



HP UFT Web 插件扩展性

软件版本： 12.50
Windows® 操作系统

开发人员指南

文档发布日期： 2015 年 7 月
软件发布日期： 2015 年 7 月

法律声明

担保

HP 产品和服务的唯一担保已在此类产品和服务随附的明示担保声明中提出。此处的任何内容均不构成额外担保。HP 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

此处所含信息如有更改，恕不另行通知。

受限权利声明

机密计算机软件。必须拥有 HP 授予的有效许可证，方可拥有、使用或复制本软件。按照 FAR 12.211 和 12.212，并根据供应商的标准商业许可的规定，商业计算机软件、计算机软件文档与商品技术数据授权给美国政府使用。

版权声明

© Copyright 1992 - 2015 Hewlett-Packard Development Company, L.P.

商标声明

Adobe® 和 Acrobat® 是 Adobe Systems Incorporated 的商标。

Google™ 和 Google Maps™ 是 Google Inc 的商标。

Intel® 和 Pentium® 是 Intel Corporation 在美国和其他国家/地区的商标。

Microsoft®、Windows®、Windows® XP 和 Windows Vista® 是 Microsoft Corporation 在美国的注册商标。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。

文档更新

此文档的标题页包含以下标识信息：

- 软件版本号，用于指示软件版本。
- 文档发布日期，该日期将在每次更新文档时更改。
- 软件发布日期，用于指示该版本软件的发布日期。

要检查是否有最新的更新，或者验证是否正在使用最新版本的文档，请访问：<https://softwaresupport.hp.com>。

需要注册 HP Passport 才能登录此站点。要注册 HP Passport ID，请访问 <https://softwaresupport.hp.com> 并单击 **Register**。

支持

请访问 HP 软件联机支持网站：<https://softwaresupport.hp.com>

此网站提供了联系信息，以及有关 HP 软件提供的产品、服务和支持的详细信息。

HP 软件联机支持提供客户自助解决功能。通过该联机支持，可快速高效地访问用于管理业务的各种交互式技术支持工具。作为尊贵的支持客户，您可以通过该支持网站获得下列支持：

- 搜索感兴趣的知识文档
- 提交并跟踪支持案例和改进请求
- 下载软件修补程序
- 管理支持合同
- 查找 HP 支持联系人

- 查看有关可用服务的信息
- 参与其他软件客户的讨论
- 研究和注册软件培训

大多数提供支持的区域都要求您注册为 HP Passport 用户再登录，很多区域还要求用户提供支持合同。要注册 HP Passport ID，请访问：<https://softwaresupport.hp.com> 并单击 **Register**。

要查找有关访问级别的详细信息，请访问：<https://softwaresupport.hp.com/web/softwaresupport/access-levels>。

HP 软件解决方案以及集成与最佳实践

访问 **HP Software Solutions Now** (<https://h20230.www2.hp.com/sc/solutions/index.jsp>)，探究 HP 软件目录中的产品如何协同工作、交换信息和解决业务需求。

访问 **Cross Portfolio Best Practices Library** (<https://hpln.hp.com/group/best-practices-hpsw>)，获取大量最佳实践文档和材料。

目录

欢迎使用 HP UFT Web 插件扩展性	7
关于 UFT Web 插件扩展性 SDK	7
关于 UFT Web 插件扩展性开发人员指南	8
其他联机资源	9
第 1 部分: 使用 Web 插件扩展性	10
第 1 章: 简介 UFT Web 插件扩展性	11
关于 UFT Web 插件扩展性	12
Extensibility Accelerator	12
标识 Web 插件扩展性的构建块	13
决定何时使用 Web 插件扩展性	13
分析示例自定义控件的默认 UFT 支持和扩展性选项	14
了解如何实现 Web 插件扩展性	18
Web 插件扩展性示例	19
第 2 章: 计划工具包的 UFT 支持	20
关于计划工具包的 UFT 支持	21
准备为自定义工具包创建支持	21
决定工具包相关信息	21
决定每个自定义控件类型的支持信息	21
了解 Web 插件扩展性计划清单	22
Web 插件扩展性计划清单	23
接下来要执行的操作	24
第 3 章: 开发工具包支持	25
关于自定义工具包支持	26
开发不受浏览器影响的支持	26
创建自定义工具包支持集	27
了解测试对象配置文件	28
UFT 如何加载测试对象配置 XML	30
了解 UFT 如何合并测试对象配置文件	31
扩展现有测试对象类	32
为自定义测试对象类提供帮助文件	33
了解工具包配置文件	33
为工具包支持集设计 JavaScript 函数	34
全局 JavaScript 方法和实用程序方法	38
示例: 使用同级控件标识对象	39
使 UFT 标识用于自定义 Web 控件的测试对象类	40
使用条件元素	41
在开发期间测试工具包支持集	44
记录和调试自定义支持	46
使用 Microsoft Windows 事件日志	46
调试 JavaScript 文件 (仅限 Internet Explorer)	48
实现对测试对象方法的支持	49

支持方法参数值的动态列表	50
实现对标识属性的支持	52
自定义测试对象名称	53
实现识别子控件的筛选	53
实现对录制的支持	55
疑难解答和限制——开发支持	57
第 4 章: 部署工具包支持集	58
关于部署自定义工具包支持	59
部署自定义工具包支持	59
设计阶段要使用的设置	60
修改已部署的支持	60
修改测试对象配置文件中的 Identification Property 属性	60
删除已部署的支持	61
第 2 部分: 教程: 学习创建 Web 自定义工具包支持	62
第 5 章: 学习为简单的自定义 Web 控件创建 UFT 支持	63
本课程的准备工作	64
计划对 Web 插件扩展性 Book 示例工具包的支持	64
Web 插件扩展性计划清单	68
开发工具包支持集	69
第 1 阶段: 创建工具包支持集	69
第 2 阶段: 将 WebExtSample 环境引入 UFT 中	70
设计工具包配置文件	70
设计测试对象配置文件	71
第 3 阶段: 使 UFT 标识、侦测和识别 Book 控件	72
部署和测试工具包支持集 (第 3 阶段)	74
第 4 阶段: 实现对 WebExtBook 测试对象方法的支持	76
部署和测试工具包支持集 (第 4 阶段)	77
第 5 阶段: 实现对 WebExtBook 标识属性的支持	78
部署和测试工具包支持集 (第 5 阶段)	79
第 6 阶段: 更改测试对象的名称	80
部署和测试工具包支持集 (第 6 阶段)	80
第 7 阶段: 实现防止识别子对象的筛选	81
部署和测试工具包支持集 (第 7 阶段)	81
第 8 阶段: 实现在 Book 控件上进行录制的支持	82
部署和测试工具包支持集 (第 8 阶段)	84
第 9 阶段: 实现对 AuthorName 值动态列表的支持	85
部署和测试工具包支持集 (第 9 阶段)	86
课程总结	87
接下来要执行的操作	87
第 6 章: 学习为复杂的自定义 Web 控件创建 UFT 支持	88
本课程的准备工作	89
计划对 Web 插件扩展性 示例 UsedBooks 控件的支持	89
Web 插件扩展性计划清单	93
开发工具包支持集	94
第 1 阶段: 扩展工具包支持集以支持其他控件	94

第 2 阶段：使 UFT 标识、侦测和识别 UsedBooks 控件	95
部署和测试工具包支持集（第 2 阶段）	96
第 3 阶段：实现对 WebExtUsedBooks 测试对象方法的支持	99
部署和测试工具包支持集（第 3 阶段）	101
第 4 阶段：实现对 WebExtUsedBooks 标识属性和测试对象名称的支持	101
部署和测试工具包支持集（第 4 阶段）	102
第 5 阶段：实现防止识别子对象的筛选	102
部署和测试工具包支持集（第 5 阶段）	103
第 6 阶段：实现在 UsedBooks 控件上进行录制的支持	103
部署和测试工具包支持集（第 6 阶段）	104
课程总结	105
接下来要执行的操作	105
 发送反馈	 106

欢迎使用 HP UFT Web 插件扩展性

HP UFT Web 插件扩展性是一个 SDK（软件开发工具包）程序包，使您能够测试使用第三方和自定义 Web 控件的应用程序，这些应用程序不受 UFT Web 插件的支持。

扩展性也可用于支持 HP Sprinter（这是 HP 为高效手动测试推出的一款解决方案），用于了解不受直接支持的 Web 对象。有关 Sprinter 的详细信息，请参见《HP Sprinter 用户指南》。

此章节包括：

- [关于 UFT Web 插件扩展性 SDK](#) 7
- [关于 UFT Web 插件扩展性开发人员指南](#) 8
- [其他联机资源](#) 9

关于 UFT Web 插件扩展性 SDK

安装 Extensibility Accelerator for HP Functional Testing 时，也会安装 UFT Web 插件扩展性 SDK，它将提供以下内容：

- API，可用于扩展 UFTWeb 插件，以支持自定义 Web 控件。
- 适用于 Microsoft Visual Studio 的
- “Web 插件扩展性 Help”，该文档包括以下内容：
 - 一本开发人员指南，包括阐述您为示例自定义控件开发支持的分步教程。
 - API Reference。
 - Toolkit Configuration Schema Help。
 - UFT Test Object Schema Help。

该帮助可从 **Start > All Programs > HP Software > HP Unified Functional Testing > Extensibility > Documentation** 获取

- 开发人员指南的打印机友好的 Adobe 可移植文档格式 (PDF) 版本（可 < **Extensibility Accelerator 安装文件夹>\help** 文件夹获取）。
- Web 插件扩展性工具包支持集，可针对以下 Web 2.0 环境扩展 UFT GUI 测试支持：
 - ASP.NET AJAX Control Toolkit
 - Google Web Toolkit
 - Yahoo User Interface
 - Dojo
- 示例 Web 工具包，其中包括名为 **Book** 和 **UsedBooksTable** 的控件（如果已安装 Extensibility Accelerator）。

在 Windows 8 操作系统中访问 UFT Web 插件扩展性

那些可在早期版本的 Windows 中从 **Start** 菜单访问的 UFT 文件，也可在 Windows 8 中从 **Start** 屏幕或 **Apps** 屏幕访问。

- **应用程序 (.exe 文件)**。在 Windows 8 中，您可以直接从 **Start** 屏幕访问 UFT 应用程序。例如：要启动 UFT，请双击 **HP Unified Functional Testing** 快捷方式。
- **非程序文件**。您可以从 **Apps** 屏幕访问文档。

备注: 正如在早期版本的 Windows 中一样，您可以通过按 **F1** 访问 UFT 中的上下文相关帮助，并从 **Help** 菜单中访问完整文档和外部链接。

关于 UFT Web 插件扩展性开发人员指南

本指南介绍如何设置 UFT Web 插件扩展性并使用它扩展针对第三方和自定义 Web 控件的 UFT GUI 测试支持。

本指南假定您熟悉 UFT 功能，且它应与“Web 插件扩展性帮助”中提供的以下文档一起使用 (**Start > All Programs > HP Software > HP Unified Functional Testing > Extensibility > Documentation > Web 插件扩展性 Help**):

- *API Reference*
- *Toolkit Configuration Schema Help*
- *Test Object Schema Help*

这些文档还应与 UFT 安装提供的以下 UFT 文档（可通过 UFT 主窗口中的 **Help > HP Unified Functional Testing Help** 获得）一起使用：

- 《HP Unified Functional Testing 用户指南》
- 《HP Unified Functional Testing 插件指南》的 Web 部分
- 《HP UFT Object Model Reference for GUI Testing》

备注:

本指南中的信息、示例和屏幕捕获专门集中在使用 UFT GUI 测试的方面。但是，使用 Web 插件扩展性开发的所有工具包支持集还可用于扩展 Sprinter 功能，使其支持那些未受到直接支持的 Web 控件。本指南中对 UFT 的所有引用均适用于 UFT 和 Sprinter。

此外，本指南中的很多信息同样适用于许多组件。

业务组件是 HP Business Process Testing 的一部分。有关详细信息，请参见《HP Unified Functional Testing 用户指南》和 HP Business Process Testing 《HP Business Process Testing 用户指南》。

在 Windows 8 中工作时，可从 **Apps** 屏幕访问 UFT 文档和其他文件。

为了使您能够更高效地搜索本指南中的特定主题或关键字，请使用以下选项：

- **AND、OR、NEAR 和 NOT** 逻辑运算符。可从搜索框旁的箭头获得。
- **搜索以前的结果**。可通过底部的搜索选项卡获得。
- **匹配相似的单词**。可通过底部的搜索选项卡获得。
- **只搜索标题**。可通过底部的搜索选项卡获得。

提示: 从搜索结果打开“帮助”页面，您搜索的字符串可能包括在折叠部分中。如果在页面上找不到字符串，则展开所有下拉部分，然后使用 Ctrl-F 组合键来搜索字符串。

要检查是否有最近更新，或验证是否正在使用文档的最新版本，请转到 HP 软件产品手册网站 (<http://h20230.www2.hp.com/selfsolve/manuals>)。

其他联机资源

您可使用以下其他联机资源：

资源	描述
疑难解答 & 知识库	HP 软件支持网站上的“Troubleshooting”页，可在其中搜索自助解决知识库。此网站的 URL 是 http://h20230.www2.hp.com/troubleshooting.jsp 。
HP 软件支持	HP 软件支持网站。通过此站点，可浏览自助解决知识库。您还可以在用户讨论论坛发帖和搜索信息、提交支持请求、下载修补程序和更新的文档等。此网站的 URL 是 www.hp.com/go/hpsoftwaresupport 。 <ul style="list-style-type: none">大多数提供支持的区域都要求您注册为 HP Passport 用户再登录，很多区域还要求用户提供支持合同。要查找有关访问级别的详细信息，请访问：http://h20230.www2.hp.com/new_access_levels.jsp要注册 HP Passport ID，请访问：http://h20229.www2.hp.com/passport-registration.html
HP 软件网站	HP 软件网站。此站点提供了有关 HP 软件产品的最新信息。包括新的软件发布、研讨会及展览会、客户支持和其他更多信息。此网站的 URL 是 www.hp.com/go/software

第 1 部分: 使用 Web 插件扩展性

第 1 章: 简介 UFT Web 插件扩展性

使用 UFT Web 插件扩展性可以为 UFT Web 插件不支持即取即用的第三方和自定义 Web 控件提供高级支持。

此章节包括:

- 关于 UFT Web 插件扩展性 12
- Extensibility Accelerator 12
- 标识 Web 插件扩展性的构建块 13
- 决定何时使用 Web 插件扩展性 13
- 了解如何实现 Web 插件扩展性 18
- Web 插件扩展性 示例 19

关于 UFT Web 插件扩展性

UFT Web 插件为许多常用 Web 控件提供内置支持。可以使用 UFT Web 插件扩展性扩展该支持并支持 UFT 识别其他 Web 控件。您也可以使用 Web 插件扩展性指示 UFT 将一组控件或 Web 元素识别为一个对象。例如, 如果一个搜索按钮和一个编辑框构成一个搜索框, 而您希望 UFT 将它们视为一个功能单元。

UFT 识别应用程序中的对象时, 会将该对象识别为属于特定测试对象类。这样可以确定 UFT 中表示应用程序对象的测试对象的标识属性和测试对象操作。

当 UFT 识别不带扩展性的网页上的控件时, 它将忽略特定类型的元素且不会创建测试对象来表示它们定义的控件。

对于不支持 Web 插件即取即用的其他 Web 控件, UFT 将创建一般 WebElement 测试对象。此类测试对象可能没有特定于正在测试的 Web 控件的某些特征。因此, 尝试使用此测试对象创建测试步骤时, 可用标识属性和测试对象操作可能不够。

例如, 将 UFT 识别的特殊类型表的自定义 Web 控件视为普通 WebElement。WebElement 测试对象不支持 **GetCellData** 操作。要创建从表中的单元格检索数据的测试步骤, 需要创建测试对象以表示表中的每个单元格, 然后创建访问相关单元格的测试对象以检索数据的复杂测试。

或者, 考虑在网页上使用一个由多个附加 html 表元素构成的表。您可以定义帮助 UFT 将此聚合表识别为一个对象的规则, 并提供相关测试对象方法。例如, 接收相对于聚合表的行数从而知道实际包含该行的 html 表的方法。

通过使用 Web 插件扩展性创建 Web 控件支持, 可以指示 UFT 将控件或控件组识别为属于特定测试对象类, 并可以指定测试对象的行为。另外还可以扩展 UFT 能够识别的可用测试对象类列表。这样就可以创建完全支持自定义 Web 控件特定行为的测试。

Extensibility Accelerator

越来越多的 Web 应用程序正在使用基于 Web 2.0 的工具包 (如 ASP.NET AJAX、Dojo、YahooUI、GWT 和 JQueryUI) 将动态和交互内容添加到其站点。这些工具包中的控件很复杂, 需要完善、灵活的测试功能。

UFT Web 插件扩展性使您可以扩展 Web 插件以自定义 UFT 如何识别不同类型的控件, 并与这些控件交互。到目前为止, Web 插件扩展性的使用包括手动开发和维护多个工具包支持集。

Extensibility Accelerator for HP Functional Testing 是类似于 Visual Studio 的 IDE, 便于设计、开发和部署这些支持集。它使所需扩展性 XML 文件的创建更快速、更轻松, 以便您可以集中精力开发使 UFT 能够使用自定义 Web 控件的 JavaScript 函数。

Extensibility Accelerator 用户界面可帮助您定义新的测试对象类、操作和属性。它还提供了点击式机制, 该机制可用于将定义的测试对象类映射到应用程序中的控件。Extensibility Accelerator 部署功能使您能够将新工具包支持集自动部署到 UFT 或对该支持集进行打包, 以便您可以与其他 UFT 用户共享该支持集。

可从 Unified Functional Testing 安装程序中的 **Add-in Extensibility and Web 2.0 Toolkits** 选项安装 Extensibility Accelerator for HP Functional Testing。

备注: 在此过程中, 会在浏览器中打开一个 html 页面。要成功完成安装, 此页面必须使用 Internet

Explorer 打开。

标识 Web 插件扩展性的构建块

以下部分描述组成 UFT 对象支持的主要元素。这些元素是 Web 插件扩展性的构建块。通过扩展其中一个或多个元素的现有支持，您可以开发所需的支持，以便创建有意义的且可维护的测试。

测试对象类

在 UFT 中，应用程序中的每个对象均由特定测试对象类的测试对象表示。测试对象类将确定 UFT 中可用于此测试对象的标识属性和测试对象方法的列表。此图标用于表示 UFT 中的测试对象，例如在 Keyword View 和 Object Repository 中，此图标也由测试对象类决定。

测试对象名称

UFT 识别对象时，将为页面上的每个测试对象创建唯一的名称。描述性测试对象名称使您可以区分相同类的测试对象，并且更容易在对象存储库和测试中标识对象。

默认情况下，测试对象使用其测试对象类的名称（如果页面上的同一个类有多个测试对象，则附加索引）。在许多情况下，这都不是自定义控件的理想名称。

测试对象的名称需要对 UFT 用户有意义，最好使用与工具包相关的术语。UFT 将在 Keyword View、Editor 和对象存储库中显示此名称。

测试对象标识属性

用于表示 Web 控件的测试对象类决定了测试对象可用的标识属性的列表。它还决定了这些标识属性中的哪些将用于唯一地标识控件，哪些标识属性可用于检查点和输出值（在“Checkpoint Properties”和“Output Value Properties”对话框中），以及检查点默认选中哪些标识属性。但是，标识属性的实际值将从 Web 控件派生。因此，由同一测试对象类的测试对象表示的几个 Web 控件对同一标识属性可能有不同的定义。

测试对象方法

用于表示 Web 控件的测试对象类决定了测试对象的测试对象方法列表。但是，对于由相同测试对象类的测试对象表示的不同 Web 控件，相同的测试对象方法可能运行方式不同。这是因为根据特定类型的 Web 控件，UFT 可能必须以不同方式执行测试对象方法。

录制事件

创建 UFT GUI 测试的方法之一是在应用程序上录制用户操作。启动录制会话时，UFT 将侦听在应用程序中的对象上发生的事件，并编写相应的测试步骤。用于表示 Web 控件的测试对象类决定了 UFT 可在 Web 控件上侦听哪些事件，以及每个事件发生时应录制哪些测试步骤。

