

T-Zero™  
*User's Manual*

2.0



**T-Zero™**

**User's Manual**

---

Version 2.0

Copyright © 2001 Command and Control Technologies Corp.

ALL RIGHTS RESERVED. This document contains proprietary and confidential information of Command and Control Technologies Corporation. No part of this publication may be disclosed to third parties, copied, or duplicated in any form without the prior written permission of Command and Control Technologies Corporation.

Command and Control Toolkit, T-Zero and RangeNet are trademarks of Command and Control Technologies Corporation. All other trademarks used in this document are the property of their respective owners.

### Contributors

Written by Rodney Davis and John Ward

Edited by Kevin Brown

Production by Patti Schill

Engineering contributions by Ross Manges and Eric Sorton

---

#### Corporate Headquarters

Command and Control Technologies Corp.  
1425 Chaffee Drive, Suite 1  
Titusville, FL 32780 USA  
Tel: 321.264.1193  
Fax: 321.383.5096  
Web: <http://www.cctcorp.com>

---

---

#### Technical Support

Tel: 321.264.1193  
Email: [support@cctcorp.com](mailto:support@cctcorp.com)  
Web: <http://www.cctcorp.com/support>

---

### LIMITED WARRANTY

**SOFTWARE.** Command and Control Technologies Corporation (CCT) warrants for a period of THIRTY (30) DAYS from the date of purchase that the T-Zero 2.0 (SOFTWARE) will execute its programming instructions as specified in the user documentation when properly installed. Due to the complex nature of computer software, CCT does not warrant that the operation of the SOFTWARE will be uninterrupted or error free.

**MEDIA.** CCT warrants the media upon which this SOFTWARE is recorded to be free of defects in materials and workmanship under normal use for a period of THIRTY (30) DAYS from the date of purchase.

**REMEDIES.** In the event this SOFTWARE fails to execute its programming instructions, or if any media proves to be defective during the warranty period, your remedy shall be to return the media to CCT for replacement. Should CCT be unable to replace the media within a reasonable time, your alternative remedy shall be a refund of the purchase price upon return of the product and all copies.

**WARRANTY VOID.** This limited warranty is void if the failure results from abuse, misapplication, or user modification of the software other than that provided for in the user documentation.

**NOTICE OF WARRANTY CLAIMS.** You must notify CCT in writing of any warranty claim prior to the expiration of the warranty period.

**NO OTHER WARRANTIES.** To the maximum extent permitted by applicable law, CCT disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE. This limited warranty gives you specific legal rights. Some jurisdictions do not allow limitations on how long an implied warranty lasts, so the above limitations or exclusion may not apply to you.

**NO LIABILITY FOR CONSEQUENTIAL DAMAGES.** To the maximum extent permitted by applicable law, in no event shall CCT be liable for any lost revenues or profits, loss of data or other special, indirect, incidental or consequential damages, even if CCT has been advised of the possibility of such damages. Because some jurisdictions do not allow the exclusion or limitations of liability for consequential or incidental damages, the above limitation may not apply to you.

**JURISDICTION.** This license is governed by the Laws of the United States.

**U.S. GOVERNMENT RESTRICTED RIGHTS.** The SOFTWARE and the documentation are "commercial items" as that term is defined in 48 C.F.R. 2.101 consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212. Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4, if the licensee hereunder is the U.S. Government or any agency or department thereof, the SOFTWARE and the documentation are licensed hereunder (i) only as a commercial items, and (ii) with only those rights as granted to all other end users pursuant to the terms and conditions hereof.

**T-Zero User's Manual**

MNL-TZ-User-110601

November 7, 2001

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>2</b>
1.1	MANUAL ORGANIZATION.....	2
1.2	REFERENCE DOCUMENTATION .....	2
1.3	CCT HELP DESK .....	3
<b>2</b>	<b>CCTK INTEGRATION.....</b>	<b>4</b>
2.1	INSTALLATION.....	4
2.2	T-ZERO CLIENT .....	4
<b>3</b>	<b>T-ZERO CLIENT GUI COMPONENTS .....</b>	<b>6</b>
3.1	MAIN WINDOW COMPONENT .....	6
3.2	MENU .....	7
3.3	CONNECTION PROPERTIES .....	8
3.4	MAIN WINDOW TOOLBAR.....	8
3.5	MAIN WINDOW CONTROLS.....	9
3.6	TIMELINE COMPONENT.....	9
3.7	TABLE CATEGORIES.....	10
<b>4</b>	<b>USING T-ZERO .....</b>	<b>12</b>
4.1	SEQUENCE CONSTRUCTION.....	12
4.2	SEQUENCE INTERACTION.....	13
	<b>APPENDIX A. DEFAULT PROPERTIES CONFIGURATION FILE .....</b>	<b>14</b>
	<b>APPENDIX B. SEQUENCE INPUT FILE DTD SPECIFICATION.....</b>	<b>16</b>
	<b>APPENDIX C. EXAMPLE SEQUENCE INPUT FILE.....</b>	<b>18</b>
	<b>APPENDIX D. COMMAND LINE OPTIONS.....</b>	<b>19</b>
	<b>GLOSSARY .....</b>	<b>21</b>
	<b>INDEX.....</b>	<b>28</b>

# 1 INTRODUCTION

T-Zero is an interactive real-time command and control product with an intuitive user interface called the “client GUI” that graphically shows task relationships in a time-based sequence. Each timeline accommodates symbols that represent process milestones, executable procedures, event status, and group decisions and interactions. Using T-Zero, your test conductor can construct sequences that gate activities by linking them to real-time data and events.

---

## 1.1 Manual Organization

This manual describes the general user operation of the T-Zero collaborative sequencer software. It describes the installation of the software, its components and its use.

Chapter 2 introduces the installation procedure and illustrates how T-Zero and CCTK software work together to provide a real-time interactive process control tool.

Chapter 3 describes each of the T-Zero buttons, menus, window controls, and other user interface elements. Chapter 4 explains how to use T-Zero.

A set of appendices provide listings of the configuration and example input files. This material is intended to be a reference for setting up and customizing the software on your system.

A glossary of commonly used terms, phrases, and acronyms for CCTK and T-Zero documentation is included at the end of the manual along with a keyword index.

---

## 1.2 Reference Documentation

The following documents complement the material presented in this manual:

- CCTK Installation Manual – Describes the steps needed to install CCTK client and CCTK server software from the CCTK CD. Installation on Microsoft Windows® and UNIX® systems is described. T-Zero installation is explained in the “index.html” file on the T-Zero CD-ROM.
- CCTK Administrator’s Manual – Describes setup, configuration, and administration of CCTK. The material addresses software setup, user administration, project configuration, hardware configuration, and other topics related to administration of a CCTK-based system.

- CCTK User's Manual – Provides an overview of the CCTK architecture and a complete description of how to use CCTK for operations.
- 

## 1.3 CCT Help Desk

The CCT Help Desk Phone Number is (321) 264-1193 (available 8:00 am to 5:00 pm, U.S. Eastern Time, Monday through Friday, excluding holidays).

