

## Automation Test Tools

# Попугаи для тестеров

По статистике, стоимость тестирования программы составляет в среднем 50% цены начальной разработки и 70% цены поддержки ПО. Если полагать, что для каждого теста единственно необходимый ресурс — это клавиатура, то относительным спасением качества продукта будет только одно — увеличение количества тестеров. А если вы разрабатываете программу в одиночку? В этом случае, как и во многих других, выходом может быть использование программ автоматизированного тестирования.

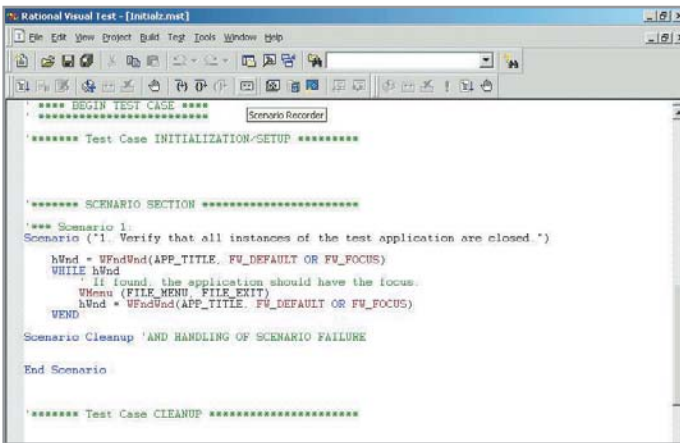
## Зачем это нужно?

Очень популярны в последнее время построенные на идее воспроизведения действий тестера, раз за разом неустанно и точно повторяющие самые монотонные проверки — «попугаи» и помощники тестеров.

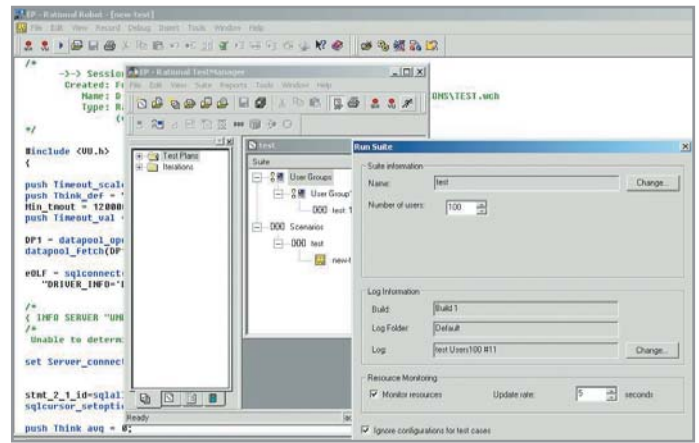
Существует несколько причин, по которым имеет смысл включить Automation Test Tools в свой арсенал. Во-первых, некоторые программы технически невозможно протестировать вручную. Например, идея ручного тестирования невидимых серверных приложений хоть и отличается определенной оригинальностью, но может поставить исполнителя в тупик.

Во-вторых, некоторые достаточно трудоемкие тесты необходимо повторять множество раз, а это неизбежно ведет к тому, что тестер пропускает их или выполняет невнимательно. Программа же не скучает, не злится на сжатые сроки и не хочет курить, а значит тесты всегда будут выполнены одинаково! В-третьих, Automation Test Tools могут позволить программисту самостоятельно тестировать плоды своих трудов без опасения стать жертвой психологической игры с собой, в ходе которой разработчик подсознательно тестирует именно «безопасные» участки кода. »





▲ Рис. 1. Visual Test — окно отладки



▲ Рис. 2. Rational Robot — создание нового тестового скрипта

## » Как научить поугая повторять?

В отличие от многих продуктов, предполагающих программирование в той или иной степени, первые шаги в освоении автоматизированного тестирования не только не отпугивают, но даже ободряют. Все просто. Открываем свеженькое отладочное окошко (рис. 1), предварительно определив папку для хранения скриптов и заполнив некоторые данные по проекту. Затем нажатием кнопки «Record» указываем программе, что можно начать фиксировать предпринимаемые действия. После этого запускаем тестируемую программу, работаем с ней, останавливаем запись. Все произведенные действия (ввод с клавиатуры и с помощью мыши) автоматически записываются в тестовый скрипт, который может быть затем «проигран»: совершенные только что действия выполняются уже без участия тестера.

