

TestDirector

Tutorial

Version 8.0

TestDirector Tutorial, Version 8.0

This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: U.S. Patent Nos. 5,701,139; 5,657,438; 5,511,185; 5,870,559; 5,958,008; 5,974,572; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332, 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; and 6,564,342.6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912 and 6,694,288. Other patents pending. All rights reserved.

Mercury, Mercury Interactive, the Mercury Interactive logo, LoadRunner, LoadRunner TestCenter, QuickTest Professional, SiteScope, SiteSeer, TestDirector, Topaz and WinRunner are trademarks or registered trademarks of Mercury Interactive Corporation or its subsidiaries, in the United States and/or other countries. The absence of a trademark from this list does not constitute a waiver of Mercury Interactive's intellectual property rights concerning that trademark.

All other company, brand and product names are registered trademarks or trademarks of their respective holders. Mercury Interactive Corporation disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury Interactive Corporation
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

© 2004 Mercury Interactive Corporation, All rights reserved

If you have any comments or suggestions regarding this document, please send them via e-mail to documentation@mercury.com.

Table of Contents

Welcome to the TestDirector Tutorial	v
Before You Begin	v
Using This Tutorial.....	vii
Documentation Updates	viii
Typographical Conventions.....	ix
Lesson 1: Introducing TestDirector	1
The TestDirector Testing Process.....	1
Starting TestDirector.....	5
The TestDirector Window	7
The Mercury Tours Sample Web Site	10
Lesson 2: Specifying Testing Requirements	15
Defining Requirements	16
Viewing Requirements	18
Modifying Requirements.....	20
Converting Requirements	21
Lesson 3: Planning Tests	27
Developing a Test Plan Tree	28
Designing Test Steps.....	31
Copying Test Steps	36
Calling Tests with Parameters	37
Creating and Viewing Requirements Coverage	41
Generating Automated Test Scripts.....	48
Lesson 4: Running Tests	51
Defining Test Sets	52
Adding Tests to a Test Set.....	57
Scheduling Test Runs	59
Running Tests Manually	67
Running Tests Automatically	74

Lesson 5: Adding and Tracking Defects	81
How to Track Defects	82
Adding New Defects	83
Matching Defects.....	85
Updating Defects.....	87
Mailing Defects.....	89
Associating Defects with Tests	91
Creating Favorite Views	94
Lesson 6: Tracing Changes	99
Triggering a Traceability Alert.....	100
Viewing the Traceability Alert.....	102
Creating Follow Up Alerts	104
Lesson 7: Analyzing the Testing Process	107
Generating Reports.....	107
Generating Graphs	114
Lesson 8: Where Do You Go from Here?	119
Getting Started	119
Getting Additional Information	121

Welcome to the TestDirector Tutorial

Welcome to the TestDirector Tutorial, a self-paced guide which teaches you how to use TestDirector, Mercury Interactive's Web-based global test management tool.

This tutorial instructs you on how to use TestDirector to manage the application testing process. It familiarizes you with specifying testing requirements, planning tests, executing tests, and tracking defects. It also shows you how to monitor the testing process by creating reports and graphs.

Before You Begin

To do this tutorial, you must have access to the following components from your machine:

- ✓ TestDirector 8.0—Mercury Interactive's Web-based test management tool. You must have access to the following modules: Requirements, Test Plan, Test Lab, and Defects.
- ✓ TestDirector_Demo—a demonstration project that helps introduce you to TestDirector and includes sample requirements, tests, test sets, test runs, and defects.

Note: It is highly recommended that you work on a new copy of the **TestDirector_Demo** project. For more information, contact your TestDirector administrator.

- ✓ Mercury Tours—a sample application which simulates a Web-based site for reserving flights, hotel rooms, car rentals, cruises, and vacation deals.

Note: To access the **TestDirector_Demo** project and Mercury Tours, you must install these components while installing TestDirector. For more information, refer to the *TestDirector Installation Guide*.

- ✓ QuickTest Professional, Astra QuickTest, or WinRunner—testing tools to run the automated tests used in this tutorial. Make sure that you enable your testing tool to work with other Mercury Interactive tools. For more information, refer to the user guide for your testing tool.

Note:

- To integrate TestDirector with QuickTest Professional or Astra QuickTest, you must install the QuickTest Professional/Astra QuickTest Add-in and the TestDirector Connectivity Add-in from the TestDirector Add-ins page.
- To integrate TestDirector with WinRunner, you must install the TestDirector Connectivity Add-in from the TestDirector Add-ins page.

For more information, refer to the *TestDirector Installation Guide*.

Using This Tutorial

This tutorial contains the following lessons:

Lesson 1 Introducing TestDirector

Introduces you to the TestDirector testing process, and familiarizes you with the TestDirector user interface and the sample Mercury Tours Web site.

Lesson 2 Specifying Testing Requirements

Shows you how to define requirements, view and modify the requirements tree, and convert requirements.

Lesson 3 Planning Tests

Shows you how to create a test plan tree, design test steps, copy test steps, call a test with parameters, link tests to requirements, and automate your manual tests.

Lesson 4 Running Tests

Shows you how to define test sets, schedule test runs, and run manual and automated tests.

Lesson 5 Adding and Tracking Defects

Shows you how to add new defects, search for similar defects, update defects, e-mail defects, associate defects with tests, trace changes, and create favorite views.

Lesson 6 Tracing Changes

Shows you how to keep track of changes made to your requirements, tests, and defects as you perform your project testing.

Lesson 7 Analyzing the Testing Process

Shows you how to monitor the testing process by creating reports and graphs.

Lesson 8 Where Do You Go from Here?

Shows you how to start testing your own application and where to find more information about TestDirector.

Documentation Updates

Mercury Interactive is continuously updating its product documentation with new information. You can download the latest version of this document from the Customer Support Web site (<http://support.mercury.com>).

To download updated documentation:

- 1** In the Customer Support Web site, click the **Documentation** link.
- 2** Under **Select Product Name**, select TestDirector.

Note that if TestDirector does not appear in the list, you must add it to your customer profile. Click **My Account** to update your profile.
- 3** Click **Retrieve**. The Documentation page opens and lists the documentation available for the current release and for previous releases. If a document was recently updated, **Updated** appears next to the document name.
- 4** Click a document link to download the documentation.

Typographical Conventions

This book uses the following typographical conventions:

1, 2, 3	Bold numbers indicate steps in a procedure.
►	Bullets indicate options and features.
>	The greater than sign separates menu levels (for example, File > Open).
Stone Sans	The Stone Sans font indicates names of interface elements in a procedure that you perform actions upon (for example, “Click the Run button.”).
Bold	Bold text indicates function names.
<i>Italics</i>	<i>Italic</i> text indicates variable names, or introduces a new term.
Arial	The Arial font is used for examples and statements that are to be typed in literally.
<>	Angle brackets enclose a part of a URL address that needs to be typed in.
...	In a line of syntax, an ellipsis indicates that more items of the same format may be included.

1

Introducing TestDirector

Application testing is a complex process. TestDirector helps you organize and manage all phases of the application testing process, including specifying testing requirements, planning tests, executing tests, and tracking defects.

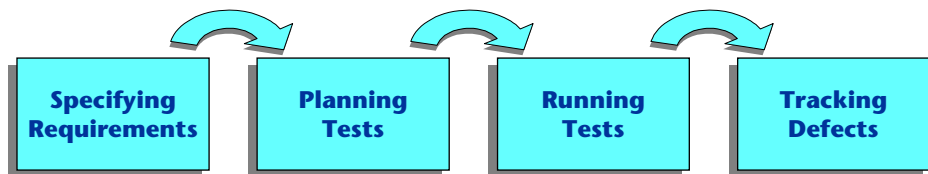
In this lesson, you will learn about:

- The TestDirector Testing Process
- Starting TestDirector
- The TestDirector Window
- The Mercury Tours Sample Web Site

The TestDirector Testing Process

TestDirector offers an organized framework for testing applications before they are deployed. Since test plans evolve with new or modified application requirements, you need a central data repository for organizing and managing the testing process. TestDirector guides you through the requirements specification, test planning, test execution, and defect tracking phases of the testing process.

The TestDirector testing process includes four phases:



Specifying Requirements

You begin the application testing process by specifying testing requirements. In this phase you perform the following tasks:

Task	Description
Define Testing Scope	Examine application documentation to determine your testing scope—test goals, objectives, and strategies.
Create Requirements	Build a <i>requirements tree</i> to define your overall testing requirements.
Detail Requirements	For each requirement topic in the requirements tree, create a list of detailed testing requirements. Describe each requirement, assign it a priority level, and add attachments if necessary.
Analyze Requirements Specification	Generate reports and graphs to assist in analyzing your testing requirements. Review your requirements to ensure they meet your testing scope.

Planning Tests

You create a test plan based on your testing requirements. In this phase you perform the following tasks:

Task	Description
Define Testing Strategy	Examine your application, system environment, and testing resources to determine your testing goals.
Define Test Subjects	Divide your application into subjects or functions to be tested. Build a <i>test plan tree</i> to hierarchically divide your application into testing units, or <i>subjects</i> .
Define Tests	Determine the types of tests you need for each subject. Add a basic definition of each test to the test plan tree.
Create Requirements Coverage	Link each test with a testing requirement(s).

Task	Description
Design Test Steps	Develop manual tests by adding steps to the tests in your test plan tree. <i>Test steps</i> describe the test operations and the expected outcome of each test. Decide which tests to automate.
Automate Tests	For tests that you decide to automate, create test scripts with a Mercury Interactive testing tool, or a custom or third-party testing tool.
Analyze Test Plan	Generate reports and graphs to assist in analyzing test planning data. Review your tests to determine their suitability to your testing goals.

Running Tests

After you build a test plan tree, you run your tests to locate defects and assess quality. In this phase you perform the following tasks:

Task	Description
Create Test Sets	Define groups of tests to meet the various testing goals in your project. These might include, for example, testing a new version or a specific function in an application. Determine which tests to include in each test set.
Schedule Runs	Schedule test execution and assign tasks to testers.
Run Tests	Execute the tests in your test set automatically or manually.
Analyze Test Results	View the results of your test runs to determine whether a defect has been detected in your application. Generate reports and graphs to help analyze these results.

Tracking Defects

Locating and repairing application defects efficiently is essential to the testing process. Defects can be detected and added during all stages of the testing process. In this phase you perform the following tasks:

Task	Description
Add Defects	Report new defects detected in your application. Quality assurance testers, developers, project managers, and end users can add defects during any phase in the testing process.
Review New Defects	Review new defects and determine which ones should be fixed.
Repair Open Defects	Correct the defects that you decided to fix.
Test New Build	Test a new build of your application. Continue this process until defects are repaired.
Analyze Defect Data	Generate reports and graphs to assist in analyzing the progress of defect repairs, and to help determine when to release the application.

Starting TestDirector

You start TestDirector from your Web browser, using the TestDirector URL.

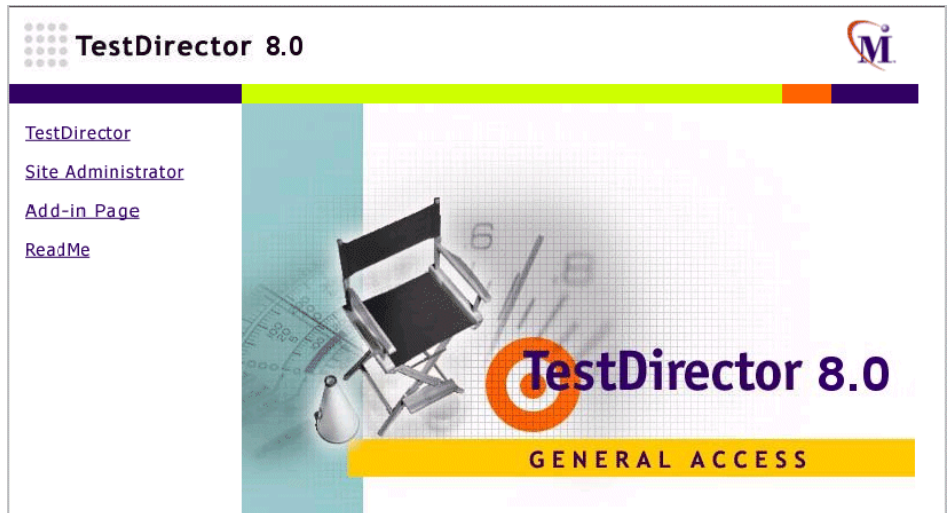
To start TestDirector:

- 1 Open the TestDirector Options window.

In your Web browser, type your TestDirector URL:

<http://<TestDirector server name>/<virtual directory name>/default.htm>

The TestDirector Options window opens.



Note: If you cannot start TestDirector, contact your system administrator to ensure that TestDirector has been installed on your company Web server.

2 Open TestDirector.

Click the **TestDirector** link.

The first time you run TestDirector, the application is downloaded to your computer. Then, each time you open TestDirector, it automatically carries out a version check. If TestDirector detects a newer version, it downloads the latest version to your machine.

The TestDirector Login window opens.



3 Select a domain.

In the **Domain** list, select **DEFAULT**.

4 Select a project.

In the **Project** list, select **TestDirector_Demo**.

Note: If more than one **TestDirector_Demo** project is listed, contact your TestDirector administrator to determine which project to use.

5 Log on to the project as a QA tester.

In the **User ID** box, type one of the following user names: **alice_td**, **cecil_td**, or **michael_td**.

Skip the **Password** box. A password was not assigned to any of the above user names.

Login

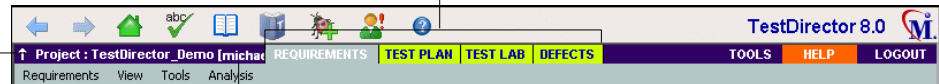
Click the **Login** button.

TestDirector opens and displays the module in which you were last working. In the title bar, TestDirector displays the project name and your user name.

Modules

Project name

User name



The TestDirector Window

In the following exercise, you will explore the TestDirector modules and their common elements.

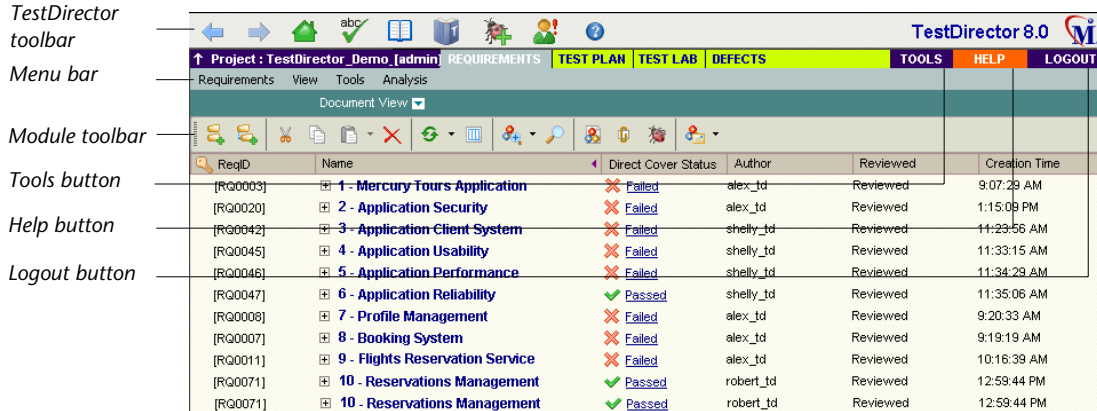
To explore the TestDirector window:

1 Explore the TestDirector modules.

- Click the **Requirements** tab. The Requirements module enables you to specify your testing requirements. This includes defining what you are testing, defining requirement topics and items, and analyzing the requirements.
- Click the **Test Plan** tab. The Test Plan module enables you to develop a test plan based on your testing requirements. This includes defining goals and strategies, dividing your plan into categories, developing tests, automating tests where beneficial, and analyzing the plan.
- Click the **Test Lab** tab. The Test Lab module enables you to run tests on your application and analyze the results.
- Click the **Defects** tab. The Defects module enables you to add defects, determine repair priorities, repair open defects, and analyze the data.

2 Explore the common TestDirector elements.

All the TestDirector modules have common elements. For example, click the **Requirements** tab.








Each of the TestDirector modules contains the following key elements:



- The *TestDirector toolbar* is located directly above the TestDirector project name. If the toolbar is not visible, click the **Show Toolbar** button. The common TestDirector toolbar is accessible from all TestDirector modules and contains the following buttons:

Button	Description
	Back: Navigates back to your previous location in TestDirector.
	Forward: If you navigated back, enables you to navigate forward.
	Home: Logs out and takes you to the TestDirector Login window.
	Check Spelling: Checks the spelling for the selected word or text box. If there are no errors, a confirmation message opens. If errors are found, the Spelling dialog box opens and displays the word together with replacement suggestions.

Button	Description
	Spelling Options: Opens the Spelling Options dialog box, enabling you to configure the way TestDirector checks spelling.
	Thesaurus: Opens the Thesaurus dialog box and displays a synonym, antonym, or related word for the selected word. You can replace the selected word or look up new words.
	Add Defect: Opens the Add Defect dialog box, enabling you to add a new defect.
	Trace All Changes: Opens the Trace All Changes dialog box, enabling you to view traceability alerts.
	Help: Opens the Online Help and displays the help topic for the current context.

- The *menu bar*, located directly below the TestDirector project name, displays the names of menus from which you select commands.
 - The *module toolbar*, located below the menu bar, contains buttons for frequently-used commands in the current TestDirector module.
- TOOLS**
- The **Tools** button, located on the upper-right side of the window, enables you to change your user password and other user properties, open the Document Generator, and view version information for each TestDirector client component.
- HELP**
- The **Help** button, located on the upper-right side of the window, enables you to access TestDirector's online resources.
- LOGOUT**
- The **Logout** button, located on the upper-right side of the window, enables you to exit and return to the TestDirector Login window.

The Mercury Tours Sample Web Site

Mercury Tours is the sample Web application used in this tutorial. It simulates a Web-based application for reserving flights, hotel rooms, car rentals, cruises, and vacation deals. You should now familiarize yourself with this application.

To explore Mercury Tours:

1 Open the Mercury Tours application.

Open a separate instance of your Web browser, and type the following URL:
<http://<TestDirector server name>/mtours/index.html>

The Mercury Tours home page opens.

The screenshot shows the Mercury Tours website interface. At the top, there is a banner for 'one cool summer' with the ARUBA logo. Below the banner is a navigation bar with links for SIGN-ON, REGISTER, SUPPORT, and CONTACT. On the left side, there is a vertical menu with links for Home, Flights, Hotels, Car Rentals, Cruises, Destinations, and Vacations. Below the menu is an 'HTML VERSION' section with a 'Use Java Version' link and a 'SAVINGS! Rent A Car' advertisement with a 'CLICK HERE' button. The main content area features a 'Featured Destination' section for ARUBA, which includes a globe icon, a photo of a beach, and a description of the island. Below this is a 'Specials' table listing flight deals. On the right side, there is a 'Find A Flight' search box, a 'Sign-In' section with fields for User Name and Password, and a 'Sign-In' button. Below the search box are sections for 'Destinations', 'Vacations', 'Register', and 'Links'.

Featured Destination

ARUBA

This island is surrounded by coral reefs, offers guaranteed sunshine and is blessed with beautiful beaches. Luxury resorts have taken up residence along most of the beachfronts on the southern coast, but there are still undeveloped areas on the exposed northern coast, and much of the interior is inhabited by nothing more substantial than goats.

Specials

Atlanta to Las Vegas	\$398
Boston to San Francisco	\$513
Los Angeles to Chicago	\$168
New York to Chicago	\$198
Phoenix to San Francisco	\$213

Find A Flight

Registered users can **sign-in here** to find the lowest fare on participating airlines.

User Name:

Password:

Sign-In

Destinations

Find detailed information about [your destination](#).

Vacations

Read about our [featured vacation destinations](#).

Register

[Register here](#) to join Mercury Tours!

Links

May 20, 2002

2 Subscribe to Mercury Tours.

Click **Register**. The Register page opens.

REGISTER

To create your account, we'll need some basic information about you. This information will be used to send reservation confirmation emails, mail tickets when needed and contact you if your travel arrangements change. Please fill in the form completely.

Note: Please fill all fields bellow to complete the registration.

Contact Information

First Name:

Last Name:

Phone:

Email:

Mailing Information

Address:

City:

State/Province:

Postal Code:

Country:

User Information

User Name:

Password:

Confirm Password:

SUBMIT

Type your **Contact Information**, **Mailing Information**, and **User Information**, in the relevant fields.

Click **Submit**. A confirmation of your registration is displayed in the Register page.

3 Log on to Mercury Tours.

Click the **sign-in** link. The Sign-on page opens.

The screenshot shows the Mercury Tours website's sign-in page. At the top left is the Mercury Tours logo. To its right is a banner for 'one cool summer ARUBA' with a photo of people on a boat. Below the banner is a navigation bar with links for SIGN-ON, REGISTER, SUPPORT, and CONTACT. A yellow 'SIGN-ON' button is highlighted. The main content area contains a welcome message: 'Welcome back to Mercury Tours! Enter your user information to access the member-only areas of this site. If you don't have a log-in, please fill out the [registration form](#).' Below this are two input fields: 'User Name:' and 'Password:'. A yellow 'SUBMIT' button is positioned below the password field. On the left side, there is a vertical menu with links for Home, Flights, Hotels, Car Rentals, Cruises, Destinations, and Vacations. Below the menu is a section for 'HTML VERSION' with a link to 'Use Java Version'. At the bottom left, there is a 'SAVINGS! Rent A Car' advertisement with a 'CLICK HERE' link. A copyright notice '© 2000, Mercury Interactive (v. 020213-1.01-061)' is located at the bottom center.

In the **User Name** and **Password** boxes, type your user name and password. Click **Submit**.

The Flight Finder page opens.

MERCURY TOURS

one cool summer **ARUBA**

SIGN-OFF ITINERARY PROFILE SUPPORT CONTACT

FLIGHT FINDER

Use our Flight Finder to search for the lowest fare on participating airlines. Once you've booked your flight, don't forget to visit the Mercury Tours Hotel Finder to reserve lodging in your destination city.

Flight Details

Type: Round Trip One Way

Passengers: 1

Departing From: Acapulco

On: May 20 View Calendar

Arriving In: Zurich

Returning: May 21 View Calendar

Preferences

Service Class: Economy class
 Business class
 First class

Airline: No Preference

CONTINUE

HTML VERSION
Use Java Version

SAVINGS!
Rent A Car
CLICK HERE

4 Reserve a flight.

Follow the on-screen instructions to reserve a flight.

5 Explore other options.

Explore the following options: **Hotels**, **Car Rentals**, **Cruises**, **Destinations**, and **Vacations**.

6 End your Mercury Tours session.

Click **SIGN-OFF** to log off.



Now that you are familiar with the TestDirector testing process, the TestDirector modules, and the Mercury Tours sample Web site, you can proceed to Lesson 2, “Specifying Testing Requirements.”

2

Specifying Testing Requirements

You begin the testing process by specifying testing requirements in TestDirector's Requirements module. Requirements describe in detail what needs to be tested in your application and provide the test team with the foundation on which the entire testing process is based.

You define the requirements in TestDirector by creating a *requirements tree*. This is a graphical representation of your requirements specification, displaying your requirements hierarchically. You can group and sort requirements in the tree, monitor task allocation and progress of requirements, and generate detailed reports and graphs.