CCT provides free unlimited telephone and e-mail support for 30 days after purchase as part of our standard warranty. Customers who purchase the technical support upgrade option have unlimited telephone or email support for one year after purchase or renewal. For questions or comments about CCTK or other CCT products or services please call (321) 264-1193 and ask to speak to a customer representative. For e-mail information, contact [info@cctcorp.com](mailto:info@cctcorp.com) or visit our website at <http://www.cctcorp.com>.

It is important when calling for help with a problem to preserve the project environment as much as possible to allow CCT engineers to efficiently diagnose the problem. If at all possible, leave the project and operation in its current state at the time of the problem and call the Help Desk. If it is not possible to stop and troubleshoot during an operation, please perform the following steps:

1. Completely document all of the symptoms of the problem and as much of the state and configuration of the project as possible.
2. Copy the KPATH directory to a location outside of the project environment (e.g., */tmp/projectname/date/*).
3. Note the time and date of the problem.

Additional help may be found at the CCT Technical Support web site <http://www.cctcorp.com/support>. Also, please review the RELEASE\_NOTES file in the root directory of the CCTK CD for additional information and instructions.

## 2 CCTK INTEGRATION

T-Zero is a plug-in application for the Command and Control Toolkit™ from CCT. The two software products work together to provide a graphically interactive method for controlling real-time operations.

### 2.1 Installation

Detailed installation instructions are contained in a file on the T-Zero program CD-ROM. Insert the CD into your Windows or UNIX computer and access it (browse to the CD using Windows Explorer or mount the CD using Linux). View the “index.html” file in the root directory of the CD. This file contains detailed instructions for installing the T-Zero software.

Please note that CCTK 2.0 must be installed prior to installing T-Zero 2.0.

### 2.2 T-Zero Client

The T-Zero Client GUI shown below in Figure 1 is a Java-based client/server application that communicates with the T-Zero server to provide the operator with a graphical task list vs. time sequence data view.

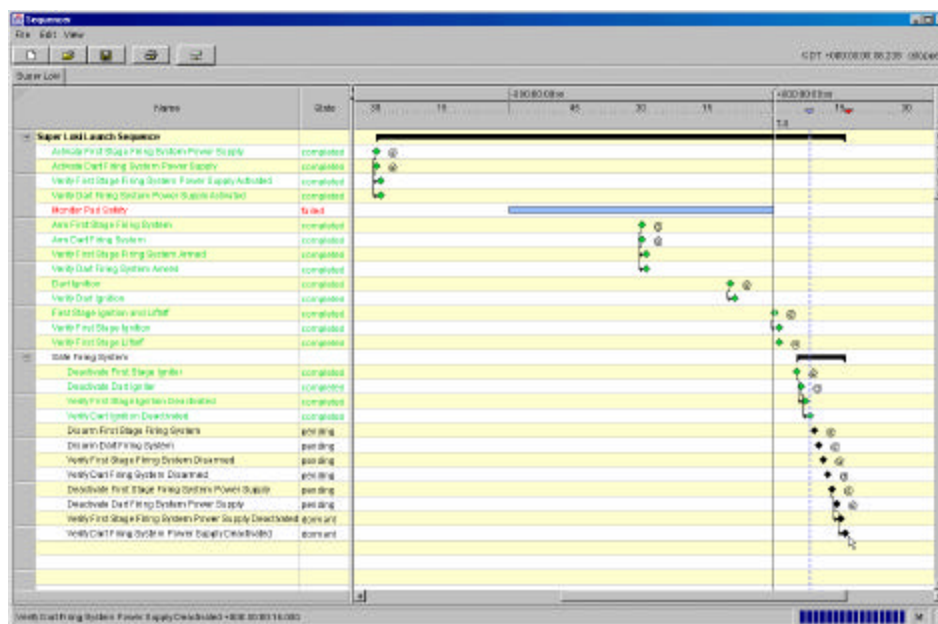


Figure 1: T-Zero Client GUI running Super-Loki Sequence

Figure 2 depicts the integration of T-Zero with the Command and Control Toolkit (CCTK).

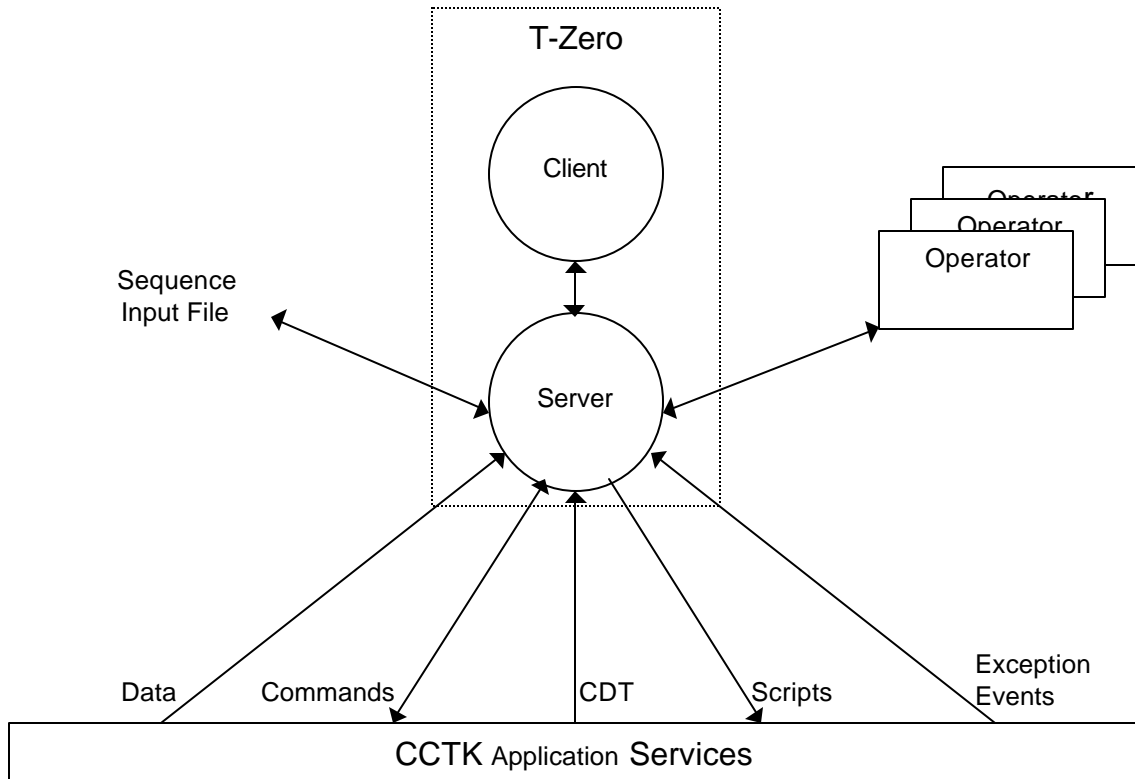


Figure 2: CCTK Integration Concept

The CCTK Application Services layer provides all of the necessary interfaces for T-Zero to both monitor and control a test operation.

