Test Focus: TOC formats: modern

[1. Introduction 2](#_Toc216171188)

[1) System Requirements 2](#_Toc216171189)

[I. Hardware Requirements 2](#_Toc216171190)

[1) 128-500MB of RAM 2](#_Toc216171191)

[2) Pentium II or faster processor 2](#_Toc216171192)

[3) 50MB of hard drive space 2](#_Toc216171193)

[4) Central storage unit recommended for test script storage 3](#_Toc216171194)

[5) Platform Support 3](#_Toc216171195)

[6) Browser Support 8](#_Toc216171196)

[2) Price and Licensing 9](#_Toc216171197)

[2. General Challenges 10](#_Toc216171198)

[3. Technical Details 12](#_Toc216171199)

# Introduction

## System Requirements

### Hardware Requirements

#### 128-500MB of RAM

#### Pentium II or faster processor

#### 50MB of hard drive space

#### Central storage unit recommended for test script storage

Typical hardware requirements. Lots of memory is crucial. Typically XDE Tester 2.0 will be part of a large testing suite. Although it will likely be possible to pick and choose the test tools desired, there may still some overhead required for memory and storage. For example, you may have to install a license configuration manager. Additional requirements such as these may require the implementation of client and server software on separate machines. This may be an issue for ECD TVT because of the requirements of WebSphere Commerce Server.

The alpha release did not have any extravagant hardware requirements.

#### Platform Support

##### Windows NT 4.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

###### NT 1.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

###### NT 2.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

###### NT 3.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

###### NT 4.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

###### NT 5.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

###### NT 6.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

##### Windows 2000

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

##### Windows XP

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

##### Windows Vista

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

##### Red Hat 6.2 or later

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

#### Browser Support

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

##### Microsoft Internet Explorer 5.0-6.0

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

##### Netscape Navigator 7 or later

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

## Price and Licensing

XDE Tester 2.0 is freely distributed internally within IBM. As such, there are no current or foreseeable licensing issues that need to be addressed at this time.

# General Challenges

***Ease of Installation and Setup***

The installation and setup of the tool was very easy. The alpha version was provided in a standalone package. There were no setup patches required for the environment that the evaluation occurred in. The installation was simple and there were no workarounds to be followed. For the edition evaluated, there was no license configuration manager to install or configure. This will likely be different in the final release.

The tester must also create a datastore for which the test data will be stored when scripts are created. Datastores, if shared, must be manually imported into XDE Tester 2.0 in order for test scripts to function.

Before using XDE Tester 2.0, the Java enablement environment must be setup. Java environments must be enabled in order begin testing Java applications or brower-based applications. XDE Tester 2.0 will automatically enable sample JREs to help speed up the configuration process.

XDE Tester 2.0 will not automatically enable tests to run in your browser. The process to setup this very intuitive and the process only takes a few minutes.

***Ease of Test Case Execution and Evaluation***

The procedure to follow to record and execute test cases appear fairly straightforward. Once the test application is enabled (i.e. for web-based applications, the test application is the browser), than we simply need to follow simple directives from the client tool. According to the help file, there is support for executing test cases from the command line. Thus, executing a test case can be integrated into the test environment without a great of effort.

With regard to evaluation, logs are generated that keep track of failures and proper defect analysis can be conducted from test execution results.

For the purposes of TVT automation, test case execution can be challenging. At the moment, for double byte languages, QTP must be installed natively on the machine with the language being tested. This can be a time consuming process especially with the number of workarounds and patches that are required to be installed. This may or may not be avoidable and we are looking into possible third-party support to create an environment that would allow QTP to run test scripts against WCS in any language.

.

The generated log files are of excellent value. With the logs, the experienced user can easily identify the cause to any test script failure and determine if it’s a defect and to what jurisdiction the defect belongs.

***Vendor Technical Support***

Rational Tools has always exhibited excellent support. Because Rational Tools is now part of IBM, free internal support is easily obtainable. There is an online forum available for technical questions. Responses and follow-ups are conducted very quickly, although sometimes the support does not properly address the problem. Often questions were passed along the chains of support several times before a workable solution was provided.

