RAILROAD & CO.®

TrainController™ +Net™

Multi User/Multi Computer
Model Railroad Control

Manual

Version September 2005

Contact: Freiwald Software
Kreuzberg 16 B
D-85658 Egmating, Germany
e-mail: contact@freiwald.com
http://www.freiwald.com

Table of Contents

Overview

RAILROAD & CO. is the leading product line of computer programs for digitally or conventionally controlled model railroads. It contains the following members:

- **TrainController™** is the world's leading software for computer controlled model railroads.
- **TrainProgrammer™** is the program, which makes programming of DCC decoders as simple as a few clicks with your mouse.
- **+Net™** is a module, that allows to control your layout with a network of several computers running **TrainController™**.
- **+4DSound™** is a module, that recreates realistic spatial sound effects for each model railroad layout controlled by **TrainController™** without the need to install on-board sound into each decoder.
- **TrainMonitor™** is the world's first program, which is especially made for indication of train positions on the computer screen based on train detection and train tracking.
- **RAILROAD & CO. Handheld** is the world's first remote control especially designed for computer controlled model railroads.

By reading this manual one can obtain information about how to control a model railroad layout with more than one computer running **TrainController™**.

An overview of the basic concepts of **TrainController™** is provided in the **TrainController Users Guide**. It is assumed, that you are familiar with the contents of that document.

In the following it is also assumed, that you know how to setup and to operate a simple TCP/IP network between your computers. If you are not familiar with TCP/IP based networking, it is strongly recommended to study concerning literature first.

1 Introduction

1.1 Overview

TrainController™ is a system to operate a model railroad layout from a Personal Computer running MS Windows 98 or 95, Windows ME, Windows XP, Windows 2000 or Windows NT.

TrainController™ provides you with the ease of point and click to operate your switches, signals, routes and other accessories displayed on track diagram panels. Track diagram panels are individually created for each yard or section, as desired. You can run your trains with on-screen throttles, external hand held throttles connected to your computer, or with your favourite throttles or hand held throttles supported by your digital system. You can operate digital engines equipped with their own decoders, as well as conventional models without decoders. Digital and conventional engines can run on the same track. Far-reaching automation features make railroad operations manageable by one person and match those found on the largest club layouts. You can see on the screen which engine/train is on which track.

1.2 Multi User / Multi Computer operation with + Net™

+Net™ is an additional component, that can be added to **TrainController™** to enable you to operate a model railroad layout with several computers, that are connected with each other by a network. **+Net™** utilizes standard network technologies and protocols (TCP/IP), that are available on every modern personal computer running Windows.

+Net™ also provides a mode which allows testing of distributed functions by starting **TrainController™+Net™** several times on one single computer without any physical network.

Unlike other network based solutions for model railroad computer control, that support only sharing of the same digital system between different computers or focus on enabling a remote computer to access a remotely connected digital system, **+Net™** focuses on sharing and on distribution of the high level model railroad control processes to different computers.

The architecture of **+Net™** is peer-to-peer rather than client/server. That means, that control of the model railroad is shared between different computers with equal rights. In general there is no dedicated “server computer” nor a centralized interface to the model railroad.

Assume the following sample layout:

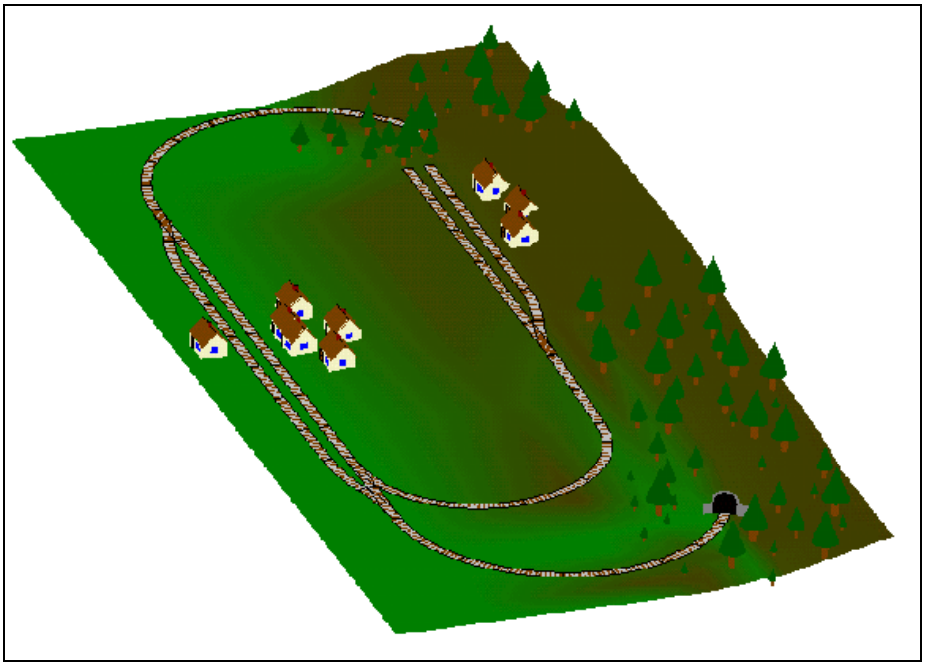

Diagram 1: Sample Layout

The layout has two stations: “Southtown” located on the left side of the layout and “Northville” located at the end of the branch line. There is an additional hidden yard that is covered by the mountain.

This can be seen better in the track plan displayed below:

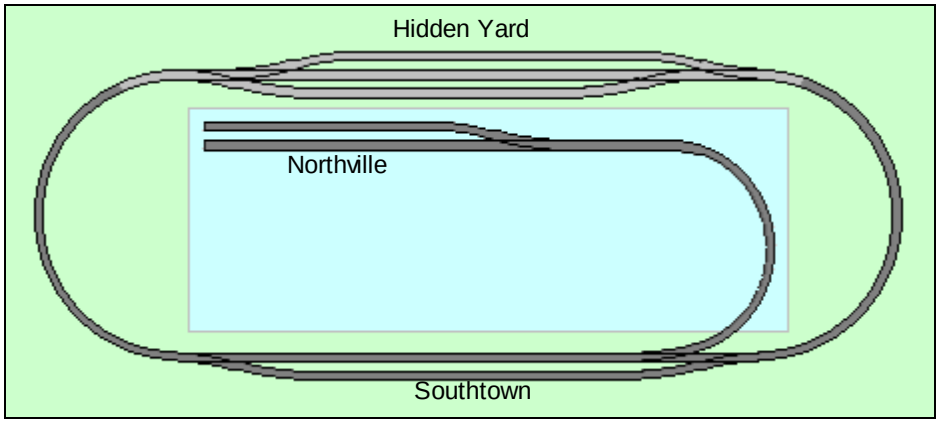

Diagram 2: Track Plan of the Sample Layout

The control of this layout shall be distributed to two computers. The main line, i.e. the loop that connects “Hidden Yard” and “Southtown” (green area), will be operated by the first computer (called “Green” throughout this document). The branch line from “Southtown” to “Northville” (blue area) will be operated by the second computer (called “Blue” here).