- Data – Real-time end-item data updates, health, status, and fused information can all be used for aggregating status and events on the display.
- Commands – Commanding support is provided for both end-item and system maintenance.
- Countdown Time – Continuous time event trigger input.
- Scripts – Execution of scripts or other support programs based on sequence state conditions.

# 3 T-ZERO CLIENT GUI COMPONENTS

## 3.1 Main Window Component

The T-Zero main window (Figure 3) provides the focus for user interaction.

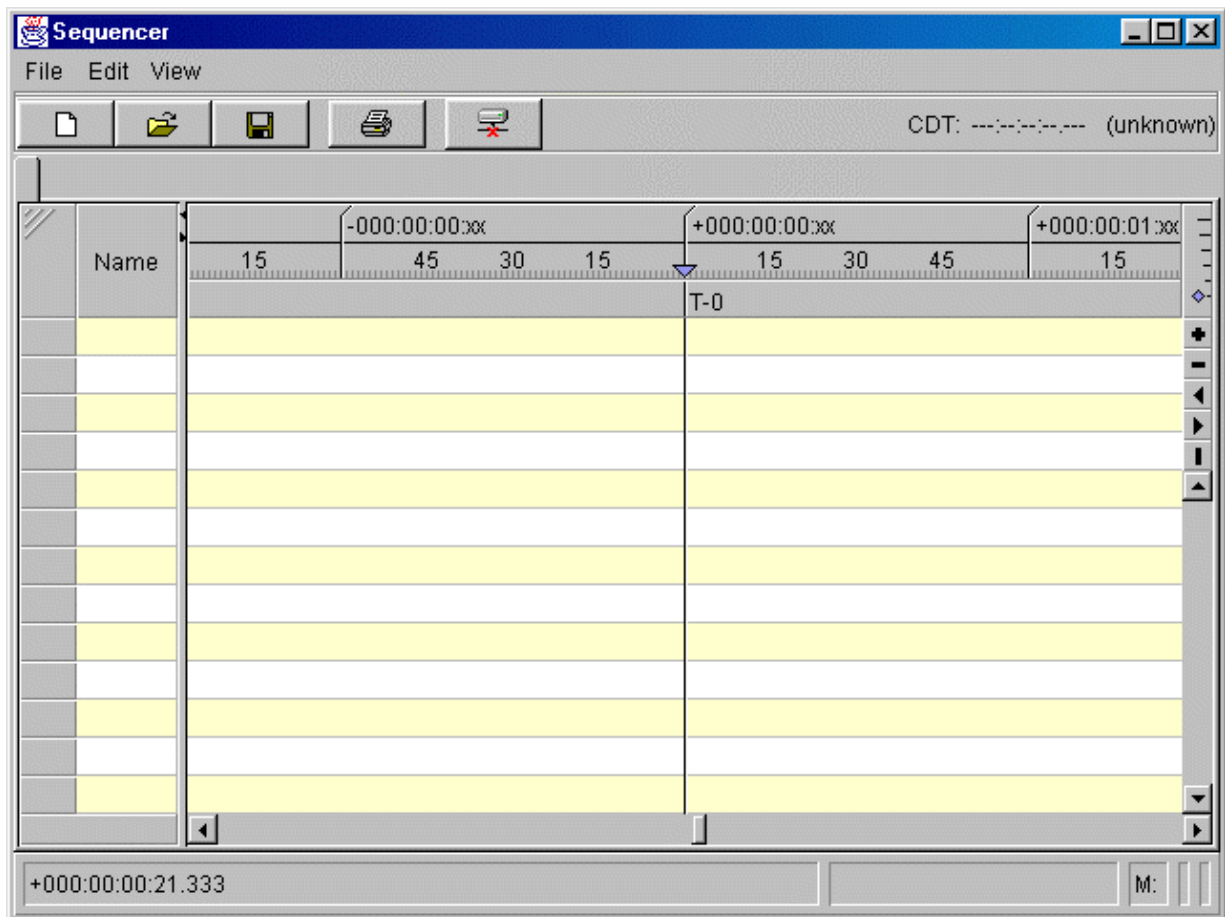


Figure 3: Main Window

---

## 3.2 Menu

This section presents descriptions of all the options on the T-Zero Client GUI's menus:

### *File*

Connect – Connect to a T-Zero Server application specified by the current connection property values.

Disconnect – Disconnect from an existing server connection.

New – Create a new sequence sheet.

Open – Open a sequence input file. Refer to Sequence Input File DTD Specification, and Example Sequence Input File for details on the input file format.

Save and Save As – Save the current sequence to a file.

Print – Print a snapshot of the current sequence view.

Close – Close the current sequence and disconnect from the server if appropriate.

Exit – Exit the T-Zero Client application, and disconnect from the server if appropriate.

### *Edit*

Preferences – Display the connection properties dialog box to modify the T-Zero server connection (network address and port) properties (see Figure 4: Connection Properties Dialog Box).

### *View*

Annotations – When active, this mode will cause event annotations to pop up when the user moves the mouse cursor on top of an entry.

Auto Scroll – When active, this mode will cause the T-Zero Client GUI to automatically scroll the event data window in order to keep the time “now” line visible at all times.

### Expand/Collapse

Expand All and Collapse All – Expand/Collapse the table of events to show/hide the entire event tree hierarchy.

### Relations

Dependencies – When active, this mode will show all event dependency connections on the event data window.

Time – When active, this mode shows time dependencies for each applicable event.

---

## 3.3 Connection Properties

The properties shown in Figure 4 allow you to specify changes in the default connection properties:

- Server Connection Address – Specifies the host name or IP address of the machine running the T-Zero server application.
- Server Connection Port – Specifies the port number that the T-Zero server application listening for client connections.
- Multicast Connection Address – Specifies the IP address used to deliver the sequence data.
- Multicast Connection Port – Specifies the port number the client listens to for sequence data.

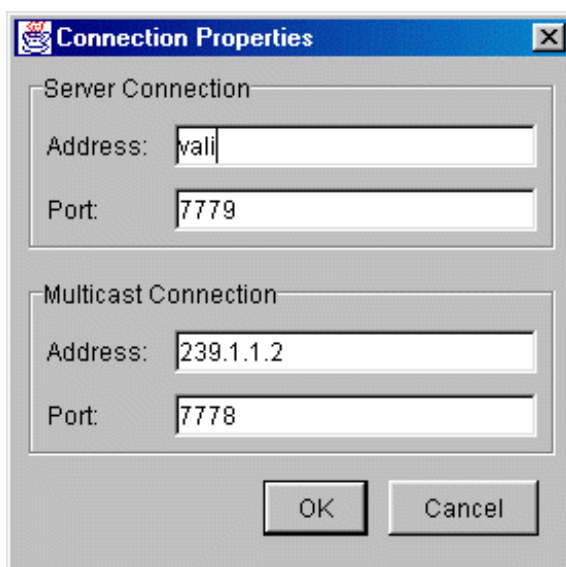


Figure 4: Connection Properties Dialog Box

---






## 3.4 Main Window Toolbar



Figure 5: Toolbar

The buttons on the main window toolbar are described in Table 1 below.

