MTS TestSuite™

TW Elite™ Test Design Guide for Criterion™ Systems
MTS Trademarks
MTS, be certain., Bionix, Echo, ElastomerExpress, FlatTrac, FlexTest, Just In Case, Landmark, Level Plus, MTS Acumen, MTS Criterion, MTS Echo, MTS EM Extend, MTS Exceed, MTS Insight, MTS Landmark, MTS TestSuite, RPC, SWIFT, Temposonics, TestWare, TestWorks are registered trademarks of MTS Systems Corporation within the United States. Acumen, AdapTrac, Advantage, Aero ST, Aero-90, AeroPro, Criterion, cRPC, Exceed, First Road, Landmark, MAST, MicroProfiler, MPT, MTS Exceed, MTS Fundamentals, MTS TestSuite, ReNew, SilentFlo, TempoGuard, TestLine, Tytron, Virtual Test Lab, and VTL are trademarks of MTS Systems Corporation within the United States. These trademarks may be registered in other countries.
All other trademarks are the property of their respective holders.

Proprietary Software
Software use and license is governed by MTS’ End User License Agreement which defines all rights retained by MTS and granted to the End User. All Software is proprietary, confidential, and owned by MTS Systems Corporation and cannot be copied, reproduced, disassembled, decompiled, reverse engineered, or distributed without express written consent of MTS.

Software Verification and Validation
MTS software is developed using established quality practices in accordance with the requirements detailed in the ISO 9001 standards. Because MTS-authored software is delivered in binary format, it is not user accessible. This software will not change over time. Many releases are written to be backwards compatible, creating another form of verification. The status and validity of MTS’ operating software is also checked during system verification and routine calibration of MTS hardware. These controlled calibration processes compare the final test results after statistical analysis against the predicted response of the calibration standards. With these established methods, MTS assures its customers that MTS products meet MTS’ exacting quality standards when initially installed and will continue to perform as intended over time.

<table>
<thead>
<tr>
<th>Manual Part Number</th>
<th>Publication Date</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-256-490 G</td>
<td>August 2015</td>
<td>MTS TestSuite TW 4.1 or later</td>
</tr>
<tr>
<td>100-256-490 F</td>
<td>October 2014</td>
<td>MTS TestSuite TW 3.0</td>
</tr>
<tr>
<td>100-256-490 E</td>
<td>February 2014</td>
<td>MTS TestSuite TW 2.3.4</td>
</tr>
<tr>
<td>100-256-490 D</td>
<td>September 2013</td>
<td>MTS TestSuite TW 2.3.1</td>
</tr>
<tr>
<td>100-256-490 C</td>
<td>August 2013</td>
<td>MTS TestSuite TW 2.3</td>
</tr>
</tbody>
</table>
# Contents

## Technical Support

How to Get Technical Support ................................................................. 7
   Start with your manuals ..................................................................... 7
   Technical support methods .................................................................. 7
   Outside the U.S. .................................................................................. 7

Before You Contact MTS ....................................................................... 7
   Know your site number and system number ....................................... 7
   Know information from prior technical assistance ............................. 8
   Identify the problem ............................................................................ 8
   Know relevant computer information ............................................... 8
   Know relevant software information ............................................... 8

If You Contact MTS by Phone ............................................................... 9
   Identify system type ........................................................................... 9
   Be prepared to troubleshoot ............................................................... 9
   Write down relevant information ....................................................... 9
   After you call ..................................................................................... 10

Problem Submittal Form ........................................................................ 10

## Preface

Before You Begin ................................................................................... 11
   Safety first! ......................................................................................... 11
   Other MTS manuals .......................................................................... 11

Documentation Conventions ................................................................. 11
   Hazard conventions .......................................................................... 11
   Other special text conventions ......................................................... 12
   Special terms .................................................................................... 12
   Illustrations ...................................................................................... 12
   Electronic manual conventions ......................................................... 12
   Hypertext links .................................................................................. 12

## Introduction

About this Guide .................................................................................. 14
   Summary .......................................................................................... 14
   Information pertaining to TW Express .............................................. 14

Start TWE and Open the Example Test .............................................. 14
   Install TWE in simulation .................................................................. 14
   Open the Example EM Tension (Simplified) Test .............................. 14

TWE Main Window ............................................................................... 15
   Main Window Toolbar ....................................................................... 17
   About the Multi-Head Run Section .................................................... 19
## Contents

Multi-Head Run Section Properties .................................................. 20

### Examine the Example Test .......................................................... 23

#### About This Chapter ................................................................. 24

#### Design Flow ............................................................................ 24
  - Select tab ............................................................................. 24
  - Define tab ............................................................................. 24
  - Monitor tab ........................................................................... 25
  - Review tab ............................................................................. 25
  - Runtime Values tab ................................................................. 25

#### Define Tab ................................................................................ 25
  - Outline View in the Procedure Subtab ..................................... 25
  - Workflow in the Procedure Subtab .......................................... 26
  - Test-Run Display Subtab ....................................................... 26
  - Variables Subtab ..................................................................... 36
  - Report Templates Subtab ....................................................... 37
  - Functions Subtab ..................................................................... 40
  - Resources Subtab ..................................................................... 41
  - Test Definition Subtab ........................................................... 44

#### Monitor Tab .............................................................................. 45
  - Monitor Tab View Before the Test Starts ............................... 45
  - Test Opens to Monitor Tab for Operators ............................... 45
  - Editing Pretest Inputs on the Variables Subtab ...................... 45
  - Monitor Tab View After the Test Starts .................................. 46

#### Review Tab ............................................................................... 47
  - Review Tab Toolbar ............................................................... 48
  - Test Run Results Table .......................................................... 49
  - Statistics Table ...................................................................... 50
  - Panel of Charts and Tables .................................................... 50
  - Four-Panel View ..................................................................... 52

#### Runtime Values Tab .................................................................. 53

### Design Guidelines ..................................................................... 55

#### About This Chapter ................................................................. 56

#### Designing Tests ...................................................................... 56
  - Test Design Checklist ............................................................ 56
  - Tests and Templates ............................................................... 57
  - Modifying Tests ..................................................................... 60
  - Working with Variables .......................................................... 66
  - Configuring Data Acquisition ................................................ 72
Contents

Polarity Considerations for MTS Criterion Series 40 Systems ........................................ 74
Debugging ................................................................................................. 77

Modify the Example Test ........................................................................ 79

About This Chapter .............................................................................. 80
Prerequisites ......................................................................................... 80
Practice in Simulation .......................................................................... 80

Modifying Tests .................................................................................... 80
Methods Used to Modify Tests .............................................................. 80
Methods Used to Modify the Example Test .............................................. 81

Design Scenario .................................................................................... 81
Modifying the Set Up Section ................................................................. 81
Modifying the Run Section .................................................................... 84
Modifying the Finish Section ................................................................. 86

TW Elite for TestWorks 4 Users .............................................................. 89

About This Chapter .............................................................................. 90
Basic Concepts ..................................................................................... 90
Terminology Differences ...................................................................... 90
Functional Differences .......................................................................... 94
Converting TestWorks 4 Methods and Sample Files ............................... 97

Appendix ............................................................................................... 99

NI M Series Multifunction DAQ ............................................................. 100
Configuration ......................................................................................... 100

Index .................................................................................................... 101
Technical Support

How to Get Technical Support

Start with your manuals
The manuals supplied by MTS provide most of the information you need to use and maintain your equipment. If your equipment includes software, look for online help and README files that contain additional product information.

Technical support methods
MTS provides a full range of support services after your system is installed. If you have any questions about a system or product, contact Technical Support in one of the following ways.

<table>
<thead>
<tr>
<th>Type of Support</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web site</td>
<td><a href="#">www.mts.com &gt; Contact Us &gt; In the Subject field, choose To escalate a problem; Problem Submittal Form</a></td>
</tr>
</tbody>
</table>
| E-mail          | Worldwide: tech.support@mts.com  
|                 | Europe: techsupport.europe@mts.com |
| Telephone       | Worldwide: 1 800 328 2255 - toll free in U.S.; +1 952 937 4000 - outside U.S.  
|                 | Europe: +800 81002 222, International toll free in Europe |

Outside the U.S.
For technical support outside the United States, contact your local sales and service office. For a list of worldwide sales and service locations and contact information, use the Global MTS link at the MTS web site:

[www.mts.com > About MTS Systems > Global Presence > Choose a Region](#)

Before You Contact MTS
MTS can help you more efficiently if you have the following information available when you contact us for support.

Know your site number and system number
The site number contains your company number and identifies your equipment type (such as material testing or simulation). The number is typically written on a label on your equipment before the system leaves MTS. If you do not know your MTS site number, contact your sales engineer.

Example site number: 571167
When you have more than one MTS system, the system job number identifies your system. You can find your job number in your order paperwork.

Example system number: US1.42460

**Know information from prior technical assistance**

If you have contacted MTS about this problem before, we can recall your file based on the:

- MTS case number
- Name of the person who helped you

**Identify the problem**

Describe the problem and know the answers to the following questions:

- How long and how often has the problem occurred?
- Can you reproduce the problem?
- Were any hardware or software changes made to the system before the problem started?
- What are the equipment model numbers?
- What is the controller model (if applicable)?
- What is the system configuration?

**Know relevant computer information**

For a computer problem, have the following information available:

- Manufacturer’s name and model number
- Operating software type and service patch information
- Amount of system memory
- Amount of free space on the hard drive where the application resides
- Current status of hard-drive fragmentation
- Connection status to a corporate network

**Know relevant software information**

For software application problems, have the following information available:

- The software application’s name, version number, build number, and (if available) software patch number. This information can typically be found in the About selection in the Help menu.
- The names of other applications on your computer, such as:
  - Anti-virus software
  - Screen savers
  - Keyboard enhancers
  - Print spoolers
  - Messaging applications
If You Contact MTS by Phone

A Call Center agent registers your call before connecting you with a technical support specialist. The agent asks you for your:

- Site number
- Email address
- Name
- Company name
- Company address
- Phone number where you can be reached

If your issue has a case number, please provide that number. A new issue will be assigned a unique case number.

Identify system type

To enable the Call Center agent to connect you with the most qualified technical support specialist available, identify your system as one of the following types:

- Electrodynamic material test system
- Electromechanical material test system
- Hydromechanical material test system
- Vehicle test system
- Vehicle component test system
- Aero test system

Be prepared to troubleshoot

Prepare to perform troubleshooting while on the phone:

- Call from a telephone close to the system so that you can implement suggestions made over the phone.
- Have the original operating and application software media available.
- If you are not familiar with all aspects of the equipment operation, have an experienced user nearby to assist you.

Write down relevant information

In case Technical Support must call you:

- Verify the case number.
- Record the name of the person who helped you.
- Write down any specific instructions.
After you call
MTS logs and tracks all calls to ensure that you receive assistance for your problem or request. If you have questions about the status of your problem or have additional information to report, please contact Technical Support again and provide your original case number.

Problem Submittal Form
Use the Problem Submittal Form to communicate problems with your software, hardware, manuals, or service that are not resolved to your satisfaction through the technical support process. The form includes check boxes that allow you to indicate the urgency of your problem and your expectation of an acceptable response time. We guarantee a timely response—your feedback is important to us.

You can access the Problem Submittal Form at www.mts.com > Contact Us (upper-right corner) > In the Subject field, choose To escalate a problem; Problem Submittal Form
Preface

Before You Begin

Safety first!
Before you use your MTS product or system, read and understand the safety information provided with your system. Improper installation, operation, or maintenance can result in hazardous conditions that can cause severe personal injury or death, or damage to your equipment and specimen. Again, read and understand the safety information provided with your system before you continue. It is very important that you remain aware of hazards that apply to your system.

Other MTS manuals
In addition to this manual, you may receive additional manuals in paper or electronic form.

You may also receive an MTS System Documentation CD. It contains an electronic copy of the manuals that pertain to your test system.

Controller and application software manuals are typically included on the software CD distribution disc(s).

Documentation Conventions
The following paragraphs describe some of the conventions that are used in your MTS manuals.

Hazard conventions
Hazard notices may be embedded in this manual. These notices contain safety information that is specific to the activity to be performed. Hazard notices immediately precede the step or procedure that may lead to an associated hazard. Read all hazard notices carefully and follow all directions and recommendations. Three different levels of hazard notices may appear in your manuals. Following are examples of all three levels. (For general safety information, see the safety information provided with your system.)

⚠️ Danger: Danger notices indicate the presence of a hazard with a high level of risk which, if ignored, will result in death, severe personal injury, or substantial property damage.

⚠️ Warning: Warning notices indicate the presence of a hazard with a medium level of risk which, if ignored, can result in death, severe personal injury, or substantial property damage.

⚠️ Caution: Caution notices indicate the presence of a hazard with a low level of risk which, if ignored, could cause moderate or minor personal injury or equipment damage, or could endanger test integrity.
Other special text conventions

- **Important:**
  Important notices provide information about your system that is essential to its proper function. While not safety-related, if the important information is ignored, test results may not be reliable, or your system may not operate properly.

- **Note:**
  Notes provide additional information about operating your system or highlight easily overlooked information.

- **Recommended:**
  Recommended notes provide a suggested way to accomplish a task based on what MTS has found to be most effective.

- **Tip:**
  Tips provide helpful information or a hint about how to most efficiently accomplish a task.

- **Access:**
  Access provides the route you should follow to a referenced item in the software.

**Example:** Examples show specific scenarios relating to your product and appear with a shaded background.

Special terms
The first occurrence of special terms is shown in italics.

Illustrations
Illustrations appear in this manual to clarify text. They are examples only and do not necessarily represent your actual system configuration, test application, or software.

Electronic manual conventions
This manual is available as an electronic document in the Portable Document File (PDF) format. It can be viewed on any computer that has Adobe Acrobat Reader installed.

Hypertext links
The electronic document has many hypertext links displayed in a blue font. All blue words in the body text, along with all contents entries and index page numbers, are hypertext links. When you click a hypertext link, the application jumps to the corresponding topic.
Introduction

About this Guide ................................................................. 14
Start TWE and Open the Example Test .................................. 14
TWE Main Window ............................................................... 15
About this Guide
The MTS TestSuite TW Elite Test Design Guide contains the information you need to begin customizing templates with the TW Elite (TWE) application. It includes the fundamental concepts of using the TWE application with MTS control software to create and run tests.

This guide introduces the most important features of the application in the context of customizing tests and does not explain every application feature.

Summary
Introduction: provides an overview of the TWE interface. In this section, you will open the example test included with the MTS TWE application. The example test was created to run in simulation and includes test run data. This section describes the basic parts of the interface with the example test loaded.

Examine the Example Test: examines the example test supplied with the system in detail. Understanding the example test reveals application capabilities and design principles you can apply to your own tests.

Design Guidelines: provides design guidelines for TWE tests in general.

Modify the Example Test: provides a tutorial for how to modify the example test with typical enhancements using the design guidelines, tips, and best practices.