It is certainly not necessary to use two computers to control such small layout. For demonstration purposes, however, such small layout is well suited.

1.3 Connecting multiple computers to the model railroad

In general and in the most recommended case, the computers are connected to the layout through one or more digital systems and by a separate computer interface for each computer as shown in the following image:

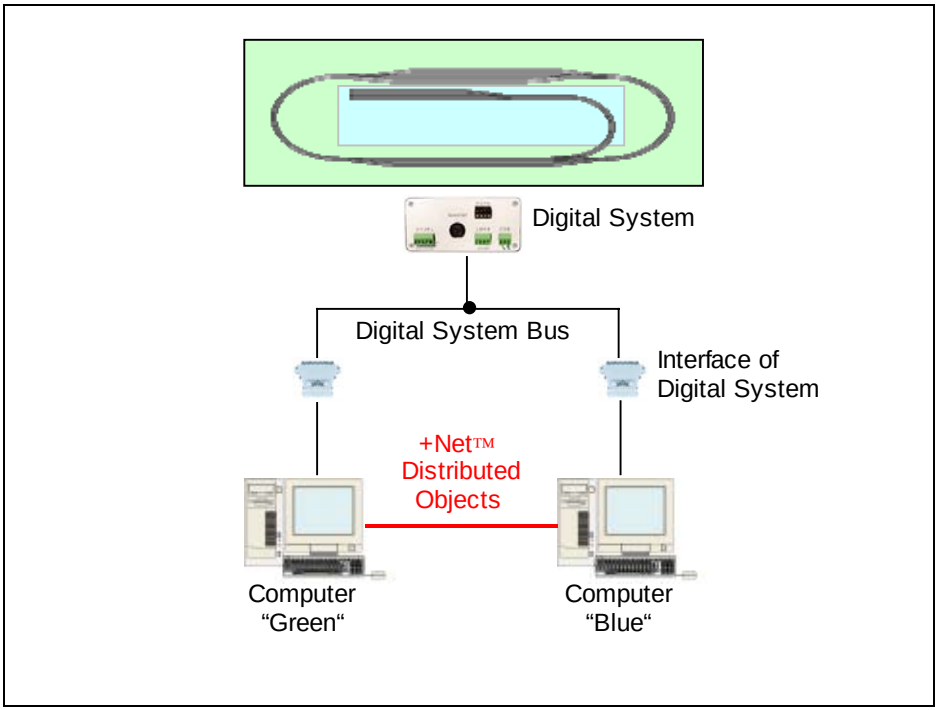

Diagram 3: Utilizing the Digital System Bus

The red line in the above image represents the core feature of **+Net™**, i.e. the possibility to distribute objects or to make objects “visible” across the network. This feature is based on TCP/IP connections between the computers of the network.

In the above image the digital system is shared between different computers by utilizing the possibility, to connect several computers directly to the digital system through the Digital System Bus.

- Examples of Digital System Busses are Lenz XpressNet, Digitrax LocoNet or Trix Selectrix SX.
- Examples of Interfaces are Lenz LI101F, Digitrax MS100, an Interface to the SX bus or any other interface offered for the related digital system.

Users of other digital systems can share their digital system between different computers, too, by installing **+Net/D™**, which is an extended variant of **+Net ™**. This is outlined in detail in chapter 3.

In an extended configuration it is also possible, to connect “local” digital systems to one computer, that are not visible to the other computers in the network. It is for example possible to connect a second digital system to computer “Green”, that is responsible for turnout control or feedback monitoring in the Hidden Yard. This is shown in the image below:

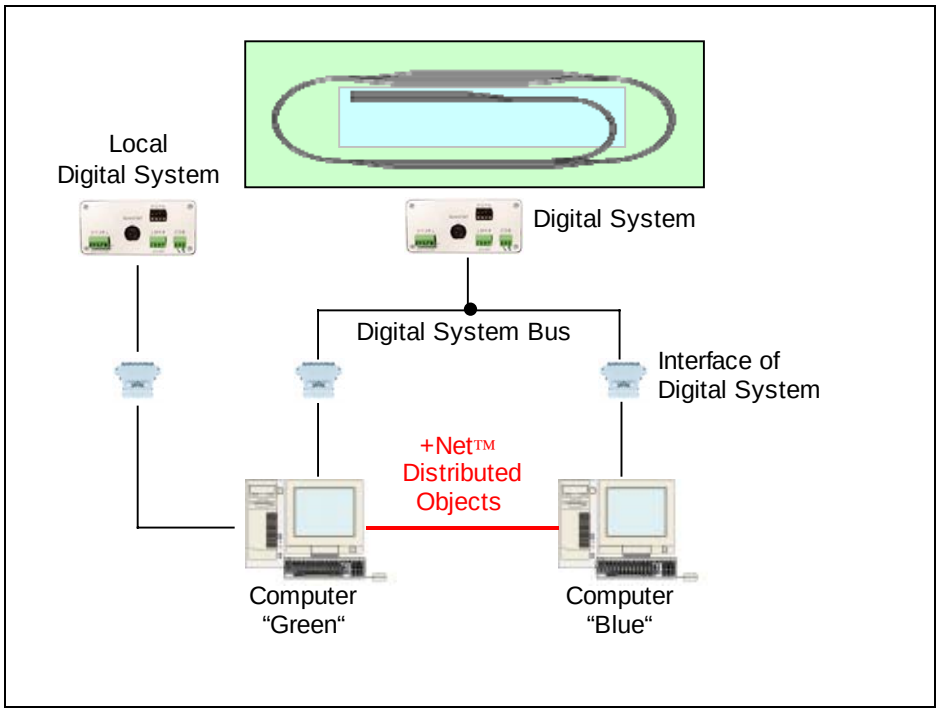

Diagram 4: Connecting a Local Digital System

In the above image the local digital system is only operated by computer “Green”. The digital system is not visible to computer “Blue”.