Table 1: Toolbar Icons

Icon	Function	Description
	Open File	Open a valid sequence input file.
	New File	Create a new sequence sheet.
	Save File	Save the current sequence to a file.
	Connect Disconnect	Connect to a T-Zero Server application specified by the current connection property values. Or disconnect from an existing server connection.
	Print	Print a snapshot of the current sequence view.
	CDT	Displays Countdown Time (CDT) and status .

### 3.5 Main Window Controls

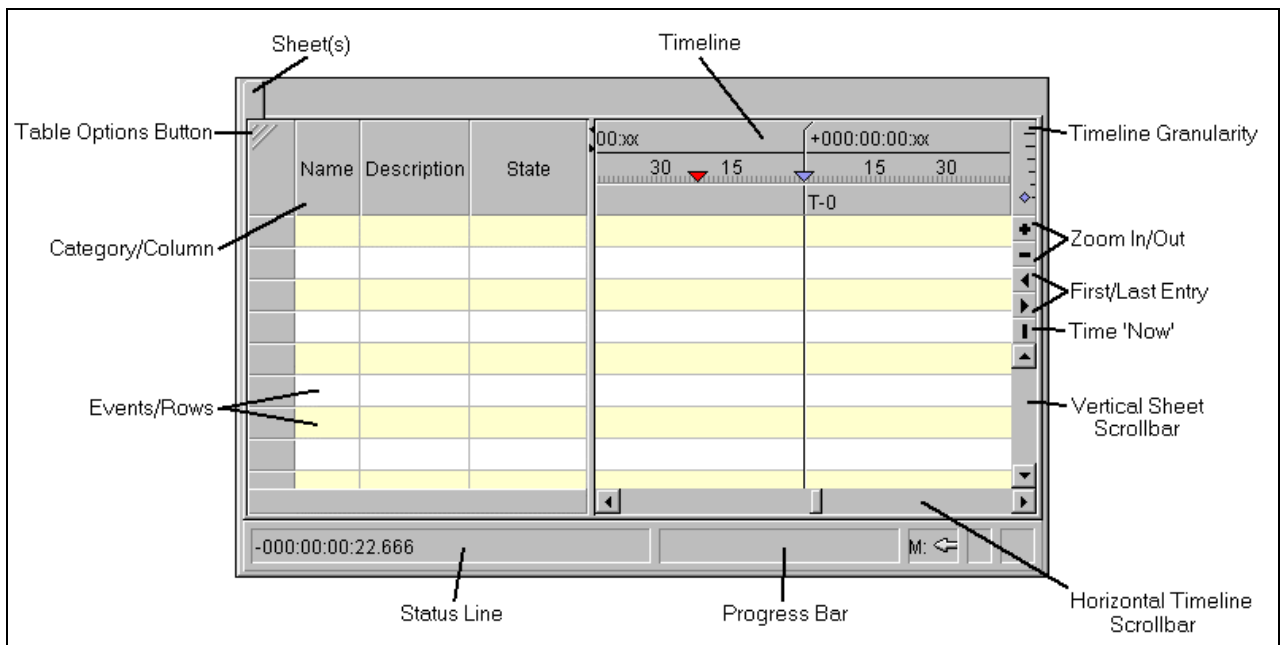


Figure 6: Main Window Controls

### 3.6 Timeline Component

The timeline component is primarily used to correlate events with countdown time on the event data display of the main window. In addition, the operator can interact with the timeline to get location and time value information, as well as scroll the timeline forward and backward in a similar fashion as the horizontal scrollbar. You can invoke an option menu on the timeline by right-clicking on the timeline component. The list on the next page describes the menu options.

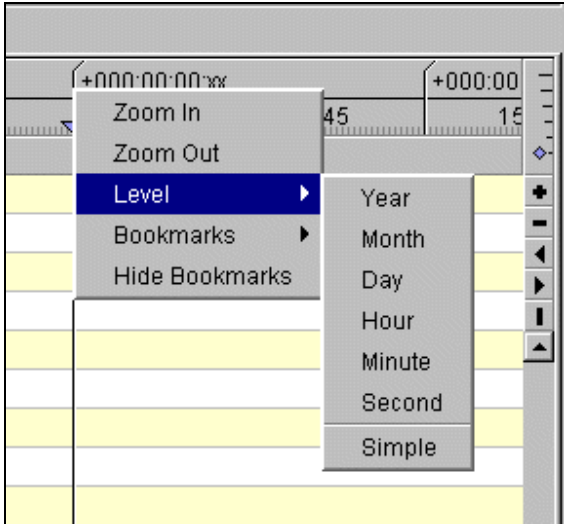


Figure 7: Timeline Level Menu

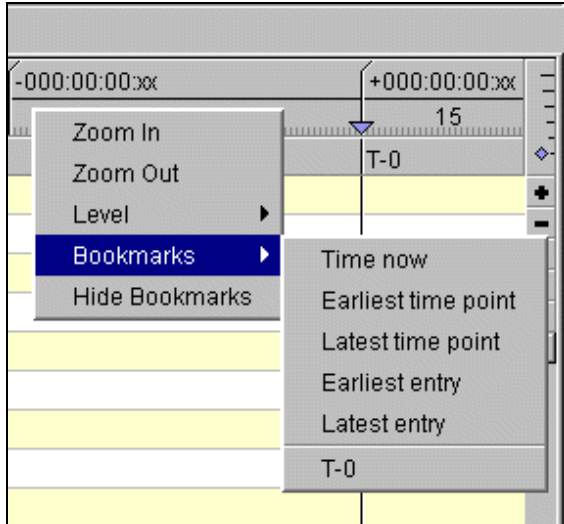


Figure 8: Timeline Bookmarks Menu

### Timeline Menu:

Zoom In, Zoom Out – Zoom in/out off the timeline, steps sequentially through each of the timeline granularities.

Level Submenu (see Figure 7: Timeline Level Menu)

Year, Month, Day, Hour, Minute, Second – Each of the above are timeline view granularities currently supported.

Simple – A simple form where time units are numbered sequentially (1, 2, 3, ...)

Bookmarks Submenu (see Figure 8: Timeline Bookmarks Menu)

*The following is a list of default bookmarks that allow the operator to navigate to predefined locations in the current sequence on the T-Zero Client GUI display.*

Time now – This is the current Countdown Time (CDT) known by the client.

Earliest time point – This is the earliest time point of interest, you can't scroll in front of this position.

Latest time point – This is the latest time you can scroll to.

Earliest entry – This is the first event entry in the sequence, in time sorted order.

Latest entry – This is the last event entry in the sequence, in time sorted order.

T-0 – This is a user defined bookmark for the T-0 Countdown time location.

Hide Bookmarks

Hide/Show all user defined bookmark labels and icons on the timeline control.