TWE for MTS TestWorks 4 Users: provides information for MTS TestWorks 4 users that will help them transition to TWE.

Information pertaining to TW Express
MTS offers a companion application to TW Elite named TW Express (TWX). TWX cannot be used to create or modify tests; it can only run existing tests. In cases where this manual refers to functionality shared by both applications (such as running tests), the term “TW application” is used.

Start TWE and Open the Example Test
Install TWE in simulation
Installation of MTS TWE consists of running a self-extracting executable file. When running the MTS TestSuite InstallShield wizard, ensure you select “Insight/Criterion Simulator” when choosing the controller type.

Open the Example EM Tension (Simplified) Test

1. Start the TWE application.
Double-click the TWE icon on your desktop, or click Start > All Programs > MTS TestSuite > TW Elite. The application opens and connects to the simulated MTS Insight controller (the MTS Insight icon appears on the taskbar).

Note: The example test is in Project 1, which is the default project. If Project 1 is not selected as the default project, select Preferences > Configuration > Project > Project 1, right-click the mouse, and then select Set as System Default Project.

2. Open the TW - EM folder.
3. Open the Example MTS EM Tension (Simplified) Test.

Note: If you see a red "X" on the Define and Resources tabs when you first open the application, you must change the controller resource type of some signals from calculation to signal in order to run this test. To do this, click the Resources subtab, expand the Float Signals line, and then change the Controller Resource Type of each float signal from "Calculation" to "Signal".

Note: As a test designer, you typically log on to the TWE application as an Engineer or Administrator. When you open a test, the main window displays the workflow. If you open a test while logged on to TWE as an Operator, or when you open a test with the TW Express (TWX_ application), the workflow is hidden, and the main window shows the test inputs on the Monitor tab.

**TWE Main Window**

When you open a test, click the Define tab and review the basic features in the TWE application as shown in the following figures and tables.
### TWE Main Window Description

<table>
<thead>
<tr>
<th>Number</th>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Menu bar and Quick Access panel</td>
<td>Provides menus and quick-access icons that allow you to perform tasks such as opening tests, opening tests from templates, and saving tests as templates.</td>
</tr>
<tr>
<td>2</td>
<td>Control panel</td>
<td>Allows you to resolve interlocks; show fault status; view the direction in which the crosshead is moving; position the crosshead; and start, stop, and hold the test.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When control is provided by a handset, the crosshead controls will be locked and overlaid by the handset exclusive control icon:</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Status panel</td>
<td>Shows test run information, including name, status, and running time. This panel includes a control that allows you to terminate the current test run.</td>
</tr>
<tr>
<td>4</td>
<td>Toolbox panel</td>
<td>The contents of the Toolbox panel change depending on the subtab selected in the Define tab:</td>
</tr>
<tr>
<td></td>
<td>- <strong>When you select the Procedure subtab,</strong> the Toolbox panel contains components that pertain to test activities such as commands, data acquisition, test flow control, and program actions.</td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5</td>
<td>Outline button</td>
<td>The Outline button shows a hierarchical view of the test in the center panel when selected.</td>
</tr>
<tr>
<td>6</td>
<td>Workflow</td>
<td>Provides a work area to edit tests. You drag test activities from the toolbox to the workflow to edit tests. The workflow contains three sections: Set Up, Run, and Finish.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the Set Up section for activities that you want to perform before you start test runs, such as configuring the load train (setting up fixtures, etc.) and entering a test name. The Set Up section runs only once when you start a test, that is, before the first test run in the test.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the Run section for applying forces to the specimen while performing test runs. The Run repeats for all test runs in the test, and by default shows the Review tab after every test run.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use the Finish section for operations such as printing reports, exporting files, saving test data, and archiving. The Finish section runs when the number of test runs set for the test is complete, or when the operator clicks the Run Finish Section button on the Review tab.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If you are using a multi-head frame and template, your work area will show a Multi-Head Run section instead of a Run section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The workflow includes Pan and Zoom features to help you view complex test procedures.</td>
</tr>
<tr>
<td>7</td>
<td>Properties panel</td>
<td>Allows you to define or change the information, characteristics, and appearance of the selected procedure activities and runtime display components. For example, you can use the Properties panel to change the amplitude of a cycle command test activity in a procedure, or the Y-axis signal selection on a signal scope in a test-run display.</td>
</tr>
</tbody>
</table>

**Main Window Toolbar**
## Application Toolbar Description

<table>
<thead>
<tr>
<th>Number</th>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Controller panel</td>
<td>Allows you to reset interlocks.</td>
</tr>
<tr>
<td>2</td>
<td>Status panel</td>
<td>Shows fault status information. This indicator is primarily for diagnosing system errors with MTS service personnel.</td>
</tr>
<tr>
<td>3</td>
<td>Direction panel</td>
<td>Indicates when the crosshead/actuator is moving and in what physical direction. This relates to physical motion, regardless of whether feedback is increasing or decreasing.</td>
</tr>
<tr>
<td>4</td>
<td>Crosshead Controls panel</td>
<td>Contains controls that allow you to move the crosshead up and down for specimen installation, and for returning the crosshead to the zero or a preset location.</td>
</tr>
<tr>
<td>5</td>
<td>Test Controls panel</td>
<td>Contains controls to run, hold, and stop the test run.</td>
</tr>
<tr>
<td>6</td>
<td>Status panel</td>
<td>Shows test run information, including name, status, and running time. It includes a Cancel Test Run button you can use to cancel, or back out of, the current test run. When the test is stopped, pressing this button cancels the current test run. Pressing the Run button starts a new test run.</td>
</tr>
</tbody>
</table>

**Status Panel with Cancel Test Run Button**

![Status Panel with Cancel Test Run Button](image.png)
About the Multi-Head Run Section

Note: The Multi-Head option is a separately licensed feature.

Note: 
Prerequisite: Only analog data is collected from the multiple active load cells in a multi-head frame. You must have National Instruments Data Acquisition (NI-DAQ) hardware and software or an extensometer installed which will convert the analog data to digital signals.

Multi-head frames can contain up to five active load cells which allow the simultaneous collection of test-runs for multiple specimens. By default, the test run collects the results for all specimens within one file. You can create a separate test-run file for each of the specimens by using the appropriate multi-head template which contains the Multi-Head Run section in the test procedure work area. This section is similar to the Run section found in templates for single head frames which contains activities for applying forces to the specimen while performing test runs. Unlike the Run section, the Multi-Head Run section contains a Properties Panel where you can input configuration data. The Multi-Head Run is only run once. Data is collected simultaneously from the Ni-DAQ analog signals. After the test is completed, the data is split into multiple tests runs. At the end of the test, a number of test runs will be created and populated with the collected data. The new test runs appear post-test as if they were individually collected.

At the end of the multi-head test, the array data is split into multiple test-run files by moving the collected data into the normal force, extension, and optional strain variables.
To create a new multi-head template, you must start with an existing test or template that contains the **Multi-Head Run** section. To save a test as a template, you must be assigned the Administrator or Engineer privilege.

**Multi-Head Run Section Properties**

<table>
<thead>
<tr>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
</tr>
<tr>
<td>General</td>
</tr>
<tr>
<td>Progress Table</td>
</tr>
<tr>
<td>Number of Active Load Cells: 3</td>
</tr>
</tbody>
</table>

**Test Run Variables**

<table>
<thead>
<tr>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
</tr>
<tr>
<td>Width</td>
</tr>
</tbody>
</table>

| Automatically Map Variables | Configure... |

**Load Array**

<table>
<thead>
<tr>
<th>Force Array Variable: Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Variable Mapping: Configure...</td>
</tr>
</tbody>
</table>

**Extension Array**

<table>
<thead>
<tr>
<th>Enable Extension Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension Array Variable:</td>
</tr>
<tr>
<td>Configure Variable Mapping: Configure...</td>
</tr>
</tbody>
</table>

**Trim Data**

<table>
<thead>
<tr>
<th>Use Data Trimming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate after Data Point: Break Index</td>
</tr>
</tbody>
</table>

MTS TestSuite
## Multi-Head Run Section Properties

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Active Load Cells</strong></td>
<td>Specify how many test-runs should be created. For example, if you are testing four specimens, enter 4. When the test run has ended, four test run files will be created. Default: 5</td>
</tr>
<tr>
<td></td>
<td>Click the <strong>Automatically Map Variables</strong> checkbox to have the system map variables, or click the <strong>Configure</strong> button to open the Map Variables window and manually map variables.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you select the <strong>Automatically Map</strong> option and then change the <strong>Number of Active Load Cells</strong> value, some variables will be created or removed.</td>
</tr>
<tr>
<td></td>
<td>If you map variables manually, and then select <strong>Auto Map Variables</strong>, your manually created variables will be overwritten by the automatically-generated variables.</td>
</tr>
<tr>
<td><strong>Test Run Variables</strong></td>
<td>Specify variables that are not in the <strong>Common</strong> category. For example, if <strong>Number of Active Load Cells</strong> is 5 and the <strong>Width</strong> variable is added to this list, the other five variables should be mapped to the <strong>Width</strong> variable (such as <strong>Width1, Width2… Width5</strong>). At the end of the test, the variable values from <strong>Width1, Width2</strong>, and so on, must be moved to the <strong>Width</strong> variable in the appropriate test runs. Click the plus sign to add a variable. To delete a variable, select a variable in the list and click the red minus sign.</td>
</tr>
<tr>
<td></td>
<td>Click the <strong>Configure</strong> button to open the Map Variables window.</td>
</tr>
<tr>
<td><strong>Load Array Variable</strong></td>
<td>Specify the name of the array variable that is used to gather data from the Force (Load) channel. Map it to the other array variables as described above in the <strong>Test Run Variables</strong> description. At the end of the test, the variable values from mapped arrays must be moved to the selected Load Array variable in the appropriate test runs. Click the <strong>Configure</strong> button to open the Map Variables window.</td>
</tr>
<tr>
<td><strong>Enable Extension Channels</strong></td>
<td>Select to map additional array variables to the other array variables. You can also move data from mapped variables to the selected array variable at the end of the test. Click the <strong>Configure</strong> button to open the Map Variables window.</td>
</tr>
<tr>
<td><strong>Trim Data</strong></td>
<td>Select to have the trimming process performed after the data has been split into test-run files. The trimming process will keep the point identified by the variable. In other words, the data will be trimmed at the variable index plus one. If the variable is invalid, no truncation will take place and a message will be logged. Default for <strong>Truncate after data point</strong>: Break</td>
</tr>
</tbody>
</table>
Examine the Example Test

About This Chapter ........................................................................................................... 24
Design Flow ......................................................................................................................... 24
Define Tab ............................................................................................................................. 25
Monitor Tab .......................................................................................................................... 45
Review Tab ............................................................................................................................ 47
Runtime Values Tab .............................................................................................................. 53
Examine the Example Test

About This Chapter
This chapter provides a detailed review of the example test provided with the TWE application.

You will examine the test’s workflow, test activities, use of variables, and so on. Understanding how the example test was created with the TWE application and how the test appears in the interface will help you understand how to create tests of your own.

For information about modifying the example test for a different scenario, see “Modify the Example Test” on page 79.

For More Information
“Start TWE and Open the Example Test” on page 14
“Modify the Example Test” on page 79

Design Flow
The flow for designing tests with TWE is defined by the tabs shown on the main display after you open a test:

- **Select** tab (or the initial open test/template window)
- **Define** tab
- **Monitor** tab
- **Review** tab
- **Runtime Values** tab

![Main Tabs in TWE](image)

Note:
If the selected test has validation errors, the test opens to the **Define** tab so operators can use the **Resource** subtab to resolve resource conflicts.

**Select tab**
Use the initial open test/template window or the **Select** tab to choose an existing test or template to modify.

**Define tab**
Use the **Define** tab to modify the selected test by modifying the workflow, editing variables, applying functions, and so on.
Monitor tab
Use the Monitor tab to review pretest variables (inputs that typically apply to all of the test runs in the test) shown to the operator. You may create additional pretest variables by editing variable properties in the Variables subtab of the Define tab.

Review tab
Use the Review tab to view test results, customize table and chart content, define markers, and customize report templates with the Reporter Add-In for Microsoft Excel.

Runtime Values tab
Use the Runtime Values tab to view current variable values and activity configuration values.

Define Tab
You perform most test design tasks on the Define tab, which includes the following subtabs:

- Procedure
- Test-Run Display
- Variables
- Report Templates
- Functions
- Resources
- Test Definition

Outline View in the Procedure Subtab
When a test is opened in TWE, the Define tab and Procedure subtab are shown by default. You can select Outline view by clicking the Outline button. The Outline view shows the procedure as a hierarchy.

Outline View and the Find Feature
Examine the Example Test

Tip
The **Find** feature is particularly useful for locating activities nested within other activities. For instance, to locate the **Go To + DAQ + Detection** command activity in the example test, enter **Go To** and press return. The outline hierarchy expands to show the **Go To + DAQ + Detection** activity, and the activity is highlighted in the workflow.

Workflow in the Procedure Subtab
The center of the main window shows the workflow. The workflow consists of a sequence of connected activities divided into test sections; it is a graphical representation of the test.

Each section performs a specific function within the test. All TW tests include the following sections:

- Set Up
- Run
- Finish

Set Up Section
The Set Up section runs only once before the first test run.

You typically use this section for activities that you want to perform before you start test runs, such as configuring the load train (setting up fixtures, and so forth) and entering a test name.
Examine the Example Test

The Set Up section of the example test contains three disabled activities: **Input Variables** activity, **Read data** activity, and **Calculate Variables** activity.

- When enabled, the **Input Variables** activity shows a list of selected variables and their current values to an operator so they can enter variable values.
- When enabled, the **Read Data** activity allows the test to read variable values from an XML file and apply those values to the current test run.
- When enabled, the **Calculate Variables** activity calculates all variables assigned to the activity.

For more information about activities, see “Modify the Example Test” on page 79.

### Set Up Section of the Example Test

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Input Variables</strong> activity (disabled)</td>
</tr>
<tr>
<td>2</td>
<td><strong>Read data</strong> activity (disabled)</td>
</tr>
<tr>
<td>3</td>
<td><strong>Calculate Variables</strong> activity (disabled)</td>
</tr>
</tbody>
</table>

For More Information

“Modify the Example Test” on page 79

### Run Section

The Run section performs a test run when the operator clicks the Run button. After each test run, the **Review** tab appears, which shows test data in tables and charts.
The Run section of the example test has been performed multiple times, so the **Review** tab includes data from multiple test runs.

> **Note:**
The following figure shows the **Parallel Paths** activity at the root of the Run section collapsed.

