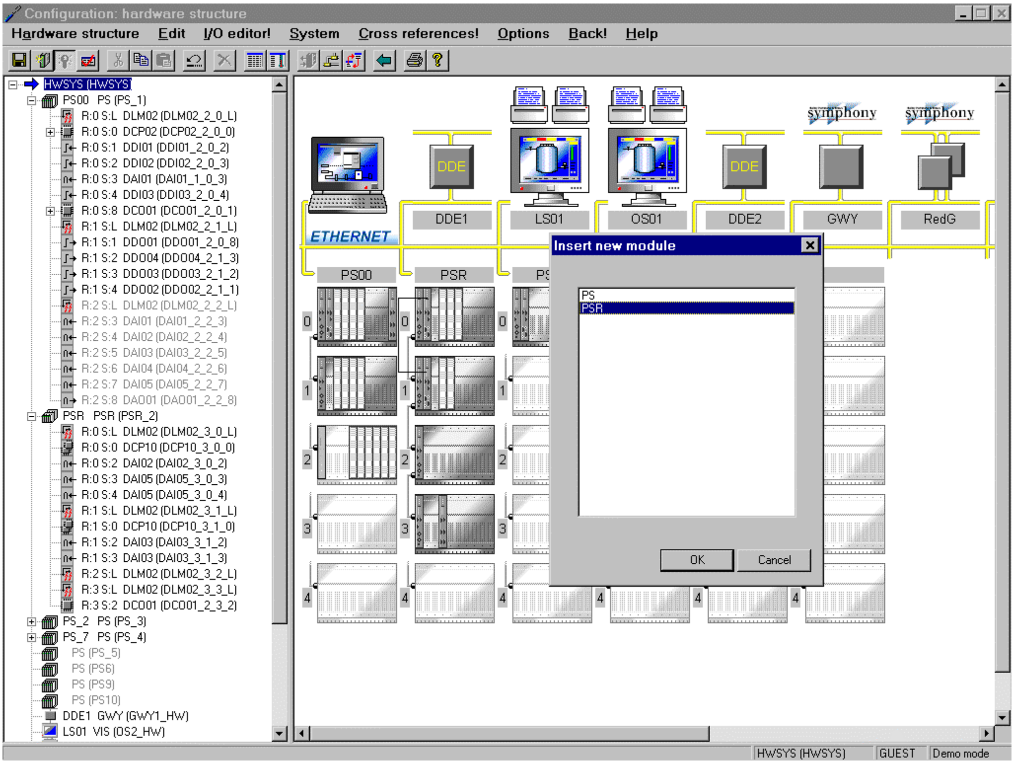
Exporting the individual process stations makes it possible in the reimport to accept the modules mounting and the I/O channel assignments, which would otherwise be lost.

7.2.2 Mount and activate redundant hardware

A redundant process station type must be configured in the hardware structure.

The following steps are necessary for this:

- Select the system node (HWSYS) in the tree view or a process station position in the graphic view
- Select *Insert*
- Select the redundant process station type PSR
- Allocate a redundant resource (typed PS/RED in the project tree) with *Resource allocation*
- Mount both CPU modules



```
HWSYS (HWSYS)
   PS00  PS (PS_1)
      R:0 S:L  DLM02 (DLM02_2_0_L)
      R:0 S:0  DCP02 (DCP02_2_0_0)
      R:0 S:1  DDI01 (DDI01_2_0_2)
      R:0 S:2  DDI02 (DDI02_2_0_3)
      R:0 S:3  DAI01 (DAI01_1_0_3)
      R:0 S:4  DDI03 (DDI03_2_0_4)
      R:0 S:8  DCO01 (DCO01_2_0_1)
      R:1 S:L  DLM02 (DLM02_2_1_L)
      R:1 S:1  DDO01 (DDO01_2_0_8)
      R:1 S:2  DDO04 (DDO04_2_1_3)
      R:1 S:3  DDO03 (DDO03_2_1_2)
      R:1 S:4  DDO02 (DDO02_2_1_1)
      R:2 S:L  DLM02 (DLM02_2_2_L)
      R:2 S:3  DAI01 (DAI01_2_2_3)
      R:2 S:4  DAI02 (DAI02_2_2_4)
      R:2 S:5  DAI03 (DAI03_2_2_5)
      R:2 S:6  DAI04 (DAI04_2_2_6)
      R:2 S:7  DAI05 (DAI05_2_2_7)
      R:2 S:8  DAO01 (DAO01_2_2_8)
   PSR  PSR (PSR_2)
      R:0 S:L  DLM02 (DLM02_3_0_L)
      R:0 S:0  DCP10 (DCP10_3_0_0)
      R:0 S:2  DAI02 (DAI02_3_0_2)
      R:0 S:3  DAI05 (DAI05_3_0_3)
      R:0 S:4  DAI05 (DAI05_3_0_4)
      R:1 S:L  DLM02 (DLM02_3_1_L)
      R:1 S:0  DCP10 (DCP10_3_1_0)
      R:1 S:2  DAI03 (DAI03_3_1_2)
      R:1 S:3  DAI03 (DAI03_3_1_3)
      R:2 S:L  DLM02 (DLM02_3_2_L)
      R:3 S:L  DLM02 (DLM02_3_3_L)
      R:3 S:2  DCO01 (DCO01_2_3_2)
   PS_2  PS (PS_3)
   PS_7  PS (PS_4)
      PS (PS_5)
      PS (PS6)
      PS (PS9)
      PS (PS10)
   DDE1  GWY (GWY1_HW)
   LS01  VIS (OS2_HW)
```

di7152us.bmp

A redundant process station always features 2 CPU modules. Redundancy link is automatic when inserting a second CPU module.