Происходящее сильно напоминает манипуляции с видеоманитофоном: запись, остановка записи, последующее проигрывание записанного, пауза в случае необходимости. Снабдив полученный скрипт «точками верификации», можно будет проверять факт наличия и статус тестируемого объекта интерфейса, соответствие появившегося изображения ожидаемому результату или корректность изменения значения в заданной таблице базы данных. Разумеется, в конечном счете запись и воспроизведение все реже будут становиться частью процесса автоматизации, поскольку для непосредственного тестирования объектов, базы данных и т. д. более надежно использовать встроенные функции.

Опять же автоматизация тестирования реального проекта может привести и к ситуациям, когда стандартный язык используемой те-

стовой программы не поддерживает какой-либо функции. Однако и эту проблему во многих случаях можно решить созданием DLL-файла или расширением языка каким-либо образом, позволяющим совершить требуемое действие. Эта тема довольно сложная и может рассматриваться тестерами, которые используют Automation Test Tools по крайней мере 6–12 месяцев. Заметим только, что некоторые программы предоставляют расширение функционала для создания определяемых пользователем функций, методов, классов и т. д.

## Что невозможно без автоматикки?

Если для функционального тестирования Automation Test Tools приветствуются, то при нагрузочном тестировании без них уже просто не обойтись. В этом случае использова-

ние автоматизации сводится к моделированию сотен и тысяч одновременно работающих пользователей, что действительно нереально организовать силами не только одного, но и даже нескольких тестировщиков. На самом деле моделирование такого количества пользователей происходит на уровне прикладного протокола системы, то есть в случае web-приложений моделируется соответствующий трафик HTTP(S), в случае DCOM-среды — вызовы удаленных объектов, в случае баз данных Oracle — OCI-вызовы, и т. д.

После генерации скрипта, которая осуществляется так же, как и для функционального тестирования, просто указывается количество «виртуальных пользователей» (рис. 2). Кроме этого, происходит имитация одной или нескольких платформ (Windows NT/2000/Windows XP/Unix), с которых идет нагрузка от «виртуальных пользователей». Учи- »



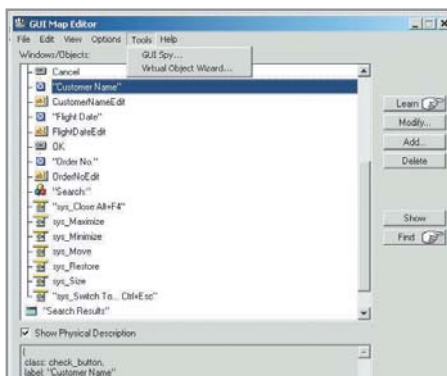
## Программа определения объекта

# Инструмент для любителей точности

В обычных условиях тестирования исходный код — недоступная тестеру информация. Идеологически это верно, но иногда внутренняя информация о некоторых свойствах тестируемого объекта все-таки необходима. Особенно если специфицированы такие конкретные требования, как, например, номер шрифта или высота строк в гридах, которые некорректно тестировать «на глаз». В Automation Test Tools, как правило, встроена специальная программа для определения свойств объектов. Это как бы «шпионская программа», которая просматривает «внутренности» объекта и передает вам его название, идентификационные свойства

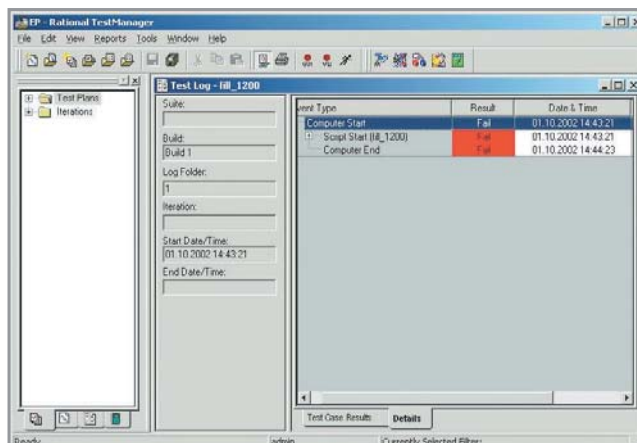
и другие подобные данные. Это позволяет соотнести выбранный объект с вызовом функции.