**Run Section of the Example Test**

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Input Variables</strong> activity</td>
</tr>
<tr>
<td>2</td>
<td><strong>Read data</strong> activity (disabled)</td>
</tr>
<tr>
<td>3</td>
<td><strong>Calculate Variables</strong> activity</td>
</tr>
<tr>
<td>4</td>
<td><strong>Parallel Paths</strong> activity</td>
</tr>
<tr>
<td>5</td>
<td><strong>Input Variables</strong> activity (disabled)</td>
</tr>
<tr>
<td>6</td>
<td><strong>Calculate Variables</strong> activity (disabled)</td>
</tr>
<tr>
<td>Number</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>Run Report activity (disabled)</td>
</tr>
<tr>
<td>8</td>
<td>Write data activity (disabled)</td>
</tr>
<tr>
<td>9</td>
<td>Return to Zero activity</td>
</tr>
</tbody>
</table>

For more information on the Parallel Paths activity (4), see “Program Command in the Example Test” on page 30.

**Examining the Logic in the Run Section**

**Default Test Behavior**

The operator starts the test by clicking the run button on the Test Controls panel.

At the beginning of the test, the operator is prompted for specimen dimensions per the Input Variables activity (1).

Then the Calculate Variables activity (2) uses the operator input to calculate area (this does not affect the interface).

Next, the Monitor tab shows a graph of Load versus Extension as the Go To + DAQ + Detection activity performs a ramp. The Go To + DAQ + Detection activity is nested within the Parallel Paths activity (4), which performs additional functions, such as data acquisition and break detection.

**Note:** In any instance where a Data Acquisition activity (DAQ) is in a parallel path, the software starts recording data inside the parallel path from left to right. To reduce the delay in recording the first data point, place the first DAQ to the far left in the parallel path.

When a break is detected, the ramp stops and the operator is prompted to return to zero per the Return to Zero activity (9).

Next, test run results are shown on the Review tab. At this point, the operator may start another test run by clicking the Run button on the Test Controls panel, or end the test by clicking the Run Finish Section button on the Review tab.

**Behavior With the Other Activities Enabled**

Read data activity (2)—This activity allows the test to read variable values from an XML file and apply those values to the current test run. You can configure the activity to prompt the operator for the file name during test runtime, or allow the activity to automatically select the file while running a test.

Input Variables activity (5)—When this activity becomes active in the test workflow, a list of selected variables and their current values is shown to the operator. The operator can edit the variable values as required.

Calculate Variables activity (6)—This activity calculates all variables assigned to the activity at a specific point in workflow.
Examine the Example Test

**Run Report** activity (7)—This activity generates a test report based on the selected report template. The generated report is a Microsoft Excel file by default. You can also select other report format options, such as PDF. You can configure the Run Report activity to send the report to one or more e-mail addresses.

**Write data** activity (8)—This activity directs the MTS TestSuite application to write variable values from a test run, such as array variable data and other results, to an XML file.

**Program Command in the Example Test**

In the example test, program command is issued from the Go To + DAQ + Detection activity nested within the Parallel Paths activity at the root of the Run section. This is the only activity in the example test that applies forces to the specimen.

### Branches in the Parallel Activity

![Diagram of Parallel Activity](image)

### Description of Branches in the Parallel Activity

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Auto Offset and Go To + DAQ + Detection activities. For more information, see Path for the Typical Test Flow.</td>
</tr>
<tr>
<td>2</td>
<td>State Change Detector activity. For more information, see Path with the State Change Detector.</td>
</tr>
<tr>
<td>3</td>
<td>Limit Detection activity: Specimen. For more information, see Path with the Limit Detector for the specimen.</td>
</tr>
<tr>
<td>4</td>
<td>Extensometer Removal activity. For more information, see Path with the Extensometer Removal activity.</td>
</tr>
</tbody>
</table>
Branches in the Parallel Paths Activity
The Parallel Paths activity in the example test contains four parallel branches. All of the branches are terminal, meaning the completion of any of the paths ends all of the other paths.

Path for the Typical Test Flow
In a typical test flow, the path that contains the Auto Offset and the Go To + DAQ + Detection activities (1) ends the other paths when the specimen fails. The Auto Offset activity resets the running time of the test run to zero, and the Go To + DAQ + Detection activity provides program command.

Note: In any instance where a Data Acquisition activity (DAQ) is in a parallel path, the software starts recording data inside the parallel path from left to right. To reduce the delay in recording the first data point, place the first DAQ to the far left in the parallel path.

Path with the State Change Detector
The State Change Detector path (2) detects a stopped test state. If the operator clicks the Stop button while the Go To + DAQ + Detection activity issues command, the Stop Change Detector activity detects the stopped state and ends the Parallel Path activity. So if the operator clicks the Stop control while the system is ramping, the ramp stops immediately and the “Return to Zero?” prompt appears. Because of this, the operator cannot stop and then resume command in the example test.

Path with the Limit Detector for the specimen
The first Limit Detection activity (3) detects minimum and maximum load values applied to the specimen while the Parallel Path activity is active. The minimum and maximum values are defined by Load Limit Low and Load Limit High variables. The action for both limits is set to No Action. This means if either of the specified values are detected, the detector trips and the Parallel Path activity ends.

This activity protects the specimen by allowing you to define the maximum compressive and tensile forces applied to the specimen while the command is applied.

Path with the Extensometer Removal activity
The Extensometer Removal activity (4) is disabled by default. When enabled, this activity allows a test designer to specify the extensometer removal point.

About Detector Action Resources
Actions are test resources. By default, the only action available to the detectors in the example test is No Action. The MTS Insight controller also supports Program Hold, Program Stop, and Interlock actions, but these actions have been removed from the test because they are not used.

About Restoring Detector Actions
You can restore unused controller resources to tests by importing them with the controls in the Resources tab (Define tab > Resources subtab). In “Modify the Example Test” on page 79, you will restore all controller actions to the example test.

For More Information
“Modify the Example Test” on page 79
Examine the Example Test

Go To + DAQ + Detection Activity
This composite activity performs multiple functions in parallel associated with the following tabs on the Properties editor:

- Go To
- DAQ
- Limit Detection
- Break Detection
- Wait for Operator Action

For most tests, including the example test, the command content of the test is provided by one or more Go To + DAQ + Detection activities in the Run section.

Go To Tab
The Go To tab is configured to ramp the crosshead at a rate defined by the Test Rate variable. The Termination Condition on this tab is not enabled, so the ramp is configured to continue until a specimen failure is detected by the condition specified in the Break Detection tab. If that condition is not detected, the ramp continues until one of the other branches of the Parallel Branch activity (such as a limit detection branch) completes.

Go To Tab and Edit Variables Window
Examine the Example Test

DAQ Tab
The DAQ (data acquisition) tab is configured to continuously acquire timed load (force) and crosshead (length) data at a sample rate defined by the Data Acquisition Rate variable. It is also configured to map acquired data to array variables.

**DAQ Tab, Trigger Properties Window, and Map Variables Window**

---

About Mapping Data to Variables
In all MTS templates created for TWE, the **Save data to variables?** control is enabled by default. Mapping data to arrays significantly extends the functionality of TW. Array variables provide the data in the charts and tables in the **Review** tab, support the use of several types of test-run displays (for example, the Array-Variable Chart, the Variable Data Table, and the Variable Meter), and support the use of the Reporter Add-In for Microsoft Excel.

Limit Detection Tab
Limit detection is not used in this tab in the example test.

Break Detection Tab
The **Break Detection** tab detects when the specimen breaks while the **Parallel Path** activity is active.
Examine the Example Test

The break detector monitors the peak load applied to the specimen. When the current value exceeds the value defined by the Break Threshold variable, the detector is armed.

When the detector is armed and the peak load value drops by the percentage defined by the Break Sensitivity variable, a break is detected. The action is set to No Action. This means if a break is detected, the detector trips and the Parallel Path activity ends.

For many types of tensile tests, the test run completes when the specimen fails and the break detector trips.

**Break Detection Tab**

![Break Detection Tab]

**Wait for Operator Action Tab**

The Wait for Operator Action tab enables the operator to click a button shown on the Monitor tab while the test run is being performed. Clicking this button prematurely ends the Parallel Path activity.

The button label shown on the Monitor tab is determined by the text entered in the Display Text box.

In the example test, the button is labeled Next Segment. When the operator clicks the Next Segment button on the Monitor tab while the test is running, program command stops and the "Return to Zero?" message is shown.
Examine the Example Test

Wait for Operator Action Tab and the “Next Segment” Button Shown on the Monitor tab

Finish Section
The Finish section runs:

- Automatically, when the Maximum Test Runs variable equals the number of test runs performed, or
- Manually, when the operator clicks the Run the Finish Section button on the Review tab toolbar. The Review tab appears automatically after each test run.

In the example test, the Maximum Test Runs variable is set to 9999 (default), so it is designed to be ended manually.

The Finish section is typically used for operations such as printing reports, exporting files, saving test data, and archiving.

The example test contains four disabled activities: Input Variables activity, Calculate Variables activity, Run Report activity, and Write data activity.

- When enabled, the Input Variables activity shows a list of selected variables and their current values to an operator so they can enter variable values.
- When enabled, the Calculate Variables activity calculates all variables assigned to the activity.
- When enabled, the Run Report activity generates a test report based on the selected test report template.
- When enabled, the Write data activity directs the MTS TestSuite application to write variable values from a test run, such as array variable data and other results, to an XML file.
Examine the Example Test

Finish Section of the Example Test

1. Input Variables activity (disabled)
2. Calculate Variables activity (disabled)
3. Run Report activity (disabled)
4. Write data activity (disabled)

Run the Finish Section Button on the Review Tab

Test-Run Display Subtab

When you click the Test-Run Display subtab of the Define tab, the center of the main window shows the designer view of the test-run displays used in the test. You can think of a test-run display as a readout device—like a meter or scope—that shows test signals or other information on the Monitor tab during test runs.

The only test-run display used in the example test is the Array-Variable chart. It uses the Load variable for the Y-axis and the Extension variable for the X-axis, as shown on the Properties panel.
You can add any of the test-run displays in the Toolbox to the test to view test information during test runs.

**Array-Variable Chart on the Test-Run Display Subtab**

When you click the **Variables** subtab of the **Define** tab, the main window shows the variables used in the test.

**Variables Subtab and Variable Availability Property**
Variable Definition
A variable is a place with a name in which you can store one or more values. Variable types and examples used in the test include:

- Simple variables (also referred to as scalar variables) store either a value or a string. In the example test, the variable used to set the rate at which data is acquired is a simple variable. It has a display name of Data Acquisition Rate, an internal name of DataAcqRate, and default value of 10 Hz.

- Array variables store an array, which is an ordered collection of numbers or strings. You can refer to individual values in an array by specifying an index. In the example test, the variable used to store acquired load data is an array variable. It has a display name of _Load, an internal name of _LoadArray, and a default value of 0.000 kN (array variables resize automatically, so you can disregard the default value).

- Calculated variables are variables (simple or array) that include a mathematical expression. In the example test, the variable used to store stress data is a calculated variable. It has a display name of Stress, an internal name of _StressArray, a default value of 0.000 kN/mm, and a calculation of _LoadArray/Area.

Variable Use in the Example Test
Variables are used extensively in the example test. While using variables is not a requirement for test design, using variables significantly extends the functionality of tests created with MTS TestSuite software.

If variables were not used in the example test:

- To change the properties of activities that are currently variables, you would have to navigate through the layers in the workflow and locate specific property fields.

- You could not use the array-variable chart on the Test-Run Display subtab, or any of the other test-run displays in the Variable Views category (or other test-run displays that use variables). You would only be able to use test-run displays in the Signal Views category, such as signal meter and signal scope.

- The charts and tables in the Review tab, which are based on variables, would not contain useful information.

- You would not be able to show test data in charts and graphs using the Reporter Add-in for Excel. In this case, test data would be available only as tables of sampled signal data values.

Variable Availability Attributes
Variables have one or more of the following Availability attributes:

- Preallocate
- Pretest
- During-Test
- Result
- Editable Post-Test
Examine the Example Test

The **Pretest, During Test, and Editable Post Test** attributes specify when the operator can change the variable during the test.

**Preallocate Attribute**
Variables with this attribute can be pre-defined by a test designer. This allows the value of the input variable for individual test runs to be entered all at once for a batch of specimens, for example.

**Pretest Attribute**
Variables with this attribute can be edited before the test is initialized and before each test run. In the example test, the Data Acquisition Rate, Test Rate, and Grip Separation variables have this attribute. They appear on the **Monitor** tab before the test is started and each time the operator clicks the **Run** button.

**During Test Attribute**
Variables with this attribute can be edited in any of the test sections (Set Up, Run, or Finish) after the test is started. In the example test, the **Input Parameters** activity in the Run section prompts the operator for the width and thickness of the specimen, which apply to the Width and Thickness variables.

**Editable Post-Test Attribute**
Variables with this attribute can be edited in the Variables table on the **Review** tab after the test is complete (1). The **Variables** table does not appear by default. To view the **Variables** table, you must change the default panel view on the **Review** tab by clicking the Two or Four-Panel Views button (2). In the example test, the Break Index variable has this attribute (3).

![Access to the Variables Table and Editable Post-Test Variables](image)

**Result Attribute**
The Result attribute specifies the variables that appear in the **Result** table and **Variables** tables in the **Review** tab. In the example test, the Peak Load variable has the Result attribute, and appears on the **Results** table.
Examine the Example Test

Peak Load Variable on the Result Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Peak Load (kN)</th>
<th>Peak Stress (kN/mm²)</th>
<th>Strain at Break (mm/mm)</th>
<th>Modulus (kN/mm²)</th>
<th>Width (mm)</th>
<th>Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Run 1</td>
<td>5.000</td>
<td>0.1</td>
<td>0.225</td>
<td>0.520</td>
<td>10.300</td>
<td>12.700</td>
</tr>
<tr>
<td>Test Run 2</td>
<td>5.000</td>
<td>0.1</td>
<td>0.235</td>
<td>0.630</td>
<td>12.700</td>
<td></td>
</tr>
<tr>
<td>Test Run 3</td>
<td>5.000</td>
<td>0.1</td>
<td>0.235</td>
<td>0.630</td>
<td>12.700</td>
<td></td>
</tr>
</tbody>
</table>

Variables with Both Results and Editable Post Test Attributes
Some variables include both Editable Post Test and Result attributes. In the example test, this applies to the Thickness variable. In this case, the variable appears in two locations on the Review tab:

- In a non-editable form in the Results table
- In an editable form in the Variables table (appears when you select a multi-panel view)

Report Templates Subtab
The Report Templates subtab shows the report templates available to the example test. Report templates are Microsoft Excel files that specify what and how test information appears in generated reports. Report templates are workbook files (.xltx); you can generate reports without Excel installed. Excel must be installed to create or modify report templates.

Default Test-Run Report Template
Use the default Test-Run Report template for generating reports for individual test runs. The example test does not include an activity in the workflow to generate a report for test runs. However, the operator could generate a report for test runs using the toolbar controls on the Review tab.

