

QTP 自动化测试



自动化测试的好处

- 快速** QuickTest 执行测试比人工测试速度快多了。
- 可靠** QuickTest 每一次的测试都可以正确的执行相同的动作，可以避免人工测试的错误。
- 可重复** QuickTest 可以重复执行相同的测试。
- 可程序化** QuickTest 可以以程序的方式，撰写复杂的测试脚本，以带出隐藏应用程序中的信息。
- 广泛性** QuickTest 可以建立广泛的测试脚本，涵盖应用程序的所有功能。
- 可再使用** QuickTest 可以重复使用测试脚本，即使应用程序的使用接口已经改变。



QTP背景

- *BTO: Business Technology Optimization* 业务科技优化方案
 - 使IT系统既能满足质量和成本的需求，又能适应多变的业务需求，可随之改变
- QTP是HP花费45亿美元购买的Mecury公司测试产品套件之一



- *QTP*是继*WR*后推出以*VBS*为内嵌语言的测试工具
- 用于功能测试
- *QC*产品协作进行自动化测试开发 (*BPT*) *BusinessProcessTesting*
- 特点：价格高昂但是容易掌握



QTP应用范围

- 自带插件
- *ActiveX Database StdWindows TEA VisualBasic Web Xml*
- 额外购买插件
- *Java DotNet Peoplesoft VisualAge siebel Oracle*

