

---

**IBM Rational<sup>®</sup> XDE<sup>™</sup>  
Tester v2003**

*Evaluation Guide*

# Table of Contents

- Preface .....3**
- Introducing Rational XDE Tester .....3**
  - The Challenges of Regression Testing .....3
  - Is Rational XDE Tester the tool for me?.....4
- Getting Started with Rational XDE Tester .....6**
  - Install the Rational XDE Tester Evaluation Software .....6
  - Limitations on the Evaluation Version of Rational XDE Tester .....7
  - Install and Configure the Java Runtime Environment .....7
  - Create an Rational XDE Tester Datastore .....8
- Using IBM Rational XDE Tester .....9**
  - About the Sample Application .....9
  - Step One: Script Creation .....9
  - Step Two: Script Validation .....13
  - Step Three: Examining the Object Map .....16
  - Step Four: Script Execution .....19
- Next Steps.....21**
- Rational Developer Network .....23**
- Summary.....24**
- Technical Support Contact Information.....25**

## Preface

---

Welcome to the IBM Rational® XDE Tester v2003 Evaluation Guide.

This guide is designed to introduce the first time user, to IBM Rational XDE Tester. The guide will help you create, execute and edit your first script Rational XDE Tester regression testing script. Along the way we'll explore the challenges of regression testing and the solutions offered by Rational XDE Tester to address these challenges.

## Introducing Rational XDE Tester

---

Rational XDE Tester is an automated regression testing tool that lets users test Java and HTML applications running on Windows and Linux platforms. Rational XDE Tester scripts are created by recording a user's actions against a Java or HTML application. The scripts are coded in Java, and presented to the user in the Rational XDE Tester perspective of the Eclipse editor. The Eclipse editor is an open source development environment which offers powerful test code development and editing features. Once any optional code modifications have been completed, users can execute their Rational XDE Tester scripts against the current and future builds of their application.

## The Challenges of Regression Testing

Eventually, in any software system, users will uncover defects or bugs in the program. Typically, these bugs are fixed, the fixes are tested, and the updated software is released back to the user community. However, due to the interconnected nature of software, even the smallest change can wreak unpredictable havoc when implications of the change are not properly understood. Any software change, even for the purpose of correcting a known defect, can impact the system in an unforeseen manner and potentially cause greater problems for the user than the issue the change was originally trying to address.

Regression testing is the practice of retesting of a software system that has been modified to ensure that no previously-working functions have failed as a result of defect reparations or newly added functionality. Comprehensive regression testing fully ensures that a software system is functioning as designed. Comprehensive regression testing, however, is rarely feasible given the time and resource constraints placed on the software development team. As a software system expands and evolves it becomes more and more difficult to test every piece of functionality. This problem is

compounded by the frequency of software builds. In an environment where software builds are done on a nightly basis, comprehensive regression testing of every build is essentially impossible. Typically in these environments, the testing of previous functionality is forgone to allow for testing of new fixes and new functionality. This leaves the software team vulnerable to releasing software with undiscovered defects.

Rational XDE Tester is an automated regression testing tool which addresses these challenges. With Rational XDE Tester, test scripts are created for the purpose of re-testing system functionality. With every build of a new system, the scripts are executed to validate previously working functionality. Scripts are generally executed in an unattended mode, freeing up the quality assurance team to test the most recent fixes and functionality. As the software expands and evolves, so does the library of test scripts ensuring that every build of the application is adequately tested from beginning to end.

## Is Rational XDE Tester the tool for me?

Ask yourself the following two questions to determine whether or not Rational XDE Tester is the tool for you:

- Am I testing a Java or Web/Browser based application?
- Is there some familiarity with, or eagerness to learn Java on your test team?

If you answered **Yes** to both of the above questions, then Rational XDE Tester could be the tool for you! Rational XDE Tester was designed to test Java and Web/Browser based applications exclusively. If you are testing Visual Basic, C++ or other Windows 32 bit applications, Rational XDE Tester is not the tool for you. For users operating in these environments, IBM Rational offers Rational Robot. Similar in concept to Rational XDE Tester, Rational Robot enables automated regression testing of Windows based applications.

Rational XDE Tester scripts are coded in Java. To be successful, you'll need to have some familiarity with Java on your test team. Not everyone on the team needs to be a Java guru, but there has to be some knowledge on the team with basic Java skills to edit the scripts as necessary. Remember, test automation is really creating a program to test a program. Some Java coding will be necessary along the way. Rational XDE Tester is a powerful tool, but it can't do everything for you!