Такая программа особенно важна при автоматизированном тестировании для получения данных, которые уникально определяют тестируемую часть объекта или окно. Тестеру предоставляется указатель и служба идентификации. С помощью мышки вы можете указать на какой-то объект и в определенном окне увидеть все его свойства. Большинство программ дают возможность просматривать все открытые приложения параллельно и выдавать результат в виде дерева, которым вы можете пользоваться при необходимости.



▲ Рис. 3. WinRunner Map — карта тестируемого проекта

Рис. 4. Rational Test Manager — автоматический отчет по результатам тестирования блока



» тывая степень важности автоматизации нагрузочного тестирования, практически все Automation Test Tools в той или иной степени предполагают такую возможность.

### Лучшие породы «попугаев»

В виду существования большого количества средств автоматизированного тестирования, за основу для этого документа выбраны только наиболее популярные средства: WinRunner (Mercury Interactive Corporate), LoadRunner (Mercury Interactive Corporate), Rational Robot (Rational), Visual Test (Rational). Все они решают в принципе одну задачу, поэтому выделим только основные их «идеологические» отличия.

### WinRunner

Если проектов много и автоматизация сводится к созданию и прогону больших блоков регрессионных тестов на различных платформах, то наиболее подходящим представ-

ляется WinRunner. Наличие очень удобного GUI MAP (рис. 3) позволяет существенно облегчить тестирование с переменными данными, когда имена, адреса, цифры, прошитые в коде, заменяются на переменные значения из внешнего источника, например из CSV-файла, крупноформатной таблицы или базы данных.

Все программы дают возможность для поиска и замены, но лучше всего реализованы те, которые предоставляют центральный репозиторий для хранения уникальных ссылок этих объектов (это и есть упомянутый выше GUI MAP). Суть в том, что гораздо проще изменить ссылку в одном месте, чем прокручивать каждый сценарий заново и таким образом изменять его.

### LoadRunner

LoadRunner несколько выбивается из общего ряда, поскольку является программой, «заточенной» именно под нагрузочное тестирова-

ние. Для диагностики системы во время теста в LoadRunner Controller есть средства мониторинга как для различных компонентов системы, так и для ее разных уровней. Например, для контроля за загруженностью сети на всем пути прохождения трафика служит функция Network Delay Monitor (рис. 5). Измеряя время реакции, сетевую задержку, а также производительность серверной и прикладной частей приложения, система тестирования позволяет определить «узкие места» проверяемой программы и настроить ее на максимальную производительность.

### Rational Robot

Если автоматизировать предполагается разнообразные проекты с большой долей автоматизированных скриптов, сложной структурой и возможностью учета результатов автотеста в общем мониторинге проекта, то наиболее полномасштабным представляется Rational Robot. Планирование автоматизированного тестирования с возможностью собирать различные тесткейсы в блоки, прогон группы таких блоков по мере необходимости, структурированный отчет по результатам прогона скриптов в Test Manager (рис. 4) — это только малая часть возможностей, предо- »

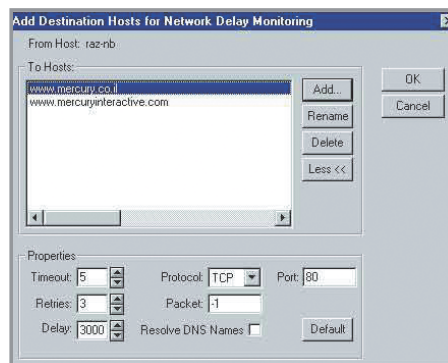
## Недостатки Automation Test Tools

### Нет в мире совершенства

Поскольку первое знакомство с Automation Test Tools вызывает искушение с самого начала составить большое количество тестовых скриптов, то имеет смысл напомнить любителям повальной автоматизации и о некоторых недостатках такого тестирования:

- ▶ Тестовые сценарии сами могут быть источниками ошибок.
- ▶ Хороший блок тестовых скриптов требует затрат на планирование и отладку. Если увлечься, то они превысят затраты на ручное тестирование, что лишит использование такой программы смысла.