After you create tests in the Test Plan module, you can link requirements to tests (see "Planning Tests" on page 27). Later, after you begin logging defects, you can also associate requirements with defects (see "Adding and Tracking Defects" on page 81). In this way, you can keep track of your testing needs at all stages of the testing process. If a testing requirement changes, you can immediately identify which tests and defects are affected, and who is responsible for them.

In this lesson, you will learn about:

- Defining Requirements
- Viewing Requirements
- Modifying Requirements
- Converting Requirements

Defining Requirements

In the following exercise, you will define requirements for testing the functionality of reserving cruises in Mercury Tours.

To define a requirement:

1 Open the TestDirector_Demo project.

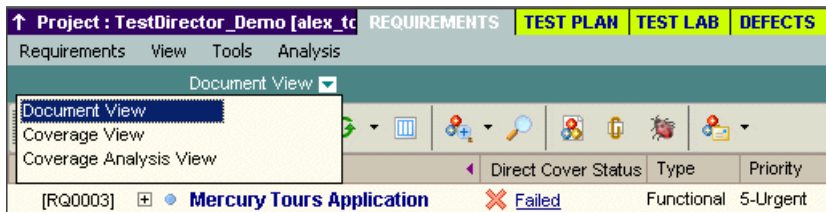
If the **TestDirector_Demo** project is not already open, log on to the project. For more information, see “Starting TestDirector,” on page 5.

2 Display the Requirements module.

Click the **Requirements** tab. The Requirements module displays the requirements tree.

3 Display the requirements tree in Document View.

Make sure the **Document View** of the requirements tree is displayed.



Note: This lesson uses the **Document View**. For more information on the other views, see “Creating and Viewing Requirements Coverage” on page 41.

4 Create a new requirement.



Click the **New Requirement** button. The New Requirement dialog box opens.

Type or select the following:

Name: Cruise Reservation

Product: Mercury Tours (HTML Edition)

Priority: 4-Very High

Type: Functional

Click **OK**. TestDirector adds the **Cruise Reservation** requirement to the requirements tree.

5 Add a child requirement.



Click the **New Child Requirement** button to add the next requirement below **Cruise Reservation**, at a lower hierarchical level. The New Requirement dialog box opens.

Type or select the following:

Name: Cruise Search

Product: Mercury Tours (HTML Edition)

Priority: 4-Very High

Type: Functional

Click **OK**. TestDirector adds the **Cruise Search** requirement as a child of the **Cruise Reservation** requirement.

6 Add an additional child requirement.

In the requirements tree, select **Cruise Reservation**.

Repeat step 5. This time in the **Name** box, type Cruise Booking.

TestDirector adds the **Cruise Booking** requirement as a child of the **Cruise Reservation** requirement.

ReqID	Name	Direct Cover Status	Type	Priority	Planned Version	Reviewed	Reviewer	Author
[RQ0003]	Mercury Tours Application	Failed	Functional	5-Urgent	Version 1.0	Reviewed	robert_td	alex_td
[RQ0140]	Cruise Reservation	Not Covered	Functional	4-Very High		Not Reviewed		michael_td
[RQ0141]	Cruise Search	Not Covered	Functional	4-Very High		Not Reviewed		michael_td
[RQ0142]	Cruise Booking	Not Covered	Functional	4-Very High		Not Reviewed		michael_td
[RQ0020]	Application Security	Failed	Standard	5-Urgent	Version 1.0	Reviewed	robert_td	alex_td
[RQ0042]	Application Client System	Failed	System	3-High	Version 1.0	Reviewed	alex_td	shelly_td

Viewing Requirements

You can change the way TestDirector displays requirements in the requirements tree. In the following exercise, you will learn how to zoom in and out of the tree and display numeration.

To view requirements:

1 Make sure the Requirements module is displayed.

If the Requirements module is not displayed, click the **Requirements** tab. The Requirements module displays the requirements tree.

Make sure the requirements tree is displayed in **Document View**.

2 Zoom in and out of the requirement.



To zoom in, select **Cruise Reservation** in the requirements tree.

Click the **Zoom In** button on the toolbar. The requirements tree displays only the **Cruise Reservation** child requirements.

Requirements View Tools Analysis
Document View

Zoomed in to: Cruise Reservation

ReqID	Name	Direct Cover Status	Type	Priority	Planned Version	Reviewed	Review
[RQ0141]	▫ Cruise Search	? Not Covered	Functional	4-Very High		Not Reviewed	
[RQ0142]	▫ Cruise Booking	? Not Covered	Functional	4-Very High		Not Reviewed	

To reverse the zoom in action and display the entire requirements tree, click the **Zoom In** arrow and choose **Zoom Out To Root**.

3 Display numeration in the requirements tree.

To assign hierarchical numbers to each requirement in the tree, choose **View > Numeration**. As you make changes to the tree, TestDirector automatically renumbers the requirements. Note that the numbers are not related to the unique Req ID assigned to each requirement.

Requirements View Tools Analysis
Document View

ReqID	Name	Direct Cover Status	Type	Priority	Planned Version	Reviewed	Review
[RQ0003]	1 - Mercury Tours Application	✗ Failed	Functional	5-Urgent	Version 1.0	Reviewed	robert_t
[RQ0140]	2 - Cruise Reservation	? Not Covered	Functional	4-Very High		Not Reviewed	
[RQ0142]	2.1 - Cruise Booking	? Not Covered	Functional	4-Very High		Not Reviewed	
[RQ0141]	2.2 - Cruise Search	? Not Covered	Functional	4-Very High		Not Reviewed	
[RQ0020]	3 - Application Security	✗ Failed	Standard	5-Urgent	Version 1.0	Reviewed	robert_t
[RQ0042]	4 - Application Client System	✗ Failed	System	3-High	Version 1.0	Reviewed	alex_td

4 Remove the numeration from the requirements tree.

To remove the hierarchical numbering, choose **View > Numeration**.

Modifying Requirements

You can modify the requirements in the requirements tree. In the following exercise, you will learn how to copy, rename, move, or delete requirements.

To modify requirements:

1 Make sure the Requirements module is displayed.

If the Requirements module is not displayed, click the **Requirements** tab. The Requirements module displays the requirements tree.

2 Copy a requirement.



In the requirements tree, select **Cruise Reservation** and click the **Copy** button.



Click the **Paste** button.

A warning box opens because you are duplicating the requirement name. Click **OK**. The requirement is pasted below the selected requirement, at the same hierarchical level. **_Copy_** is added to the end of the requirement's name.

3 Rename the Cruise Reservation_Copy_ requirement.

Right-click the **Cruise Reservation_Copy_** requirement and choose **Rename**.

Edit the requirement name to **Hotel Reservation** and press **Enter**.

4 Move the Hotel Reservation requirement to a different location in the requirements tree.

Select **Hotel Reservation**.



Click the **Cut** button.

Select **Reservations Management**.



To paste **Hotel Reservation** below the selected requirement, click the **Paste** arrow and choose **Paste as Child**.

Click **Yes** to confirm.

[RQ0071]	☐	◆ Reservations Management	✓ Passed	Functional	3-High
[RQ0072]	☐	▫ View Reservations	✓ Passed	Functional	3-High
[RQ0073]	☐	▫ Cancel Reservations	✓ Passed	Functional	3-High
[RQ0154]	☐	◆ Hotel Reservation	? Not Covered	Functional	4-Very High
[RQ0155]	☐	▫ Cruise Booking	? Not Covered	Functional	4-Very High
[RQ0156]	☐	▫ Cruise Search	? Not Covered	Functional	4-Very High



Tip: You can also drag requirements to a new location in the requirements tree.

5 Delete the Hotel Reservation requirement.

Select **Hotel Reservation**.



Click the **Delete** button.

Click **Yes** to confirm. TestDirector deletes the requirement and its children.

Converting Requirements

After you create the requirements tree, you use the requirements as a basis for defining your *test plan tree* in the Test Plan module. For more information, see Lesson 3, “Planning Tests” on page 27.

You can use the Convert to Tests wizard to assist you when designing your test plan tree. The wizard enables you to convert selected requirements or all requirements in the requirements tree to tests or subjects in the test plan tree.

In the following exercise, you will convert the **Cruise Reservation** requirement to a subject in the test plan tree and the **Cruise Reservation** child requirements to tests.

To convert a requirement:

1 Make sure the Requirements module is displayed.

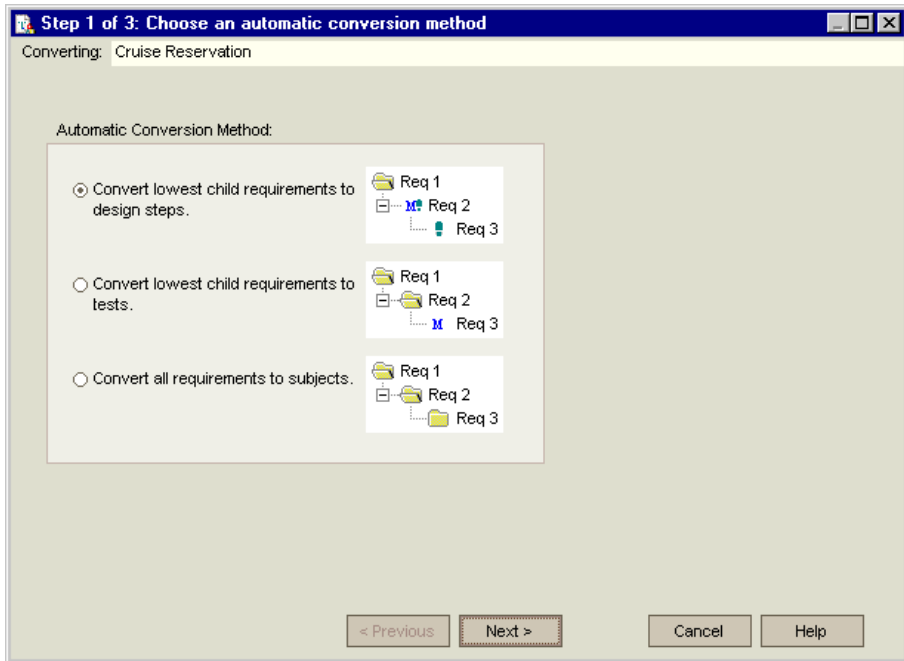
If the Requirements module is not displayed, click the **Requirements** tab. The Requirements module displays the requirements tree.

2 Select a requirement.

In the requirements tree, select **Cruise Reservation**.

3 Open the Convert to Tests wizard.

Choose **Tools > Convert to Test > Convert Selected**. The Step 1 dialog box opens.

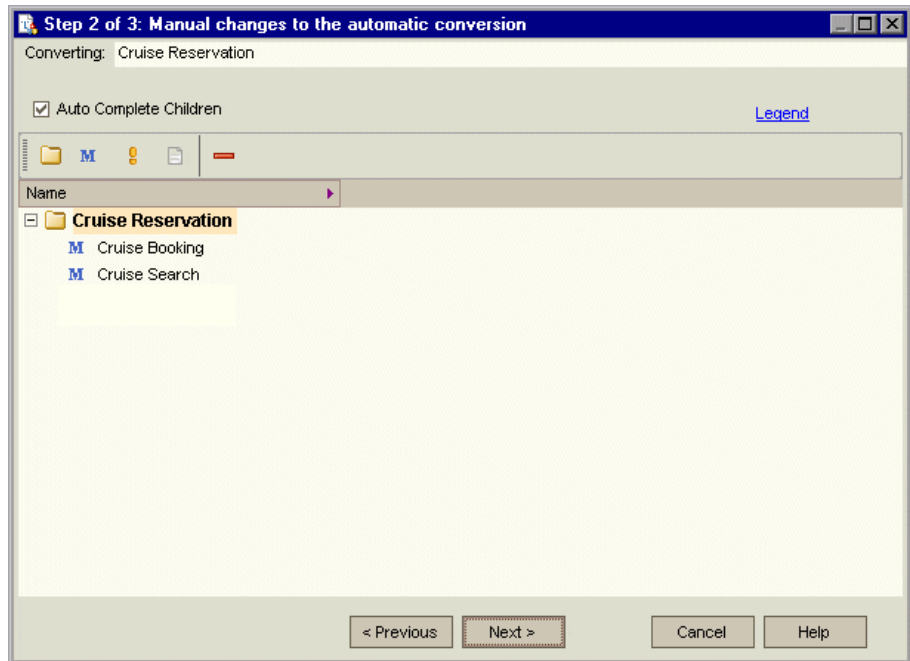


4 Choose an automatic conversion method.

Select the second option, **Convert lowest child requirements to tests**, to convert the selected requirements to tests.

5 Start the conversion process.

Click **Next** to begin converting the requirements. When the conversion process is complete, TestDirector displays the results in the Step 2 dialog box.



6 Convert Cruise Search to a step and restore it.

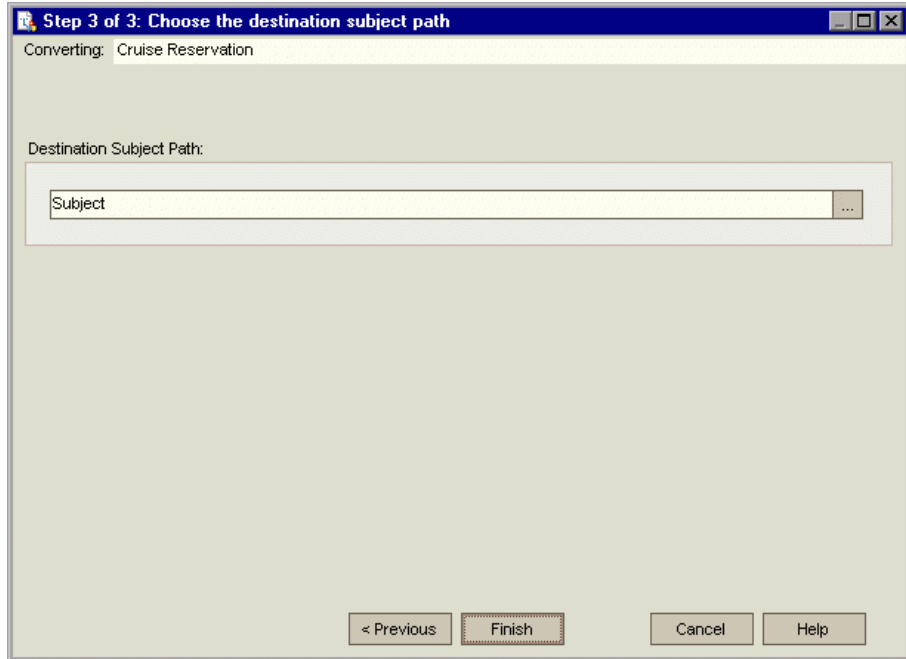


Select **Cruise Search** and click the **Convert to Step** button. TestDirector converts the **Cruise Search** test to a step.



With **Cruise Search** selected, click the **Convert to Test** button. TestDirector converts the **Cruise Search** step back to a test.

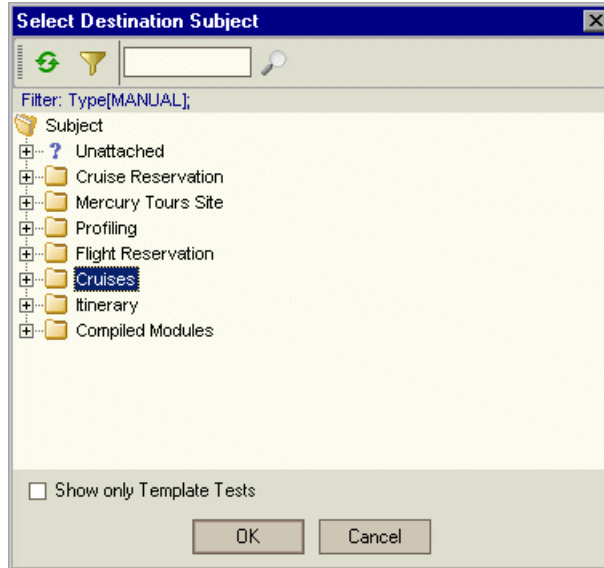
Click **Next**. The Step 3 dialog box opens.



7 Choose the destination subject path.

In the **Destination Subject Path**, click the browse button. The Select Destination Subject dialog box opens.

In the test plan tree, select the **Cruises** subject.



Click **OK** to close the Select Destination Subject dialog box. The **Destination Subject Path** box now indicates the following path:



8 Finalize the conversion process.

Click **Finish**.

Click **OK** to confirm.

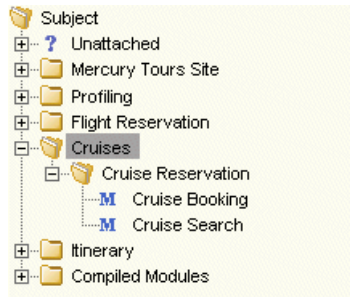
9 View the tests in the test plan tree.

Click the **Test Plan** tab to display the Test Plan module.



In the test plan tree, select **Cruises** and click the **Refresh Selected** button. The test plan tree displays **Cruise Reservation** under **Cruises**.

Expand **Cruise Reservation**. The test plan tree displays the **Cruise Booking** and **Cruise Search** manual tests.



Now that you are familiar with defining requirements, viewing and modifying the requirements tree, and converting requirements, you can proceed to Lesson 3, “Planning Tests”. In Lesson 3, you will learn how to define your test plan tree.

3

Planning Tests

After you define your requirements, you need to determine your testing goals. To do this, examine your application, system environment, and testing process to outline the testing strategy for achieving your goals.

After you determine your testing goals, you build a *test plan tree*, which hierarchically divides your application into testing units, or *subjects*. For each subject in the test plan tree, you define tests that contain steps.

For each test step, you specify the actions to be performed on your application and the expected result. You can increase the flexibility of a test step by adding parameters.

To keep track of the relationship between your tests and your requirements, you can add links between them. By creating links, you ensure compliance with your requirements throughout the testing process.

After you design your tests, you can decide which tests to automate. When you automate a test, you can generate a test script and then complete it using other Mercury Interactive testing tools (for example, QuickTest Professional, Astra QuickTest, or WinRunner).

In this lesson, you will learn about:

- Developing a Test Plan Tree
- Designing Test Steps
- Copying Test Steps
- Calling Tests with Parameters
- Creating and Viewing Requirements Coverage
- Generating Automated Test Scripts

Developing a Test Plan Tree

The typical application is too large to test as a whole. The Test Plan module enables you to divide your application according to functionality. You divide your application into units, or *subjects*, by creating a *test plan tree*. The test plan tree is a graphical representation of your test plan, displaying your tests according to the hierarchical relationship of their functions. After you define subjects in the tree, you decide which tests to create for each subject and add them to the tree.

In Lesson 2, you converted the **Cruise Reservation** requirement and its child requirements to subjects and tests in the test plan tree (see “Converting Requirements,” on page 21). In the following exercise, you will add a subject and a test to the test plan tree in the Test Plan module.

To develop a test plan tree:

1 Open the TestDirector_Demo project.

If the **TestDirector_Demo** project is not already open, log on to the project. For more information, see “Starting TestDirector,” on page 5.

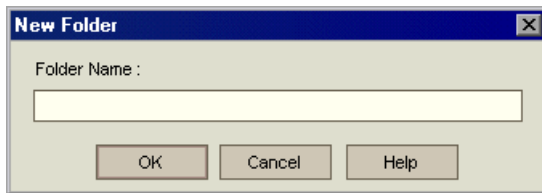
2 Display the Test Plan module.

Click the **Test Plan** tab.

3 Add a subject folder to the test plan tree.



Select the **Cruises** subject folder and click the **New Folder** button. The New Folder dialog box opens.



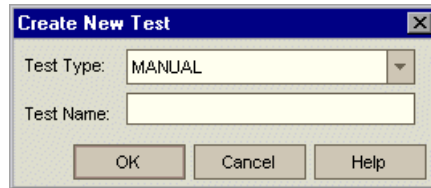
In the **Folder Name** box, type Cruise Cancellation. Click **OK**. The new subject folder appears under the **Cruises** subject folder in the test plan tree.

In the **Description** tab in the right pane, type a description of the subject: This folder contains tests that verify the Cancel Reservation functionality.

4 Add a test to the subject folder.



Select **Cruise Cancellation** and click the **New Test** button. The Create New Test dialog box opens.



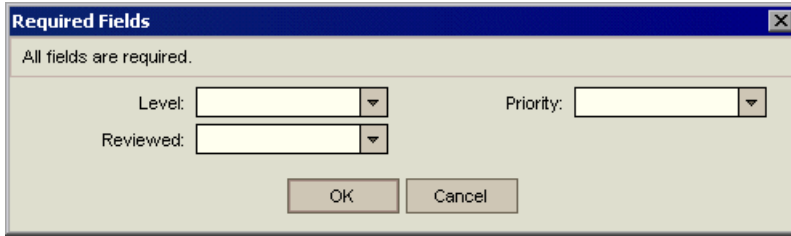
In the **Test Type** box, select **QUICKTEST_TEST** to create a QuickTest Professional/Astra QuickTest test, or select **WR-AUTOMATED** to create a WinRunner test.

Notes:

- The **QUICKTEST_TEST** test type is only available if you have installed the QuickTest Professional/Astra QuickTest Add-in from the TestDirector Add-ins page. For more information on installing the add-in, refer to the *TestDirector Installation Guide*.
 - If you selected **QUICKTEST_TEST** from the **Test Type** list, the **Template** box is available. You can create your new test based on another QuickTest Professional/Astra QuickTest test, defined as a template test. TestDirector copies the template test to your new test without the test results.
-

In the **Test Name** box, type a name for the test: Cancel All Reservations.

Click **OK**. The Required Fields dialog box opens.



Select the following:

Level: Basic

Reviewed: Not Reviewed

Priority: 4-Very High

Click **OK**.

The new test is added to the test plan tree under the **Cruise Cancellation** subject folder.



5 Add a test description.

In the **Details** tab, you can see the test name, test designer, creation date, test status, and other information.

In the **Description** box, type a description for the test: The test verifies cancellation of cruise reservations in the Itinerary page.

Designing Test Steps

After you add a test to the test plan tree and define basic test information, you define test steps—detailed, step-by-step instructions that specify how to execute a test. A step includes the actions to be performed on your application and the expected results.

You can create test steps for both manual and automated tests. For manual tests, you complete test planning by designing the test steps. Using your plan, you can begin test execution immediately. For automated tests, you create an automated test script using a Mercury Interactive testing tool, a custom testing tool, or a third-party testing tool.

In the following exercise, you will create the **Cruise Booking** test. This test verifies the process of booking a cruise through the Mercury Tours site.

To design a test step:

1 Make sure the Test Plan module is displayed.

If the Test Plan module is not displayed, click the **Test Plan** tab.