Now, consider these questions:

- Are you forced to forgo some tests due to time constraints?
- Are you spending too much time on test script maintenance?
- Are you testing a complex application that is hostile to record and playback?

If you answered **Yes** to any of the above then you need to take a good look at what Rational XDE Tester can do for you. Continue on through this evaluation guide and explore the tool. In about an hour, you'll go through the process of recording, editing, executing and analyzing your first Rational XDE Tester script. Once complete, you should be in a good position to understand the value that Rational XDE Tester can bring to your software development process.

## Getting Started with Rational XDE Tester

---

Getting started with Rational XDE Tester involves three steps:

1. Install the Rational XDE Tester software
2. Install and/or Configure the Java Runtime Environment
3. Create a Test Datastore

Let's walk through these three steps now.

### Install the Rational XDE Tester Evaluation Software

#### Remember Your License Key

In order to run the evaluation version of Rational XDE Tester, you need a license key. This key should have been emailed to you automatically when you completed the evaluation request form before downloading the software.

Complete the following steps to install the Rational XDE Tester software:

1. Download the self extracting zip file, and double click to extract the files and start the installation. You should have received the download location via e-mail.
2. Click on the "Evaluation Installation" link on the application page. At the security warning select "Open" to begin the installation.
3. During the Rational XDE Tester installation, you will be prompted for two decisions during the install.
  - a. You will be first prompted for an install location. Any location will do.
  - b. You will be then prompted for the Rational XDE Tester Shell Instance. The Rational XDE Tester evaluation is limited to the installation of a new instance of the WebSphere Studio Workbench 2.0. Installation into WebSphere Studio Application Developer 5.0 is only supported in the full version of Rational XDE Tester.
4. After the install completes, reboot your computer as directed.
5. Once the reboot is complete, the License Key Administrator Wizard will be launched for you. If it does not launch, start it from Start→Programs→Rational Software→Rational License Key Administrator.
6. From the Wizard, select "Import a Rational License File", click Next and browse to the location where you saved the license file that you received via e-mail.
7. Click on Import to import the license. Once imported, you may shut down the License Key Administrator. You're ready to [configure your environment](#).

## Limitations on the Evaluation Version of Rational XDE Tester

The evaluation version of Rational XDE Tester is a stand alone version of the product. The integrations to other IBM Rational products such as Rational TestManager and Rational ClearCase have been disabled in this version. This functionality is only available in the full version of Rational XDE Tester.

## Install and Configure the Java Runtime Environment

As you'll soon experience, Rational XDE Tester creates test logs to inform you of what has transpired during script execution. By default, these logs are HTML documents. However to provide additional information, Rational XDE Tester will use Java comparator applications to highlight differences between expected and actual results. In order for Rational XDE Tester to run these Java applications from the browser where the HTML logs are displayed, the Java Runtime Environment must be installed and configured.

### Is the Java Runtime Environment already installed on my machine?

It's quite possible that you have a Java Runtime Environment (JRE) on your machine. If you do, you'll find a Java configuration icon in your control panel. Select Start→Settings→Control Panel. If, in the list of applications, you see a Java Plug-in option, then the JRE is installed on your machine. Complete the steps in [Configuring your Java Runtime Environment](#). If you do not see this icon, move forward to the next section, Installing the Java Runtime Environment.

## Installing the Java Runtime Environment

The Java Runtime Environment is available for download from <http://java.sun.com/j2se/1.4.1/download.html>. Select the appropriate link for your environment. Most users will want to select the Windows (US English Only) JRE link. Once you have downloaded and installed the software, you can skip ahead to step 7 of the Configuring your Java Runtime Environment section.

## Configuring your Java Runtime Environment

1. In the Control Panel, double-click the Java Plug-in icon to open the Java Plug-in Control Panel.
2. If you have more than one plug-in listed, use one that is version 1.3.1\_03 or later.
3. In the Java Plug-in Control Panel, click the Browser tab.
4. Select Microsoft Internet Explorer and/or Netscape. (If you have both browsers on your system, select them both.)
5. Click Apply.
6. Close Control Panel.
7. Launch Rational XDE Tester.