The above examples show configurations with only two computers. With **+Net™** it is possible, however, to create a network with as many computers as desired.

1.4 Distributed Model Railroad Computer Control

The key features of **+Net™** are listed below:

- Distributed control of manual switchboard control panels for separate parts of the layout. It is for example possible to run a switchboard control panel for “South-town” on computer “Green” and a switchboard control panel for “Northville” on computer “Blue”.
- Distributed start/destination keys
- Distributed interlocking and remote blocking.
- Transferring control of manually or automatically running trains from one computer to another.
- Network wide train tracking.
- Distributed execution of semi-automatic control mechanisms.
- Remote execution of manual and/or automatic schedules.
- Remote linking of schedules on different computers in the same successor chain.

Even though **+Net™** offer so many possibilities for operation of a model railroad layout with multiple computers you will find, that setup of **+Net™** in the software is amazingly simple. This is shown in the following.

2 Distributed Objects

The core mechanism for model railroad control with multiple computers running **+Net™** is the distribution of objects. Distribution of objects means to make objects stored on one computer "visible" to other computers. This is a powerful, though amazingly simple mechanism.

Distribution of objects is based on the following: in order to communicate with each other, each distributed object is associated with a counterpart on one or more other computers. A turnout, for example, can be distributed by creating a turnout symbol on two different computers and specifying the same address for both objects. Other objects, who do not have digital addresses (such as blocks or schedules) are connected with each other by a logical association, that is based on a common name, the *network name*.

Objects with digital addresses (turnouts, signals, feedbacks, push buttons with address, etc.) are associated with "sibling" objects on other computers by their address. They communicate through the digital system bus (or as an alternate through the Virtual Digital System Bus, if the digital system does not support such bus – see chapter 3). These objects can "see" their counterpart through the digital system bus (or the Virtual Digital System Bus).

Objects without digital addresses (routes, blocks, flagmen, push buttons without address, macros, schedules, etc.) cannot "see" each other through the digital system bus. These objects are associated with each other by a logical connection, which is shown in Diagram 3 as a red line. This logical connection is based on the network name. Two objects located on different computers are associated with each other by assigning the same network name to them.

Engines and trains, finally, are not actually distributed or associated with each other across the net. Instead the same engine or train data must be loaded on all computers, where the engines and trains shall be visible. In this way each engine and train "exists" only once in the network, but can be seen on all computers, if necessary.

This is outlined in more details in the following.

The following objects can be distributed (for details about these objects refer to the **TrainController™** Users Guide please):

- Turnouts and signals

- Feedback Indicators
- Push Buttons and On-Off Switches
- Flagman Indicators
- Engines and Trains
- Macros
- Routes
- Blocks
- Schedules
- The Clock

This is outlined in more detail in the following sections.

Turnouts and Signals

Turnouts and signals are implicitly distributed by means of their digital address and their assignment to a particular digital system. If two turnout or signal symbols defined with the same address on different computers are associated with the same digital system, then these symbols are automatically shared between these computers, if the concerning digital system is shared between these computers, too. Sharing of a digital system between different computers is possible by means of the hardware based Digital System Bus of this digital system or by a Virtual Digital System Bus as outlined in chapter 3.

To distribute a turnout or signal symbol between two or more computers connect these computers to the same digital system and assign the same address and this digital system to the concerning symbol on each computer. If the turnout or signal symbol is operated on one computer you will se the change of the associated symbol on all other computers connected in this way.

There is no limit with regard to the number of computers, to which a particular signal or turnout symbol can be distributed.

Please note, that all turnout and signal symbols, that are associated with each other, usually show the same status on all concerning computers. **Conditions**, if any, may prevent a certain turnout or signal symbol from being actively operated; though they do not prevent the symbol from following the status changes of sibling objects on remote computers passively.

Feedback Indicators

Feedback indicators are implicitly distributed by means of their digital address and their assignment to a particular digital system like turnout or signal symbols. The content of the previous section applies to feedback indicators accordingly.

If a feedback sensor, that is associated with several feedback indicators on several computers, is turned on, then the associated feedback symbols will be turned on on all concerning computers. Like distributed turnout and signals the digital system, to which the feedback sensor is connected, must be shared between the particular computers by means of the hardware based digital system bus of this digital system or by a Virtual Digital System Bus as outlined in chapter 3.

Please note, that **Memory** settings of contact indicators are only locally effective on the particular computer, where they are defined. Contact indicators, that are associated with each other across the network, may show a different status, if the **Memory** settings of these indicators are different on the particular computers, too.

Push Buttons and On-Off Switches

If a push button or on-off switch is associated with a digital address, then it can be distributed implicitly by means of its digital address and its assignment to a particular digital system like a turnout or signal symbol.

Push buttons and on-off switches without assignment to a digital address can be distributed as well. If **+Net™** is installed on your computer, then the **Push Button Properties** dialog and the **On-Off Switch Properties** dialog contain an additional tab labeled **Network**.

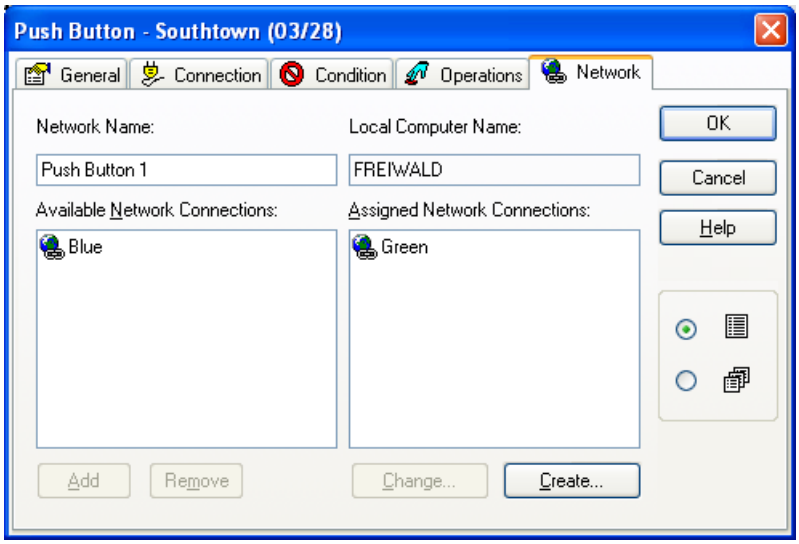

Diagram 5: Network Tab

This tab offers the following options:

Network Name:
Specify a unique name for the edited object, which with this object can be unambiguously identified on all computers, where this object is distributed on. If a push button, for example, is distributed to two computers, then the network names of both associated push button symbols must be identical on both computers. In a certain sense this network name acts like a digital address. This name may be an arbitrary sequence of at most 16 printable characters.