2 Display the Cruise Booking test.

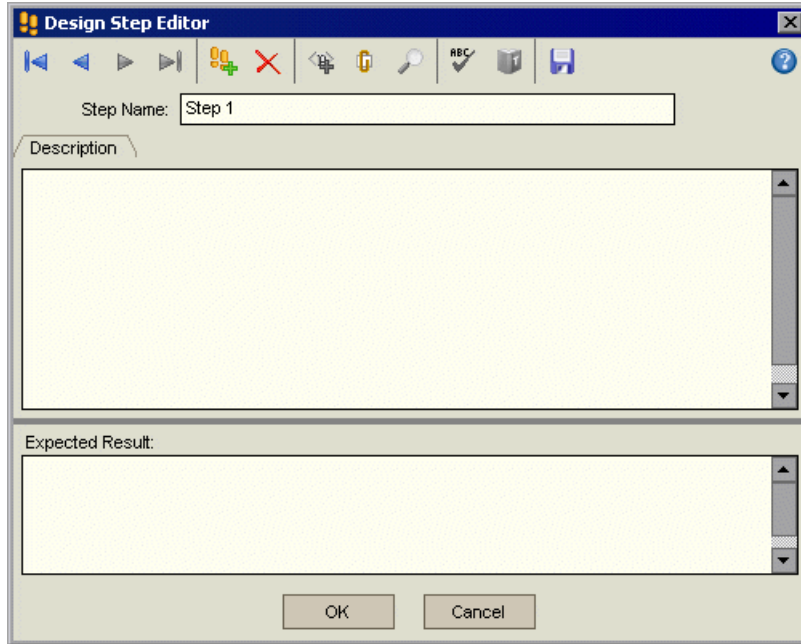
Under the **Cruise Reservation** folder, select the **Cruise Booking** test.

3 Open the Design Step Editor.

Click the **Design Steps** tab.



Click the **New Step** button. The Design Step Editor opens.



In the **Step Name** box, TestDirector displays a step name. The default name is the sequential number of the test step (**Step 1** if you are adding steps to a test for the first time).

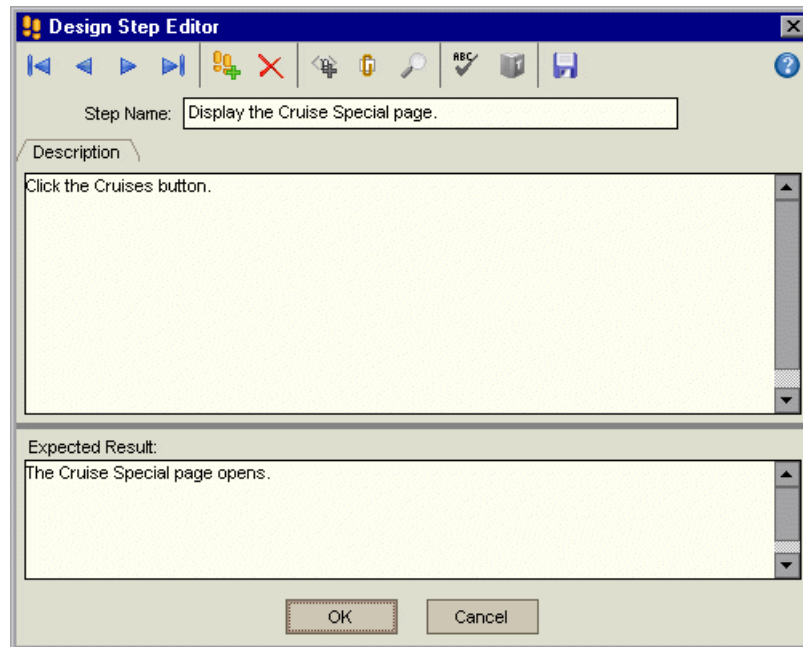
4 Define a step for displaying the Cruise Special page.

In the Design Step Editor, type the following:

Step Name: Display the Cruise Special page.

Description: Click the Cruises button.

Expected Result: The Cruise Special page opens.



5 Define a step for reserving the cruise.



In the Design Step Editor, click the **New Step** button. The **Step Name** box displays **Step 2**.

Type the following:

Step Name: Display the Cruise Reservation page.

Description: Click the Now Accepting Reservations button.

Expected Result: The Cruise Reservation page opens.

6 Define a step for booking the cruise.



In the Design Step Editor, click the **New Step** button. The **Step Name** box displays **Step 3**.

Type the following:

Step Name: Book the cruise.

Description: Enter passenger name, credit card information, and address. Click OK.

Expected Result: The Cruise Confirmation page opens.

7 Define a step for printing the cruise confirmation information.



In the Design Step Editor, click the **New Step** button. The **Step Name** box displays **Step 4**.

Type the following:

Step Name: Print cruise confirmation.

Description: Click the Print button.

Expected Result: A confirmation page is printed.

8 Define a step for logging off the Mercury Tours site.



In the Design Step Editor, click the **New Step** button. The **Step Name** box displays **Step 5**.

Type the following:

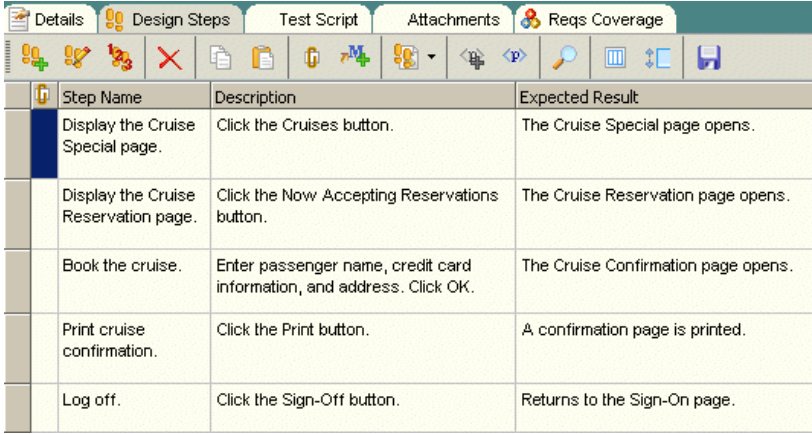
Step Name: Log off.

Description: Click the Sign-Off button.

Expected Result: Returns to the Sign-On page.

9 Close the Design Step Editor.

Click **OK**. The Design Steps tab displays the design steps.



The screenshot shows a software interface with a menu bar at the top containing 'Details', 'Design Steps', 'Test Script', 'Attachments', and 'Reqs Coverage'. Below the menu bar is a toolbar with various icons. The main area displays a table with three columns: 'Step Name', 'Description', and 'Expected Result'. The table contains five rows of design steps.

Step Name	Description	Expected Result
Display the Cruise Special page.	Click the Cruises button.	The Cruise Special page opens.
Display the Cruise Reservation page.	Click the Now Accepting Reservations button.	The Cruise Reservation page opens.
Book the cruise.	Enter passenger name, credit card information, and address. Click OK.	The Cruise Confirmation page opens.
Print cruise confirmation.	Click the Print button.	A confirmation page is printed.
Log off.	Click the Sign-Off button.	Returns to the Sign-On page.

Copying Test Steps

You can copy steps from another test in the same project or from a different project. In the following exercise, you will copy the test steps from the **Cruise Booking** test and paste them into the **Cruise Search** test.


To copy a test step:

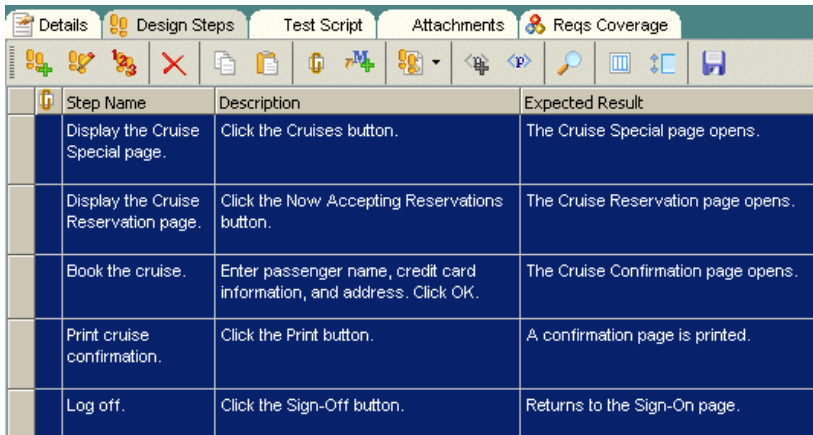
1 Display the Design Steps tab for the Cruise Booking test.

In the test plan tree, under **Cruise Reservation**, select the **Cruise Booking** test.

Click the **Design Steps** tab.

2 Select the steps that you want to copy.

Position the mouse pointer in the gray sidebar on the left. The mouse pointer changes to . Press the **Shift** or **Ctrl** key and select all the steps.



	Step Name	Description	Expected Result
	Display the Cruise Special page.	Click the Cruises button.	The Cruise Special page opens.
	Display the Cruise Reservation page.	Click the Now Accepting Reservations button.	The Cruise Reservation page opens.
	Book the cruise.	Enter passenger name, credit card information, and address. Click OK.	The Cruise Confirmation page opens.
	Print cruise confirmation.	Click the Print button.	A confirmation page is printed.
	Log off.	Click the Sign-Off button.	Returns to the Sign-On page.

3 Copy the selected steps.



Click the **Copy Steps** button.

4 Paste the steps in the Cruise Search test.

In the test plan tree, under **Cruise Reservation**, select the **Cruise Search** test.



In the **Design Steps** tab, click the **Paste Steps** button. TestDirector copies the test steps to the Design Steps tab.

Calling Tests with Parameters

When you design test steps, you can include a call to a manual test. When you run the test, the test steps include the steps from the called test as part of the test. The test that you call is a *template* test. This is a reusable test that can be called by other tests. A template test can include *parameters*. A parameter is a variable that replaces a fixed value. You can modify the value of a parameter according to the test that is calling it, or for various instances of the same test.

For example, suppose you have a test which logs in a user with a specific password when you start your application. You need to call this test at the beginning of each test. In some cases, you may want to log on as a regular user while in other cases, you may want to log on as the administrator. To accomplish this, you create two parameters, <<<user name>>> and <<<password>>>, and modify the value of each parameter according to the type of test that is calling your template test.

In “Designing Test Steps” on page 31, you created a test for booking a cruise. In the following exercise, you will enhance your test by calling the **Connect And Sign-On** test. This template test includes parameters for the Mercury Tours URL address and for the user name and password used to log on to the site.

To call a test with parameters:

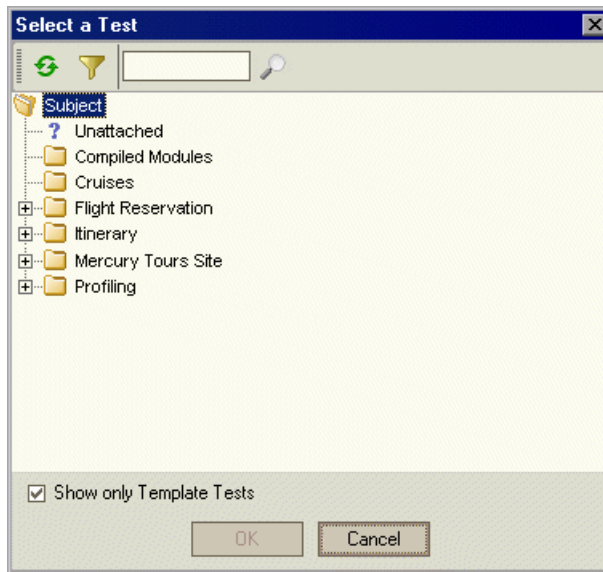
1 Display the Design Steps tab for the Cruise Booking test.

In the test plan tree, under **Cruise Reservation**, select the **Cruise Booking** test.

Click the **Design Steps** tab.

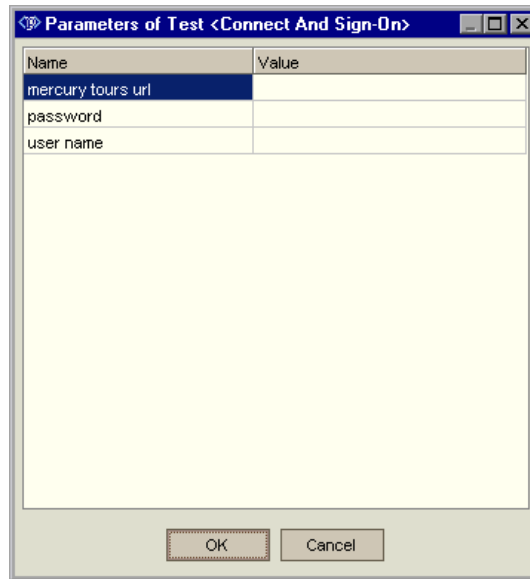
2 Select the test with parameters that you want to call.

Click the **Call to Test** button. The Select a Test dialog box opens.



In the **Find** box, type **Connect**, and click the **Find** button. TestDirector highlights the **Connect And Sign-On** test.

Click **OK**. The Parameters of Test dialog box opens and displays the parameters contained in the called test.



3 Assign values to the parameters.

In the **Value** column, type the following:

mercury tours url: http://<TestDirector server name>/mtours/index.html

password: Leave blank. You will assign a value to this parameter when you run your test (see “Running Tests Manually” on page 67).

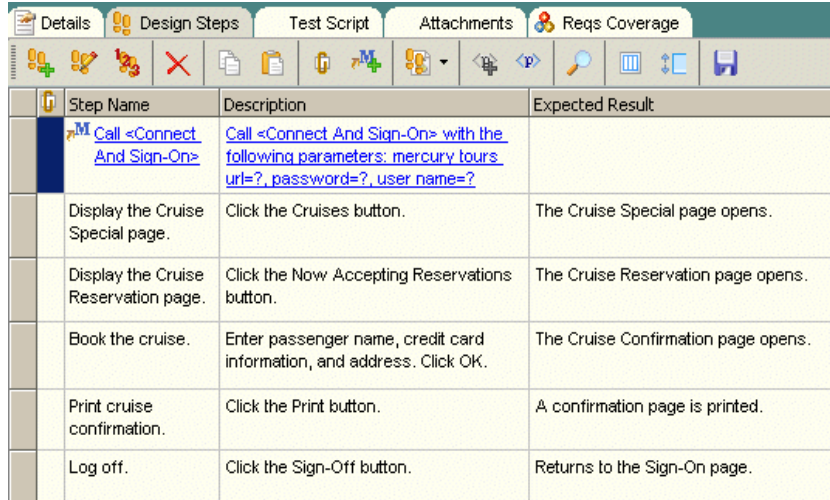
user name: your user name (use the same name as the one used in Mercury Tours on page 10).

Note: You can also assign values to parameters when you create a test that calls your test, when you add your test to a test set, or when you run your test.

Click **OK**. TestDirector adds the **Call Connect And Sign-On** step to your design steps.

4 Reorder the steps.

Position the mouse pointer on the gray sidebar to the left of the **Call Connect And Sign-On** step. The mouse pointer changes to ➡. Click and drag the step to the top row.



Step Name	Description	Expected Result
Call <Connect And Sign-On>	Call <Connect And Sign-On> with the following parameters: mercury.tours.url=?, password=?, user name=?	
Display the Cruise Special page.	Click the Cruises button.	The Cruise Special page opens.
Display the Cruise Reservation page.	Click the Now Accepting Reservations button.	The Cruise Reservation page opens.
Book the cruise.	Enter passenger name, credit card information, and address. Click OK.	The Cruise Confirmation page opens.
Print cruise confirmation.	Click the Print button.	A confirmation page is printed.
Log off.	Click the Sign-Off button.	Returns to the Sign-On page.

5 Adjust the size of the steps.



Click the **Adjust Rows Height** button. TestDirector expands the size of the rows in which the text is too long to view. Note that if you close and reopen the Design Steps tab, TestDirector restores the default step size.

Creating and Viewing Requirements Coverage

It is essential that the tests in your test plan comply with your testing requirements. To help ensure compliance throughout the testing process, you can add links between your tests in the Test Plan module and your requirements in the Requirements module.

In the Test Plan module, you create *requirements coverage* by selecting requirements to link to a test. Alternatively, in the Requirements module, you create *tests coverage* by selecting tests to link to a requirement. A test can cover more than one requirement, and a requirement can be covered by more than one test.

To further ensure compliance with your testing requirements, after you log defects, you can link your tests to defects (see “Associating Defects with Tests” on page 91). This ensures that if a requirement changes, you can identify which tests and defects are affected, and who is responsible for them.

In the following exercises, you will create requirements coverage and tests coverage. You will also view a graphic representation of tests coverage.

Linking Requirements to a Test

In the following exercise you will view existing requirements coverage for the **Cruise Booking** test and create new requirements coverage by linking the **View Reservations** requirement to the **Cruise Booking** test.

To link a requirement to a test:

1 Make sure that the Test Plan module is displayed.

If the Test Plan module is not displayed, click the **Test Plan** tab.

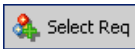
2 Display the Cruise Booking test.

Under the **Cruise Reservation** folder, select the **Cruise Booking** test.

3 Display the Reqs Coverage tab.

In the right pane, click the **Reqs Coverage** tab. TestDirector displays the existing requirements coverage in the coverage grid. Note that the **Cruise Booking** requirement is already linked to the **Cruise Booking** test because you converted the requirement to a test (see “Converting Requirements,” on page 21).

4 Display the requirements tree.



Click the **Select Req** button to show the requirements tree on the right.

coverage grid

requirements tree

Requirement	Reviewed	Description
Cruise Booking	Not Reviewed	

- Mercury Tours Application
- Cruise Reservation
- Application Security
- Application Client System
- Application Usability
- Application Performance
- Application Reliability
- Profile Management
- Booking System
- Flights Reservation Service
- Reservations Management

5 Search for the View Reservations requirement in the requirements tree.



In the **Find** box, type View and click the **Find** button. TestDirector highlights the **View Reservations** requirement in the tree.

6 Add the requirement to the coverage grid.



Click the **Add to Coverage (Include Children)** button. TestDirector adds the **View Reservations** requirement to the coverage grid.



Tip: You can also drag a requirement or requirement topic in the requirements tree to the coverage grid.

7 Hide the requirements tree.



Click the **Close** button.

Linking Tests to a Requirement

In the following exercise, you will create tests coverage by linking the **Cruise Search** test to the **Cruise Booking** requirement.

To link a test to a requirement:

1 Display the Requirements module.

Click the **Requirements** tab.

2 Display the requirements tree in Coverage View.

Select the **Coverage View** of the requirements tree.

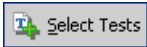


The Test Coverage tab is displayed in the right pane.

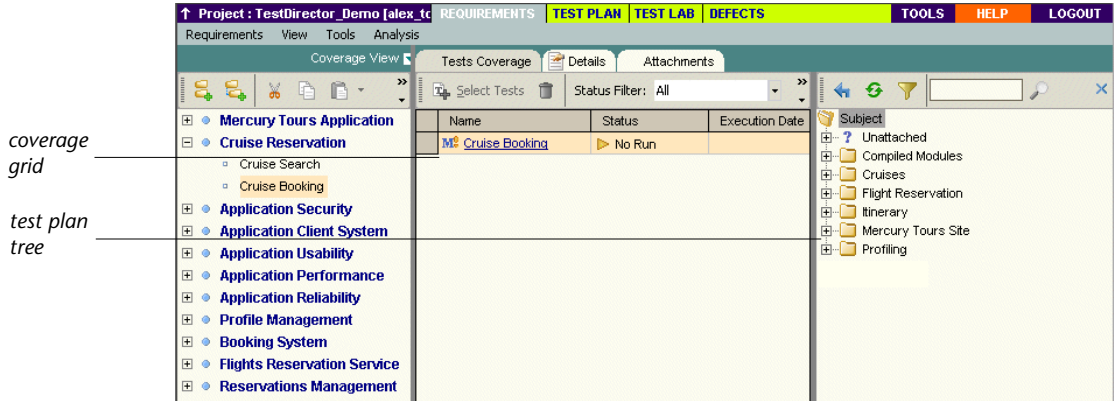
3 Display the Cruise Booking requirement.

In the requirements tree, under **Cruise Reservation**, select the **Cruise Booking** requirement. TestDirector displays the existing tests coverage in the coverage grid. Note that the **Cruise Booking** test is already linked to the **Cruise Booking** requirement because you generated the test from the requirement (see “Converting Requirements,” on page 21).

4 Display the test plan tree.



In the Tests Coverage tab, click the **Select Tests** button to show the test plan tree on the right.



5 Select the Cruise Search test in the test plan tree.

In the **Cruises** folder, expand the **Cruise Reservation** sub-folder.

Select the **Cruise Search** test.

6 Add the test to the coverage grid.



Click the **Add to Coverage** button. TestDirector adds the **Cruise Search** test to the coverage grid.



Tip: You can also drag a test or a subject folder in the test plan tree to the coverage grid.

7 Hide the test plan tree.



Click the **Close** button.

Analyzing Tests Coverage

After you create tests coverage, you can use the *Coverage Analysis View* in the Requirements module to analyze the breakdown of child requirements according to tests coverage.

In this exercise, you will analyze the **Mercury Tours Application** requirement. In the Coverage Analysis View, you will see that five of the requirement's children have a "Failed" status (one or more of the tests covered by the requirement failed) and seven have a "Not Covered" status (the requirement has not been linked to a test). You will then analyze the failed requirements in more detail.

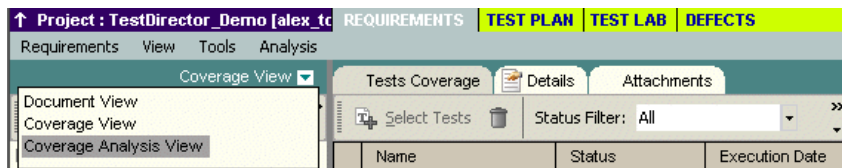
To analyze tests coverage:

1 Make sure that the Requirements module is displayed.

If the Requirements module is not displayed, click the **Requirements** tab.

2 Display the requirements tree in Coverage Analysis View.

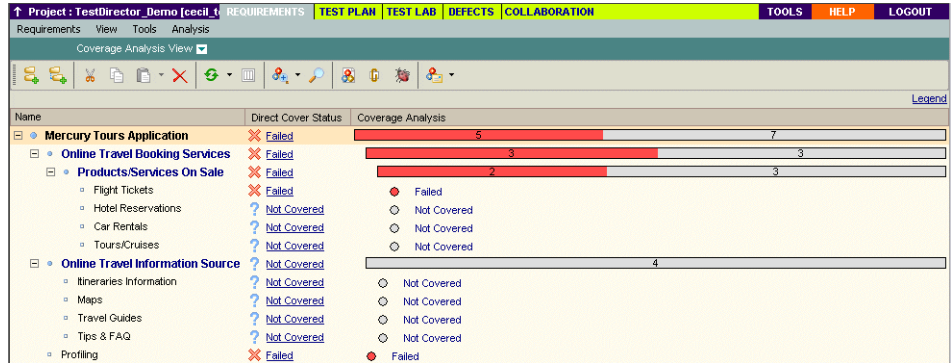
Select the **Coverage Analysis View** of the requirements tree.



TestDirector displays the Coverage Analysis View.

3 Display the Mercury Tours Application requirement in Coverage Analysis View.

Expand the **Mercury Tours Application** requirement and its children.

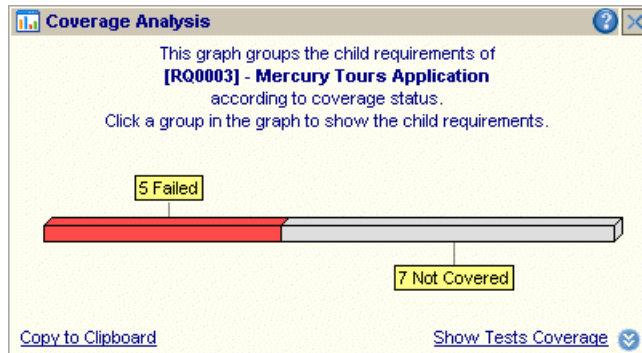


You can see that this requirement has a **Direct Cover Status** of “Failed”. In the Coverage Analysis column, you can see graphically that of the 12 children, only five have “Failed” and seven are not yet covered.