8. In Rational XDE Tester, click Configure → Enable Environments for Testing to open the Enable Environments dialog box.
9. Click the Java Environments tab.
10. Click the Select All button to select all current Java environments.
11. Click Disable to disable them.
12. Click Search. The Search for Java Environments dialog opens. Select one of the following search mechanisms.
  - a. Quick Search can only be used on Windows systems. It searches the registry for the Java environments, and is quicker than searching your hard drive(s).
  - b. Search All Drives scans all of your hard drives or partitions to locate all the Java environments on your system.

Note: You should not use the Search All Drives option to find JREs on Linux or UNIX systems. Instead use the Search In option and browse for the JRE.
  - c. Select Search In to browse to a specific drive or root directory to search.
13. After choosing one of the search mechanisms, click the Search button.
14. When the search is complete, it will list the JREs in the Java Environments list. The list includes the full path name of each environment.
15. Click the Select All button under the list to enable all of them.
16. Click Enable.
17. The selected environments will be enabled for Java testing. The enabled environments will be indicated in parentheses after each JRE name in the list.

## Create an Rational XDE Tester Datastore

An Rational XDE Tester Datastore is the mechanism Rational XDE Tester will use to store your test scripts, expected results and test logs. To create your first Rational XDE Tester Datastore, complete the following steps:

1. Select Datastore → Create Datastore...
2. Provide a descriptive name for your Datastore. Throughout this guide, we'll use the name "Eval Datastore"
3. You may select any location for this datastore to be located on your hard drive.
4. Click on Finish to complete the process.



## Using Rational XDE Tester

---

Let's get started. You're about to record, edit and playback your first Rational XDE Tester script. Along the way, you'll learn how the ScriptAssure™ feature will greatly simplify your dynamic data testing and make your scripts more resilient to application changes. You'll work within the flexible and extensible Eclipse shell, and leverage the power of Java as your scripting language.

### About the Sample Application

You'll be using the Classics Online sample application that is installed with Rational XDE Tester. The Classics Online application allows users to browse and place orders for music CDs. We have two versions of the application, Build A and Build B. Having two builds will allow us to perform true regression testing. We'll record tests against Build A and execute the tests against Build B.

### Step One: Script Creation

#### Re-use of recorded scripts.

When you record scripts against a given build of an application, you're investing in future testing efforts. As soon as the next build becomes available, you can play back your scripts and test the new build in a fraction of the time it would take to perform the tests manually.

In this Step, we'll record a script that places an order for a CD. Our script will validate both static and dynamic data. Once complete, we'll have a working test script, written in Java, that we can edit and execute.

#### 1. Start Recording a script


- a. Confirm that you are in the Test perspective within Rational XDE Tester by checking the main title bar. It should be entitled **Test – Rational XDE Tester**
- b. Select **File > New > XDE Tester Script Using Recorder**. The Record a New XDE Tester Script dialog box opens. Give your script the name "Order" and click on Finish.

#### Version Control

With the assistance of Rational ClearCase, scripts can be placed under Version Control. This manages the creation of scripts in a multi-user environment, and enables you to track of the evolution of your scripts.

## 2. Recording User Actions

As you begin testing your application, Rational XDE Tester will record your mouse clicks and keystrokes. As Rational XDE Tester is recording your actions, you can monitor the recording activity in the Recording Monitor window.

- a. After you have started the recording process and the Record toolbar is visible, you can use the Start Application button  to run your application.

### Configuring Applications

To test your application, you must first configure XDE Tester to launch the application. You can learn how to configure your own application in the [“Next Steps...”](#) section of this document.

- b. The Start Application button brings up the Start Application dialog. If you have already configured your application for testing with Rational XDE Tester, your application name will appear in the list box. For the purpose of this evaluation select the preconfigured ClassicsJavaA application.
- c. Place an order for Haydn’s Violin Concertos by completing the following steps:
  1. Click on the [+] next to Haydn to expand this composer
  2. Select Violin Concertos
  3. Click Place Order button

The Member Logon window will appear. Login as follows:

4. Select the default customer Trent Culpito
5. Enter the password: secret

### Important Step!

Make sure to click this checkbox - you'll see why when we get to playback time!

6. Click the Remember Password Box.
7. Click Ok.


The Place an Order window will appear. Complete the order as follows:

8. Enter the Credit Card Number: 1414 2525 3636 4747
9. Enter the Expiration Date: 12/05
10. Click Place Order to place the order.