Default Test Report Template
Use the default Test Report template for generating reports that apply to all the test runs in the test. The example test includes a disabled Run Report activity in the Finish section of the workflow. To generate a test report, the designer would have to enable this activity. Otherwise, the operator could generate a test report using the toolbar controls on the Review tab, as shown
Examine the Example Test

Generate Reports Buttons on the Review Tab

Functions Subtab
The Functions subtab shows the custom functions available to the example test. The MTS TestSuite TW application uses functions to perform complex mathematical operations during tests.

The example test includes the SimulatedLoad function, which derives load data from extension data provided by the MTS Insight Controller simulator. This function was created with Python. To view the Python code for this function (as shown), select the function and click Edit, or double-click the function.

Using Python with MTS TestSuite
Python is an interpreted, general-purpose, high-level programming language with an extensive resource library. You can use Python to create custom functions to extend the functionality of MTS TestSuite software. You can use Python for reading and writing to files, dynamic mathematical calculations, and manipulating variables.

Python Code for the SimulatedLoad Function in the Function Editor

Resources Subtab
The Resources subtab of the Define tab shows the mapping of test resources to controller resources.
Examine the Example Test

Description of Resources Subtab

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connection indicator</td>
</tr>
<tr>
<td>2</td>
<td>Test resources</td>
</tr>
<tr>
<td>3</td>
<td>Controller resources</td>
</tr>
<tr>
<td>4</td>
<td>Resource management controls</td>
</tr>
</tbody>
</table>

Controller identification

The connection indicator (1) shows the controller configuration associated with the current test when MTS TW is connected to a controller. In the example test, the controller configuration is labeled “C42.503 (Simulator)”.

The main window shows the test name, project, application name, controller name, and controller configuration. In the example test, this is:

“Example MTS EM Tension (Simplified) Test: Project 1: TW Elite (Insight:C42.503 (Simulator))”.

Test resources

The resources in (2) are test resources. Test resources are required by the test and must map to equivalent resources in the Controller Resources column when connected to a controller.

In the example test, the control mode labeled “Crosshead” is a test resource. You can use test resources for various test elements, such as activities, runtime display devices, and variables.

Note:

For MTS Criterion and MTS Insight systems, load and strain control modes are available only if you have purchased the Advanced Rate Control license.
Controller resources
The resources in (3) are controller resources. A controller resource refers to a hardware resource in the controller as it appears in the controller configuration. In the example test, the control mode labeled “_Extension” is a controller resource.

When connected to a station, controller resources reflect the resources in the controller configuration. When not connected to a controller, controller resources are the resources MTS TWE expects to find when connected to a controller.

Managing resources
Resource management controls (4) allow you to add, import, delete, and rename test resources.

It is important to understand that only the controller resources required by the test are typically shown on the Resources subtab. To view all of the resources available in the connected controller, click the Import Resources... > Import All Unused Controller Resources.

In the example test, the default Resources subtab shows just one control mode resource; when all unused resources are imported, it shows multiple control modes, as shown.

Resource mapping
A test resource is not required to have the same name as the equivalent controller resource. For instance, in the example test, the control mode test resource is labeled “Crosshead”, and the equivalent controller resource is labeled “_Extension”.

However, if the names are different when MTS TWE connects to the controller, you must map the test resource to the equivalent controller resource.

Suppose you disconnect the example test from the controller and reconnect to a different controller in which the equivalent resource is labeled “_Displacement”. In this case, MTS TWE would be unable to automatically map the new controller resource “_Displacement” to the test resource “Crosshead”, and a validation icon would appear in the window.

To resolve the error, you would have to manually map the controller resource to the test. To do this, you would click the list icon next to the “_Displacement” test resource and select “Crosshead.”
Examine the Example Test

Resource mapping is saved with the test.

⚠️ Note:
When you create tests while the MTS TWE application is connected to a controller, it is referred to as “online” test design. You can also perform “offline” test design.

**Test Definition Subtab**
The Test Definition subtab of the Define tab shows general settings for the test, including:

- The Formula Assistant File
- The Log Type, which determines whether an audit trail is created for the test.
- Whether the test can be started remotely with a digital input.
- Whether the display automatically switches to the Review tab at the completion of a test run.

⚠️ Note:
All fields are read-only until you click the Edit button.

**Formula Assistant File**
The Formula Assistant File provides a base set of system variables, calculations, and functions associated with a specific type of test. The example test uses a formula assistant file labeled “Electro-Mechanical Formula Assistant”.

If you need to change the Formula Assistant File associated with a template, contact your MTS Field Service Engineer for assistance.

When a test variable is associated with the Formula Assistant File, an Option list becomes available in the Calculation panel for the variable. The options available depend on the selected variable.

For example, when you select the Area variable on the Variables subtab of the Define tab, several specimen geometry types are available in the Options list.

**Log Type**
The log type selections include Basic or Audit Trail. The example test uses the Basic log type.

The Audit Trail feature is for tracking changes to the test. When selected, each time you run a test on an MTS Criterion or MTS Insight system, the application writes the TEDS model and serial number to the test run log. When the test is complete, you can right-click a test run in the Review tab and validate the raw data for that test run.
Remote Start
This control is disabled by default, and is disabled for the example test.

This control works in tandem with the Digital Input control in the Preferences menu (Preferences > Configuration > Test > Select additional methods to start the test > Digital Input).

If you enable both controls, you can select a digital input source and specify a transition type to start the test remotely.

Switching to the Review Tab
This control is enabled by default, and is enabled for the example test.

If you disable this control, the display remains on the Monitor tab after each test run instead of automatically switching to the Review tab.

For More Information
“Change the Specimen Geometry” on page 82

Monitor Tab

Monitor Tab View Before the Test Starts
Before the operator starts the test by clicking the Run button, the Monitor tab shows the pretest inputs the operator can change.

In the example test, this includes the Data Acquisition Rate, Test Rate, and Grip Separation variables, as shown.

Test Opens to Monitor Tab for Operators
If you open a test while logged on to TWE as an Operator, or when you open a test with the TWX application, the workflow is hidden, and the main window shows the test inputs on the Monitor tab. If the test contains validation errors, it opens to the Resource subtab of the Define tab so the operator can resolve the errors.

Editing Pretest Inputs on the Variables Subtab
You can edit the pretest inputs that are shown to the operator in the Variables subtab of the Define tab. To view variables that have the pretest attribute, use the Column Chooser to add the Pretest column to the table, then edit as desired in the Availability panel.
Examine the Example Test

Edit Pretest Inputs

1. Click this icon to show the Column Chooser.
2. Select Pretest in the Column Chooser.
3. Pretest attribute appears in the table.

Description of Variables Tab

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click this icon to show the <strong>Column Chooser</strong>.</td>
</tr>
<tr>
<td>2</td>
<td>Select <strong>Pretest</strong> in the Column Chooser</td>
</tr>
<tr>
<td>3</td>
<td>Pretest attribute appears in the table.</td>
</tr>
</tbody>
</table>

Monitor Tab View After the Test Starts

After the operator starts the test, the **Monitor** tab shows test information on the test-run display included in the test design until the test run is complete. In the example test, the Array-Variable Chart is used, as shown.
Review Tab

The **Review** tab shows test results, which includes:

- Test run data
- Compiled statistical data as multiple test runs are completed.

Test data is shown in tables and charts. The default view of the **Review** tab for the example test is shown below.
Examine the Example Test

Default View of Review Tab

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Toolbar</td>
</tr>
<tr>
<td>2</td>
<td>Test Run Results table</td>
</tr>
<tr>
<td>3</td>
<td>Statistics table</td>
</tr>
<tr>
<td>4</td>
<td>Review chart</td>
</tr>
</tbody>
</table>

Review Tab Toolbar

The **Review** tab toolbar contains icons that allow you to show, analyze, and manage test data.
Review Tab Toolbar Description

<table>
<thead>
<tr>
<th>Number</th>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Revert and Recalculate Test Run/Test Variables buttons</td>
<td>The Variables table on the Review tab allows you to create what-if analysis scenarios by changing test variable values and observing the test results. The Revert and Recalculate buttons allow you revert changes that pertain to individual test runs and to the test (which includes all the test runs in the test).</td>
</tr>
<tr>
<td>2</td>
<td>Run Finish Section button</td>
<td>Runs the Finish section. You may also run the Finish section automatically by entering the desired number of test runs in the Maximum Test Runs variable, which has a default value of 9999.</td>
</tr>
<tr>
<td>3</td>
<td>Generate a report for Test Runs/Test buttons</td>
<td>Generates reports that you can open using Microsoft Excel. You can generate reports that pertain to individual test runs or the test. You can choose pre-made or customize report templates.</td>
</tr>
<tr>
<td>4</td>
<td>Print a report for Test Runs/Test buttons</td>
<td>Prints reports generated from individual test runs or for the test.</td>
</tr>
</tbody>
</table>
| 5      | One, Two, or Four Panel View buttons | Selects the number of panels shown on the Review tab.  

**Note:**  
The default Review tab shows only the Results table, which does not show variables you can edit (variables with the Edit Post Test attribute). To show the Variables table, which includes variables you can edit, click the Two or Four Panel View buttons. |
| 6      | Actions button | Opens a menu where you can save your display layout, open the Display Manager to work with your displays, and view details about generated reports in the Reports window. |

Test Run Results Table

The Test Run Results table shows the results of each test run. By selecting test runs in the Compare column, you determine the test-run data that appear in the various charts and tables. The variables listed in the columns of this table are not editable. To view editable variables, you must click the two or four panel view buttons in the toolbar to show the Variables and Common Variables tables.
Examine the Example Test

Right-click Menu for the Test Run Results Table
Right-click menus allow you to:
- Select test runs for comparison
- Copy and delete test runs
- Export raw data
- Configure Autotag rules

Export Raw Data
The Export Raw Data feature allows you to export raw data (sampled from signals versus data mapped to variables) as tab or comma-delimited text (.txt) files. You can show or manipulate exported data with other applications, such as Microsoft Excel or Word.

Configure Autotag Rules
The Autotag feature allows you to create rules that determine when a test run is automatically tagged (excluded). The Autotag rules are applied as each test run is run or can be defined and applied post test.

Tagging Specimens Manually
You can manually tag (exclude) test runs by selecting the test runs in the Tagged column. Tagged test run data is not compiled in the Statistics table.

Statistics Table
The Statistics table contains statistical data calculated from test runs. These calculations include data from all of the completed test runs that are not tagged (excluded). The example test includes Mean and Standard Deviation values.

Configuring Statistics
You can add additional statistical values to the Statistics table with the Configure Statistics feature on the right-click menu.

Panel of Charts and Tables
The Charts and Tables Panel shows a chart of the active (highlighted) test run, in addition to the test runs selected (checked) in the Compare column. For more information about the table and chart views that can be shown in the panel, see the MTS TestSuite TW Elite User Guide.

In the example test, the chart in the panel shows an Array-Variable Chart that plots load versus extension. It includes the following markers:
- B (Slope 1 Index, or beginning of modulus)
- M (Slope 2 Index, or end of modulus)
- F (Break Index, or point of failure)
- Y (Yield Index, or point of yield)
Note:
In the example test, the F (Break Index) and Y (Yield Index) markers are at the same point, and the marker symbols are positioned on top of one another.

Chart Markers
Chart markers are variables. Each marker on the chart defines a data point in the test run. Chart marker variables can be defined by either a calculation or the occurrence of a run-time event. In either case, the marker variable saves an index into a data array. When the marker is added to a chart, its position represents the X and Y axis values associated with the index.

Chart Marker Variable Manipulation
Examine the Example Test

Description of Chart Marker Variable Manipulation

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Right-click the M marker (Slope 2 Index variable) and select Edit Variable.</td>
</tr>
<tr>
<td>2</td>
<td>Use the Edit Variable window to review and edit the marker variable properties.</td>
</tr>
<tr>
<td>3</td>
<td>You can move the M marker by entering a new value in the Variables table or by selecting the Move Selected Marker in the marker right-click menu.</td>
</tr>
</tbody>
</table>

**Calculated Markers**

Calculated markers are typically derived from a calculation and are often used in a calculation. For example, the least squares calculation for the modulus line in the example test uses two calculated marker variables that appear on a chart. (Marker B, the Slope 1 Index variable; and Marker M, the Slope 2 Index variable).

**Using Chart Variables in Test Reports**

You can add, remove, and edit the markers on the chart. You can also use chart variables elsewhere.

For example, if you add a marker at the maximum load (yield) point on the chart, you can use the X-Y (load, strain) values associated with that marker in a test report that specifies the maximum load value.

**Four-Panel View**

The following figure shows the default four-panel view of the example test. Each panel is independent; the charts and tables do not correspond to one another.

Charts have a right-click menu that allow you to:

- Zoom
- Configure the chart by editing the appearance of trace, curve fit, limit, and axis lines
- Add text
- Add or remove markers
- Copy values and images
- Switch, add, and manage views
Four-Panel View of Review Tab

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Review Chart view in the four-panel view is the same Review chart shown in the one-panel view. It shows a chart of the active (highlighted) test run, in addition to the test runs selected (checked) in the Compare column.</td>
</tr>
<tr>
<td>2</td>
<td>The Review Chart for Multiple Runs view shows a chart of the selected (checked) test runs, in addition to the active (highlighted) test run, without markers.</td>
</tr>
<tr>
<td>3</td>
<td>The Variable Table view shows a table of variables of various categories (except Common) which pertain to the active (highlighted) test run. When you select more than one Compare check box on the Test Run Results table, a drop-down test-run list appears in the Variables table.</td>
</tr>
<tr>
<td>4</td>
<td>The Common Variable Table view shows a table of variables of the Common category which pertain to all of the test runs.</td>
</tr>
</tbody>
</table>

Runtime Values Tab

The **Runtime Values** tab shows and updates variable values in real time while you perform a test run. Test designers can use the **Runtime Values** tab to validate their test designs.

When you initially open a test, this tab will not appear on the main window until you run the test.

When you start the test (and when you start each subsequent test run in the test), the display switches to the **Monitor** tab. If desired, you can click the **Runtime Values** tab and monitor the variables values. When the test run is complete, the display automatically switches to the **Review** tab.
Examine the Example Test

The figure below shows array-variables updating on the **Runtime Values** tab during a test run.

**Runtime Values Tab During a Test Run**
Design Guidelines

About This Chapter .................................................................................................................. 56
Designing Tests ......................................................................................................................... 56
Design Guidelines

About This Chapter
This chapter contains concepts, design considerations, and recommended practices for designing tests with TWE.

For related procedural information and interface reference, see the MTS TestSuite TW Elite User Guide installed with the TWE application (Start > Programs > MTSTestSuite > Documentation > TWEliteUserGuide).

For a step-by-step tutorial for modifying the Sample Tension (Simplified) Test, see “Modify the Example Test” on page 79.

This section contains tips and recommended practices.

For More Information
“Modify the Example Test” on page 79

Designing Tests

✅ Recommended:
   It is not possible to create a test from scratch using the TW application. Choose an existing test or template that is as close as possible to your desired design and modify as needed.