4 Display coverage analysis for the Mercury Tours Application requirement.



Select the **Mercury Tours Application** requirement and click the **Coverage Analysis** button. The Coverage Analysis dialog box opens.



This graph displays the five “Failed” requirements in red and the seven “Not Covered” requirements in gray.

5 Display the child requirements with a “Failed” status.

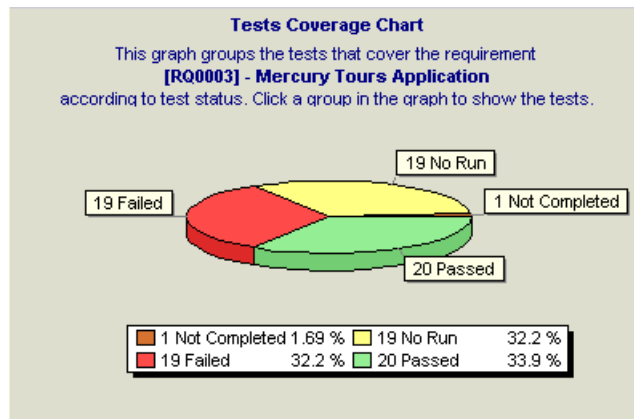
Click the red **Failed** area of the graph. TestDirector lists the five child requirements with a “Failed” status.



Note that you can select a requirement and click **Go To** to highlight the requirement in the requirements tree.

6 Display tests coverage for the requirement.

Click the **Show Tests Coverage** link to extend the Coverage Analysis dialog box and display the Tests Coverage Chart.



You can see, for example, that 19 of the tests associated with the **Mercury Tours Application** requirement have a “Failed” status. The **Mercury Tours**

Application requirement’s direct cover status of “Failed” therefore means that 32.2% of the tests associated with this requirement failed.

Note that you can click a section of the chart to open the Tests Coverage dialog box and display the list of tests with the selected status.

7 Close the Coverage Analysis dialog box.



Click the **Close** button.

Generating Automated Test Scripts

Test planning involves deciding which tests to automate. If you choose to perform tests manually, the tests are ready for execution as soon as you define the test steps. If you choose to automate tests, you can generate test scripts and complete them using other Mercury Interactive testing tools (for example, QuickTest Professional, Astra QuickTest, or WinRunner).

Consider the following issues when deciding whether to automate a test.

Do automate:	<ul style="list-style-type: none"> • Tests that will run with each new version of your application to check the stability of basic functionality across the entire application (regression test). • Tests that use multiple data values for the same operation (data-driven tests). • Tests that are run many times (stress tests) and tests that check a multi-user client/server system (load tests).
Do not automate:	<ul style="list-style-type: none"> • Tests that will be executed only once. • Tests that require immediate execution. • Tests that check how easy the application is to use (usability tests). • Tests that do not have predictable results.

In the following exercise, you will generate an automated test script for the **Cruise Search** test.

To generate an automated test script:

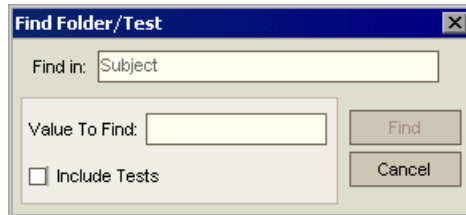
1 Display the Test Plan module.

Click the **Test Plan** tab.

2 Locate the Cruise Search manual test to automate.



Select the **Subject** folder at the root of the test plan tree and click the **Find Folder/Test** button. The Find Folder/Test dialog box opens.



Do the following:

Value to Find: type: Cruise

Include Tests: Select this checkbox to instruct TestDirector to search for folders and tests.

Click **Find**. The Search Results dialog box opens and displays a list of possible matches.

Select **Cruises\Cruise Reservation\Cruise Search** and click the **Go To** button to highlight the test in the test plan tree.

Click **Close** to close the Search Results dialog box.

3 Display the Design Steps tab.

In the right pane, click the **Design Steps** tab.

4 Generate a test script.



Click the **Generate Script** button.

Choose **QUICKTEST_TEST** to generate a QuickTest Professional or Astra QuickTest test, or choose **WR-AUTOMATED** to generate a WinRunner test.

Note: The **QUICKTEST-TEST** test type is only available if you have installed the **QuickTest Professional/Astra QuickTest** add-in from the TestDirector Add-ins page. For more information on installing the add-in, refer to the *TestDirector Installation Guide*.

TestDirector uses the steps in the **Cruise Search** test to create an automated test script. In the test plan tree, note that the manual test icon **M** next to the test is now replaced with the automated test icon.

5 View the test script.

Click the **Test Script** tab.

To display and modify your test script in the testing tool in which it was created, click the **Launch** button.



Now that you are familiar with creating a test plan tree, designing test steps, copying test steps, calling a test with parameters, linking tests to requirements, analyzing tests coverage, and automating your manual tests, you can proceed to Lesson 4, “Running Tests”. In Lesson 4, you will learn how to run your manual and automated tests.

4

Running Tests

Running tests is the core of the testing process. As your application changes, you run manual and automated tests in your project to locate defects and assess quality.

You start by creating *test sets* and choosing which tests to include in each set. A test set is a group of tests in a TestDirector project designed to achieve specific testing goals. TestDirector enables you to control the execution of tests in a test set by setting conditions and scheduling the date and time for executing your tests.

After you define test sets, you can begin to execute your tests. When you run a test manually, you execute the test steps you defined in test planning. You pass or fail each step, depending on whether the actual results match the expected output. When you run a test automatically, TestDirector opens the selected testing tool, runs the test, and exports the test results to TestDirector.

In this lesson, you will learn about:

- ▶ Defining Test Sets
- ▶ Adding Tests to a Test Set
- ▶ Scheduling Test Runs
- ▶ Running Tests Manually
- ▶ Running Tests Automatically

Defining Test Sets

After you design tests in the Test Plan module, you create a *test sets tree*. A test sets tree enables you to organize your testing process by grouping *test sets* in folders and organizing them in different hierarchical levels in the Test Lab module. Test sets can include both manual and automated tests. You can also include the same test in different test sets.

To decide which test sets to create, think about the testing goals you defined at the beginning of the testing process. Consider issues such as the current state of the application and the addition or modification of new features.

Following are examples of general categories of test sets you can create:

Test Set	Description
Sanity	Tests the entire application at a basic level to check that it is functional and stable.
Normal	Tests the system in a more in-depth manner than the sanity test. A Normal test set can contain both positive and negative checks. Positive checks test that the application responds to input as expected. Negative tests attempt to crash an application to demonstrate that the application is not functioning properly.
Advanced	Checks the entire application, including its most advanced features.
Regression	Verifies that a change to one part of the application does not prevent the rest of the application from functioning.
Function	Tests a specific feature or a group of features in the application.

In the following exercise, you will define the **Mercury Tours Site** test set.

To define a test set:

1 Open the TestDirector_Demo project.

If the **TestDirector_Demo** project is not already open, log on to the project. For more information, see “Starting TestDirector,” on page 5.

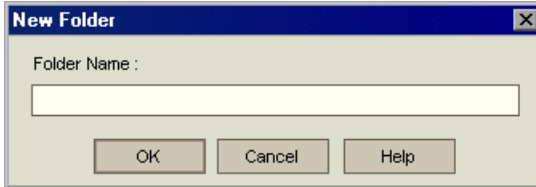
2 Display the Test Lab module.

Click the **Test Lab** tab.

3 Add a folder to the test sets tree.

In the test sets tree in the left pane, select the **Root** folder.

Click the **New Folder** button. The New Folder dialog box opens.

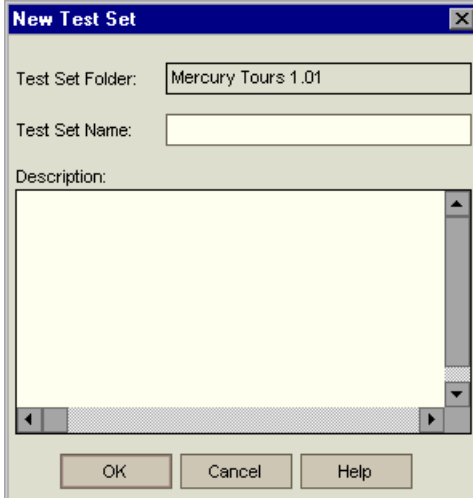



The **New Folder** dialog box has a title bar with a close button. It contains a label **Folder Name :** followed by a text input field. At the bottom, there are three buttons: **OK**, **Cancel**, and **Help**.

In the **Folder Name** box, type **Mercury Tours 1.01** and click **OK**.

4 Add a test set to the Test Sets list.

Click the **New Test Set** button. The New Test Set dialog box opens.

The **New Test Set** dialog box has a title bar with a close button. It contains three input fields: **Test Set Folder:** with the text **Mercury Tours 1.01**, **Test Set Name:** with an empty field, and **Description:** with a large text area. At the bottom, there are three buttons: **OK**, **Cancel**, and **Help**.

Type the following:

Test Set Name: Mercury Tours Site

Description: This test set includes tests that verify the functionality of the Mercury Tours site.

Click **OK**. TestDirector adds the **Mercury Tours Site** test set to the test sets tree in the left pane.

5 Define the test set details.

Click the **Test Set Properties** tab and select the **Details** link.

The screenshot shows the 'Test Set Properties' dialog box with the 'Details' link selected. The 'Test Set Information' section contains a description field with the text 'This test set includes tests that verify the functionality of the Mercury Tours site.' Below the description are fields for 'Close Date', 'Open Date' (set to 6/3/2003), and 'Status' (set to Open). The 'User Defined Fields' section includes fields for 'Level', 'Version', and 'Project'.

By default, the **Status** indicates that the test set is **Open**.

Do the following:

Open Date: Select a date from the calendar. By default, TestDirector displays the current date.

Close Date: Select the planned closing date for the test set.

6 Set rules for the automated tests in the test set in the event of a test failure.

Click the **On Failure** link.

The screenshot shows the 'Test Set Properties' dialog box with the 'On Failure' tab selected. The left sidebar contains links for 'Notifications', 'On Failure', 'Attachments', and 'Details'. The main area is divided into two sections: 'On automated test failure' and 'On final test failure'. In the 'On automated test failure' section, the first checkbox 'If an automated test fails, rerun the test up to 1 times' is checked. The second checkbox 'Run cleanup test <None>' is unchecked. A link 'Settings per test...' is visible. In the 'On final test failure' section, the radio button 'Do nothing' is selected, and the 'Maximal number of times the test set can be rerun is 1' is set.

Execution Grid | Execution Flow | Test Set Properties

[Notifications](#)

[On Failure](#)

[Attachments](#)

[Details](#)

On automated test failure

If an automated test fails, rerun the test up to 1 times

Run cleanup test <None> before each rerun of the test

[Settings per test...](#)

On final test failure

On the final failure of any test in the test set

Do nothing

Stop the test set

Rerun the test set

Maximal number of times the test set can be rerun is 1

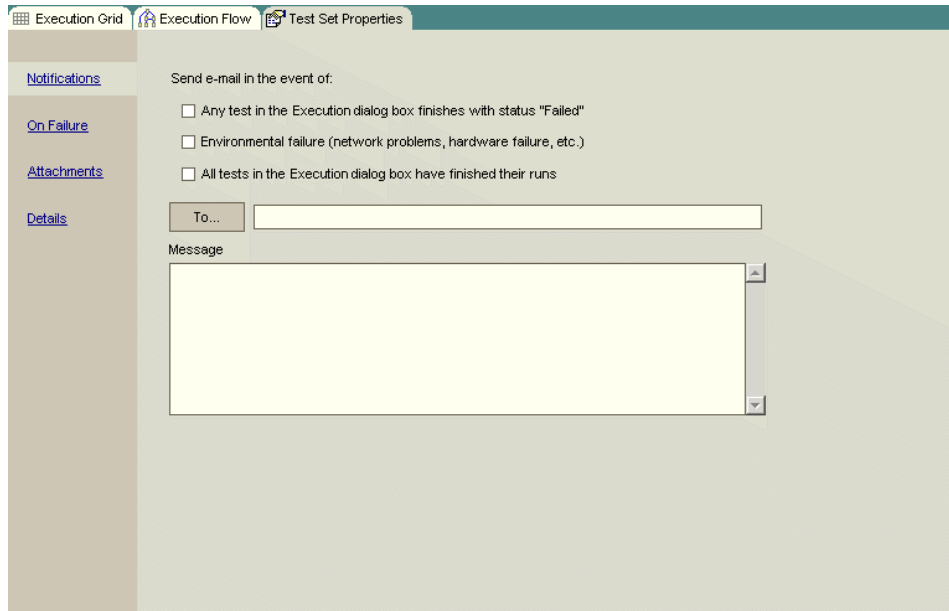
Do the following:

On automated test failure: Select the first check box and make sure that the number of times an automated test can be rerun is set to **1**.

On final test failure: Make sure that the **Do nothing** option is selected.

7 Instruct TestDirector to send an e-mail to specified users if certain events occur.

Click the **Notifications** link.



Do the following:

Send e-mail in the event of: Select the first check box to send an e-mail notification if any test in the test set fails.

To: Type your e-mail address.

Message: Type the following:

This test failed. Please review the test results and submit a defect.

Adding Tests to a Test Set

After you define a test set, you can add copies of tests from the project to your test set. In the following exercise, you will add tests to the **Mercury Tours Site** test set.

To add a test to a test set:

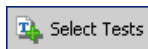
1 Make sure the Test Lab module is displayed.

If the Test Lab module is not displayed, click the **Test Lab** tab.

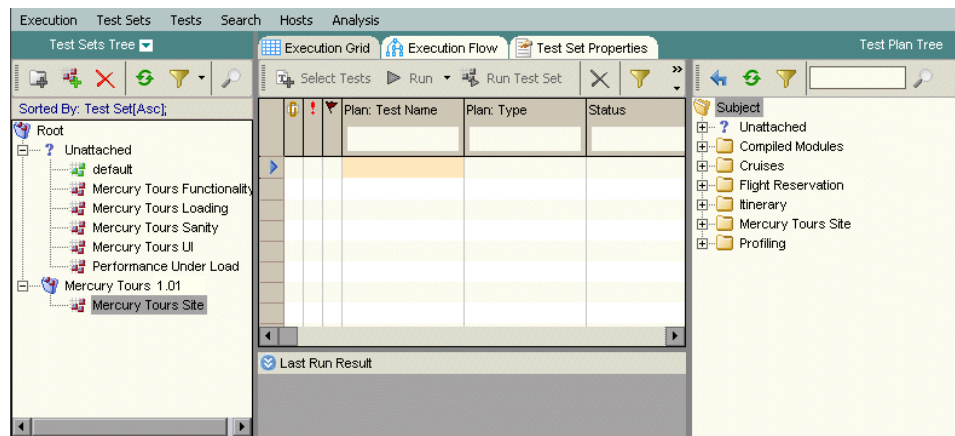
2 Display the Execution Grid tab.

From the test sets tree, select the **Mercury Tours Site** test set and click the **Execution Grid** tab.

3 Display the test plan tree.



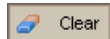
Click the **Select Tests** button. The right pane displays the test plan tree.



Note: If any filters are applied, it is recommended that you clear them:



- In the test plan tree, click the **Set Filter** button. In the Filter dialog box, click the **Clear** button and click **OK** to confirm.



- In the Execution Grid, click the **Clear Filter/Sort** button and click **Yes** to confirm.

4 Add the Cruises folder to the test set.



Select the **Cruises** folder and click the **Add Tests to Test Set** button.

Click **Yes** to confirm. The Parameters of Test dialog box opens because you are adding a test with an unassigned parameter value to a test set. Click **Cancel** to close the dialog box. You will assign this parameter value when you run the Cruise Booking test (see “Running Tests Manually” on page 67).

TestDirector adds the tests to the test set.

5 Add the Airline Preference test to the test set.



To search for the test, in the **Find** box, type airline and click the **Find** button. TestDirector highlights the **Airline Preference** test in the test plan tree.



Click the **Add Tests to Test Set** button. TestDirector adds the test to the test set.

6 Add the Number of Passengers test to the test set.



To search for the test, in the **Find** box, type Number of Passengers and click the **Find** button. TestDirector highlights the **Number of Passengers** test in the test plan tree.



Click the **Add Tests to Test Set** button. TestDirector adds the test to the test set.



Tip: You can also add tests by dragging a folder or test in the test plan tree to the Execution Grid or Execution Flow.

7 Close the test plan tree pane.

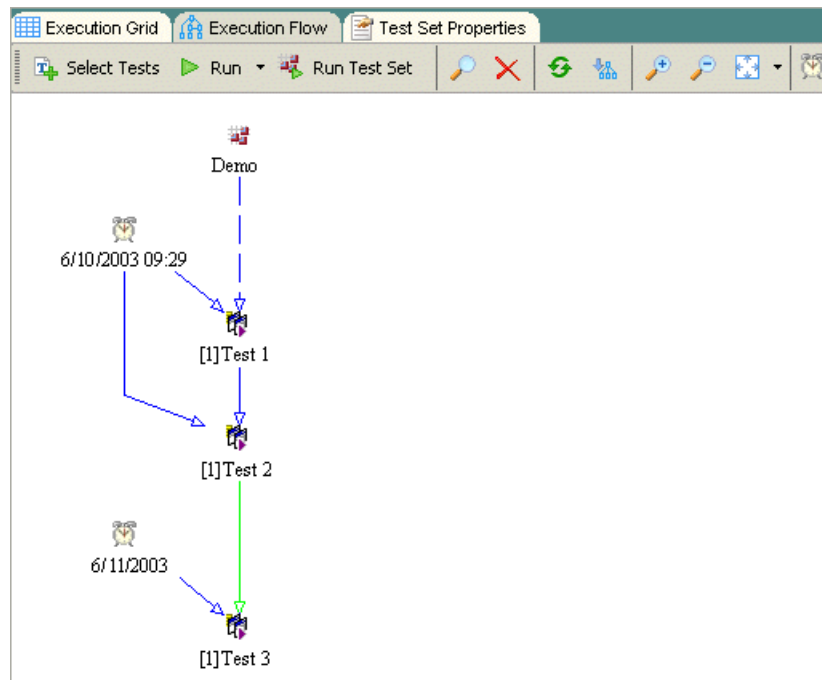


Click the **Close** button.




Scheduling Test Runs

The Execution Flow tab enables you to specify a date and time to execute a test and set conditions for it. A *condition* is based on the results of another specified test in the Execution Flow. By setting conditions, you can instruct TestDirector to postpone the execution of the current test until another specified test finishes running or passes. You can also set the sequence in which to execute the tests.

For example, you can schedule *Test 2* to run only after *Test 1* finishes, and *Test 3* to run only if *Test 2* passes. You can also schedule *Test 1* and *Test 2* to run a day before *Test 3*. The Execution Flow displays the tests and their conditions in a diagram.



Note:

- ▶ A **dashed line**  arrow indicates a test with no conditions.
 - ▶ A **solid line**  arrow indicates a condition and can be blue or green. If the solid line is blue, it indicates that the condition status is set to “Finished”. If the solid line is green, it indicates that the condition is set to “Passed”.
 - ▶ A **Time Dependency**  icon is displayed for time dependent tests.
-

In the following exercise, you will create a new test set and add to it three tests that verify the login procedure on the Sign-On page of the Mercury Tours site. Then, you will set the conditions for each test and specify when each test is to be executed.

To schedule a test run:

1 Make sure the Test Lab module is displayed.

If the Test Lab module is not displayed, click the **Test Lab** tab.

2 Create a new test set.

In the Test Lab module, click the **Execution Flow** tab.



In the test sets tree, choose the **Mercury Tours 1.01** folder and click the **New Test Set** button. The New Test Set dialog box opens.

New Test Set

Test Set Folder: Mercury Tours 1.01

Test Set Name:

Description:

OK Cancel Help

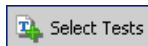
Type the following:

Test Set Name: Test Run Schedule

Description: This test set is used to explain how to schedule a test run.

Click **OK**. TestDirector adds the **Test Run Schedule** test set to the test sets tree in the left pane.

3 Add a test from the Sign-On/Sign-Off folder to the Test Run Schedule test set.



Click the **Select Tests** button. TestDirector displays the test plan tree in the right pane.



In the **Find** box in the test plan tree, type **sign** and click the **Find** button to search for the **Sign-On/Sign-Off** folder. TestDirector highlights the **Sign-On/Sign-Off** folder in the test plan tree.

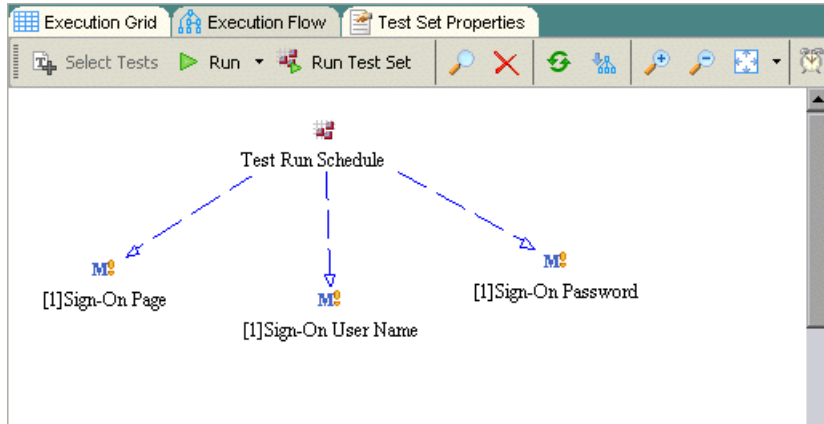


Select the **Sign-On Page** test. Click the **Add Tests to Test Set** button. TestDirector adds the test to the test set.

4 Add two additional tests to the test set.

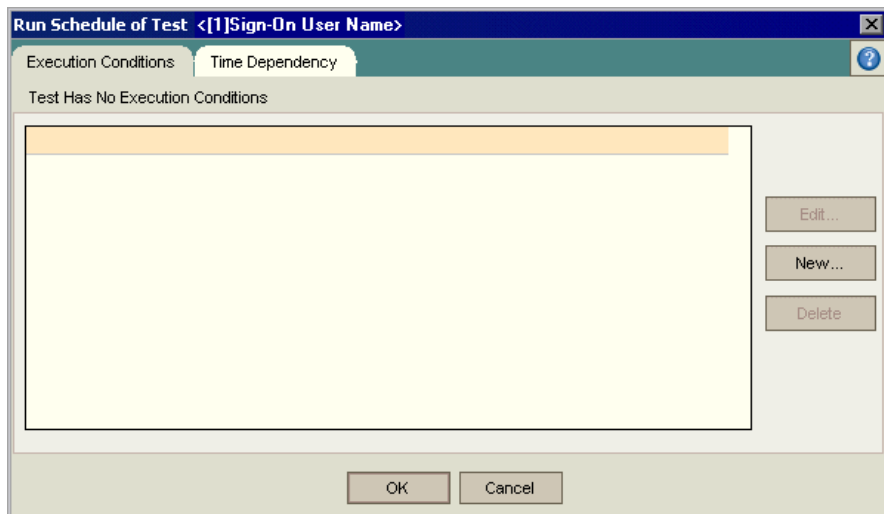
Drag the **Sign-On User Name** test to the Execution Flow area.

Double-click the **Sign-On Password** test to add it to the Execution Flow.

