COBALT Reference Manual

This document describes the installation, configuration and use of the COBALT software. It is divided into the following sections:

- Overview
- Hardware and Software Requirements
- Networking Primer
- How TEMPO and COBALT Work Together
- Installing and Configuring COBALT
- Procedure for Starting and Stopping COBALT
- Testing COBALT
- COBALT Databases
- COBALT Commands
- Integrating COBALT Into Your TEMPO Protocols

1 Overview

The COBALT software is an add-on product to TEMPO that allows you to collect high speed analog data that is synchronized with TEMPO data. Typical applications for COBALT include collecting high-speed raw analog data and real-time analysis of high speed analog data such as spike sorting and spike-triggered averaging.

COBALT collects analog data in parallel and independently from TEMPO's data collection. A COBALT system can collect up to 16 analog channels simultaneously. The maximum resolution per sample and the maximum sample rate is limited by the data acquisition board used in the COBALT system. The PCI-DAS1602/12 has a maximum aggregate sample rate of 330,000 12-bit samples per second while the PCI-DAS1602/16 has a maximum aggregate sample rate of 200,000 16-bit samples per second. For example, the PCI-DAS1602/12 can collect 3 analog channels at 100,000 samples per second per channel (aggregate rate of 300,000 samples per second). Alternatively, can collect 16 analog channels at 20,000 samples per second per channel (aggregate rate of 320,000 samples per second) or any other combination of channels and rate that the analog board is capable of acquiring.

COBALT databases have a structure similar to TEMPO Analog Append (XAPP type) databases. Each COBALT database is defined by a number of parameters such as the database number, the database tag, the period of an epoch, the number of epochs, amount of data collected prior to the trigger (offset), amount of time to suspend data accumulation (extension), the first analog channel to collect, the number of contiguous analog channels to collect and channel subsampling. The trigger cancellation mechanism is identical to TEMPO; the CANCEL bits in TEMPO are used to cancel pending trigger requests in COBALT.

COBALT databases are triggered (peri-event) on protocol triggers executed by your TEMPO protocol. COBALT uses real-time automatic time-synchronization to align COBALT's analog data stream with TEMPO's data. This allows data collected by COBALT to be time-aligned on TEMPO's protocol triggers. For example, it is possible for a single TEMPO trigger to cause an epoch of data to be collected on both the TEMPO system and the COBALT system. The epochs (on the TEMPO system and on the COBALT system) will be aligned on the same trigger.

The COBALT Architecture allows you to add one or more COBALT systems to a single TEMPO system. Each COBALT system is seamlessly integrated with TEMPO. This architecture effectively allows you to increase the number of analog channels that can be collected simultaneously. The COBALT Architecture is similar to the TEMPO Architecture: the COBALT real-time server acquires the analog data and streams it over a local area network to the COBALT client. The COBALT client synchronizes COBALT's data with TEMPO and builds one or more analog database by appending epochs of analog data when a TEMPO's triggers occurs. Thus a COBALT system comprises of a COBALT server computer networked to a COBALT client computer.

All COBALT systems are seamlessly integrated into your TEMPO system. Commands can be sent from any system to any other system. For example, your TEMPO protocol can send a command to any or all COBALT systems (i.e., to save their databases to a disk file, start or stop their clock, to load a new configuration, etc.)

1.1 COBALT Features

The following is an list of features currently available in COBALT.

Fast Analog Data Acquisition. COBALT can collect analog data much faster than TEMPO can. COBALT is limited only by the maximum aggregate acquisition rate of the analog acquisition board. Each COBALT system can collect 12 bit or 16 bit analog data, depending on the analog board in that system. Each COBALT system can be configured to acquire 2 to 16 analog channels. The rate (speed) at which the analog data is sampled and the number of analog channels are user defined. For example, with the PCI-DAS1602/16 board, COBALT can be configured to collect 2 16-bit analog channels at 100,000 samples per second per channel, 8 channels at 25,000 samples per second per channel or 16 analog channels at 12,500 samples per second.

Databases. COBALT can build multiple databases simultaneously, collecting 12 bit or 16 bit analog data into one or more epochs triggered by a TEMPO trigger. COBALT databases share the same trigger tag number space as TEMPO so a COBALT databases can use the same trigger tag as one of your TEMPO databases or it can have its own unique trigger tag. Trigger cancellation works the same way as in TEMPO: Each database has a CANCEL OVERRIDE mask which is used to make the database selectively sensitive or insensitive to TEMPO's CANCEL bits. Post-trigger and pre-trigger data can be collected. Databases parameters such as period, offset, extension, number of epochs, first and number of analog channels and subsampling can be specified. Databases are accumulated in the COBALT client's memory and can be written to a disk file that is compatible with TEMPO's HTB file format.

On-Line Viewing. The COBALT client lets you create one or more database pages, each of which can display one or more COBALT databases. COBALT's data viewing capabilities are similar to TEMPO's viewing capabilities. Database pages can be positioned anywhere on the screen as can database views within a database page. Each database view lets you select the visible channels, the gain and duration of the view. Database views are updated each time the database changes so you see the most recent epoch. Alternatively, you can examine previously acquired epochs during an experiment. In a view, up to 20 epochs can be overlayed in one view. Each view lets you define the colors of various components of the view.

Multiple COBALT Systems. Each TEMPO system supports multiple COBALT systems. The SETUPTN program and the TEMPONET Configuration File format (.TN) has been enhanced to support all of the parameters for each COBALT system associated with a TEMPO system. SETUPTN also allows you to create COBALT Server and COBALT Diagnostic diskettes for each COBALT system.

COBALT Commands. The COBALT client implements a set of commands that are similar to TEMPO's commands. A COBALT configuration file stores the COBALT commands used to configure the COBALT system for a particular protocol (i.e., COBALT's acquisition speed, key macros, details of database parameters, database pages and views, etc.). COBALT commands can be entered manually in the COBALT client's command line window, from a COBALT command file, or from a dialog.

Distributed Control. From your TEMPO client, you can send a command to any or all COBALT systems. Your TEMPO protocol can use this method to control your COBALT clients (i.e., stop or start a COBALT client, load a COBALT configuration or save a COBALT database.) Conversely, each COBALT client can send a command to the TEMPO client or to other COBALT clients. It is even possible for one TEMPO client to send commands to another TEMPO client.

1.2 Planned Functionality

The following functionality is currently being developed and will be added to COBALT in subsequent releases. It is not currently available.

Spike Sorting.

- Multiple spike sorting algorithms, each with user-defined parameters
- New Spike Database Format (with variable length epochs, time stamped records)
- Automatic streaming of time stamped spike epochs to a disk file
- Automatic streaming of time stamped spike epochs to an external UDP application

Spike Triggered Averaging.

Dynamic Data Exchange (DDE) and UDP servers.

The COBALT Grid - A platform for experimenting with the distribution of real-time processing over multiple computers in order to perform a complex analysis of the pipelined, high speed analog spike data as it is acquired during an experiment. This includes:

- Pipelining of raw analog data to external applications.
- Pipelining of COBALT epoch data to external applications.
- COBALT database server for external applications (similar to TEMPO's DDESRV).

COBALT Dialogs.

A mechanism for COBALT to send information back to the protocol as COBALT's databases change. This allows both the TEMPO client and the TEMPO protocol to query COBALT systems for information such as the number of epochs acquired in a particular COBALT database.

Per-epoch user-defined tags for TEMPO and COBALT databases. This facilitates the process of associating epoch data in TEMPO databases with epoch data in COBALT databases.

2 Hardware and Software Requirements

The hardware and software requirements for COBALT are similar to the hardware and software requirements for TEMPO. COBALT is a client-server based system so two networked computers are required per COBALT system.

Please refer to COMPUTER.PDF for a complete description of hardware and software requirements and WIREPCI.PDF for wiring diagrams.

2.1 Definitions

For purposes of this discussion, the following terms are defined. For more networking terms, refer to the Networking Primer section in this chapter.

A *COBALT system* consists of one COBALT client computer and one COBALT server computer. The COBALT server software acquires raw, analog data (from your electrodes) and sends it to the COBALT client computer. The COBALT client software receives the analog data, builds databases and provides the user interface to control the COBALT system. Thus, a COBALT system architecture is similar to the TEMPO system architecture: both are implemented with a client-server architecture.

The word *network* refers to a *physical* ethernet-based local area network (LAN). A network can operate at a speed of 10, 100 or 1000 megabits per second (Mbps). Networks operating at 1000 Mbps are often referred to as *1 gigabit* networks. A network can be a simple crossover cable between two computers or it can be a series of wires that connect multiple computers using switches or routers.

COBALT utilizes two standard *network protocols*: UDP (a cousin of TCP/IP) and Netbeui (a high speed protocol from IBM). Network protocols are conventions used by programs to transfer information across a network. Both UDP and Netbeui are supported by Windows. Both UDP and Netbeui can coexist compatibly on a single network. Please do not confuse *network protocol*, the network language computers use to communicate with each other, with a *TEMPO protocol*, a description of your experiment that you write using the PCL language.

The word *connection* refers to a *logical* communication across a network between two programs. A program establishes a connection across a network to another program using a particular network protocol (UDP or Netbeui).

An *8255 chip* is a Digital I/O chip that contains 24 TTLs. A TTL (transistor-transistor logic) is an electrical signal that is in one of two states: 0v (low) or +5v (high). Measurement Computing offers several boards that contain 8255 chips (PCI-DIO24H contains one 8255 chip, the PCI-DIO48H contains two 8255 chips and the PCI-DIO96H contains four 8255 chips). You can install one or more of these boards to gain access to multiple 8255 chips on a single computer.

The 8255 chips on two computers are connected to each other via a *ribbon cable.* For example, the COBALTCable-50C ribbon cable connects all 24 TTLs on both 8255 chips of a PCI-DIO48H board to the corresponding TTLs on the two 8255 chips of another PCI-DIO48H board. The ribbon cable must not connect +/- 5 volt pins. You can make this cable or you can purchase it from Reflective Computing. See the COBALTCable50-C specification for more details.