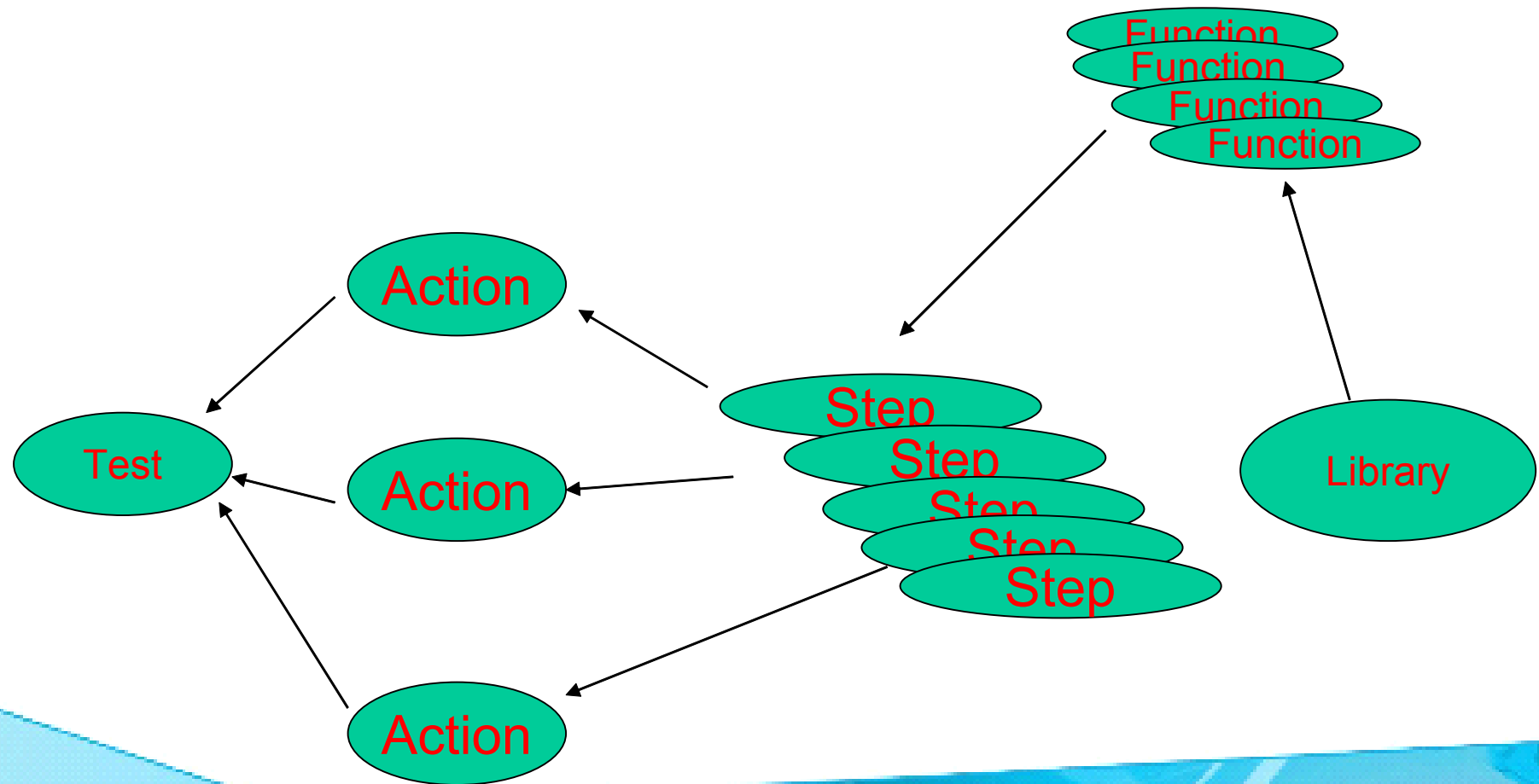


QTP基本功能介绍

- *HP QuickTest Professional* 支持功能测试和回归测试自动化，用于每个主要软件应用程序和环境。此解决方案使用关键字驱动测试概念，简化了测试创建和维护过程。它使测试人员能够使用专业的不感技术直接从应用程序屏幕中捕获流程来构建测试案例。测试专家还可以通过集成的脚本和调试环境完全访问内在测试和对象属性。



QTP的基本功能介绍



以Test为测试运行单位，每个Test包含若干类型Action。
以Action为最小单位映射对象库，多个Action之间可以共享统一对象库
Action内的Step可以包含用户录制操作或者函数库内对Function的调用

使用QuickTest进行测试

- 测试和业务组件
- 测试：组织成一个或多个操作的步骤集合，用于验证应用程序是否按预期执行。
- 业务组件：表示应用程序中单任务的步骤集合。业务组件（也称为组件）在 *Mercury Quality Center* 中由业务流程测试组合为特定的场景以建立业务流程测试。



测试流程

- 使用 *QuickTest* 测试包括三个主要阶段

创建测试或组件

运行测试或组件

分析结果

创建测试或组件

- 以下列一种或两种方式向测试中添加步骤
 - 在应用程序或网站上录制会话。
 - 建立对象库并使用这些对象在关键字视图或专家视图中手动添加步骤
- 然后，使用特殊的测试选项或编程语句来修改测试或组件。
- 通过用参数替换固定值拓宽测试或组件的范围
 - 使用设计的数据表中的数据
 - *QuickTest* 生成随机数据
- 使用 *QuickTest* 中众多的功能测试功能来增强测试或组件，或添加编写语句来实现更复杂的测试目标。



运行测试或组件

- 运行测试或组件检查网站或应用程序
 - 运行测试或者组件以对其进行调试
 - 单步执行
 - 单步跳过
 - 单步退出
- 设置断点使测试或组件在预定点暂停



分析结果

- 在“结果”窗口中查看结果。
- 报告在运行会话过程中检测到的缺陷。



QuickTest窗口

- 标题栏&菜单栏
- 文件工具栏
- 调试工具栏
- 测试工具栏
- 操作工具栏
- 测试窗格：关键字、专家试图
- 数据表
- 运行窗口
- 调试查看器



基本使用介绍

- 各个按钮介绍



使用测试对象

- 什么是对象？



QuickTest如何识别对象

- 确定对象符合的测试对象类
- 对测试对象进行分类
 - 标准 *Windows* 对话框(*Dialog*)
 - *Web* 按钮 (*WebButton*)
 - *Visual Basic* 滚动条对象 (*VbScrollBar*)
- 强制属性列表
- 添加辅助属性
- 特殊的顺序标识



应用测试对象模型

- 在录制时执行应用程序的相应操作，则 *QuickTest* 将
 - 标识 *QuickTest* 测试对象类（表示执行了操作的对象），并创建相应的测试对象
 - 读取应用程序中对象属性的当前值，然后将属性和属性值列表与测试对象一起存储
 - 选择该对象的唯一名称，一般使用该对象某个重要属性的值
 - 使用适当的 *QuickTest* 测试对象方法录制对对象执行的操作



实例

- `<INPUT TYPE="submit" NAME="Find" VALUE="Find">`
- 对象标识为 **WebButton** 测试对象
- 创建一个名为 *Find* 的 *WebButton* 对象
- 为 *Find WebButton* 对象存储属性和属性值

属性	值
type	Find
name	reset
Html tag	input

实例（续）

- 录制对 *WebButton* 执行了 **Click** 方法
- 关键字视图

	项	操作	文档
• 专家视图	Action1		
- Brows	Mercury - Business Tech...		
- WebE	Mercury - Business ...		
	search_right	Click	单击 "search_right" image。