## 3.7 Table Categories

By default, the T-Zero Client GUI displays the event name as the only table category. The user can alter this both in the custom preferences (see Default Properties Configuration File)

or by opening the table options dialog box (see Figure 9: Table Options Dialog Box) by selecting the table options button on the main window (see Figure 6: Main Window Controls)

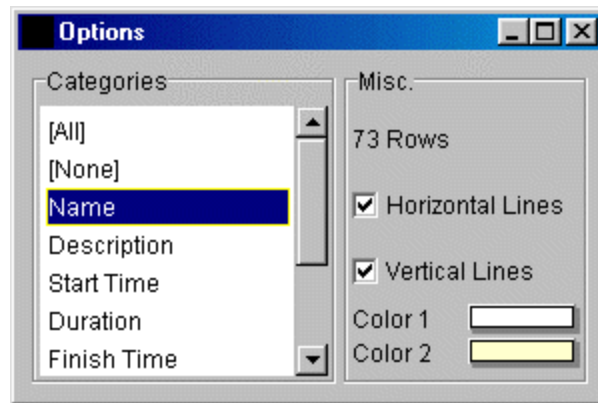


Figure 9: Table Options Dialog Box

The following column categories are available:

- Name
- Description
- Start Time
- Duration
- Finish Time
- Hyperlink
- Resource
- % Complete
- State

## 4 USING T-ZERO

This section describes how to use the T-Zero components to construct, view, and interact with sequences running on the CCTK T-Zero server.

### 4.1 Sequence Construction

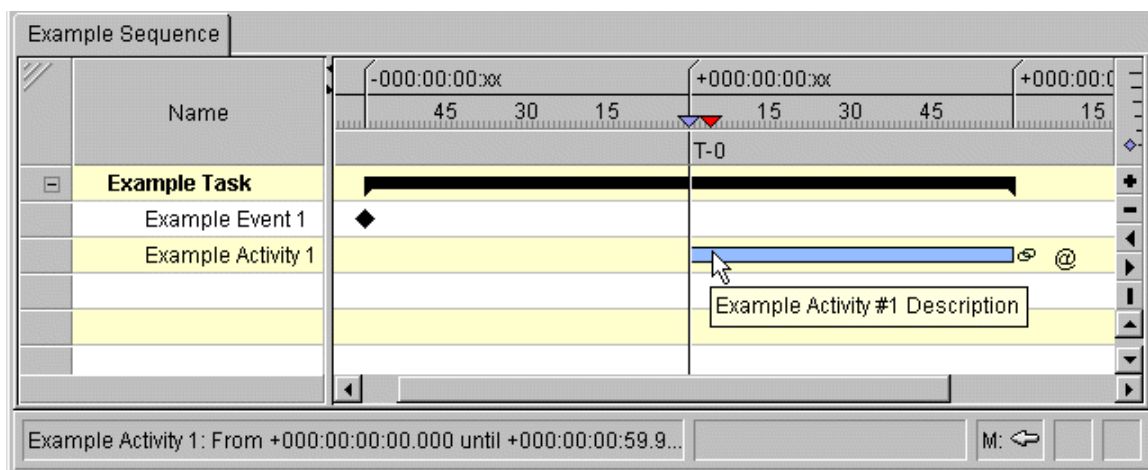


Figure 10: Example Sequence Output

Figure 10 is the resulting output from the input file provided in Example Sequence Input File shown in Appendix C. The sequence input file definition is based on the Extensible Markup Language (XML). The format rules for the sequence input file are defined in the Sequence Input File DTD Specification (shown in Appendix B). The simple example above illustrates the three basic elements of the sequence definition used within T-Zero:

- **Tasks** – Organize events, activities, and subtasks.

For example:

```
<Task name="Example Task" />
```

- **Events** – Represent an instantaneous event at some defined point in time.

For example:

```
<Event name="Example Event 1" start="-000:00:00:59.999" />
```

- **Activities** – Represent an event with an associated duration of time.

For example:

```
<Event name="Example Activity 1" start="+000:00:00:00.000" duration="60000" />
```

Both an element name and time information are required for all events and activities. The following properties can also be associated with each element in a sequence definition:

- *Description* – A more detailed description to accompany the element name.
- *Resource* – A name used to associate ownership or responsibility for the element.
- *Hyperlink* – An HTML document is linked with the element (double-click to follow).
- *Dependency* – An associated predecessor or time dependency for the element.
- *Dialog* – A pop-up dialog request for response associated with the element.
- *Action* – A server side shell activity associated with the element.

---

## 4.2 Sequence Interaction

The T-Zero display provides the operator with up-to-date sequence event information, refer to Table 2: Default State Colors that identify the default color scheme and associated state for events during run-time. Each of these colors can be customized via the configuration file, refer to the section Default Properties Configuration File for information on changing these defaults.

Table 2: Default State Colors

Color	State
BLACK	Inactive
BLUE	Active (Pending)
GREEN	Completed
RED	Failed

Additionally, there are various interactions between the operator and the GUI that may occur during the execution of a sequence. Since all data in T-Zero is based on the CDT clock, managing CDT is a fundamental (but external) operation during operations. While clock management is a function of CCTK, the T-Zero Client GUI is responsible for displaying countdown time and mode information. Valid CDT modes include:

- Cancelled
- Stopped
- Running

Each event or activity within a sequence can have an associated dialog box. The purpose of this pop-up window is to relay some event specific information to the operator in a “request to proceed” fashion. The operator must acknowledge, not acknowledge, or dismiss the dialog before any further action will be taken in regards to that particular element in the sequence.

# Appendix A. DEFAULT PROPERTIES CONFIGURATION FILE

The T-Zero Client GUI uses a collection of settings called properties that allow you to customize T-Zero's features. To view or change these properties, copy or modify the existing defaults file and customize the settings.

To run the T-Zero Client GUI with an alternate properties file, execute the application with the addition of the following command line argument:

```
java ... -properties <filename>
```