A *TimeLink* connection is a ribbon cable that connects an 8255 chip on the TEMPO server computer to a corresponding 8255 chip on a COBALT server computer. The COBALT software utilizes TimeLink connections to communicate real-time time synchronization information between two computers.

2.2 COBALT Server Computer Hardware

The COBALT server computer has the similar requirements as the TEMPO server computer (see COMPUTER.PDF). In addition, the COBALT server computer requires:

- A PCI-DAS1602/12 or PCI-DAS1602/16 analog acquisition board from Measurement Computing, used for high speed analog data acquisition.
- A dedicated 8255 Digital I/O chip (i.e. PCI-DIO24, PCI-DIO48H, PCI-DIO96H from Measurement Computing) for its TimeLink connection to the TEMPO server computer.
- A COBALTCable-50C or equivalent ribbon cable connecting the COBALT server computer to the TEMPO server computer (for the TimeLink connection).
- Network connectivity between the COBALT client computer and the COBALT server computer. A 100 mbps network may work in some circumstances. We recommend a 1000 mbps network. COBALT supports the Intel PRO 1000 MT network controller.
- An IBM PC DOS 2000 US Version License (the CDROM or Electronic Distribution required).
- A diskette drive is required.

The COBALT server computer does not require Windows, but you may find it helpful to have Windows (or some other operating system) installed for testing the hardware and network connectivity.

The COBALT server software does not access the hard drive or the CDROM drive.

2.3 COBALT Client Computer Hardware

The COBALT client computer has the following requirements:

- Network connectivity between the COBALT client computer and the COBALT server computer. A 100 mbps network may work in some circumstances. We recommend a 1000 Mbps network.
- COBALT client software supports any Windows compatible network board.
- Network connectivity between the COBALT client computer and the TEMPO client computer. TCP/IP protocol is required. This is a low bandwidth connection so any network speed should work fine.
- Windows 2000 is recommended (98 or XP are acceptable).
- We recommend 512 Mb or more of RAM memory
- A fast CPU (i.e., 2.0 Ghz or faster).

The COBALT client performance will improve with additional RAM, a faster CPU speed (including multiple CPUs) and a faster network connection.

2.4 TEMPO Server Computer Hardware

In additional to the TEMPO server requirements (see COMPUTER.PDF), the TEMPO server computer must have:

- A PCI-DAS1602 based TEMPO kernel (KP*xxx*.EXE)
- A dedicated 8255 chip for a TimeLink connection for each COBALT server computer.
- A COBALTCable-50C or equivalent ribbon cable between the TEMPO server computer and each COBALT server computer (for TimeLink).

2.5 TEMPO Client Computer Hardware

In additional to all the other TEMPO client computer requirements (see COMPUTER.PDF), the TEMPO client must have:

• 10/100/1000 network connectivity to each COBALT client using TCP/IP protocol.

3 Networking Primer

The COBALT client and server computers use local area networking (LAN) to communicate with each other. The COBALT client also communicates with the TEMPO client using the LAN.

This section is intended to present some basic concepts about networking for the beginner. It is not intended to be a complete discussion of all networking issues.

First time readers can skip this section now and refer back to it later.

A *network protocol* is a convention used by two or more computers to communicate with one another. It defines in detail the format of data packets and the behavior of network components. In many respects, a network protocol is like a language (i.e, Japanese, English) in that two computers must use the same network protocol in order to communicate with one another.

The TEMPO and COBALT software utilize two network protocols: NetBEUI and TCP/IP. This section discusses the TCP/IP network. For information on NetBEUI, please refer to Chapter 1 of the TEMPO Reference Manual.

3.1 Host Computers, Names, Network Interface Card, Network Cable

A *host* computer is a computer that supports networking. Every host computer has a *host name* which must be unique on the LAN. The COBALT client computer and COBALT server computer are examples of host computers.

In order to communicate with other computers on a LAN, a host computer needs to have a *network interface card* or *network interface chip* (NIC). This is typically a PCI board (i.e., Intel Pro 1000 MT) or a network chip that is built into the mother board of the computer (i.e., Broadcom 5700 10/100/1000 network chip).

NICs can be implemented in software. For example, if you use a dial up connection to access the internet, a piece of software (called a dialer) simulates a NIC and allows you to access the internet. If you use a Virtual Public Network (VPN) to access remote computers over the internet, the VPN software adds a NIC simulator to your system.

Computers are typically connected to one another with cables. In some cases, wireless NICs can be used to allow computers to communicate over a radio-based network. In all of our examples, however, we assume your computers are connected via network cables.

The standard CAT-5, CAT-5e or CAT-6 twisted-pair network cable are common network cables. A *crossover cable* can be used to connect two hosts computers directly to one another, forming a tiny network of just the two host computers.

To connect more than two host computers to a network, you need a network switch.

3.2 Network Switches

A *Network Switch* is a device that, when connected to a LAN, allows you add multiple host computers to a LAN. Switches typically come in configurations that support 4, 8, 16 or more. With one or more switches on your network, you can have many host computers connected to your LAN.

For example, the Asante FriendlyNet GX5-800P network switch allows up to 8 computers to communicate with one another (see www.asante.com).

A LAN can have zero or more switches on it.

3.3 Bandwidth, Half-Duplex, Full-Duplex

The *bandwidth* of a connection refers to the maximum number of bits of data that can be communicated over that connection, typically measured in millions of bits per second (Mbps). Most devices such as NICs and switches are capable of 10 Mbps or 100 Mbps. Newer NICs and switches can also operate at 1000 Mbps. Most devices automatically detect and operate at the fastest bandwidth of a connection. For example, the Asante FriendlyNet GX5-800P network switch can communicate independently with up to 8 host computers at 10, 100 or 1000 Mbps.

Some devices are capable of sending and receiving simultaneously and at different speeds. Devices that can do this are said to operate in *full duplex* mode. Devices that can not send and receive simultaneously are said to operate in *half duplex* mode.

3.4 IP Address

Each host computer is assigned a unique (on the LAN) *IP address*. This is typically displayed as four numbers, each from 1 to 255, separated by periods (i.e., 192.168.0.2). This form is called a *dotted quad*. This IP address refers to a particular NIC on the host computer.

3.5 Dynamic and Static Addressing, DHCP Servers

A host IP address can be assigned *statically* by the user. When assigned statically, it does not change from its assigned value; this is called *static addressing*. The user must insure that each IP address on their LAN is unique.

As an alternative to static IP addressing, a host IP address can be *dynamically* assigned. When the IP address is dynamically assigned, it is obtained (*leased)* from a Dynamic Host Control Protocol (DHCP) server for a period of time, typically 3 days. This is done when the host computer is first booted but IP address assignment can also occur in other circumstances. This means that a host computer's dynamic IP address can change unpredictably from time to time without any warning. This is referred to as *dynamic addressing*.

DHCP servers can be implemented as software on a host computer or inside a router or network switch.

Dynamic addressing is often easier to maintain than static addressing. With static addressing, the user must configure the IP address (and gateway and subnet, see below) of each computer on the LAN and insure that no two computers have the same IP address. With dynamic addressing, this configuration is done automatically by the DHCP server, resulting in less management effort by the user.

When dynamic addressing is used, the subnet and gateway are also dynamically assigned.

3.6 Network Segment and Subnet Mask

Two computers are on the same *network segment* if they are able to communicate with one another without having to go through a router. Two computers connected to each other through zero or more switches without an intervening router are on the same network segment.

It is possible to subdivide the computers on a network segment into logical groups or *subnets*. Each logical group of computers on a network segment are configured to communicate with one another and can not communicate with a computer outside their subnet, even if that computer is on the same network segment.

These logical groups are defined in terms of a *subnet mask*. The subnet mask looks like an IP address (because it is specified as a dotted quad) but it is used as a bit-wise mask to define a subnet.

For example, suppose you have three computers A, B and C, with the following IP and subnet masks. This table shows the range of IP addresses on each subnet.

	IP address	Subnet Mask	Range of IP Addresses in Subnet
Computer A	192.168.0.2	255.255.255.0	192.168.0.0 to 192.168.0.255
Computer B	192.168.0.3	255.255.255.0	192.168.0.0 to 192.168.0.255
Computer C	122.2.1.2	255.255.255.0	122.2.1.0 to 122.2.1.255

Computers A and B are on the same subnet while computer C is on a different subnet from A and B.

The following test is used to determine if two computers X and Y are on the same subnet:

```
IF ((X.ip & X.subnet) == (Y.ip & Y.subnet))
       THEN X and Y are on the same subnet.

OTHERWISE, X and Y are not on the same subnet.
```

where X.ip refers to the IP address of computer X, X.subnet refers to the subnet mask of computer X, the '&' operator is a bitwise AND operation between the bits in the IP address and the bits in the subnet mask and the '==' operator is the bit-wise equivalence operator.

Writing the values in hexadecimal from the above example,

```
Value                       Hexadecimal
A.ip                        C0.A8.00.02
A.subnet                    FF.FF.FF.00
(A.ip & A.subnet)           C0.A8.00.00 <-------|
                                                |
B.ip                        C0.A8.00.03         | Same subnet
B.subnet                    FF.FF.FF.00         |
(B.ip & B.subnet)           C0.A8.00.00 <-------|

C.ip                        7A.02.01.01
C.subnet                    FF.FF.FF.00
(C.ip & C.subnet)           7A.02.01.00 <-------| Different subnet from A and B
```

From this result, it is easy to see that A and B are on the same subnet but C is on a different subnet.

Thus, the network segment defines a set of computer that are physically connected to the same LAN without an intervening router while the subnet mask defines logical subsets of computers within a network segment.

If computer A is on the same subnet as computer B and computer B is on the same subnet as computer C, this does not imply that computer A is on the same subnet as computer C! The reason is that computer B can be multi-homed: B can have several IP addresses and can communicate on several different subnets simultaneously. See below for more information on multi-homed computers.

3.7 Gateways and Routers

The networking standards makes communication between computers on the same subnet very efficient. But there are situations in which one computer needs to communicate with another computer that is on a different subnet (i.e., a computer in another country).

A *router* is a device that allows computers on one subnet to communicate with computers on other subnets. When it receives a packet from a computer on a LAN, it forwards the packet to zero or more routers until the packet reaches the destination computer. Routers can be implemented as dedicated boxes or in software on a host computer.