Local Computer Name:
The name of the current computer in the underlying TCP/IP network is displayed here. The name is displayed for informational purposes, e.g. to help you to link to this computer from other computers.

Available Network Connections:
This list contains the names of all computers, to which connections have been already established.

Assigned Network Connections:
This list contains the names of all computers, to which this symbol shall be additionally distributed. In the example above a push button symbol is created, that is distributed on the local computer “FREIWALD” and the remote computer “Green” by using the unique network name “Push Button 1”. This list may contain an arbitrary number of entries.

Add:
Use this option, to add the name of the computer, that is selected in the list of all computers, to the list of assigned computers.

Remove:
Use this option, to remove the name of the computer, that is selected in the list of assigned computers, from this list.

Change:
Use this option, to change the name of the computer, that is selected in the list of assigned computers. Please note, that this change affects all distributed elements on this computer with a network connection to the selected computer. If “Green” is changed to “Yellow”, then all network connections of all elements in the current session, that point to “Green” will be reassigned to “Yellow”. In this way you can easily change all links to a certain computer by links to another computer in one single step. This is also useful to easily swap between local testing and distributed operation. Please note also, that a changed name cannot by restored by pressing the **Cancel** button of the dialog nor by **Undo** of the **Edit** menu. This is no major restriction, though, because name changes back to the previous or any other name can be done very easily at any time by using this option.

For testing it is also possible to establish a network connection to the local computer. This is done by specifying a single dot “.” as computer name. If this is done, then networking can be tested between to instances of **TrainController™**, that run on the same computer.

Create:
Use this option, to add the name of a new computer to the list of all computers.

If you want to change the assignment of the displayed element from “Green” to “Yellow” without affecting other elements, do not use the **Change** option. Instead create a new entry “Yellow”, add this to the list and remove “Green”.

Notes:
To distribute an object on different computers, fill in the **Network** tab of this object on each particular computer. Use the same unique network name on all computers. Add the name of all “other” computers to the list of assigned computers on each particular computer.

To distribute a push button symbol on computer “Blue”, “Green” and “Yellow” create a push button symbol on each computer. Specify the same unique network name on all computers. On computer “Blue” add “Green” and “Yellow” to the list of assigned computers. Do the same on the other two computers accordingly.

If this is done correctly, then pressing of the push button symbol on one of these computers will cause pressing of the button on the other two computers, too. On each computer you can assign an individual set of operations. In this way it is not only possible to transfer state information from one computer to another, but also to execute operations on one computer by pressing a push button or on-off switch on another computer. A similar “remote effect” can also be achieved, if the push buttons or on-off switches are used as start/destination keys.

Push buttons and on-off switches are distributed symmetrically. This means: it does not matter, on which computer a distributed push button symbol is being pressed. The associated symbols on the other computers will always follow accordingly.

Flagman Symbols

Flagman symbols are distributed by means of the **Network** tab of their properties as shown above. If a distributed flagman symbol is triggered on one computer, then all associated flagman symbols on the other computers will follow accordingly. In this way it is possible to signal certain status changes from one computer to other computers. This can be used for distributed interlocking etc.

Flagman symbols are distributed asymmetrically. This means: only the flagman symbol on one computer may contain a trigger ("master flagman"). The trigger of all associated flagman symbols on other computers must be left empty ("slave flagman"). The associated slave flagman symbols on other computers are solely triggered through the network. If a slave flagman has a non-empty trigger, too, then it does not follow, when the associated master flagman is turned on on the other computer. In a set of associated distributed flagman symbols there must be exactly one master flagman. This master flagman is specified by associating a trigger with it.

Engines and Trains

The data of engines and trains is not distributed on the network. Instead identical data of all engines and trains, that shall be controlled by different computers, is loaded into the concerning data files on each computer.

This is supported by the **Import** and **Export** commands of the **Train** menu. At first create the necessary data for each engine or train with the usual means of **TrainController™**. This can be done on an arbitrary computer on the network. It is recommended, however, to perform the measurement of the speed profile of each engine on a computer, that is directly connected to the digital system used for train control, i.e. not indirectly through a Virtual Digital System Bus. Then export the data of all engines and trains, that shall be controlled by other computers, too, by using the **Export** command of the **Train** menu. Finally import this data on the other computers by using the **Import** command of the **Train** menu.

Please remind to repeat the export/import procedure for all affected engines or trains, whenever you changed their data. This procedure can be made much more convenient, if a network-widely shared disk drive is being used to store the train data files.

Macros

Macros are distributed by means of the **Network** tab like push button symbols as shown above. Execution of a distributed macro on one computer causes execution of the associated macros on the other computers and vice versa. In this way it is possible to execute operations on remote computers.

Macros are distributed symmetrically. This means: it does not matter, on which computer a distributed macro is executed. The associated macros on all other computers will be executed accordingly.

Routes

Routes are distributed by means of the **Network** tab like push button symbols as shown above. State changes of a distributed route on one computer cause highlighting of the associated routes on the other computers and vice versa.

Some restrictions apply, however:

- Routes can be distributed under the limitation, that there is no network wide interlocking for such routes across computers. If a route, that is distributed on computer A and computer B, is activated on computer A, then the associated route on computer B tries to activate, too. If this is not possible, then the route on computer B remains non-active. This does not affect the activated route on computer A.
- If a route is activated on computer A, then an existing counterpart of this route on computer B is highlighted, if possible, and the contained tracks and turnouts are locked, but not operated on computer B. It is assumed, that both routes contain the same turnouts. For this reason all turnouts contained in these routes are assumed to be operated, when the route on computer A is activated. The route on computer B does not operate these turnouts again.
- If it is desired to activate one of these routes only when its counterpart can be activated, too, then undesired activation of either route must be prohibited by using distributed on-off switches or flagmen, that are added to the conditions of the routes.
- This limited distribution of routes is mainly useful for display or monitoring purposes, where computer B is mainly used to display, what is happening on computer A.
- True distributed control, where two or more computers are responsible to actually control parts of the layout should be solved in a way, that each computer is responsible for its own district, and no routes exist, that are common to more than one computer.