决定何时使用 Web 插件扩展性

UFT Web 插件为大多数 Web 控件提供特定级别的支持，但忽略定义为 DIV 或 SPAN 元素的控件。为自定义 Web 控件扩展支持之前，先从 UFT 角度进行分析以查看此支持的界限，决定需要修改的支持元素。

当分析自定义 Web 控件时, 请使用“Object Spy”、“Keyword View”、“Editor”和“Record”选项。确保检查[标识 Web 插件扩展性的构建块](#)中描述的每个元素。

如果对现有对象标识或行为不满意, 则 Web 控件将成为 Web 插件扩展性的替补, 如以下情况中所述:

- UFT 可能会使用不符合您需求的测试对象类来识别控件。可使用 Web 插件扩展性指示 UFT 将自定义控件标识为属于您创建的新测试对象类。
- UFT 对控件使用的测试对象类可能符合要求, 但您可能要自定义某些测试对象方法或标识属性的行为。您可以使用 Web 插件扩展性新建扩展了 UFT 使用的测试对象类的测试对象类, 使用自己的自定义实现覆盖这些属性和方法的实现, 并指示 UFT 使用新测试对象类。
- 您可能会发现 UFT 为特定类型控件的所有对象生成的测试对象名称完全相同 (唯一计数器除外), 或者测试对象使用的名称并不能清晰地指示它所表示的控件。您可以使用 Web 插件扩展性新建扩展了 UFT 使用的测试对象类的测试对象类, 修改 UFT 对新类的测试对象的命名方式, 并指示 UFT 使用新测试对象类。
- UFT 可识别自定义控件内的各个子控件, 但无法正确识别主控件。例如, 如果主自定义控件是一个含编辑框 (其中包含小时数字和分钟数字) 的数字时钟, 则您可能希望对时间进行更改, 使其被视为时钟控件上的 **SetTime** 操作, 而不是视为编辑框上的 **Set** 操作。您可以使用 Web 插件扩展性修改对子控件上所发生事件的处理方式。
- 录制会话期间, 当在控件上执行操作或触发事件时, UFT 可能根本不会录制步骤, 或者可能会录制并非特定于控件行为的步骤。或者, UFT 可能会录制许多应视为单个操作的事件步骤, 或者可能在不应该录制步骤时录制了步骤。

您可以通过修改 Web 事件配置为每种现有 Web 对象配置要录制的事件。有关详细信息, 请参见《HP Unified Functional Testing 插件指南》中有关配置 Web 事件录制的部分。

如果 Web 事件配置不能充分支持您自定义录制, 例如, 要修改发生特定事件时 UFT 录制的步骤, 则可以使用 Web 插件扩展性。

分析示例自定义控件的默认 UFT 支持和扩展性选项

以下示例演示了您可以如何使用 Web 插件扩展性改进自定义控件的 UFT 支持。

下方显示的 Book 控件表示在 Internet 上售卖的书。此控件在 UFT 上不受特别支持。



此控件包含的信息包括书籍的标题、作者、新版的价格, 以及可购买旧版的最低价格。

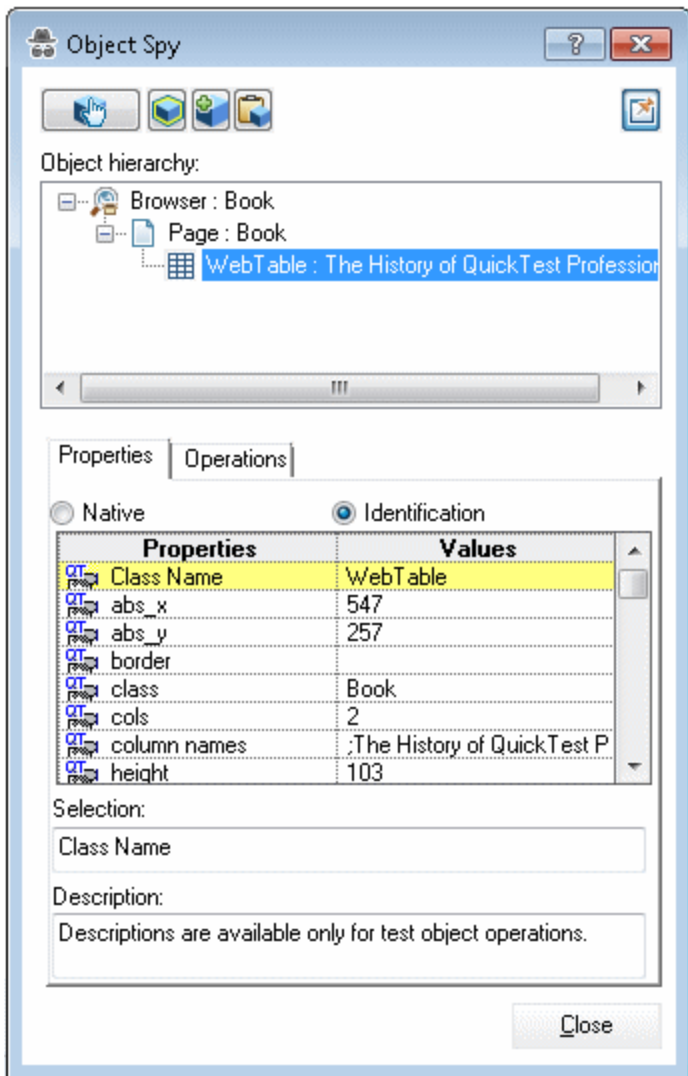
单击书籍标题可打开包含此书籍更多详细信息的页面。单击作者姓名可打开包含同一作者的书籍列表的页面。单击 **Used** 将打开“UsedBooks”页面, 列出此书籍的所有可用旧版及其价格。

Book 控件作为 Web 表实现, 如下所示:

```
<table class="Book">
  <tr>
    <td class="BookImageCell" rowspan="4">
```

```
<a href=".\\QtpHistory.htm">
  
</a>
</td>
<td class="BookCell">
  <a class="BookTitle" href=".\\QtpHistory.htm" > The History of QuickTest Professional</a>
</td>
</tr>
<tr>
  <td class="BookCell">
    By:<a href=".\\JaneDoe.htm">Jane Doe</a>,
<a href=".\\JohnDoe.htm">John Doe</a>
  </td>
</tr>
<tr>
  <td class="BookCell">
  </td>
</tr>
<tr>
  <td class="BookCell">
    New:<strong>59.99$</strong> &nbsp;<a href=".\\UsedBooks.htm">Used:</a> from
<strong>29.99$</strong>
  </td>
</tr>
</table>
```

因此，如果使用 Object Spy 指向此控件，则 UFT 会将此控件识别为 WebTable 对象，并根据书籍的标题命名。用于测试对象的图标是标准 WebTable 类图标。



如果没有实现相应支持却在 Book 控件上进行录制，则 Keyword View 如下所示：

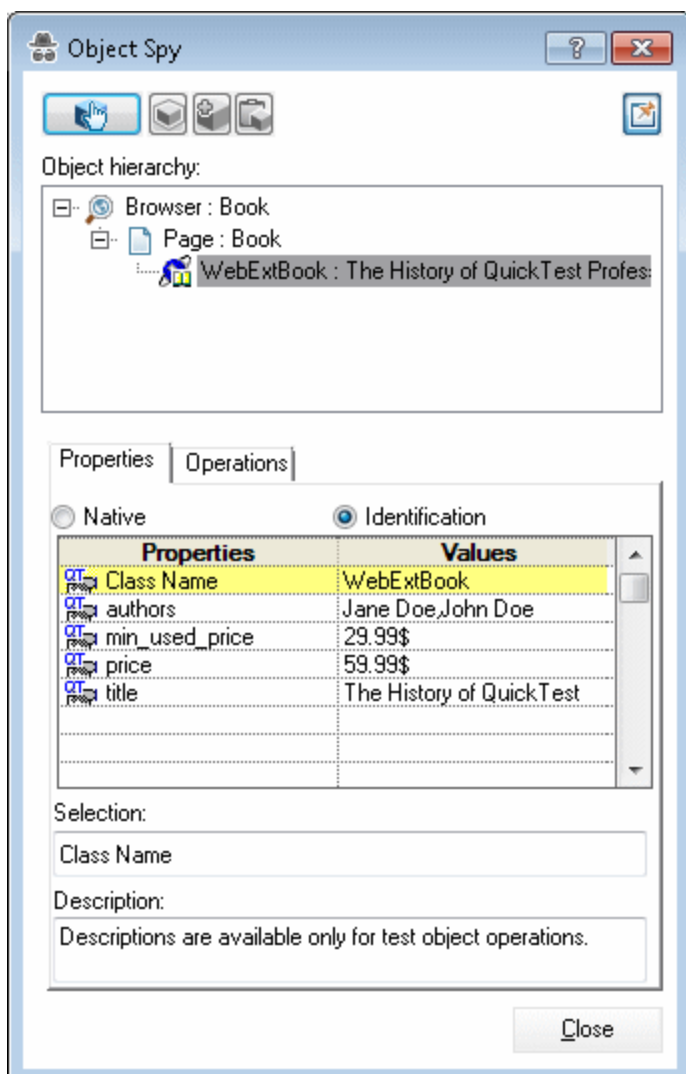
Item	Operation	Value	Documentation
▼ Action1			
▼ Book			
▼ Book			
The History of Quick...	Click		Click the "The History of QuickTest" link.
Book	Click		Click the "Book" image.
Jane Doe	Click		Click the "Jane Doe" link.
Used:	Click		Click the "Used:" link.

在 Editor 中，记录的测试如下所示：

```
Browser("Book").Page("Book").Link("The History of QuickTest").Click  
Browser("Book").Page("Book").Image("Book").Click  
Browser("Book").Page("Book").Link("Jane Doe").Click  
Browser("Book").Page("Book").Link("Used:").Click
```

注意，只录制简单的 **Click** 步骤，每个均属于 Book 控件中定义的不同对象。**Click** 操作在名称各异的 Web 链接测试对象或 Book 图像测试对象上单独录制。这些步骤在此控件的上下文中没什么帮助。

如果使用 Web 插件扩展性支持 Book 控件，则结果更有意义。UFT 将此控件识别为 **WebExtBook** 测试对象（仍根据书籍的标题命名）并使用不同的图标。标识属性包含相关信息，如 **authors** 和 **min_used_price**，提供所有书籍作者的姓名以及可购买旧版的最低价格。



当准备在控件上创建测试时，**Select**、**GoToAuthorPage** 和 **GoToUsedBookPage** 方法将受支持。这些方法可以录制，或者您可以在“Keyword View”的 **Operation** 列中手动进行选择。录制测试时，在 **Select** 步骤中产生的对书籍图像的单击和书籍标题的单击都会被录制。

还可以创建检查点来检查标识属性的值，例如 **authors**（提供包含所有书籍作者的字符串）。

在 Keyword View 中，通过记录与上面显示的测试相同的用户操作创建测试，如下所示：

Item	Operation	Value	Documentation
▼ Action1			
▼ Book			
▼ Book			
The History of QuickTest	Select		Select the "The History of QuickTest" book.
The History of QuickTest	Select		Select the "The History of QuickTest" book.
The History of QuickTest	GoToAuthorPage	"Jane Doe"	Open the Web page for "Jane Doe".
The History of QuickTest	GoToUsedBooksPage		Open the "The History of QuickTest" used books page.

在 Editor 中，此测试如下所示：

```
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").Check CheckPoint("The History of QuickTest")
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").Select
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").GoToAuthorPage "Jane Doe"
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").GoToUsedBooksPage
```

此测试更有意义，与 Book 控件的功能关系更密切。

了解如何实现 Web 插件扩展性

使用 Web 插件扩展性，可以实现对控件所有 UFT 功能的完全支持。您可以通过开发 **toolkit support set** 为一组控件（也称为 **toolkit** 或 **custom toolkit**）实现 Web 插件扩展性支持。

实现 Web 插件扩展性 包括以下阶段：

1. 计划工具包支持集。

- 决定组成自定义工具包的一组 Web 控件。
- 定义测试对象模型，方法是根据工具包支持的控件和业务流程决定要支持的测试对象和操作。

2. 开发工具包支持集。

Web 插件扩展性 工具包支持集包含以下文件：

- 一个 **test object configuration** 文件，描述了 UFT 对工具包使用的测试对象模型：
- 一个 **toolkit configuration** 文件，描述了表示工具包中每个控件的测试对象类以及 UFT 与每个控件交互的方式。
- 一个或多个包含 JavaScript 函数的文件，UFT 可调用这些函数在自定义控件上执行操作。

3. 部署和测试工具包支持集。

要部署工具包支持集并启用 UFT 支持控件，需要将创建的文件复制到 UFT 安装文件夹中的特定位置。

部署工具包支持集后，当打开 UFT 时，工具包的名称将在“Add-in Manager”中显示为“Web Add-in”节点下的子节点。如果选择了工具包的复选框，则 UFT 将使用开发的工具包支持集支持此工具包中的控件。

4. 增强工具包支持集。

为控件创建并测试基本 Web 插件扩展性 支持后，可使用一些较复杂的选项优化支持以增强工具包支持集。

有关每个阶段的详细信息，请参见：

- [计划工具包的 UFT 支持 \(第 20 页\)](#)
- [开发工具包支持 \(第 25 页\)](#)
- [部署工具包支持集 \(第 58 页\)](#)

开发 Web 插件扩展性 工具包支持集时，可以先创建简单且基本的支持集并将其部署到 UFT。这样 UFT 就可以正确识别控件，UFT 用户可以在控件上创建并运行测试。然后可通过启用较复杂的功能来增强支持，如筛选与控件一起识别的子对象，以及录制测试时更高级的事件处理方式。

通过学习教程：[学习创建 Web 自定义工具包支持 \(第 62 页\)](#)中的课程，可了解如何实际动手开发工具包支持集。

Web 插件扩展性 示例

适用于 HP Functional Testing 的 Web 2.0 功能包 提供了许多已完成的工具包支持集，从中您可以学习有关 Web 插件扩展性的更多知识。

- 在 UFT 计算机上，从该功能包安装 Unified Functional Testing 的 Web 2.0 Toolkit Support 时将在支持一些公共 Web 2.0 工具包的 Web 插件下安装子插件：ASP.NET AJAX、GWT、YahooUI 和 Dojo。这些插件是使用 Web 插件扩展性开发的工具包支持集。包含这些工具包支持集的文件可在 **<UFT 安装文件夹>\dat\Extensibility\Web** 文件夹和此文件夹的 **Toolkits** 子文件夹中找到。
- 安装 Extensibility Accelerator 时将安装包含上述 Web 2.0 工具包支持集的示例 Web 插件扩展性项目。

这些示例项目安装在 **%ALLUSERSPROFILE%\Documents**

ExtAccTool\Samples 文件夹中，还可从 Extensibility Accelerator 开始页访问。您可以在 Extensibility Accelerator 中打开这些项目，浏览这些文件、函数以及注释，从而进一步了解如何设计这些支持集。也可以修改这些示例项目，用它们做一些实验。示例项目的备份副本安装在 **<Extensibility Accelerator 安装文件夹>\Help\Samples** 文件夹中。

- 安装 Extensibility Accelerator 时还会安装示例 Book 应用程序以及包含此应用程序的工具包支持集的 Web 插件扩展性项目。
 - 此应用程序安装在 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** 中
 - 包含 Book 应用程序工具包支持集的 Web 插件扩展性项目安装在 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** 中。

在教程：[学习创建 Web 自定义工具包支持 \(第 62 页\)](#)中，您为 Book 应用程序创建类似的工具包支持集，并将其部署到 UFT（不使用 Extensibility Accelerator）。

《HP UFT Extensibility Accelerator for HP Functional Testing 用户指南》包含教程，在此教程中，使用 Extensibility Accelerator 创建对此控件的支持。

第 2 章: 计划工具包的 UFT 支持

开始创建自定义工具包支持之前，必须仔细计划支持。详细计划您希望 UFT 如何识别自定义控件，以便您能够正确构建自定义工具包支持的基本元素。

此章节包括：

- [关于计划工具包的 UFT 支持](#)21
- [准备为自定义工具包创建支持](#)21
- [决定工具包相关信息](#)21
- [决定每个自定义控件类型的支持信息](#)21
- [接下来要执行的操作](#)24

备注: 此章节假设您已熟悉 [简介 UFT Web 插件扩展性 \(第 11 页\)](#)中的概念。

关于计划工具包的 UFT 支持

扩展 UFT Web 插件支持以识别自定义控件是需要详细计划的过程。为协助您完成这一过程，此章节中的部分列出了几组与实现自定义工具包的支持及其控件相关的问题。创建工具包支持集时，可根据为这些问题提供的答案进行实现。

第一步是决定与自定义工具包相关的一般信息，然后是定义与您要支持的每个控件相关的特定信息。

准备为自定义工具包创建支持

开始为自定义 Web 控件计划支持之前，确保可完全访问这些控件并了解它们的工作方式。必须具有可供查看活动中的控件和用于实现这些控件的源代码的应用程序或网页。

您无需修改任何自定义控件的源代码就可以在 UFT 中支持这些控件，但需要熟悉它们。确保已知该控件包含的元素和属性，具有的 HTML 属性，以及可侦听的事件等。

决定工具包相关信息

计划工具包支持集时，将从决定一般工具包相关信息开始：

- 为要创建支持的工具包或环境提供一个唯一名称。
UFT 将在所有显示插件列表或受支持环境列表的对话框中显示环境的名称。例如，当 UFT 打开时，环境名称将作为“Add-in Manager”对话框中“Web Add-in”的子项显示，且 UFT 用户可指定是否为该环境加载支持。
- 决定此工具包支持集将支持的控件。
- 决定将包含为工具包支持集编写的 JavaScript 函数的文件。
 - 可为 JavaScript 函数指定一个默认文件，这些函数实现对不同 UFT 功能和不同测试对象类的支持。另外，可为不同功能和测试对象类的实现函数定义独立的文件。
 - 可以指定一个文件，其中包含可从其他文件调用的公共 JavaScript 函数。
- 决定是否使用整个工具包的一个 JavaScript 函数将测试对象类与自定义控件匹配。有关详细信息，请参见[使 UFT 标识用于自定义 Web 控件的测试对象类 \(第 40 页\)](#)。

设计工具包支持集时，在工具包配置文件中指定这些信息。有关详细信息，请参见[了解工具包配置文件 \(第 33 页\)](#)。

决定每个自定义控件类型的支持信息

为特定类型的控件计划自定义支持时，请认真考虑 UFT 如何识别此类控件——在 UFT GUI 测试中要使用哪种类型的测试对象来表示控件，要使用哪些标识属性和测试对象方法，等等。根据业务流程做出的这些决定可能要使用此类控件和用户期望在这些控件上执行的操作进行测试。

您可以运行包含自定义控件的应用程序，使用“Object Spy”、“Keyword View”和“Record”选项从 UFT 角度分析此控件。这样，您便可以了解 UFT 如何在没有自定义支持的情况下识别控件，并帮助您确定要更改的内容。

备注: Web 插件扩展性可用于在网页和框架中为 Web 控件创建支持。无法为网页或框架自身开发自定义支持。

要查看使用 UFT 分析自定义控件的示例，请参见[分析示例自定义控件的默认 UFT 支持和扩展性选项 \(第 14 页\)](#)。

此部分还包括：

- [了解 Web 插件扩展性 计划清单 \(第 22 页\)](#)
- [Web 插件扩展性 计划清单 \(第 23 页\)](#)

了解 Web 插件扩展性 计划清单

为特定类型的控件计划支持时，必须思考一系列问题。这些问题在下面进行了解答，可通过第 23 页上可打印的简易清单获得。

1. 确保您有权访问在已安装 UFT 的计算机上运行自定义控件的应用程序。
2. 是否存在可扩展以表示自定义控件的现有 Web 测试对象类？如果存在，是哪一个？如果不存在，则新测试对象类需要扩展 WebElement 类。
3. 如果新测试对象类扩展的是基本测试对象类而非 WebElement，则包含此类元素的控件是否通过基本测试对象类（也称为基本元素）正常表示？
 - 如果不是，则需要实现从 WebElement 不支持的基类继承的测试对象方法。另外，需要设计方法来返回 WebElement 不支持的所有测试类标识属性的值。
 - 如果控件包含基本元素，该元素是否为控件的根 Web 元素？
 - 如果是根元素，则 UFT 将对继承的测试对象方法和未覆盖的标识属性使用其内部实现。
 - 如果基本元素不是根元素，则需要实现返回基本元素的 JavaScript 函数。
4. 为表示 UFT GUI 测试中自定义控件的新测试对象类定义详细信息。

设计工具包支持集时，在测试对象配置文件中指定这些信息。有关详细信息，请参见[了解测试对象配置文件 \(第 28 页\)](#)。

- a. 指定测试对象类名称。
- b. 是否希望新测试对象类属于其他基本类型而非基类所属的基本类型？如果是，请指定该基本类型。（例如，如果新测试对象类扩展了 WebElement（其基本类型是 **object**），但 UFT 希望将此测试对象类与 **edit** 测试对象类组合。）
- c. 是否需要 UFT 对新的测试对象使用不同的图标？

如果是，请确保此图标文件可使用未压缩的 **.ico** 格式。推荐位置：**<UFT 安装文件夹>\dat\Extensibility\<UFT add-in name>\Toolkits\<Environment name>\Res。**
- d. 指定一个或多个可用于唯一标识控件的标识属性（除测试对象类之外）。
- e. 指定在为此类的测试对象生成步骤时显示在“Keyword View”和“Step Generator”中的默认测试对象方法。
- f. 是否要提供帮助文件？（在“Keyword View”或“Editor”中为此类的测试对象按下 **F1** 时

UFT 将其打开)

如果是, 请确保该帮助文件可使用 .chm 格式。推荐位置: <UFT 安装文件夹>\dat\Extensibility\<UFT add-in name>\Toolkits\<Environment name>\Help。

5. 决定如何设计此控件用于标识测试对象类的流程。
 - 是否要将流程限制为带有特定 HTML 标记的 HTML 元素? 如果是, 是哪种特定 HTML 标记? 确保包含所有可与自定义控件相关的 HTML 标记。
 - 将使用哪些 HTML 属性决定表示此类控件的测试对象类?
 - 是否需要创建不同的标识规则, 供控件在不同的浏览器上运行时使用?