The *gateway* is the IP address of a router.
When a computer sends a packet to a remote computer, it uses the above subnet test to determine if the destination computer is on the same subnet. If the remote computer is on the same subnet as the sending computer, the packet is sent directly to the remote computer. If the remote computer is not on the same subnet, the packet is sent to the sending computer's gateway (i.e., a router) and the router forwards it to the remote computer. It is possible that the packet may be forwarded to many routers before arriving at the remote computer.

3.8 DNS

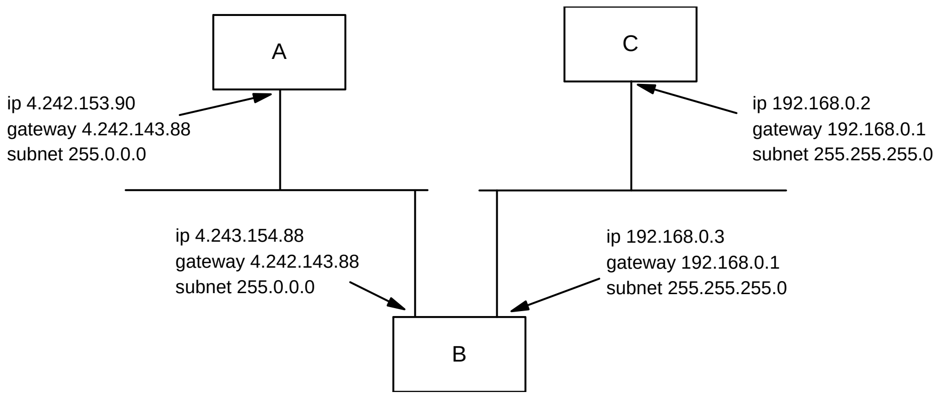
Most people find it easier to refer to their computers using names instead of dotted quads. For example, it is easier to remember www.google.com than 66.102.7.147.

You assigned each of your host computers a unique (on the LAN) host name. The association of this name to the host computer's IP address is maintained by a Domain Name Server (DNS). So when you type “www.google.com”, a request is sent to a DNS to obtain the IP address for Google's computer. Once Google's IP address is known, a request is then sent to it.

Windows implements a non-standard alternative to DNS called WINS. WINS performs the same basic functions as DNS but is extended to support Windows specific features.

3.9 Multi-homed Computers and Network Adapters

It is possible to have more than one NIC on a host computer. In this case, each NIC is assigned its own IP address. When a host computer has multiple IP addresses, it is said to be *multi-homed*.



A Multi-homed Computer (B)

Examples of multi-homed hosts are:

- Host computers that are on a LAN and use a dialer to connect to an internet provider.
- Host computers that are on a LAN and use VPN software.
- Host computers with multiple Network Interface Cards.

Each NIC has its own IP address; each dialer has its own IP address and each VPN has its own IP address.

The TEMPO and COBALT software use the phrase *network adapter* to refer to each IP address on a multi-homed host computer.

3.10 Windows IPCONFIG, Network Neighborhood, Control Panel

Windows offers several ways for you to obtain network information about your computer.

You can view your Windows network configuration using the IPCONFIG command. For example, on a Windows 98 computer, the output of IPCONFIG looks like this:

```
C:\>ipconfig

Windows 98 IP Configuration

0 Ethernet adapter :

       IP Address. . . . . . . . . : 4.242.153.88
       Subnet Mask . . . . . . . . : 255.0.0.0
       Default Gateway . . . . . . : 4.242.153.88

1 Ethernet adapter :

       IP Address. . . . . . . . . : 192.168.0.3
       Subnet Mask . . . . . . . . : 255.255.255.0
       Default Gateway . . . . . . : 192.168.0.1
C:\>
```

You can add the /ALL switch to get more information about each adapter:

```
C:\>ipconfig /all

Windows 98 IP Configuration
```

```
        Host Name . . . . . . . . . : HORIZON.ReflectiveComputing.com
        DNS Servers . . . . . . . . : 207.69.188.187
                                      207.69.188.186
                                      192.168.0.1
        Node Type . . . . . . . . . : Broadcast
        NetBIOS Scope ID. . . . . . :
        IP Routing Enabled. . . . . : No
        WINS Proxy Enabled. . . . . : No
        NetBIOS Resolution Uses DNS : No

0 Ethernet adapter :

        Description . . . . . . . . : PPP Adapter.
        Physical Address. . . . . . : 40-45-56-52-01-00
        DHCP Enabled. . . . . . . . : Yes
        IP Address. . . . . . . . . : 4.242.153.88
        Subnet Mask . . . . . . . . : 255.0.0.0
        Default Gateway . . . . . . : 4.242.153.88
        DHCP Server . . . . . . . . : 255.255.255.255
        Primary WINS Server . . . . :
        Secondary WINS Server . . . :
        Lease Obtained. . . . . . . : 01 01 80 12:00:00 AM
        Lease Expires . . . . . . . : 01 01 80 12:00:00 AM

1 Ethernet adapter :

        Description . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
        Physical Address. . . . . . : 01-09-44-CA-2B-00
        DHCP Enabled. . . . . . . . : Yes
        IP Address. . . . . . . . . : 192.168.0.3
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.168.0.1
        DHCP Server . . . . . . . . : 192.168.0.1
        Primary WINS Server . . . . :
        Secondary WINS Server . . . :
        Lease Obtained. . . . . . . : 05 24 05 6:31:49 AM
        Lease Expires . . . . . . . : 05 27 05 6:31:49 AM
C:\>
```

Here, you can see that there are two "adapters", one is the Intel PRO 1000/MT and the other a PPP Adapter used for a dial-up connection. You can also see the IP address, subnet mask and gateway address for each adapter. The *lease* information refers to duration of time the IP address is valid and indicates that dynamic addressing is used.

The Windows Network Neighborhood uses WINS to locate other Windows computers on your LAN. The Network Neighborhood is a convenient way to determine if you have network connectivity (can communicate) with another host computer on your LAN. Network Neighborhood detects only other Windows computers; TEMPO servers and COBALT servers will not appear in the Windows Network Neighborhood.

Finally, the Windows Control Panel/Networking icon gives you detailed information and lets you change the IP address, gateway and subnet.

3.11 Testing Network Connectivity

Network Connectivity refers to whether two computers are able to communicate via the LAN. There are several ways you determine if two computers are connected.

The PING Test. PING is a program available on Windows. It sends a message to a remote computer and waits for a reply. Windows, TEMPO servers and COBALT servers all respond to PING messages. To ping another computer, you can use one of several forms of the PING command.

```
C:\>ping www.google.com

Pinging www.l.google.com [66.102.7.147] with 32 bytes of data:

Reply from 66.102.7.147: bytes=32 time=140ms TTL=247
Reply from 66.102.7.147: bytes=32 time=126ms TTL=247
Reply from 66.102.7.147: bytes=32 time=126ms TTL=247
Reply from 66.102.7.147: bytes=32 time=126ms TTL=247

Ping statistics for 66.102.7.147:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 126ms, Maximum =  140ms, Average =  129ms

c:\>
```

You can also specify an IP address with PING. So PING 66.102.7.147 does the same thing as PING www.google.com.

Note that when you PING to a TEMPO or COBALT server computer, you must specify the server's IP address. The server's IP address is displayed on the top line of the server computer. TEMPO and COBALT servers will not respond to PING requests by its servername.

The BONG/BONGW Test. BONG, like PING, can be used to test network connectivity. The BONG test is described in Chapter 1 of the TEMPO Reference Manual. BONG.EXE is run on DOS computers (i.e,. on TEMPO servers and COBALT servers) while BONGW.EXE is run in a Command Prompt of Windows. BONG and BONGW work with both TCP/IP and NetBEUI configured computers.

The Network Neighborhood Test. The Network Neighborhood is used to test network connectivity between two Windows Computers. If each Windows computer displays the other computer in its Network Neighborhood, then the two Windows computers are able to communicate with each other.

3.12 Troubleshooting Network Problems

There are a number of reasons why two computers fail to communicate across a LAN. This section provides some helpful hints on how to locate and correct the problem.

Here are some of the components that can fail.

Network software (driver) is improperly installed or configured
Network Interface Card (NIC) or network chip is failing
A cable is damaged
Switch, router or other network device between the two computers is failing
A firewall device is preventing one of the computers from communicating with the other
Firewall software is preventing an application from accessing the network.
The computers are on different subnets.

Brownouts (partial or intermittent failures) can occur if:

The network is overloaded

A device (i.e., Router, switch) on the network is overloaded or intermittently failing
A cable is making a bad connection
Environmental conditions such as static, RF, ground loops, power surges, etc.
A NIC is loosely plugged into the computer

One way to determine the cause of a failure is to isolate the failing device by proving other individual components are functioning properly. Methodically proceed step by step "proving" each component is working.

For example, if you are experiencing brownouts, try wiggling all network cables, routers and switch boxes to see if you can force the failure to occur.

You may also be able to eliminate unnecessary devices between the two computers. Try connecting the two computers together with a crossover cable, eliminating all external network devices. If they can then communicate, you know that the computer software on each computer is configured correctly (and the NICs and cable are OK); the fault must be in a network device or cable.

If you are unable to establish any communication between computers with a crossover cable, then fault is on one of the computers: The network software could be improperly configured, a NIC could be faulty or the cable could be faulty. Try swapping the cable or NICs with a computer that is known to be working.