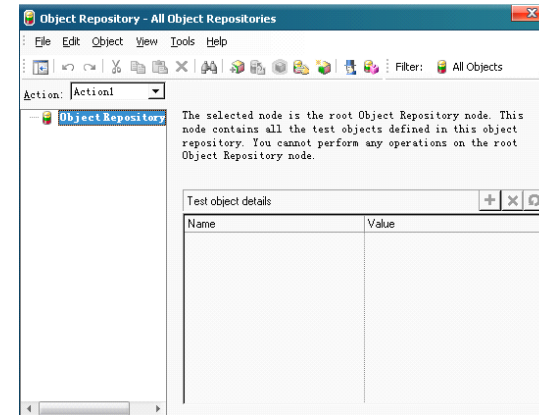
核心功能清单

- 对象库
- 描述性编程
- 检查点
- 对象属性运行期间变更
- 数据驱动
- 错误恢复
- 自动化对象



核心功能-对象库

- 入口:
tools->Object Repository
Ctrl+R
- 功能
保存被测试对象属性,
在编写执行步骤时使用



核心功能-对象库

- 对象库中的对象可以在录制过程中自动产生，也可以手动添加指定对象
- 对象库分为本地对象库和共享对象库，区别是：
 - 本地对象库只应用于本地*Action*调用
 - 共享对象库以*.tsr*扩展名文件形式存在，可以被其他*Action*调用。
- 对象库面板具有定义以及更新对象的功能，并且能够进行在被测软件和对象库之间双向查找对象。
- 对象属性可以在对象库中被任意修改，并且其属性描述支持正则表达式。

核心功能-对象库

- 智能识别是鸡肋，辅助标识符要慎用
- 对象库与描述性编程的区别：
对象库将对象收集起来，以可视化的方式呈现给用户
描述性编程在运行步骤中通过对象属性指定对象。
- 结合使用时应注意：描述性编程的子对象不可以是对象库中的对象，反之可以。
- 具体采用哪种方式指定对象因具体情况和个人喜好所定，并非描述性编程更“高人一筹”



核心功能-对象库-简单的对象识别原理

- 用户通过录制或指定对象属性的方式将一个对象添加到对象库，在录制过程中添加的对象是根据 *Tools->Object Identification* 中的设定记录的。
- 在回放的过程中 *QTP* 通过将对象库中的对象属性和被测软件中的实时对象属性进行比对，如果发现存在能够唯一确定的对象，则对对象使用相应的用户设定的方法，完成用户动作。若不能发现唯一确定的对象，则启动智能识别机制，若智能识别仍无法唯一确定对象，则报错。
- 更多细节请参见 *QTP* 帮助。



核心功能-描述性编程

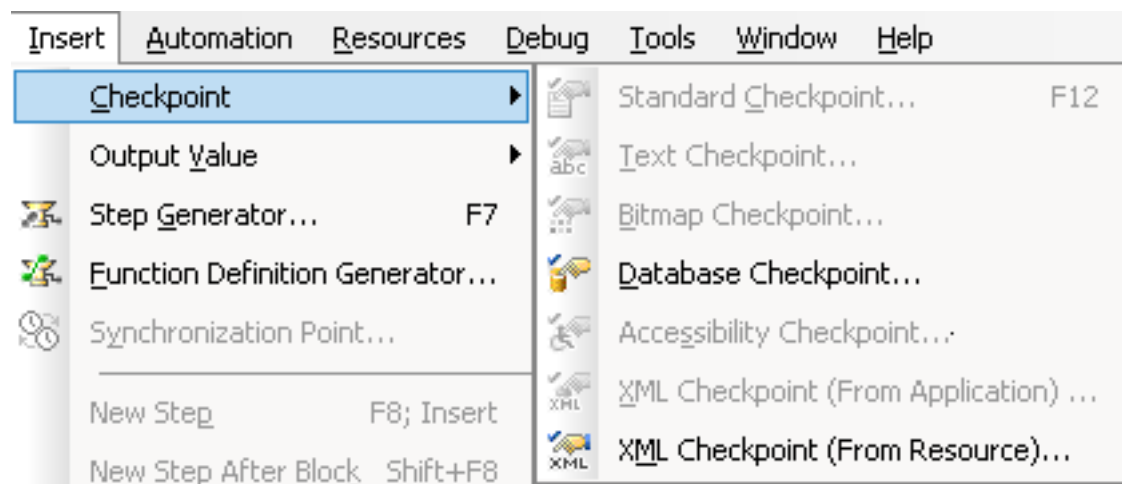
- *'Browser("name:=UML软件工程组织").Page("Title:=UML软件工程组织").link("text:=项目管理","html tag:=A").Click*
- *Set desc=Description.Create*
- *desc("text").value="项目管理"*
- *desc("html tag").value="A"*
- *Browser("name:=UML软件工程组织").Page("Title:=UML软件工程组织").link(desc).Click*