有关详细信息, 请参见[使 UFT 标识用于自定义 Web 控件的测试对象类 \(第 40 页\)](#)。

6. 为表示控件的测试对象命名指定基础。
7. 要支持哪些标识属性? 在 UFT 中的“Checkpoint Properties”和“Output Value Properties”对话框中应显示哪些属性, 在此对话框中应默认选中哪些属性? 哪些标识属性可用于智能标识?
8. 要支持哪些测试对象方法? 指定方法的参数类型和名称, 以及方法除了返回代码是否还返回值。还可以选择指定, 在“Keyword View”或“Editor”中按 **F1** 时, 或在测试对象方法的“Step Generator”中单击“Operation Help”按钮时, UFT 将打开的帮助文件的位置。
9. 是否为任何测试对象方法参数动态提供可能值的列表? 哪些参数?
10. UFT 应使用控件识别哪些类型的子控件?
11. Object Spy 是否应显示此类的测试对象?
12. 是否通过录制为创建 UFT GUI 测试提供支持?

如果是, 请列出 UFT 录制会话期间要在自定义控件上录制的事件。
13. 决定需要在工具包配置文件中设计的支持部分以及需要 JavaScript 函数的部分:

Web 插件扩展性 计划清单

使用此清单计划对自定义控件的支持。

<input checked="" type="checkbox"/>	自定义控件支持计划清单	是否在工具包 XML 中指定?	JavaScript 函数是否支持?
<input type="checkbox"/>	此自定义控件的源位于:	N/A	N/A
<input type="checkbox"/>	指定新测试对象类扩展的 Web 测试对象基类: (默认值 —— WebElement)	N/A	N/A
<input type="checkbox"/>	基本测试对象类是否为 WebElement? 是/否 如果不是, 那么是否有基本元素 (与基本测试对象类匹配的元素)? 是/否 如果有基本元素, 那么是否需要 JavaScript 函数将其返回? 是/否	是/否	是/否
<input type="checkbox"/>	指定新 Web 测试对象类的详细信息:	N/A	N/A

<input checked="" type="checkbox"/>	自定义控件支持计划清单	是否在工具包 XML 中指定?	JavaScript 函数是否支持?
	<ul style="list-style-type: none"> • 测试对象类名称: • 一般类型 (可选): • 图标文件位置 (可选): • 描述的标识属性: • 默认的测试对象方法: • 帮助文件位置: 		
<input type="checkbox"/>	指定标识控件所使用的测试对象类的基础 (考虑不同的浏览器):	是/否	是/否
<input type="checkbox"/>	指定测试对象命名基础:	N/A	是
<input type="checkbox"/>	列出要支持的标识属性。标记应可供检查点和输出值使用的标识属性 (以及应在检查点中默认选中的标识属性) 和应该用于智能标识的标识属性 (如有):	是/否	是/否
<input type="checkbox"/>	列出要支持的测试对象方法 (如果需要, 包括参数、返回值、帮助文件位置和帮助 ID):	是/否	是/否
<input type="checkbox"/>	是否提供所有测试对象方法参数值的动态列表? 是/否 (默认) 如果提供, 则列出参数:	N/A	是/否
<input type="checkbox"/>	指定 UFT 应使用控件识别的子类型:	是/否	是/否
<input type="checkbox"/>	是否在 Object Spy 中显示此类的测试对象? 是 (默认值) /否	是/否	N/A
<input type="checkbox"/>	是否提供录制支持? 是/否 如果提供, 则列出可触发录制的事件:	是/否	是/否

接下来要执行的操作

完成自定义工具包支持的计划后, 可创建工具包支持集以支持每个计划的自定义工具包。[开发工具包支持 \(第 25 页\)](#)说明了如何开发工具包支持集。

第 3 章: 开发工具包支持

此章节说明了如何创建对自定义 Web 工具包的支持。另外还说明了必须为工具包支持集创建的文件, 这些文件的结构和内容, 以及如何使用这些文件在您的环境中支持各种 UFT 功能。

有关将工具包支持集文件存储在何处以激活您设计的支持的信息, 请参见[部署工具包支持集 \(第 58 页\)](#)。

此章节包括:

· 关于自定义工具包支持	26
· 创建自定义工具包支持集	27
· 了解测试对象配置文件	28
· 了解工具包配置文件	33
· 为工具包支持集设计 JavaScript 函数	34
· 使 UFT 标识用于自定义 Web 控件的测试对象类	40
· 在开发期间测试工具包支持集	44
· 记录和调试自定义支持	46
· 实现对测试对象方法的支持	49
· 实现对标识属性的支持	52
· 实现识别子控件的筛选	53
· 实现对录制的支持	55
· 疑难解答和限制 —— 开发支持	57

关于自定义工具包支持

通过为每个要支持的 Web 工具包创建 **toolkit support set** 实现 Web 插件扩展性。工具包支持集由 XML 配置文件和 JavaScript 函数组成。XML 配置文件定义为支持自定义 Web 控件而创建的测试对象类并将它们映射到控件。此外，它们还定义 UFT 如何在自定义控件上运行。JavaScript 函数在 UFT 和正在测试的应用程序之间提供接口，检索有关控件的信息并在控件上执行操作。

此章节描述了必须包含在自定义支持集中的各种文件、定义和函数。有关详细信息，请参见“Toolkit Configuration Schema Help”、“API Reference”、“JavaScript Function Reference”和“UFT Test Object Schema Help”（可通过“UFT Web 插件扩展性帮助”获得）。

Extensibility Accelerator 提供包含工具包支持集的示例项目，这些工具包支持集扩展了 UFT 对某些公共 Web 2.0 工具包的支持。这些是 ASPAjax、GWT、YahooUI 和 Dojo 支持集，在为 Unified Functional Testing 安装 Web 2.0 Toolkit Support 时安装在了 UFT 上。这些示例还包括测试对象模型的帮助文件，UFT 使用这些模型支持这些工具包中的测试控件。

示例项目安装在 `%ALLUSERSPROFILE%\Documents\`

`ExtAccTool\Samples` 中。可在 Extensibility Accelerator 中打开这些项目，浏览整个工具包支持集文件以查看 Web 插件扩展性的实现示例。此章节中的一些示例摘自这些文件。

备注: 实际开始创建工具包支持集之前，必须仔细计划。有关详细信息，请参见[计划工具包的 UFT 支持 \(第 20 页\)](#)。

开发不受浏览器影响的支持

许多 Web 控件在不同浏览器上的实现方式和运行方式皆不同，因为不同 DOM 属性在不同浏览器上的实现方式不同。

要启用对 Microsoft Internet Explorer、Google Chrome 和 Mozilla Firefox 的自定义支持，必须清楚以下几点：

- 如果不同浏览器中控件的 DOM 结构不同，则 UFT 可能需要使用不同的逻辑来标识控件在不同浏览器上识别对象或运行步骤时使用的测试对象类。有关如何定义特定于浏览器的标识条件的信息，请参见[使用条件元素 \(第 41 页\)](#)。
- 在编写 JavaScript 函数时，使用 jQuery 函数调用可创建在不同类型和版本的浏览器中流畅运行的代码。有关详细信息，请参见[创建不受浏览器影响的支持 \(第 36 页\)](#)。
- Web 插件扩展性提供了 **GetBrowserType** 和 **GetBrowserVersion** 实用程序方法，用于确定当前运行控件的浏览器的类型和版本。这使 JavaScript 代码可以根据当前运行控件的浏览器区别对待控件。

创建自定义工具包支持集

Web 插件扩展性 工具包支持集包含以下文件:

- 一个 **test object configuration** 文件, 描述了工具包的测试对象模型: UFT 用于在工具包中表示控件的测试对象类, 以及需要测试对象支持的标识属性和测试对象方法。有关详细信息, 请参见[了解测试对象配置文件 \(第 28 页\)](#)。

备注: UFT 在打开时 (从所有测试对象配置文件) 加载所有测试对象类定义, 而不考虑创建它们的自定义工具包。这样, 您便可以在支持不同的自定义工具包时使用相同的测试对象类定义。

- 一个 **toolkit configuration** 文件, 描述了表示工具包中每个控件的测试对象类以及 UFT 与每个控件交互的方式。有关详细信息, 请参见[了解工具包配置文件 \(第 33 页\)](#)。
- 一个或多个含有 JavaScript 函数的文件, UFT 可调用这些函数从自定义控件检索信息或在自定义控件上执行操作。
- (可选) 包含图标的图标文件, 在 UFT 中用于表示您定义的测试对象类, 以及描述这些测试对象类及其方法和属性的帮助文件。

[部署工具包支持集 \(第 58 页\)](#)描述工具包支持集所需的各种文件的名称和文件夹的存储结构。

创建自定义工具包支持集:

1. 选择一个唯一名称表示所创建支持的工具包或环境。

使用自定义工具包名称构成工具包文件夹和工具包配置文件的名称。该名称必须以字母开始并且只能包含字母数字字符和下划线。

提供唯一的工具包名称后, 单个 UFT 安装即可同时支持大量的自定义工具包支持集。因此, 建议不要使用 MyToolkit 之类的名称。

2. 创建工具包支持集的文件夹。

可为此文件夹选择任何方便的名称和位置。

3. 创建以下文件夹结构:

- **<工具包支持集文件夹>\Toolkits\<工具包环境名称>\JavaScript**
- **<工具包支持集文件夹>\Toolkits\<工具包环境名称>\Res**
- **<工具包支持集文件夹>\Toolkits\<工具包环境名称>\Help**

4. 在工具包支持集文件夹中, 创建一个名为 **<工具包环境名称>TestObjects.xml** 的文件。此为测试对象配置文件。

5. 在 **<工具包支持集文件夹>\Toolkits\<工具包环境名称>** 文件夹中, 创建一个名为 **<工具包环境名称>.xml** 的文件。此为工具包配置文件。

6. 在 **<工具包支持集文件夹>\Toolkits\<工具包环境名称>\JavaScript** 文件夹中, 创建一个或多个将包含您所设计的 JavaScript 函数的文件。

作为最佳实践, 应为每个测试对象类创建一个 JavaScript 文件。另外, 可以创建一个包含 JavaScript 函数的 JavaScript 文件, 其中的 JavaScript 函数调用自其他文件中的 JavaScript 函数。

7. 您可以使用 **<工具包支持集文件夹>\Toolkits\<工具包环境名称>\Res** 文件夹存储任何提供用于表示 UFT 中的测试对象类的图标。

8. 您可以使用 <工具包支持集文件夹>\Toolkits\<工具包环境名称>\Help 文件夹存储任何描述环境中的测试对象的帮助 (.chm) 文件。

了解测试对象配置文件

开发自定义工具包支持的第一个阶段是引入 UFT 要用于测试应用程序和控件的测试对象模型。为此，您需要在测试对象配置 XML 文件中定义测试对象模型。对于要扩展或修改 UFT 支持的每一类自定义控件，您均需创建测试对象类。

在测试对象配置 XML 中，您可定义测试对象类（例如，它们支持的测试对象方法及其标识属性等）。

您可为要定义的每个测试对象类创建一个 **ClassInfo** 元素。此外，您还可定义使用测试对象类的环境名称或自定义工具包名称（在 **TypeInformation** 元素的 **PackageName** 属性中），以及这些测试对象类要扩展的 UFT 插件（在 **TypeInformation** 元素的 **AddinName** 属性中）。

如果 UFT 打开时未加载相关插件，则 UFT 将不加载此 XML 中的信息。同样地，如果环境名称或自定义工具包名称显示在“Add-in Manager”对话框中，且其复选框未选中，则不加载此 XML 中的信息。

有关详细信息，请参见[UFT 如何加载测试对象配置 XML \(第 30 页\)](#)。

以下部分描述测试对象类定义中可包括的信息。

类名和基类

新测试对象类及其属性的名称（其中包括基类），即新测试对象类要扩展的测试对象类。新测试对象类将直接或间接扩展现有 WebUFT 测试对象类。基类可以是通过 UFT 交付的类，也可以是使用 Web 插件扩展性定义的类。

默认情况下，基类是 **WebElement**。

在所有支持 UFT 用户同步加载的环境中，测试对象类名必须是唯一的。例如，定义新测试对象类时，请勿使用现有 UFT 插件中测试对象类的名称，例如：**WebButton**、**WebEdit**、等。

备注:

- 测试对象类将继承基类的测试对象操作（方法和属性）、一般类型、默认操作和图标。不继承标识属性。
- 如果创建的测试对象类将扩展在其他工具包支持集中定义的测试对象类，则需在两个工具包支持集之间创建依赖关系。每次选择加载 UFT 插件管理器中的扩展工具包支持集时，还必须选择加载它所扩展的工具包支持集。

一般类型

新测试对象类的一般类型（如果希望新测试对象类属于非基类的其他一般类型。例如，如果新测试对象类将扩展 **WebElement**（其一般类型为 **object**），但您希望 UFT 使用 **edit** 测试对象类对此测试对象类进行分组。）

一般类型可在筛选对象时使用（例如，在步骤生成器的“Select Object for Step”对话框中和在将多个测试对象添加到对象存储库时）。一般类型还可在为“Keyword View”的“Documentation”列创建文档字符串时使用（如果这些字符串在测试对象配置文件中并未专门定义）。

测试对象操作

测试对象类的操作列表，其中包括每个操作的以下信息：

- 参数，其中包括参数类型（例如，String 或 Integer）、方向（In 或 Out）、参数是否为必需参数，如果不是则是是否为默认值。
- 此参数的可能值动态列表是否可从支持的控件中检索，是否会显示在“Keyword View”、编辑器和步骤生成器中。
- 操作描述（显示在对象侦测器中，并在“Keyword View”和步骤生成器中显示为工具提示）。
- “Documentation”字符串（显示在“Keyword View”的 **Documentation** 列和步骤生成器中）。
- 返回值类型。
- 在“Keyword View”或编辑器中针对测试对象操作按 **F1** 时，或者在步骤生成器中针对该操作单击 **Operation Help** 按钮时，将打开上下文相关帮助主题。该定义包括帮助文件路径以及文件内的相关帮助 ID。

默认操作

生成类对象使用的步骤时“Keyword View”和步骤生成器中默认选中的测试对象操作。

标识属性

测试对象类的标识属性列表。您还可定义：

- 用于对象描述的标识属性。

提示: 包括对象描述中用来加快 UFT 中对象标识的 **html tag** 属性。

- 用于**智能标识**的标识属性。（此信息仅在启用测试对象类的智能标识时相关。要启用智能标识，请使用 UFT 中的“Object Identification”对话框。）
- 可在检查点和输出值中使用的标识属性。
- 默认选中的检查点（在 UFT “Checkpoint Properties”对话框中）的标识属性。

图标文件

指测试对象类使用的图标文件路径。（可选。如果未定义，则使用基类的图标。）该文件可以是 .dll、.exe 或 .ico 文件。

帮助文件

在“Keyword View”或编辑器中，针对测试对象按 **F1** 时，将打开上下文相关帮助主题。该定义包括 .chm 帮助文件路径以及文件内的相关帮助 ID。

有关测试对象配置 XML 的结构和语法信息，请参见“UFT Test Object Schema Help”（可通过“UFT Web 插件扩展性帮助”获得）。

示例测试对象配置 XML

以下示例显示了 **WebExtUsedBooks** 测试对象定义的定义，此示例属于此指南中教程部分使用的 **WebExtSample** 工具包：

```
<ClassInfo BaseClassInfoName="WebTable" GenericTypeID="Table"
  DefaultOperationName="SelectBook" Name="WebExtUsedBooks">
  <IconInfo
    IconFile="INSTALLDIR\Dat\Extensibility\Web\Toolkits\
WebExtSample\Res\WebBookList.ico"/>
  <TypeInfo>
    <Operation ExposureLevel="CommonUsed" Name="SelectBook" PropertyType="Method">
      <Description>
        Selects the radio button for the specified book and clicks Select.
      </Description>
      <Documentation>
        <![CDATA[Select the radio button for the book with index %a1 and click Select.]]>
      </Documentation>
      <Argument Name="BookIndex" IsMandatory="true" Direction="In">
        <Type VariantType="Integer"/>
      </Argument>
    </Operation>
  </TypeInfo>
  <IdentificationProperties>
    <IdentificationProperty ForDefaultVerification="true" ForVerification="true"
ForDescription="true" Name="title"/>
  </IdentificationProperties>
</ClassInfo>
```

此示例显示 **WebExtUsedBooks** 测试对象类扩展 **WebTable** 测试对象类。**WebExtUsedBooks** 测试对象类使用 **WebBookList.ico** 图标文件，其默认测试对象方法是 **SelectBook**（具有一个整数型强制输入参数：**BookIndex**），此测试对象类具有一个额外标识属性：**title**。

为 **WebExtUsedBooks** 测试对象类定义一个标识属性：**title**。此标识属性用于对象描述，可用于检查点和输出值，在 UFT 的“Checkpoint Properties”对话框中默认选中。

另请参见：

- [UFT 如何加载测试对象配置 XML \(第 30 页\)](#)
- [扩展现有测试对象类 \(第 32 页\)](#)
- [为自定义测试对象类提供帮助文件 \(第 33 页\)](#)

UFT 如何加载测试对象配置 XML

每次运行 UFT，都会读取所有测试对象配置 XML 并将来自不同 XML 的每个测试对象类的信息合并为一个测试对象类定义。有关详细信息，请参见[了解 UFT 如何合并测试对象配置文件 \(第 31 页\)](#)。

测试对象配置文件中 **Identification Property** 元素的以下属性指定可以在 UFT 中修改的信息（通过使用“Object Identification”对话框）：**AssistivePropertyValue**、**ForAssistive**、**ForBaseSmartID**、**ForDescription**、**ForOptionalSmartID** 和 **OptionalSmartIDPropertyValue**。这些属性确定 UFT 中用于不同目的的标识属性列表。

因此，默认情况下，UFT 仅会从 XML 中读取一次这些属性的值，防止覆盖用户通过使用“Object Identification”对话框所做的任何更改。这样，用户便可以通过 UFT 持续不断地定义属性列表。有关详细信息，请参见[修改测试对象配置文件中的 Identification Property 属性 \(第 60 页\)](#)。