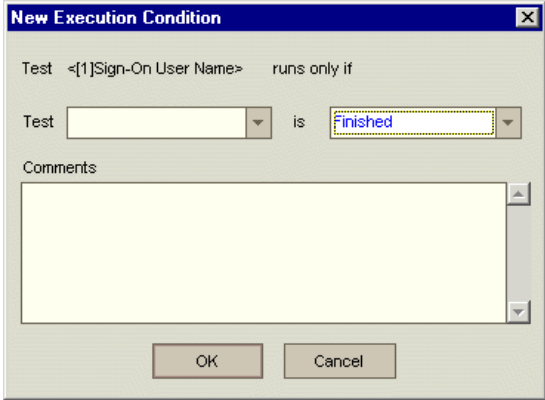


5 Add an execution condition to the Sign-On User Name test.

Right-click the **Sign-On User Name** test and choose **Test Run Schedule**. The Run Schedule of Test dialog box opens and displays the Execution Conditions tab.



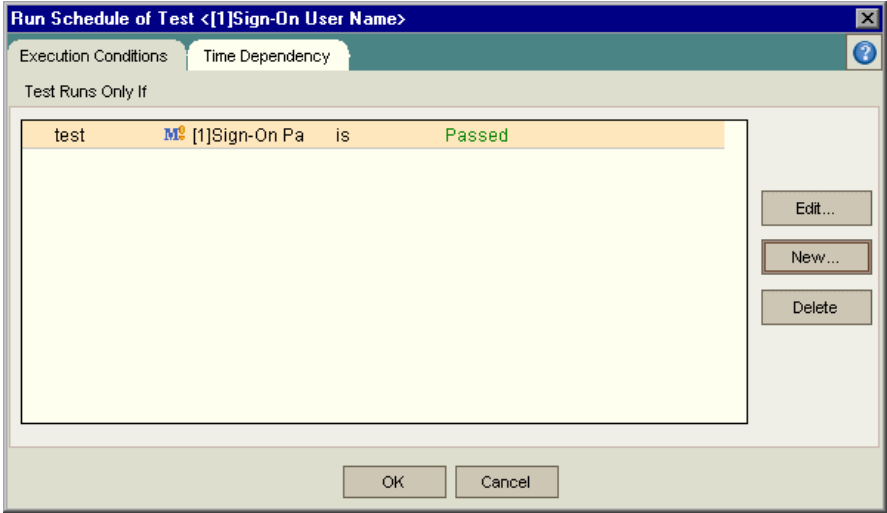
Click **New**. The New Execution Condition dialog box opens.



In the **Test** box, select <[1]Sign-On Page>.

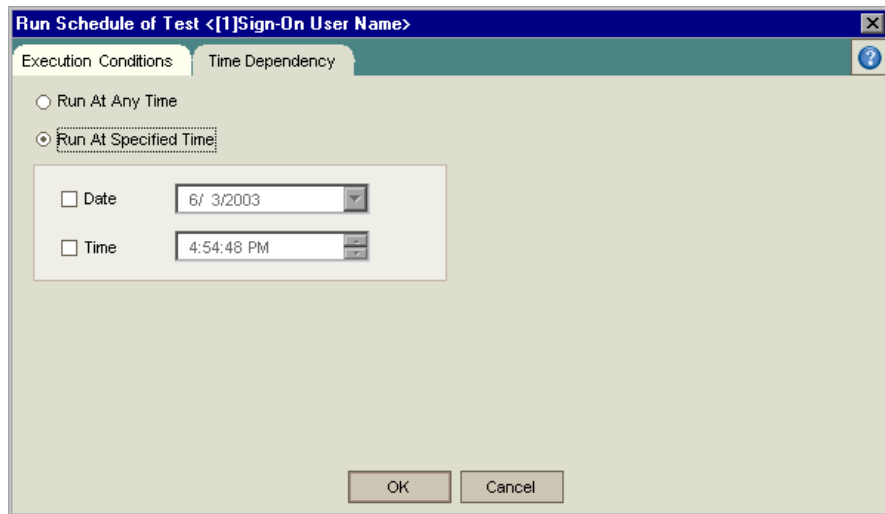
Select **Passed** to instruct TestDirector to execute the **Sign-On User Name** test only if the **Sign-On Page** test finishes executing and passes.

Click **OK**. TestDirector adds the condition to the Run Schedule of Test dialog box.



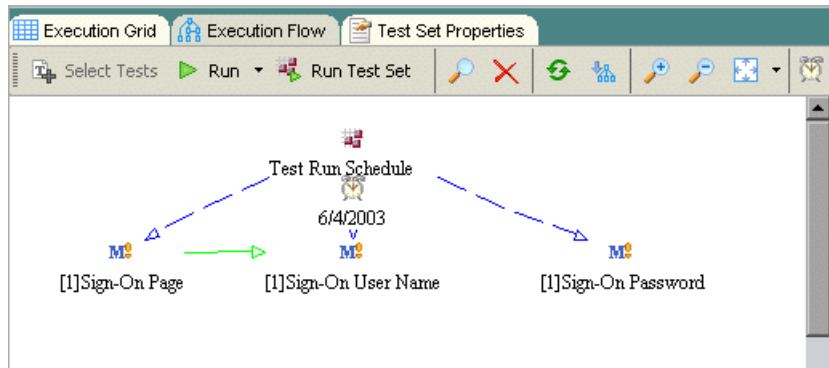
6 Add a time dependency condition to the Sign-On User Name test.

Click the **Time Dependency** tab.



Under **Run At Specified Time**, select the **Date** checkbox and select tomorrow's date.

Click **OK** to close the Run Schedule of Test dialog box. TestDirector displays your conditions in the Execution Flow diagram.



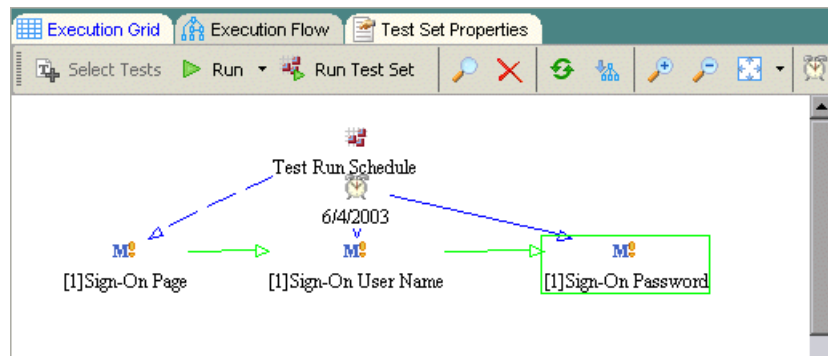
7 Add an execution condition to the Sign-On Password test.

Add the same execution condition as described in Step 5 on page 62 for the **Sign-On Password** test. This time select **Sign-On User Name** from the **Test** box in the New Execution Condition dialog box.

8 Add a time dependency condition to the Sign-On Password test.

Add the same execution condition as described in Step 6 on page 64 for the **Sign-On Password** test.

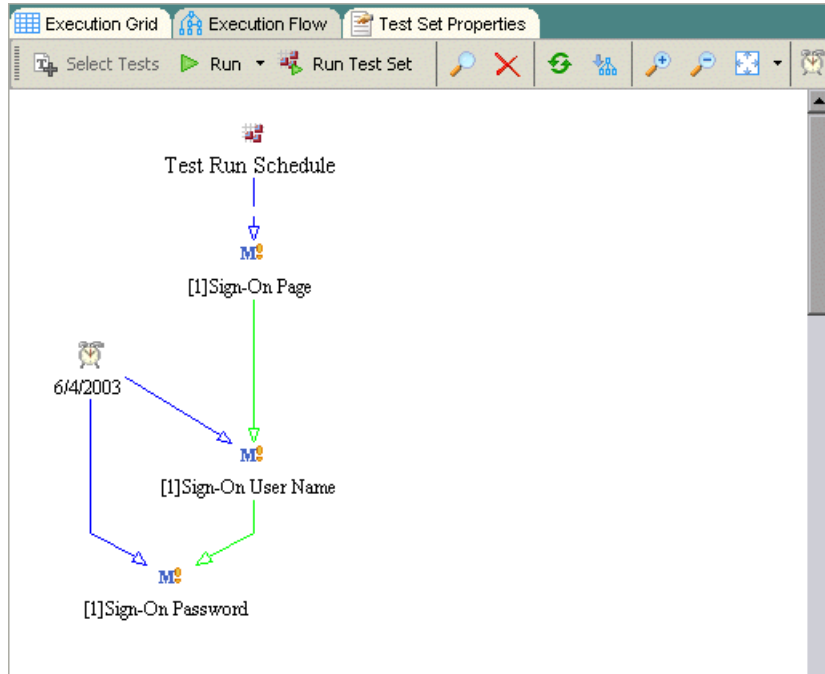
Click **OK** to close the Run Schedule of Test dialog box. TestDirector displays your conditions in the Execution flow diagram.





9 Rearrange the tests in a hierarchical layout.

Click the **Perform Layout** button to clearly view dependencies between the tests.



Running Tests Manually

When you run a test manually, you follow the test steps and perform operations on your application. Then, you compare the expected results with the actual outcome and record the results. You can execute a manual test as many times as needed. TestDirector stores the results separately for each run.

Note that you can run both manual and automated tests manually. You can also choose to run a single test or to run an entire test set.

In the following exercise, you will run the **Cruise Booking** test.

To run a test manually:

1 Make sure the Test Lab module is displayed.

If the Test Lab module is not displayed, click the **Test Lab** tab.

2 Select the Mercury Tours Site test set.

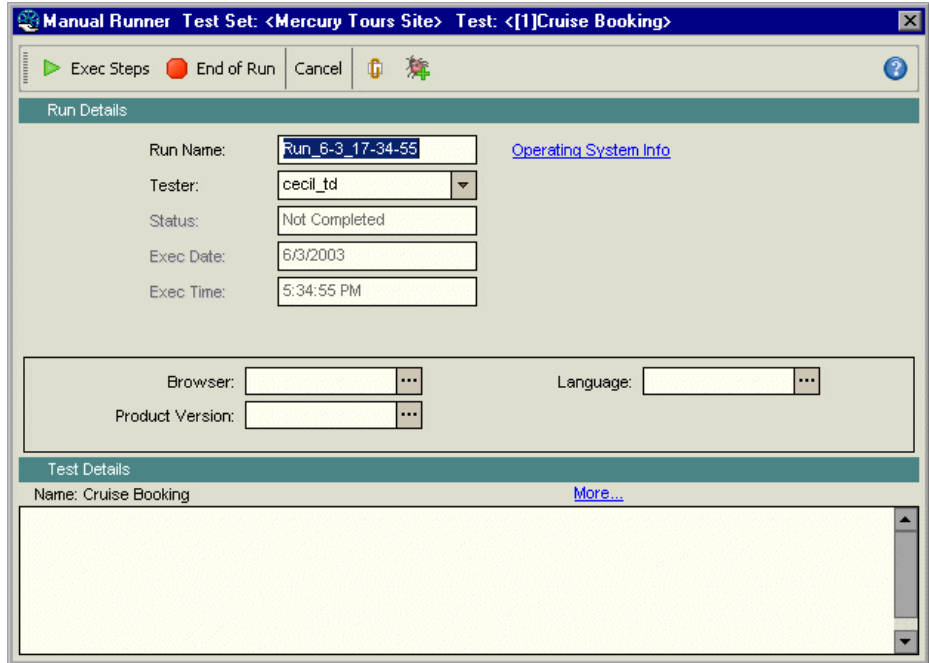
In the test sets tree, select the **Mercury Tours Site** test set.

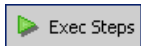
3 Select the Cruise Booking test from the Execution Grid.

In the Execution Grid tab, select the **Cruise Booking** test.

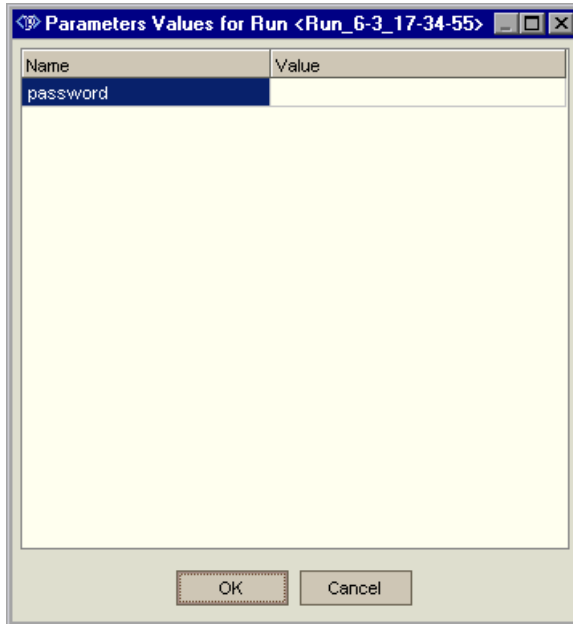


Click the **Run** button. The Manual Runner dialog box opens.



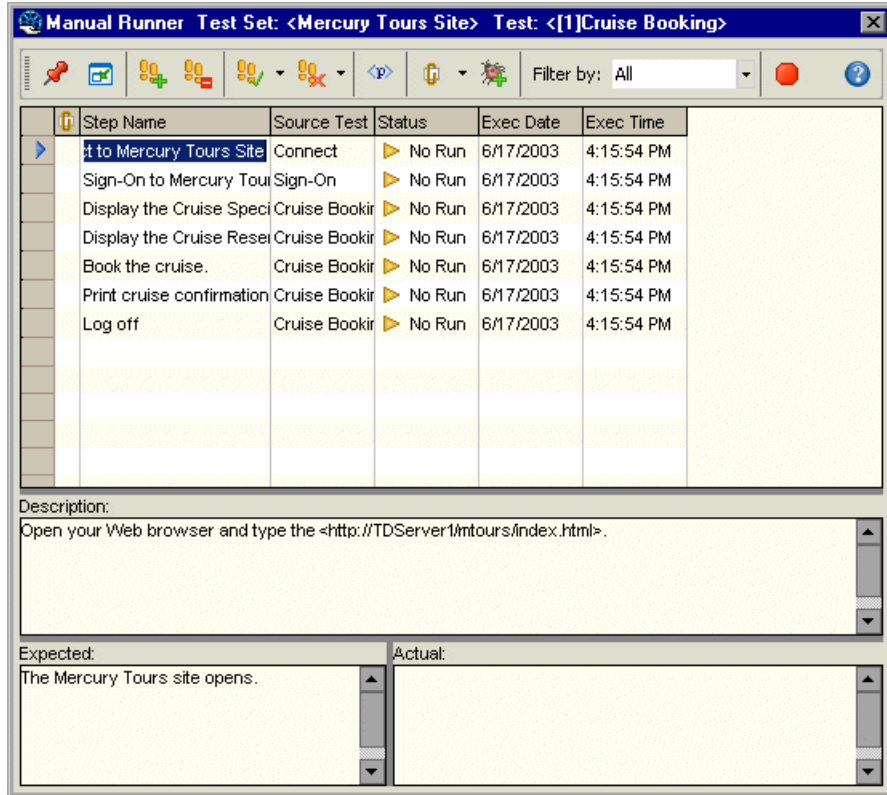
4 Start the test run.

Click the **Exec Steps** button. The Parameters Values for Run dialog box opens because you have an unassigned parameter in the test.

**5 Assign a value for the password parameter.**

Click the **Value** box for **password** and type the same password you used in “The Mercury Tours Sample Web Site” on page 10.

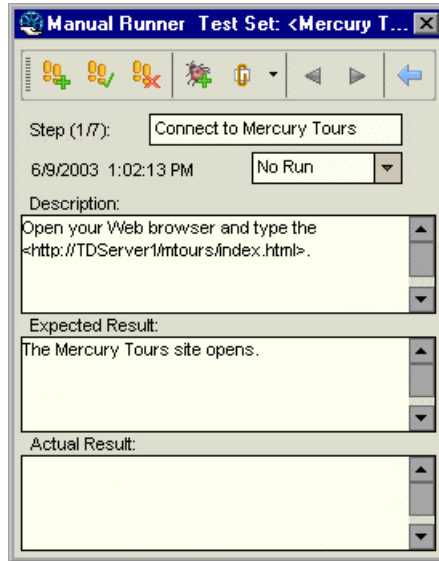
Click **OK**. The Manual Runner: Step Details dialog box opens.



6 Display the Manual Runner dialog box in a compact view.



Click the **Compact View** button. This enables you to conveniently read each step and record the results.



7 Perform the first step.

Perform the procedure described in the **Description** box.

If the actual result is the same as the expected result, in the **Actual** box, type: The Mercury Tours site opens.



Click the **Pass Selected** button. TestDirector displays step 2.

8 Perform the second step.

Perform the procedure described in the **Description** box.

If the actual result is the same as the expected result, in the **Actual** box, type: The Flight Finder page opens.



Click the **Pass Selected** button. TestDirector displays step 3.

9 Perform the third step.

Perform the procedure described in the **Description** box.

If the actual result is the same as the expected result, in the **Actual** box, type: The Cruise Special page opens.



Click the **Pass Selected** button. TestDirector displays step 4.

10 Perform the fourth step.

Perform the procedure described in the **Description** box.

If the actual result is different than the expected result, in the **Actual** box, type: The Flight Finder page opens instead of the Cruise Reservation page.



Click the **Fail Selected** button.

Note: When you detect an application flaw while running the test, you can click the **Add Defect** button to open the Add Defect dialog box and add a defect. For the purpose of this exercise, you will submit this defect in “Adding New Defects,” on page 83.

11 Return to the default display of the Manual Runner.



Click the **Back to Steps Grid** button. The default display of the Manual Runner: Step Details dialog box is displayed.

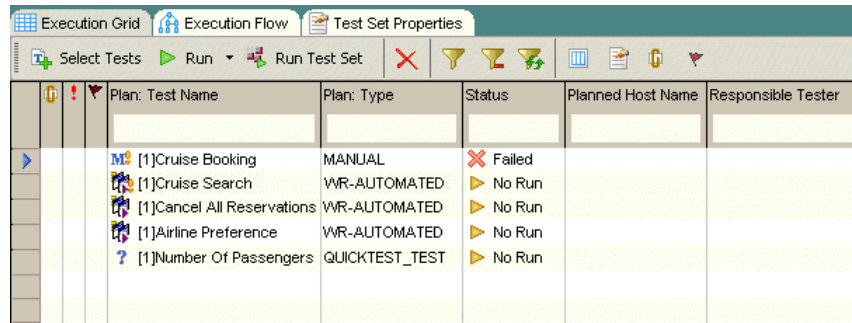
12 End the test run.



Click the **End of Run** button to end your test run.

13 View the test run results in the Execution Grid.

Following the execution of your test, you can view the test run results of your last run in the Execution Grid. Note that TestDirector updates the test run status from “No Run” to “Failed”.

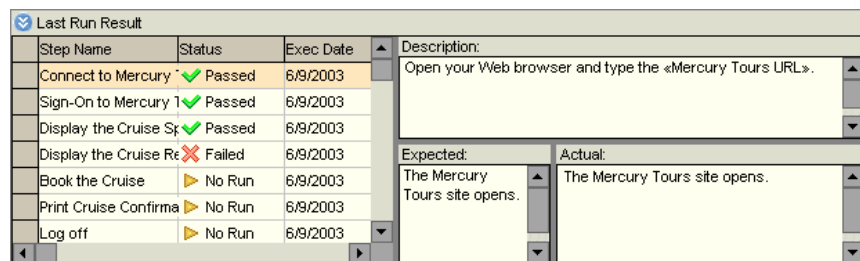


Plan:	Test Name	Plan: Type	Status	Planned Host Name	Responsible Tester
	[1]Cruise Booking	MANUAL	Failed		
	[1]Cruise Search	WR-AUTOMATED	No Run		
	[1]Cancel All Reservations	WR-AUTOMATED	No Run		
	[1]Airline Preference	WR-AUTOMATED	No Run		
	[1]Number Of Passengers	QUICKTEST_TEST	No Run		

14 View the results of each test step in the Last Run Result pane.



If the Last Run Result pane is not displayed, click the **Last Run Result** button on the bottom of the pane. The Last Run Result pane is displayed below the Execution Grid.



Step Name	Status	Exec Date	Description:
Connect to Mercury	Passed	6/9/2003	Open your Web browser and type the «Mercury Tours URL».
Sign-On to Mercury 1	Passed	6/9/2003	
Display the Cruise Sp	Passed	6/9/2003	
Display the Cruise Re	Failed	6/9/2003	
Book the Cruise	No Run	6/9/2003	
Print Cruise Confirma	No Run	6/9/2003	
Log off	No Run	6/9/2003	

Expected:	Actual:
The Mercury Tours site opens.	The Mercury Tours site opens.

Click each step to view the step’s description, as well as the expected and actual results.

Note that you can view more detailed results in the Test Run Properties dialog box (see “Running Tests Automatically” on page 74).

Running Tests Automatically

When you run an automated test, TestDirector opens the selected testing tool automatically, runs the test on your local machine or on remote hosts, and exports the results to TestDirector.

Note that you can run all tests in a test set or run specific tests. You can run tests from the Execution Grid tab or the Execution Flow tab.

In the following exercise, you will run an automated test from the Mercury Tours Site test set. After the test run is complete, you will view the test results from the Test Run Properties dialog box.

Note: To perform the following exercise, you must have one of the following tools installed on your computer:

- ▶ QuickTest Professional
- ▶ Astra QuickTest
- ▶ WinRunner

To launch QuickTest Professional or Astra QuickTest from TestDirector, you must install the QuickTest Professional/Astra QuickTest Add-in and the TestDirector Connectivity Add-in from the TestDirector Add-ins page. For more information, refer to the *TestDirector Installation Guide*.

To run a test automatically:

1 Make sure the Test Lab module is displayed.

If the Test Lab module is not displayed, click the **Test Lab** tab.

2 Display the Mercury Tours Site test set in the Execution Grid.

Click the **Execution Grid** tab.

In the test sets tree, select the **Mercury Tours Site** test set.

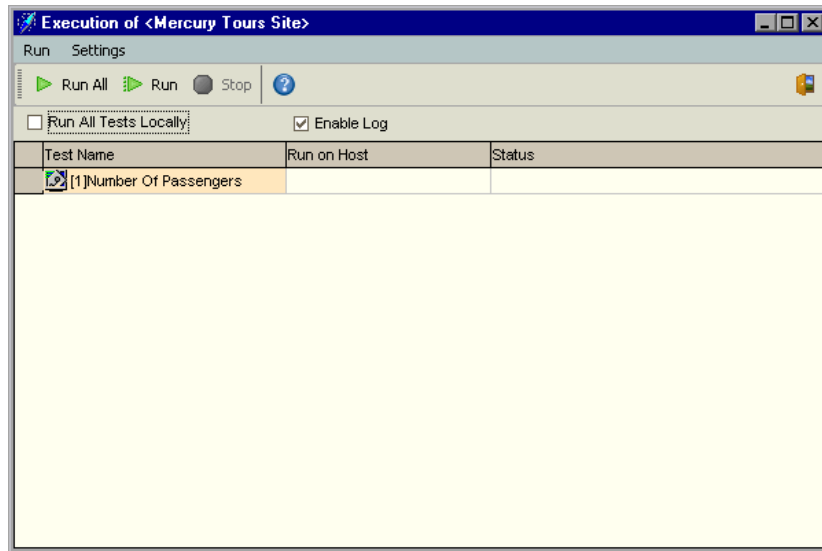
3 Select a test.

