

产品简介

自动测试简介

自动测试过程就是通过模拟人工操作，完成对被测试系统的输入，并且对输出进行检验的过程。

自动测试是由软件代替人工操作，对被测试系统的 GUI 发出指令，模拟操作，完成自动测试过程。

● 测试脚本

自动测试，就是使用一个程序来测试另一个程序（被测试的应用系统）功能的正确性。如果用来测试的程序本身非常复杂，也需要被测试，或者编写困难，那么自动测试就失去了意义。

因此，用来测试另外一个程序的程序往往是非常简单的，我们把这个程序称为“测试脚本”。

测试脚本通常在测试工具的 IDE 里执行，并且获得 IDE 的支持。

● 自动记录

当我们编写测试脚本的时候，往往发现编写脚本本身是很困难的：了解脚本的语法、了解测试过程、把测试过程转换称为测试脚本语句。自动记录，就是通过记录一个操作过程来获得测试脚本的功能。通过自动记录，我们就能够得到一个操作的基本的脚本，通过修改这个脚本，我们得到更通用的测试脚本。

● 同步点

在执行测试脚本的时候，测试脚本语句的操作对象是 GUI 的组件。测试脚本通过这个组件的属性（如：名称、位置、winclass、disable 等）来确定哪个组件是我们需要操作的组件。这个查找组件的过程如果失败，意味着：第一，应用系统的响应比较慢，需要等待一段时间再进行一次定位；第二，该组件不存在。这个查找、定位组件的过程，我们称为同步点。AutoRunner 的同步点都是隐含方式的：在操作对象的时候进行自动同步，自动设置同步点。

● 检查点

测试的目的是检查数据是否正确。在测试的过程中，我们需要检查某个组件的某些属性满足某个条件。这个检查的位置和条件，我们称为检查点。在 AutoRunner 中，使用 check("objectname", "property", "期望值") 来作为检查点的脚本语句，它检查对象 objectname 的属性 property 是否和期望值一致。在使用中，可以使用检查点来检验对象的各个方面属性。

● 循环参数表与数据驱动

测试脚本是针对一个测试过程的。一个测试过程往往需要众多的数据来测试。通过自动录制得到的脚本，所有的输入数据都是常数，是固定的。如果需要使用一个测试脚本测试多组数据，就需要对脚本进行参数化，把固定的常数修改为来自数据源变量。这个过程我们称为参数化。采用了参数化的脚本，我们称为数据驱动的模式。使用 AutoRunner 完成自动测试：通过录制的方式自动生成测试脚本，不需要用户通过编写测试脚本来创建；通过检查点向导来创建检查点，只面向业务，不需要手工修改脚本；自动的参数化和数据驱动支持，一步到位的创建数据驱动脚本；

AutoRunner 功能测试工具简介

AutoRunner 是自动化的功能测试工具。功能测试的目标是根据 GUI 的界面或者报表来检查软件的实际功能是否和需求定义的功能相一致。下面简要介绍一下自动测试的基本概念。

AutoRunner 自动测试工具，是黑盒测试工具，可以用来完成功能测试、回归测试、每日构建测试、自动回归测试等测试工作。

AutoRunner 是具有脚本语言、提供针对脚本完善的跟踪功能的、支持 IE 测试和 Windows Native 测试工具。

产品可以进行：

- 基于 B/S 架构测试——对 B/S 系统进行功能测试，支持各种 B/S 应用和 Web 网站的测试。
- 基于 C/S 架构测试——对 C/S 系统进行功能测试，支持大多数的 C/S 系统测试。
- Net 测试——对 .Net 类型的应用软件进行功能测试；支持标准 Windows 应用程序测试和 .Net 应用程序测试；

产品特点：

- 使用 BeanShell 语言作为脚本语言，使脚本更少，更易于理解。
- 提供了强大的测试案例编辑功能。
- 支持各种同步点，包括识别自动同步点和自己扩展的同步点。
- 支持手动参数化和自动参数化。
- 支持模糊识别，通过对组件设置和权重识别来实现模糊识别，提高了脚本执行的可靠性和兼容性。
- 支持 JAVA 语法分析的编辑器，关键字提示，提高脚本编写的效率。
- 支持测试过程的错误提示功能。
- 支持 Java 组件、IE 组件的录制和回放。

- 测试脚本支持校验点，保证脚本能够准确执行，对数据进行检验。允许用户在某个时刻从被测试系统中获取组件的信息，例如：一个对话框上的按钮的名字和是否被激活等属性信息。
- AutoRunner 支持数据驱动脚本，能够通过数据驱动的方式来控制脚本运行的分支。
- 有权重的模糊识别组件技术。在脚本执行过程中，支持模式识别组件，提高了识别效率，提高了脚本的兼容性和可靠性。组件的识别，可以通过 IDE 的组件的识别权重进行设置。AutoRunner 提供了两种方法来设置模糊识别权重：第一，通过修改系统的配置文件来修改默认的认识权重。它控制了所有录制得到的系统属性的识别权重；第二，可以在 IDE 中单个修改某个组件的权重。模糊识别的价值，在于当用户的 UI 不断被修改的时候，测试脚本仍然能够位置发生改变或者其他非重要信息发生改变的组件，降低脚本被重写的可能性，提高工作效率。
- JAVAGUI 测试，采用 JAVA Accessible bridge 技术。实现 JAVA Accessible Bridge 技术，完美的实现了针对 JAVAGUI 应用的测试，可以实现自动录制和脚本执行。被测试系统不需要做任何修改就可以进行测试。
- 丰富的脚本命令函数。AutoRunner3.0 新增了许多命令函数，有利于测试人员进行各种功能测试，熟练掌握这些命令函数，能够让测试人员编写出更简练、更高效的测试脚本。

上海泽众软件科技有限公司

www.spasvo.com

021-31357887