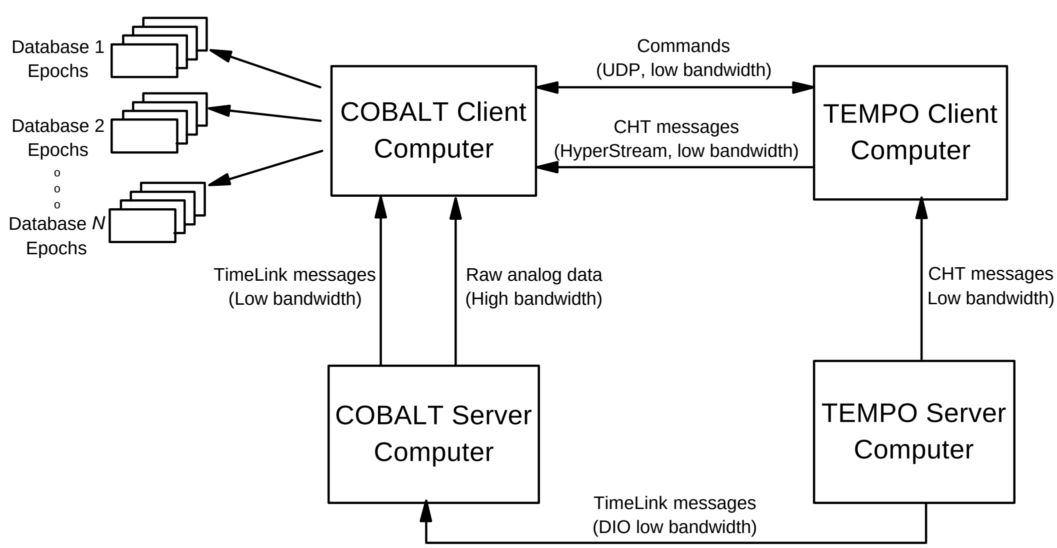
If each computer is able to communicate with other computers but they can't communicate with each other, verify that they both have the same subnet mask (i.e., using IPCONFIG).

4 How TEMPO and COBALT Work Together

TEMPO and COBALT use two types of communication, the TimeLink connection and a LAN, to collect data and synchronize database triggers.

4.1 COBALT Architecture

The following diagram illustrates the logical connections and data flow between a TEMPO system and a COBALT system.



COBALT Architecture

The COBALT server and TEMPO server independently acquire analog data from their pci-das1602 boards. This analog data is transmitted to their respective clients over a network connection.

The TEMPO server transmits synchronization information to each COBALT server via the TimeLink (ribbon cable) connection. Each TimeLink50 ribbon cable (for the pci-dio48h) is designed to support two COBALT servers or one COBALT server and one VideoSYNC computer. The TEMPO server can also use one or more pci-dio96 boards each of which supports up to four computers. See COMPUTER.PDF for more details.

CANCEL, HARVEST and TRIGGER (CHT) information is sent from the TEMPO server to the TEMPO client which forwards it to each COBALT client.

The COBALT client collates the analog and timing information it receives from the COBALT server with the CHT information it receives from the TEMPO client and builds its databases.

Finally, each client can send comments to any or all other clients. This allows the TEMPO client, for example, to tell all COBALT systems to load a configuration or to start their clocks. This means you can centralize the control of all of the COBALT system from your TEMPO client or even from within your TEMPO protocol.

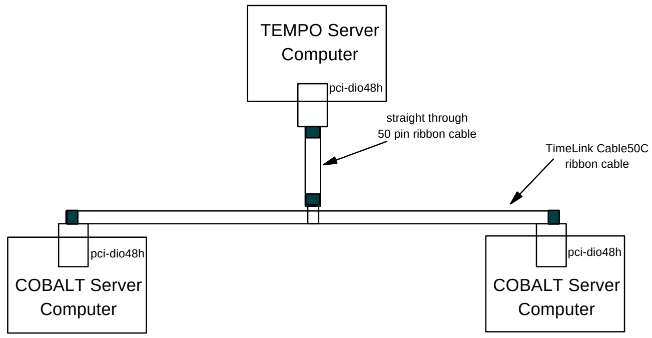
Note that the logical network connections shown above can operate on physically distinct networks or they can all be on the same physical network (i.e., using a network switch). If your network is heavily loaded, for performance reasons you may want to use a separate physical network between the COBALT client and COBALT server computers because this logical connection requires a high bandwidth network.

This modular architecture is designed to allow you to add additional COBALT systems to your TEMPO system as your acquisition capacity requirements increase.

4.2 TimeLink Connections

The TimeLink connection is used for the real-time time synchronization that occurs between the TEMPO server computer and each of the COBALT server computers. A private, dedicated connection is required between each COBALT server computer and the TEMPO server computer. Each TimeLink connection uses a dedicated 8255 chip.

The TimeLink Cable-50C is used to connect one or two COBALT server computers to a TEMPO server computer using a PCI-DIO48H board in each computer.



TimeLink Connections between TEMPO Server computer and one or two COBALT server computers

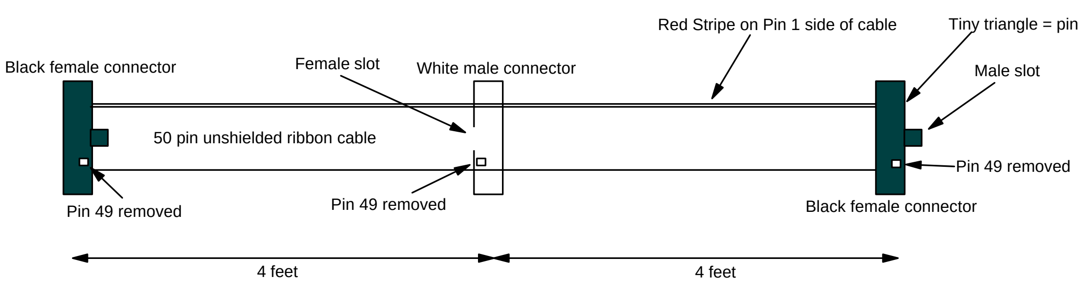
The PCI-DIO48H boards contain one 8255 chip and may be used to connect a TEMPO server computer to a COBALT server computer. A PCI-DIO98H board contains four 8255 chips. The TEMPO server computer can use a PCI-DIO96H board for TimeLink connections to up to 4 COBALT servers. When more COBALT servers are needed, more PCI-DIO96H boards can be installed in the TEMPO server computer.

4.3 TimeLink Cable Specification

The TimeLink Cable50-C is an 8 foot (approximate 2 m) unshielded 50 pin ribbon cable (28 AWG) that is designed to be used with the PCI-DIO48H, PCI-DIO96H and CIO-MINI50 boards. It connects the TEMPO server to one or two COBALT servers. It can also be used with VideoSYNC or for general purpose digital I/O.

The cable has three connectors: a black (female) connector at each end of the cable and a white (male) connector in the middle of the cable. The black connectors fit directly into the PCI-DIO48H or CIO-MINI50 boards.

All pins are connected except pin 49 (+5v), which is physically removed from each connector. You can use a flashlight to verify that there is no copper inside pin 49 of the black connectors. It is easy to see that pin 49 on the white connector has also been removed. When making the cable, you can use a small needle nose plyers to remove the gold pin inside each connector. The connectors are then crimped onto the cable as shown below.



TimeLink Cable50-C
All pins connected except pin 49

The red stripe on the cable corresponds to the pin 1 side of the cable. Each connector is slotted in a way that forces you to orient the cable correctly when plugging it into the PCI-DIO48H or CIO-MINI50 boards.

On the black connectors, there is a tiny triangle on the slotted side of the connector near the red stripe. This indicates pin 1 of the cable.

4.4 Network Connections

The COBALT server uses the network to send analog data to the COBALT client.

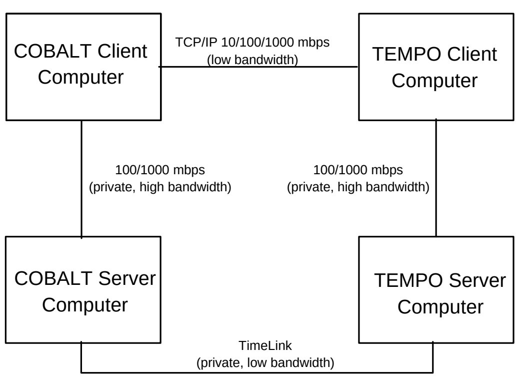
The COBALT client also receives TRIGGER messages from the TEMPO client when ever your protocol (on the TEMPO server) executes a TRIGGER statement. Other information is also sent including the CANCEL register and HARVEST statements. The COBALT client uses this information to build its COBALT databases.

The new command, the REMOTE command, is available on the TEMPO client and all COBALT clients and can be used to send commands or transfer files between computers.

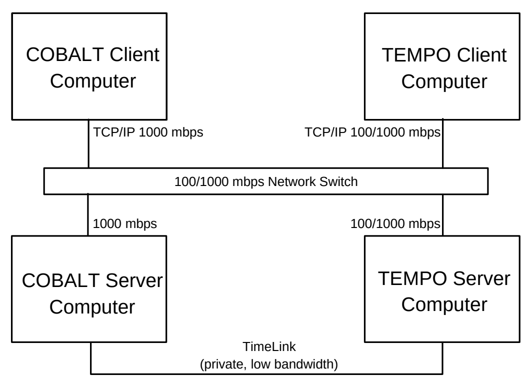
4.4.1 Local Area Network Connections

The COBALT client computer requires a low bandwidth UDP network connection to the TEMPO client computer. The COBALT client computer also requires a high bandwidth Netbeui network connection to its COBALT server computer.

There are several ways to configure the two networks when one COBALT system is connected to a TEMPO system: as separate networks or as one network.