### 3. Creating Verification Points in Scripts

A verification point is a place in a script where you perform a validation. Essentially you are capturing a baseline snapshot against which future builds will be compared.


- a. When your order is placed, a Message dialog will appear with an order confirmation message.

The Verification Point and Action Wizard enables you to record verification points. Click on the Verification Point and Action Wizard button  and drag it over the confirmation message in the dialog box. Release the mouse button when the message becomes encased in a red square.

- b. Click Next to show the data being captured
- c. Click Finish to capture this baseline data.
- d. Click OK on the Message dialog with the order confirmation.

### 4. Complete the recording

Continue on to view your order information, and then complete the recording.

- a. Select Order > View Existing Order Status. Click OK to continue.
- b. Create a verification point on the order information by dragging the Verification Point and Action Wizard over the data in the grid.
- c. In the Data Value drop down list box of the wizard, select "Table Contents".
- d. Click Next to show the data being captured
- e. Click Finish to capture this baseline data.
- f. Click on Cancel Selected Order.
- g. Click on Close
- h. Now close the application using the Close button in the top right corner of the application window.
- i. When you have finished recording user actions and verification points, you can stop recording by clicking the Stop button  on the User Action toolbar.

## Review

Congratulations! Using Rational XDE Tester's powerful recording mechanism, you have just created your first Java test script which can be used to test this build of the application as well as subsequent builds of the Classics application. By investing a little time now, you'll save lots of time later when this script can be run to validate the functionality of future builds.

Your script is written in Java. Using an industry standard language means that testers will always have a good base of resources upon which to draw when it comes to modify and extend the functionality of their scripts.

## Step Two: Script Validation

### Dynamic Data Validation

Dynamic Data validation involves ensuring that the system's output is correct even when the content of this output is not constant. Validating data such as times, dates, totals, sequential product or order numbers can present a significant challenge to testers. Typically, testing such ever changing data requires some coding effort – but not with Rational XDE Tester!

In this Step, we'll playback our test and modify it to handle the dynamic content. The order number in this message will be different every time an order is placed. We'll walk through the process of creating a dynamic data validation point.

#### 1. Running Scripts from Rational XDE Tester

- a. Rational XDE Tester test scripts can be run from the command line, from Rational TestManager, or directly from Rational XDE Tester. Run your test directly from Rational XDE Tester by selecting Script > Run.
- b. When you run an Rational XDE Tester script, a playback monitor will appear and will display all the actions you recorded in your script very quickly. Watch your script run. It should perform the exact actions you just recorded. When playback has completed, the test log will appear.

#### 2. View Rational XDE Tester Logs

By default your default browser will open and the results will be displayed in the HTML format. Rational XDE Tester can log results as Plain Text, HTML or in Rational TestManager log format.

- a. As you review the log file you will see your first verification point has failed during playback. Click on the "View Results" link to get additional detail.



- b. Why did this test fail? A new order number is generated each time an order is placed in the Classics application. As recorded, your test is designed to look for an exact match of the order number that was generated when the script was recorded. We created a static verification point to test dynamic data!

### Differences Not Bugs

One thing you'll soon learn about automated scripts is that they find discrepancies not defects. It's up to you as a tester to determine when a discrepancy is a defect.

## 3. Convert a Static Verification Point to Dynamic

Let's now modify the static verification point to handle the dynamic data.

- a. Click the Load Baseline to edit  button
- b. Click the Edit Text button  above the Baseline Value pane to open the Editor dialog.
- c. Click the regular expression button **RE:** to convert the static verification to a dynamic verification.
- d. Edit the text to delete *only* the order number and replace it with **[0-9]{2}**. This is a wild-card that will match any two digit order number between 00 and 99.

### Regular Expressions

The notation of this wild card is called Regular Expression Notation. This is a standard pattern matching notation commonly used in PERL and many other programming languages.

- e. To ensure that your regular expression is working correctly, click the Expression Evaluator button **EE:**.
- f. Click Evaluate. If "Match" appears in green than you're conversion has worked. If not, ensure you've entered the expression exactly as noted above. Make sure your spacing is correct.
- g. Click OK to finish.
- h. Close the Edit dialog.
- i. Click File→Save to save your edits
- j. Close the Verification Point Editor
- k. Run the script again and ensure that your verification point passes

## Review

Dynamic Data validation is a common testing challenge. Using Rational XDE Tester's powerful pattern matching feature, you can easily create tests to validate dynamic data without writing a single line of code.