***Learning Curve***

XDE Tester 2.0 uses the Java for the test scripts created. Thus, a good solid knowledge of Java is required to take advantage of the full potential of the software. A good reference manual is highly recommended. With IBM’s tremendous support for Java, XDE Tester 2.0 Java environment should not be greatly intimidating. It can be a comforting to the Java user.

To assist the new user, XDE Tester 2.0 comes with some quick tutorials.

The included help files do not follow the standard Windows help file format. Using the help files proved to be a challenge. Searching for information on supported methods proved to be a challenge. There is no index for the help file although a search option is provided. When attempting to search for a particular method, I received only non-relevant search results when looking for information regarding the .click() method. Further, there appears to be an absence of example source code snipits in the help system. Having a working code snippet corresponding to a particular function would be a great option.

Lastly, when search for internalization, globalization, and language support for this evaluation, there were no relevant search results.

***Backward and Forward Compatibility***

Rational Tool claims that XDE Tester 2.0 is fully backward compatible with Robot and RobotJ test scripts and datastores. There was no explicit supported noted in supporting use of XDE Tester 2.0 test scripts or datastores with either Robot or RobotJ. Backward and forward compatibility issues were not examined in detail for the evaluation because Robot and RobotJ test scripts or datastores were not available.

# Technical Details

***Support for Your Application Type***

XDE Tester 2.0 fully supports Java script and Java applets which are crucial for web-based applications. The support for Java applets are an integral part of the application. In preliminary tests, XDE Tester 2.0 was able to correctly record and play back interaction with Java applets with no additional setup or configuration. This is important because there are often used in web based applications such as WebSphere Commerce Server (WCS) (i.e. Java applets used for Customer Care).

XDE Tester 2.0 fully supports Java scripts. XDE Tester 2.0 will see the web page content as it is displayed to the user. On some pages, the html content is dynamic and by viewing page source cannot reveal the current html that is actually generated/displayed to the viewer. However, QTP does see the html content as it is displayed and, therefore, will manipulate the web content as if it were not dynamically generated.

XDE Tester 2.0, like Quick Test Professional, interprets the content of the browser as it is exhibited to the user. Thus, dynamic web pages appeared to work correctly under XDE Tester 2.0. XDE Tester 2.0 correctly manipulates objects as they appear to the user. Further, html dynamically generated by Java scripts proved not to be a problem.

Legacy Java script and Java applets that are used for components such as selection trees and Customer Care required no test script alterations that other tools such as QTP required (recall, for QTP, some common functions needed to be introduced ensure proper execution of test scripts on these components).

***Support for Single Byte Languages***

XDE Tester 2.0 is fully complient with single byte encoding of text. Test scripts were correctly generated while recording under English, Spanish, Portuguese, Italian and German. The IDE saved the code syntax as it appeared on the machine at the time of the recording. Objects appeared in XDE Tester 2.0 as they appeared in the test application without an internal or visual data corruption. Test scripts programmed with a particular single-byte language can be edited and played back in any other language.

***Support for Double Byte Languages***

XDE Tester 2.0 is not fully compatible with double byte languages. During the evaluation, I attempted to record a Japanese test script using an English workstation with the correct the correct language support for Japanese installed natively on the operating system. XDE Tester 2.0 happily recorded data from the Unicode application (WCS) but refused to play back the test scripts because the double byte data that was recorded was incorrectly noted by a series of questions marks (i.e. the XDE Tester 2.0 trying to make reference to object\_???.click which is not of proper form). When play back was attempted, the XDE Tester 2.0 reported compilation errors with the script.

Test scripts that were recorded on an operating system that were recorded with full support of the native language (i.e. recording Japanese test scripts on top of machine with the Japanese operating system) were successful when played back on the native machine. When the test scripts were loaded on a language that did not natively support the language suffered the symptoms discussed earlier.

The above difficulty with Japanese may or may not be reflective of general double byte problems experienced with XDE Tester 2.0. Because Java natively supports Unicode, many of these problems encountered may be resolved by the final release of the product or by some adjustment additional workaround in installation and/or setup.

***Vulnerability to Change***

XDE Tester 2.0 test scripts are very vulnerable to change. Change in translation, ordering of elements or index, and other functional changes immediately results in test scripts that hung or crashed. The only solution that worked was to rerecord the object or operation.