对象类型("识别属性1: =属性值").子对象属性("识别属性1: =属性值","识别属性2: =属性值")。。。方法

此外，描述性编程中属性值默认是支持正则表达式的，在使用中请留意。

由于支持了正则表达式，就可以一次获取多个测试对象，便可以逐个处理这个相似对象集。

核心功能-检查点



可检查类型包括上图中所有Checkpoint中子菜单项，包括文字，位图，XML，数据库（数据表）等检查点。

每个检查点的执行结果都会在Automation-Result中存在相应的记录。

有些检查点需要在录制状态下才能够使用。

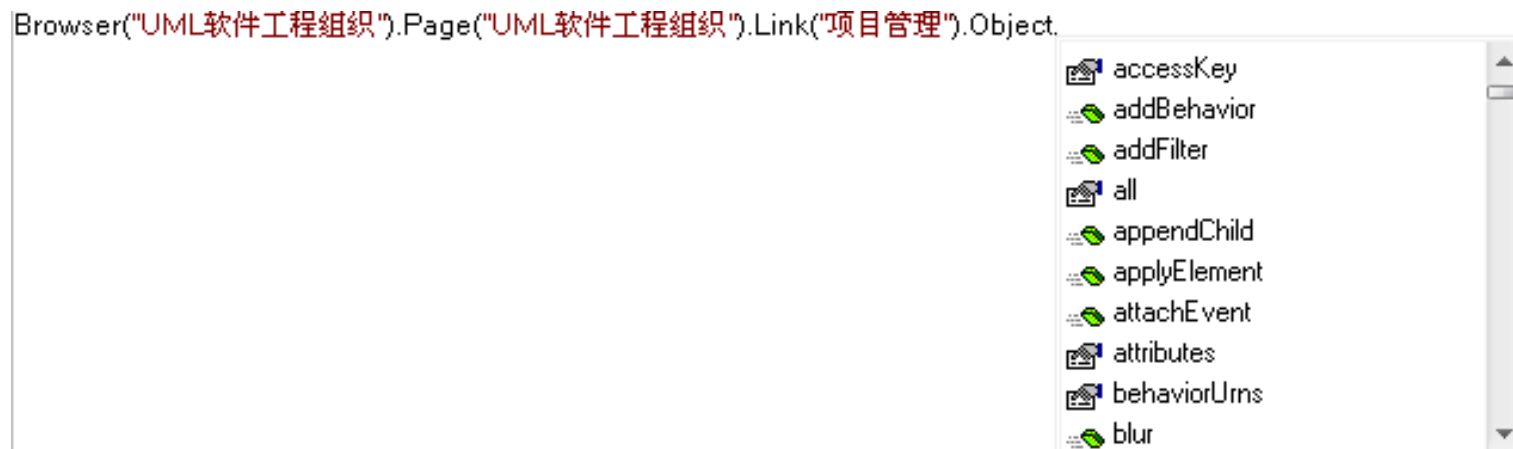
核心功能-对象属性运行期间变更

- *Object(description).***SetTOProperty** *Property, Value*
- 通过此方法能够在运行期间更改对象库中对象的识别属性值。尤其在一个循环逻辑中更为常用，通过每次循环修改属性值，完成对一系列类似对象的相同动作。此方法弥补了对象库相对于描述性编程的不足。
Browser("UML软件工程组织").Page("UML软件工程组织").Link("项目管理").SetTOProperty "text","需求管理"
- *Browser("UML软件工程组织").Page("UML软件工程组织").Link("项目管理").Click*



核心功能-对象属性运行期间变更

- 更多方法



通过 .object 对象，能够使用对象本身（没有被 QTP 封装）的更多方法或属性值设定。

核心功能-数据驱动

- 通过变更不同的数据组合，执行相同的逻辑步骤，完成对一个测试用例的测试。
- 数据来源可以是：*Excel*表格，数据库（数据表），*XML*文件，*TXT*文件。任何有规则的*VBS*可以操纵的文件类型都可以作为*QTP*的数据来源。当然最常用的还是*Excel*表。