了解 UFT 如何合并测试对象配置文件

每次打开 UFT 时，它都会读取位于 <UFT 安装文件夹>\dat\Extensibility\<UFT 插件名称> 文件夹中的所有测试对象配置文件。然后，UFT 将根据每个测试对象配置文件的优先级，将每个测试对象类的信息从不同文件合并到单个测试对象类定义中。

在以下情况下，UFT 将忽略测试对象配置文件中的定义：

- **TypeInformation** 元素的 **Load** 属性设置为 false。
- 与测试对象配置文件相关的环境将显示在“Add-in Manager”对话框中，且 UFT 用户选择不加载该环境。

通过使用 **TypeInformation** 元素的 **Priority** 属性，可以定义每个测试对象配置文件的优先级。

如果测试对象配置文件的优先级高于现有类定义，则它将替代任何现有的测试对象类定义，其中包括内置 UFT 信息。因此，在更改测试对象配置文件的优先级之前，请务必注意任何将被替代的内置功能。

如果存在多个测试对象类定义，则 UFT 必须处理出现的任何冲突。以下部分描述了多个测试对象配置文件中定义了 **ClassInfo**、**ListOfValues** 和 **Operation** 元素时 UFT 所遵循的流程。某一特定测试对象类的所有 **IdentificationProperty** 元素必须仅通过一个测试对象配置文件进行定义。

ClassInfo 元素

- 如果 **ClassInfo** 元素通过优先级高于现有定义的测试对象配置文件进行定义，则该信息将附加到任何现有定义。如果不同文件中的 **ClassInfo** 定义发生冲突，则优先级较高文件中的定义将替代（替换）优先级较低文件中的信息。
- 如果 **ClassInfo** 元素通过优先级等于或低于现有定义的测试对象配置文件进行定义，则这一不同信息将附加到现有定义。如果不同文件中的 **ClassInfo** 定义发生冲突，则系统将忽略优先级较低文件中的定义。

ListOfValues 元素

- 如果不同文件中的 **ListOfValues** 定义发生冲突，则优先级较高文件中的定义将替代（替换）优先级较低文件中的信息（其定义不会合并）。
- 如果 **ListOfValues** 定义替代现有列表，则对于在同一测试对象配置文件中为类操作定义的类型为 **Enumeration** 的所有参数，系统将更新新列表。
- 如果 **ListOfValues** 通过优先级低于现有定义的配置文件加以定义，则系统将忽略优先级较低的定义。

Operation 元素

- 根据测试对象配置文件的优先级，可以添加、忽略或替代 **Operation** 元素定义。
- 如果 **Operation** 元素通过优先级高于现有定义的测试对象配置文件进行定义，则该操作将添加到现有的类定义。如果不同文件中的 **Operation** 定义发生冲突，则优先级较高文件中的定义将替代（替换）优先级较低文件中的定义（其定义不会合并）。

有关详细信息，请参见《HP UFT Test Object Schema Help》（可通过“Web 插件扩展性 SDK 帮助”获得）。

扩展现有测试对象类

当为控件提供部分支持的现有测试对象类需要进行一些修改时，例如，类中测试对象使用其他命名约定，或使用附加的或修改的测试对象方法，则可以新建测试对象类来表示此控件。新建测试对象类时，此测试对象类应基于现有测试对象类，并继承它的所有测试对象方法。

然后可通过定义和实现其他测试对象方法和标识属性来扩展此测试对象类的功能。此外，您还可以通过为它们提供可选实现来替代现有测试对象方法。在测试对象配置文件中定义全新的或更改的方法和属性，并使用 JavaScript 函数设计它们的实现。

要扩展现有测试对象类，需要在测试对象配置文件中为新测试对象类定义 **ClassInfo** 元素 **ClassInfo\BaseClassInfoName** 属性中的基本测试对象类的名称。这会声明新测试对象类支持基本测试对象类的所有测试对象方法，以及为新测试对象类定义的任何测试对象方法。

如何选择基类

选择测试对象类进行扩展时，请注意以下几点：

- 您的测试对象类将从测试对象类继承操作定义。请选择带有一系列与控件相关的操作功能的测试对象类。
- 要继承基类操作的实现，您的控件必须包含类型与基本测试对象类匹配的元素。

如何确保实现所有方法和属性

必须确保实现所有继承的测试对象方法，而不仅仅是声明。要实现这一点，一种方式是编写支持每个继承的测试对象方法的 JavaScript 函数。另一种更简单的方式是确保控件包含类型与基本测试对象类匹配的元素。此元素被称为 **base element**。UFT 可使用继承的测试对象方法的内部实现（不专门针对自定义控件实现）与基本元素通信。

另外，如果控件包含基本元素，则 UFT 在满足以下条件时使用基本测试对象类实现来检索标识属性值：

- 在测试对象配置文件中，将新测试对象类的标识属性名称定义为与基本测试对象类的属性名称相同。
- 不实现检索这些属性值的 JavaScript 函数。

当需要将基本元素返回到 UFT

如果此基本元素是您所支持的控件的根元素，则 UFT 将识别此基本元素，并使用您未实现的测试对象方法和标识属性的基本测试类实现。

如果此基本元素不是控件的根元素，则必须编写 JavaScript 函数将此基本元素返回到 UFT，然后在工具包配置文件中指定 JavaScript 函数的名称和位置。在工具包配置文件的 **Control\Settings** 元素中指定此信息。例如，如果控件是一种特殊的表，且被定义为包含表元素（UFT 通常使用 WebTable 元素表示）的 IDV 元素（UFT 通常会忽略），则可以创建扩展 WebTable 的 MyWebTable 测试对象类，并将 DIV 元素映射到 MyWebTable 测试对象类。

要返回基本元素，请在名为 **HPTable.js** 的文件中实现一个名为 **get_base_table** 的 JavaScript 函数。在 MyWebTable 控件元素的工具包配置 **Settings** 元素中，如下定义 **func_to_get_base_elem**：

```
<Control TestObjectClass=MyWebTable>
```



```
<Settings>  
  <variable name="default_imp_file" value="HPTable.js"/>  
  <variable name="func_to_get_base_elem" value="get_base_table"/>  
</Settings>  
</Control>
```

为自定义测试对象类提供帮助文件

作为 UFT 联机帮助的一部分，UFT 为定义的测试对象类提供对象模型参考。此参考旨在帮助 UFT 用户在自己的测试中使用 UFT 测试对象、方法和属性。此外，在“Keyword View”或编辑器中针对测试对象或测试对象方法按下 F1 之后，或者在步骤生成器中针对测试对象方法单击“Operation Help”按钮之后，UFT 会将对象模型参考打开到相关位置。

修改测试对象类或定义新测试对象类时，通过使用测试对象配置 XML 定义，您可以为定义的测试对象、方法和属性提供类似的 (.chm) 帮助文件。您可将这些帮助文件作为工具包支持集的一部分进行部署，并通知用户可在何处找到这些文件。您可以将帮助文件存储在任何方便的位置。例如，存储在 **<UFT 安装文件夹>\dat\Extensibility\<UFT 插件名称>\Toolkits\<环境名称>\help** 文件夹中。在测试对象配置 XML 中，可以为测试对象类和测试对象方法定义 **HelpInfo** 元素，从而指定帮助文件路径和文件内的相关帮助 ID。

警告: 帮助文件名不得与 **<UFT 安装文件夹>\help** 文件夹中提供的帮助文件的名称相同。

针对已定义 **HelpInfo** 元素的测试对象类或测试对象方法按下 F1 或“Operation Help”按钮之后，UFT 将打开您指定的上下文相关帮助主题。对于没有 **HelpInfo** 元素的已继承测试对象方法，UFT 将打开为基本测试对象类提供的帮助。

在您为测试对象类提供的帮助中，为每个测试对象类指定以下信息十分有用：

- 此测试对象类所扩展的基本测试对象类（其中包括介绍此帮助中不涵盖、但可在基本测试对象类相关帮助中找到的已继承测试对象方法相关描述注释）。
- 可用测试对象方法的列表（其中包括继承的方法）。
- 您定义的测试对象方法的描述，其中包括方法的用途、语法、参数、返回值以及任何其他相关信息。
- 可用标识属性的列表以及这些属性的描述。

了解工具包配置文件

要开始开发工具包支持集，需要定义一个基本工具包配置文件。可对比 **<Extensibility Accelerator 安装文件夹>\dat\Toolkit.xsd** 文件验证设计的工具包配置文件。（此文件也位于 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits** 文件夹中。）

在工具包配置 XML 中，对于计划用来支持工具包中控件的每个测试对象类，必须定义 **Control** 元素。每个 **Control** 元素均必须包含一个 **TestObjectClass** 属性，指定其要应用到的测试对象类的名称。

Control 元素包含在一个 **Controls** 元素内，表示整个工具包。

工具包配置文件必须提供相关信息，使 UFT 能够识别工具包中的每个控件要使用何种测试对象类。正如下面的部分所述，此信息可在工具包级别或控件级别中提供。

工具包配置 XML 还可包含其他信息。以下简单概述了此 XML 的可能内容。有关如何设计和使用工具包配置 XML 的详细信息, 将在本指南的后续部分介绍。

工具包特定的信息可包括:

- 工具包描述。
当用户在可用插件列表中选择工具包支持集的环境名称时, UFT 将在插件管理器中显示此描述。如果正在开发此工具包支持集以便分发, 请在此描述中提供 **Provided by** 子句, 指定相关的个人或公司。
- 工具包的优先级。当 UFT 尝试识别映射到自定义控件的测试对象类时, 它将按照其优先级在不同的工具包中搜索 (优先级最高的最先)。
- 标识函数 (也可包括含有该函数的文件的名称), 用于识别工具包中的每个控件要使用的测试对象类。
- 从中调用实现函数 (如果没有为测试对象类专门定义文件) 的默认文件的名称。
- UFT 将注入网页中的 JavaScript 库, 使您能够从设计的支持功能中调用库函数。

测试对象类元素可包括:

- HTML 标记和标识条件或标识函数 (也可包括含有该函数的文件的名称), 用于识别此测试对象类应表示的自定义控件。
- 从中调用实现函数 (如果没有为函数专门定义文件) 的默认文件的名称。
- 一个返回具有以下特点的元素的方法 (也可包括含有该函数的文件的名称): 其测试对象将实现已从基类中继承的、且未针对此控件实现的属性和测试对象方法。
- 多个用来实现此测试对象类的测试对象方法的方法 (也可包括含有此类函数的文件的名称)。UFT 将调用这些函数在控件上执行测试步骤。如果未定义函数, 则 UFT 将调用与测试对象方法同名的实现函数。
- 可从控件中检索标识属性的函数 (也可包括含有该函数的文件的名称)。如果没有定义函数, 则 UFT 将调用名为 **get_property_value** 的函数。
- 多个具有以下功能的元素: 指示是否使用默认 Web 事件配置来处理控件及其子控件上的事件, 且可指定 UFT 执行自定义事件的侦听和处理时应要运行的函数。
- 多个指示何时识别控件及其子控件的元素, 以及 (可选) 一个指定识别哪个子控件的函数。
- 多个指示何时在对象侦测器中显示此类的测试对象的元素。
- 一个可从控件中检索测试对象方法参数的可能值列表的函数 (也可包括含有该函数的文件的名称)。如果没有定义函数, 则 UFT 将调用名为 **get_list_of_values** 的函数。
- 一个可检索受支持工具包状态的函数 (也可包括含有此类函数的文件的名称), 以检查该工具包是否已在当前网页上完全加载。

备注: 为此文件中的 **Control** 元素规划顺序时, 请注意, UFT 将在搜索与控件匹配的测试对象类时遵循此顺序。它将使用第一个匹配的测试对象类。

有关此 XML 的结构和语法信息, 请参见 “Toolkit Configuration Schema Help” (可通过 “UFT Web 插件扩展性帮助” 获得)。

为工具包支持集设计 JavaScript 函数

作为工具包支持集的一部分, 设计 UFT 调用的 JavaScript 函数。 “UFT Web 插件扩展性 JavaScript Function Reference” (可通过 “Web 插件扩展性帮助” 获得) 描述您实现的 JavaScript 函数。

编写 JavaScript 函数时, 还要考虑以下部分的信息:

- [可使用的约定、实用程序对象和全局函数 \(第 35 页\)](#)
- [创建不受浏览器影响的支持 \(第 36 页\)](#)
- [全局变量范围 \(第 37 页\)](#)
- [编写全局 JavaScript 函数 \(第 37 页\)](#)
- [从外部 JavaScript 库调用函数 \(第 37 页\)](#)
- [处理异常 \(第 38 页\)](#)

可使用的约定、实用程序对象和全局函数

“**Web 插件扩展性 API**” 提供以下约定、常量、实用程序对象方法和全局 JavaScript 函数供您在 JavaScript 函数中使用:

- **_elem**。表示 UFT 当前处理的 Web 元素的令牌。在标识函数中, _elem 令牌表示识别的 HTML 元素。在所有其他上下文中, 它表示自定义控件的根元素。这些上下文包括:
 - 事件处理程序和事件注册函数。
 - 返回基本元素、属性值和可能值列表的函数。
 - 筛选函数。

例如, 以下 JavaScript 函数将返回当前 Web 元素 **id** 标识属性的值。

```
function get_property_value(prop)
{
  if (prop == "id")
  {
    return _elem.id;
  }
}
```

- **_util**。其方法可用于指示 UFT 执行不同操作的实用程序对象。有关详细信息, 请参见[全局 JavaScript 方法和实用程序方法 \(第 38 页\)](#)。
- **全局 JavaScript 函数**和在 **<Extensibility Accelerator 安装文件夹>\bin\PackagesToLoad** 文件夹内的 **common.js** 文件中定义的常量。(此文件也位于 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits** 文件夹中。)有关详细信息, 请参见[全局 JavaScript 方法和实用程序方法 \(第 38 页\)](#)。
- **窗口**。可用于访问您测试的 Web 浏览器窗口所对应的窗口对象。可使用此对象访问浏览器命名空间中的对象和函数, 并管理要创建支持的控件。

例如, 假设网页包含的全局 JavaScript 函数 **getTree()** 返回管理特定树控件的 **Tree** 对象。此 **Tree** 对象可能具有 **Collapse**、**Expand** 和 **Select** 方法。您可以编写以下 JavaScript 方法来支持运行 **Collapse** 测试对象方法:

```
function Collapse (treeBranch)
{
  var tree = window.getTree(_elem);
```

```
tree.Collapse(treeBranch);  
}
```

创建不受浏览器影响的支持

要使工具包支持集在不同类型和版本的浏览器上流畅运行，请在开发的扩展性代码中满足下面的一个或两个条件：

- 如果需要在不同浏览器中区别对待控件，则调用 **GetBrowserType** 或 **GetBrowserVersion** 实用程序方法以检索相关浏览器信息。有关详细信息，请参见[全局 JavaScript 方法和实用程序方法 \(第 38 页\)](#)。
- 使用外部不受浏览器影响的 JavaScript 函数库中的函数（如 **jQuery**），创建可在各种类型和版本的浏览器上流畅运行的代码。

jQuery（1.3.2 版）和 **jQuery.simulate** 库随 UFT 安装，这些库提供不受浏览器影响的函数。使用 **jQuery** 函数调用可检索信息和执行操作，使用 **jQuery.simulate** 函数调用可处理分发事件。

例如，如果需要在控件上执行单击操作，**jQuery** 可能会根据当前浏览器使用一行代码替换以各种方式发送事件的复杂代码块。

使用：

```
$_(elem).simulate("click");
```

而不是：

```
if ( _util.GetBrowserType()==QtpConstants.FireFox)  
{  
    var myEvent = window.document.createEvent("MouseEvents")  
    myEvent.initEvent("click", true, true)  
    _elem.dispatchEvent(myEvent)  
}  
else  
    _elem.click();
```

有关 **jQuery** JavaScript 库和 **jQuery.simulate.js** 的信息，请参见 <http://jquery.com/> 和 <http://code.google.com/p/jqueryjs/source/browse/trunk/plugins/simulate/jquery.simulate.js?r=6163>

有关使用 **jQuery** 库和其他外部 JavaScript 函数库的详细信息，请参见[从外部 JavaScript 库调用函数 \(第 37 页\)](#)。

- 在需要在 Mozilla Firefox 上运行的 JavaScript 函数中，需要显式地使用 **window** 对象来访问全局命名空间中的任何项。例如，当 **document** 仅在 Internet Explorer 上运行时，**window.document** 将可以在 Internet Explorer 和 Mozilla Firefox 中正确访问 **document** 对象。
- 针对 Chrome 进行开发时，请考虑以下事项：
 - 如果使用 **<body>** 标记标识某个控件，并且 **<body>** 是具有 **blank** 或 **about:blank SRC** 值的 **<iframe>** 的子项，则 UFT 只能在 Chrome 37 或更高版本中标识该控件。

- 在 Chrome 中, 如果自定义控件包含两个或两个以上同级控件, 则您必须在对象标识中包含这两个控件。例如, 这在您的自定义控件由一个编辑框及其旁边的提交按钮组成或一个复选框及其标签组成时相关。有关代码示例, 请参阅[示例: 使用同级控件标识对象 \(第 39 页\)](#)。
- 虽然设计的代码不受浏览器影响, 但每个浏览器使用不同的逻辑运行, 所以请在所有浏览器上验证您的脚本。

全局变量范围

当在 JavaScript 函数中定义全局变量时, 变量将存在于浏览器中运行的 JavaScript 引擎范围内。如果 UFT 在浏览器之前打开, 则 UFT 将只能连接到 JavaScript 引擎。因此, 在浏览器或 UFT 关闭之前, 可以访问全局 JavaScript 变量。

编写全局 JavaScript 函数

作为最佳实践, 应为每个测试对象类创建一个 JavaScript 文件。但是, 如果您要从多个测试对象的 JavaScript 函数调用同一个 JavaScript 函数, 则可以将这些全局函数存储在独立的文件中, 此文件将被用作工具包支持集的公共 JavaScript 文件。在工具包配置文件中的 **Controls\Settings\Variable** 元素中指定公共 JavaScript 文件的路径。有关详细信息, 请参见“Toolkit Configuration Schema Help”(可通过“UFT Web 插件扩展性帮助”获得)。

从外部 JavaScript 库调用函数

在设计 JavaScript 函数中, 可调用 UFT 测试的网页中包含的任何 JavaScript 函数。

请记住, 如果已加载工具包支持集, UFT 可能会调用在浏览器中打开的任何网页上的代码。例如, 当识别网页上的对象时, 即使此页面不含有任何控件, 也不包含您的函数库, UFT 也可能会调用您设计的标识函数。

要支持在不包含您自己的 JavaScript 库的页面上使用这些库, 您可以指示 UFT 将 JavaScript 库注入到每一页中。为此, 请在工具包配置文件的 **JSLibrariesToInject\JSLibrary** 元素中指定到该库的文件系统路径。有关详细信息, 请参见“Toolkit Configuration Schema Help”(可通过“UFT Web 插件扩展性帮助”获得)。

当 UFT 在已加载工具包支持集的情况下打开时, UFT 将指定的 JavaScript 库注入到在浏览器中打开的每个网页中。

建议设计您自己的 JavaScript 库, 仅当网页中未包含此库时才加载。

将工具包支持集部署到 UFT 时, 确保参考的 JavaScript 库位于您在工具包配置文件中指定的路径中。

使用 UFT 提供的 jQuery 库

jQuery (1.3.2 版) 和 **jQuery.simulate** 库都随 UFT 安装。这些库位于: **<UFT 安装文件夹>\bin\JSFiles** 中

要从 JavaScript 函数调用 **jQuery** 库函数, 请在工具包配置文件中指定库的位置:

```
<Controls>
  <JSLibrariesToInject>
    <JSLibrary path="INSTALLDIR\bin\JSFiles\jQuery-1.3.2.js" />
    <JSLibrary path="INSTALLDIR\bin\JSFiles\jQuery.simulate.js" />
  </JSLibrariesToInject>
```

```
</Controls>
```