Blocks

Blocks are distributed by means of the **Network** tab like push button symbols as shown above. State changes of a distributed block on one computer cause state changes of the associated blocks on the other computers and vice versa. This concerns also the reservation of blocks by engines and trains. In this way it is possible to perform remote blocking and network wide train tracking from one computer to others.

The following state changes are reported over the network

- Reservation by engines and trains (including engine orientation)
- Exit Locks
- Entry Locks

- Preference of Blocks
- State changes of block signals

Block occupancy is not reported explicitly over the network. This can be done by distribution of the associated indicator symbols as outlined above.

Distribution of blocks is important for distributed control of a layout, where two or more computers are responsible to actually control separate parts of the layout. A distributed block is created at all locations of the layout, where control of trains is passed from one computer to another. In this way distributed blocks act as kind of "interfaces" between different parts of the layout.

Blocks are distributed symmetrically. This means: it does not matter, on which computer a distributed block changes its state. The state of all associated blocks on all other computers will follow accordingly.

Schedules

If **+Net™** is installed on your computer, then an additional type of schedule is available, the so called *remote schedule*. Remote schedules can be used to start a schedule on one computer from another. Remote schedules are always associated with a usual schedule on another computer. By creating a remote schedule on computer "Green", for example, that is associated with a regular schedule on computer "Blue", it is possible to start the schedule on computer "Blue" remotely on computer "Green".

Remote schedules are created by calling the **Create Remote Schedule** command of the **Schedule** menu. Then the **Network** tab must be filled in in a similar way as for other objects. The distinctive feature here is the fact, that only one remote computer can be assigned, because the remote schedule can only be used to start one specific schedule on one other computer. On the remote computer, where the actual schedule to be started is hosted, do the same by assigning the computer name of the remote schedule.
If this is done, then the Schedule can be started remotely. If, for example, a schedule on the branch line to "Northville" is located on computer "Blue", then this schedule can be started by computer "Green" by creating a remote schedule on "Green", that is associated with the actual schedule by specifying a unique network name and a connection to "Blue". The actual schedule on computer "Blue" must be inversely associated with the remote schedule on computer "Green" by using the same network name and assigning a connection to "Green".

The remote schedule on computer "Green" can then be assigned to schedule selections or as successor to other schedules located on computer "Green". Especially with the lat-

ter it is possible to extend automatic, schedule based operation of all engines and trains over the network. If a certain schedule on the first computer is about to be terminated, and a remote schedule associated with a schedule on another computer is available as successor of this terminating schedule, then control of the train can be passed via the remote schedule to another schedule (or schedule selection) on the other computer. It is of course possible, to mix remote and usual schedules in the same selection or set of successors of a schedule selection or a normal schedule, respectively.

Some differences between usual schedules and remote schedules are to be taken in account, though. It is generally assumed, that it is always possible to start remote schedules (or more exactly: the actual schedule associated with the remote schedule on the other computer). In cases, where a remote schedule is contained in a list of successors, **TrainController™** will always treat remote schedules as "possible to be executed", when selecting an appropriate schedule. While usual schedules are investigated, whether a train is available or whether the blocks and routes ahead are available, this is always assumed to be true for remote schedules (because the software cannot perform such look ahead on a remote computer). For this reason the computer, where the actual schedule is to be started, should signal the permission, whether this schedule can be started or not, in advance (e.g. by mans of distributed flagman symbols, etc.). This permission should be included in the **condition** of the remote schedule on the other computer.

If, for example, the schedule on the branch line to "Northville" on computer "Blue" must not be started, if both blocks in "Northville" are occupied, then you should create a flagman on "Blue", which is turned on, when both blocks are occupied. Distribute this flagman to computer "Green" and enter the associated flagman symbol on computer "Green" to the condition of the remote schedule on that computer.

If remote schedules are being used as successors of schedules for passing of control of trains from one computer to another, then transfer of control should take place while the train is stopped. In other words: schedules should "intersect" in (distributed) blocks, where trains normally stop.

It is also possible to transfer control fluently, i.e. without stopping of trains. If a remote schedule, that is listed as successor, is not restricted by a condition, then **TrainController™** will start the remote schedule without stopping the train. The previously running schedule on the first computer is terminated and control of the train is released without stopping the train. **TrainController™** leaves it up to the second computer to take over control. This is done by starting the actual schedule on the second computer, that is associated with the remote schedule. Usually the train will continue without stopping. If the second computer determines, that the train must not leave the (distributed) block, where transfer of control took place, then the train is abruptly stopped regardless where

it is currently located. This is a safety measure for cases, when the train must not leave the start block on the second computer.

This fluent transfer of control of running trains requires appropriately configured software on both computers. In cases, where the second computer is not able to start the “receiving” schedule, e.g. due to a misconfiguration of its condition or rules, the train will continue running without control of a schedule. For this reason it is more safe to transfer control of a train between computers, while the train is stopped.

The Clock

The clock is distributed by defining one computer in the network to contain the master clock. From time to time the clocks on other computers in the network will be synchronized with the master clock. During synchronization the slave clocks are set to the same time as the master clock.

Master and slave clocks are defined as shown in Diagram 8 and on page 31.

Do not select more than one clock in the network as master clock. If no clock is selected as master clock, than the clocks on all computers are running as local clocks without synchronization.

Summary

The following table summarizes, how objects are distributed or made visible across the network:

Object Type	Associated with each other by
turnouts, signals, toggle switches	digital address
feedback indicators	digital address
push buttons and on-off switches with digital address	digital address
push buttons and on-off switches without digital address	logical network name
flagman indicators	logical network name
engines and trains	exist only once in the network
macros	logical network name
routes	logical network name
blocks	logical network name
schedules	remote schedules and logical network name

3 Establishing a Virtual Digital System Bus with +Net/D™

The recommended and optimal way to share the same digital system between different computers is utilizing the digital system bus by connecting the computers directly to the digital system through an appropriate interface.

In cases where this is not possible (for example because the digital system does not allow to connect several computers directly), it is possible to use **+Net/D™**. **+Net/D™** is an extended variant of **+Net™**. It provides all features of **+Net™**. Additionally **+Net/D™** contains a virtual, software based Digital System Bus, to which additional computers can be connected in order to share the same digital system*.