- ▶ Если блок скриптов запущен на выполнение, то другие виды деятельности на этом компьютере придется отложить.
- ▶ Необходимо также учитывать психологический фактор, который заставляет полагаться на результаты автоматизированного тестирования, а значит увеличивает риск пропущенных ошибок при ручном тестировании этой части программы.
- ▶ Автоматизация нагрузочного тестирования предъявляет достаточно высокие требования к компьютеру, на котором она будет организована.



▲ Рис. 5. Load Test → Network Delay Monitor

» ставляемых TestStudio (Rational), в составе которой поставляется Rational Robot.

Связь с программой Rational ClearQuest дает возможность конвертировать любой результат непройденного теста в ошибку, что используется при управлении дефектами и отслеживании динамики производственного процесса. Rational SiteCheck, являясь частью Rational Robot, обеспечивает функциональное тестирование web-сайта, автоматически проходя его насквозь и проверяя все гиперссылки (внешние и внутренние) и актив-

ные страницы. Link Wizard позволяет быстро обнаруживать разорванные связи и страницы, на которые нет ссылок, и предоставляет встроенный HTML-редактор, функцию Drag&Drop, реорганизацию файлов и многое другое. Также есть возможность тестировать защищенные сайты по SSL-соединению.

### Visual Test

Наиболее экономным командам, а также имеющим небольшие запросы по части автоматизации процесса отлично подойдет Visual

Test. Эта программа в пять раз дешевле, чем другие, правда, этому есть причина: всего спектра решений, наличествующего у других, она все же не предоставляет. Продукт имеет интерфейс, сходный с VisualStudio от компании Microsoft. То есть при изучении программы тратится минимум времени.

Основу гибкости Visual Test составляет производный от Visual Basic расширенный язык программирования TestBasic со специальными конструкциями для облегчения тестирования, простого доступа к Windows API »

Программа	LoadRunner	Rational Robot	Visual Test	WinRunner
Цена, €	от 44 000	Поставляется в двух комплектах: TeamTest — 7235 TestStudio Suite — 4825	от 689,9 (при этом тесты можно распределять по разным машинам, что не требует нескольких лицензий)	от 6400
Операционная система	Windows NT, 2000, XP Sun Solaris, HP-UX, IBM AIX, Linux	Windows 95, 98, NT, 2000, XP	Windows 95, 98, NT, 2000	Windows 98, NT, 2000, XP
Дистрибутив, Мбайт	от 100 до 300 в зависимости от модификации	не менее 28,2 в составе TeamTest или TestStudio Suite	21,3	132
<b>Характеристики</b>				
Поддерживает	Java, ActiveX, XML, SAP, PeopleSoft, HTTP(S), WAP, LDAP, SSL	HTML, DHTML, Java, Visual Basic, Visual C++, Oracle Developer/2000, Delphi, SAP, PeopleSoft, Sybase Powerbuilder	Продукты фирмы Microsoft C++, Power Builder, Visual Basic; web-приложения.	При установке Java add-in – Java приложения и апплеты, Delphi add-in – Delphi приложения
Язык скриптов	TSL (Test Script Language) – аналог C	SQABasic — аналог Visual Basic	Основан на Visual Basic	TSL (Test Script Language) – аналог C
Взаимодействие с другими программами	Test Director, WinRunner	Интеграция с LoadTest, ClearQuest, Purify, Quantify, Purecoverage, RequisitePro, Rose с TestManager Экспорт данных: Word, Excel	PureCoverage, Purify, Quantify, Clearcase, ClearQuest, Microsoft Visual Studio и со всеми продуктами, используемыми Visual C++ IDE.	LoadRunner, Test Director Экспорт данных: Excel
Программа определения объекта	+	+	+	+
Карта объекта	+	-	-	+
Тестирование интернет-приложений	отлично	отлично	удовлетворительно	отлично
Тестирование баз данных	отлично	отлично	плохо	отлично
Тестирование изображений	хорошо	отлично	хорошо	отлично
Восстановление тестирования после ошибки	отлично	хорошо	хорошо	хорошо
<b>Итог</b>				
Достоинства	Полный мониторинг процесса тестирования с любой графикой. Стабильность	Полнота функционального тестирования, взаимодействие с многочисленными внешними программами	Очень удобная программа определения объекта. Простота в использовании	Рациональный центральный репозиторий для хранения объектов и их свойств. Очень хорошая сопроводительная документация
Недостатки	Крайне высокая цена	Информация об элементах интерфейса хранится непосредственно в тексте скрипта	Ограниченная область применения	Слабая расширяемость функциональных возможностей языка
Комментарии	Оптимальное средство для нагрузочного тестирования действительно крупных проектов	Мощное средство для всестороннего тестирования продукта. Особенно подходит командам, уже использующим продукты Rational Software	Компактная, удобная программа для тестирования продуктов, разработанных средствами Microsoft	Полнофункциональная программа с отличными «мастерами». Оптимальна для функционального тестирования web-приложений

