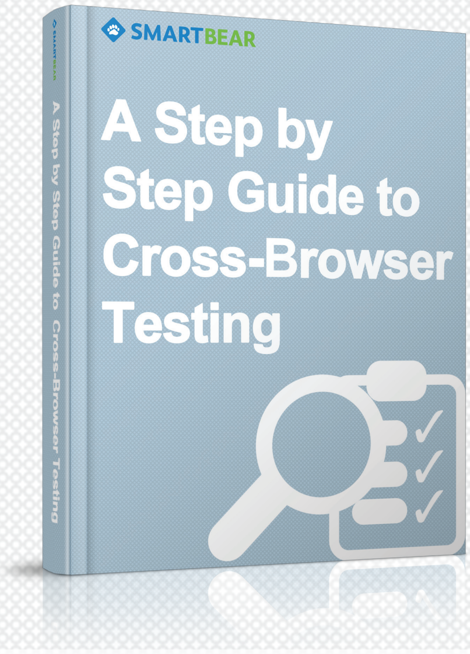
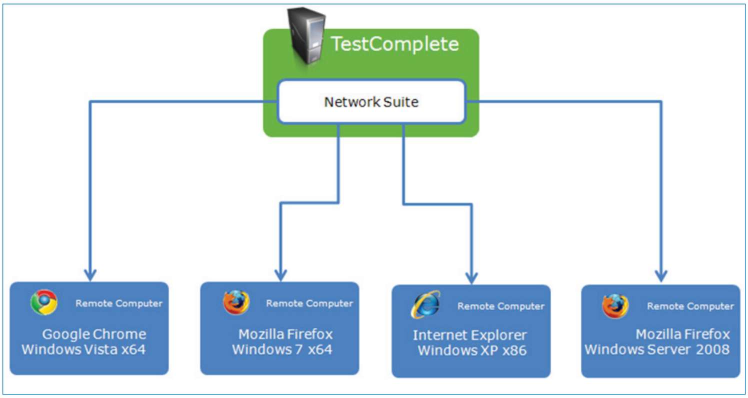
TestComplete
by SMARTBEAR



Cross-browser testing is essential for ensuring the application compatibility with different web browsers. Ideally, cross-browser testing should involve multiple computers with various combinations of operating systems. This also includes the need to centrally organize simultaneous or sequential test execution across multiple computers.

SMARTBEAR

SmartBear TestComplete provides an integrated solution for functional web testing across all major browsers as well as for distributed testing and test monitoring on remote computers, whether physical, virtual or cloud. In this article we'll explain how to leverage TestComplete to organize an automated framework for distributed cross-browser testing



## *Four-Step Distributed Testing Process*

Distributed testing requires multiple computers to carry out coordinated test actions. When using TestComplete, the distributed test (also called network suite) is initiated from the master computer and executed on the remote computers called slave computers. All of the distributed testing settings— the list of remote computers, the tests to be run on them and so on—are specified in a single test project on the master computer.

Distributed cross-browser testing process consists of the following four phases:

1. Preparing master and remote computers:
    a. Installing and configuring web browsers
    b. Configuring operating systems for distributed testing

2. Creating cross-browser web tests on the master computer

3. Configuring the distributed test on the master computer:

    a. Adding the Network Suite item to the master project (this item adds the distributed testing functionality)

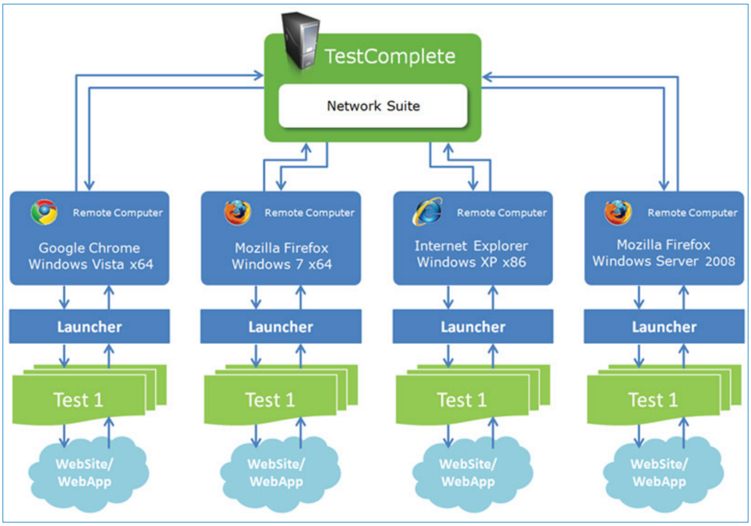
    b. Setting up connections to the remote computers

    c. Setting up automatic deployment of test projects to the remote computers

    d. Selecting test to be run on the remote computers

4. Running and monitoring the distributed test

We will describe each of these in more detail below.