Separate Network Cabling



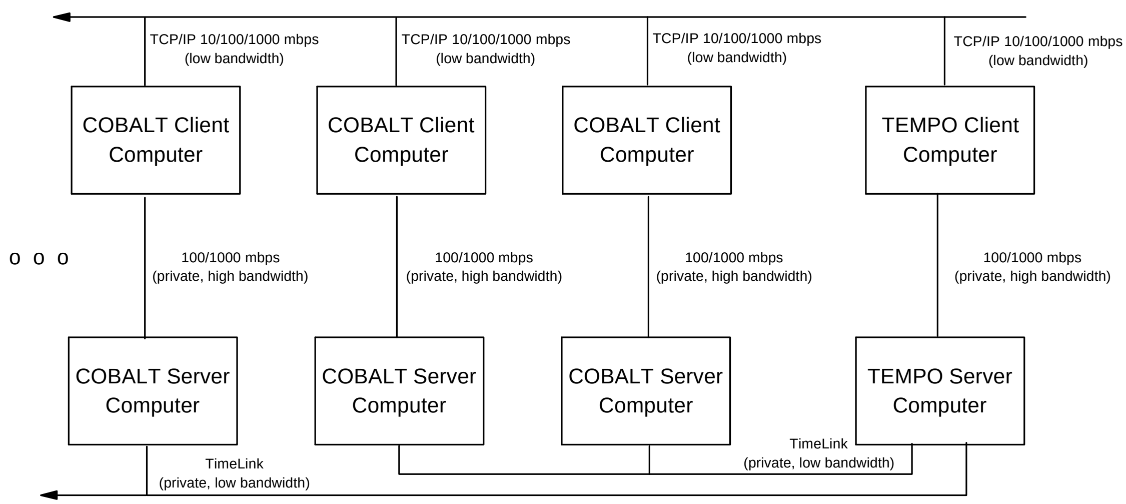
Shared Network Cabling

As one network, the COBALT client computer and COBALT server computer share the network with the TEMPO client computer and TEMPO server computer. Each computer has one network card. When one COBALT system is used with the TEMPO system, this may be sufficient. In this case, we recommend using a 1000 mbps network. Note that the communication between the TEMPO client computer and TEMPO server computer is also a high-bandwidth connection. Sharing two high-bandwidth connections on a single network may cause network traffic delays on a 100 Mbit network. A shared 1000 mbps maybe sufficient depending on the acquisition rates of the server computers and the network load.

When separate networks are used, it is possible to use a crossover cable between the COBALT server computer and COBALT client computer (and between the TEMPO server computer and TEMPO client computer). This is the preferred configuration, especially when multiple COBALT systems are used, because it separates the high-speed network traffic between the client computers and server computers from the lower speed network traffic between the COBALT client and TEMPO client. But it means that the COBALT client computer needs two network controllers: one connected to the COBALT server computer and one connected to the TEMPO server computer.

When multiple COBALT systems are used, it is preferable to keep each COBALT client computer and COBALT server computer networks separate. This keeps the high bandwidth network traffic between each COBALT client computers and server computers from colliding on the network, creating delays due to traffic interference from multiple COBALT server computers.

HyperStream is a proprietary network protocol that allows for high bandwidth data to be transferred reliably over an otherwise unreliable UDP connection. A HyperStream connection is used to send CANCEL, HARVEST and TRIGGER (CHT) messages from the TEMPO client to a COBALT client. The TEMPO client uses a HyperStream Send connection for sending CHT messages to each COBALT client. Each COBALT client uses a HyperStream Receive connection to receive CHT messages from the TEMPO client.



Multiple COBALT Systems With Separate Network Cables

In general, the HyperStream connections need to be set up only once. The TEMPO client HyperStream connection information in the TEMPO.TPO file; the COBALT client saves the HyperStream connection information in the COBALT.TPO file. Both clients automatically reestablish their HyperStream connections each time they are run.

4.4.2 Receiving Commands from Another Client

Each TEMPO client and COBALT client contains a UDP command server receives and executes command sent to it from another client. See the UDPSRV command for more information.

4.4.3 Sending Commands to Another Client

Before a client can send a command to another client, it must first *connect* to the other client. Each client maintains a list of zero or more clients to which it is connected. The TEMPO client stores this information in the TEMPO.TPO file. COBALT clients store this information in the COBALT.TPO file.

5 Installing and Configuring COBALT

This section describes a step-by-step procedure for installing a COBALT system to work with an existing TEMPO system. For information on installing a TEMPO system, please refer to Chapter 1 of the TEMPO Reference Manual.

These directions make the following assumptions:

1. Your COBALT client and server computers have Windows installed.
2. Your COBALT client, COBALT server and TEMPO client computers all appear in each other's Network Neighborhood.
3. The TEMPO software is installed and working on your TEMPO client and server computers.

In the following discussion, we use the following abbreviations:

CS	COBALT Server computer
CC	COBALT Client computer
TS	TEMPO Server computer
TC	TEMPO Client computer.

5.1 Install Hardware on COBALT Server

CS: Install the pci-das1602 and pci-dio boards.

CS: Install the network board (if required).

CS: Install the hardware KEY.

Also install the TimeLink cable between CS and TS.

5.2 Copy COBALT Software to TEMPO Client and COBALT Client

The TEMPO Client computer needs a copy of the COBALT specific files in the COBALT software distribution.

Copy the COBALT software from the COBALT distribution to the CC c:\tempo directory. Follow these steps on both the TEMPO Client computer and the COBALT Client computer:

1. `cd c:\tempo`
2. `pkunzip -d -o CO.ZIP`

5.3 Add C:\TEMPO to COBALT Client Path

Add C:\TEMPO to the PATH environment variable on the COBALT Client computer.

Follow the same directions as for TEMPO Client in Chapter 1 of the TEMPO Reference Manual.

5.4 Check Subnet and Network Connectivity

The subnet mask on the TEMPO client and COBALT client computers must match.

For static addressing the subnet mask for the COBALT server, specified by the COBALTServer1/Network/subnet parameter, must match the subnet on the COBALT client computer.

1. Run IPCONFIG on both TC and CC and verify that the subnets match.
2. Verify network connectivity between TC and CC.

   You can use any of the methods mentioned in the Network section of this chapter to test network connectivity between TC and CC. The BONGW test is easy.

   TC: Run BONGW SEND in a Command Prompt window
   CC: Run BONGW RECV in a Command Prompt window

   If CC receives messages from TC, they are connected.

5.5 Configure TEMPO Server's TimeLink to COBALT

You must inform the TEMPO server which 8255 digital i/o chip to use for sending TimeLink information to the COBALT system.

There are eight parameters under the TEMPOServer/Timelink section, one for each COBALT server, which specify which 8255 chip is used for TimeLink to each COBALT system.

For example, to use chip 0 of the pci-dio48h board for TimeLink, specify

```
TEMPOServer/TimeLink/TimeSend1_base=pci-dio48h,0,0
```

5.6 Configure COBALT Server

There are several steps required to add and configure a COBALT server to a SETUPTN TN file.

5.6.1 Add COBALT Server To TN File

Each COBALT server must have a unique (on your LAN) servername. If you plan to have multiple COBALT systems and multiple TEMPO systems, you may want to encode the TEMPO system number and the COBALT system number into the COBALT server name. For instance, for TEMPO 3's second COBALT system, the COBALT server name would be COBALT32. This is how Reflective Computing will refer to your COBALT systems. Of course, you can choose any name you want as long as it is unique.

The COBALT server name must contain only alpha-numerics, case is significant, and must be 15 characters or less.

To add a COBALT server's parameter tree to your TN file, follow these steps:

1. Edit your TN file with a text editor. At the end of the file, add a line that reads:

   ```
   /COBALTServer1/Network/servername=cobalt21
   ```

   This adds the cobalt21 system to the tempo2 server.

   If the TEMPO 2 system has multiple COBALT servers, each COBALT server is numbered consecutively: COBALTServer21, COBALTServer22, ..., etc.

```
/COBALTServer2/Network/servername=cobalt22
/COBALTServer3/Network/servername=cobalt23
```

2. Run SETUPTN, open the TN file and you should now see the tree of parameters for your COBALT Server(s).

5.6.2 Configure Parameters For COBALT Server

The COBALT specific SETUPTN parameters are stored in the COBALTServer*N* parameter tree, one set of parameters for each COBALT server. Most of COBALT's parameters are analogous to the TEMPO server's parameters. For purposes of our example, we will assume this is TEMPO 2 system's first COBALT server so the COBALT server name is *cobalt21.*

Here are some parameters that are particular important for configuring the COBALT server. Typical settings are shown. The **bolded** entries are the ones you may need to change. The unbolded entries are generally not changed by the user.

Parameter	Meaning
COBALTServer1/Configuration/OSdirectory=\PCDOS	*CS's operating system*
COBALTServer1/Configuration/kernel=cpd	*CS's "kernel"*
COBALTServer1/Network/servername=cobalt21	*CS's network server name*
COBALTServer1/Network/nwProtocol=tcp	*CS's network protocol*
COBALTServer1/Network/ksrv=csrvU	*CS's network server program*
COBALTServer1/Network/ethernet=E1000	*CS's NIC (network interface board)*
COBALTServer1/Network/ip=	*IP address of CS*
COBALTServer1/Network/subnet=	*CS subnet must match CC subnet*
COBALTServer1/Network/gateway=	*This is the IP of the gateway*
COBALTServer1/Network/lana=255	*CS's listening port*
COBALTServer1/AnalogInput/achannels=8	*Number of analog channels to acquire*
COBALTServer1/AnalogInput/speed=25000	*Default acquisition rate*
COBALTServer1/AnalogInput/asets=25	*Number of analog sets per interrupt*
COBALTServer1/AnalogInput/samples=0	*Size of analog buffer (0=autoselect)*
COBALTServer1/TimeLink/timeRecv_base=pci-dio48h,0,0	*TimeLink 8255 chip (from TEMPO)*

The SPEED and ASETS parameters determine the rate at which the analog input board will interrupt the COBALT computer during acquisition. Due to limitations on the analog board hardware, there may be certain combinations of SPEED and ASETS that are not possible.

The interrupt rate in Hertz is specified by the formula:

Interrupt rate (Hz) = speed / asets

It is important to keep the interrupt rate between 200 Hz and 2000 Hz or your COBALT server may experience data loss; ideally try to keep the interrupt rate at 1000 Hz. For example, if you want to acquire at 40,000 Hz, set

```
COBALTServer1/AnalogInput/asets=40
```

```
COBALTServer1/AnalogInput/speed=40000
```

You can change the SPEED from the COBALT client but you can not change ASETS from the COBALT client. This allows different protocols to collect at different speeds. For example, if you have ASETS=40 and you change the speed to SPEED 20000 in the COBALT client, the interrupt rate becomes 500 Hz.

The pci-das1602/12 has a maximum aggregate sample rate of 330,000 samples per second; the pci-das1602/16 has a maximum aggregate sample rate of 200,000 samples per second. You must not exceed your analog input board's maximum aggregate rate or the COBALT server will not acquire data. The aggregate rate is determine by the formula:

*Aggregate rate = speed * achannels*

For dynamic addressing, leave the IP, SUBNET and GATEWAY entries empty. For static addressing, they must be specified and the subnet must match the subnet on the COBALT client computer.