This is shown in the image below:

* For technical reasons **+Net/D™** is only offered in countries, where digital systems without own Digital System Bus are frequently used.

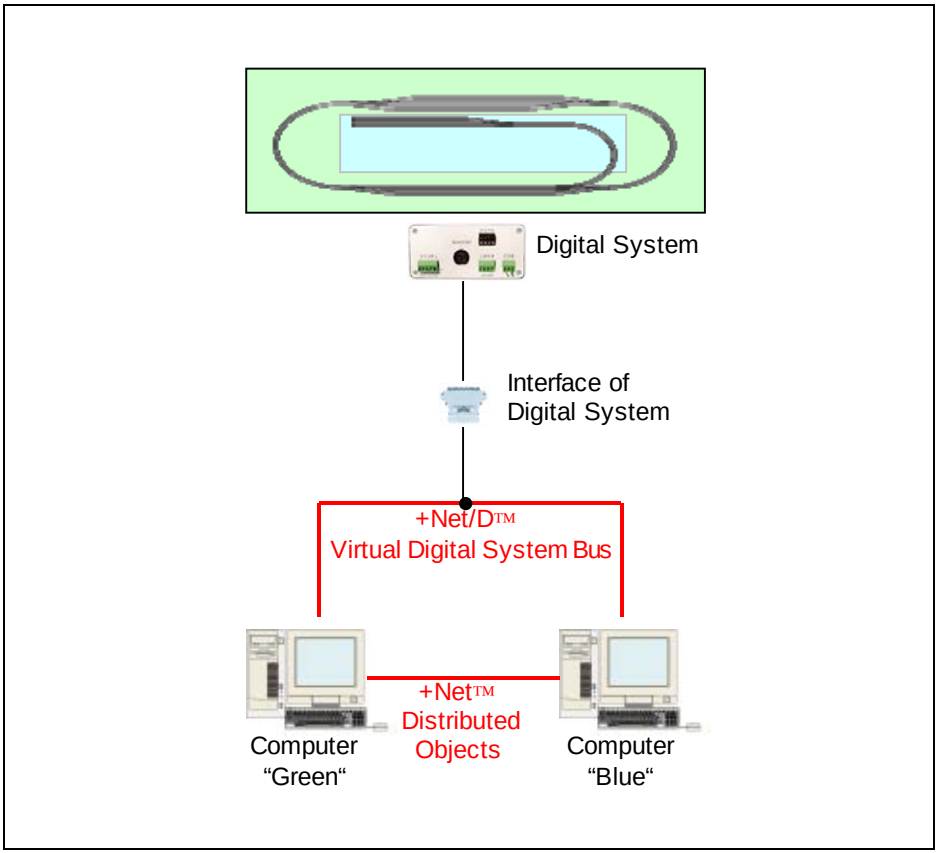

Diagram 6: Virtual Digital System Bus

To create one or more Virtual Digital System Busses between several computers, **+Net/D™** must be installed on all participating computers; the interface of the digital system is to be connected to a free serial or USB port of one of these computers. Each installed Virtual Digital System Bus allows sharing of one digital system between two or more computers.

+Net/D™ allows creation of a Virtual Digital System Bus for each digital system connected to the layout. It is for example possible, to extend the configuration of Diagram 6 in a way, that a previous local digital system connected to computer “Green” is shared with computer “Blue” by a second Virtual Digital System Bus. The network topology can be created with almost no restrictions. It is also possible to mix the types of connections (original bus of the digital system, Virtual Digital System Bus, direct connection to a local digital system) as desired. It is for example possible, to extend the configura-

tion of Diagram 4 in a way, that the local digital system connected to computer “Green” is shared with computer “Blue” by a Virtual Digital System Bus, that is operated in addition to the existing hardware based system bus of the main digital system.

In other words: the configuration types shown in the images of this document can be combined with almost no restriction.

There is one restriction, however, that applies, if two or more Virtual Digital System Busses connect the same set of computers. If two or more Virtual Digital System Busses connect two or more computers, then the digital systems associated with the particular Virtual Digital System Busses must be connected to the same computer. If, for example, computer “Green” and computer “Blue” share two digital systems through two Virtual Digital System Busses, then both digital systems must be connected to the same computer, i.e. either to “Green” or to “Blue”. It is not allowed to connect one digital system to “Green” and the other to “Blue”. This limitation, however, is not caused by the architecture of the Virtual Digital System Bus, but caused by the method, how **TrainController™** initializes its data after session start.

The above restriction does not apply to locally connected digital systems. In any of the configurations discussed in this document it is allowed to connect as many local digital systems to computer “Green” and/or “Blue” as desired.

A Virtual Digital System Bus must be established, if a digital system, that does not support an own digital system bus, shall be shared between two or more computers.

For the following digital systems and their system busses no Virtual Digital System Bus is needed:

- Digitrax / Digitrax LocoNet
- Lenz / Lenz XpressNet
- Selectrix, MÜT, Rautenhaus / SX Bus

If you are using these systems, then it is recommended to connect the particular computers directly to the digital system bus with an appropriate interface as displayed in Diagram 3.

If another digital system shall be shared between two or more computers, a Virtual Digital System Bus must be established, that connect these computers.

- **+Net/D™** supports all digital systems supported by **TrainController™**.
- One Virtual Digital System Bus allows sharing of one digital system between two or more computers.

- If more than one digital system is to be shared, a separate Virtual Digital System Bus must be established for each digital system.
- The shared digital system must be directly connected to one of the sharing computers by a free serial or USB connector. This computer is called *primary computer* of the Virtual Digital System Bus. The other computers connected to the same bus are called *secondary computers* of the Virtual Digital System Bus.
- A Virtual Digital System Bus connects one primary computer with one or more secondary computers.
- If two or more digital systems are shared between several computers with the appropriate number of Virtual Digital System Busses, then all digital systems must be connected to the same (primary) computer. If, for example, computer “Green” and computer “Blue” share two digital systems through two Virtual Digital System Busses, then both digital systems must be connected to the same computer, i.e. either “Green” or “Blue” must be primary computer of both Virtual Digital System Busses. It is not allowed to connect one digital system to “Green” and the other to “Blue”.
- Configurations in the following form are allowed, though: computer “Green” is primary computer of a Virtual Digital System Bus, that connects computer “Green“ and ”Blue“. Computer ”Yellow“ is primary computer of a Virtual Digital System Bus, that connects computer ”Yellow“ and ”Blue“. In such configuration, however, “Green” and “Yellow” must not be connected by a common Virtual Digital System Bus.