To run a WinRunner test, select the **Airline Preference** test.

To run a QuickTest Professional or Astra QuickTest test, select the **Number of Passengers** test.



Click the **Run** button. The Execution dialog box opens and displays the selected test.



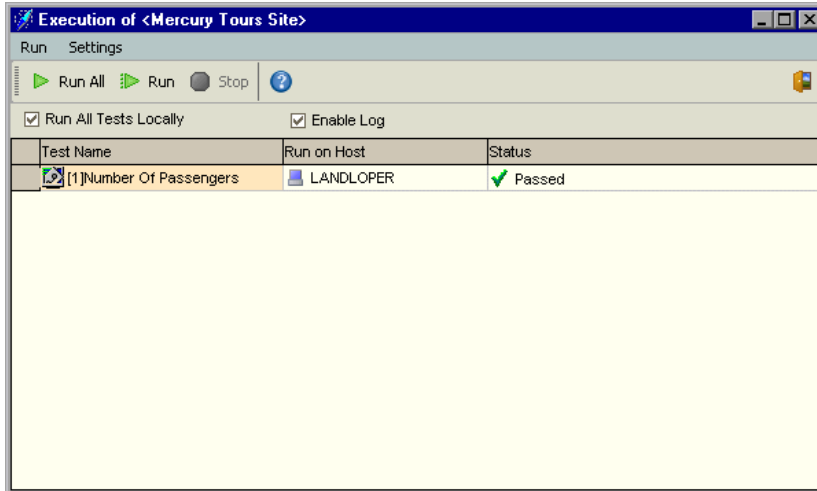
4 Set the test run settings.

Select the **Run All Tests Locally** check box to run the test on your local computer.

5 Run the test.



Click the **Run** button. TestDirector opens the selected testing tool automatically and runs the test. You view the test execution progress in the **Status** column.



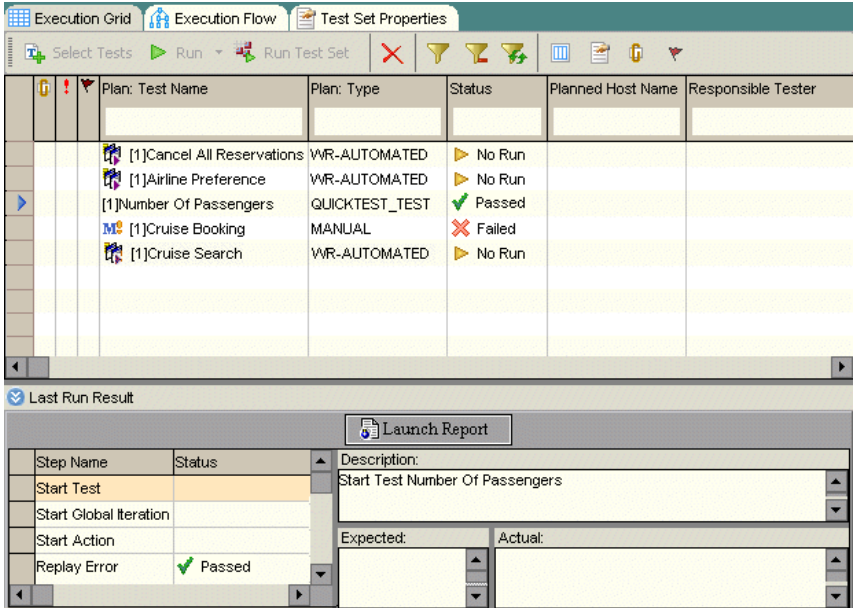
6 Close the Execution dialog box.



After the test run is complete, click **Exit**.

7 View a summary of test results in the Execution Grid.

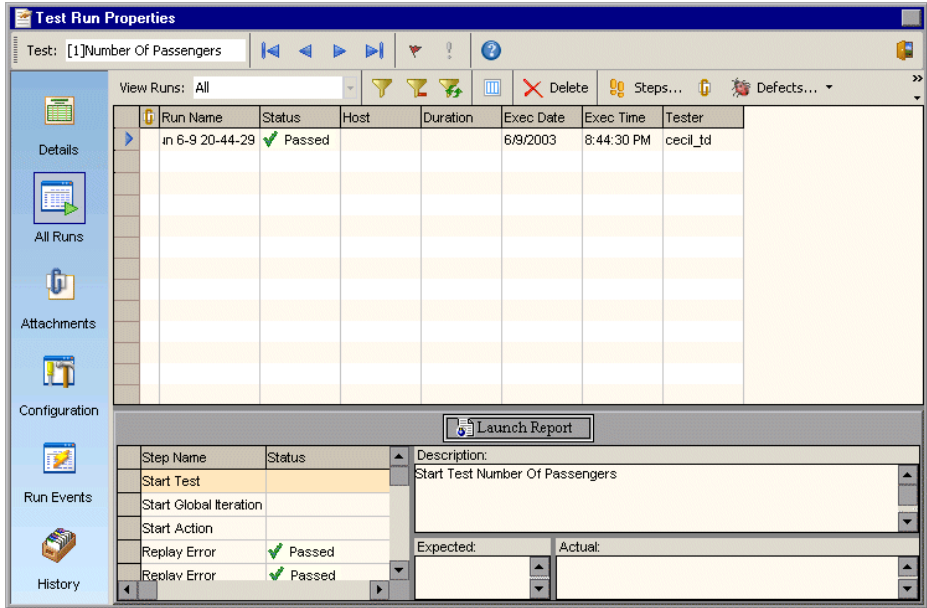
The Execution Grid displays the updated status for the test run. Results for each test step appear in the Last Run Result pane.



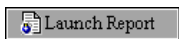
8 View detailed test results from the Test Run Properties dialog box.



In the Execution Grid, make sure your test is selected. Click the **Test Run Properties** button. The Test Run Properties dialog box opens and displays the All Runs view.



9 View the test results in your selected testing tool.



➤ To view the test results in QuickTest Professional or Astra QuickTest, click the **Launch Report** button.



➤ To view the test results in WinRunner, click the **View Report** button.

10 View other test run information in the Test Run Properties dialog box.

- To view run details of the test, in the left menu, click **Details**.
- To view any attachments to a test, in the left menu, click **Attachments**.
- To view the parameters for a manual or WinRunner test, in the left menu, click **Configuration**. Note that any changes that you make will be implemented in the next test run.

- To view the on failure rules for an automated test, in the left menu, click **Run Events**. This view also enables you to change your rules. Note that any changes that you make will be implemented in the next test run.
- To view a list of changes made to the test run fields, in the left menu, click **History**.

**11 Close the Test Run Properties dialog box.**

Click **Exit**.

12 Close your selected testing tool.

In QuickTest Professional, Astra QuickTest, or WinRunner, choose **File > Exit**.



Now that you are familiar with how to define and build test sets, schedule test runs, and run tests manually or automatically, you can proceed to Lesson 5, “Adding and Tracking Defects”. In Lesson 5, you will learn how to add and track defects.

5

Adding and Tracking Defects

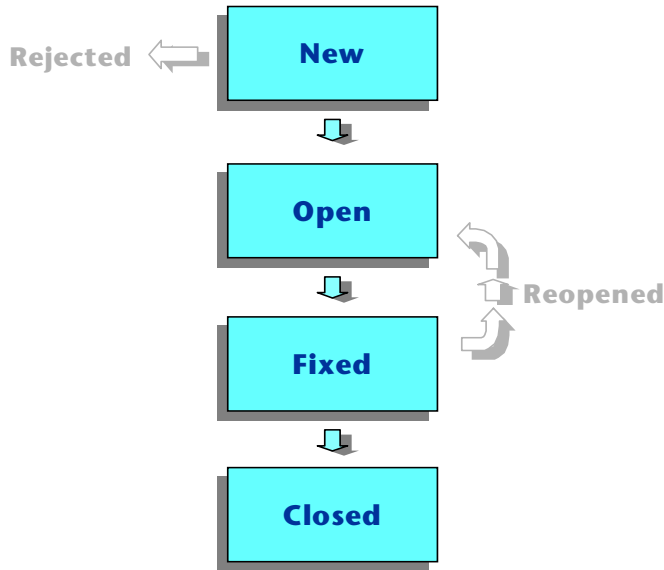
Locating and repairing defects is an essential phase in application development. Defects can be detected and submitted by developers, testers, and end users in all stages of the testing process. Using TestDirector, you can submit defects detected in the application and track them until they are repaired.

In this lesson, you will learn about:

- How to Track Defects
- Adding New Defects
- Matching Defects
- Updating Defects
- Mailing Defects
- Associating Defects with Tests
- Creating Favorite Views

How to Track Defects

When you submit a defect to a TestDirector project, it is tracked through the following stages: *New*, *Open*, *Fixed*, and *Closed*. A defect may also be *Rejected* or *Reopened* after it is fixed.



When you initially report the defect to the TestDirector project, it is assigned the status *New*, by default. A quality assurance or project manager reviews the defect and determines whether or not to consider the defect for repair. If the defect is refused, it is assigned the status *Rejected*. If the defect is accepted, the quality assurance or project manager determines a repair priority, changes its status to *Open*, and assigns it to a member of the development team. A developer repairs the defect and assigns it the status *Fixed*. You retest the application, making sure that the defect does not recur.

If the defect recurs, the quality assurance or project manager assigns it the status *Reopened*. If the defect is actually repaired, the quality assurance or project manager assigns it the status *Closed*.

Adding New Defects

You can add a new defect to a TestDirector project at any stage of the testing process. In the following exercise you will submit the defect that was detected while running the **Cruise Booking** test.

To add a new defect:

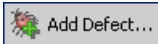
1 Open the TestDirector_Demo project.

If the **TestDirector_Demo** project is not already open, log on to the project. For more information, see “Starting TestDirector,” on page 5.

2 Display the Defects module.

Click the **Defects** tab. The Defects Grid displays defect data in a grid. Each row in the grid displays a separate defect record.

3 Open the Add Defect dialog box.



Click the **Add Defect** button. The Add Defect dialog box opens. Note that fields in red are required.

A screenshot of the "Add Defect" dialog box in TestDirector. The window has a title bar "Add Defect" and a toolbar with "Clear", "Attach:", and various icons. The "Defect Information:" section contains several fields, some marked with a red asterisk to indicate they are required. The fields are: Summary (text box), Category (dropdown), Detected By (dropdown with "pccil_td" selected), Project (dropdown with "Mercury Tours (HT)" selected), Severity (dropdown), Reproducible (text box with "Y"), Priority (dropdown), Subject (dropdown), Detected on Date (dropdown with "6/5/2003" selected), Detected in Version (dropdown), Status (dropdown with "New" selected), Regression (dropdown with "N" selected), Assigned To (dropdown), Planned Closing Version (dropdown), Estimated Fix Time (text box), Modified (dropdown), and Task Status (dropdown). Below these fields is a "Description" section with a large text area. At the bottom are "Submit" and "Close" buttons.

4 Describe the defect.

Type or select the following:

Summary: Unable to reserve a cruise from the Cruise page.

Category: Defect

Severity: 2-Medium

Subject: Cruises

Detected in Version: Version 1.01

Description: The defect was detected in the Cruise Booking test. When you click the Now Accepting Reservations button, the Flight Finder page opens instead of the Cruise Reservation page.

5 Attach the URL address for the Mercury Tours page where the defect was detected.



Click the **Attach URL** button. The Attach URL dialog box opens.

Type the URL address of the Mercury Tours page:

`http://<server name>/mtours/servlet/com.mercurytours.servlet.CruisesServlet`

Click **OK**. TestDirector displays the URL above the **Description** box.

6 Spell check your text.



Place the cursor in the **Description** box and click the **Check Spelling** button. If there are no errors, a confirmation message box opens. If errors are found, the Spelling dialog box opens and displays the word together with replacement suggestions.

7 Add the defect to the TestDirector project.

Click the **Submit** button. Click **OK** to confirm.

8 Close the Add Defect dialog box.

Click **Close**. TestDirector adds the defect to the Defects Grid.

Matching Defects

Matching defects enables you to eliminate duplicate or similar defects in your project. Each time you add a new defect, TestDirector stores lists of keywords from the **Summary** and **Description** fields. When you search for similar defects, keywords in these fields are matched against other defects.

Note that keywords are more than two characters and letter case does not affect your results. TestDirector ignores the following: articles (a, an, the); coordinate conjunctions (and, but, for, nor, or); boolean operators (and, or, not, if, or then); and wildcards (?, *, []).

In the following exercise, you will match defects by comparing a selected defect with all other existing defects in the **TestDirector_Demo** project.

To match defects:

1 Make sure that the Defects module is displayed.

If the Defects module is not displayed, click the **Defects** tab.

2 Select Defect ID 37.

In the Defects Grid, select **Defect ID 37**.



Note: If you cannot find **Defect ID 37**, click the **Clear Filter/Sort** button to clear the filter that is applied to the grid.

3 Find similar defects.

Click the **Find Similar Defects** button.



TestDirector displays the results in the Similar Defects dialog box. TestDirector displays similar defects according to the percentage of detected similarity.

Similar Defects				
Defect ID	Status	Similar	Summary	
37	Open	41%	The list of flights is given when Departing date after Returning	
38	Closed	41%	Cancel All Reservation button is available when there are no reser	
39	Open	41%	There is no confirmation message on Cancel Reservations	
1	Open	42%	The list of flights is given even when past date set as Departing de	
34	Closed	44%	Clicking on Back to Home button opens Find Flight page	
2	Reopen	46%	The list of available flights is not given, when Departing and Return	
14	Closed	49%	All flight are shown when time preference is set to NOON	

Description:

Test Set: Mercury Tours Functionality
 Test: [1]Returning Date
 Run: Run_9-2_10-51-7
 Step: Step 3: Departing Date After Returning Date

R&D Comments:



Click **Close** to close the Similar Defects dialog box.

Updating Defects

Tracking the repair of defects in a project requires that you periodically update defects. You can do so directly in the Defects Grid or in the Defect Details dialog box. Note that the ability to update some defect fields depends on your permission settings.

In this exercise, you will update your defect information by assigning the defect to a member of the development team, changing the severity of the defect, and adding a comment.

To update a defect:

1 Make sure that the Defects module is displayed.

If the Defects module is not displayed, click the **Defects** tab.

2 Update the defect directly in the Defects Grid.

In the Defects Grid, select the defect you added in “Adding New Defects” on page 83.

To assign the defect to a member of the development team, in the **Assigned to** box in the defect record, click the down arrow and select **james_td** from the list.

3 Open the Defect Details dialog box.



Click the **Defect Details** button. The Defect Details dialog box opens.

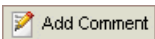
4 Change the severity level of the defect.

In the **Severity** box, select **5-Urgent**.



5 Add a new R&D comment to explain the change in the severity level.

Click **Description** in the left menu.



Click the **Add Comment** button. A new section is added to the **Comments** box, displaying your user name and the current date.

Type: This defect also occurs in Mercury Tours version 1.0.



6 View the Attachments.

Click **Attachments** in the left menu. Note that the URL attachment is listed.

**7 View the History.**

Click **History** in the left menu to view the history of changes made to the defect. For each changed field, TestDirector displays the date of the change, the name of the person who made the change, and the new value.

8 Close the Defect Details dialog box.

Click **OK** to exit the dialog box and save your changes.

Mailing Defects

You can send an e-mail about a defect to another user. This enables you to routinely inform development and quality assurance personnel about defect repair activity. TestDirector includes a **Go To Defect** link in the e-mail which enables the recipient to go directly to the defect.

In the following exercise, you will e-mail your defect to your mailbox.

To mail a defect:

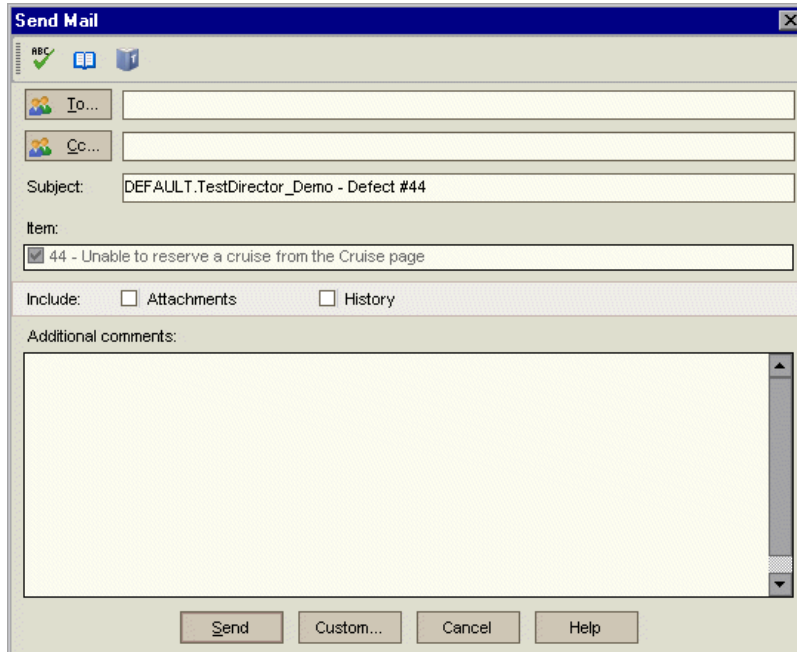
1 Make sure that the Defects module is displayed.

If the Defects module is not displayed, click the **Defects** tab.

2 Select a defect.



Select the defect you added in “Adding New Defects” on page 83, and click the **Mail Defects** button. The Send Mail dialog box opens.



3 Type a valid e-mail address.

In the **To** box, type your e-mail address.

4 Include the attachments and history of the defect.

In the **Include** box, select **Attachments** and **History**.

5 Add your comments.

Under **Additional comments**, type: I will send you more information tomorrow.

6 E-mail the defect.

Click **Send**. A message box opens. Click **OK**.

7 View the e-mail.

Open your mailbox and view the defect you sent.

Associating Defects with Tests

You can associate a test in your test plan with a specific defect in the Defects Grid. This is useful, for example, when a new test is created specifically for a known defect. By creating an association, you can determine if the test should be run based on the status of the defect. Note that any requirements covered by the test are also associated with the defect.

You can also create an association during a manual test run. When you submit a defect, TestDirector automatically creates an association between the test run and the new defect.

In the following exercise, you will associate your defect with the **Cruise Booking** test in the Test Plan module, and view the associated test in the Defects Grid.

To associate a defect with a test:

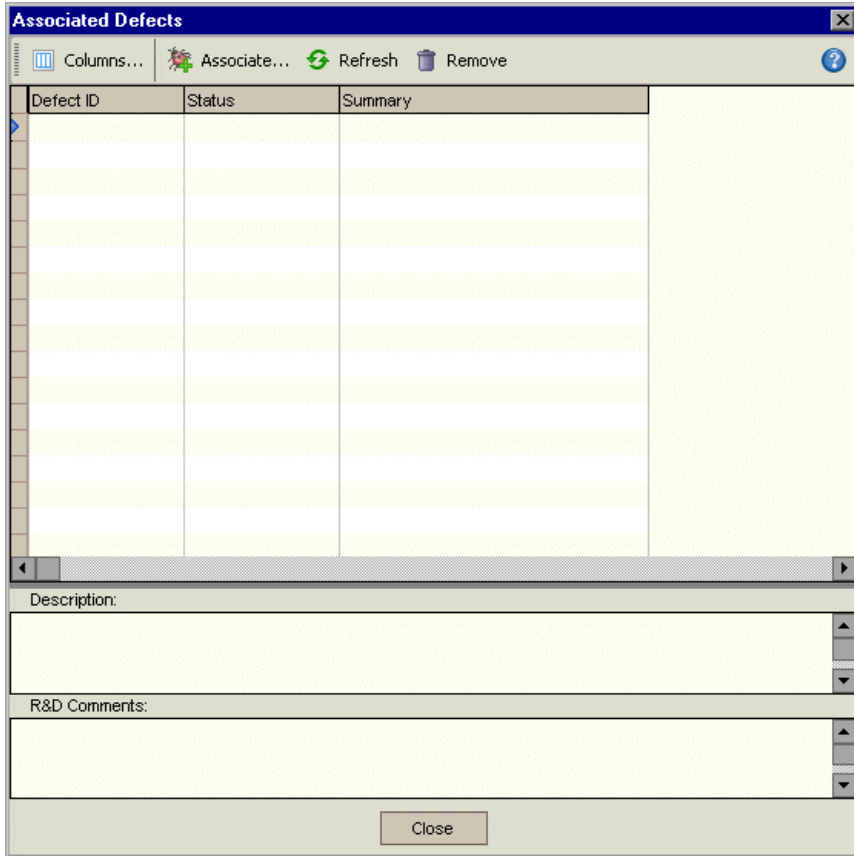
1 Display the Test Plan module.

Click the **Test Plan** tab.

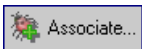
2 Select the Cruise Booking test.

In the test plan tree, under **Cruises**, expand **Cruise Reservation** and right-click the **Cruise Booking** test. Choose **Associated Defects**.

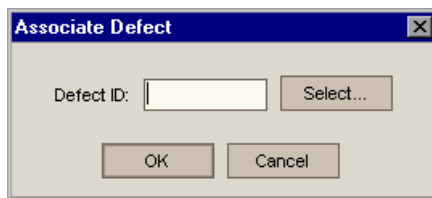
The Associated Defects dialog box opens.



3 Add an associated defect.



Click the **Associate** button. The Associate Defect dialog box opens.



Click the **Select** button. The Associate Defect dialog box opens.

Select your defect and click the **Associate** button.



Note: If you cannot find your defect in the Associated Defect grid, click the **Clear Filter/Sort** button to clear the filter that is applied to the grid.

Click **OK** to confirm. Click **Close** to close the Associate Defect dialog box. TestDirector adds your defect to the Associated Defects grid.

Click **Close** to close the Associated Defects dialog box.

4 View the associated test in the Defects Grid.

Click the **Defects** tab.

Select your defect in the Defects Grid, and choose **View > Associated Test**. The Associated Test dialog box opens.

- The **Details** tab displays a description of the test.
- The **Design Steps** tab lists the test steps.
- The **Test Script** tab displays the test script if the test is automated.
- The **Attachments** tab lists any attachments.
- The **Reqs Coverage** tab displays the requirements covered by the test.

Creating Favorite Views

A *favorite view* is a view of a TestDirector window with the settings you applied to it. For example, in the Defects Grid, you may want to apply a filter to display only the defects that were detected by you, are assigned to you, or have the status “Not Closed”.

Note: You can also save a favorite view of the Test Grid, Execution Grid, Defects Grid, and all TestDirector reports and graphs.

In the following exercise, you will create a favorite view in the Defects Grid.

To create a favorite view:

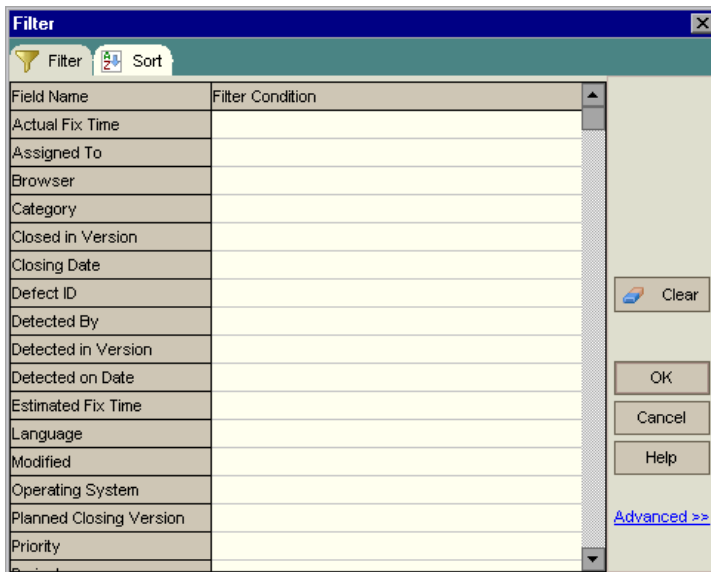
1 Make sure that the Defects module is displayed.