5.7 Make COBALT Diskettes

Once the CS's parameters are set to their desired values, you must make a COBALT Server diskette and a COBALT Diagnostic diskette. You will use the COBALT Server diskette to boot the COBALT server computer and the Diagnostic diskette to run the COBALT diagnostic program.

To make the Server and Diagnostic diskettes, click the NEXT button (in SETUPTN's Parameters dialog). In the Diskettes dialog, click the button corresponding to the diskette you with to make.

Any time you change the COBALT server parameters in SETUPTN, you must remake the Server diskette and reboot the COBALT server computer with the newly made diskette.

5.8 License the COBALT Server Computer

You must obtain a KEY file from Reflective Computing in order to run the COBALT server. This procedure is similar to licensing a TEMPO server computer. Here is a summary of the license process.

The first time you boot the COBALT server computer with the COBALT Server diskette, you will be asked to insert the COBALT Diagnostic diskette and press any key. When you do this, the diagnostic program runs and performs many tests. Some of these tests asks you to do things such as run BONGW SEND on the COBALT Client computer or move the server's mouse.

The test results from the diagnostic are written to the A:\COBALTS.LOG file on the Diagnostic diskette. Email this file to Reflective Computing and we will email you a KEY file which you install in SETUPTN's TN file for the COBALT Server. You then remake the COBALT Server diskette, reboot the COBALT server computer and the COBALT server runs.

To license a COBALT server computer, follow these steps.

1. Install hardware key into COBALT server computer's printer port.
2. Boot COBALT Server computer with COBALT Server diskette.
3. When asked to do so, insert COBALT Diagnostic diskette and press any key. This runs the Diagnostic program.

4. The diagnostic program will ask you to perform tests similar to what you did when running the Diagnostic program on the TEMPO server.
5. When the diagnostic completes, it stores the test results in the A:\COBALTS.LOG file (on the Diagnostic diskette). Email the A:\COBALTS.LOG file to Reflective Computing.

We will analyze the COBALTS.LOG file for any problems and send you a KEY file. To install the KEY file, please follow these steps:

1. Copy the KEY file to the C:\TEMPO directory
2. Run SETUPTN and OPEN your TEMPONET configuration.
3. In the Parameters dialog, click INSTALL KEY FILE button. (The button is in the upper right corner of Parameters dialog.)
4. Select the COBALT Server you want to install the key file on and click OK.
5. Browse to C:\TEMPO directory, select the KEY file and click OPEN. This installs the KEY file for the selected server.
6. Click NEXT to go to the Diskettes dialog.
7. Remake the Server diskette for the selected server.

Once the Server diskette is made, put it into the A: drive on the TEMPONET or COBALT server computer and reboot the computer. The Server software should start.

5.9 The Timelink Connection

The Timelink connection allows TEMPO server to communicate precise timing information to each COBALT server. Each TEMPO server supports Timelink connections to up to 8 COBALT servers.

A single Timelink cable can connect TS computer to one CS computer, two CS computers or one CS computer and one VideoSYNC computer. TimeLink uses the same RDX communication protocol as TEMPO so the cable wiring is identical to TEMPO's RDX cable to VideoSYNC.

PCI-DIO boards are required on TS and CS: there must be an available 8255 digital i/o chip available on TS for a TimeLink connection to each CS and each CS must have an available 8255 digital i/o chip available to receive Timelink messages.

With one CS and one VideoSYNC computer, a typical configuration uses a pci-dio48h on TS, CS and VideoSYNC computers. One TimeLink cable connects all three computers (the computer connected to the middle connector requires a 50 pin straight through cable in order to connect to the middle connector of the TimeLink cable to the pci-dio48h board on the middle computer.) See COMPUTER.PDF and PCIWIRE.PDF for detailed information on hardware and wiring requirements.

If the TimeLink connection is not working for any reason, both TEMPO server and COBALT server will periodically display warning messages. You may see one or two of these messages while you are booting the TEMPO and COBALT servers; these can be disregarded. If you don't see the periodic warning messages once the COBALT and TEMPO servers are running, your TimeLink connection is operational and you can skip this section.

If these periodic warning messages persist, the TimeLink connection may not be operational and you can use the steps in this section to determine the source of the problem.

5.9.1 Testing The TimeLink Connection

To test a TimeLink connection between a TS and a CS, follow these steps.

First, verify the pci-dio boards and cable are installed and working with the DDX Test. This is done by running the DDX.EXE program on TS and CS. This tests the TimeLink cable and the pci-dio boards. (We assume here you are using a pci-dio48h on both TS and CS.)

1. Boot TS and CS computers using their server diskettes. Exit the servers so that both computers are at the DOS prompt.
2. On TS at the DOS prompt type:

```
DDX send con pci-dio48h,0,0
```

3. On CS at the DOS prompt type:

```
DDX recv con pci-dio48h,0,0
```

4. Type some characters on the TS keyboard and press the ENTER key. If the pci-dio boards and the TimeLink cable are working, you should see the characters on the CS monitor.

If you do not see the characters on the CS monitor, there is a problem with your pci-dio boards or your TimeLink cable.

If you do see the characters you typed on the CS monitor, the pci-boards and TimeLink cable are operational. The problem is with your SETUPTN parameters for

```
TEMPOServer/TimeLink/timeSend1_base=pci-dio48h,0,0
```

or

```
COBALTServer1/TimeLink/timeRecv_base=pci-dio48h,0,0
```

parameter. Please check these to verify they are using the correct 8255 chips.

Finally, there are a number of other SETUPTN parameters, some unrelated to TimeLink, which the TEMPO server can use to access the PCI-DIO. Please verify that you are not using the TimeLink chip for another purpose. The parameters that use the pci-dio board are shown below. Each one, if set, should specify a unique 8255 chip.

```
/TEMPOServer/RDX/ddx_base=
/TEMPOServer/RDX/ddx2_base=
/TEMPOServer/DigitalInputOutput/dio0_base=
/TEMPOServer/DigitalInputOutput/dio1_base=
/TEMPOServer/DigitalInputOutput/dio2_base=
/TEMPOServer/DigitalInputOutput/dio3_base=
/TEMPOServer/DigitalInputOutput/dio4_base=
/TEMPOServer/DigitalInputOutput/dio5_base=
/TEMPOServer/DigitalInputOutput/dio6_base=
/TEMPOServer/DigitalInputOutput/dio7_base=
/TEMPOServer/TimeLink/timeSend1_base=
/TEMPOServer/TimeLink/timeSend2_base=
/TEMPOServer/TimeLink/timeSend3_base=
/TEMPOServer/TimeLink/timeSend4_base=
/TEMPOServer/TimeLink/timeSend5_base=
/TEMPOServer/TimeLink/timeSend6_base=
/TEMPOServer/TimeLink/timeSend7_base=
/TEMPOServer/TimeLink/timeSend8_base=
```

5.10 License the COBALT Client Computer

The COBALT client program, COBALT.EXE, must be licensed for the client computer before it will connect to the COBALT server computer. The procedure is slightly different than the procedure used to license the COBALT server computer.

To license the COBALT client computer, follow these steps:

1. Add C:\TEMPO to the PATH. Use the same directions as with TEMPO client.
2. Run COBALT.EXE once. This creates the serial number file C:\COBALT.SNO.
3. Run the diagnostic program (i.e., VPD CLIENT). The name of the diagnostic program is the same as the name of the COBALT kernel except with a 'V' prefix instead of a 'C' prefix. For example, use the VPD.EXE diagnostic program with the CPD.EXE kernel.

   Open a Command Prompt window and type

   ```
   CD \TEMPO
   VPD CLIENT
   ```

   The diagnostic program stores its test results in C:\TEMPO\CLIENT.LOG
4. Email C:\TEMPO\CLIENT.LOG to Reflective Computing.
5. We will analyze the results and email you the C:\TEMPO\PASSWORD.CFG file.
6. Put this file in C:\TEMPO\PASSWORD.CFG on your COBALT client computer. The COBALTW.EXE program is now licensed. You should now be able to run the C:\TEMPO\COBALT.EXE program to connect to the server.

At this point, you have now installed your COBALT software on both your COBALT server computer and your COBALT client computer. You are now ready to test your COBALT system.

6 Procedure For Starting and Stopping COBALT

This is the recommended order for starting your TEMPO and COBALT systems.

1. Boot VideoSYNC, TEMPO server and all COBALT servers
2. Start TEMPO client
3. Start all COBALT clients

This is the recommended order to shut down your TEMPO and COBALT systems.

1. Stop the acquisition clock on all servers
2. Save data and exit all COBALT clients.
3. Save data and exit TEMPO client.
4. Shutdown all TEMPO server, COBALT servers and VideoSYNC computer.

7 Testing COBALT

You must verify that your COBALT system is collecting analog data correctly. This includes verifying that the collected analog data on the COBALT system accurately measures your signal. This is usually done by measuring an known signal to verify that COBALT database contains the correct analog values.

Another important test is to verify that COBALT is synchronizing with TEMPO. When the TEMPO protocol triggers a database accumulation, the moment of the trigger on the two systems should be aligned on approximately (within hardware tolerances) the same moment in real time.

A special test protocol is used to help you check this synchronization. The CCQTEMPO protocol (in c:\tempo\prowin directory) creates an analog square wave once each process cycle and trigger a database once every few seconds. The trigger is emitted at the same time as the rising edge of the square wave. The duration of the square wave is a few tens of microseconds which is too fast for the TEMPO database to collect it. But the protocol runs COBALT at 100,000 samples per second (10 uSec per sample set) so COBALT's database should detect the square wave. The COBALT database collects two analog input channels so you can connect another (known) signal to verify that COBALT is acquiring an external signal correctly.

If the synchronization is working correctly, the left (rising) edge of the square wave should be aligned with the trigger. Slight fluctuations (approximately +/- 1 sampling period) are normal and are due to subtle differences in the behavior of the clocks on each pci-das board. The position of trailing edge of the square wave will vary depending on non-deterministic activities on the TEMPO server; this is normal artifact and does not affect alignment. The important thing to watch for in the COBALT database is that the left rising edge of the signal is aligned on the trigger (see CCQCOBAL illustration below).