## Step Three: Examining the Object Map

### The Object Map: The Key To Resilient Playback

You may have noticed that once recording was complete that Rational XDE Tester generated not only a test script, but a list of objects as well. Collectively, these objects are contained in the Rational XDE Tester Object Map. This Object Map contains information about all of the objects in your application which the scripts use to execute the playback commands. In this Step we'll explain the Object Map's role in making your scripts resilient to application changes.

In this Step, we'll open up the object map and examine the information that Rational XDE Tester has captures about all of your application's GUI objects.

#### 1. Examining the object map

The object map contains information about every GUI object you acted against in your script.

- a. In the Rational XDE Tester Script Explorer, expand the Test Objects folder if it is not already.

#### Private vs. Shared

Object Maps can be either shared or private. A private map is attached to a script, and only accessed by that specific script. A shared map is a map that is shared by multiple scripts. The advantage of a shared map comes when objects need to be updated. Using a shared map means only one update to one map to fix all your scripts.

- b. Double click on the **text** object. This is the field where you typed in a credit card number. It should be near the bottom of the list.
- c. Examine the information presented on the Recognition tab.



Recognition Administrative		
Property	Value	Weight
.class	javax.swing.JTextField	100
.classIndex	3	50
.priorLabel	Card Number (include the spaces):	25
accessibleContext.accessibleRole	text	100
name	.cardNumberField	100

You'll see that Rational XDE Tester has captured five properties of this object, .class, .classIndex, .priorLabel, accessibleContext.accessibleRole, name. Collectively, Rational XDE Tester will use all of these properties to find and act upon the credit card field.

Note that each property is weighted. These weights determine the relative importance of each property. As an example, an object name change (weight 100) is four times more significant than a prior label change (weight 25). Users can adjust these weights for greater flexibility.

When Rational XDE Tester goes to act upon this object at playback time, should it not find an exact match, it will use a scoring system to find the closest match between the object it's looking for and the objects in the map. Rational XDE Tester will determine a score for all the objects in the map by comparing the properties of the target object and the objects in the map. A perfect match scores zero, while small variations – such as a prior label change – result in small scores. Rational XDE Tester determines the smallest score, and if it's within an acceptable variation tolerance, will act upon the object. This means that small variations in your application's GUI will not break your scripts! This feature will save hours of script maintenance and rework.

Strict. This setting allows the user to control the amount of acceptable variation between object definitions giving the user better control in determining when a script playback will be halted due to differences between the original and current application.

- d. Close the object map.

### Review

Rational XDE Tester uses an Object Map to contain information about all the objects acted upon in a script. The object may be specific to a single script or shared by many scripts. Sharing an object map between scripts is a good strategy to reduce script maintenance.

At playback time, Rational XDE Tester compares the objects on the GUI to the objects in the map. Using a weighted scoring mechanism allows Rational XDE Tester to playback against an application whose GUI may have changed between recording and playback time. This feature makes Rational XDE Tester scripts resilient to GUI changes and reduces the time spent performing script maintenance.

## Step Four: Script Execution

### Script Assure Reduces Script Maintenance

You've arrived. You've created your script and you now about to reap the benefits by automatically testing a new build of your application. From this point on your biggest enemy will be script maintenance. Your goal is to minimize the amount of time you spend reworking old scripts, and maximize time spent creating new scripts. In this Step, you'll learn how Rational XDE Tester's ScriptAssure feature helps you minimize script maintenance.

In this step, we'll execute your script to validate that the latest build of the software. Your script will ensure that changes made to the application have not adversely impacted the process of placing an order.

#### 1. Playing back a script against the new build

Before you can play back the script against the new build, you need to edit the application name, so that playback executes against Build B of Classics Online.

- a. In the script window, make sure your script (Classics.java) is the active script. At the top of the script, underneath the template information, you will see the start application command. It should appear as follows:  

```
startApp("ClassicsJavaA");
```
- b. Change the ending "A" to a "B". It is case-sensitive, so make sure you capitalize the B.
- c. Run the script by selecting Script > Run.