If the Defects module is not displayed, click the **Defects** tab.

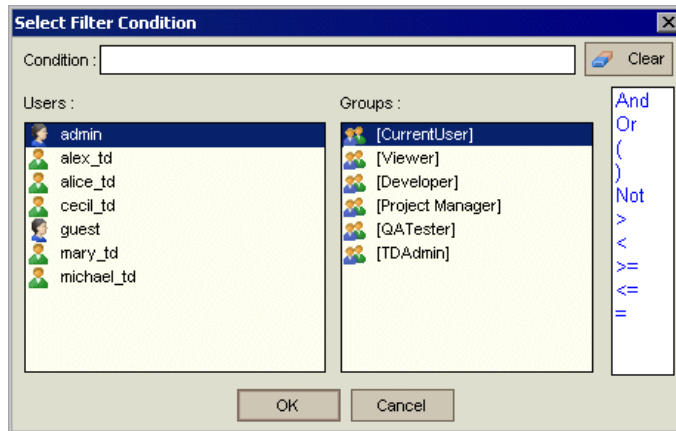
2 Define a filter to view defects you detected that are not closed.



Click the **Set Filter/Sort** button. The Filter dialog box opens.



Click the **Filter Condition** box that corresponds to **Detected By**. Click the **Browse** button. The Select Filter Condition dialog box opens.



Under **Users**, select your TestDirector login user name (**alice_td**, **cecil_td**, or **michael_td**). Click **OK** to close the Select Filter Condition dialog box.

For the **Status** field, click the **Filter Condition** box. Click the **Browse** button. The Select Filter Condition dialog box opens.

In the right pane, select the logical expression **Not**.

In the left pane, select **Closed**.



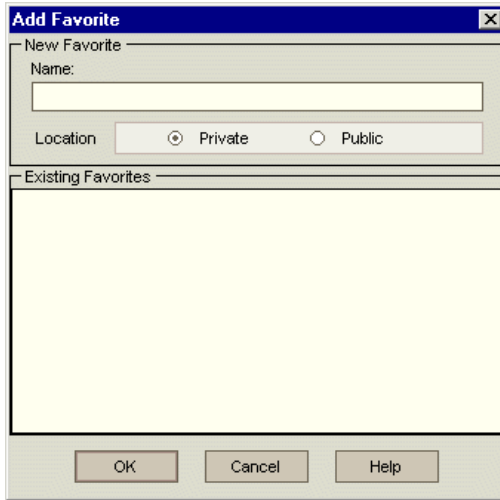
Click **OK** to close the Select Filter Condition dialog box.

Click **OK** to close the Filter dialog box. The Defects Grid displays the defects you detected that are not closed.

3 Add a favorite view.



Click the **Favorites** button and choose **Add to Favorites**. The Add Favorite dialog box opens.



In the **Name** box, type: My detected defects (status Not Closed).

You can add a favorite view to either a *public* folder or a *private* folder. Views in the public folder are accessible to all users. Views in the private folder are accessible only to the person who created them. For the purpose of this exercise, select **Private**.

Click **OK**. The new view name is added to the **Favorite** list.

Project: TestDirector_Demo [alex_td] REQUIREMENTS TEST PLAN TEST LAB DEFECTS TOOLS HELP LOGOUT										
Defects Search View Favorites Analysis										
Defects Grid										
Add Defect... [Icons] Favorite: <None> [Star]										
	Defect ID	Status	Category	Detected in	Planned Clos	Priority	Sever		Detected By	Assigned T
	31	Open	UI Suggest	Version 1.0	Future Versi	1-Low			michael_td	mary_td
	32	Open	Defect	Version 1.0	Version 1.01	4-Very High	4-Very High	The average response time for the Sign-On Page Down	cecil_td	mary_td
	33	Open	Defect	Version 1.0	Future Versi	2-Medium	2-Medium	The average response time for the Sign-Off Page Down	cecil_td	mary_td
	34	Closed	Defect	Version 1.0	Version 1.0	4-Very High	4-Very High	Clicking on Back to Home button opens Find Flight page	alice_td	peter_td
	35	Open	Defect	Version 1.0	Version 1.01	3-High	3-High	The average response time for the Search Flight transac	cecil_td	mary_td
	36	Open	Defect	Version 1.0	Version 1.01	4-Very High	4-Very High	The average response time for the Book Flight transacti	cecil_td	mary_td
	37	Open	Defect	Version 1.0	Version 1.01	4-Very High	4-Very High	The cost doesn't depend on Service class	alice_td	peter_td
	38	Closed	Defect	Version 1.0	Version 1.0	3-High	2-Medium	Cancel All Reservation button is available when there ar	alice_td	peter_td
	39	Open	Defect	Version 1.0	Version 1.01	3-High	4-Very High	There is no confirmation message on Cancel Reservatio	alice_td	mary_td
	42	New	Defect	Version 1.01			5-Urgent	Profile is registered when Password is empty	cecil_td	
	43	New	Defect	Version 1.01			4-Very High	All fields in Edit profile form are empty	michael_td	
	44	New	Defect	Version 1.01		3-High	2-Medium	Unable to reserve a cruise from the Cruise page	alex_td	



Now that you are familiar with how to add new defects, search for similar defects, update defects, e-mail defects, associate defects with tests, and create favorite views, you can proceed to Lesson 6, “Tracing Changes”. In Lesson 6, you will learn how to trace changes made to your requirements, tests, and defects.

6

Tracing Changes

You can instruct TestDirector to create alerts automatically and send e-mails to notify those responsible when changes occur in your project that may impact the testing process. You can also add your own follow up alerts.

To generate automatic *traceability alerts*, your TestDirector administrator can activate traceability notification rules, which are based on associations you make in TestDirector between requirements, tests, and defects. When an entity in your project changes, TestDirector notifies you of any associated entities that may be impacted by the change.

TestDirector also enables you to add your own *follow up flag* to a specific test, test instance, or defect to remind yourself to follow up on an issue. When the follow up date arrives, TestDirector sends you an e-mail.

In this lesson, you will learn about:

- Triggering a Traceability Alert
- Viewing the Traceability Alert
- Creating Follow Up Alerts

Triggering a Traceability Alert

When changes are made to a requirement, test, or defect in your project, TestDirector notifies you by adding a trace changes flag to the changed entity and/or sending you an e-mail notification.

TestDirector can generate traceability alerts for the following changes:

- ▶ When a requirement changes (excluding change of status), TestDirector notifies the designer(s) of the associated test(s).
- ▶ When a requirement changes and it has an associated test(s), TestDirector notifies all project users.
- ▶ When the status of a defect changes to “Fixed”, TestDirector notifies the responsible tester of the associated test instance.
- ▶ When a test runs successfully, TestDirector notifies the user(s) assigned to the associated defect(s).

In the following exercise, you will cause TestDirector to generate a traceability alert for a test by changing the associated requirement. To do this, you will log on as a different user and make a change to the **Cruise Booking** requirement. Later, when you log on again as yourself, you will see the associated **Cruise Booking** test flagged.

To trigger a traceability alert:

1 Log on to the TestDirector_Demo project as a different user.

A purple rectangular button with the word "LOGOUT" in white, uppercase letters.

If the **TestDirector_Demo** project is already open, log off by clicking the **Logout** button on the right side of the window.

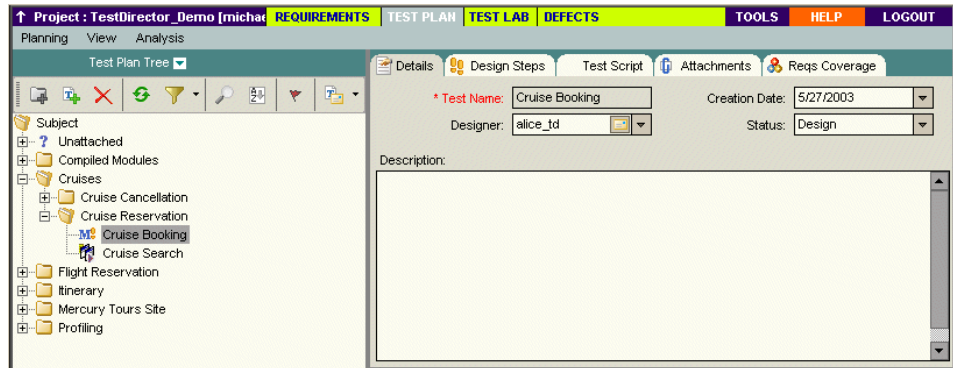
An orange rounded rectangular button with the word "Login" in white, lowercase letters.

Log on to the **TestDirector_Demo** project using a different user name than you used in the previous lessons. You can log on as: **alice_td**, **cecil_td**, or **michael_td**.

For more information on opening the **TestDirector_Demo** project, see “Starting TestDirector,” on page 5.

2 Display the Test Plan module and view the designer.

Click the **Test Plan** tab. The Test Plan module displays the test plan tree. Expand the **Cruises** and **Cruise Reservation** subject folders and select the **Cruise Booking** test.



In the Details tab in the right pane, the **Designer** box displays the user name you used to design the **Cruise Booking** test. Note that when an associated requirement changes, the designer is the one who sees the traceability notification.

3 Display the requirements tree.

Click the **Requirements** tab and make sure that the **Document View** of the requirements tree is displayed.

4 Select the requirement that you want to change.

Under **Cruise Reservation**, select the **Cruise Booking** requirement.

ReqID	Name	Direct Cover Status	Author	Reviewed	Creation Time
[RQ0003]	Mercury Tours Application	Failed	alex_td	Reviewed	9:07:29 AM
[RQ0140]	Cruise Reservation	No Run	michael_td	Not Reviewed	12:09:14 PM
[RQ0141]	Cruise Search	No Run	michael_td	Not Reviewed	12:10:44 PM
[RQ0142]	Cruise Booking	No Run	michael_td	Not Reviewed	12:11:00 PM

5 Change the priority of the requirement.

Click the **Priority** down arrow and select **5-Urgent**.

ReqID	Name	Direct Cover Status	Author	Reviewed	Creation Time	Creation...	Priority
[RQ0003]	Mercury Tours Application	Failed	alex_td	Reviewed	9:07:29 AM	1/9/2001	5-Urgent
[RQ0140]	Cruise Reservation	Not Covered	michael_td	Not Reviewed	12:09:14 PM	5/27/2003	4-Very High
[RQ0142]	Cruise Search	No Run	michael_td	Not Reviewed	12:11:00 PM	5/27/2003	4-Very High
[RQ0141]	Cruise Booking	No Run	michael_td	Not Reviewed	12:10:44 PM	5/27/2003	5-Urgent
[RQ0020]	Application Security	Failed	alex_td	Reviewed	1:15:09 PM	1/9/2001	1-Low
[RQ0153]	Sightseeing Reservation	Not Covered	michael_td	Not Reviewed	6:42:06 AM	6/8/2003	2-Medium
[RQ0042]	Application Client System	Failed	shelly_td	Reviewed	11:23:56 AM	1/15/2001	3-High
[RQ0045]	Application Usability	Failed	shelly_td	Reviewed	11:33:15 AM	1/15/2001	4-Very High

This change causes TestDirector to generate a traceability alert for the test associated with this requirement. TestDirector also sends an e-mail to the designer of the test. In the next exercise, you will log on as the designer and view the traceability alert.

6 Log off the TestDirector_Demo project.

LOGOUT

Click the **Logout** button on the right side of the window.

Viewing the Traceability Alert

You can view traceability changes for a single entity or for all changed entities in your project. An entity can be a test, a test instance, or a defect.

In the following exercise, you will view the traceability alert for the **Cruise Booking** test that TestDirector generated as a result of the change you made to the **Cruise Booking** requirement.

To view the traceability alert:

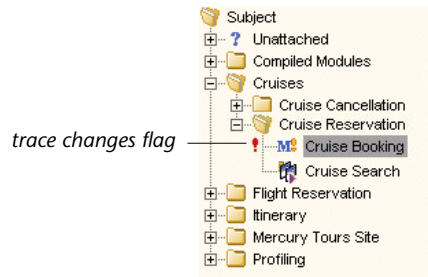
1 Log on to the TestDirector_Demo project as the designer of the Cruise Booking test.

Login


Log on to the **TestDirector_Demo** project. Make sure you use the same name you used in previous lessons as the designer of the **Cruise Booking** test.

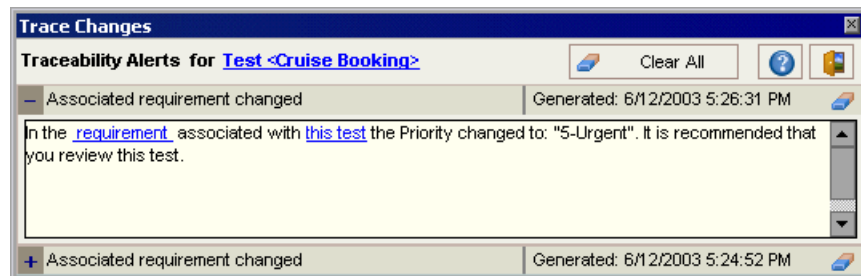
2 Display the Test Plan module and view the traceability flag.

Click the **Test Plan** tab. In the test plan tree, under **Cruises**, expand **Cruise Reservation** to display the **Cruise Booking** test. The **Cruise Booking** test has a trace changes flag, indicating that a change was made to an associated requirement.



3 View the traceability alert.

Click the **Trace Changes** flag  for the **Cruise Booking** test. The Trace Changes dialog box opens.



Click the **requirement** link. TestDirector highlights the **Cruise Booking** requirement in the Requirements module.

4 View all traceability alerts.




In the common TestDirector toolbar, click the **Trace All Changes** button. The Trace All Changes dialog box opens and displays a list of all traceability changes in your TestDirector project. For this exercise, TestDirector displays the same change as above.



Click **Close** to close the Trace All Changes dialog box.

Creating Follow Up Alerts

TestDirector enables you to add your own alerts to remind yourself to follow up on outstanding issues. In this exercise, you will add a *follow up* flag to a defect whose status you want to check one week from today. A follow up flag is a gray flag icon  that TestDirector adds to the record.

When you add a follow up flag, TestDirector also adds a yellow information bar that reminds you about the follow up alert. When the follow up date arrives, TestDirector sends you an e-mail and changes the flag icon to red.

Note that follow up flags are specific to your user name, meaning that only you can see your follow up alerts.

To create a follow up alert:

1 Display the Defects module.

Click the Defects tab.

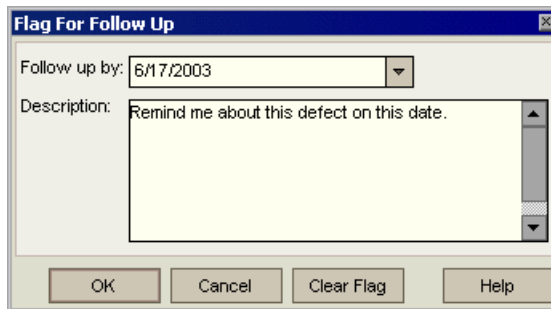
2 Select the defect that you want to flag to follow up.

In the Defects Grid, select a defect.

3 Create the follow up alert.




Click the **Flag for Follow Up** button. The Flag For Follow Up dialog box opens.



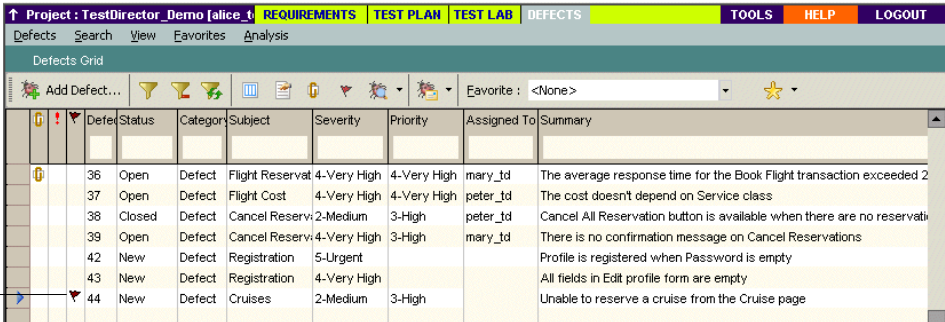
Do the following:

Follow up by: Select the date one week from today.

Description: Type: Remind me about this defect on this date.

Click **OK**. TestDirector adds the flag icon  to the defect record.

follow up flag →

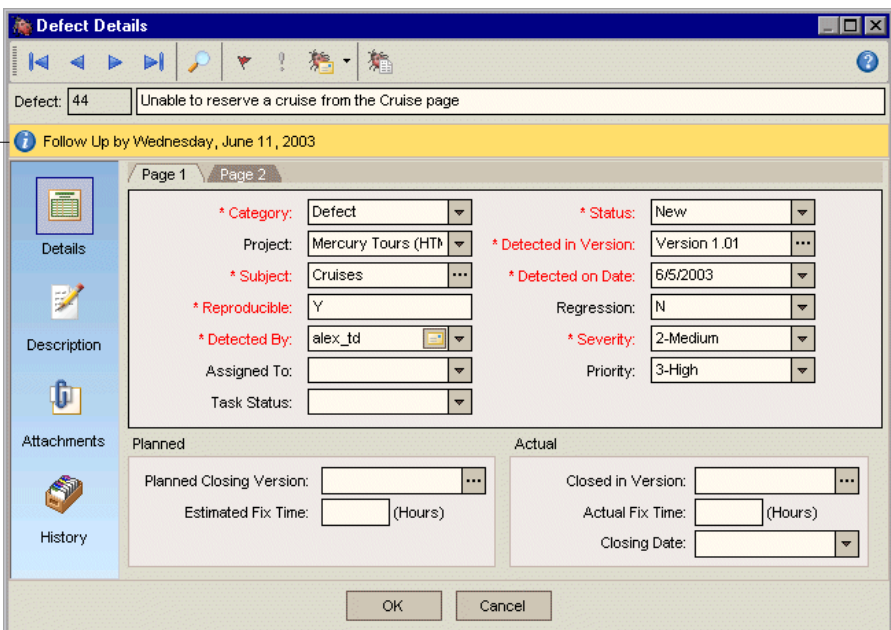


Defect ID	Status	Category	Subject	Severity	Priority	Assigned To	Summary
36	Open	Defect	Flight Reservat	4-Very High	4-Very High	mary_td	The average response time for the Book Flight transaction exceeded 2
37	Open	Defect	Flight Cost	4-Very High	4-Very High	peter_td	The cost doesn't depend on Service class
38	Closed	Defect	Cancel Reservi	2-Medium	3-High	peter_td	Cancel All Reservation button is available when there are no reservati
39	Open	Defect	Cancel Reservi	4-Very High	3-High	mary_td	There is no confirmation message on Cancel Reservations
42	New	Defect	Registration	5-Urgent			Profile is registered when Password is empty
43	New	Defect	Registration	4-Very High			All fields in Edit profile form are empty
44	New	Defect	Cruises	2-Medium	3-High		Unable to reserve a cruise from the Cruise page

4 Display the information bar for your follow up alert.

Double-click the defect with the follow up flag. The Defect Details dialog box opens and displays a yellow information bar with your follow up alert.

follow up alert information bar →



Defect: 44 Unable to reserve a cruise from the Cruise page

Follow Up by Wednesday, June 11, 2003

Page 1 Page 2

* Category: Defect * Status: New

Project: Mercury Tours (HTM) * Detected in Version: Version 1.01

* Subject: Cruises * Detected on Date: 6/5/2003

* Reproducible: Y Regression: N

* Detected By: alex_td * Severity: 2-Medium

Assigned To: Priority: 3-High

Task Status:

Planned Actual

Planned Closing Version: Closed in Version:

Estimated Fix Time: (Hours) Actual Fix Time: (Hours)

Closing Date:

OK Cancel

Click **Cancel** to close the dialog box.



Now that you are familiar with tracing changes, you can proceed to Lesson 7, “Analyzing the Testing Process.” In Lesson 7, you will learn how to generate reports and graphs to help you analyze the testing process.

7

Analyzing the Testing Process

TestDirector reports and graphs help you assess the progress of requirements, the test plan, test runs, and defect tracking. You can generate reports and graphs at any time during the testing process from each TestDirector module, using default or customized settings. When customizing a report or graph, you can apply filters and sort conditions, and display information according to your specifications. You can also save your settings as favorite views and reload them as needed.

In this lesson, you will learn about:

- Generating Reports
- Generating Graphs

Generating Reports

You can generate a report from any TestDirector module. Each TestDirector module contains various report options. After you generate a report, you can customize report properties to display information according to your specifications.

In the following exercise, you will generate a standard requirements report, customize the report for a specific user name, and add it to your favorites list.

To generate a report:

1 Open the TestDirector_Demo project.

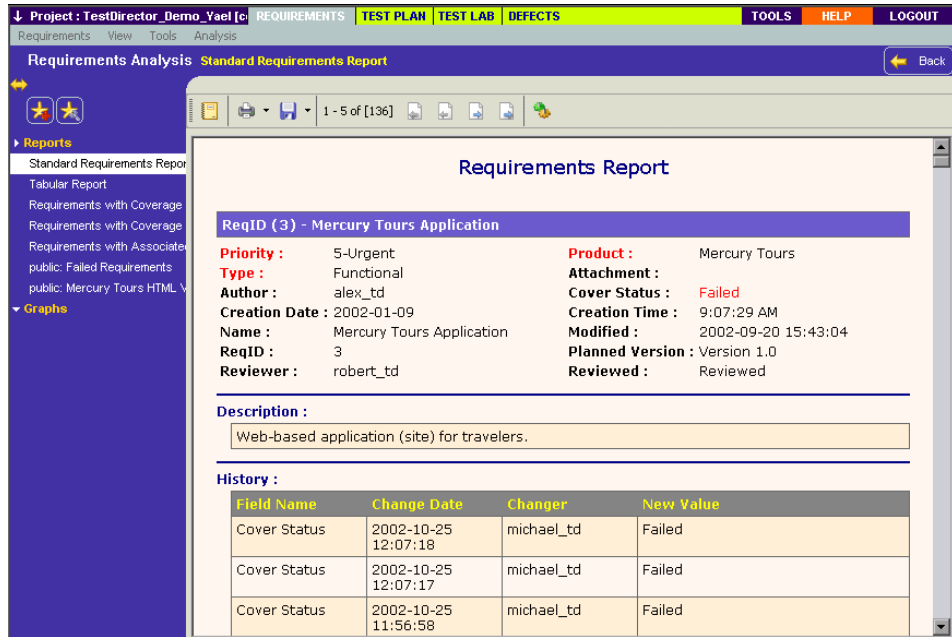
If the **TestDirector_Demo** project is not already open, log on to the project. For more information, see “Starting TestDirector,” on page 5.

2 Display the Requirements module.

Click the **Requirements** tab. The Requirements module displays the requirements tree.

3 Choose a report.

Choose **Analysis > Reports > Standard Requirements Report**. The report opens, displaying default data.