💡 Tip:
   The process of test design is iterative. It is good practice to implement your changes iteratively and run the test after each iteration.

Test Design Checklist
The following table lists the typical order of tasks a test designer may perform to prepare a test for an operator.

<table>
<thead>
<tr>
<th>Step</th>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start the TWE application and select a test or template to customize</td>
<td>Start the TWE application and select a test or template that is as close as possible to the test you want the operator to perform. When you open a template, you are actually opening a test version of the template.</td>
</tr>
<tr>
<td>2</td>
<td>Modify the test</td>
<td>Modify the test by enabling disabled activities within the template, changing Formula Assistant file options, adding resources, and editing the workflow.</td>
</tr>
<tr>
<td>3</td>
<td>Select and customize a Test Report template</td>
<td>Create or modify the existing test report template by adding test variables, etc., from your test run to the template. Editing report templates requires the optional MTS Reporter Add-In for Microsoft Excel.</td>
</tr>
<tr>
<td>4</td>
<td>Run the customized</td>
<td>You typically run your test from the operator’s perspective. You may</td>
</tr>
<tr>
<td>Step</td>
<td>Activity</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>test to validate the design</td>
<td>choose to enhance your test with special operator information.</td>
</tr>
<tr>
<td>5</td>
<td>Save your test as a template</td>
<td>When the test runs as desired, you typically save the test as a template.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When the operator opens the template, they will actually be opening a test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>version of the template.</td>
</tr>
<tr>
<td>6</td>
<td>Deliver your template to the operator</td>
<td>When your new template meets your requirements, deliver it to the operator.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The operator can run the template with TWX or with TWE (you can log on to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TWE as an Operator, Engineer, or Administrator).</td>
</tr>
</tbody>
</table>

**Tests and Templates**

**Choosing a Template to Run or Modify**

Locate a template that contains as many of your test requirements as possible.

Open the template (which becomes an editable test when you open it), modify as desired, and test it. Note that when you run your test to validate the modifications, the application automatically saves the test. Now save the test as a custom template. When the operator wants to test a batch of specimens, they begin by opening your custom template, and typically begin the testing of each subsequent batch of specimens by opening your custom template.

For test inputs, use variables to allow a central definition point for values that exist in multiple areas in a test.

For test outputs, such as acquired data, use variables to support test reporting. For instance, the data acquisition activity in the example test is configured to automatically map variables.

**File Icons**

The following table describes the icons associated with the files that you access in the Open Test, New Test from Template, and Copy Existing Test window.
File Type Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>File Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="MTS Templates" /></td>
<td><strong>MTS Templates</strong>—Only the templates for which you are licensed appear available in the MTS Templates folders. Templates that are licensed but not applicable to an application are grayed out.</td>
</tr>
<tr>
<td><img src="image" alt="Custom Templates" /></td>
<td><strong>Custom Templates</strong>—Includes any test or template saved as a template.</td>
</tr>
<tr>
<td><img src="image" alt="Tests" /></td>
<td><strong>Tests</strong>—An instance of a template that you can edit and save as a test or template.</td>
</tr>
</tbody>
</table>

### Working with Templates and Tests
When you open a template, you are in an editable copy of the template, referred to as a test file. If you want to make changes to a template, you must open a test version of the template and save it as a template (File > SaveAs > Template) with the same name as the original template.

**Note:**
- You cannot edit MTS-supplied templates.

When you export a test, TW combines the current test and its associated files into one compressed (and self-contained) file and assigns an extension of .tsproj.

Exported tests can be managed with the Windows operating system. Unexported tests must be managed within the MTS TestSuite application.

When you save a template, TW compresses the template and assigns an extension of .TWTemplate.

Templates can be managed with the Windows operating system.

### About Running Tests from Templates
Operators typically begin a new test by opening a template, which becomes a test, and saving the test with a name that reflects the specimen batch.

### About Customizing Test and Test Run Names
You can define custom names to tests or test runs by assigning the TestRun or TestRunName system variables to the Assign Variables activity and adding a calculation to define text for the variable. In addition, you can use the Input Variables activity to allow the operator to enter text for these variables during the test by adding the input variable to the calculation (see the example that follows). Custom test and test run names are shown on the Review tab and in reports, and in the case of custom test names, in test selection lists.
**Note:** If the calculated or assigned name is already in use or cannot be used for some other reason, the default name is used in place of the custom name, and an error providing more information is issued.

**Customize the Test Run Name with Operator Entry**

**Note:** An alternate way of customizing TestRun and TestRunName is to assign calculations directly in variables, and to trigger the calculation with the Calculate Variables activity in the test flow.

1. Add an **Input Variables** activity at the beginning of the **Run** section of the workflow.
2. In the **Properties** panel, click (+) in the **Variable** list to add a new item.
3. Click the **Information1** variable, then click the right arrow to add it to the **Selected Variables** list and click **OK**.
4. Click the **Variables** tab, locate the **Information1** variable, and change the **Display Name** to “Enter Test Run Name”.
5. Add an **Assign Variables** activity after the **Input Variables** activity to the **Run** section of the workflow.
6. In the **Properties** panel, click (+) in the **Variable** list to add a new item.
7. Click the **Test Run Name** variable, then click the right arrow to add it to the **Selected Variables** list and click **OK**.
8. Click the **TestRunName** variable in the **Variables List** to show the Calculation Editor window.
9. Add the following calculation: "**TestRunName=TestInfo1**", ensuring the validation is successful.

**Customize the Test Name without Operator Entry**

**Note:** An alternate way of customizing TestRun and TestRunName is to assign calculations directly in variables, and to trigger the calculation with the Calculate Variables activity in the test flow.

1. Add an **Assign Variables** activity to the **Set Up** section of the workflow.
2. In the **Properties** panel, click (+) in the **Variable** list to add a new item.
3. Click the **Test Name** variable, then click the right arrow to add it to the **Selected Variables** list and click **OK**.
4. Click the variable in the **Variables** list to show the Calculation Editor window.
5. Add the following calculation: **TestName="Desired Test Name"**, ensuring the validation is successful.
Modifying Tests

Adding and Configuring Test Activities

Test Activity Sequence
Test activities execute from top to bottom as they appear on the Test Editor. To execute activities simultaneously, add Parallel Path activities to the test flow. You can also add While Loop, If-Else Condition, and Periodic Time Event activities to control the test flow.

Test Activity Property Definition
To define the properties of an activity, select the activity icon on the Test Editor, and then select the Properties panel. You can choose to define properties as fixed values or variables.

Activity Property Example

Error Indicators Assist Property Definition
When you add an activity, red error icons appear next to the properties fields when information is missing or is incorrect.

Example
The example in the following figure shows the error indicators that appear when you initially add a Data Acquisition activity to the test editor. A similar message also appears in the Error List on the main window. In this case, all of the error messages disappear when you select a signal and a trigger.

Error Indicator and Help for the Data Acquisition Activity
Skipping Command Segments or Stopping a Test Prematurely
When you run a test, the Test-Run Display shows the progress of the test on a chart (such as the Array-Variable chart used in the sample test). While developing a test, it may not be practical to allow the entire command to play out while iterating between modifying and running a test.

To speed up iterations, you can use the **Wait for Operator Action** activity. Place a **Wait for Operator Action** activity in a parallel path with the command activity. When you run the test and the command activity is executing, a button appears on the Test-Run Display that allows you to skip to the next activity.

**Wait for Operator Activity in Parallel with a Command Activity**

Another option is to add the **End Test** activity which forces the test run to end before completing all stages of the activity. When the test procedure reaches the **End Test** activity, workflow activities that follow and are parallel to the **End Test** activity do not execute, and the procedure immediately proceeds to the end of the test. A message is written to the test run log indicating that the test run was stopped due to the **End Test** activity.
About Disabled Resources
To add versatility to templates, you can include variables, activities, and resources that you can disable or enable depending on the specific type of test you wish to perform. This allows one test procedure to contain resources for the various functions, while allowing the operator to “turn off” portions of the test procedure as needed. Typically, a combination of variables, activities, and resources might all need to be disabled in order to effectively turn off a portion of a template.

If a disabled resource is referenced by an enabled resource, variable, or activity, a validation error occurs for that reference, and the test will not run.

Disabled resources appear grayed out on the Resources tab, and in resource lists in other areas of the application.

Note: Disabled calculations may be referenced in the Additional Calculations list of a data acquisition activity without causing a validation error. In these cases, the disabled references are ignored during a test run.

Disable Resources
1. On the Resources tab, locate the resources that you want to disable. If a validation error occurs, use the associated error icons to help locate resources that should be disabled.
2. Right-click the resource and select Disable (resource).

Considerations for Test Sections
Set Up Section
The Set Up section is for activities you want the operator to perform before test runs begin, such as:

- Entering values for test variables common to all test runs
- Configuring the load train (setting up fixtures, and so forth.)
- Entering the test name, specimen geometry, lot number, batch name, and so forth.

The Set Up section runs only once per test, before the first test run.

Run Section
The Run section performs test runs. Operators typically perform the following for each test run:

- Remove the specimen from the previous test run
- Enter values for nominal specimen dimensions
- Install the new specimen
- Observe the test as forces are applied to the specimen
- Respond to a message that indicates the frame will move to the return position
- Observe the test run information shown on the Review tab
Finish Section
The Finish section is for activities you want the operator to perform when test runs are complete, such as:

- Generate and print reports
- Export data
- Archive data

Note:
It is typical to design tests so that the Finish section runs when test runs are complete. However, running the Finish section does not end the test. If the operator continues to click the Run button after the Finish section is complete, new test runs will be added to the test.

Note:
When using a Run Report activity in the Finish section, use the report template for tests. Do not use the report template for test runs, which contains test run variables not available to the Finish section.

About Running the Finish Section and Ending Tests
An operator can run the Finish section in one of two ways:

- After a test run is complete and the Review page appears (by default, the Review page appears automatically), the operator can click the Run Finish button. This is the manual way to run the Finish section.

Run the Finish Section Button on the Review Tab

- You can enter a value for the Maximum Test Runs variable (Define tab > Variables subtab > Maximum Test Runs). When the number of test runs equals this value, the test will run the Finish section. By default, this value is set to 9999.
Internal Name and Display Name

Internal Name

Generally, this is a name that is used internally to the application or by a calculation. These names do not usually contain spaces and have character restrictions. In some cases, internal names are assigned by the application and cannot change, and in other cases they are defined by the user and can change. Because other parts of the system use the internal name, some additional restrictions are in place to make sure that changing the internal name does not cause the program to fail. In some cases (such as with variables), the name is adjusted where it can be, but in others (like test resources) it will not allow the change to take place. These strings do not change when the language is changed.

Display Name

This is a display version of a string. It has some restrictions on allowed characters, but is much more open than the Internal Names. These strings can be changed by the user without changing any other part of the application. Some character restrictions apply, and some validation is done so two items at the same level do not have the same name. These strings are translated and will change based on language if it was saved in multiple languages (such as with templates).

Configuring What Is Displayed after Test Runs

By default, the application shows the Review tab after each test run. This is set in the Test Definition subtab of the Define tab.

If you disable this control, when a test run is complete the application shows the test input variables on the Monitor tab in preparation for the next test run.

Interacting with the Operator During Tests

You can add information for your operator, such as hardware installation instructions or safety information, directly into your test design. This allows you to create the information that the operator is exposed to during the test, and control where the information is shown in the test flow.

To do this, add special test activities and run-time devices to your test.

- The CustomMessageWindow activity allows you to show a custom message to the operator anytime the test is running.
- The LaunchExternalApplication activity allows you to show documents that require external applications to launch, such as word processing, spreadsheet, and video documents.
The CustomHelp runtime display device allows you to show custom help information, such as files in the CHM format, that you create for the test.

Run-Time Markers
Run-time markers are user-initiated markers used to mark an event that occurs while a test segment is performed. For instance, if a specimen begins to tear before it fails, the operator can assign a run-time marker to mark the point when the tear began. To use run-time markers, you must modify your test. Run-time markers are typically set by a digital input triggered by one of the F keys (F1 or F2) on the handset. You can also use hard-wired digital inputs to trigger a marker.

Batch Mode Tests
You can create batch tests that automatically perform test run loops without the operator clicking the Run button for each test run, and can run unattended.

To do this, you must put the Run section in a while loop by:

- Adding a **While Loop** activity after the Run section
- Dragging the Run section into the while loop
- Disabling any activities (such as Custom Message Activities) that require operator response

A potential problem with this design is stop detection. If stop detection is present and the operator clicks the Stop button during the test, the stop detector triggers an end of the Run section. The while loop then restarts the Run section and creates a new test run. However, since the test is already stopped, the stop detector immediately ends the Run section, causing the process to repeat itself until it has performed all the test runs. This functionally locks the user interface.

To remedy this, you must provide an exit from the loop if the stop state is detected. The simplest thing to do is to add the following expression to the conditional path of the **While Loop** activity:

```
(Signal("Run/Stop") != 0)
```

This causes the **While Loop** activity to terminate if the stop state is present when the Run section ends.

Pre-Allocating Multiple Test Runs
If you are testing multiple specimens, you can improve the speed and reliability of your testing by pre-allocating test runs for each specimen. When you pre-allocate test runs, you can specify unique variable values for each specimen prior to running tests.

Both test designers and test operators can pre-allocate specimens. The strategy you use to implement pre-allocation depends upon your lab’s workflow. The following two examples illustrate scenarios in which either the test designer or operator would use the pre-allocate feature:

- When designing a test, the test designer knows that 10 specimens will be tested. The test designer also knows the properties of each specimen. So, the test designer pre-allocates 10 test runs and enters the variable values for each specimen. The test designer then changes the names of the test runs to Specimen 1, Specimen 2, Specimen 3, and so on for each
Design Guidelines

specimen. Finally, the test designer marks each specimen with the corresponding number. When the operator receives the test and batch of specimens marked 1-10, he or she selects the appropriate specimen for each pre-allocated test run.

- The operator often receives batches of specimens that have varying properties. Because of this, the lab has decided that it is not efficient for the test designer to spend time designing a new test for each batch of specimens. So, when the operator receives a new batch of specimens, her or she pre-allocates test runs for each specimen based on the specimen properties that were recorded on a data sheet or USB flash drive that was sent with the batch of specimens. Additionally, because it is possible to open the Review tab while a test is in progress, the operator runs the first pre-allocated test run and then works on configuring the remaining test runs while the first test run is in progress. To pre-allocate multiple test runs, perform the following:

1. On the Review tab, click the Actions button.
2. Select Preallocate.
3. In the window that appears, enter the number of test runs you want to pre-allocate.
4. Click OK. The top-most table on the Review tab is populated with the pre-allocated test runs.

   Note: When you pre-allocate test runs, each test run uses the test’s default variable values. When you select a test run, you can edit the variable values for each test run in the lower table of the Review tab.