### GUI Redesign

Notice when the script executes that the layout of the application has changed significantly. These changes do not impact XDE Tester's ability to playback. Thanks to the Object Map, XDE Tester does not rely on the geographic placement of controls. GUI reorganization will not impact script playback!

- d. During playback the script will pause for about 20 seconds at the Login window while it attempts to find the Remember Password checkbox. This new build contains a label and object name change on the Remember password object. The label now reads “Remember **the** Password” instead of “Remember Password”.

Thanks to ScriptAssure, this object change will not prevent script playback. ScriptAssure’s fuzzy logic will determine that the object is a close enough match to the original, and continued with script execution.

- e. Wait for the script to complete and the test log to appear.

You’ll note in the log that Rational XDE Tester has raised a warning that reads “Object Recognition is weak”. Here Rational XDE Tester is informing you that an object has changed between the original recording and the current application.

It’s your choice now whether or not to ignore the problem or update the script. Ignoring the problem will not impede playback, as you’ve just seen. However, you will continue to get warnings. Updating the map will put the script in sync with the application and remove the warnings. For the adventurous, this process is documented in the Online Rational XDE Tester Quick Tour, section 20.

This is an example of the power of Rational XDE Tester’s ScriptAssure feature. In this case the visible name and underlying logical name of the “remember password” checkbox. ScriptAssure used its fuzzy logic matching to help identify and adapt to changes in the application to enable script execution to complete.

With any other automated testing tool, this type of change would have halted test script playback and required you to manually update the object map or script. This type of scripting error can be particularly time consuming, especially when this script is part of an overnight test run. One small script error can waste an entire night of testing!

### Review

Rational XDE Tester has the unique ability to easily “adapt” to application changes that are identified during playback to enable a script to continue execution. This drastically minimizes ongoing script maintenance which can be a significant drain on the test team’s resources.

---

## Next Steps...

---

Let's examine what you've just accomplished:

- You have created your first Rational XDE Tester recording
- You have used created a test that validates dynamic data
- You have had ScriptAssure ensure test playback in the face of an application change

What's next?

You've tried Rational XDE Tester against our demo application, now it's time to test against your own application. To get started, your first step will be to configure your application.

### To configure your application follow these steps:

1. Select Configure > Configure Applications for Testing...
2. Select Add...
3. You have three choices at this point. Select:
  - a. Java Application if you application is launched from a java .class or .jar file
  - b. HTML Application if you are testing a browser based application
  - c. Executable or Batch File if your java application is launched from a batch or executable file
4. Click on the Run button to test your configuration.

If your application launches, then you have successfully configured your application. You may begin to start recording your first tests against your application. If your application does not launch, revisit the information in the Detailed Information fields to ensure that all entries are valid and correct.

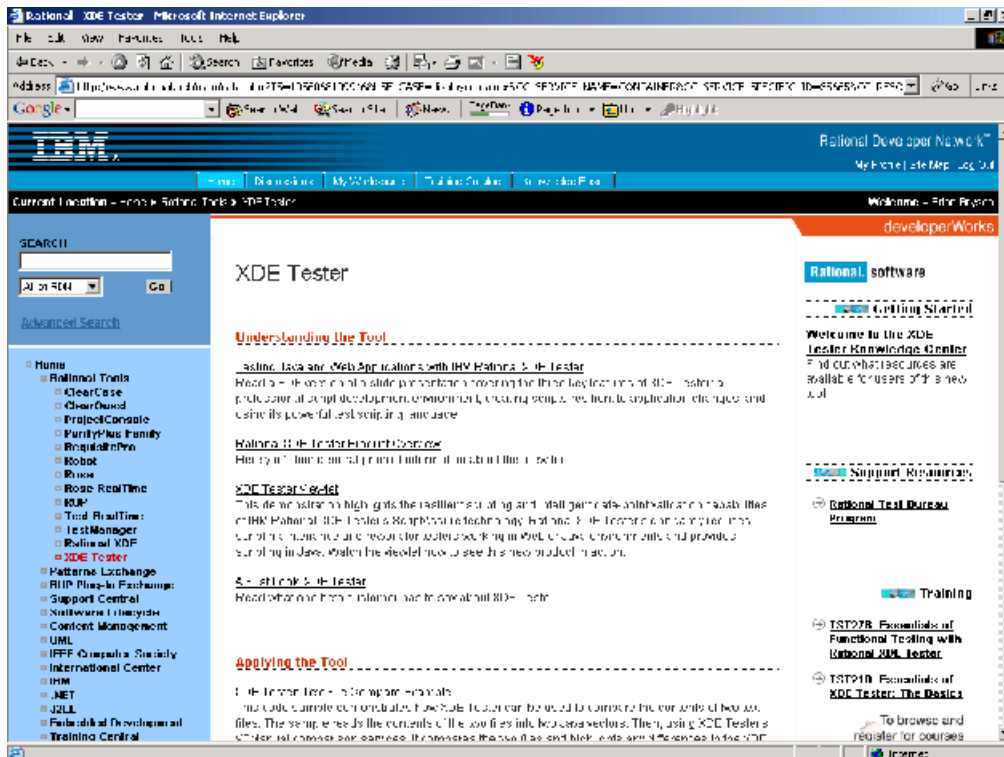
Next, you'll want to ensure that your browser or JVM is enabled. In this walk through, you configured your JVM in the [Install and Configure the Java Runtime Environment](#). If your application is using this JVM, then you're all set. If not, repeat these same steps, selecting your JVM.