For general changes, such as indexing, I was unable to discover how to change an object’s properties after record time.

With some changes, the tester should be on a lookout for changes in UI or functionality that do not necessarily break the automation but may change the behaviour or outcome of the test case. Some changes can be apparent while others are not.

Some future challenges can b avoided with proper preparation. One of the biggest problems that plagued the TVT automation for WCS is handling program menus. In functional testing, test scripts navigate java script menus by the display name of each menu item. For TVT, because the menu items do not have name tags or any other unique identifier associated with them, indexing is the only way to handle menu navigation. After all the scripts were recorded with indices recorded in the object repository, a change in the ordering (i.e. addition or removal of a menu item) will affect every subsequent menu item’s index. The ordering of the menu varies on the user access role and store model, amongst other properties. With WCS, a massive change such as the above scenario occurred just recently when the Commerce Enhancement Pack was integrated with the next release of WCS. This resulted in re-indexing over 60 menu items for each category of test script. The end result is manual configuration of thousands of menu navigations. Because the indices are stored it the object repository, there is no way to automate the change. The average time to navigate the GUI to make a single index change is 10 seconds.

Problems such as this can be avoided with clever thinking. For high risk objects, such as navigation menus, a generic library function can be created that house the indices for the navigation menu with respect to the user role, store model and build being tested. Now, instead of storing indices in the test script itself, when the test script needs to navigate a menu, it simply makes passes a logical name for the menu item, that both parties agreed to, to the function. The function will return the correct index value of the menu item. The test script can then use the index to navigate the menu.

With the above system, we can create robust scripts that can handle multiple roles and multiple store types. Most importantly, a change in the indexing of the menu can be implemented by directly editing the common function. By editing the function itself, which doesn’t make use of the object repository, we need only make the change once, and the change itself is easier and faster to implement.

**Enabling TVT Automation**

***Catch-22***

1. ***Write Once and Run on All Languages***

By design, XDE Tester 2.0 does not support the recording of a single test script that can be run on all languages. The XDE Tester 2.0 can create a common test script that to run all languages, however, the test script will be comprised of a logical test of the native language at each step. For example, if I want my test script to activate a button, I would have a if-elseif conditions that tested the language. So, if I test ten languages, I will program every action at least ten times. This is necessary because the Java used in XDE Tester 2.0 is language specific. The display name is embedded into the object’s name, and since display names are language specific, we require this procedure to follow.

1. ***Verify Translation***

Because of the method to which XDE Tester 2.0 is not necessarily setup for TVT automation, any changes to the language will affect the functional execution of test scripts. Language and functionality are a common entity in XDE scripts developed by TVT. In a TVT environment, XDE scripts would misbehave due to change in language or functionality.

We create verification points to track changes in translation. When we create a verification points, we are telling XDE Tester 2.0 to look for a particular language-specific component. If it differs or is missing in an execution, then XDE Tester 2.0 records the test case as a failure. Therefore, any slight change in translation will be caught. But, in the same process, any normal formatting changes in the html document and any functional changes (i.e. arrangement of onscreen content) will be detected. It will ultimately be up to the tester to determine if the detected change is a translation change. Once investigated, the test script’s expected output can be manually adjusted by editing the Java code. There is no single automated procedure or update run in XDE Tester 2.0. With some third party support, we can reduce the effort required to detect deviations.

***Additional Features***

The link between XDE Tester 2.0 and the browser is seamless. There were no problems encountered. One of the advantages of XDE Tester 2.0 over other client tools is the ability to continue to use other Internet Explorer instances while executing tests. XDE Tester 2.0 will open an independent Internet Explorer instance that are not tied to other windows.

There also appeared to be no corruption to Internet Explorer after installation of the client tool. XDE Tester 2.0 can accomplish it’s task without replacing crucial files that belong to Internet Explorer. Thus, we avoid possible conflicts with different applications using Internet Explorer.

***Known Problems, Disadvantages, and Improvement Recommendations***

XDE Tester 2.0 is fairly productive. There are some tasks in XDE Tester 2.0 that become routine. It takes some time and knowledge to get a successful and robust script working. To save time and resources, XDE Tester 2.0 can share classes and methods among test scripts. With this functionality, you can create a reusable class once and make calls to it from many scripts.