这些 jQuery 库被设计为在页面已含有某版本的库时，不再加载这些库。

处理异常

如果设计的 JavaScript 函数抛出异常，则 UFT 将显示运行时错误消息，且测试步骤将失败。此异常提供的消息将被添加到失败步骤的测试报告详细信息中。

如果要将消息添加到失败步骤的运行结果中，而不引起运行时错误，请从 JavaScript 函数调用 `_util.Report` 方法。

备注: 如果包含在异常中的消息是简单的字符串，则 UFT 显示的运行时错误将包含抛出异常的代码行。如果消息使用格式包含字符串中的变量，则运行时错误将不包含代码行。

全局 JavaScript 方法和实用程序方法

作为工具包支持集的一部分，设计 UFT 调用的 JavaScript 函数。JavaScript 函数可调用 UFT 提供的全局方法和实用程序方法。全局方法在 `common.js` 文件中作为 JavaScript 函数提供。实用程序方法由 `_util` 对象公开。

`_util` 实用程序对象公开了以下可在 JavaScript 函数中调用的方法：

- **Alert (*message*)**。打开显示指定消息的模型消息框。
- **GetBrowserType/GetBrowserVersion ()**。返回当前运行控件的浏览器的名称或版本。
- **LogLine (*text, severity, id, category*)**。向 Microsoft Windows 事件日志添加条目（可使用事件查看器查看），或在 Chrome 控制台中向 Chrome 添加条目。使用此方法有助于分析支持集的性能或调试其功能。
有关使用事件查看器调试工具包支持集的详细信息，请参见[使用 Microsoft Windows 事件日志 \(第 46 页\)](#)。
- **Record (*method, arrParams, delay*)**。将步骤添加到测试，并将测试对象添加到对象存储库（如果测试对象尚未位于对象存储库中）。使用 JavaScript 函数中的此方法在控件上发生事件后记录测试中的步骤。有关开发录制支持的详细信息，请参见[实现对录制的支持 \(第 55 页\)](#)。
- **RegisterForEvent (*element, eventName, handler, [HandlerParam]*)**。注册以侦听特定 Web 元素上的特定事件。使用 JavaScript 函数中控制侦听事件的方法来支持录制。还可从事件处理程序调用此方法。有关开发录制支持的详细信息，请参见[实现对录制的支持 \(第 55 页\)](#)。
- **Report (*status, method, arrParams, details*)**。将有关步骤结果的信息添加到运行结果。使用 JavaScript 函数中的此方法在控件上执行步骤。有关开发运行测试支持的详细信息，请参见[实现对测试对象方法的支持 \(第 49 页\)](#)。
- **Wait (*milliseconds*)**。将执行挂起指定的毫秒数。

`common.js` 中最常用的函数是：

- **getFrameElement**。返回包含当前元素的框架元素。
- **toSafeArray(inArray)**。将 JavaScript Array 对象转换为 COM SafeArray。
- **trim(inString)**。返回去除首尾空格后的字符串。

`common.js` 文件还包含带有预设值，可在代码中用作常量的变量。

common.js 文件位于 **<Extensibility Accelerator 安装文件夹>\bin\PackagesToLoad** 文件夹和 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits** 文件夹中。

有关这些方法和常量的详细信息，请参见“UFT Web Add-in Extensibility API Reference”（可通过“Web 插件扩展性帮助”获得）。

示例：使用同级控件标识对象

在 Chrome 中，如果自定义控件包含两个或两个以上同级控件，则您必须在对象标识中包含这两个控件。

此示例演示在包含 input 和 label 的自定义控件中如何执行上述操作：

```
<input type="checkbox" role="hidden-checkbox-input" class="ui-helper-hidden"/><label for="check" class="ui-toggle-button" role="button">Toggle</label>
```

在其他浏览器中，您可以通过仅指定 **input** 标记来标识控件：

SampleToolkit.xml:

```
<Control TestObjectClass="SampleToggleButton">
...
<Identification type="javascript" function="isSampleToggleButton">
  <Browser name="*">
    <HTMLTags>
      <Tag name="input" />
    </HTMLTags>
  </Browser>
</Identification>
...
</Control>
```

SampleToggleButton.js:

```
function isSampleToggleButton ()
{
  return _elem.getAttribute("role") === "hidden-checkbox-input";
}
```

在 Chrome 中，您应同时使用 input 和 label 标记标识控件。

要实现此操作，请向 **SampleToolkit.xml** 再添加一个标记：

```
<Control TestObjectClass="SampleToggleButton ">
...
<Identification type="javascript" function=" isSampleToggleButton ">
  <Browser name="*">
```

```
<HTMLTags>
  <Tag name="input" />
  <Tag name="label" />
</HTMLTags>
</Browser>
</Identification>
...
</Control>
```

然后对 **SampleToggleButton.js** 函数进行相应更新:

```
function isSampleToggleButton ()
{
  if (_elem.getAttribute("role") === "hidden-checkbox-input" && _elem.tagName.toUpperCase() === "INPUT")
    return true;

  if (_elem.tagName.toUpperCase() === "LABEL" && _elem.getAttribute("role") == "button")
  {
    var sibling = _elem.previousSibling;
    if (sibling && sibling.tagName.toUpperCase() === "INPUT" && sibling.getAttribute("role") === "hidden-checkbox-input")
    {
      return sibling; // use the sibling input element to replace label to serve as _elem.
    }
  }

  return false;
}
```

使 UFT 标识用于自定义 Web 控件的测试对象类

定义希望 UFT 用于表示控件的测试对象类后, 需要将每种控件映射到特定的测试对象类。

此标识可使用 JavaScript 函数或条件元素执行, 检查控件的属性并决定应表示此控件的测试对象类。要提高 UFT 识别自定义控件和在控件上运行步骤的性能, 在定义标识元素时请尽量不要使用 JavaScript 函数调用。

可将自定义控件的标识过程限制为带有指定 HTML 标记的 HTML 元素。这样可进一步提高性能, 比定义检查 **tagName** 属性的条件更有效。

提供启用 UFT 的信息来标识用于工具包配置文件 **Controls\CommonIdentification** 元素或 **Control\Identification** 元素中各种控件的测试对象类。在这些元素中, 您可以定义以下内容:

- **每个测试对象类的一组 HTML 标记。** UFT 处理每个控件时, 仅对带有指定 HTML 标记的 HTML 元素, 根据 **Identification** 元素中的定义继续标识。

- **每个测试对象类的一组条件。** UFT 处理每个控件时，UFT 将根据在每个 **Control** 元素中定义的条件检查该控件的 HTML 属性，并查找控件满足其条件的第一个属性。

备注: 您可以定义 UFT 在各种浏览器上运行时使用的各种条件和 HTML 标记。

- **整个工具包的一个 JavaScript 标识函数。** UFT 处理每个控件时，UFT 将调用此函数来检查控件的属性并返回相应测试对象类的名称。

备注: 如果在工具包级别指定标识函数（在 **Controls\CommonIdentification** 元素中），并且在测试对象类级别指定条件（在 **Control\Identification** 元素中），则 UFT 在调用 JavaScript 函数之前将检查这些条件以避免不必要的调用。

- **每个测试对象类的一个 JavaScript 标识函数。** UFT 处理每个控件时，UFT 将按照在工具包配置文件中定义测试对象的顺序调用这些 JavaScript 函数。每个函数将检查定义它的测试对象类是否应表示该控件。UFT 将使用函数返回 **true** 的第一个测试对象类。

应尽量避免使用此标识方法，因为对于每个控件，此方法需要调用许多 JavaScript 函数，从而显著影响标识效率。

备注: 如果在工具包级别指定标识函数（在 **Controls\CommonIdentification** 元素中），则 UFT 不会调用在测试对象类级别指定的任何标识函数（在 **Control\Identification** 元素中）。

- 还可以组合使用条件与 JavaScript 函数，根据控件属性定义条件来限制对所定义 JavaScript 函数的调用。有关详细信息，请参见[使用条件元素 \(第 41 页\)](#)。

有关为 Web 插件扩展性编写 JavaScript 函数的详细信息，请参见[为工具包支持集设计 JavaScript 函数 \(第 34 页\)](#)。

有关标识元素的结构和语法信息，请参见“Toolkit Configuration Schema Help”，可通过“UFT Web 插件扩展性帮助”获得。

使 UFT 标识用于自定义控件的测试对象类后，便可以测试工具包支持集的基本功能。有关详细信息，请参见[在开发期间测试工具包支持集 \(第 44 页\)](#)。

使用条件元素

可在为测试对象类定义的 **Control\Identification\Browser** 元素中定义 **Conditions** 元素。这样 UFT 便可以根据控件的属性标识通过此测试对象类表示的控件，而不需要调用标识函数。

或者，也可以定义标识函数（针对每个测试对象类或整个工具包），使用条件来限制 UFT 调用该函数的次数。要实现这一点，可以定义仅在控件属性符合特定条件时调用标识函数。

可定义不同组条件供 UFT 在不同浏览器上运行时使用，也可以定义一组条件并指定该条件与所有浏览器相关。有关详细信息，请参见“Toolkit Configuration Schema Help”（可通过“UFT Web 插件扩展性帮助”获得）中的 **Browser** 元素描述。

将一组 **Condition** 元素通过 **and** 或 **or** 逻辑组合成 **Conditions**。每个 **Condition** 元素指定 HTML 控件的某个属性，以及该属性的期望值。如果控件的属性值与指定的值匹配则满足该条件（您可以在条件中指定此值是必须等于指定值还是不等于指定值）。您可以嵌套 **Conditions** 元素来创建复杂的逻辑。

针对每个顶级 **Conditions** 元素的每组条件，指定以下某种类型，指示 UFT 在控件属性符合元素内条件时如何处理此控件。

- **IdentifyIfPropMatch**。如果满足此元素内的条件，则使用当前测试对象类表示此控件。
- **CallIDFuncIfPropMatch**。如果满足此元素内的条件，则调用标识函数来检查此控件。另外，不要使用当前测试对象类表示此控件。
- **SkipIfPropMatch**。如果满足此元素内的条件，则不使用当前测试对象类表示此控件。

如果定义了标识函数，且检查条件不支持 UFT 决定是否使用当前测试对象类表示控件，则将调用标识函数。

- 如果定义了 **IdentifyIfPropMatch** 类型的条件元素，则将在其他类型之前检查此条件。
- 如果同时定义了 **CallIDFuncIfPropMatch** 和 **IdentifyIfPropMatch** 条件元素，则仅当不满足 **IdentifyIfPropMatch** 类型的条件时才检查 **CallIDFuncIfPropMatch** 类型的条件。
- 如果同时定义了 **SkipIfPropMatch** 和 **IdentifyIfPropMatch** 条件元素，则仅当不满足 **IdentifyIfPropMatch** 类型的条件时才检查 **SkipIfPropMatch** 类型的条件。
- 如果定义了 **CallIDFuncIfPropMatch** 类型的条件元素，则不检查 **SkipIfPropMatch** 类型的条件元素。
- 如果嵌套 **Conditions** 元素，则忽略嵌套元素的类型属性。

了解 UFT 如何执行标识

对于所处理的每个自定义控件，UFT 执行以下步骤来决定自定义控件使用的测试对象类：

1. 如果标识函数是在工具包级别（在工具包配置文件的 **CommonIdentification** 元素中）定义的，则首先检查所有 **Control** 元素中的 **Identification\Conditions** 元素，以决定是否可避免调用此函数。

对于每个 **Control** 元素：

- a. 如果定义了 **IdentifyIfPropMatch** 条件，且自定义控件的属性满足条件的要求，则使用此 **Control** 元素的测试对象类。否则，继续执行以下步骤。
- b. 如果定义了 **CallIDFuncIfPropMatch** 条件，且自定义控件的属性不满足条件的要求，则不使用此 **Control** 元素的测试对象类。
如果此 **Control** 元素不包含 **CallIDFuncIfPropMatch** 条件，则继续执行以下步骤。
- c. 如果定义了 **SkipIfPropMatch** 条件，且自定义控件的属性满足条件的要求，则不使用此 **Control** 元素的测试对象类。

检查所有条件后，请调用工具包级别的标识函数，除非找到匹配的 **Control** 元素或 **Control** 元素的 **all** 中的条件指示不使用此 **Control** 元素的测试对象类。

标识函数返回测试对象类的名称用于表示此自定义控件。

2. 如果在工具包级别未定义任何标识函数，则逐一检查 **Control** 元素。每个 **Control** 元素针对一个测试对象类定义。

对于每个 **Control** 元素，请执行以下步骤决定自定义控件是否使用此元素的测试对象类。使用第一个匹配的 **Control** 元素：

- a. 如果定义了 **IdentifyIfPropMatch** 条件，且自定义控件的属性满足条件的要求，则使用此 **Control** 元素的测试对象类。否则，继续执行以下步骤。
- b. 如果定义了 **CallIDFuncIfPropMatch** 条件，且自定义控件的属性满足条件的要求，则继续执行步骤 d。如果自定义控件的属性不满足条件的要求，则不使用此 **Control** 元素的测试对象类。
如果此 **Control** 元素不包含 **CallIDFuncIfPropMatch** 条件，则继续执行以下步骤。
- c. 如果定义了 **SkipIfPropMatch** 条件，且自定义控件的属性满足条件的要求，则不使用此

Control 元素的测试对象类。否则, 继续执行以下步骤。

- d. 如果标识函数在此 **Control** 元素内定义, 则运行此函数以决定自定义控件是否使用此元素的测试对象类。此函数将返回 `true` 或 `false`。如果此 **Control** 元素不包含标识函数, 则使用此 **Control** 元素的测试对象类。

了解 HTML 属性如何比较条件

将条件中指定的 HTML 属性值与指定期望值进行比较时, 将遵守以下规则:

- 字符串值比较不区分大小写。
- 可以指示 UFT 将期望值中提供的字符串视为常规表达式。在这种情况下, 将 `is_reg_exp` 属性设置为 `true`。
- 数字值比较将仅比较数字值。
- 比较布尔值时, 值 `true`、`1` 和 `yes` 都将认为是 `true`。值 `false`、`0` 和 `no` 都将认为是 `false`。
- 如果检查的 HTML 属性返回一个对象, 则将 `valid` 和 `null` 用作期望值。如果成功返回一个对象, 则认为属性 `valid`, 如果未返回对象, 则为 `null`。
- 如果要检查属性是否未包含特定值, 则将 **Condition** 元素中的 `equal` 属性设置为 `false`。
- 如果希望 UFT 在评估条件之前删除属性值和期望值中的首尾空格, 则将 **Condition** 元素中的 `trim` 属性设置为 `true`。

以下示例说明了可使用 **Conditions** 元素的不同方式。

示例 1: 标识函数和条件

如果自定义控件定义了 **CalendarBehavior** 或 **PopupBehavior** 属性, 则 UFT 将调用 `isCalendar` JavaScript 函数以决定是否使用 **TSGCalendar** 测试对象来表示控件。否则, 将继续执行工具包配置文件中的下一个 **Control** 元素, 并检查该元素是否与自定义控件匹配。

```
<Control TestObjectClass="TSGCalendar">
  <Identification type="javascript" function="isCalendar">
    <Browser name="*">
      <Conditions type="CallIDFuncIfPropMatch" logic="or">
        <Condition prop_name="CalendarBehavior" expected_value="valid"/>
        <Condition prop_name="PopupBehavior" expected_value="valid"/>
      </Conditions>
    </Browser>
  </Identification>
</Control>
```

示例 2: 仅条件

如果自定义控件定义了 **AccordionBehavior** 属性, 则 UFT 使用 **TSGAccordion** 测试对象来表示此控件。否则, 将继续执行工具包配置文件中的下一个 **Control** 元素, 并检查该元素是否与自定义控件匹配。

```
<Control TestObjectClass="TSGAccordion">
  <Identification>
    <Browser name="*">
```

```
<Conditions type="IdentifyIfPropMatch" logic="and">
  <Condition prop_name="AccordionBehavior" expected_value="valid"/>
</Conditions>
</Browser>
</Identification>
</Control>
```

示例 3: 嵌套条件节点

如果自定义控件的 **className** 属性值等于 **tsg**, 同时控件的 **id** 属性值等于 **tsg_table** 或控件的 **myType** 属性值等于 **table**, 则 UFT 使用 **TSGTable** 测试对象来表示此控件。

如果自定义控件的 **className** 属性值等于 **tsg**, 但控件的 **id** 属性值不等于 **tsg_table**, **myType** 属性值不等于 **table**, 则 UFT 调用 **isTable** JavaScript 函数以决定是否使用 **TSGTable** 测试对象。

如果自定义控件的 **className** 属性值不等于 **tsg**, UFT 将在不调用 **isTable** 标识函数的情况下继续执行工具包配置文件中的下一个 **Control** 元素。

```
<Control TestObjectClass="TSGTable">
  <Identification type="javascript" function="isTable">
    <Browser name="*">
      <Conditions type="IdentifyIfPropMatch" logic="and">
        <Condition prop_name="className" expected_value="tsg"/>
        <Conditions logic="or">
          <Condition prop_name="id" expected_value="tsg_table"/>
          <Condition prop_name="myType" expected_value="table"/>
        </Conditions>
      </Conditions>
      <Conditions type="SkipIfPropMatch" logic="and">
        <Condition prop_name="className" expected_value="tsg" equal="false"/>
      </Conditions>
    </Browser>
  </Identification>
</Control>
```

有关 **Conditions** 和 **Condition** 元素的结构和语法信息, 请参见“Toolkit Configuration Schema Help” (可通过“UFT Web 插件扩展性帮助”获得)。

在开发期间测试工具包支持集

在测试对象配置文件中定义测试对象模型, 并定义支持 UFT 标识不同控件使用的测试对象类的基本工具包配置文件后, 可以测试工具包支持集的现有功能。为此, 您需要部署工具包支持集并测试 UFT 如何与环境中的控件交互。

完成开发环境支持的其他阶段后, 可再次部署工具包支持集并测试 UFT 和控件间交互的其他方面 (识别测试对象、运行测试等)。

如果设计的支持用于在不同浏览器上运行, 请确保对其进行相应测试。

在定义测试对象类并将其映射到自定义 Web 元素后，要测试工具包支持集：


1. 在测试对象配置文件中，将 **TypeInformation** 元素的 **DevelopmentMode** 属性设置为 **true**，以确保 UFT 从每次打开的文件中读取所有测试对象类信息。完成工具包支持集的开发后，确保将此属性设置为 **false**。
2. 通过将支持集的文件复制到 UFT 安装文件夹中的正确位置，在 UFT 计算机上部署工具包支持集。有关详细信息，请参见[部署工具包支持集 \(第 58 页\)](#)。
3. 打开 UFT 并打开 GUI 测试。确保您为工具包支持集定义的环境名称作为 Web 插件的子插件显示在“Add-in Manager”对话框中。（如果在打开 UFT 时未打开“Add-in Manager”对话框，请参见《HP Unified Functional Testing 插件指南》中的说明。）

提示: 显示在插件管理器中的环境名称与工具包配置文件名相同。如果环境在插件管理器中显示不正确，请确保工具包配置文件命名正确，位于正确的文件夹中且语法正确。


4. 选择环境的复选框（此时会自动选中“Web Add-in”）然后单击 **OK** 以加载工具包的支持。
5. 使用“Define New Test Object”对话框从定义的测试对象类新建测试对象。环境的名称将显示在 **Environment** 列表中。如果从列表中选择环境的名称，则在测试对象配置文件中定义的测试对象类将显示在 **Class** 列表中。

有关详细信息，请参见《HP Unified Functional Testing 用户指南》中关于在对象存储库中定义新测试对象的部分。

6. 打开包含自定义控件的应用程序。
7. UFT 已可以识别您的控件，您可以创建带有自定义测试对象的测试步骤。（有关在 UFT 中使用以下选项的详细信息，请参见《HP Unified Functional Testing 用户指南》。）

- a. 使用“Object Spy”  查看受控件支持的标识属性和测试对象操作。
 - 对于定义的所有测试对象类，Object Spy 将显示在测试对象配置文件中定义的所有标识属性。新标识属性显示时不含任何值，因为您尚未实现从该控件检索值的方法。
对于与基本测试对象类属性同名的标识属性，如果控件的根 Web 元素与基本测试对象类匹配，则 UFT 使用基类实现来检索属性值。有关详细信息，请参见[实现对标识属性的支持 \(第 52 页\)](#)。