## *1. Preparation Steps*

To run cross-browser tests on multiple computers using TestComplete, you need to prepare these computers to enable distributed test execution. For example, you need to install and configure TestComplete (or its lightweight test runner TestExecute, see below) and the needed browsers on each computer, allow TestComplete, its service and ports through firewall and so on. For details, please refer to TestComplete documentation:

» Distributed Testing - Requirements

### Tip: Gain maximum leverage by installing TestExecute on remote machines

The distributed testing scenario has one aspect in particular that you can use to your advantage – the tests are typically  created and configured on only one (master) computer  and are executed on multiple (slave) computers. Hence all the powerful features of TestComplete would not be required on remote machines, and it is redundant and more costly to have TestComplete installed on every remote machine.

A more favorable solution, in this case, would be to equip remote computers with TestExecute - a resource-friendly tool that was specifically engineered for running TestComplete test projects on computers where TestComplete is not installed. TestExecute is less expensive than TestComplete, so you can afford several TestExecute instances for the price of one TestComplete license, which  comes bundled with TestComplete Enterprise edition. Additional licenses can be purchased separately. TestExecute offers a Floating User license, so it is not locked to some particular machine and can be installed and run on multiple physical and virtual machines. However, the overall number of simultaneously running TestExecute instances should not exceed the number of purchased licenses.

Thus you would use TestComplete as a “designer” for creating and adjusting tests, and TestExecute as a “runner” for running pre-configured tests and obtaining results on the remote machines.

## *2. Creating Cross-Browser Tests on Master Machine*

You start creating a distributed cross-browser test by recording or creating web tests on one specific computer. For example, you can create tests on the master computer— (the one that will be later used to initiate and monitor the distributed test run).

Creating a cross-browser web test includes the following steps:

1. **Configuring web browsers** to eliminate browser-specific behavior and facilitate cross-browser compatibility.

2. **Creating and configuring a test project.** A cross-browser test project must have the Web tree model option set to Tree and the Use legacy web testing features option disabled. These options are set by default in new test projects. If you are using an existing test project, you may need to set these settings manually. For more information about these options, please refer to TestComplete documentation.

3. **Creating or recording the test** in a specific browser (for example, Internet Explorer).

4. **Editing and debugging the test** in the original browser to en sure it plays back correctly.

5. **Running the test against other browsers** to make sure it executes successfully.

## *3. Configuring Tests to Run in All or Specific Browsers*

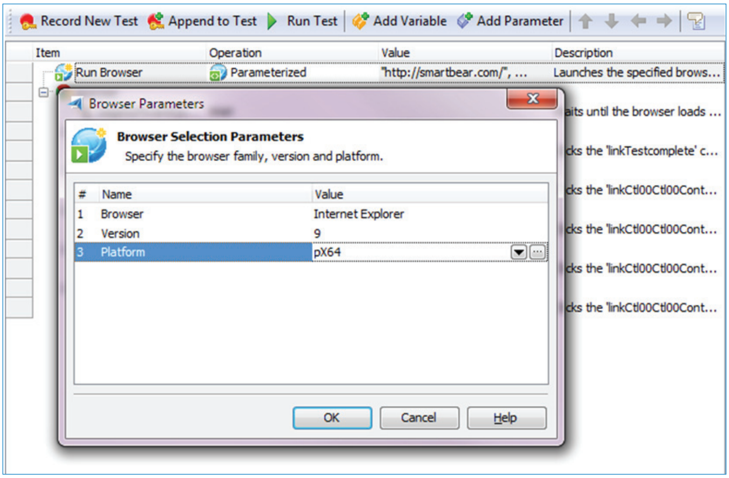
There are two main scenarios for running web tests:

- Running the tests in a specific browser.
- Running the tests against all browsers installed on the computer.