» и открытой архитектурой, которая делает этот язык расширяемым. Visual Test, мне кажется, докажет еще свою конкурентоспособность — после того как расширится функционал этой программы, поскольку до недавнего времени она не поддерживала тестирование интернет-приложений.

### Когда автоматика мешает

Практика показывает, что автоматизация идеально подходит для таких видов тестирования как приемочное, конфигурационное и нагрузочное. Приемочное выглядит в этом случае как запуск блока скриптов на тестовой машине, например перед уходом домой. За ночь все они будут пройдены с выводом результатов. И новый рабочий день можно начинать или с обоснованной отправки версии на доработку, или уже с функционального тестирования. Автоматизация конфигурационного тестирования позволяет избежать повторения раз за разом одних и тех же проверок на различных платформах и в изменяющемся окружении. И наконец, на

грузочное тестирование без автоматизации просто не может быть проведено корректно.

Не приветствуется использование Automation Test Tools на самых ранних стадиях разработки софта. Постоянные изменения интерфейса и даже функционала заставляют переписывать скрипты чуть ли не перед каждой проверкой, и зря теряет свою целесообразность. Лишь когда софт в рамках проекта приобретет известную статичность, имеет смысл взяться за выбор тестов, нуждающихся в автоматизации.

Выбор этот делается на основании достаточно простых расчетов. Если среднее время на создание теста при автоматическом тестировании примерно в пять раз превышает этот показатель для ручного тестирования, то при прогоне теста ситуация кардинально меняется: среднее время на прогон автоматизированного теста почти в десять раз меньше, чем аналогичный прогон вручную. Из этого естественным образом следует, что тесты, которые выполняются всего несколько раз, вообще

автоматизировать не следует. И в то же время, если создаваемый тест предполагается прогонять более десяти раз, то он — явный кандидат на автоматизацию.

Итак, Automation Test Tools помогают забыть о самых скучных участках тестовой деятельности, сэкономить время, организовать полноценное нагрузочное тестирование. Известно, что большинство программ предоставляют одинаковые функциональные возможности, но разными путями. Поэтому, если среда разработки тестируемого продукта поддерживается всеми представленными программами, то выбор здесь обусловят скорее субъективные предпочтения. А когда «свой» Automation Test Tool будет выбран, останется только удержаться от желания полностью довериться в плане тестирования этому замечательному помощнику. При всей своей мощи и гибкости подобные программы не исключают тестирования вручную — как попугаи, они способны только повторять то, что им приказали. Не более.

■ ■ ■ Светлана Кондратенко

**В каждом номере на Chip CD**

- ▶ популярные freeware и shareware программы для Windows, Linux и MacOS
- ▶ тесты программного обеспечения и аналитика
- ▶ утилиты и драйверы
- ▶ обзоры игр
- ▶ демо-версии новейших продуктов
- ▶ материалы, не вошедшие в номер
- ▶ электронная версия журнала

CD-ROM content includes: NetAnt, ZoneAlarm 2.0, Setup Generator, Napster и другие, Linux KDE 2.1, Moorhuhn, Тест messenger'ов, Windows Media 8.

**CHIP CD**