When you click the green Run button, the top-most test run in the list begins. When it completes, you can click the green Run button again to begin the second test run in the list, and so on until you reach the last test run at the bottom of the list. The names of test runs that have not yet been run are italicized. To run a specific test run as opposed to running tests in the list from top-to-bottom, right-click a test run and select Run Test.

   Note: If the test definition is changed during test design (for example, variables, resources, and workflows are edited), any changes made will not be applied to existing pre-allocated test runs. In this case, the test designer must delete the existing pre-allocated test runs and pre-allocate new test runs.

Working with Variables

Variable Concepts
A variable is a place with a name in which you can store one or more values. Variable types and examples used in the test include:

- Simple variables (also referred to as scalar variables) store either a value or a string.
- Array variables store an array, which is an ordered collection of numbers or strings. You can refer to individual values in an array by specifying an index.
- Calculated variables are variables (simple or array) that include a mathematical expression.
Adding and Editing Variables
You can add or edit variables from within the various variable editing windows and editors (such as the Map Variables window or the Calculation Editor). You can click the Add Variable (plus sign) or Edit Variable (...) icon or right-click and select New Variable. To edit a variable, right-click a variable name and select Edit Variable.

Benefits of Using Variables

Recommended
Variables are central to the utility of MTS TestSuite applications. It is good practice to use variables in tests whenever possible.

- Variables allow information to be shared between different areas of the test.
- Variables allow you to define test input values from a central location.
- Variables allow you to apply calculations to values and show them in charts and graphs.

Common Use Cases for Variables
The following common use cases illustrate the use of variables in test design:

- Create a rate variable, then add an operator entry activity to the workflow in which the operator enters a rate value during the test, and use this variable value in the command activity to control rate.
- Create a variable for the area of specimen, and use that variable when calculating stress from load.
- Acquire data mapped to variables from which modulus is calculated, then include modulus on a report.
- Place a marker at the yield point on the graph on the Review tab, and use the marker variable in a report.

Using Variables in Activities
MTS TestSuite applications allow you to define many of the individual properties of test activities as variables.

Normally, when you enter a value for a property in an activity, you can change the value only when the test is stopped and the test procedure is unlocked. In this sense, the property is “fixed.”

With the use of variables, you can associate a property of an activity with a variable. This allows you to change the variable value without selecting the activity.

Variable Toggle Button
The Numeric-Variable or Item-Variable toggle button appears throughout the application. Click the button to toggle between two entry options. The following table indicates the icon and its meaning.
### Design Guidelines

**Icon** | **File Type**  
--- | ---  
| ![Variable selection](image) | Variable selection; click to select or enter a variable  
| ![Item selection](image) | Item selection; click to select an item from the drop-down menu.  
| ![Direct numeric entry](image) | Direct numeric entry; click to enter a numeric value.  

### Variable Example

Suppose you want the example test to perform the ramp in the **GoTo + DAQ + Detection** activity at a different rate for some tests, and at the original rate for others. Rather than manually entering the new rate for the ramp each time you want to use it, you could create a new variable with the new rate and select the new variable when desired.

To do this, you would perform the following:

- Locate the **Rate** property on the property editor (1)
- Click the **Test Rate** variable and select `<New Variable>` (2)
- On the Edit Variables window, create a new variable for the desired test rate. In the example that follows, the new test rate has a display name of "Test Rate 2", and a default value of 2 mm/sec (3).
Manipulating Variables in the Workflow

The TWE application includes test activities you can use to manipulate variables in the workflow, as follows:

- Assign Variables activity
- Calculate Variables activity
- Input Variables activity
- Read data activity
- Write data activity

You can use these activities in a variety of ways to manipulate variables. For example, you can use an Input Variables activity to allow the operator to change variable values while the test is running.
Assigning Variable Availability
Variables have the following Availability attributes:

- Preallocate
- Pretest
- During Test
- Editable Post Test
- Result

The Pretest, During Test, and Editable Post Test attributes specify when the operator can change the variable during the test.

Preallocate Attribute
Variables with this attribute can be pre-defined by a test designer. This allows the value of the input variable for individual test runs to be entered all at once for a batch of specimens, for example.

Pretest Attribute
Variables with this attribute can be edited before the test is initialized and the test runs begin. They appear on the Monitor tab and can be changed before the test is started.

During Test Attribute
Variables with this attribute can be edited in any of the test sections (Set Up, Run, or Finish) after the test is started.

Editable Post Test Attribute
Variables with this attribute can be edited in the Variables table on the Review tab after the test is complete.

Result Attribute
The Result attribute specifies the variables that appear in the Result table in the Review tab. Variables shown in the Results table are not editable.

Variables with Results and Editable Post Test Attributes
Some variables include both Editable Post Test and Results attributes. In this case, the variable appears in two locations on the Review tab:

- In a non-editable form in the Test Run Results table
- In an editable form in the Variables table

Default Availability of New Variables
By default, new variables are assigned to the Variables category, with an availability of During-Test, and Editable Post Test.

Adding Calculations to Variables
You can add mathematical expressions to variables to create calculated variables. To do this, use the Variables subtab of the Define tab to add calculations.
**Recommended**
The application always performs calculations in base (system) units. To avoid unit-conversion errors when adding a calculation to a variable, create variables to hold constants used in the calculation. Assign the desired units to each variable and the application will convert the unit type to the base unit type when performing the calculation. For more information, see the *MTS TW Elite User Guide*.

**Managing Variables**
Managing variables is important when designing tests, especially for complex tests. The Variables subtab of the Define tab has a Variables Actions button that includes the following display options to assist with managing variables:

- Show All Variables
- Show/Hide Read-Only Variables
- Show/Hide Disabled Variables
- Rename Category
- Update Used By
- Delete Unused
- Set Variable Order

The Variables subtab toolbar also includes a search tool.

**Update Used By Feature**

*Note:*
To show the Used By column, click the Variable Actions button and then click Update Used By.

**Variable Subtab of the Define Tab**

![Variable Subtab of the Define Tab](image)
### Description of Variable Tab

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Search tool</td>
</tr>
<tr>
<td>2</td>
<td>Variable Actions button</td>
</tr>
<tr>
<td>3</td>
<td>Select Update Used By to show the Used By column</td>
</tr>
</tbody>
</table>

#### Used By Table Column

The Used By table column shows where and how many times each variable is used in the test. This is useful for:

- Editing variables used by more than one activity. If a variable is used by multiple activities, you may want to create additional variables to manage the unintended changes.
- Troubleshooting complex tests.
- Deleting variables. If you delete a variable that is in use, validation errors occur where the deleted variable is referenced.

This column includes a drop-down list that shows where the variable is active. To locate where the variable is used in context of the test, click the entry, and the context is shown. For example:

- For a calculation, the entry for the calculated variable becomes selected.
- For an activity, the workflow appears with the activity selected.
- For a monitor view, the monitor display appears when you select the view.
- For data acquisition, the Data Acquisition window appears with the corresponding entry selected.

#### Configuring Data Acquisition

**Data Acquisition Properties Page**

For new users, the Data Acquisition properties page may not be intuitive. When setting up data acquisition, you should determine the properties you need in advance and access them in the properties page as required.

**Mapping Variables to Acquired Data**

**Recommended**

In almost all instances, you should choose to map variables to acquired data. The default selection to the **Save data to variables?** control is set to **Yes, automatically map variables**.
If you choose not to map variables:

- You will not be able to use any of the monitor displays in the **Variable Views** category (or other monitor displays that use variables). You will only be able to use monitor displays in the **Signal Views** category, such as the Signal Meter and Signal Scope.
- The charts and tables in the **Review** tab, which are based on variables, will not contain useful information.
- You will not be able to show test data in charts and graphs using the MTS Reporter Add-in for Excel.

**Acquiring Data in a Continuous Stream**

For typical tensile tests, you configure data to be acquired in a continuous stream using the **Go To + DAQ + Detection** or **Dwell + DAQ + Detection** activities.

For “multi-cycle” tests, you configure data to be acquired in blocks.

**About Blocks**

Block boundaries are typically divided according to command content. This is sometimes loosely referred to as “cycles” of data, though the blocks could be a group of ramps, ramps and cycles, or a custom waveform.

**Acquiring Data in Blocks Without Mode Switching**

If you want to acquire data in blocks without performing control mode switches within the blocks, you can use either the **Custom Waveform + DAQ** or **Cycle + DAQ** activities.

**Acquiring Data in Blocks With Mode Switching**

If you want to acquire data in blocks that include command elements that switch control mode within the blocks, you need to use the generic **Data Acquisition** activity in parallel with a **While Loop** activity that includes the command activities and increments the block count.

In this scenario, you must set up custom blocking using the **Configure** properties. To emulate the automatic blocking performed by the **Custom Waveform + DAQ** or **Cycle + DAQ** activities, you need to configure a variable that counts two segments (one cycle equals two segments).

**Setting Buffer Size**

You choose the buffer size by clicking the **Advanced** button on the Data Acquisition Properties page.

The default buffer size is set to 100. There are instances when this is too large. Symptoms that the buffer size is too large include:

- Calculations do not run in a timely manner.
- Monitor displays do not update as often as necessary.
Design Guidelines

Recommended
While many factors determine ideal buffer size, observe the following:

- When acquiring data in a continuous stream, it is good practice to set the buffer size relatively high, so that the buffer fills every 0.1 to 1.0 second.
- When acquiring data divided into custom blocks, it’s good practice to set the buffer size relatively low, so that it accommodates the anticipated size of the blocks.

Data Acquisition for Test-Run Displays
In the Test-Run Display Toolbox, the Cycle View category pertains only to data that is divided into blocks.

This is an important consideration, because if you attempt to use one of these display devices to view data that is not blocked, the view will appear blank, because the device is waiting for the first block boundary.

For example, if you want to view hysteresis by examining load versus displacement in a test that does not use blocking, you can use:

- The signal Scope (Signal Views category)
- The Array-Variable Chart (Variable Views category) if you map data to variables when configuring data acquisition

If you use either the Custom Waveform + DAQ or Cycle + DAQ activities (which provide automatic blocking), or configure the generic Data Acquisition activity for blocking, then you can also use the Hysteresis Chart (Cycle Views).

Polarity Considerations for MTS Criterion Series 40 Systems
When designing tests, it is important to be aware of the various settings that affect the polarity of your system. The combined effect of these settings control:

- The direction that the crosshead will move when command activities (such as Go To or Cycle) are run.
- Whether the values in meters, scopes, and data acquisitions either increase (become more positive) or decrease (become more negative).

For example, you can customize a tension test so that when the crosshead applies tension (that is, pulls the specimen apart) the measurements in meters, scopes, and data acquisitions become either more positive or more negative.

Important:
When adjusting the polarity settings of your system, always use a disposable specimen that you are comfortable with destroying. Modifying the polarity settings of a test may cause unexpected crosshead movement.

For example, consider the following analogy in which you have designed a prototype of a car and want to submit it to a crash test. The car will crash into the back wall if you do not shift the transmission from reverse to forward even though you have designed the prototype of the car
perfectly. As a result, your prototype will be destroyed and you will not be able to collect any crash test data.

In this analogy, shifting the transmission of the car is similar to changing the polarity of a float signal because you can effectively reverse the direction that the crosshead moves during a test run. Because of this, it is important to validate the polarity settings with a disposable specimen to ensure that you receive the anticipated crosshead movement before you test real specimens.

When designing a new test, the following settings that affect polarity are available on the Resources subtab of the Define tab:
Settings that Affect Polarity

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Upper and lower workspace selection. These buttons are only available on test systems equipped with dual workspaces. These buttons allow operators to easily switch between different test zones without manipulating the test resources or using a different test that has unique polarity settings for different workspaces. For example, consider a scenario in which you have designed and validated a test that you expect will be run in the lower workspace and the Polarity of all float signals used by the test are set to Normal. To run the same test in the upper workspace, the operator would need to change the polarity setting of all float signals in the test that measure values relative to the bottom of the frame from Normal to Inverted in order to achieve the appropriate test results. By doing this, the operator would be changing the test and a future operator may not be aware of these changes. In contrast, using the workspace selection buttons only requires the operator to select the upper workspace in order to achieve the same result when changing workspaces. When the operator selects the appropriate workspace, he or she achieves the desired test results without having to modify the test. As a result, this allows the operator to run a single test using either the upper or lower workspace.</td>
</tr>
<tr>
<td>2</td>
<td>Polarity selection. Use this setting to change the polarity of a float signal. Changing this setting affects the Effect of Increasing Extension setting. Available settings are Normal and Inverted.</td>
</tr>
<tr>
<td>3</td>
<td>Effect of Increasing Extension selection. This setting defines the relationship between the crosshead/displacement channel and each float signal.</td>
</tr>
<tr>
<td></td>
<td>• Select Increases Value when an increase in the crosshead or actuator extension will increase the value of the signal.</td>
</tr>
<tr>
<td></td>
<td>• Select Decreases Value when an increase in the crosshead or actuator extension will decrease the value of the signal.</td>
</tr>
<tr>
<td></td>
<td>• Select Indefinite for signals (such as time) where there is no direct relationship to the crosshead or actuator extension.</td>
</tr>
</tbody>
</table>

Configure Settings That Affect Polarity

When designing your test, use the following procedure to ensure that the settings that affect polarity are appropriate:

1. Import the resources that are required by your test.
2. If your load frame has upper and lower workspaces, select the desired workspace.
3. For each float signal, select the polarity that produces the desired output.
4. For each Float Signal, select the desired Effect of Increasing Extension. You can determine the appropriate setting by adding a meter for each signal, installing a disposable
specimen, increasing extension, and then observing how the signal output changes.

Note:
All command activities in your test that have the **Direction** property set to Auto will use the **Effect of Increasing** setting to determine which direction the crosshead must travel to reach the termination condition specified in the activity.

If the **Direction** setting of a command activity is either Increase or Decrease, the crosshead will move in a direction that will either increase or decrease the control-mode feedback signal value regardless of whether that movement will ever satisfy the activity’s terminal condition.

**User Scenario: Configuring Polarity for a System Equipped with Dual Workspaces**

For example, consider a scenario in which you want to design a tension test that uses a system equipped with dual workspaces. While most of your specimens will break when approximately 5 kN of force is applied, you know that some specimens will break when less than 1 kN of force is applied.

Instead of simply installing the 10 kN load cell into the lower workspace and only using the lower workspace to test all of your specimens, you install a 1 kN load cell in the upper workspace for use when testing the weaker specimens. Now, you can test the weaker specimens in the upper workspace and take advantage of the improved resolution that the smaller 1 kN load cell offers.

Next, you want to ensure that your polarity settings are correct. First, you import all unused resources into the test. Since you are designing the test in the lower workspace, you ensure that the lower workspace is selected. Because you are performing a simple tension test and you want the values in the scope and data acquisitions to become more positive as you apply tensile force, you leave the **Polarity** setting at the default value of Normal and the **Effect of Increasing Extension** at the default value of Value Increases. However, if you wanted the load signal to become more negative as tensile force is applied, you would change Polarity to Inverted. Then, because you changed the polarity to inverted, you would need to change the **Effect of Increasing Extension** on the Load float signal to **Decreases Value**.