Virtual Digital System Busses are created with **+Net/D™**. The only difference between **+Net/D™** and **+Net™** is the support of Virtual Digital System Busses provided by **+Net/D™**. Thus everything written in this manual with the exception of the current chapter apply to both, **+Net/D™** and **+Net™**. The content of the current chapter only applies to **+Net/D™**, however.

The creation of a Virtual Digital System Bus is quite simple. At first setup the shared digital systems on the primary computer. This is done in the usual way with the **Setup Digital Systems** dialog.

On the secondary computers a new Virtual Digital System Bus is added in the same way as a usual digital system. After adding a new entry to the list of available digital systems in the **Digital System Setup** dialog, fill in the **Digital System** dialog as outlined in the following. If **+Net/D™** is installed, then the **Digital System** dialog contains some additional options as displayed below:

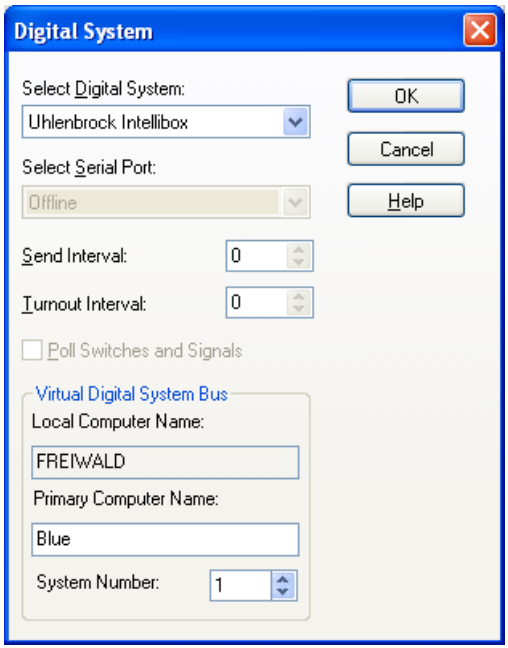

Diagram 7: Configuration of a Virtual Digital System Bus

The following additional options are available:

Local Computer Name:
The name of the current computer in the underlying TCP/IP network is displayed here. The name is displayed for informational purposes, e.g. to help you to link to this computer from other computers.

Primary Computer Name:
Specify the name of the remote computer, to which the concerning digital system is physically connected. In the diagram displayed above it is assumed, that the digital system is directly connected to computer “Blue”. The local, secondary computer is connected to this digital system through a Virtual Digital System Bus, that connects the local computer with the primary computer “Blue”.
Leave this option blank on primary computers.

System Number:
Specify the number of the digital system on the remote computer. If, for example, three digital systems are directly connected to the remote computer, then enter 1, 2 or 3 here, to specify, to which of the three digital systems the Virtual Digital System Bus shall es-

tablish a connection. In the example above a Virtual Digital System Bus is established to the first connected digital system directly connected to computer “Blue”.
Specify 0 on primary computers.

- The **Digital System** dialog must be filled in according to Diagram 7 on all secondary computers of the concerning Virtual Digital System Bus.
- The selection of the correct type of digital system on a secondary computer is not mandatory. The type of digital system connected to the primary computer will always override this setting. It is recommended , however, to select the type of digital system on the secondary computers correctly, too, because if you are testing your configuration offline, i.e. with the Virtual Digital System Bus disabled, it is better, when the correct type of digital system is known to the software.

4 Installation, Licensing and Initialization

Installation

At first install a physical, TCP/IP based network between all computers, that you want to use for distributed model railroad control with **+Net™**. Ensure, that this connection is properly working.

Installation of **+Net™** is very simple and self-explaining. Install **+Net™** on all computers in the network to the same directory, where **TrainController™** is already installed. If this is not done correctly, the **Network** tab displayed in Diagram 5 and the additional options displayed in Diagram 7 will not become visible.

+Net™ and **+Net/D™** are installed with the same installation file. The installation files of both products are identical. The program code of both products is always installed on your computer. Depending on the used license code (see the following section) either the functions of **+Net™** or additionally the functions of **+Net/D™** are available after program start.

Ensure also, that the same version of **TrainController™** is installed on all computers in the network. If this is not done correctly, connection between the computers will fail without further notice.

Licensing

A separate license of **+Net™** or **+Net/D™** is required on each computer, where **+Net™** is installed. Since each network contains at least two computers, at least two licenses of **+Net™** or **+Net/D™** are required to use the product. In contrast with the usual license conditions of **TrainController™**, that do not allow to run the non-network enabled version of **TrainController™** on different computers at the same time with the same license, you may run **TrainController™** simultaneously on different computers with your existing **TrainController™** license, if a valid and authorized license of **+Net™** is available for each computer in the network.

In other words: the combination of a **TrainController™** license, that may be common for all computers in the network, and a **+Net™** or **+Net/D™** license, that is individual for the particular computer, results in an individual license of **TrainController™+Net™** on the particular computer. This individual license may be used simulta-

neously with different individual licenses on other computers of the same network at the same time.

Licenses of **+Net/D™** are only needed, if you want to establish Virtual Digital System Busses (see chapter 3) between your computers. In this case a separate license of **+Net/D™** is required for each computer, that is connected to at least one Virtual Digital System Bus.

If the same **+Net™** or **+Net/D™** license is being used on different computers at the same time, then connection between these computers will fail without further notice.

It is only possible to install either a license of **+Net™** or a license of **+Net/D™** on the same computer, but not both. Depending on the installed license either the functions of **+Net™** or additionally the functions of **+Net/D™** are available after program start.

As long as no license is installed it is only possible to test the functions of **+Net™** and **+Net/D™** locally by starting **TrainController™** several times on the same computer. (see section “Local Testing”, page 34).