If no alternate file is specified then the "defaultProperties" file in the T-Zero installation directory will be used by default. Refer to Default Properties Configuration File for an example of all available properties and their associated defaults.

```
#####  
# T-Zero Client GUI Default Properties, $Revision: 1.4 $  
#####  
#  
# cct.tzero.sequencerui.serverAddress =  
# Default = server  
#  
# cct.tzero.sequencerui.serverPort =  
# Default = 7779  
#  
# cct.tzero.sequencerui.multicastAddress =  
# Default = 239.1.1.2  
# cct.tzero.sequencerui.multicastPort =  
# Default = 7778  
#  
# cct.tzero.sequencerui.startDirectory =  
# Default = .  
#  
# cct.tzero.sequencerui.event.state.default.color =  
# Default = 0, 0, 0 (Black)  
#  
# cct.tzero.sequencerui.event.state.active.color =  
# Default = 0, 0, 255 (Blue)  
#  
# cct.tzero.sequencerui.event.state.completed.color =  
# Default = 0, 255, 0 (Green)
```

```

#
# cct.tzero.sequencerui.event.state.failed.color =
# Default = 255, 0, 0 (Red)
#
# cct.tzero.sequencerui.gantt.annotations =
# Default = off (on | off)
#
# cct.tzero.sequencerui.table.autoExpand =
# Default = off (on | off)
#
# cct.tzero.sequencerui.gantt.relations =
# Default = off (on | off)
#
# cct.tzero.sequencerui.rowColor1 =
# Default = 255, 255, 255 (White)
#
# cct.tzero.sequencerui.rowColor2 =
# Default = 254, 255, 204 (Pastel Yellow)
#
# cct.tzero.sequencerui.table.categories =
# Default = Name
# (Name, Description, Start Time, Duration, Finish Time, Resource, % Complete, State)
#
# cct.tzero.sequencerui.table.horizontalLines =
# Default = on (on | off)
#
# cct.tzero.sequencerui.table.verticalLines =
# Default = on (on | off)
#
# cct.tzero.sequencerui.gantt.zoomLevel =
# Default = Second (Year, Month, Day, Hour, Minute, Second)
#
# cct.tzero.sequencerui.gantt.autoScroll =
# Default = off (on | off)
#
# cct.tzero.sequencerui.gantt.autoScrollThreshold =
# Default = 100 (percent)
#
# cct.tzero.sequencerui.gantt.autoScrollStep =
# Default = 100 (percent)
#
# cct.tzero.sequencerui.autoConnect =
# Default = off (on | off)
#
# cct.tzero.sequencerui.size.width = 1280
# Default = 640 (pixels)
#
# cct.tzero.sequencerui.size.height = 850
# Default = 480 (pixels)
#
# cct.tzero.sequencerui.position.x = 0
# Default = ? (pixels)
#
# cct.tzero.sequencerui.position.y = 150
# Default = ? (pixels)

```

# Appendix B. SEQUENCE INPUT FILE DTD SPECIFICATION

```
<?xml version="1.0">

<!-- $Id: sequence.dtd,v 1.2 2000/09/11 15:54:23 wardj Exp $ -->
<!-- DTD for the CCTK T-Zero Sequencer Application. -->

<!--
  Explanation of events, activities, and tasks:
  - An event is a singular event that happens at some instance of time.
  - An activity is a event or collection of events that have both a start time and a
    duration.
  - A task is an aggregated activity, sum total of all the child events/activities.
-->

<!ELEMENT Sequence (Task*, Event*, CDT?)>
<!ATTLIST Sequence
name CDATA #IMPLIED>

<!-- The "Task" tag can contain both events and subtasks -->

<!ELEMENT Task (Description?, Event*, Task*)>
<!ATTLIST Task
name CDATA #IMPLIED>

<!-- The "Event" tag is used to describe both events and activities -->

<!ELEMENT Event (Description?, Resource?, Hyperlink?, Dependency*, Action?, Dialog?,
  Status?)>
<!ATTLIST Event
name CDATA #IMPLIED
start CDATA #REQUIRED      <!-- +000:00:00:00.000 -->
duration CDATA "0">      <!-- in milliseconds -->

<!ELEMENT Description (#PCDATA)>
<!ELEMENT Resource (#PCDATA)>
<!ELEMENT Hyperlink (#PCDATA)>
```

```

<!-- The "Dependency" tag is used to indicate both time and predecessor dependencies. A
      predecessor dependency is specified in the parsed character data portion of the tag.
      -->

<!ELEMENT Dependency (#PCDATA)>
<!ATTLIST Dependency
time (true | false) "false">

<!ELEMENT Action (#PCDATA)>
<!ATTLIST Action
type (script | other) #REQUIRED>

<!ELEMENT Dialog (#PCDATA)>
<!ATTLIST Dialog
active (true | false) "false"
yesOption CDATA #IMPLIED
noOption CDATA #IMPLIED
cancelOption CDATA #IMPLIED>

<!-- The following elements are dynamically generated during run-time, but may be used
      within the input file to initialize specific values -->

<!ELEMENT Status EMPTY>
<!ATTLIST Status
percent CDATA "0%"
state (dormant | pending | active | completed | failed) #IMPLIED>

<!ELEMENT CDT EMPTY>
<!ATTLIST CDT
value CDATA #REQUIRED
state (invalid | cancelled | stopped | running | unknown) #IMPLIED>

```

# Appendix C. EXAMPLE SEQUENCE INPUT FILE

```
<?xml version="1.0"?>

<Sequence name="Example Sequence">

  <Task name="Example Task">

    <!-- A very simple example -->
    <Event name="Example Event 1" start="-000:00:00:59.999" />

    <!-- A more complicated example using all fields available -->
    <Event name="Example Activity 1" start="+000:00:00:00.000" duration="60000">
      <Description>Example Activity #1 Description</Description>
      <Resource>User 1</Resource>
      <Hyperlink>file://F:/My Documents/ReferenceDoc.html</Hyperlink>
    <Dependency time="true">Example Event 1</Dependency>
    <Dialog yesOption="Ready" noOption="Not Ready" cancelOption="Ignore">
    Are you ready?
    </Dialog>
    <Action type="script">
    exec CCTKsh &lt;&lt;EOF
    set timeout 30
    for {set i 0} {\$i &lt; \$timeout} {incr i} {
    if [[:cctk::GetMeasValue NLM_UINT_014] > 25] {
    exit 0
    } else {
    after 1000
    }
    }
    exit 1
    #EOF
    </Action>
  </Event>

</Task>

</Sequence>
```

# Appendix D. COMMAND LINE OPTIONS

```
SequencerEngine [ options ] filename
```

Options:

```
[ -c des_name ]  
[ -i channel_name ]  
[ -k kpath ]  
[ -m address ]  
[ -p port ]  
[ -s des_name ]  
[ -S port ]  
[ -u des_name ]
```

`-c des_name`

The application may be configured to use a different descriptor for Countdown Time (CDT).

(Default: `CURRENT_CDTIME_TIME_VAL_FD`)

`-i channel_name`

The application may be configured to use a specific input channel for receiving data.

(Default: `NEXT_AVAIL_CHAN`)

`-k kpath`

This option is used to specify an explicit path to be used as the CCTK `KPATH` for this process. The default path is taken from the `$KPATH` environment variable.

`-m address`

This option is used to specify an explicit address to be used for client multicast data connections.

(Default: 239.1.1.2)

-p port

This option is used to specify an explicit port to be used for client multicast data connections.

(Default: 7778)

-s des\_name

The application may be configured to use a different descriptor for Countdown Time (CDT) Status.

(Default: CURRENT\_CDTIME\_STATUS\_FD)

-S port

This option is used to specify an explicit port to be used for server socket connections.

(Default: 7779)

-u des\_name

The application may be configured with a descriptor used for receiving sequence specific update packets during run-time.

# GLOSSARY

---

The following terms and abbreviations are used throughout the CCTK documentation set:

**Actor** – Command and measurement models used in CCTK simulation.

**Actor groups** – Collections of actors.

**Analog measurement** – Measurement data that is represented as a continuously variable quantity.

**ANSI** – American National Standards Institute. See <http://www.ansi.org/>

**API** – See Application Programming Interface.

**Application programming interface** – A library of functions and associated data structures that allows application software to link with CCTK.