When you are finished designing the test, you want to validate your test to ensure the results are what you expect. So, you insert a disposable specimen into the lower workspace and run the test. To ensure that the upper workspace produces the desired output, you select the upper workspace, insert a disposable specimen, and then run the test and observe the result.

**Debugging**
This section contains considerations and techniques test designers should be aware of when creating, testing, and debugging test procedures.

**Automatic Saves and Experimentation**
When you run a test, the application performs an automatic save. If you make changes to a test that you want to delete after you ran a test, you will have to manually back out of the changes.

**Tip**
If you want to experiment with a test, perform a Save As operation.
Rapid Design Iterations

Running the Set Up Section without Starting a New Test
The Set Up section runs only once per test before the first test run. All subsequent test runs in the test skip the Set Up section. If you are developing a test and need to run it from the beginning to the end iteratively, this can be inconvenient, because it requires you to open a new test after each iteration.

Tip
To run the Set Up section after performing test runs, delete all test runs (on the Review tab, right-click a test run, and then select Delete > (All)).

Activity State in Outline View
The outline view (Define > Procedure > Outline) provides a compact view of the test hierarchy. Use this view to check nesting levels and disable and enable activities.

Test Procedure Test-Run Display
The Test Procedure monitor display shows the test procedure during runtime. It allows you to follow the progress of the test by identifying the active test activity, which is shown with a green triangle. Completed activities are identified with a blue check mark.

Runtime Values Tab
The Runtime Values tab updates variable values as the test is running. Array variable values include array size, first value, and last value.
Modify the Example Test

About This Chapter ................................................................. 80
Modifying Tests ....................................................................... 80
Design Scenario ....................................................................... 81
Modify the Example Test

About This Chapter
This chapter contains instructions for performing modifications to the Example MTS Tension (Simplified) Test supplied with the TWE application. While the modifications are unlikely to result in a test that you will use in your facility, the process of making the modifications will help you learn how to use the application.

The example test is reviewed in detail in “Examine the Example Test” on page 23. You will apply the modifications presented in the “Design Scenario” on page 81.

The modifications applied to the example test follow the guidelines presented in “Design Guidelines” on page 55.

Note:
To benefit from this chapter, it is not necessary to perform the instructions precisely. Even if you just review the information without using your software, you will still learn the fundamentals of how to modify test templates.

Prerequisites
Start the TW Elite application in simulation and open the Example MTS EM Tension (Simplified) test.

Practice in Simulation
MTS recommends that new users perform the tasks in this chapter in simulation. While it is possible to perform these tasks on any MTS system equipped with the proper resources, it is good practice to learn about the application in simulation.

For More Information
“Examine the Example Test” on page 23
“Design Scenario” on page 81
“Design Guidelines” on page 55

Modifying Tests

Methods Used to Modify Tests
To add capability to a test created from a template, you typically perform one or more of the following:

- Enable disabled activities (templates typically include disabled activities that extended functionality when enabled)
- Import unused resources (Define > Resources > Import Resources)
- Change variable options associated with the assigned Formula Assistant File
- Add activities to the workflow
- Change settings in the various tabs in the Configuration window (Preferences > Configuration)
Methods Used to Modify the Example Test

The procedure for modifying the example test includes most of these methods. The procedure does not include information about optional controls. For more information about user controls, see the MTS TestSuite TW Elite Users Guide (Press F1 while the TWE application is open, or see Start > Programs > MTS TestSuite > Documentation > TW Elite Users Guide).

Note:
The process of test design is iterative. It is good practice run the test frequently while making modifications.

Note:
You may want to run the modified test with the TW Express application to validate the operator experience.

Design Scenario

For learning purposes, suppose you need to modify the example test according to the following requirements:

- The operator needs to switch the specimen geometry type from rectangular to round.
- The operator needs to enter the specimen batch name at the beginning of each test, but not for every test run.
- The operator needs to view a video you have created for installing the specimen in the fixture at the beginning of each test. You also want the operator to be able to exit the video prematurely if a review is not needed. You want this video to be available only before test runs begin.
- The operator needs to be informed to stay clear of the test fixture before the frame moves before each test run.
- The test needs to ramp in displacement control at 3 mm/s not to exceed an ending condition of 1 kN.
- The test needs to slow down to minimize overshoot of the ending condition.
- The test needs to automatically generate a test report and send it to a manager when twenty four test runs have been performed.

Modifying the Set Up Section

Requirements

- The test needs to use geometry type of round instead of rectangular.
- The operator needs to enter the specimen batch name at the beginning of each test, but not for every test run.
Modify the Example Test

** Modifications to the Set Up section 
- Change the specimen geometry type. To do this, you will change the Area variable option of the Formula Assistant File from rectangular to round.
- Add an **Input Variables** activity to the Set Up section, enter a message to prompt the operator for a batch name, and assign the name to a variable.

**Formula Assistant File**

**Overview**
The Formula Assistant File provides a base set of system variables, calculations, and functions associated with a specific type of test. The example test uses a formula assistant file labeled “Electro-Mechanical Formula Assistant”.

When a test variable is associated with the Formula Assistant File, an **Option** list becomes available in the **Calculation** panel for the variable.

**Warning:**
Mouse clicks are queued, so the machine may continue to move after you have stopped clicking.

Unexpected machine movement can cause injury or death to anyone who is near the machine.

Make adjustments only as fast as the machine can respond. To stop the machine if adjustments are queued, press an Emergency Stop button. Do not click the mouse or the window controls faster than the machine can respond.

A common scenario for changing a variable calculation option with the Formula Assistant is the Area variable in the Specimen Geometry Category. In this example, the Area variable in the MTS EM Simplified Tension Test is changed from a Rectangular to a Round calculation for a specimen.

**Change the Specimen Geometry**
To change the specimen geometry:

1. Open the MTS EM Tension (Simplified) template.
2. Click the **Variables** sub tab under the **Define** tab. The **Area** variable is currently set to calculate a Rectangular specimen, which is Width * Thickness as shown below.
3. Scroll down and notice that Width and Thickness are also variables in the Test Definition.

You can also view the current calculation of the variable in the formula box of the **Calculation** panel.

4. Select Round from the **Option** list and click **Apply**. Notice that the Pi() formula to calculate a round area has automatically been populated for you.

After you click **Apply**, the formula is replaced in both the **Calculation** and the **Variables** panels.

5. Observe that the variables used for calculating a rectangular area (Width and Thickness) are automatically removed from the **Variables** panel, and the variable for Diameter is automatically added to the **Variables** panel.
Modify the Example Test

Enable the Input Variables Activity

Overview
This task enables an **Input Variables** activity to the workflow and defines its properties.

Procedure

| Note: | The Set Up section runs only once per test, before the first test run. Subsequent test runs in the test skip the Set Up section. Because of this, to test the runtime effect of this modification, you must delete all test runs. To do this, click the **Review** tab, right-click the name of any test run, select **Delete > All**.

1. Right-click the **Input Variables** activity in the Set Up section, and select **Enable**.

2. Enter the following information in the **Properties** panel:

   **Input Variables Activity Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>Enter “Enter Batch Name”</td>
</tr>
<tr>
<td>Message</td>
<td>Enter “Enter the batch name as it appears on the label on the box of specimens.”</td>
</tr>
<tr>
<td>Variable List</td>
<td><strong>Note:</strong> Click the plus sign and use the Variable Selection window to add variables. Select <strong>Test ID</strong> from the Available Variables list, and then click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>

Runtime effect
The TWE application shows the prompt in a window before the first test run when the operator starts the test (clicks the Run button). The window includes a text entry field and an **OK** button. When the operator enters a batch name and clicks **OK**, the application captures the entry in the **TestID** variable.

Modifying the Run Section

Requirements

- The operator needs to be informed to stay clear of the test fixture before the frame moves before each test run.
- The test needs to ramp in displacement control at 3 mm/s not to exceed a 1 kN load level.
- The test needs to slow down to minimize overshoot of the ending condition.

Modifications to the Run Section

- Modify the **Test Rate** variable from the **Go To + DAQ + Detection** activity.
- Add a **TerminationCondition** of 1 kN to the **Go To + DAQ + Detection** activity.
- Add a **Braking Distance** of 100 N to the **Go To + DAQ + Detection** activity.
Modify the Example Test

Edit the Go To + DAQ + Detection Activity

Overview
This task changes properties of the Go To + DAQ + Detection activity, which applies forces to the specimen by moving the frame. You will change the Test Speed variable, enable and configure the Termination Condition control, and enable and configure the Braking Distance control.

Procedure
1. In the workflow, click to expand the Run section and the Parallel Paths activity.
2. In the left-most branch of the Parallel Path activity, click the Go To + DAQ + Detection activity to show its properties.
3. In the Properties panel, enter the following:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>Enter “Apply Program Command”</td>
</tr>
<tr>
<td>Rate</td>
<td>Right-click Rate and select Edit to show the Edit Variable window. In the Edit Variable window, change the Default Value to 3 mm/s.</td>
</tr>
<tr>
<td>Termination Condition</td>
<td>Enable (check)</td>
</tr>
</tbody>
</table>

Note: The controls that follow are in the Go To tab.
### Modify the Example Test

**Runtime Effect**
Modified properties are configured to ramp the crosshead at a rate of 3 mm/s until either:
- Load becomes greater than 1.00 kN (Termination Condition)
- Specimen failure is detected
- One of the other branches of the Parallel Branch activity (such as a limit detection branch) completes

The **Brake Distance** control provides control-mode braking, which slows the command to help avoid overshoot. The brake value is subtracted from the **Termination Condition** value to determine where control-mode braking starts. In this case, the **Termination Condition** is 1.00 kN and the **Brake Distance** is set to 100 N, at 900 N the command slows to 10% of the **Test Rate** (0.3 mm/s), and at 990 N, command slows to 1% of the **Test Rate** (0.03 mm/s).

### Modifying the Finish Section

**Requirements**
The test needs to automatically generate a test report and send it to a manager via e-mail when 24 test runs have been performed.

**Modifications to the Finish Section**
- Change the **Maximum Test Runs** variable to 24.
- Enable the **Run Report** activity and select the **Send report in e-mail** option.

### Change the Maximum Test Runs Variable

1. Click the **Variables** subtab (Define tab > Variables subtab).
2. Select the Maximum Test Runs variable from the list of available variables and, in the Properties panel, change the default value to 24.

**Runtime Effect**
The TWE application automatically runs the Finish section, which contains the Run Report activity, when 24 test runs are complete.

**Enable and Configure the Run Report Activity**

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following task requires e-mail address and server properties to be properly configured. These properties are set in Preferences &gt; Configuration &gt; E-Mail. Check with your e-mail administrator for the correct settings.</td>
</tr>
</tbody>
</table>

**Overview**
This task enables the Run Report activity, which generates a test report. It also enables the e-mail option, which sends a copy of the report to an e-mail address.

**Procedure**
1. In the workflow, click to expand the Finish section.
2. Right-click the Run Report activity and select Enable.
3. Enable (check) the Send report to e-mail control.
4. Enter e-mail parameters as desired.

**Runtime Effect**

When 24 test runs are complete and the TWE application runs the Finish section, a test report is generated using the default test report template (**<Report Templates>\Default Test Report.xltx**.) The test report is saved to the **Report Directory** (Preferences > Configuration > E-Mail). A copy of the report is sent to the e-mail address entered in the **Run Report** activity.
About This Chapter
If you are familiar with MTS TestWorks 4 software, this section will help you make the transition to TW Elite. It provides a side-by-side comparison of the concepts, terminology, and functionality of both applications.

Basic Concepts

TestWorks 4 and MTS TestSuite TW Basic Concepts

<table>
<thead>
<tr>
<th>TestWorks 4</th>
<th>MTS TestSuite TW</th>
</tr>
</thead>
<tbody>
<tr>
<td>You choose a method to test a collection of</td>
<td>You choose a template to test a collection of</td>
</tr>
<tr>
<td>physical specimens referred to as a sample.</td>
<td>physical specimens.</td>
</tr>
<tr>
<td>To open a method, you typically select <strong>Open</strong></td>
<td>To open a template, you typically select <strong>New Test</strong></td>
</tr>
<tr>
<td><strong>Method</strong> from the <strong>Method</strong> menu. When you do</td>
<td>From Template on the <strong>File</strong> menu. When you do this,</td>
</tr>
<tr>
<td>this, you create a container file called a sample</td>
<td>you create a container file called a test that is</td>
</tr>
<tr>
<td>that is used to collect data and contains a copy</td>
<td>used to collect data and contains a copy of the</td>
</tr>
<tr>
<td>of the method.</td>
<td>template.</td>
</tr>
<tr>
<td>Each time you run the method on a physical</td>
<td>Each time you run the template on a physical</td>
</tr>
<tr>
<td>specimen, the application creates a specimen.</td>
<td>specimen, the application creates a test run.</td>
</tr>
<tr>
<td>The specimen is a data set that populates the</td>
<td>The test run is a data set that populates the test.</td>
</tr>
<tr>
<td>sample. When you run the method against all of</td>
<td>When you run the template against all of the</td>
</tr>
<tr>
<td>the physical specimens in the sample, the</td>
<td>physical specimens in the collection, the test is</td>
</tr>
<tr>
<td>sample is complete.</td>
<td>complete.</td>
</tr>
</tbody>
</table>