The **+Net™** licenses can be managed by using the **Network Setup** command of the **Railroad** menu. This command opens the following dialog:

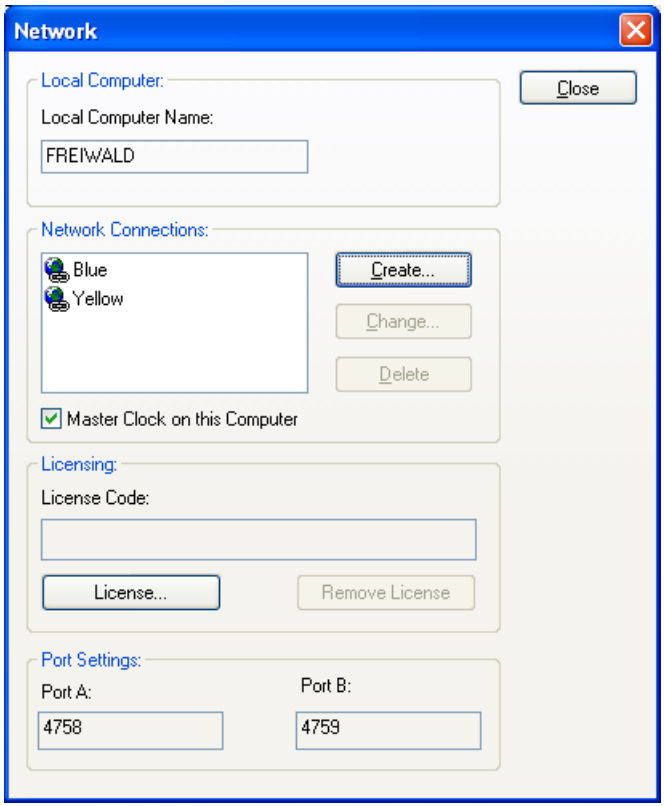

Diagram 8: License management and other Network options

Local Computer Name:
The name of the current computer in the underlying TCP/IP network is displayed here. The name is displayed for informational purposes, e.g. to help you to link to this computer from other computers.

Network Connections:
This list shows all defined network connections used for distribution of objects (see chapter 2, “Distributed Objects”). With the options **Create**, **Change** and **Delete** it is possible to create new connections, to change the computer name of existing connections or to delete connections, that are no needed anymore.

Master Clock on this Computer:
Select this option, if you want the clock on this computer be the master clock for other computers in the network, too. If this option is selected, then the clocks on all computers displayed in the list will follow the clock on this computer.
If this option is not selected, then the clock on this computer is not treated as master clock. It will be synchronized from time to time by a suitable master clock on another computer.

License Code:
The already installed code of your **+Net™** license valid for this computer is displayed here. The code cannot be changed here.

License…:
If no license for **+Net™** or **+Net/D™** was installed yet on this computer, use this option to install a new license on your computer.

Remove License:
Use this option to remove the installed **+Net™** or **+Net/D™** license, if any, from your computer. This option is useful, if you want to move a license from one computer in the network to another or if you want to replace a **+Net™** license by a **+Net/D™** license or vice versa.

Port A / Port B:
+Net™ uses two TCP/IP ports for communication. By default, the port numbers 4758 and 4759 are being used. The port numbers cannot be changed in this dialog.
In cases, however, where other applications occupy the same port numbers, it is possible to change the used port numbers by adding the following entries to RAILROAD.INI:

[Connections]
NetworkPortA=xxxx
NetworkPortB=yyyy

Where xxxx and yyyy specify the desired numbers of still available ports. Handle these options with extreme care! The ports must be identical on all computers in the network! Change these port numbers only in the rare case, when nothing else helps to avoid conflicts with other applications in conjunction with occupied port numbers!

Initialization and Synchronization

To start a network wide session with **TrainController™+Net™**, start **TrainController™** as usually and load the data file, that belongs to this particular computer.

If this has been done correctly call the **Connect** command of the **Railroad** menu again on all participating computers.

A green **Network** indicator in the status line of each **TrainController™** instance indicates proper connection.

Connection may fail among others for the following reasons:

- The underlying TCP/IP connections between the participating computers were not established properly.
- The **TrainController™** versions installed on the particular computers are not identical.
- Identical **+Net™** or **+Net/D™** license codes are installed on different computers.
- Improper configuration of computer names, network names etc.

In case of network errors an error message is displayed in the **Message Window** of **TrainController™**. These messages contain an error code, that describes the particular error any further. For a list of the most frequent error codes refer to the appendix of this document, please.

If the status of certain objects gets out of sync during a session for what reason ever, then it is possible to synchronize the status again by using the **Synchronize to Remote Computer** or **Synchronize from Remote Computer** command of the **Railroad** menu. The first sends the status of all distributed objects from the local computer to their counterpart located on one or more remote computers; the second requests one or more remote computers to send the status of all distributed objects stored on these computers to their counterpart stored on the local computer. The **Synchronize to** command synchronizes remote computers with the status of the local computer; the **Synchronize from** command does the opposite. These commands are also available in the context menu of certain objects. This menu can be opened with the right mouse button. If called from this context menu, then these commands synchronize only the status of the selected object, if this object is distributed.

Synchronization is only performed for objects, that are distributed by their logical network name; it is not provided for objects, that are associated with each other by their digital address.

Local Testing

Local testing of the configuration is possible by starting **TrainController™** twice on the same computer. Change the name of all network connections (Virtual Digital System Busses on a secondary computer and/or connections of distributed objects) to "." Note, that it is sufficient to do this for the connections of only one distributed object according to Diagram 5, because changing the computer name of the connection of one object affects all other objects, that use this connection, too.

Another possibility is to call the **Offline** command of the **Railroad** menu. This closes all connections to connected digital systems and all network connections and enables you to test everything in offline mode. The previous connections can be established again by calling the **Connect** command of the **Railroad** menu.

Error Codes

In case of network errors **TrainController™** displays an error message in the Message Window, that contains an error code.

In the following the most frequent error codes and there meaning are listed:

22: The specified computer is not known. Often displayed, when the network is down or when a computer name was specified, that does not belong to a known computer in the network.

51: Network is unreachable. Often displayed, when the network is down or when an IP address was specified, that does not belong to a computer, that can be reached on the network. Ensure, that a running network connection exists between the concerning computers.

60: Connection timed out. Often displayed, when the specified computer can be reached but **+Net™** is not running on the other computer. This message can be ignored in many cases, because the connection is usually automatically established later, when **+Net™** is started on the other computer.

61: Connection refused. Very similar to error code 60. See there for further explanation.

65: The specified computer cannot be reached. Very similar to error code 51. See there for further explanation.

1001: The versions of **TrainController™**, that are installed on the particular computers, do not match.

1002: Licensing problem. Often displayed, when the same **+Net™** license code was installed on different computers.

1003: Timeout. The specified computer did not respond in a timely manner.

1004: Lost connection to the specified computer. Often displayed, when **+Net™** is abnormally terminated on the other computer.

Index