**ARS** – Archive and Retrieval Subsystem.

**BDT** – Bus Descriptor Table.

**CCB** – See Channel control block.

**CCT** – Command and Control Technologies Corporation.

**CCT\_HOME** – Environment variable that specifies the installation directory of CCTK.

**CCTK** – Command and Control Toolkit.™

**CCTK Client** – The general reference to the program *CctkClient*, which is the main user interface for operating CCTK.

**CCTKPROJECTLABEL** – An environment variable that specifies an alternate term for a CCTK project.

**CDB** – See Configuration Database.

**CDT** – See Command Descriptor Table.

**Channel** – A channel is a means of interprocess communications. There are two types of channels, private and public. A private channel is an exclusive point-to-point connection between two processes. Private channels do not go through a distributor. A public channel is one to which any process can write.

**Channel control block** – An area of memory contained within the channel shared memory block that stores state information related to the channels configured in a particular instance of CCTK.

**Client** – Any process that requires services from another process within CCTK.

**Command** – A command represents a control input to an end item or operational system. The characteristics of a command are defined in command descriptions.

**Command Descriptor Table** – Contains information, including name and attributes, associated with system commands.

**Command Response** – The data sent in response to a command.

**Condition Notice** – A condition notice is the set of static attributes of a system message defined in the configuration database. Condition notices do not “occur.” There is no temporal aspect associated with a condition notice and they are not passed around the system. There are a set number of condition notices defined for each project. A condition notice is analogous to a “class” and a system message is analogous to an “object” of that class. When a system message is generated, an instance of a condition notice is instantiated in the system.

**Configuration Database** – Persistent configuration repository for all measurement parameters, command parameters, and interface specifications within CCTK.

**Configuration Descriptor** – A configuration record or object.

**Configuration Item** – A configuration item is a hardware or software entity, or an aggregation of both, which is designated for configuration management. Each configuration item has a unique set of function and/or physical attributes.

**DAC** – Data acquisition; see PCM data acquisition application.

**Data Compression** – Filtering of data to reduce bandwidth requirements.

**Data Conversion** – Linearization and end conversion of data.

**Data Retrieval Request** – Request for near real-time or permanent archived data.

**Descriptor** – A single named object within CCTK.

**Discrete Measurement** – Measurement data that is represented as one of two discrete states, such as ON or OFF.

**Display Monitoring** – Display monitoring is a mechanism by which a user can select and view the current value(s) of one or more measurements.

**Distributor** – A process that routes data through the Kernel and to external interfaces.

**DMON** – See Display Monitoring.

**DTD** – Document Type Definition, which is an XML construct that defines a document structure with a list of XML elements. See <http://www.xml101.com/> for more information.

**DTDPATH** – An environment variable that specifies the location of CCTK DTD files.

**Dynamic Displays** – Dynamic Displays associate attributes (e.g. color, blink, position, etc.) with screen objects that can, in turn, be associated with dynamic values (e.g. measurement values).

**EIA** – Electronic Industries Alliance.

**End Item** – Application device or system under the control of CCTK. The end item is a source of raw measurement data, recipient of commands, and generator of command responses.

**EOF** – End of file.

**Exception Condition** – A condition created when a measurement satisfies an event against boundary values, expected values, and masks.

**Exception Event** – The transition between states (positive and negative) that occurs when an expression is evaluated.

**Exception Expression** – An algorithm that determines the states associated with measurement values. An expression may be atomic or compound formulas consisting of operators and conjunctions.

**Exception Monitoring** – Generic, system provided application for monitoring exception conditions. Exception monitoring presents exception messages to the user upon detection of exception conditions detailing the conditions of the exception.

**Exception Notice** – An asynchronous notification of an event provided to an application.

**Exception State** – Persistent representation of the last exception event associated with a single measurement.

**FD** – See Function Designator.

**Fieldbus** – An open standard for industrial control I/O buses defined by ANSI/ISA-50.02, Part 2-1992.

**FIFO** – First in, first out.

**FFI** – See Frame Format Identifier.

**Frame Format Identifier** – An identification number within a PCM frame used to uniquely identify the contents of a frame.

**Front End** – Communication interfaces that connect end items to CCTK.

**FSP** – Frame Sync Pattern.

**Function Designator** – A unique symbolic name representing related data within CCTK. Function designators are defined by CCTK users and are usually associated with sensors and effectors whose value is resolved by a CCTK external interface or system application during project operations. The FD is the principal handle for data and command access within CCTK. FD's are also associated with CCTK information such as condition notices and processing chains.

**GLG** – Genlogic; GLG is the graphic builder and display creation tool from Genlogic Corp.

**GMT** – Greenwich Mean Time.

**GPS** – Global Positioning System.

**Graphical User Interface** – The interface through which clients access system functionality. This interface includes software elements (i.e. displays, controls) as well as hardware elements (i.e. workstations, input devices). Graphical and textual screen formats used to display data and accept client input.

**GUI** – See Graphical User Interface.

**HCI** – Human-computer interface; see Graphical User Interface.

**Health and Status Data** – Information about CCTK kernel processes that indicate of the state of the system.

**HTTP** – Hypertext transfer protocol.

**Human-Computer Interface** – See Graphical User Interface.

**IBS** – Interbus; IBS is a SCADA bus protocol from Phoenix Contact Corp.

**Interlock Processing** – The processing of a set of conditions upon which command issuance is contingent. Also referred to as “Prerequisite Control Logic.”

**IP** – Internet protocol.

**IPC** – Inter-process Communication. See UNIX system call documentation.

**IRIX** – The UNIX-based operating system produced by Silicon Graphics.

**ISA** – Instrumentation, Systems and Automation Society. See <http://www.isa.org/>.

**Kernel** – The core real-time services of CCTK. The kernel provides services to applications and end item interfaces for data processing, data and command distribution, and time management.

**Kernel Programming Interface** – An application programming interface consisting of functions provided directly by the CCTK kernel.

**KMSG** – Directory containing message queue information.

**KPATH** – An environment variable that points to the current project directory. *KPATH* is used by every CCTK process to locate the resources associated with a particular project. *KPATH* typically needs to be set if the user is going to be performing operations from the UNIX command line.

**KPI** – See Kernel programming interface.

**KSHM** – An environment variable that indicates the directory containing information on the shared memory segments created for a CCTK project. The name of the each file in the *KSHM* directory equates to the name of the shared memory segment. The contents of the file contains the shared memory key associated with the segment. Using this key, it is possible to associate the shared memory segments listed by the UNIX *ipcs* command with those used by a CCTK project.

**LDT** – See Link Descriptor Table.

**Link Descriptor Table** – An internal CCTK table that contains information associated with external commands and link records which link a measurement with an external interface.

**Linked Data** – A single raw measurement sample that has been acquired, time stamped, and associated with its FD handle, but not yet processed.