The protocol collects a number of epochs of data. So after collecting a few epochs, you can stop it, HSAVE the COBALT database and use the HTB.EXE utility program to inspect the analog data collected.

The CCQTEMPO.PRO is very simple. You may want to review it before running it. It enables a special test mode for TEMPO which causes the TEMPO kernel to automatically set the rising edge of the analog output channel on each process cycle. All the protocol has to do to create the square wave is to clear the analog output. The protocol also executes a TRIGGER periodically so that the COBALT database can collect epochs. Because the TRIGGER is always aligned on the first sample of the process cycle, the rising edge of the square wave should always be aligned with the trigger.

7.1 Setting SETUPTN Parameters For CCQTEMPO Test

The CCQTEMPO protocol requires that the TEMPO and COBALT servers have certain SETUPTN parameters set to particular values. Run SETUPTN and set the following parameters:

```
TEMPOServer/AnalogOutputs/das_daout0=0
TEMPOServer/Protocol/asets=5

COBALTServer1/AnalogInput/achannels=2
COBALTServer1/AnalogInput/speed=100000
COBALTServer1/AnalogInput/asets=100
COBALTServer1/AnalogInput/samples=0
COBALTServer1/AnalogInput/das_gain=0
COBALTServer1/AnalogInput/das_differential=no
COBALTServer1/AnalogInput/das_polarity=bipolar
```

Remake the TEMPO Server and COBALT Server diskettes. If you want to use different parameters for your normal operation, you can use a special set of diskettes for this test. Reboot the TEMPO and COBALT servers with their newly made diskettes.

7.2 Connect Analog Output Channel 1 on TEMPO to Analog Input Channel 2 on COBALT

A square wave is produced by the CCQTEMPO protocol using TEMPO's analog output channel 1. The COBALT system acquires this signal on its analog input channel 2. So you must install a wire to connect pins on the two pci-das cards.

Install a wire from TEMPO's pci-das1602 board pin 36 (Measurement Computing's "D/A Out 0") to COBALT's pci-das1602 pin 4 (Measurement Computing's "Analog Input Ch 1 High").

If you want, you can connect COBALT's analog input channel 1 (Measurement Computing's "Analog Input Ch 0" on pin 2) to a known signal (i.e., from a wave form generator) and record it at the same time.

7.3 Load and Run Protocol

On the TEMPO Client computer, follow these steps:

1. Run the TEMPO Client (TEMPOW.EXE).
2. CLOAD the CCQTEMPO.CCF file. It is in the C:\TEMPO\PROWIN\ directory on the TEMPO client computer.
3. Press the F1 key. This should display a simple dialog with a STOP, CLS, ZERO and START buttons

On the COBALT Client computer, follow these steps:

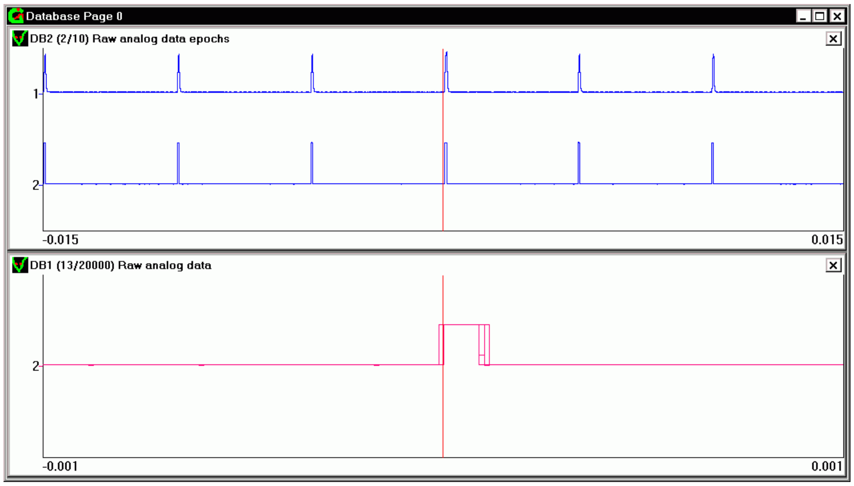
1. Run the COBALT Client (COBALT.EXE) and connect to the COBALT server.
2. CLOAD the CCQCOBAL.CCF file. It is in the C:\TEMPO\PROWIN\ directory on the COBALT client computer.

On the TEMPO Client, click the dialog buttons in this order: STOP, CLS, ZERO and START.

The STOP button stops both TEMPO and COBALT clocks.
The CLS button clears the message windows on both clients.
The ZERO button executes HZERO * on both systems, zeroing all databases.
The START button starts the acquisition clock on both computers.

You can use this dialog to repeated stop, zero and start the protocol.

COBALT's databases should look like this.



CCQCOBAL Database View
Note the left edge of the spikes are aligned with the trigger

In the above figure, you can see the left rising edge of Channel 2 square wave is aligned on the trigger. This is what you should see on your system.

The slight fluctuation or *jitter* (lower graph) of DB1 Channel 2's left edge (1 sampling period, 10 uSec) is due to slight fluctuations in the relationship between the two pci-das clocks. No two clocks tick exactly at the same rate so they have slight changes relative to one another, called *clock drift*. COBALT automatically compensates for clock drift to within +/- one sample set.

The jitter in the trailing edge results from the jitter in the CPU on the TEMPO server. This artifact occurs because of the variation from one process cycle to another in the time it takes the CPU to turn off the analog output.

Note that in the above example, Channel 1 is not connected. The data resulting from disconnected channels can be random and must be disregarded. We've observed that a disconnected channel tends to inherit similar but weaker signals from neighboring channels, which can be observed in this example.

This test protocol also provides a simple example of how to send commands to all of your COBALT systems from your TEMPO system using the REMOTE command. If you have multiple COBALT systems connected to a TEMPO system, this protocol sends commands to all of them.

8 COBALT Databases

Like TEMPO, each COBALT system supports multiple databases, each with its own parameters. COBALT's database parameters are also similar to TEMPO's databases. The period, offset, extension, number of channels, first channel, cancel override, trigger tag and TPB all behave the same as for a TEMPO database.

COBALT databases are created with the HEDIT dialog or the HOPEN command on the COBALT client computer. At present, only XSUM type (analog append) databases are available in COBALT.

Other H commands are available for zeroing, deleting, saving, loading and copying database formats (see COBALT commands in this chapter).

When you HSAVE a COBALT database to an HTB file, the file format is compatible with TEMPO's HTB database files (see Chapter 8 of TEMPO Reference Manual). The data file is stored on the COBALT client computer unless you specify a network drive in the file name.

You can use the HTB.EXE utility program to manipulate COBALT database files saved by HSAVE. See TEMPO Reference Manual Chapter 8 for more information on the HTB utility program and the HTB file format.

The rules COBALT uses for triggering and cancellation are also similar to TEMPO. The extension, CANCEL and CANCEL_OVERRIDE, TPB and HARVEST operate the same as they do in TEMPO.

The trigger tag you specify in your COBALT databases is the tag that your protocol must use in the TRIGGER statement to trigger the accumulation of an epoch into the COBALT database.

This implies that there is only one "trigger tag space": The trigger tags on TEMPO and COBALT all form a unified range of tags. COBALT database tags may be any number from 1 to 65535. A unified trigger tag space on COBALT and TEMPO systems makes it possible to have some databases with tags that are unique to each system *and* have databases (on different systems) that have the same tag and are, therefore, accumulated synchronously. Databases that have the same tag will be accumulated on the same TRIGGER statement regardless of whether the database resides on a COBALT system or on the TEMPO system.

For example, you can collect Analog, Spike and Event databases on your TEMPO system all triggered by trigger tag 1. You can also collect an Analog database on two COBALT systems, both with tag 1. All five databases will accumulate an epoch when TRIGGER 1 is executed by the protocol.

9 COBALT Client Commands

The COBALT client contains a command interpreter that processes commands.

The following is a list of available COBALT commands.

Command	Meaning
`hsave`	Save database
`hzero`	Zero database data
`hremove`	Delete database
`hcopy`	Copy database format
`hopen`	Create a database
`hfile`	Set database save file
`htitle`	Set database title
`hedit`	Modify database parameters
`stop`	Acquisition is stopped
`start`	Acquisition is started
`speed`	Set acquisition rate
`exit`	Close the client
`connect`	Connect to COBALT server

Command	Meaning
`disconnect`	Disconnect from COBALT server
`connectadd`	Add COBALT server
`connectdel`	Delete COBALT server
`connectauto`	Autoconnect to COBALT server
`connectsave`	Save connection information
`flush`	Write output log file
`log`	Open output log file
`cls`	Clear message window
`msg`	Add message to LOG file
`key`	Define KEY macro
`cclear`	Clear COBALT configuration
`cload`	Load COBALT configuration
`csave`	Save COBALT configuration
`edit`	Edit a file
`dos`	Execute DOS command (wait)
`dosx`	Execute DOS command (no wait)
`cd`	Change current directory
`cmd`	Execute command file
`show`	Show status display
`remote`	Communicate with remote client
`udpsrv`	Start command server

COBALT commands can be processed by entering them into the COBALT Command Window, from a command file, a KEY macro, a dialog button or from another client computer.

COBALT commands are designed to be similar to TEMPO's commands in syntax and meaning. However some TEMPO commands are not implemented in COBALT and others have slightly different meaning on the two systems.

9.1 HOPEN, HEDIT, HSAVE, etc.

The same 'H' commands found in TEMPO (HOPEN, HEDIT, HSAVE, HLOAD, HCOPY, HFILE, HTITLE, HZERO and HREMOVE) are also available in COBALT.

9.2 CSAVE and CLOAD Commands, COBALT Configurations

After creating one or more COBALT databases on the COBALT computer, you must save the configuration using the CSAVE command. This saves the configuration to a COBALT Configuration File (file extension CCF). Configurations can be later loaded with COBALT's CLOAD command. COBALT's CSAVE and CLOAD command behave the same as TEMPO's commands except that COBALT saves its configuration to a file on the COBALT client computer.