## Running Tests in a Specific Browser and Version

If you need to test a web application in a specific browser, you can simply record the test in this browser. TestComplete will identify the browser used and record the corresponding operations that would launch this browser and open web pages in it.

To configure the test to use specific browser version and/or platform bitness, you can modify the recorded test and change the corresponding operation parameters.



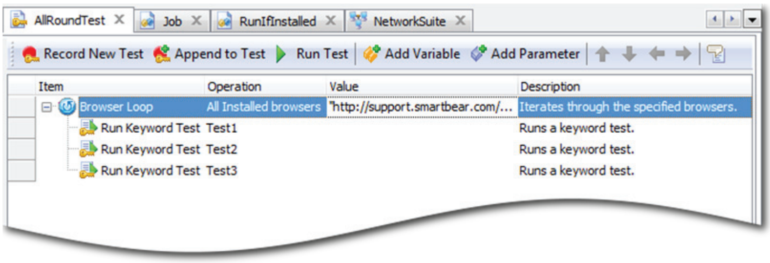
| Item | Operation | Value | Description |
|---|---|---|---|
| Run Browser | Parameterized | "http://smartbear.com/", ... | Launches the specified brows... |

**Browser Selection Parameters**
Specify the browser family, version and platform.

| # | Name | Value |
|---|---|---|
| 1 | Browser | Internet Explorer |
| 2 | Version | 9 |
| 3 | Platform | pX64 |

## Running Tests in All Browsers

To run a test in all browsers installed on a computer, you can create the main "launcher" test that would loop through all installed browsers and run the tests against each browser in turn.



| Item | Operation | Value | Description |
|---|---|---|---|
| Browser Loop | All Installed browsers | "http://support.smartbear.com/... | Iterates through the specified browsers. |
| Run Keyword Test | Test1 | | Runs a keyword test. |
| Run Keyword Test | Test2 | | Runs a keyword test. |
| Run Keyword Test | Test3 | | Runs a keyword test. |

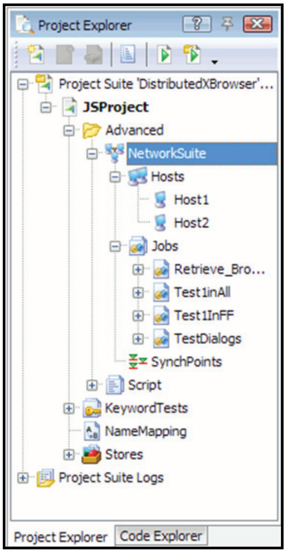
## 3. Understanding the NetworkSuite Project Item

To make a TestComplete project a distributed testing project, you need to add the NetworkSuite item to the project. This item must be added in each project participating in the distributed test – whether master (controlling) or slave (driven).

The NetworkSuite item specifies the distributed testing settings, contains a list of computers participating in the distributed test and the tests run on these computers.

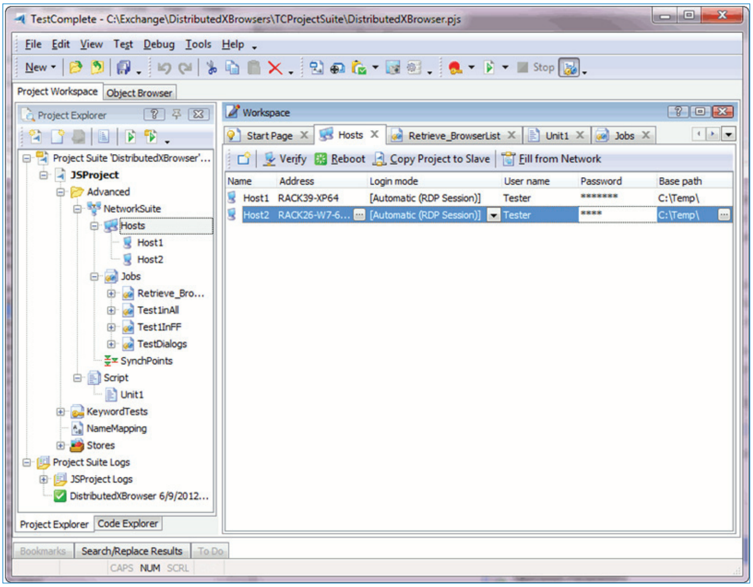
The NetworkSuite project item contains the following items:

- **Hosts** – a list of remote computers to run tests on
- **Jobs** – a set of tests to run on the remote computers. The distributed test project runs jobs one after another; each job in its turn consists of one or more tasks (individual tests) that are run in parallel within the job
- **Synchpoints** – synchronization points for coordinating the test execution across different computers



## *4. Setting Up Connections to Remote Computers*

Remote computers (hosts) participating in distributed testing can be physical computers, virtual machines or cloud computers on your network. To manage the list of remote computers and configure connections to these computers, you use the Network Suite | Hosts item in the master test project.



| Name | Address | Login mode | User name | Password | Base path |
|---|---|---|---|---|---|
| Host1 | RACK39-XP64 | [Automatic (RDP Session)] | Tester | ******* | C:\Temp\ |
| Host2 | RACK26-W7-6... | [Automatic (RDP Session)] | Tester | **** | C:\Temp\ |

For each computer, you need to specify the following parameters:

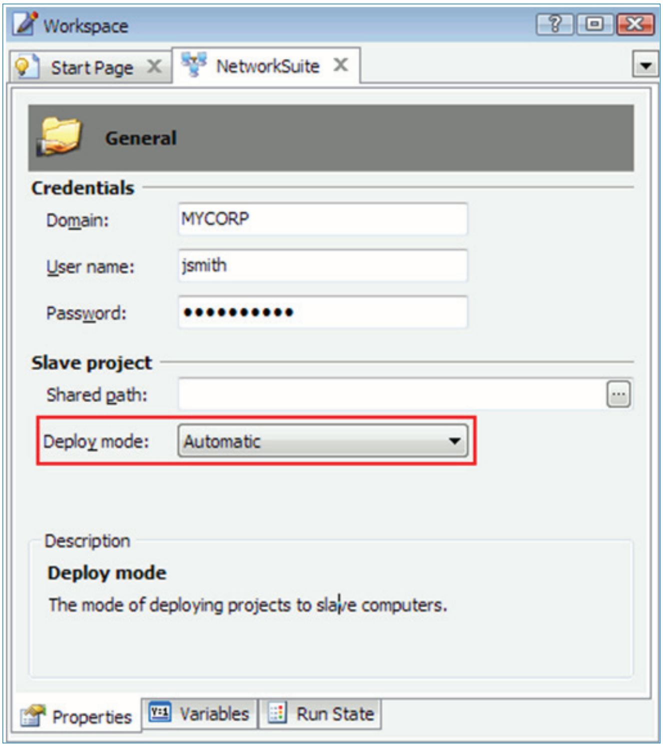
1. **Address** – the network name or IP address of the remote computer
2. **User name** and **Password** - the user account that TestComplete will be logging into the remote computer with
3. **Base path** - the folder on the remote computer where to automatically deploy the test project

To **verify** that the specified host parameters are valid, you can use the Verify button on the toolbar of the Hosts editor.

## *5. Deploying Test Project to Remote Computers*

TestComplete can automatically copy the test project from the master computer to remote computers. For this purpose, we need to configure the following options on the master computer:

- Specify the **Base Path** setting for each remote computer
- Set the Network Suite's **Deploy mode** setting to ***Automatic***