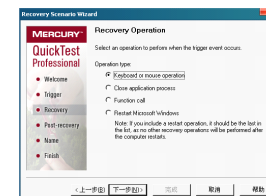
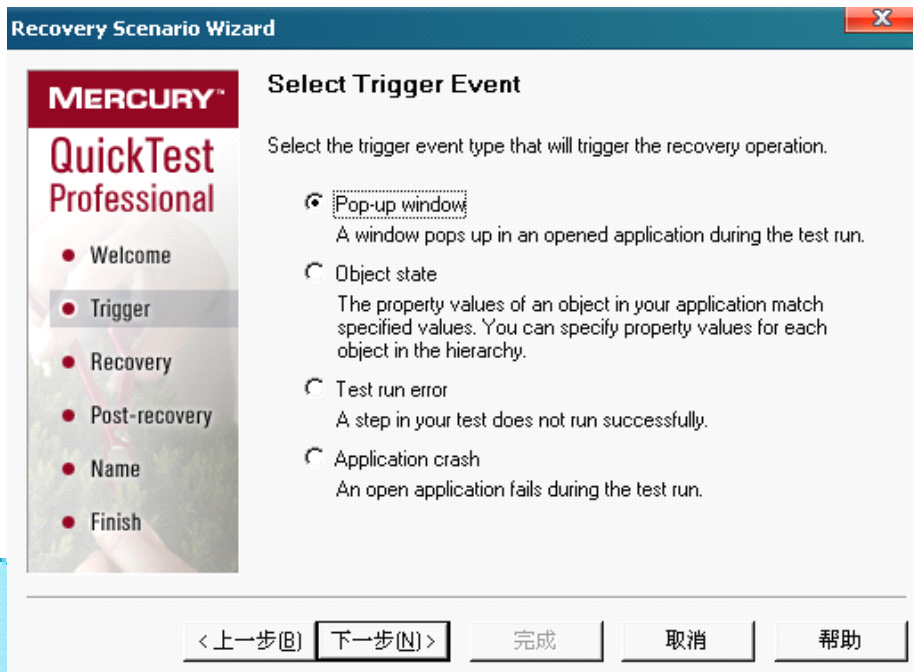
核心功能-错误恢复

- *QTP*软件本身自带了错误恢复手段 *Recovery Scenario Manager*。通过在发生不同错误的时候，执行指定的操作，保证测试脚本能够**24*7**的不间断运行。
- 用户也可以通过在容易发生错误的步骤或函数调用处执行相应的操作，保证无人值守时不宕机，脚本不中断。



核心功能-错误恢复-Recovery Scenario Manager

- 入口: *Resources->Recovery Scenario Manager*



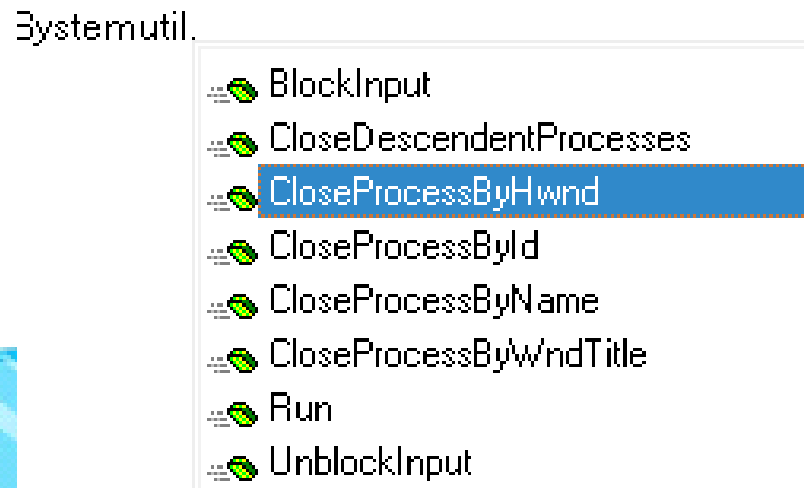
不同的错误条件和用户可执行操作如上图所示，具体到问题的处理，还需要QTP的使用者在开发过程中不断揣摩优化。

核心功能-错误恢复-用户自定义错误恢复

- 举例

在某一个用户动作执行时，总会产生某个不需要的，并且会阻止测试正常运行的软件打开，或者过程存在。这个时候就要进行人为干预。

Browser("xxxx").close ,或者



核心功能-自动化对象

QTP自身就了实现自动化对象的COM，可以通过创建QTP对象完成对QTP一些运行环境及错误收集等细节的指定。具体请参考下图中手册。网上很多，自己下载。