4 Display the customization options.



Click the **Configure Report and Sub-Reports** button. The Report Customization page opens with the default options selected.

Requirement Report

Page

Limit items per page to

All items in one page

Template

URL

Filter

Fields

All Fields (auto-layout)

Custom Fields (layout)

Keep Parent-Child Order (disables filter sort order)

Show Paragraph Number

Show Full Coverage (applied to test sub-report)

Grid View

Attachments

History

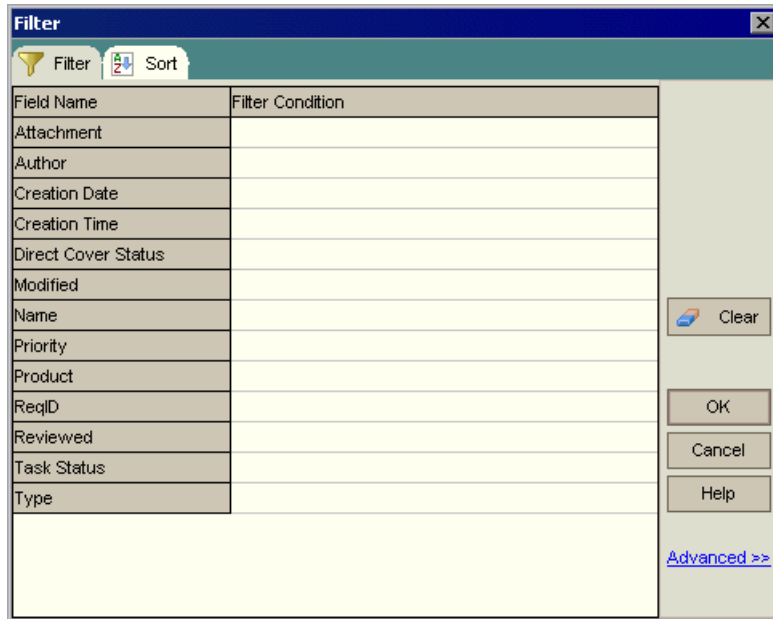
5 Set the number of items per display page.

Select **All items in one page** to display all items in one page.

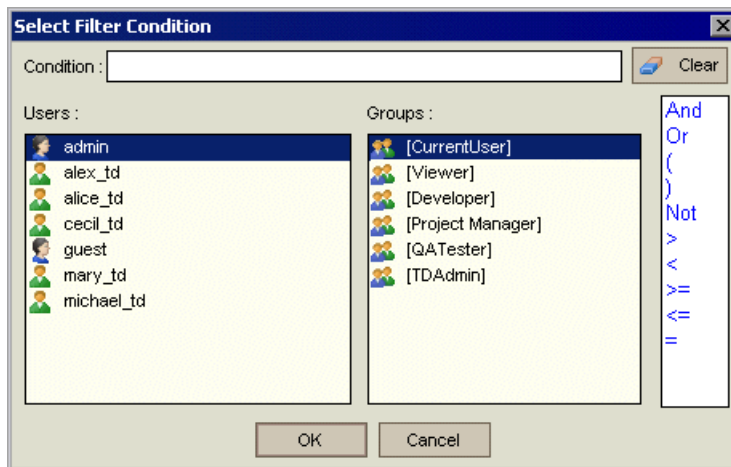
6 Define a filter to view requirements created by a specific user name.



Click the **Set Filter/Sort** button. The Filter dialog box opens.



For the **Author** field, click the **Filter Condition** box. Click the **Browse** button. The Select Filter Condition dialog box opens.



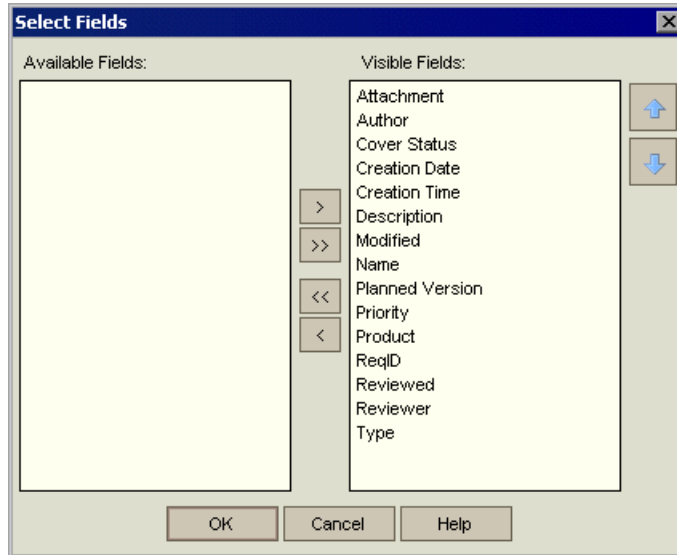
Under **Users**, select your TestDirector login user name (**alice_td**, **cecil_td**, or **michael_td**). Click **OK** to close the Select Filter Condition dialog box.

Click **OK** to close the Filter dialog box.

7 Specify the fields and the order in which they are displayed.



Under **Fields**, select **Custom Fields (layout)** and click the **Select Fields** button. The Select Fields dialog box opens.



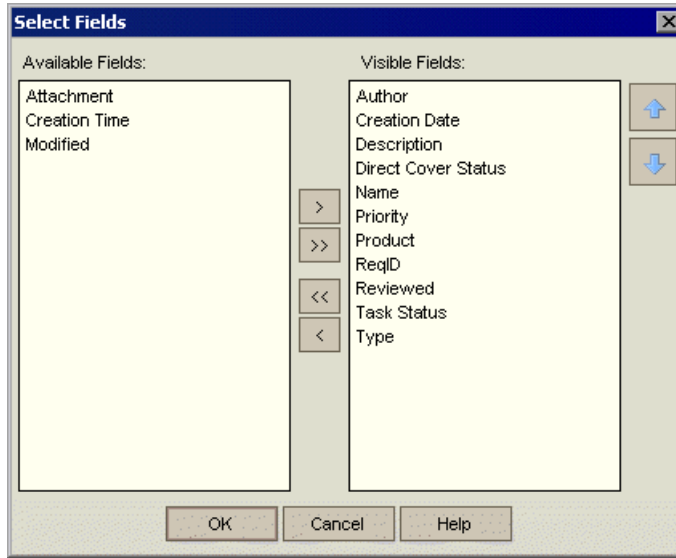
The **Available Fields** box lists fields that are not currently displayed. The **Visible Fields** box lists fields that are currently displayed.



In the **Visible Fields** box, select **Attachment** and click the left arrow button to move the field to **Available Fields**. Move **Creation Time** and **Modified** to **Available Fields**.



To set the appearance order of the visible fields, select **ReqID** and use the up arrow button to move the field under **Author**. Move **Name** and **Description** so that they appear below **ReqID**.



Tip: You can also use a drag-and-drop operation to specify visible fields and the order in which they are displayed.

Click **OK** to close the dialog box and apply the new field order.

8 Clear the Attachments and History options.

Under the list of options, clear the **Attachments** and **History** check boxes.

9 Generate a new report.



Click the **Generate Report** button. TestDirector displays the customized Requirements Report.

Project : TestDirector_Demo [alex_td] REQUIREMENTS TEST PLAN TEST LAB DEFECTS TOOLS HELP LOGOUT

Requirements View Tools Analysis

Requirements Analysis Standard Requirements Report Back

1 - 5 of [9]

Requirements Report

Filter : Author[cecil_td];

ReqID (126) - Connect To Site Transaction

Author :	cecil_td	ReqID :	126
Name :	Connect To Site Transaction	Direct Cover Status :	Failed
Creation Date :	2001-02-28	Priority :	4-Very High
Planned Version :	Version 1.0	Reviewed :	Reviewed
Product :	Mercury Tours	Task Status :	
Reviewer :	alex_td		
Type :	Quality		

Description :

The time it takes to connect to the application home page.

Threshold: 10 seconds.
Plan: 8 seconds.

10 Add the report as a favorite view.



Click the **Add to Favorites** button. The Add Favorite dialog box opens.

In the **Name** box, type: Standard Requirements by Author. Click **OK**. The report is added to the Reports list in the left pane.

11 Close the report.



Click the **Back** button. The Requirements module is displayed.



Tip: You can generate a quick report for a single node in a tree or row in a grid. Select the requirement, test, or defect, and choose **Analysis > Report Selected**.

Generating Graphs

TestDirector graphs help you analyze the relationships between different types of data in a project. Each TestDirector module contains various graph options. After you generate a graph, you can customize graph properties to display information according to your specifications.

In the following exercise, you will generate a defects graph to summarize the defects by status and priority level.

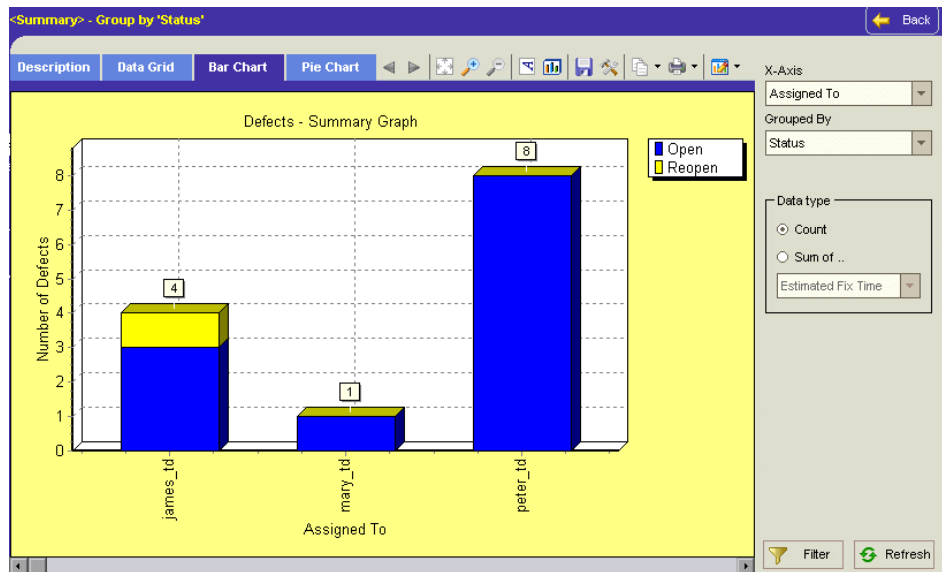
To generate a graph:

1 Display the Defects module.

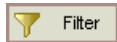
Click the **Defects** tab. The Defects module displays the Defects Grid.

2 Choose a graph.

Choose **Analysis > Graphs > Summary - Group by 'Status'**. The Defects - Summary Graph opens. By default, the graph is grouped by **Status**.



3 Clear the default filter.

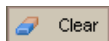


Click the **Filter** button. The Filter dialog box opens.

 A screenshot of the "Filter" dialog box. It has a blue title bar with the word "Filter" and a close button (X). The main area is a table with two columns: "Field Name" and "Filter Condition". The "Detected By" field is currently set to "[CurrentUser]". To the right of the table are four buttons: "Clear" (with a blue eraser icon), "OK", "Cancel", and "Help". Below the "Help" button is a blue link that says "Advanced >>".

Field Name	Filter Condition
Actual Fix Time	
Assigned To	
Browser	
Category	
Closed in Version	
Closing Date	
Defect ID	
Detected By	[CurrentUser]
Detected in Version	
Detected on Date	
Estimated Fix Time	
Language	
Modified	
Operating System	
Planned Closing Version	
Priority	
Project	
Regression	

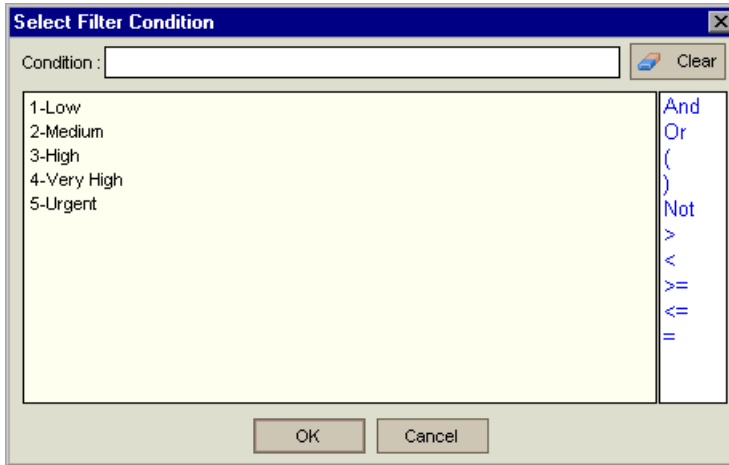
By default, the **Detected By** field is set to the current user name—your TestDirector login user name (**alice_td**, **cecil_td**, or **michael_td**).



Click the **Clear** button. TestDirector removes the applied filter.

4 Define a filter to view defects with high to urgent priority.

In the Filter dialog box, for the **Priority** field, click the **Filter Condition** box. Click the **Browse** button. The Select Filter Condition dialog box opens.



In the right pane, select the logical expression \geq .

In the left pane, select **3-High**.



Click **OK** to close the Select Filter Condition dialog box.

5 Define a filter to view defects that are not closed.

For the **Status** field, click the **Filter Condition** box. Click the **Browse** button. The Select Filter Condition dialog box opens.

In the right pane, select the logical expression **Not**.

In the left pane, select **Closed**.



Click **OK** to close the Select Filter Condition dialog box.

Click **OK** to close the Filter dialog box.

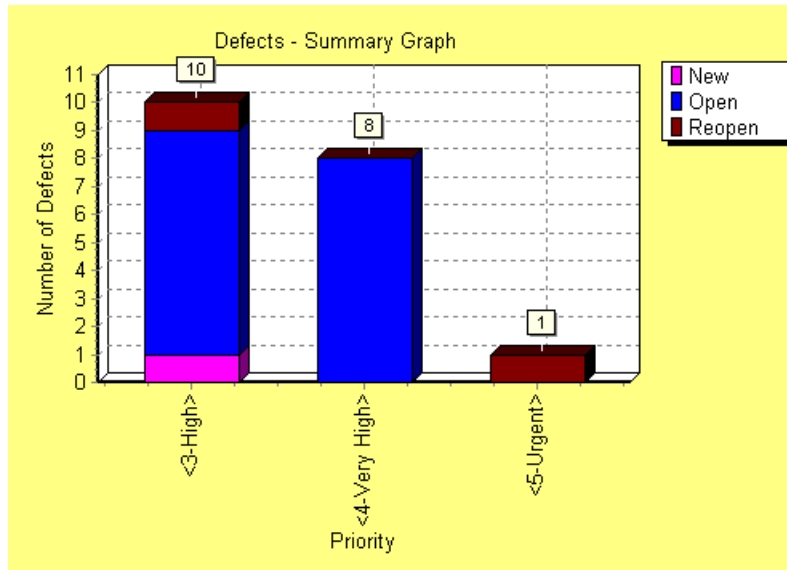
6 Set the X-axis of the graph.

From the **X-Axis** list on the right side of the window, select **Priority** to view the number of defects by priority.

7 Refresh the graph.



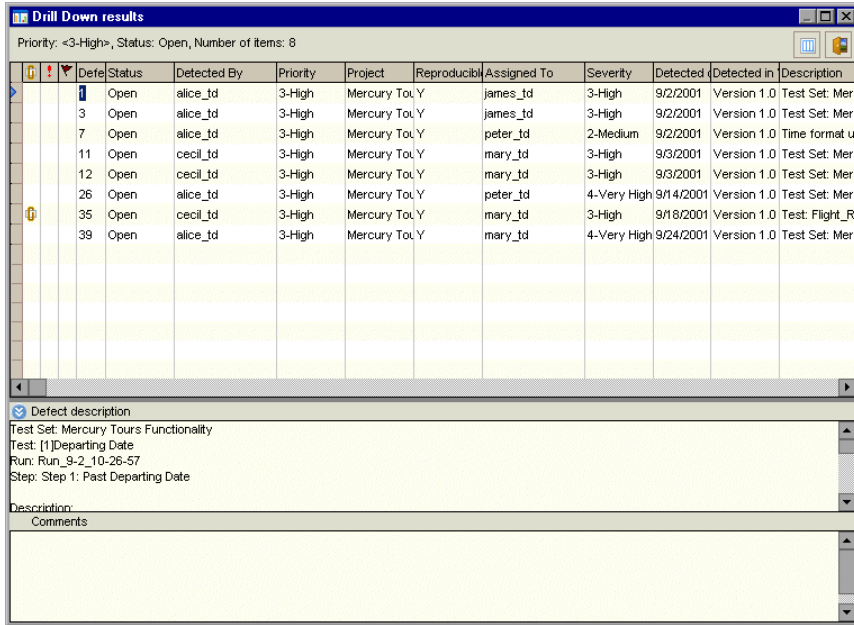
Click the **Refresh** button. TestDirector displays a new graph.



In this exercise, ten defects are assigned **3-High**; eight defects are assigned **4-Very High**; and one defect is assigned **5-Urgent**.

8 Display additional defect details.

Click a bar segment in the graph. The Drill Down Results dialog box opens and displays the defects that define the bar segment.



Drill Down results
Priority: <3-High>, Status: Open, Number of items: 8

Defect ID	Status	Detected By	Priority	Project	Reproducibility	Assigned To	Severity	Detected	Detected in	Description
1	Open	alice_td	3-High	Mercury Tou	Y	james_td	3-High	9/2/2001	Version 1.0	Test Set: Merc
3	Open	alice_td	3-High	Mercury Tou	Y	james_td	3-High	9/2/2001	Version 1.0	Test Set: Merc
7	Open	alice_td	3-High	Mercury Tou	Y	peter_td	2-Medium	9/2/2001	Version 1.0	Time format u
11	Open	cecil_td	3-High	Mercury Tou	Y	mary_td	3-High	9/3/2001	Version 1.0	Test Set: Merc
12	Open	cecil_td	3-High	Mercury Tou	Y	mary_td	3-High	9/3/2001	Version 1.0	Test Set: Merc
26	Open	alice_td	3-High	Mercury Tou	Y	peter_td	4-Very High	9/14/2001	Version 1.0	Test Set: Merc
35	Open	cecil_td	3-High	Mercury Tou	Y	mary_td	3-High	9/18/2001	Version 1.0	Test: Flight_R
39	Open	alice_td	3-High	Mercury Tou	Y	mary_td	4-Very High	9/24/2001	Version 1.0	Test Set: Merc

Defect description
 Test Set: Mercury Tours Functionality
 Test: [1]Departing Date
 Run: Run_9-2_10-26-57
 Step: Step 1: Past Departing Date

Description:
 Comments



Click **Close** to close the Drill Down Results dialog box.

9 Display other graph views.

Click the **Pie Chart** tab to display the graph as a pie graph.

Click the **Data Grid** tab to display the graph as a grid.

10 Close the graph.



Click the **Back** button. The Defects module is displayed.



Now that you are familiar with generating reports and graphs, you can proceed to Lesson 8, "Where Do You Go from Here?". In Lesson 8, you will apply the TestDirector concepts and skills you learned to your own application.

8

Where Do You Go from Here?

Now that you have completed exercises in Lessons 1 through 6, you are ready to apply the TestDirector concepts and skills you learned to your own application.

In this lesson, you will learn about:

- Getting Started
- Getting Additional Information

Getting Started

Test management with TestDirector includes four phases: specifying requirements, planning tests, running tests, and tracking defects. Throughout each phase, you can analyze data by generating detailed reports and graphs.

To get started:

1 Analyze your application and determine your testing requirements.

To specify your testing requirements, consider the following steps:

- Examine the application documentation to determine your testing scope—test goals, objectives, and strategies.
- Build a requirements tree to define your overall testing requirements.
- For each requirement topic in the requirements tree, create a list of detailed testing requirements. Describe each requirement, assign it a priority level, and add attachments if necessary.

- ▶ Generate reports and graphs to assist you in analyzing your testing requirements. Review your requirements to ensure they meet your testing scope.

2 Create a test plan, based on your testing requirements.

To create a test plan, consider the following steps:

- ▶ Examine your application, system environment, and testing resources to determine your testing goals.
- ▶ Divide your application into modules or functions to be tested. Build a test plan tree to hierarchically divide your application into testing units, or subjects.
- ▶ Determine the types of tests you need for each module. Add a basic definition of each test to the test plan tree.
- ▶ Link each test to a testing requirement(s).
- ▶ Develop manual tests by adding steps to the tests in your test plan tree. Test steps describe the test operations and the expected outcome of each test. Decide which tests to automate.
- ▶ Create test scripts for tests that you want to automate. You can automate tests using a Mercury Interactive testing tool, a custom testing tool, or a third-party testing tool.
- ▶ Generate reports and graphs to assist in analyzing test planning data. Review your tests to determine their suitability to your testing goals.

3 Create test sets and perform test runs.

- ▶ Define groups of tests to meet the various testing goals in your project. These might include, for example, testing a new application version or a specific function in an application. Decide which tests to include in each test set.
- ▶ Schedule test execution and assign tasks to application testers.
- ▶ Execute the tests in your test set automatically or manually.
- ▶ View the results of your test runs to determine if a defect was detected in your application. Generate reports and graphs to help analyze these results.

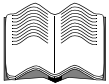
4 Submit defects detected in your application and track the progress of defect fixes.

- Submit new defects detected in your application. Quality assurance testers, developers, project managers, and end users can add defects during any phase of the testing process.
- Review new defects and determine which ones should be fixed.
- Correct the defects that you decide to fix.
- Test a new build of your application. Repeat this process until defects are fixed.
- Generate reports and graphs to assist in analyzing the progress of defect fixes and to help determine when to release the application.

Getting Additional Information

For more information on TestDirector, refer to the documentation set and online resources.

TestDirector Documentation Set



In addition to this guide, TestDirector comes with the following printed documentation:

TestDirector Installation Guide explains how to install TestDirector and the client database software needed to connect TestDirector to your databases.

TestDirector User's Guide explains how to use TestDirector to organize and execute all phases of the testing process. It describes how to define requirements, plan tests, run tests, and track defects.

TestDirector Administrator's Guide explains how to customize projects using Project Customization, and how to create and maintain projects using the Site Administrator.

TestDirector Open Test Architecture Guide explains how to use TestDirector's open test architecture to integrate your own configuration management, defect tracking, and home-grown testing tools with a TestDirector project. It includes a complete reference to TestDirector's new COM-based API.

Online Resources



TestDirector includes the following online resources:

Read Me First provides last-minute news and information about TestDirector.

What's New in TestDirector describes the newest features in the latest versions of TestDirector.

Books Online displays the complete documentation set in .PDF format. Online books can be read and printed using Adobe Reader which can be downloaded from the Adobe Web site (<http://www.adobe.com>).

TestDirector Online Help provides immediate answers to questions that arise as you work with TestDirector. It describes menu commands and dialog boxes, and shows you how to perform TestDirector tasks. Check Mercury Interactive's Customer Support Web site (<http://support.mercury.com>) for updates to TestDirector help files.

Customer Support Online uses your default Web browser to open Mercury Interactive's Customer Support Web site. The URL for this Web site is <http://support.mercury.com>. This site enables you to browse the Mercury Support Knowledge Base and add your own articles, post to and search user discussion forums, submit support requests, download patches and updated documentation, and more.

Mercury Interactive on the Web uses your default Web browser to open Mercury Interactive's home page. This site provides the most up-to-date information on Mercury Interactive and its products. This includes new software releases, seminars and trade shows, customer support, educational services, and more. The URL for this Web site is *<http://www.mercury.com>*.