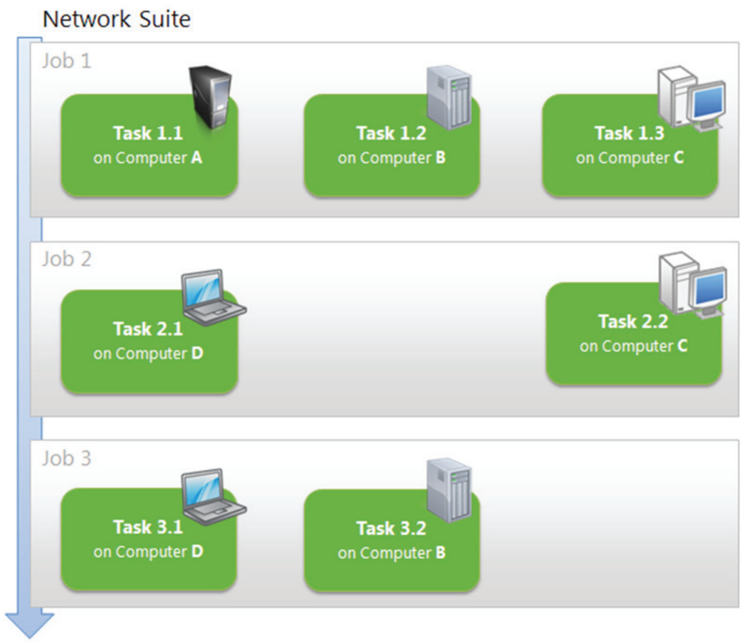


Credentials
Domain: MYCORP
User name: jsmith
Password: ••••••••••
Slave project
Shared path:
Deploy mode: Automatic
Description
**Deploy mode**
The mode of deploying projects to slave computers.

## *6. Specifying Tests to be Run on Remote Computers*

To specify the tests to be run on the remote computers, you use the **Network Suite | Jobs** project item. It specifies which tests will be run on which computers, their execution order and other settings.

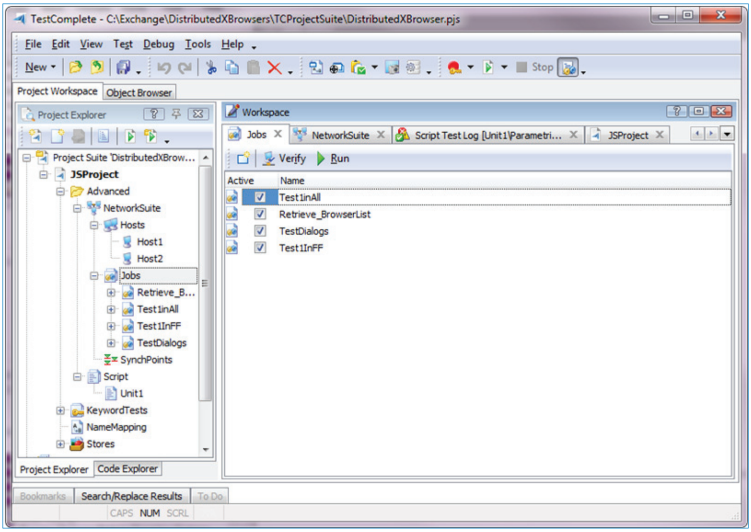
The tests (called **tasks**) are organized into groups – **jobs**. Tests within each job are run in parallel on different remote computers. Jobs, in their turn, are executed one after another. This way you can organize a complex testing process involving parallel and sequential test runs.



To specify the tests you want to run on remote computers, follow these steps:

1. **Add one or more Job to the NetworkSuite | Jobs collection.**

Each job will define a group of simultaneously executed remote tests. Since jobs are run one after another, their order in the editor is important.



| Active | Name |
|---|---|
| ✓ | Test1inAll |
| ✓ | Retrieve_BrowserList |
| ✓ | TestDialogs |
| ✓ | Test1InFF |

2. **Add tasks to each job.** A task specifies which test project (or project suite) should be run on which remote computer. In general, you need to create at least one task for each remote computer involved in the distributed test. Spacing?

Since all tasks of a job are executed in parallel, their order in the editor is not important.

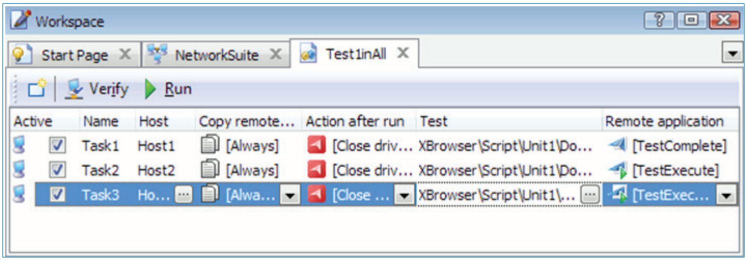
**3. Specify the task parameters:**

a. **Host** – The remote computer the where the task will be run.

b. **Test** - The test project (project suite or individual test) to be run on the remote computer. Required.