Thus, each of your TEMPO protocols you will have a corresponding COBALT CCF file on each COBALT client computer. When you CLOAD your TEMPO protocol, you will also CLOAD the corresponding CCF file on each COBALT system.

Note that with the COBALT command, it is possible for your TEMPO client to send CLOAD commands to all of your COBALT clients to load their CCF files.

9.3 The REMOTE Command

The REMOTE command is available on all TEMPO and COBALT client computers. It is used to send a command to one or more other clients on your network. The command that is sent is interpreted and by and executed on the remote client(s).

The REMOTE command has several subfunctions. The subfunction follows the word REMOTE in the command. The syntax for the REMOTE command is:

Command	Meaning
`REMOTE [/opt] CONNECT /`*`rHost`*`,..,`*`rHost Rport`*`, `*`Lport`*	Connect UDP to remote system
`REMOTE [/opt] DISCONNECT [/`*`rHost`*`,..,`*`rHost`*`]`	Disconnect UDP from remote host
`REMOTE [/opt] CONNECTHPR /`*`rHost hprRport`*`, `*`hprLport`*	Connect Hyperstream to remote host
`REMOTE [/opt] DISCONNECTHPR [/`*`rHost`*`]`	Disconnect Hyperstream
`REMOTE [/opt] INFO [/`*`rHost`*`,..,`*`rHost`*`]`	Display information remote host
`REMOTE [/opt] EXEC [/`*`rHost`*`,..,`*`rHost`*`] `*`command`*	Execute command on remote host
`REMOTE [/opt] GET [/`*`rHost`*`,..,`*`rHost`*`] `*`lFile`*` [`*`rFile`*`]`	Copy file from remote system
`REMOTE [/opt] PUT [/`*`rHost`*`,..,`*`rHost`*`] `*`lFile`*` [`*`rFile`*`]`	Copy file to remote system

where

[/opt] indicates optional switches (i.e., /timeout=nMs),
[..] indicates additional host names,
`[/`*`rHost`*`,..,`*`rHost`*`]` is an optional list of remote host names (no intervening spaces),
If `/`*`rHost`*`,..,`*`rHost`* is omitted, the action is performed for each connected remote host.

The REMOTE CONNECT and DISCONNECT Commands. When the COBALT client is started, it automatically establishes a connection with the TEMPO client. When the TEMPO client program is started, it automatically establishes a connection to each COBALT client. So you don't typically have to establish these connections manually. However, it is possible for you to manually control which clients the REMOTE command will send commands to. For example, you may want to connect two TEMPO clients so that one can send commands to the other.

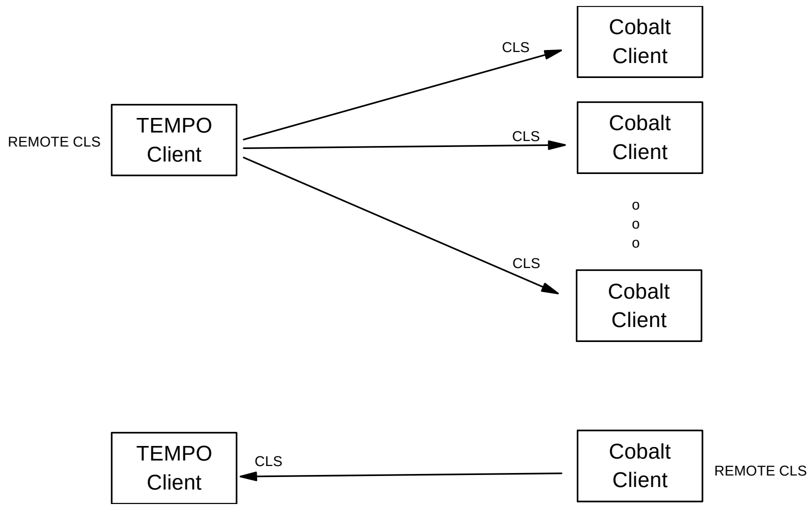
To connect to a remote client computer, use the REMOTE CONNECT command. For example, to connect to a remote client (TEMPO or COBALT) with the name *spike*, type:

```
REMOTE CONNECT /spike
```

The UDP remote and local port numbers assumes default values unless you specify otherwise.

Connection information to remote clients is saved to a file called COBALT.TPO (on TEMPO clients, its called TEMPO.TPO) so each time you run the client program, the connection information is automatically restored when the client program is started.

We suggest that you connect your TEMPO client to each of its COBALT clients; connect each COBALT client to its TEMPO client (but not to the other COBALT clients). From the TEMPO client, you can control all COBALT systems and from each COBALT system, you can control the TEMPO client.



The REMOTE command connections

The REMOTE DISCONNECT command breaks the connection to another client and removes it from the list of connected computers.

The REMOTE EXEC Command. To send a command to one or more connected clients, use the REMOTE EXEC subcommand. If you don't specify a particular client (or clients), the command is sent to all connected clients.

For example, to send CLOAD MYPRO.CCF command to all connected clients, type

```
REMOTE EXEC CLOAD MYPRO.CCF
```

The TEMPO client (TEMPOW.EXE) sends the CLOAD MYPRO.CCF command to all connected (i.e., COBALT) clients in sequence. After each connected client executes the CLOAD command, TEMPOW sends the CLOAD MYPRO.CCF command to the next client.

To send the command to only a client called *spike*, type

```
REMOTE EXEC /spike CLOAD MYPRO.CCF.
```

If you want to send a command to a list of clients, specify those client names, separated by commas, in the REMOTE EXEC command with the /*rHost* syntax (no spaces allowed in /*rHost* syntax) :

```
REMOTE EXEC /spike,wilbur CLOAD MYPRO.CCF.
```

The REMOTE command can be used in command files, TEMPO dialog buttons, key macros and can be used within your protocols via the system() and systemf() PCL functions. For example, within your protocol, you can use the systemf() PCL function:

```
system("REMOTE exec cload mypro.ccf");
while (system()) nexttick;       // Wait for all COBALT clients
```

The protocol waits in the while() statement for the REMOTE command on TEMPOW to complete. Once TEMPOW has sent CLOAD MYPRO.CCF to every connected client, the REMOTE command is complete and the system() function returns 0, terminating the while loop.

REMOTE GET and REMOTE PUT Commands. These commands let you transfer files to and from a remote client. The default local path is the current default path for the local client (see the CD command). The default remote path is the current default path on the remote client. Wildcard characters (i.e., asterisk (*) and question mark (?)) are not allowed. If you specify a path, use the forward slash (/) instead of the back slash (\) in the path.

The REMOTE INFO Command. This command provides information about a remote client and is used for diagnostic purposes.

9.4 SPEED, START, STOP

The acquisition rate is controlled by the SPEED command. Without arguments, the SPEED command displays the current acquisition rate. A single argument specifies the acquisition rate. For example, SPEED 40000 sets the acquisition rate to 40,000 sample sets per second. If the acquisition clock has been started, it is first stopped, the rate is changed and the clock is restarted.

The START and STOP commands start and stop the acquisition.

9.5 UDPSRV

The UDP command server is automatically started before a client can receive commands from another client. In general, you will not need to use the UDPSRV command because it is automatically started when you run the TEMPO or COBALT client program.

The UDPSRV command is used to start and stop the UDP server. The syntax is as shown below.

Command	Meaning
UDPSRV	Display UDP server command help
UDPSRV STATUS	Display status of UDP server
UDPSRV START	Start UDP server on default local UDP port
UDPSRV START, *n*	Start UDP server on local UDP port *n*
UDPSRV STOP	Stop UDP server

The UDPSRV command without any arguments displays a short help page.

The UDPSRV STATUS command displays the current status of the UDP server.

The UDPSRV START command starts the UDP server. An optional UDP port number may be specified (we recommend you use the default UDP port number).

The UDPSRV STOP command stops the UDP server.

On COBALT client computers, information about the UDP server is saved in the COBALT.TPO command file and is restored each time you start the COBALT client. Typically, you will need to start the UDP server once; after that, the UDP server will be started each time you run the COBALT client.

On TEMPO client computers, information about the UDP server is saved in the TEMPO.TPO command file and is restored each time you start the TEMPO client. Typically, you will need to start the UDP server once; after that, the UDP server will be started each time you run the TEMPO client.

10 Integrating COBALT Into Your TEMPO Protocols

This section presents a summary of information that is intended to help you integrate COBALT into your existing TEMPO protocols.

Please refer to the CPRIMER.PDF file for more information.

The following summarizes the steps you will take to integrating COBALT into an existing TEMPO protocol.

1. **Set COBALT Parameters**. On the COBALT client, set SPEED to the desired value. Also set the LOG file, if desired.
2. **Create COBALT Databases**. Use COBALT's HEDIT dialog or the HOPEN command to create the COBALT databases you want to collect. If you create your TEMPO databases using system() calls in the protocol, add system("COBALT EXEC HOPEN ... "); calls to create your COBALT databases from your protocol.
3. **Create COBALT Database Views**. Open one or more COBALT database pages and position the database views on the screen.
4. **Save COBALT Configuration**. Use COBALT's CSAVE to save the current configuration to a COBALT configuration file. This allows you to CLOAD it when you want to run the protocol.
5. **Add TRIGGER Statements to Your TEMPO Protocol**. If your COBALT database tags are different from your TEMPO database tags, add TRIGGER statements to your TEMPO protocol to trigger the COBALT databases at the desired time.
6. **Loading COBALT Configuration**. If you use a TEMPO dialog or key macros to CLOAD TEMPO protocols, add REMOTE EXEC CLOAD to CLOAD the corresponding COBALT configurations.
7. **Starting and Stopping COBALT**. If you use TEMPO dialogs or key macros to start and stop your protocol, add REMOTE EXEC START and REMOTE EXEC STOP commands to those dialogs or key macros to start and stop acquisition on the COBALT system.
8. **Zeroing and Saving Databases**. If you use TEMPO dialogs, key macros or system() calls in your protocol to zero and save your TEMPO databases, add REMOTE EXEC HZERO and REMOTE EXEC HSAVE commands for zeroing and saving your COBALT databases.

Once you have completed these steps, your COBALT system(s) can be completely controlled from your TEMPO system.