备注: UFT 在标识属性名称中仅使用小写字母。如果在测试对象配置文件中的标识属性名称含有大写字母，则这些大写字母将转换为小写字母。

- Object Spy 将显示定义用于表示控件的每个测试对象类的所有可用测试对象操作。其中包括从基本测试对象类继承的测试对象方法和属性，以及在测试对象配置文件中定义的测试对象方法。
- b. 使用“Object Repository”对话框中的 **Add Objects to Local**  按钮识别控件。确保使用正确的图标表示对象存储库中的测试对象。

- c. 在 Keyword View 中, 从定义的类创建带有测试对象的测试步骤。
 - 如果为此类定义了默认操作, 则该操作将在选择 **Item** 列中的测试对象后显示在 **Operation** 列中。
 - **Operation** 列中的可用操作列表反映了测试对象配置文件中的定义, 还包括从基本测试对象类继承的操作。
 - 选择操作后, **Value** 单元格根据选定操作的参数数量进行分区, 如果您为此操作定义了可能的值 (在 ListOfValues 元素中), 则这些值将显示在列表中。
 - 您在测试对象配置文件中为测试对象方法定义的描述和文档字符串分别显示在工具提示和 **Documentation** 列中。
- d. 在编辑器中, 从定义的类创建带有测试对象的测试步骤。语句完成功能将基于测试对象配置文件中的定义 (如果相关) 显示测试对象的所有可用操作, 以及这些操作可用的输入值。(继承的测试对象方法也会显示。)
- e. 在“Step Generator”中, 从定义的类创建带有测试对象的测试步骤。在测试对象配置文件中定义的操作将显示在“Operation”列表中, 为这些操作定义的描述将显示为工具提示。(继承的测试对象方法也会显示。)

开发对其他 UFT 功能的支持后, 要测试工具包支持集:

1. 按照上一过程中的步骤部署工具包支持集, 打开 UFT, 加载此支持, 然后运行带有您环境中控件的应用程序。
2. 根据其开发支持的 UFT 功能, 在应用程序上执行相关的 UFT 操作, 以测试该支持。例如, 识别应用程序中的控件、在应用程序上运行测试、记录应用程序上的测试步骤等。

默认情况下, UFT 在网页打开时将您开发的工具包支持集 JavaScript 文件注入网页。要测试您在 JavaScript 文件中所做的后续更改, 需要关闭并重新打开包含自定义控件的网页。

或者, 可在工具包配置文件中定义 **Controls\Settings\Variable** 元素, 将其 **name** 属性设置为 cache scripts, 将其 **value** 属性设置为 false。此设置将指示 UFT 在每次刷新网页时重新将 JavaScript 文件注入网页。因此, 修改 JavaScript 函数并进行部署后, 如果尚未在 XML 文件中进行任何更改或未重新打开 UFT, 则仅需要通过刷新网页来测试更改。

完成工具包支持集的开发后, 请确保删除此元素或将值属性设置为 true, 以避免性能降低。

记录和调试自定义支持

当为工具包支持集设计 JavaScript 函数时, 可使用 **_util.LogLine** 方法将日志消息添加到 Microsoft Windows 事件日志中。当 UFT 使用您设计的支持运行测试或组件时, 可在事件查看器中查看这些消息。有关详细信息, 请参见[使用 Microsoft Windows 事件日志 \(第 46 页\)](#)。

当测试您设计的工具包支持集时, 可使用 Microsoft Script Debugger 或 Microsoft Visual Studio 调试器, 像调试任何其他 JavaScript 文件一样调试您的 JavaScript 文件。有关详细信息, 请参见[调试 JavaScript 文件 \(仅限 Internet Explorer\) \(第 48 页\)](#)。

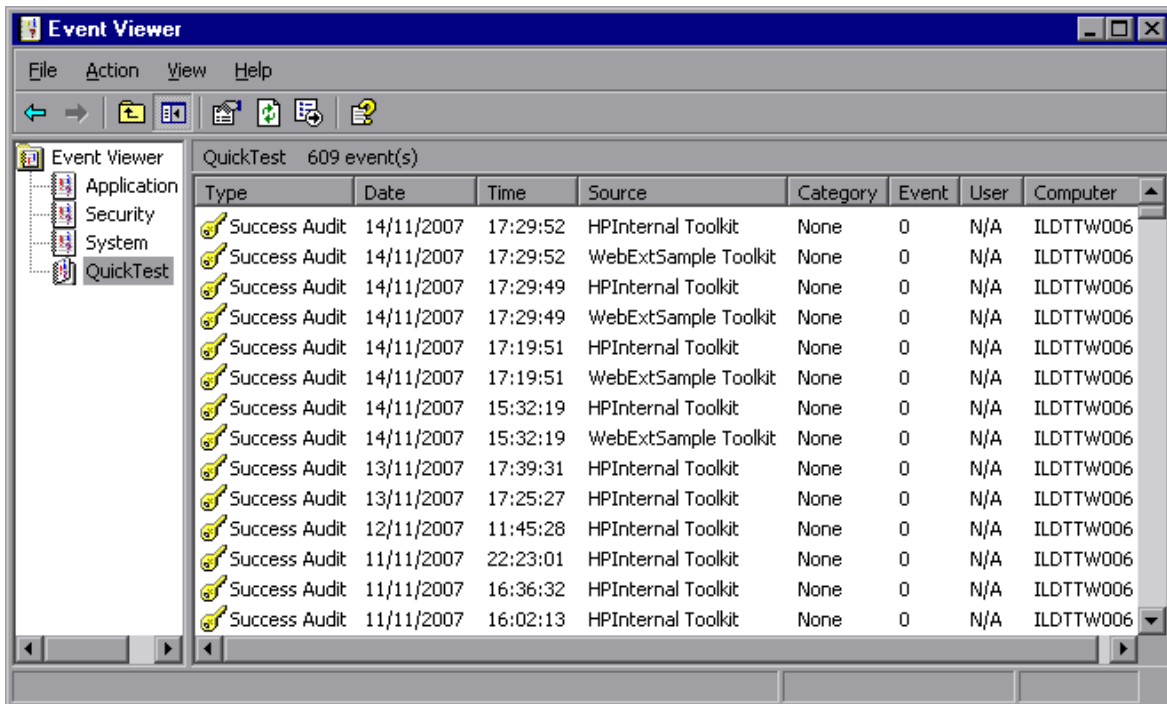
使用 Microsoft Windows 事件日志

可使用 JavaScript 函数中的 **_util.LogLine** 方法将消息添加到 Microsoft Windows 事件日志中。有关详细信息, 请参见“UFT Web 插件扩展性 API Reference” (可通过“Web 插件扩展性帮助”获得)。

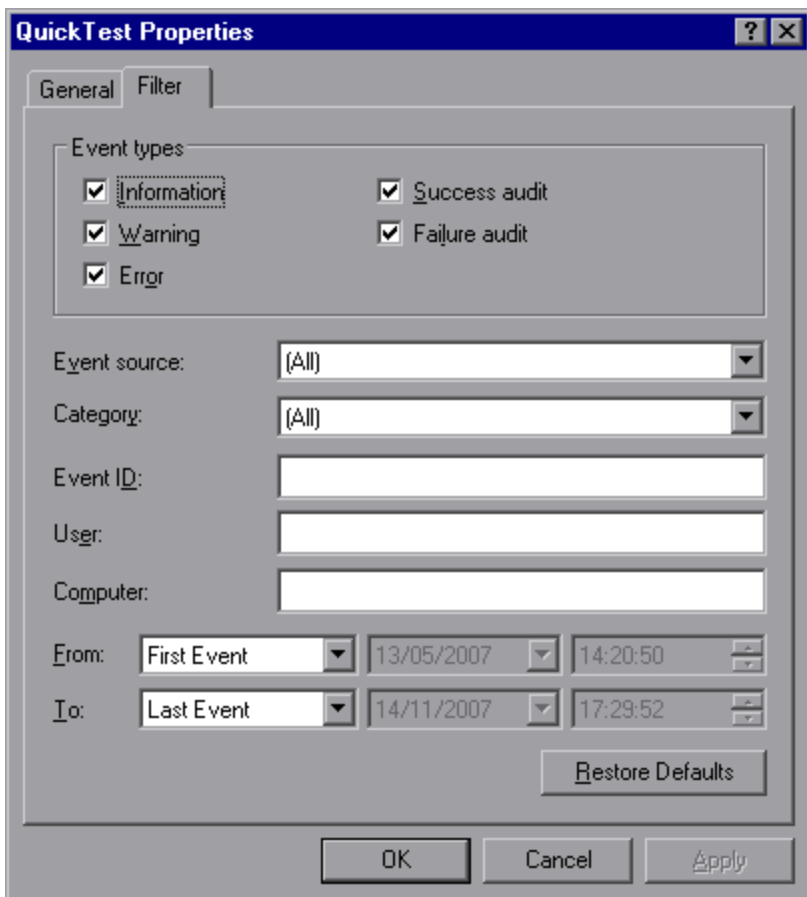
您为日志条目提供日志消息文本和严重级别，还可以选择提供 ID 和类别编号。UFT 将为您提供的信息添加工具包名称以及时间和日期戳，并将该条目添加到事件日志中。

另外，在识别 Web 插件扩展性支持的对象和在对象上执行测试时，UFT 也会将日志和错误消息写入事件日志中。

要查看事件日志并分析工具包支持集的性能，请打开事件查看器（在 Windows XP 和 Windows 2000 中，选择 **Start > Settings > Control Panel > Administrative Tools > Computer Management**，在 **Computer Management** 树中展开 **Event Viewer** 节点），然后选择 UFT 节点。双击特定日志条目查看其文本。



可根据严重度和其他消息字段筛选事件查看器中显示的日志消息。在“Computer Management”工具栏上, 选择 **View > Filter**。有关详细信息, 请参见“Event Viewer Help”(在事件查看器中选择 **Action > Help**)。



调试 JavaScript 文件 (仅限 Internet Explorer)

可使用 Microsoft Script Debugger 或 Microsoft Visual Studio 调试器调试您为工具包支持集编写的 JavaScript 文件。

要启用调试功能, 必须取消选中 Internet Explorer 高级选项 (**Tools > Internet Options > Advanced**) 中的 **Disable script debugging (Internet Explorer)** 和 **Disable script debugging (other)** 选项。更改此选项后, 需要重启 Internet Explorer, 更改才能生效。

如果希望实时调试器将 Microsoft Visual Studio 调试器列为可用的调试器之一, 则需要在“Visual Studio Options”对话框中选择 **Script** 选项 (**Visual Studio > Tools > Options > Debugging > Just-In-Time**)。

启用这些调试选项后, 可使用所有标准方法来调试 JavaScript 函数。例如:

- 可以将函数附加到 Internet Explorer 进程, 在函数中设置断点。
- 可在 JavaScript 函数中包含 `debugger;` 语句, 在函数运行时启动实时调试器。
- 当发生异常导致实时调试器打开时, 可使用实时调试器。

实现对测试对象方法的支持

使 UFT 能够识别自定义控件后，必须提供运行测试对象方法的支持。如果在未实现这些方法的情况下，尝试使用在自定义测试对象上运行的步骤运行测试，则测试将失败，并发生运行时错误。

对于在测试对象配置文件中定义的每个测试对象方法，都必须编写 JavaScript 函数，供 UFT 在控件上运行以执行步骤。除了执行步骤，还可以设计 JavaScript 函数，用于将行添加到运行结果、将日志消息添加到事件日志，以及根据需要向 UFT 用户显示消息框。有关详细信息，请参见[为工具包支持集设计 JavaScript 函数 \(第 34 页\)](#)。

在工具包配置文件中，您需要指定 UFT 将在其中查找 JavaScript 函数的 JavaScript 文件，还可以选择指定每个测试对象方法使用的函数名称。

可以在工具包配置文件中指定每个测试对象方法的 JavaScript 文件和函数，也可以定义每个工具包（**Controls** 元素）或每个测试对象类（**Control** 元素）的默认 JavaScript 文件。UFT 对所有未指定实现文件的测试对象方法使用这些默认文件。默认情况下，UFT 在指定文件中搜索与测试对象方法同名的 JavaScript 函数。因此，您无需在工具包配置文件中指定函数名称，除非您选择将函数命名为不同于相应测试对象方法的名称。有关详细信息，请参见“Toolkit Configuration Schema Help”（可通过“UFT Web 插件扩展性帮助”获得）。

警告: 不要创建名为 `get_property_value` 或 `get_list_of_values` 的 JavaScript 函数。这些名称已预留给 UFT 调用（默认情况下）用于检索测试对象标识属性的运行时值并列出测试对象方法参数的可能值列表的 JavaScript 函数。

创建对运行测试对象方法的支持后，可在自定义测试对象上运行 UFT GUI 测试，并验证工具包支持集是否正确执行。有关测试工具包支持集的详细信息，请参见[在开发期间测试工具包支持集 \(第 44 页\)](#)。

以下示例来自于 ASPAajax 工具包支持集，其中包含了对 ASPAajaxTabs 测试对象上的 **Select** 方法的支持。在 ASPAajax 测试对象配置文件中，声明如下：

```
<ClassInfo BaseClassInfoName="WebElement" Name="ASPAjaxTabs"
  DefaultOperationName="Select" >
  <TypeInfo>
    <Operation ExposureLevel="CommonUsed" Name="Select" PropertyType="Method">
      <Description>Selects the specified tab.</Description>
      <Documentation>
        <![CDATA[Select the tab with index %a1.]]>
      </Documentation>
      <Argument Name="Index" IsMandatory="true" Direction="In">
        <Type VariantType="Integer"/>
        <Description>
          The index value of the tab to select.
        </Description>
      </Argument>
    </Operation>
  </TypeInfo>
</ClassInfo>
```

在工具包配置文件中，ASPAjaxTabs 测试对象类的 **Control\Settings** 元素定义如下：

```
<Control TestObjectClass="ASPAjaxTabs">
  <Settings>
    <!-- Indicates the location of the JavaScript file that contains the implementation of the script -
    ->
    <variable name="default_imp_file" value="Tabs.js"/>
  </Settings>
</Control>
```

没有为此测试对象类定义任何 **Run** 元素。因此，当 UFT 使用 **Select** 测试对象方法运行步骤时，UFT 将在 **Tabs.js** 文件中（取决于工具包配置文件中的定义）搜索名为 **Select** 的 JavaScript 函数。

在 **Tabs.js** 文件中，通过实现 **Select** JavaScript 函数接受索引并选择带有此索引的选项卡。

支持方法参数值的动态列表

当 UFT 用户使用测试对象方法创建测试步骤时，UFT 可以为该方法的参数显示一组预定义的可能值。例如，如果某参数为布尔参数，则 UFT 可将 **true** 和 **false** 显示为可能值；或者，对于 **month** 参数，UFT 可显示所有月份的名称列表。但有时，参数仅限于一组特定的可能值，不过具体取决于步骤所针对的特定对象。例如：

- 与函数 **Table(<表名称>).SetCellData (行, 列)** 的整型 **row** 和 **column** 参数实际相关的值仅限于特定表中的行数和列数。
- 与函数 **Tree(<树名称>).Select (路径)** 的字符串 **path** 参数相关的值仅限于特定树中存在的路径。

通过使用扩展性，您可以使 UFT 能够为测试对象方法参数动态提供值列表。UFT 仅当使用语句完成功能时，才会在编辑器中提供此列表。

支持值的动态列表：

1. 在测试对象配置文件中，将 **Argument** 元素的 **DynamicListOfValues** 属性设置为 **true**。
2. 在工具包配置文件中，您可以指定文件名和 UFT 必须调用以检索值列表的 JavaScript 函数的函数名。默认情况下，UFT 将从您在 **Control\Settings** 元素的 **default_imp_file** 变量内为测试对象类指定的默认实现文件中调用 **get_list_of_values** JavaScript 函数，以此请求值的列表。有关详细信息，请参见“Toolkit Configuration Schema Help”，该文档可从“UFT Web 插件扩展性联机帮助”获取。

UFT 将为测试对象配置文件中 **DynamicListOfValues** 属性设置为 **true** 的每个参数调用 JavaScript 函数。为此函数提供的参数指示请求值的测试对象方法和参数。

3. 编写一个 JavaScript 函数，该函数必须能够接受测试对象方法和参数的名称，并针对当前元素返回与指定参数相关的值列表。该函数将返回连接到一个字符串的字符串值，且每个值的两边加上引号。

备注：值的动态列表从待测试应用程序的控件中检索。因此，要显示值的动态列表，则编辑测试时相关控件必须在应用程序中可见。

例如, 在 WebExtSample 环境的工具包支持集中 (位于 **<Web 插件扩展性 SDK 安装文件夹>\samples\WebExtSample** 文件夹中), 对于测试对象类 WebExtBook 的 **GoToAuthorPage** 测试对象方法中的 **AuthorName** 参数, 系统支持值的动态列表。

- 在 **WebExtSampleTestObjects.xml** 测试对象配置文件中, 该参数定义如下:

```
<Operation Name="GoToAuthorPage" PropertyType="Method">
  <Description>
    Opens the Web page for the specified author.
  </Description>
  <Argument Name="AuthorName" IsMandatory="true" Direction="In"
    DynamicListOfValues="true">
    <Type VariantType="String"/>
    <Description>The author.</Description>
  </Argument>
</Operation>
```

- 在 **WebExtBook.js** 文件中 (此文件已被定义为 **WebExtSample.xml** 工具包配置文件中的 WebExtBook 测试对象类的默认实现文件), 以下 JavaScript 函数旨在返回书籍作者的列表, 每位作者的两边加上引号:

```
// Dynamic list of values implementation
// ~~~~~
function get_list_of_values( method, argIndex )
{
  // When creating a step with the GoToAuthorPage test object
  // method, provide a list of the authors of this book, that
  // can be used for the method's argument.
  if (method == "GoToAuthorPage")
  {
    return get_GoToAuthorPage_list_of_values(argIndex);
  }
  return null;
}
function get_GoToAuthorPage_list_of_values(argIndex)
{
  var arr = new Array();
  if( argIndex > 1 )
    return toSafeArray(arr);
  // Retrieve all authors
  var AuthorsCount = 0;
  var authors = window.$(_elem.rows[1].cells[0]).children("A");
  for( var i = 0 ; i < authors.length ; ++i )
  {
    arr[AuthorsCount] = "\"" + authors.eq(i).text() + "\"";
    AuthorsCount++;
  }
  return toSafeArray(arr);
}
```

```
}
```

备注: \$ 指示此代码使用 jQuery JavaScript 函数库提供不受浏览器影响的支持。

实现对标识属性的支持

在测试对象配置文件中，您为测试对象类定义了标识属性。UFT 在运行测试时需要检索这些属性的值。UFT 在各种测试对象方法（如 **GetROProperty**）中将使用标识属性运行时值。各种基本功能（如创建检查点和输出值）也需要使用标识属性运行时值。

要支持检索标识属性的运行时值，需要实现接受 **PropertyName** 参数并返回 UFT 请求的任意属性值的 JavaScript 函数。（UFT 在标识属性名称中仅使用小写字母。如果测试对象配置文件中的标识属性名称含有大写字母，则这些大写字母将转换为小写字母。）

此函数必须以下列某种格式返回属性值：**String**、**Integer**、**Boolean** 或 **array**。返回数组时，请使用 **toSafeArray** 函数将数组转换为 UFT 要求的格式。当以数组格式提供标识属性值时，UFT 将数组转换为分号分隔的字符串。

有关为 Web 插件扩展性编写 JavaScript 函数的详细信息，请参见[为工具包支持集设计 JavaScript 函数 \(第 34 页\)](#)。

当满足以下条件时，UFT 将使用基本测试对象类实现来检索标识属性值：

- 控件包含基本元素（有关详细信息，请参见[扩展现有测试对象类 \(第 32 页\)](#)）。
- 标识属性在测试对象配置文件中定义为与基本测试对象类属性同名。
- 未提供返回该标识属性值的函数。