c. **Remote application** - Specifies what testing application, TestComplete or TestExecute, will be used on the remote workstation. Tests are equally played back in either of these applications. TestExecute is recommended as it consumes fewer resources.

d. **Copy remote log** - Specifies whether and in which cases Test Complete will copy the remote test log to the master computer.



| Active | Name | Host | Copy remote... | Action after run | Test | Remote application |
|---|---|---|---|---|---|---|
| ✓ | Task1 | Host1 | [Always] | [Close driv... | XBrowser\Script\Unit1\Do... | [TestComplete] |
| ✓ | Task2 | Host2 | [Always] | [Close driv... | XBrowser\Script\Unit1\Do... | [TestExecute] |
| ✓ | Task3 | Ho... | [Alwa... | [Close ... | XBrowser\Script\Unit1\... | [TestExec... |

## 7. Running Cross-Browser Tests on the Remote Computers

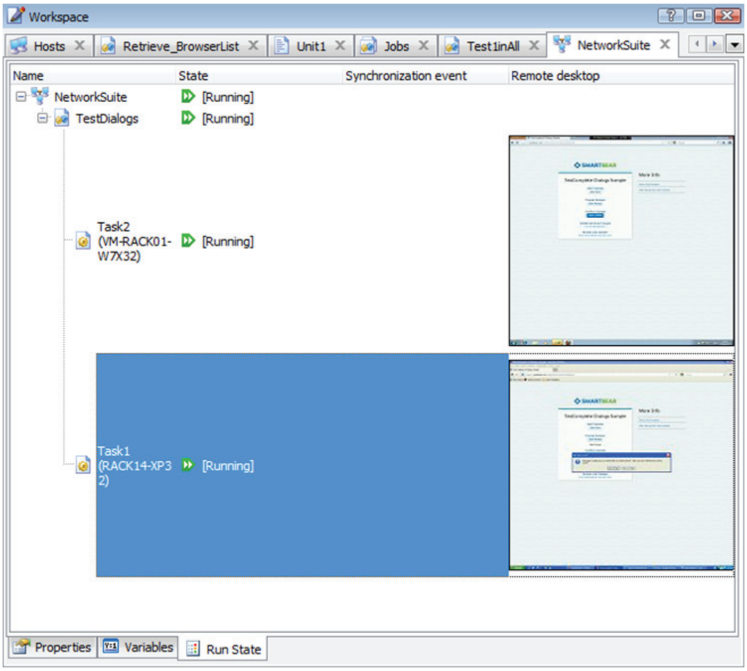
Once web tests are created, remote machines are prepared and the distributed test is configured, you can start the test run.

You initiate the distributed test run on the master computer. For this purpose, you need to open the test project in TestComplete, right-click the NetworkSuite project item in the Project Explorer and select Run from the context menu. Similarly, you can run individual tasks and jobs if needed (note that tasks inside a job are executed in parallel).

Before starting the test, TestComplete performs the following preliminary operations to ensure that the test can be executed successfully:

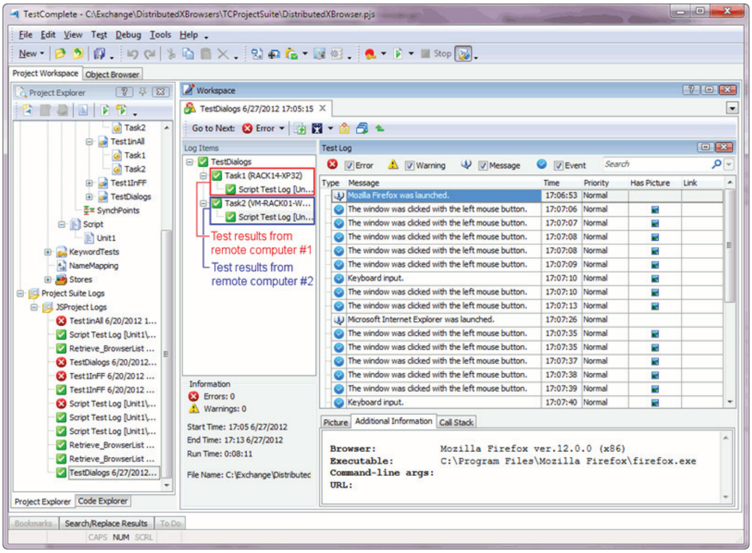
1. Verifies that the remote computers are accessible
2. Copies the test project to remote computers (if auto-deployment is configured)
3. Verifies that the needed tasks can be started