Terminology Differences

TestWorks 4 and MTS TestSuite Terminology Differences

<table>
<thead>
<tr>
<th>TestWorks 4 Term</th>
<th>Definition</th>
<th>MTS TestSuite TW Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specimen (1)</td>
<td>The physical part being tested.</td>
<td>Specimen</td>
<td>The physical part being tested.</td>
</tr>
<tr>
<td>Specimen (2)</td>
<td>A single instance of running a method. The data and results of a specimen</td>
<td>Test Run</td>
<td>A single instance of running a template (a template is similar to a method,</td>
</tr>
<tr>
<td></td>
<td>that is stored inside a sample file, along with other specimens that have</td>
<td></td>
<td>see Template below). The data is stored in a test file with a copy of the</td>
</tr>
<tr>
<td></td>
<td>been tested under this sample name.</td>
<td></td>
<td>template.</td>
</tr>
<tr>
<td>Sample</td>
<td>A file that contains the data for a group of specimens. It includes a copy</td>
<td>Test</td>
<td>A file that contains test run data (if test runs were</td>
</tr>
<tr>
<td>TestWorks 4 Term</td>
<td>Definition</td>
<td>MTS TestSuite TW Term</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-----------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Method</td>
<td>A file that contains all of the TestWorks 4 settings necessary to perform a specific test. It includes settings for the user interface, machine control, testing procedures, and calculations.</td>
<td>Template</td>
<td>A file that contains all of the MTS TestSuite TW settings necessary to perform a specific test. Templates cannot be edited directly. When you open a test from a template, MTS TestSuite TW creates a test which contains an editable copy of the template. If the template is user-defined, you can edit the copy of the template in the test and overwrite the original template by saving it with the same name (Save As Template function). MTS-defined templates are write-protected. If the template is MTS-defined, you can edit the copy of the template in the test, but you must save it with a unique name and create a new template (Save As Template function).</td>
</tr>
<tr>
<td>Channel</td>
<td>A hardware channel is both the stream of data from the controller and the container of collected data. A software channel is based on calculations from hardware channels or other software channels. Both types of channels yield a stream of data at the system rate that always has a current value and can be measured by meters even when the test is not running.</td>
<td>Signal</td>
<td>Channel functionality is broken into three parts; the data stream from the controller is a resource; the container that holds data is an array variable; and a combination of feedback selection, control algorithm, and tuning parameters is the control mode. Channel resources yield a stream of data at the system rate that always has a current value and can be measured by meters even when the test is not running.</td>
</tr>
<tr>
<td>TestWorks 4 Term</td>
<td>Definition</td>
<td>MTS TestSuite TW Term</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-----------------------</td>
<td>------------</td>
</tr>
<tr>
<td>When data acquisition is specified for a segment, the type of data specified is collected for all channels in an array of the same channel name.</td>
<td>rate that always has a current value and can be measured.</td>
<td>Calculated channels are based on calculations of controller channels.</td>
<td>Data acquisition maps data to a unique array for the specified channels.</td>
</tr>
<tr>
<td>Note: MTS TestWorks 4 does not have a direct equivalent to MTS TW control modes.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Batch</td>
<td>Multiple samples.</td>
<td>Multiple Tests</td>
<td>Multiple tests.</td>
</tr>
<tr>
<td>Calibration</td>
<td>The act of calculating a calibration factor for the device, and then comparing it to the original calibration factor. <em>(Tools &gt; Calibrate).</em></td>
<td>Verify</td>
<td>The act of checking the current calibration factor against a previously determined calibration factor. <em>(Station &gt; TEDS Devices &gt; Verify).</em></td>
</tr>
<tr>
<td>Pause</td>
<td>A test control that suspends a test already in progress. Clicking on <strong>Pause</strong> again resumes the test. Note: The &lt;F4&gt; key on the keyboard performs the same function.</td>
<td>Hold</td>
<td>A test control that suspends a test already in progress. Clicking <strong>Run</strong> resumes the test.</td>
</tr>
<tr>
<td>TestWorks 4 Term</td>
<td>Definition</td>
<td>MTS TestSuite TW Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GoTo</td>
<td>A test segment that moves the crosshead to a specified end-level or condition. It may be set to stop at a specific point, or it may be set to run until a break is detected or a limit condition has been reached. A Data Collection task running in parallel with the GoTo can be configured to acquire data.</td>
<td>Go To + DAQ + Detection</td>
<td>A composite test activity with equivalent functionality to the GoTo segment in the TestWorks 4 application. Note: MTS TestSuite TW also includes a non-composite Go To activity that does not include data acquisition or detection functionality.</td>
</tr>
<tr>
<td>Hold</td>
<td>A test segment that moves the crosshead in order to maintain a specified channel value. It may be set to stop at a specific point, or it may be set to run until a break is detected or a limit condition has been reached. A Data Collection task running in parallel with the Hold can be configured to acquire data.</td>
<td>Dwell + DAQ + Detection</td>
<td>A composite test activity with equivalent functionality to the Hold segment in TestWorks 4. Note: MTS TestSuite TW also includes a non-composite Dwell activity that does not include data acquisition or detection functionality.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Sections</th>
<th>In the Define tab, the Test Flow view of a method includes the following sections:</th>
<th>Test Sections</th>
<th>In the Procedure tab, the template includes the following sections:</th>
</tr>
</thead>
</table>
|                 | • Idle  
|                 | • Pre-Sample  
|                 | • Pre-Specimen  
|                 | • Specimen  
|                 | • Post-Specimen  
|                 | • Post-Sample  | • Setup  
|                 |                                                         | • Run  
|                 |                                                         | • Finish  |
|                 |                                                         | Note: These sections provide the capability of TestWorks 4 test sections as follows:             |
|                 |                                                         | • The Setup section provides the capability of the Pre-Sample section.                           |
|                 |                                                         | • The Run section combines the capabilities of the Pre-Specimen, Specimen and Post-Specimen sections. |
### Functional Differences

**TestWorks 4 and MTS TestSuite TW Behavioral Differences**

<table>
<thead>
<tr>
<th>Function</th>
<th>TestWorks 4 Behavior</th>
<th>MTS TestSuite TW Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performing File Operations</td>
<td>You perform file operations for samples from the File menu. You perform file operations for methods from the Methods menu.</td>
<td>You perform all file operations for tests and templates from the File menu. The File menu includes the following Open and New selections:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Open Test</strong>: Opens an existing test.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>New Test From Template</strong>: Creates a new test that includes a copy of the template without test run data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>New Test From Existing Test</strong>: Creates a new test that includes a copy of the test without test run data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>New Test From File</strong>: Creates a new test from an XML file; typically used in automation.</td>
</tr>
</tbody>
</table>
| Packaging or Exporting    | When you package a method, TestWorks 4 combines the current method and its associated files into one compressed file. The compressed file typically has an extension of “mpk”.  

Packaging makes it easier to distribute a method among many systems.  

Method packages are unpacked using the Method Unpack window.  

|                           | When you export a test, MTS TestSuite TW combines the current test and its associated files into one compressed (and self-contained) file and assigns an extension of “tsproj”.  

Exported tests can be managed with the operating system. Unexported tests must be managed within the MTS TestSuite application.  

When you save a template, MTS TestSuite TW compresses the template and assigns an extension of “TWTemplate”.  

Templates can be managed with the operating system.  

Note: Template files are not fully self-contained; they contain references to other files (e.g., |
<table>
<thead>
<tr>
<th>Function</th>
<th>TestWorks 4 Behavior</th>
<th>MTS TestSuite TW Behavior</th>
</tr>
</thead>
</table>
| **Working with Hardware Resources** | When you add a hardware channel to a method, the hardware channel automatically includes a variable that is configured to acquire data. | When you add a hardware resource to a test, you must also do the following to acquire data:  
  - Add an array variable.  
  - Locate a data acquisition activity and add it to the workflow.  
  - Add the desired signal to the list.  
  - Map the signal to the array variable (automatic by default). |
| **Working with Calculated Resources** | When you add a software channel to a method, the software channel automatically includes a variable which is configured to acquire data. | When you add an array variable to a test, you must also add the array variable to data acquisition calculations. |
|                                  | You can use a calculated channel as a control channel.                             | You must add a signal and make it a calculation. In addition, the calculation of these signals requires the use of the Signal function to access the signal and not the variable. |
| **Acquiring Data**               | You can turn on data acquisition in a segment (such as a GoTo segment).            | You can place a data acquisition activity in parallel with a command activity in the workflow, or you can use the data acquisition functionality within a composite activity, such as the Go To + DAQ + Detection activity. |
|                                  | You can define up to fives types of data to be acquired (peak-valley, timed, level-crossing, etc.) per segment. |                                                                 |
|                                  | The separate data acquisition streams are combined in the channel arrays.           | **Note:** In any instance where a Data Acquisition activity (DAQ) is in a parallel path, the software starts recording data inside the parallel path from left to right. To reduce the delay in recording the first data point, place the first DAQ to the far left in the parallel path. |
|                                  |                                                                                     | By default, acquired data is mapped to array variables.  
  You can specify the type of data you want to acquire for individual signals. |
| **Meters**                       | Meters are always live, regardless of whether the data is based on calculations or not. Calculated | Meters are available as signal meters and variable meters.  
  Signal meters show signal data at all times. |
<table>
<thead>
<tr>
<th>Function</th>
<th>TestWorks 4 Behavior</th>
<th>MTS TestSuite TW Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated Channels with MTS FlexTest Controllers</td>
<td>Calculations for calculated signals are performed in the application (TestWorks 4).</td>
<td>Calculations for calculated signals must be performed in the controller (MTS FlexTest). Converted tests that include calculations formerly performed by TestWorks 4 will generate validation errors. To resolve these errors, test designers must configure the calculated signals with MTS FlexTest software.</td>
</tr>
<tr>
<td>Working with Variable Calculations</td>
<td>When you configure a variable it contains flags which define when it is calculated.</td>
<td>You must place a variable in an appropriate activity to determine when the variable is calculated.</td>
</tr>
<tr>
<td>Creating Markers</td>
<td>You create markers by setting a flag on a variable.</td>
<td>You must go to the chart on the <strong>Review</strong> tab to create markers.</td>
</tr>
<tr>
<td>Creating Test Reports</td>
<td>You generate reports by using the report template language and Microsoft Word to produce RTF template files, or by using the drag-and-drop TestWorks Report Template Generator program to produce HTML template files. You can use preformatted report templates, or create your own from scratch.</td>
<td>You create test reports with the aide of the MTS Reporter Add-in for Microsoft Excel. You can use preformatted report templates, or create your own from scratch. Report templates use variables which are replaced by the corresponding values from the test run or test (multiple test runs) file when the report is generated.</td>
</tr>
<tr>
<td>Starting a Test Remotely</td>
<td>You start a test remotely by configuring a method to use a digital input to start the test (in the Idle test segment Digital Input task) rather than configuring a method to use a digital input to start the test.</td>
<td>You start a test remotely by configuring both a global setting (<strong>Configuration</strong> &gt; <strong>Preferences</strong> &gt; <strong>Select additional methods to start the test</strong>) and a test definition setting (<strong>Define tab</strong> &gt; <strong>Test Definition</strong> tab &gt; <strong>General Settings</strong> tab &gt;)</td>
</tr>
<tr>
<td>Function</td>
<td>TestWorks 4 Behavior</td>
<td>MTS TestSuite TW Behavior</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Pre-allocating</td>
<td>You use the Tools menu to open the Preallocate Specimens window. When test runs are</td>
<td>You click the <strong>Actions</strong> button on the <strong>Review</strong> tab to open the <strong>Preallocate Test Runs</strong> window. Test runs still appear on the <strong>Review</strong> tab, but no longer use an icon to indicate their status. Instead, the names of test runs that have not been run are italicized. When the test run completes, the text becomes normal (non-italics).</td>
</tr>
<tr>
<td>Test Runs</td>
<td>pre-allocated, they appear on the Review tab. The icon for test runs that have not</td>
<td></td>
</tr>
<tr>
<td></td>
<td>been started was hollow, and it was filled in when the test run completed.</td>
<td></td>
</tr>
<tr>
<td>Outer Loop</td>
<td>If you use advanced rate control in TW4, the outer-loop control mode is not</td>
<td>When using HD control modes in TWE, the outer-loop control mode is maintained when</td>
</tr>
<tr>
<td>Control Modes</td>
<td>maintained when transitioning from a GoTo activity to another activity in a test.</td>
<td>transitioning between GoTo activities in a test. However, similarly to TW4, the crosshead control mode will become the active control mode if an interlock occurs. This behavior prevents the crosshead from moving suddenly after resetting an interlock.</td>
</tr>
</tbody>
</table>

**Converting TestWorks 4 Methods and Sample Files**

**Fastpath:**
Start menu > **All Programs** > **MTS TestSuite** > **TestWorks 4 Converter**

The TestWorks 4 Converter allows you to convert TestWorks 4 methods and sample files, or entire folders, so they can be used with the MTS TestSuite TW application.

There are differences in file formats and pre-set parameters (such as safety parameters) between the two programs, therefore, you must review each converted template or test to confirm the accuracy of the conversion. Although the intent of the converter is to provide a completely converted file, MTS does not guarantee that it can or will permit complete conversion of your TestWorks 4 method or sample file into an MTS TestSuite TW template or test.
Appendix

NI M Series Multifunction DAQ ................................................................. 100
NI M Series Multifunction DAQ

**Note:**
This feature only applies to MTS Criterion and MTS Insight systems.

**Warning:**
The information contained in this document should only be used by qualified personnel.

Misunderstood, misread, or misapplied information used to set up and operate an MTS test system can expose personnel and equipment to severe hazards. This can result in damage to equipment (including test articles) and injury or death to personnel.

Before you use the information in this document, verify your qualifications with your system administrator or MTS.

MTS TestSuite supports the National Instruments (NI) M Series Multifunction Data Acquisition (DAQ) driver. It supports Analog Input (AI), Digital Input (DI), and Digital Output (DO) signals.

Analog Input (AI) signals can be configured to be single-ended or differential, but this configuration applies to all analog input signals. You cannot mix differential and single-ended signals on a test system. The virtual Transducer Electronic Data Sheet (TEDS) defines the scaling, dimension, and units of the AI signal.

The DI and DO signals are grouped into ports of 8 digital signals which are either all inputs or all outputs. DO signals support Toggle, Pulse, Set, and Clear.

AI and DI signals can be used in limit detectors.

For more information about National Instruments Data Acquisition products, see their website at National Instruments Data Acquisition (DAQ).

**Note:**
Before purchasing NI M series hardware, contact MTS Technical Support to confirm your requirements.

**Configuration**
To configure the NI M Series Multifunction DAQ for your test, you must modify the TW Diag configuration file. Contact MTS technical support for this procedure.

Once configured, the devices will appear in your test and can be modified from the TW application Resources tab.
Index

A

activities
  assign variables 69
  calculate variables 69
  custom message window 64
  End Test 61
  error indicators 60
  input parameters 69
  input variables 84
  launch external application 64
  Wait for Operator Action 61
  add a variable 67
  array variables 66
  Assign Variables activity 69

C

calculate variable activity 69-70
  calculated variables 66
  conventions 11
  converting
    TestWorks 4 method to TS template 97
  custom help runtime display 64
  custom message window activity 64

D

data acquisition
  error list 60
  disabled resources 62
  Display 84
  Display Name 64

E
edit a variable 67
  End Test activity 61
  error indicator 60

F

files 97
  form
    problem submittal 10

H

handset exclusive control 16

I

input parameter
  activity 69
input variables
  activity 84
  operator entry activity 84
  Internal Name 64

L

launch external application activity 64

M

manuals 7
  mapping resources 43
  Message 84

O

operator entry activities
  input variables 84
  operator information 64

P

polarity 74
  pre-allocate test runs 65

R

ramp and cycle test 81
resource
  mapping 43
  station 43
resources
  disabled 62
  station 43
  test 42
  runtime display
    custom help 64

S

simple variables 66
simultaneous activities 60
station resources 43
support
  phone 9
  technical 7

T

technical support 7
test activity
  properties 60
  sequence 60
test activity variables 69
test resources 42
TestWorks 4
  converting method to TS template 97

V

variable
  calculations 70
  test activities 69
Variable Editor 69
variable selection 67
variable toggle 67
variables
  add 67
  array 66
  calculated 66
  edit 67
  simple 66
  types 66

W

Wait for Operator Action activity 61