在下列情况下，实现 JavaScript 函数以返回测试对象配置文件中定义的标识属性值：

- 检索此标识属性值的基本测试对象类实现不可用。
- 基本测试对象类实现未满足您的要求。

通过设计返回标识属性值的函数来返回 **logical_name** 属性值，您可以控制 UFT 如何为此测试对象类的测试对象命名。有关详细信息，请参见[自定义测试对象名称 \(第 53 页\)](#)。

备注:

在工具包配置文件中，您可以指定实现检索属性值的 JavaScript 函数的 JavaScript 文件。还可以指定为此实现的函数的名称（在 **Control\Run\Properties** 元素中）。但是，如果未指定函数名称，UFT 将调用 **get_property_value (PropertyName)**，此为必须实现的函数。如果不指定文件名，UFT 将从您在 **Control\Settings** 元素（在测试对象类级别）或 **Controls\Settings** 元素（在工具包级别）中指定的默认 JavaScript 文件中调用函数。有关详细信息，请参见“Toolkit Configuration Schema Help”，可通过“UFT Web 插件扩展性帮助”获得。

创建对检索标识属性运行时值的支持后，可测试工具包支持集是否能够正确支持 UFT 在 Web 元素上运行检查点、输出属性值、在 Object Spy 中显示属性值，并使用 **GetROProperty** 操作运行测试步骤。有关测试工具包支持集的详细信息，请参见[在开发期间测试工具包支持集 \(第 44 页\)](#)。

自定义测试对象名称

UFT 识别对象时，将为页面上的每个测试对象创建唯一的名称。描述性测试对象名称使您可以区分相同类的测试对象，并且更容易在对象存储库和测试中标识对象。

默认情况下，测试对象使用其测试对象类的名称（如果页面上的同一个类有多个测试对象，则附加索引）。在许多情况下，这都不是自定义控件的理想名称。

测试对象的名称需要对 UFT 用户有意义，最好使用与工具包相关的术语。UFT 将在 Keyword View、Editor 和对象存储库中显示此名称。测试对象名称不用于对象标识，因此无需在应用程序中保持不变。

例如，测试对象名称可能依赖于语言。UFT 用户可以使用以一种语言运行的应用程序创建测试，从而创建该语言名称的测试对象。然后，用户可以使用其他语言在该应用程序上运行此测试。测试中测试对象的名称将保持原来的语言，但 UFT 可以基于描述正确识别测试对象并在对象上执行操作。

要控制 UFT 对测试对象类的测试对象的命名方式，可设计返回标识属性值以返回 **logical_name** 属性值的函数。UFT 将此值用作测试对象的名称。

例如，WebExtSample 环境中的 **WebExtBook** 测试对象将根据其书籍标题命名。WebExtSample 环境的示例工具包支持集位于 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** 文件夹中。**WebExtBook.js** 文件被定义为 **WebExtSample.xml** 工具包配置文件中的默认实现文件。返回标识属性运行时值的 JavaScript 函数在 **WebExtBook.js** 文件中定义如下：

```
function get_property_value(prop)
{
  if ( prop == "logical_name" || prop == "title" )
    // For the "title" identification property, as well as the
    // "logical_name" property, return the inner text of the
    // second cell in the first row.
    {
      return window.$(_elem.rows[0].cells[1]).text();
    }
}
```

备注: \$ 指示此代码使用 jQuery JavaScript 函数库提供不受浏览器影响的支持。

实现识别子控件的筛选

指示 UFT 识别网页时，将打开“Define Object Filter”对话框，用于确定应识别哪些网页的子网页。选择 **All object types** 会指示 UFT 识别自定义控件及其父网页，自定义控件包含的所有控件也会识别为网页的子网页（以及控件自身的同级控件）。

在某些情况下，无需为控件的所有子控件创建测试对象。例如，当没有重要操作要在子控件上执行时，没有要检索的属性时，或者当出于测试目的，将子控件上执行的操作作为父控件上执行的操作进行查看时。例如，在含有按钮控件的计算器控件上，将无需为数字按钮创建测试对象。在计算器对象自身上按数字按钮执行 **Set** 操作，可为计算器操作提供数字输入。

通过为创建的测试对象类定义识别筛选可决定 UFT 识别的控件。可使用工具包配置文件中的 **Control\Filter\Learn** 元素定义基本筛选，或可通过编写 JavaScript 函数实现复杂筛选。如果使用 JavaScript 函数设计筛选，请在工具包配置文件中指定函数的位置和名称。

在工具包配置文件中的 **Control\Filter\Learn** 元素内，可以定义：

- 是否识别此测试对象类表示的控件。还可以指定 UFT 应仅识别包含子控件的此类控件。
- 是否识别通过此测试对象类表示的控件内包含的控件。还可指定需要调用 JavaScript 函数来决定要识别的子对象。

如果编写 JavaScript 函数来实现此筛选，则该函数必须返回包含您希望 UFT 识别的所有 Web 子元素的 SafeArray。有关详细信息，请参见[为工具包支持集设计 JavaScript 函数 \(第 34 页\)](#)。

有关详细信息，请参见“Toolkit Configuration Schema Help”，可通过“UFT Web 插件扩展性帮助”获得。

您可以在 WebExtSample 环境的示例工具包支持集中看到定义识别筛选的示例，该支持集位于 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** 文件夹中。

- **WebExtBook** 测试对象类的 **Filter** 元素（在 **WebExtSample.xml** 文件中）定义如下：

```
<Filter>
  <Learn learn_control="Yes" learn_children="No"/>
</Filter>
```

此文本指示 UFT 识别 **WebExtBook** 测试对象而不识别其子对象。

- **WebExtUsedBooks** 测试对象类的 **Filter** 元素定义如下：

```
<Filter>
  <Learn learn_control="Yes" learn_children="CallFilterFunc"
    type="javascript" function="GetChildrenToLearn" />
</Filter>
```

这段代码指示 UFT 识别 **WebExtUsedBooks** 测试对象，并调用决定要识别哪些子对象的 **CallFilterFunc** JavaScript 函数。

GetChildrenToLearn JavaScript 函数位于 **WebExtUsedBooks.js** 文件中，此文件在 **WebExtSample.xml** 工具包配置文件中被定义为默认实现文件。**GetChildrenToLearn** JavaScript 函数返回 UsedBooks 表控件的所有单选按钮子对象：

```
function GetChildrenToLearn()
{
  // Return all of the radio buttons in the used books table
  return toSafeArray(window.$(_elem).children()[0].getElementsByTagName("input"));
}
```

备注： \$ 指示此代码使用 jQuery JavaScript 函数库提供不受浏览器影响的支持。

实现识别筛选后，可指示 UFT 识别您的自定义控件，并验证工具包支持集是否可以正确控制要识别的子控件。有关测试工具包支持集的详细信息，请参见[在开发期间测试工具包支持集 \(第 44 页\)](#)。

实现对录制的支持

将对象添加到对象存储库中并在 UFT 中创建测试的一种方式就是录制。要录制测试，UFT 将注册以侦听 Web 元素上的事件，在事件发生时，UFT 将相关步骤添加到测试。默认情况下，UFT 使用标准的 Web 事件配置来决定针对每个 Web 元素要侦听的事件，以及在每个事件发生时，要在测试中录制的步骤。

如果要对您定义的测试对象类上的录制进行自定义，则必须指定要录制的事件，以及在这些事件发生时 UFT 要添加到测试的步骤。

针对每个要自定义录制的测试对象类，请在工具包配置文件中定义 **Control\Record\EventListening** 元素。在此元素中，您可以指定是否使用标准 Web 事件配置来处理使用此测试对象类表示的控件上的事件。另外，您可以指定是否使用标准 Web 事件配置来处理在这些控件的子控件上发生的事件。

除了指定 UFT 是否应使用标准 Web 事件配置外，您还可以指定提供更具体事件注册的 JavaScript 函数（还可以选择指定包含此函数的文件名）。有关详细信息，请参见“Toolkit Configuration Schema Help”（可通过“UFT Web 插件扩展性帮助”获得）。

除了工具包配置文件中的定义，您还将编写以下类型的 JavaScript 函数：

- 使用 **_util** 实用程序对象中的 **RegisterForEvent** 函数注册以侦听正确元素上正确事件的 JavaScript 函数（在工具包配置文件中指定）。此函数的参数也决定了在每个事件发生时 UFT 将调用的 JavaScript 函数。

UFT 根据标准 Web 事件配置在注册以侦听事件后调用此函数。此函数执行的事件注册将覆盖同一事件之前的任何注册。对于此函数未处理的事件，将使用标准注册。

- 一个或多个处理事件的 JavaScript 函数，在事件发生时，通过调用 **_util** 实用程序对象中的 **Record** 函数向 UFT 通知要添加到测试中的步骤。

Record 函数，以及其他实用程序对象函数需要 **SafeArray** 类型参数。要将数组转换为 **SafeArray**，可使用 Web 插件扩展性提供的 `toSafeArray(array)` 函数。此函数在 **<Extensibility Accelerator 安装文件夹>\bin\PackagesToLoad\common.js** 中定义。（此文件也位于 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits** 文件夹中。）

有关实用程序对象函数的语法信息，请参见“UFT Web 插件扩展性 API Reference”中的 **_util** 部分（可通过“Web 插件扩展性帮助”获得）。有关为 Web 插件扩展性编写 JavaScript 函数的详细信息，请参见 [为工具包支持集设计 JavaScript 函数 \(第 34 页\)](#)。

您可以在 WebExtSample 环境的示例工具包支持集中看到自定义记录的示例，该支持集位于 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** 文件夹中。

- 在工具包配置文件中 **WebExtBook** 测试对象类的 **Control** 元素内，定义以下 **Record\EventListening** 元素：

```
<Record>
  <EventListening
    use_default_event_handling_for_children="false"
    use_default_event_handling="false"
    type="javascript" function="RegisterToEvents"/>
</Record>
```

此文本指示 UFT 在 Book 控件及其子控件上不使用默认“Web Event Configuration”处理事件，而是调用 JavaScript 函数 **RegisterToEvents**。因为未定义 JavaScript 文件，所以 UFT 将在您在 WebExtBook 测试对象类的 **Control** 级别上指定的 **WebExtBook.js** 文件中查找 JavaScript 函数。

- 在 **WebExtBook.js** 文件中，定义以下 **RegisterToEvents** 函数：

```
function RegisterToEvents( elem )
{
  // Connect to the "Select" event:When the book name or the
  // book icon is clicked, call OnSelectClicked.
  _util.RegisterForEvent( window.$(_elem.rows[0].cells[0]).
    children[0], "onclick", "OnSelectClicked" );
  _util.RegisterForEvent( window.$(_elem.rows[0].cells[0]),
    "onclick", "OnSelectClicked");
}
```

备注: \$ 指示此代码使用 jQuery JavaScript 函数库提供不受浏览器影响的支持。

此函数注册 UFT 以侦听书籍标题和图像上的单击事件。注册事件时，此函数将指定 UFT 在事件发生时必须调用的 JavaScript 函数。

- 在 **WebExtBook.js** 文件中，添加以下事件处理程序 JavaScript 函数：

```
function OnSelectClicked( handlerParam , eventObj )
{
  // Record the "Select" step
  var arr = new Array();
  _util.Record( "Select", toSafeArray(arr) , 0 );
  return true;
}
```

此函数在书籍标题或图像被单击时记录 WebExtBook 测试对象上的 **Select** 测试对象方法。

实现对记录的支持后，可记录环境中控件上的测试，并验证工具包支持集是否正确执行。有关测试工具包支持集的详细信息，请参见[在开发期间测试工具包支持集 \(第 44 页\)](#)。

疑难解答和限制 —— 开发支持

此部分描述有关开发 Web 插件扩展性支持的疑难解答和限制。

- Web 插件扩展性可用于在网页和框架中为 Web 控件创建支持。无法为网页或框架自身开发自定义支持。也就是说，无法使用 Web 插件扩展性将网页或框架映射到开发的测试对象类中。
- 要使 UFT 成功使用通过 Web 插件扩展性开发的支持，浏览器的安全设置必须设置为允许活动脚本。（例如，在 Internet Explorer 7 中，您可以在以下位置找到这些安全设置：**Tools > Internet Options > Security > Custom Level > Scripting > Active scripting**。）
- 设计 JavaScript 函数时，使用简单的 **Click** 或 **FireEvent** 方法调用无法单击 Mozilla Firefox 页面上的链接。Web 插件扩展性 **_util** 对象提供可用于在此情况下单击链接的特殊 **FireEvent** 方法。（“API Reference”中未介绍此方法，因为此方法仅用作这种类型的变通方法）。

下面是如何使用此方法的示例：

```
function clickOnLink(link) {
    if (_util.GetBrowserType() == QtpConstants.IE) {
        link.click();
    }
    else {
        //Firefox.
        var evObj = window.document.createEvent("MouseEvents");
        evObj.initEvent("click", true, true);
        _util.FireEvent(link, "click", evObj);
    }
}
```

第 4 章: 部署工具包支持集

为自定义工具包扩展 UFT 支持的最后一个阶段是部署工具包支持集。这意味着要将您创建的所有文件放在已安装 UFT 的计算机上的正确位置，使 UFT 能够识别工具包中的控件并对其运行测试。

开发工具包支持集时，如果将其部署到 UFT，则可测试并调试您创建的支持。工具包支持集完成后，可以将其部署至已安装 UFT 的任何一台计算机上，以扩展 Web 插件。

此章节包括：

- [关于部署自定义工具包支持](#) 59
- [部署自定义工具包支持](#) 59
- [修改已部署的支持](#) 60
- [删除已部署的支持](#) 61

关于部署自定义工具包支持

从 UFT 用户的角度来看，在已安装 UFT 的计算机上部署工具包支持集之后，工具包支持集可用作 UFT 插件。

UFT 打开后，它会将插件管理器中的工具包支持集的环境名称显示为 Web 插件节点下的子节点。针对您的环境选中该复选框，可以指示 UFT 使用您开发的工具包支持集来加载环境支持。

如果已加载环境支持，则：

- UFT 将识别环境中的控件，且能够对其运行测试。
- UFT 将在所有显示插件列表或受支持环境列表的对话框中显示环境的名称。
- UFT 将显示由对话框（该对话框显示每个插件适用的测试对象类列表）中的工具包支持集所定义的测试对象类列表。（例如：“Define New Test Object”对话框、“Object Identification”对话框。）

部署自定义工具包支持

要部署您创建的工具包支持集，必须将文件放在 UFT 安装文件夹内的特定位置。

备注：开始之前，在 <UFT 安装文件夹>\dat\Extensibility\Web 文件夹中，使用自定义工具包名称创建一个文件夹（如果此文件夹不存在）。

下表描述了每个工具包支持文件的相应位置：

文件名	位置
<自定义工具包名称>TestObjects.xml 备注： 这是建议的文件名约定。您可以拥有多个测试对象配置 XML 文件，并按需对它们命名。	<ul style="list-style-type: none">• <UFT 安装文件夹>\dat\Extensibility\Web• <UFT Add-in for ALM 安装文件夹>\dat\Extensibility\Web (可选。仅在文件夹存在时需要，这意味着 UFT Add-in for ALM 是单独从 ALM 插件页安装的，不是 UFT 安装的一部分。)
<自定义工具包名称>.xml	<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits \<自定义工具包名称>
JavaScript 文件	在 <自定义工具包名称>.xml 内的 中指定位置
新测试对象类的图标文件 (可选)	该文件可以是 .dll 、 .exe 或 .ico 文件，位于已安装 UFT 的计算机上或可访问的网络位置中。 在 <自定义工具包名称>TestObjects.xml 中指定位置
测试对象类的帮助文件 (可选)	必须为 .chm 文件，位于已安装 UFT 的计算机上。 在 <自定义工具包名称>TestObjects.xml 中指定位置
外部 JavaScript 库 这些库不属于工具包支持集的一部分，但在部署支持集时，必须确保库文件位于工具包配置文件中指定的位置。	.js 文件，位于已安装 UFT 的计算机上或可访问的网络位置中。 在 <自定义工具包名称>.xml 中指定位置

建议的文件位置

您可为工具包支持集的配置文件中的 JavaScript、帮助和图标文件指定位置。您可以使用相对路径指定这些位置。有关详细信息，请参见“Test Object Schema Help”和“Toolkit Configuration Schema Help”“Toolkit Configuration Schema Help”（可通过“Web 插件扩展性帮助”获得）。

这些文件的建议位置如下表所述：

文件名	位置
JavaScript 文件	<UFT 安装文件夹>\dat\Extensibility\ Web\ Toolkits \<自定义工具包名称>\ JavaScript
图标文件	<UFT 安装文件夹>\dat\Extensibility\ Web\ Toolkits \<自定义工具包名称>\Res
帮助文件	<UFT 安装文件夹>\dat\Extensibility\ Web\ Toolkits \<自定义工具包名称>\Help

设计阶段要使用的设置

- 如果修改测试对象配置文件中的 **Identification Property** 元素属性，请在自定义工具包支持的设计阶段，将 **TypeInformation** 元素的 **DevelopmentMode** 属性设置为 true。在部署自定义工具包支持集进行常规使用之前，请确保删除此属性（或将其设置为 false）。有关详细信息，请参见[修改测试对象配置文件中的 Identification Property 属性 \(第 60 页\)](#)。
- 要指示 UFT 在每次刷新时均将 JavaScript 文件重新注入网页，请在工具包配置文件中定义 **Controls\Settings\Variable** 元素，将 **name** 属性设置为 cache scripts，并将 **value** 属性设置为 false。如果已执行此操作，则在修改并部署 JavaScript 函数之后，如果您尚未在 XML 文件中进行任何更改或未重新打开 UFT，则仅需刷新包含自定义控件的网页，即可测试所做的更改。无需关闭并重新打开该网页。
在部署常规使用的自定义工具包支持集之前，确保删除此元素或将其 **value** 属性设置为 true，以避免性能下降。

修改已部署的支持

如果修改已部署的工具包支持集，则必须重新打开 UFT 并重新运行 Web 应用程序，才能使所做的更改生效。

如果更改指定在 UFT 中使用属性的功能的标识属性定义，请参见[修改测试对象配置文件中的 Identification Property 属性](#)。

修改测试对象配置文件中的 Identification Property 属性

测试对象配置文件中 **Identification Property** 元素的以下属性指定可以在 UFT 中修改的信息（通过使用“Object Identification”对话框）：**AssistivePropertyValue**、**ForAssistive**、**ForBaseSmartID**、**ForDescription**、**ForOptionalSmartID** 和 **OptionalSmartIDPropertyValue**。这些属性确定 UFT 中用于不同目的的标识属性列表。有关详细信息，请参见“UFTTest Object Schema Help”，该文档可从“UFT Web 插件扩展性帮助”获取。

因此，默认情况下，UFT 仅会从 XML 文件中读取一次这些属性的值，防止覆盖用户通过使用“Object Identification”对话框所做的任何更改。这样，用户便可以通过 UFT 持续不断地定义属性列表。

如果用户单击“Object Identification”对话框中的 **Reset Test Object** 按钮，则系统将从 XML 重新加载属性值。

如果 XML 自上次加载以来发生了更改（根据系统中的文件修改日期看出），则 UFT 将从 XML 中读取属性。UFT 会根据这些属性的值将标识属性添加到相关列表（并根据需要调整顺序），但不会从列表中删除任何现有的标识属性。

要指示 UFT 在每次打开 UFT 时根据 XML 中定义的属性完全刷新标识属性列表，请将此测试对象配置文件中的 **TypeInformation** 元素的 **DevelopmentMode** 属性设置为 true。

修改 Identification Property 属性时的注意事项

- 如果修改测试对象配置文件中的 **Identification Property** 元素属性，请在自定义工具包支持的设计阶段，将 **TypeInformation** 元素的 **DevelopmentMode** 属性设置为 true。这可确保 UFT 使用您对文件所做的所有更改。
- 在部署常规使用的工具包支持集之前，请确保删除 **TypeInformation** 元素的 **DevelopmentMode** 属性（或将其设置为 false）。否则，UFT 每次打开时，均会根据测试对象配置文件中的定义刷新属性列表。如果 UFT 用户使用“Object Identification”对话框更改属性列表，则重新打开 UFT 时，所做的更改将会丢失。
- 尽管 UFT 在读取修改后的测试对象配置文件时，不会从属性列表中删除现有的属性（除非 **DevelopmentMode** 属性设置为 true），但它确实会添加属性并根据文件中的定义调整列表的顺序。如果 UFT 用户从列表中删除了属性，或者使用“Object Identification”对话框修改了顺序，则加载修改后的文件时，所做的更改将会丢失。