While the distributed test is running, TestComplete lets you monitor the test execution in real time. It shows Remote Desktop views of the test computers and the current status of tests run on these computers.



| Name | State | Synchronization event | Remote desktop |
|---|---|---|---|
| NetworkSuite | [Running] | | |
| TestDialogs | [Running] | | |
| Task2 (VM-RACK01-W7X32) | [Running] | | |
| Task1 (RACK14-XP32) | [Running] | | |

After the distributed test is finished, TestComplete can retrieve the test logs from the remote computers and combine them into a single accumulated log on the master computer. This behavior is determined by the Copy remote log property of each task in the test. By default, the remote test logs are always copied to the master computer, so that they can be viewed when remote computers are currently unavailable.


TestComplete - C:\Exchange\DistributedXBrowsers\TCProjectSuite\DistributedXBrowser.pjs
File Edit View Test Debug Tools Help
New Stop
Project Workspace Object Browser
Project Explorer
Task2
Test1inAll
Task1
Task2
Test1InFF
TestDialogs
SynchPoints
Script
Unit1
KeywordTests
NameMapping
Stores
Project Suite Logs
JSProject Logs
Test1inAll 6/20/2012 1...
Script Test Log [Unit1\...
Retrieve_BrowserList ...
TestDialogs 6/20/2012...
Test1InFF 6/20/2012 ...
Test1InFF 6/20/2012 ...
Script Test Log [Unit1\...
Script Test Log [Unit1\...
Script Test Log [Unit1\...
Retrieve_BrowserList ...
Retrieve_BrowserList ...
TestDialogs 6/27/2012...
Project Explorer Code Explorer
Workspace
TestDialogs 6/27/2012 17:05:15
Go to Next: Error
Log Items
TestDialogs
Task1 (RACK14-XP32)
Script Test Log [Un...
Task2 (VM-RACK01-W...
Script Test Log [Un...
Test results from remote computer #1
Test results from remote computer #2
Test Log
Error Warning Message Event Search
Picture Additional Information Call Stack
Bookmarks Search/Replace Results To Do
CAPS NUM SCRL

Information
Errors: 0
Warnings: 0

Start Time: 17:05 6/27/2012
End Time: 17:13 6/27/2012
Run Time: 0:08:11

File Name: C:\Exchange\Distributec

| Type | Message | Time | Priority | Has Picture | Link |
|---|---|---|---|---|---|
|  | Mozilla Firefox was launched. | 17:06:53 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:06 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:07 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:08 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:08 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:09 | Normal |  |  |
|  | Keyboard input. | 17:07:10 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:10 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:13 | Normal |  |  |
|  | Microsoft Internet Explorer was launched. | 17:07:26 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:35 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:35 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:37 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:38 | Normal |  |  |
|  | The window was clicked with the left mouse button. | 17:07:39 | Normal |  |  |
|  | Keyboard input. | 17:07:40 | Normal |  |  |

```
Browser:             Mozilla Firefox ver.12.0.0 (x86)
Executable:          C:\Program Files\Mozilla Firefox\firefox.exe
Command-line args:
URL:
```

## Conclusion

With TestComplete, you can create automated cross-browser tests and distribute the tests across multiple physical, virtual and cloud computers. Distributed testing with TestComplete includes automated deployment of test projects to remote computers, flexible configuration of the testing process and centralized monitoring of distributed test execution. To run the tests on the remote computers you can use either TestComplete, or the TestExecute solution that was designed as "test runner" for machines without TestComplete installed.

In this article, we provided an overview of distributed cross-browser testing using TestComplete and outlined the main steps of this process. To start automating your cross-browser and distributed tests today, download now and try TestComplete for free.

## About SmartBear Software

More than one million developers, testers and operations professionals use SmartBear tools to ensure the quality and performance of their desktop, mobile, Web and cloud-based applications. SmartBear products are easy to use and deploy, are affordable and available for trial at the website. Learn more about SmartBear, the company's award-winning tools or join the active user community at http://www.smartbear.com, on Facebook or follow us on Twitter @smartbear and Google+.