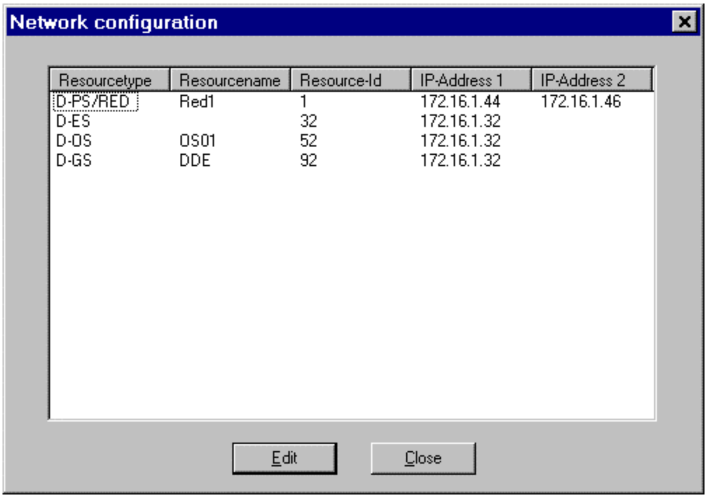
The following slot combinations are possible for a redundant process station:

CPU module 1	CPU module 2	Rack DRA 01	Rack DRA 02
Rack 0/Slot **0**	Rack 0/Slot **1**	not possible	OK
Rack 0/Slot **0**	Rack 1/Slot **0**	OK	OK
Rack 0/Slot **0**	Rack 2/ Slot **0**	OK	OK
Rack 0/Slot **0**	Rack 3/ Slot **0**	OK	OK
Rack 0/Slot **0**	Rack 4/ Slot **0**	OK	OK

7.2.3 Network configuration

A redundant process station has a resource-ID (previously station number) and a dedicated IP address for each CPU module.

☞ Which of the CPU modules is Primary and which is Secondary is **not** defined by IP address 1 and IP address 2.



Resourcetype	Resourcename	Resource-Id	IP-Address 1	IP-Address 2
D-PS/RED	Red1	1	172.16.1.44	172.16.1.46
D-ES		32	172.16.1.32	
D-OS	OS01	52	172.16.1.32	
D-GS	DDE	92	172.16.1.32	

di7154us.bmp

7.3 Technical performance

- The link monitoring between the CPU module (Primary) and the I/O modules is performed at 250 ms / 500 KBit/s (1250 ms / 100 KBit/s)
- The CPU modules (Primary and Secondary) are monitoring each other. The Secondary-CPU detects a failure of the Primary (redundancy toggle) within 20 ms / 500 KBit/s (40 ms / 100 KBit/s). The Primary detects a failure of the Secondary within 250 ms / 500 KBit/s (1250 ms / 100 KBit/s).
- An automatic redundancy toggle is performed within less than 5 seconds if the active CPU module (Primary) fails.
- The adjustment of the operating system and configuration between the Primary and Secondary CPU are performed in less than 30 seconds after loading the process station. This is valid for loading the configuration (load whole station, changed objects, ...) and for plugging a new CPU module.

7.4 Changing a not redundant process station into a redundant one

- Export from every process station which should be redundant
  a) the block of the individual resources in the project tree **and**
  b) the block of all modules of the process stations in the hardware structure.

☞ You can automatically make a resource redundant by performing a block import of partial projects, using the menu items *Edit → Import block as redundant.* The project elements are then created, and all variables are written using the process image. If the process stations are exported, their modules and the I/O channel assignment can be taken over when they are re-imported and will not be lost.

- The variables of the task to be implemented redundantly must be described in process image mode, otherwise they will not be redundancy-capable. Please check your user programs for variables which are not described in the process image (column P in the variables list). If you consider that these variables cannot be written in process image mode because the information within the task must be immediately available and must not only be updated at the end of a task cycle, then you should renew the resource assignment of the variables or move the program list and sequential function charts individually (see also next item).
- The resources intended to operate as redundant resources must be deleted. The redundant resources will otherwise be renamed automatically when imported into the project pool, because objects with the same names will certainly still be present in the project tree. Alternatively insert the redundant task under newly-inserted redundant resources and move the program list and sequential programs individually. Please do not forget the content of the system task. This method must be chosen where it is not possible to describe all variables in process image mode or where it is not necessary to operate all tasks redundantly
- In the project tree and using *Edit → Import block as redundant,* insert the resources which were previously exported as redundant resources into the project pool

- Renew the resources assignment for the variables. This can be done manually or automatically
- The global system variables of the previously non-redundant resources which were used in programs or graphics still have the former resource allocation as an integral part and must be changed possibly to a new resource allocation.

☞ These now undefined identifiers are reported as errors in the plausibility check.

- In the hardware structure then import all individually exported blocks of the process stations into a redundant process station so that all information about the equipment and the I/O modules is there once again.

☞ When changing a rack-based process station PS into a redundant FieldController AC 800FR, always import the I/O modules individually as a block under the CAN master DNETP.

- Allocate the redundant resource to the redundant process station
- Plausibility check the project.

Index

ABB

Gross Automation, 1725 South Johnson Road, New Berlin, WI 53146, www.ssacsales.com, 800-349-5827