### To enable your browser, complete the following steps:

1. Select Configure > Enable Environments for Testing
2. On the Web Browsers tab, select the web browser to enable, and click enable

3. On the Java Environments tab, select all JVMs and click enable. JVM enablement is necessary to test browsers, and is done automatically. This step is only a precaution.

## Rational Developer Network



To further assist you in your testing, visit the [Rational Developer Network](http://www.rational.net) at [www.rational.net](http://www.rational.net). On the developer network you will find the Rational XDE Tester Knowledge Center which contains many white papers and technical papers with information to help you get the most out of IBM Rational XDE Tester. Some of the valuable Rational XDE Tester-related resources you will find on RDN are:

- Getting started technical articles and demos of Rational XDE Tester written and created by tester/developers using Rational XDE Tester to build real-world solutions.
- A Rational Automated Testing Discussion Group that includes Rational XDE Tester and that provides a central place for Rational XDE Tester customers to exchange ideas and experiences.
- Web-based training for Rational XDE Tester and related training regarding Java & HTML testing
- Various code samples and utilities to help solve common Rational XDE Tester challenges

## Summary

---

This Evaluation guide has shown you how to set up Rational XDE Tester for testing, how to record and play back scripts and how to validate dynamic data. There are many other features and use cases of Rational XDE Tester that have only been covered at a very high level, such as the Rational ClearCase integration, or aren't covered at all, such as, cross-platform playback. Check out these powerful features and other important information in the Rational XDE Tester User Guide.



## Technical Support Contact Information

Technical Support is available to you during your evaluation of Rational XDE Tester. Should you need to contact support with any Rational XDE Tester related issues, please be prepared to supply the following information:

- Name, telephone number, and company name
- Product name and version number
- Operating system and version number (for example, Windows NT 4.0, Solaris 2.5.1/2.6/2.7, or HP-UX 10.20)
- Computer make and model
- Your case id (if you're calling about a previously reported problem)
- A summary description of the problem, related errors, and how it was made to occur

If your organization has a designated, on-site support person, please try to contact that person before contacting Rational Technical Support.

In addition, you can obtain technical assistance by sending an e-mail to Technical Support (see details below). E-mail is acknowledged immediately and is usually answered within one working day of its arrival at Rational. When sending an e-mail, place "**Rational XDE Tester or Testing**" in the subject line, and include a description of your problem in the body of your message.

North America	Europe, Mid East & Africa	Asia Pacific
Rational Software, 18880 Homestead Road, Cupertino, CA 95014 voice: (800) 433-5444 fax: (408) 863-4001 e-mail: <a href="mailto:support@rational.com">support@rational.com</a>	Rational Software, Beechavenue 30, 1119 PV Schiphol-Rijk, The Netherlands voice: +31 20 454 6200 fax: +31 20 454 6201 e-mail: <a href="mailto:support@europe.rational.com">support@europe.rational.com</a>	Asia Pacific: Rational Software Corporation Pty Ltd, Level 13, Tower A, Zenith Centre, 821 Pacific Highway, Chatswood NSW 2067, Australia voice: +61 2-9419-0111 fax: +61 2-9419-0123 e-mail: <a href="mailto:support@apac.rational.com">support@apac.rational.com</a>