**LSB** – Least significant bit.

**MDI** – Multiple document interface.

**Measurement** – A measurement is a unit of information that is received from an end item or application. The characteristics of a measurement are defined in the CCTK configuration database. Measurements can be represented by an analog, discrete, integer, or byte array value.

**Minor Frame** – A minor frame is a constant sized sequence of data including a synchronization pattern for hardware identification and optionally a frame ID for software identification to uniquely identify the frame. A major frame consists of one or more minor frames.

**MSB** – Most significant bit.

**NACK** – Negative acknowledge.

**NDT** – Notice Descriptor Table.

**NRZ** – Non-return to zero.

**NTP** – Network Time Protocol.

**Operator** – A person concerned with the operation and/or administration of a control system associated with CCTK; often used interchangeably with “user.”

**Packet** – Any set of data associated with a packet header allowing it to be communicated through the Kernel. A packet may be the transport structure for a processed data measurement, a message, a command, a configuration descriptor, or any other type of data that circulates through the system.

**PAM** – Permanent Archive Media.

**PCI** – Personal Computer Interface. The electronic interface between Intel-class processors and peripherals.

**PCM** – Pulse code modulation.

**PCM Data Acquisition Application (DAC)** – The CCTK software program that processes PCM telemetry data.

**PCMPATH** – An environment variable that indicates the location of the PCM Telemetry Interface software. The PCMPATH environment variable is used by the installation script and the user environment scripts.

**PDI** – Payload data interleaver (a data format associated with the Space Shuttle).

**Permanent Archive Media** – Long-term storage for recorded data. Each PAM has a type of media and a volume. A volume is a single instance of a type of media. It may also be referred to as historical archive media. The media may be any standard storage device format, e.g., DAT or CD.

**Point-to-Point Channel** – A channel that is directly connected to two processes without the need for a distributor for routing the data.

**POSIX** – Portable operating standard for Unix. See *POSIX.4 Programming for the Real World*, O’Reilly & Associates, Sebastopol, CA, 1995.

**Pseudo Function Designator** – A special type of Function Designator that can be used for storage of data and to aid in communication. Typically used to designate derived data.

**Qt** – A Trolltech product that supports portable user interface applications; see <http://doc.trolltech.com/>.

**Raw Frame** – A block of data that is acquired from the end item.

**Raw Measurements** – Data that has not been linearized or scaled (counts).

**Real-Time Control Server** – The computer that hosts the CCTK server software.

**Recorded Data** – All measurement, command, event, message and other data that are recorded on temporary and permanent archive media for retrieval and Analysis. This includes End item Data and Configuration data.

**RT** – Real time.

**RTCS** – See Real-time control server.

**RTTM** – Real-time table manager.

**SCADA** – Supervisory control and data acquisition.

**SFID** – See Sub-frame identification.

**SGI** – Silicon Graphics, Inc.

**SID** – See Software Identifier.

**SIGCHANSIGNAL** – The CCTK signal used to indicate a packet has been placed on a channel when operating a channel asynchronously.

**SIGCHANWAKEUP** – An internal channel notification signal.

**Simulated Universal Time** – A simulated clock used by the CCTK simulation function.

**Software Identifier (SID)** – Internal CCTK index key for access to command measurement information. The SID is generated at configuration build time. There is one unique SID for every Function Designator (FD).

**SQL** – Standard Query Language.

**Stale Data** – Data that has not changed in value for a minimum of a user specified number of data samples.

**STDARCH** – The standard CCTK channel for information archival. All data sent to the STDARCH channel will be archived by the ARS.

**STDDIST** – Legacy CCTK channel that is no longer used.

**STDMSG** – The standard CCTK channel for system messages. All messages sent to the STDMSG channel will be processed and forwarded to interested processes, the system message file, and the archive subsystem.

**STDRESP** – Standard CCTK channel for command responses.

**STDSTATUS** – The standard CCTK channel for status information.

**Sub-frame identification** – A cyclic sequence number contained within a telemetry minor frame used by the decommutator hardware and/or software to verify major frame lock.

**Super packet** – A packet which contains multiple subpackets. Superpackets are used to increase efficiency by reducing overhead when transmitting data.

**SUTC** – See Simulated Universal Time.

**System Message** – An asynchronous event that notifies the user some action has occurred. System messages have a set of static attributes defined in the configuration database and a set of dynamic attributes generated at instantiation.

**TAM** – See Temporary archive media.

**Temporary Archive** – An archive that stores data in the local file system before being written to permanent archive.

**Temporary Archive Media** – Archive media represented by a standard UNIX file where data is stored before being written to permanent media.

**TCL** – Tool Command Language, a freely available platform-independent, string-based, interpreted command language. See <http://www.neosoft.com/tcl> and <http://hegel.ittc.ukans.edu/topics/tcltk/index.html> for more information.

**Telemetry Data Stream** – A continuous serial bit stream of time division multiplexed data.

**TIX** – Tk interface extension. Tix is an open open source extended widget set for Tcl/Tk. See <http://tix.mne.com/> and <http://tix.sourceforge.net/> more information.

**UTC** – Universal Time, Coordinated. See <http://www.time.gov/timezone.cgi?UTC/s/0/java>.

**Valid** – Raw and processed data values lie within a defined, valid range.

**X Protocol** – X Window System technology provides display and management of graphical information for UNIX-based computers. The underlying protocol is defined by X.org; see <http://www.x.org>.

**XCDB** – XML Configuration Database – See Configuration Database.

**XIO** – Extended input/out.

**XML** – Extensible Markup Language; a self-describing markup language defined by the World Wide Web Consortium designed to describe data. XML is an open technology; see <http://www.w3.org/XML/>.

# INDEX

<b>A</b>	
<b>activities</b>	
definition .....	13
<b>B</b>	
bookmarks .....	10
<b>C</b>	
CCTK .....	2, 3, 5, 12, 13, 16, 19
CDT .....	<i>See</i> countdown time
client .....	2, 4, 8, 10, 20
clock .....	<i>See</i> countdown time
colors .....	13
columns	
labels .....	10
command line options .....	19
countdown time .....	5, 9, 10, 13, 16, 17, 19, 20
customize .....	14
<b>E</b>	
event	
dialogue box .....	13
<b>events</b>	
definition .....	12
<b>H</b>	
health .....	5
hierarchy .....	7
<i>Hyperlink</i> .....	13
<b>I</b>	
IP 8	
port .....	8

<b>J</b>	
Java .....	4
<b>M</b>	
multicast .....	8
<b>R</b>	
Real-time .....	5
relationships	
dependencies .....	7
<b>S</b>	
scripts .....	5
segment .....	24
sequence	
input file DTD .....	16
input file example .....	18
server .....	2, 4, 7, 8, 9, 12, 13, 14, 20
shared memory segments .....	24
status .....	2, 5, 9
<b>T</b>	
<b>tasks</b>	
definition .....	12
timeline	
entry point .....	10
example .....	12
level .....	10
T-0 .....	10
<b>X</b>	
XML .....	12