如果您为第三方提供自定义工具包支持集，并交付一个包含修改后的测试对象配置文件的升级程序，则需考虑向 UFT 用户指出，标识属性列表可能会存在此类潜在更改。

删除已部署的支持

打开 UFT 时，UFT 用户可以使用插件管理器指示 UFT 是否加载为任何特定环境或工具包提供的支持。

如果要在部署后从 UFT 中删除自定义工具包的支持，则必须从以下位置中删除其工具包配置文件：**UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\<自定义工具包名称>**

如果测试对象配置文件中的所有测试对象类定义均未用于表示任何自定义控件（意味着不再需要它们），则将其从以下位置删除：**UFT 安装文件夹>\dat\Extensibility\Web** 和 **UFT Add-in for ALM 安装文件夹>\dat\Extensibility\Web**（如果相关）。

第 2 部分: 教程: 学习创建 Web 自定义工具包支持

此教程中的课程均面向 Internet Explorer。也即您创建的 JavaScript 代码均以 Internet Explorer 样式编写，并且您为示例自定义控件创建的支持将只能在 Microsoft Internet Explorer 上正确运行。自行创建 Web 插件扩展性支持时，也可以将其设计为在 Mozilla Firefox 和 Google Chrome 上运行。有关详细信息，请参见 [开发不受浏览器影响的支持 \(第 26 页\)](#)。

第 5 章: 学习为简单的自定义 Web 控件创建 UFT 支持

在此课程，您将在随 Extensibility Accelerator for HP Functional Testing 安装的 Web 插件扩展性 Book 示例工具包中手动创建对 Book 控件的支持。创建对 Book 控件的支持只需要极少的自定义操作，使您能够了解创建工具包支持集的基本知识。

创建对自定义控件的支持时，请使用 Extensibility Accelerator 设计支持。Extensibility Accelerator 创建工具包支持集文件并自动设置 XML 文件，使您可以集中精力设计 JavaScript 函数。在此课程手动创建支持可使您更好地了解 Web 插件扩展性工具包支持集的结构和内容。

%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples

WebExtSample 文件夹包含此示例的完整工具包支持集，学习此课程时可供参考。

学习此课程之前，请确保已阅读[简介 UFT Web 插件扩展性 \(第 11 页\)](#)。

此课程包括：

- [本课程的准备工作](#) 64
- [计划对 Web 插件扩展性 Book 示例工具包的支持](#) 64
- [开发工具包支持集](#) 69
- [课程总结](#) 87

本课程的准备工作

扩展 UFT 对自定义控件的支持之前，必须能够访问其源文件。您无需修改自定义控件的任何源，即可在 UFT 中支持该控件，但您需要熟悉这些源。确保您知道组成控件的元素和属性、此控件上可能发生的事件等。设计支持类时将使用这些信息。

Book 控件的源文件位于 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm**。

打开文件以运行控件。



运行控件，打开控件的源文件并研究控件的行为和实现。

Book 控件包含的信息包括书籍的标题、作者、新版的价格，以及可购买旧版的最低价格。

单击书籍的标题或图像将打开一个页面，显示有关该书籍的更多详细信息（但在此示例中未实现）。单击作者姓名将打开一个页面，提供该作者的书籍列表（但在此示例中未实现）。单击 **Used** 将打开“UsedBooks”页面，列出此书籍的所有可用旧版及其价格。“UsedBooks”表是您将在此课程学习支持的更复杂的控件，[学习为复杂的自定义 Web 控件创建 UFT 支持 \(第 88 页\)](#)。

计划对 Web 插件扩展性 Book 示例工具包的支持

在此部分中，将根据您的控件知识，分析 UFT 当前识别 Book 控件的方式及其应采用的控件识别方式。接下来，确定[了解 Web 插件扩展性计划清单 \(第 22 页\)](#)中问题的答案，并相应地填写在[Web 插件扩展性计划清单 \(第 23 页\)](#)中。

执行此操作的最佳方式，一方面是使用“Object Spy”、“Keyword View”和“Record”选项分析 UFT 如何识别 Book 控件，另一方面是考虑如何实现控件及其目的。

1. 打开 UFT 并运行 Book 控件。


打开 UFT 并加载 Web 插件。

关闭 Book 控件的所有已打开的实例，然后通过打开

%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm 文件将其打开。

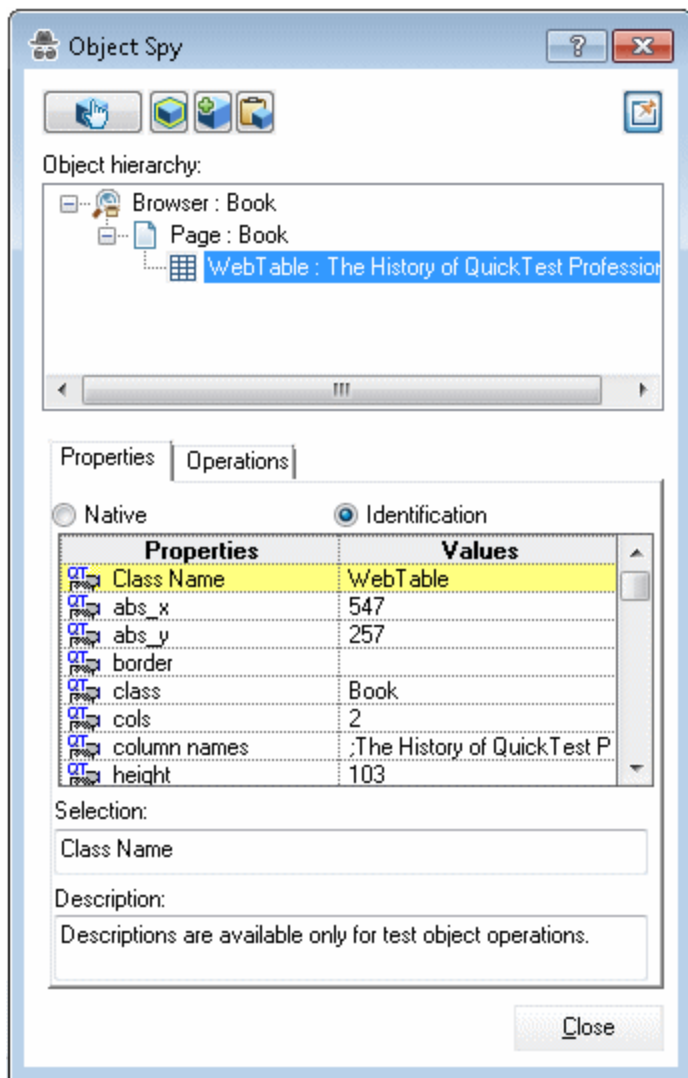
2. 使用 Object Spy 查看 Book 属性和操作。

在 UFT 中，打开 GUI 测试并选择 **Tools > Object Spy**，或单击 **Object Spy** 工具栏按钮  以打开“Object Spy”对话框。单击 **Properties** 选项卡并选择 **Identification Properties**。

在“Object Spy”对话框中，单击指向手 ，然后单击 Book 控件。

Book 控件作为 Web 表实现，已内置 UFT 支持，因此会将该控件识别为 **WebTable**，根据书籍标题命名。用于测试对象的图标是标准 WebTable 类图标。

下面，您可以查看根据上述信息完成的清单。



关闭 Object Spy。

3. 在 Book 控件上录制操作。

在 UFT 中，选择 **Record > Record and Run Settings** 打开“Record and Run Settings”对话框。在“Web”选项卡中，选择 **Record and run test on any open browser**。单击 **OK**。

单击 **Record** 按钮，或选择 **Record > Record**。单击 Book 控件中不同的链接（每次单击后必须返回到上一页以返回到 Book 控件）：书籍标题、控件中的图像、作者姓名和 **Used** 链接。


```
<td class="BookCell">
  </td>
</tr>
<tr>
  <td class="BookCell">
    New:<strong>59.99$</strong> &nbsp;<a href=".\\UsedBooks.htm">Used:</a> from
<strong>29.99$</strong>
  </td>
</tr>
</table>
```

此部分描述计划 Book 控件的支持时需要作出的决定，然后总结支持计划清单中的信息。

a. 选择表示自定义控件的测试对象类：

Book 控件作为 Web 表控件实现，帮助美化其外观。要在此控件上执行测试，UFT 无需将 Book 控件识别为表。另一方面，UFT 使用 WebElement 对象为一般 Web 元素提供的基本支持对 Book 控件来说不够具体。因此，需要新建名为 **WebExtBook** 的测试对象类，扩展 **WebElement** 并使 UFT 将其标识为表示 Book 控件的测试对象类。

b. 定义 UFT 如何标识用于表示控件的测试对象类：

如果控件的 **tagName** 属性是 **table** 且 **className** 属性是 **Book**，则使用 **WebExtBook** 测试对象表示此控件。

c. 确定新测试对象类的详细信息：

- 新测试对象类由以下图标文件表示：**<UFT 安装文件夹>\Dat\Extensibility\Web\Toolkits\WebExtSample\Res\WebBook.ico**
- 未提供任何帮助文件。
- 支持的新标识属性为：**title**、**authors**、**price** 和 **min_used_price**。这些标识属性都应显示在 UFT “Checkpoint Properties” 和 “Output Value Properties” 对话框中，默认情况下在检查点内选中。无属性用于智能标识。
唯一定义对象的标识属性是书籍的标题及其作者的姓名。
- 测试对象本身的名称应与其标题标识属性相同。

d. 决定自定义控件支持的测试对象方法：

WebExtBook 测试对象类提供以下测试对象方法：

- **Select**。模拟单击书籍的标题或图像。这是默认测试对象方法。
- **GoToAuthorPage**。模拟单击指定的作者姓名（运行期间应从特定控件检索可用的作者姓名）。
- **GoToUsedBooksPage**。模拟单击 **Used** 链接。

e. 决定是否需要支持任何方法参数值的动态列表：

是，**GoToAuthorPage** 方法中的 **AuthorName** 参数需要动态值列表。（这需要修改工具包配置文件以指定提供这些值的 JavaScript 函数，并设计相关的 JavaScript 函数）。

f. 定义 UFT 识别控件时应识别的子控件：

为方便测试，UFT 应识别所有操作，即使这些操作已在 Book 控件自身上执行，甚至已在其内部控件上得到技术性执行。因此，测试对象无需识别和表示任何控件的子控件。

g. 决定 Object Spy 是否应显示 **WebExtBook** 测试对象：是。

h. 定义是否支持录制以及录制哪些事件:

侦听控件中以下元素上发生的鼠标单击: title、image、authors 和 Used。单击这些元素中的某一个时, 将在测试中录制相关步骤。

i. 决定需要在工具包配置文件中设计的支持部分以及需要 JavaScript 函数的部分:

- 对于简单的 Book 控件, 测试对象类标识仅仅基于 HTML 属性值, 因此可使用不带 JavaScript 函数的工具包配置文件进行支持。
- 测试对象方法和标识属性可受使用默认命名约定的 JavaScript 函数支持, 因此无需在工具包配置文件中进行任何更改。
- 指示 UFT 不对控件的子控件进行识别可在工具包配置文件中设计, 不需要使用 JavaScript 函数。
- 要支持录制, 请将工具包配置文件修改为关闭默认 “Web Event Configuration”, 并指定注册 UFT 以侦听正确事件的 JavaScript 函数。另外, 请设计一个处理事件注册的 JavaScript 函数, 以及指示 UFT 在事件发生时录制相关步骤的其他 JavaScript 函数。

下面, 您可以查看根据上述信息完成的[清单](#)。

Web 插件扩展性计划清单

<input checked="" type="checkbox"/>	自定义控件支持计划清单	是否 在工 具包 XML 中指 定?	JavaScript 函数是否 支持?
<input checked="" type="checkbox"/>	此自定义控件的源位于: %ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\ WebExtSample\Application\Book.htm	N/A	N/A
<input checked="" type="checkbox"/>	指定新测试对象类扩展的 Web 测试对象基类: (默认值 —— WebElement) WebElement	N/A	N/A
<input checked="" type="checkbox"/>	基本测试对象类是否为 WebElement? 是 如果不是, 那么是否有基元素 (与基本测试对象类匹配的元素)? N/A 如果有基元素, 那么是否需要 JavaScript 函数将其返回? N/A	否	否
<input checked="" type="checkbox"/>	指定新 Web 测试对象类的详细信息: • 测试对象类名称: WebExtBook • 图标文件位置 (可选): <UFT 安装文件夹>\Dat\Extensibility\Web\ Toolkits\WebExtSample\Res\WebBook.ico • 描述的标识属性: title、authors • 默认测试对象方法: Select • 帮助文件位置: N/A	N/A	N/A
<input checked="" type="checkbox"/>	指定标识控件所使用的测试对象类的基础: tagName = table, className = Book.	是	否
<input checked="" type="checkbox"/>	指定测试对象命名基础: Use the book title	N/A	是
	列出要支持的标识属性。标记检查点可用的标识属性 (以及默认选中的标识属性) 和用于智	否	是

<input checked="" type="checkbox"/>	自定义控件支持计划清单	是否在工具包 XML 中指定?	JavaScript 函数是否支持?
<input checked="" type="checkbox"/>	能标识的标识属性 (如有) : title、authors、price、min_used_price (所有标识属性均可用于检查点并默认选中, 但不用于智能标识)		
<input checked="" type="checkbox"/>	列出要支持的测试对象方法 (如果需要, 包括参数、返回值、帮助文件位置和帮助 ID) : Select () GoToAuthorPage (AuthorName) GoToUsedBooksPage ()	否	是
<input checked="" type="checkbox"/>	是否提供所有测试对象方法参数值的动态列表? 是 如果提供, 则列出参数: GoToAuthorPage 方法的 AuthorName	N/A	是
<input checked="" type="checkbox"/>	指定 UFT 应使用控件识别的子类型: 无	是	否
<input checked="" type="checkbox"/>	是否在 Object Spy 中显示此类的测试对象? 是	否	N/A
<input checked="" type="checkbox"/>	是否提供录制支持? 是 如果提供, 则列出可触发录制的事件: 单击标题、图像、作者姓名和 “Used” 链接	是	是

开发工具包支持集

按照此部分的步骤为 WebExtSample 工具包开发工具包支持集并学习 Web 插件扩展性的基本知识。开发此工具包支持集包括以下阶段:

- [第 1 阶段: 创建工具包支持集](#), 如第 69 页所述
- [第 2 阶段: 将 WebExtSample 环境引入 UFT 中](#), 如第 70 页所述
- [第 3 阶段: 使 UFT 标识、侦测和识别 Book 控件](#), 如第 72 页所述
- [第 4 阶段: 实现对 WebExtBook 测试对象方法的支持](#), 如第 76 页所述
- [第 5 阶段: 实现对 WebExtBook 标识属性的支持](#), 如第 78 页所述
- [第 6 阶段: 更改测试对象的名称。](#), 如第 80 页所述
- [第 7 阶段: 实现防止识别子对象的筛选](#), 如第 81 页所述
- [第 8 阶段: 实现在 Book 控件上进行录制的支持](#), 如第 82 页中所述
- [第 9 阶段: 实现对 AuthorName 值动态列表的支持。](#), 如第 85 页所述

第 1 阶段: 创建工具包支持集

在此部分, 您可以为 WebExtSample 工具包创建组成工具包支持集的文件和文件夹。

创建工具包支持集:

1. 创建工具包支持集的文件夹。
可为此文件夹选择任何方便的名称和位置。
2. 在工具包支持集文件夹中, 创建一个名为 **WebExtSampleTestObjects.xml** 的文件。此为测试对象配置文件。
3. 在工具包支持集文件夹中, 创建一个名为 **Toolkits** 的文件夹。
4. 在 **Toolkits** 文件夹中, 创建一个名为 **WebExtSample** 的文件夹。
5. 在 **Toolkits\WebExtSample** 文件夹中, 创建以下内容:
 - 一个名为 **WebExtSample.xml** 的文件 (此为工具包配置文件)。
 - 一个名为 **WebExtBook.js** 的文件 (此为设计用于支持 Book 控件的所有 JavaScript 函数的文件)。
 - 一个包含 **WebBook.ico** 图标文件的 **Res** 文件夹 (可从 `%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Res` 复制此图标文件)。

第 2 阶段: 将 WebExtSample 环境引入 UFT 中

在此部分中, 您将使用工具包配置文件和测试对象配置文件, 将 WebExtSample 环境引入 UFT。Web 插件扩展性的第一层是将环境引入 UFT 中。工具包配置文件向 UFT 通知新环境 (及其名称), 测试对象配置文件描述设计用于此环境的测试对象模型。

设计工具包配置文件

工具包配置文件的首要作用是向 UFT 通知新的受支持环境。

要通知 UFT 有新环境受支持, 只要创建与环境同名的基本工具包配置文件即可。基本工具包配置文件必须包含一个 **Controls** 元素和至少一个 **Control** 元素 (描述一个测试对象类)。有关工具包配置文件中元素和属性的详细信息, 请参见“UFT Java Add-in Extensibility Toolkit Configuration Schema Help” (可通过“Web 插件扩展性帮助”获得)。

通知 UFT 关于 WebExtSample 环境的信息:

在 [第 1 阶段: 创建工具包支持集 \(第 69 页\)](#) 中创建的 **WebExtSample.xml** 文件中输入以下文本:

```
<?xml version="1.0" encoding="UTF-8"?>
<Controls>
  <Control TestObjectClass="WebExtBook"/>
</Controls>
```

将此文件部署在 UFT 计算机的正确位置上后, 打开 UFT 时, 将 WebExtSample 环境在插件管理器中显示为“Web Add-in”下的子节点。如果选择了 WebExtSample 的复选框, 则 UFT 加载您为此环境提供的支持。

稍后在此课程中，您将在此 **Control** 元素内添加其他元素，提供完成工具包支持集的 JavaScript 函数的位置以及为以下 UFT 功能提供支持的信息：

- 标识用于表示控件的测试对象类（用于支持 Object Spy 和识别控件）
- 识别控件时筛选子控件
- 侦听控件上的事件以录制测试步骤

设计测试对象配置文件

使用测试对象配置文件将 WebExtSample 环境及其测试对象模型引入到 UFT 中。

TypeInformation 元素中的 **PackageName** 属性将此测试对象配置文件（以及在此配置文件中定义的测试对象）与 WebExtSample 环境关联。如果在 UFT 打开时您没有选择 WebExtSample 环境，则 UFT 将忽略此文件中的测试对象类定义。

有关测试对象配置文件中的元素和属性的详细信息，请参见“UFT Test Object Schema Help”（可通过“Web 插件扩展性帮助”获得）。

在测试对象配置文件中定义 WebExtSample 测试对象模型：

在 [第 1 阶段：创建工具包支持集 \(第 69 页\)](#) 中创建的 **WebExtSampleTestObjects.xml** 文件中输入以下文本：此文本根据 [Web 插件扩展性计划清单 \(第 68 页\)](#) 中描述的详细信息定义 **WebExtSample** 环境和 **WebExtBook** 测试对象类（包括其测试对象方法和标识属性）。

```
<?xml version="1.0" encoding="UTF-8"?>
<TypeInformation Load="true" AddinName="Web"
  PackageName="WebExtSample">
  <ClassInfo BaseClassInfoName="WebElement"
    Name="WebExtBook"
    DefaultOperationName="Select" >
  <IconInfo IconFile="INSTALLDIR\dat\Extensibility\Web\
    Toolkits\WebExtSample\Res\WebBook.ico"/>
  <TypeInfo>
  <Operation ExposureLevel="CommonUsed"
    Name="Select"
    PropertyType="Method">
  <Description>Selects the book.</Description>
  <Documentation>
    <![CDATA[Select the %l book.]]>
  </Documentation>
  </Operation>
  <Operation ExposureLevel="CommonUsed"
    Name="GoToAuthorPage"
    PropertyType="Method">
  <Description>
    Opens the Web page for the specified author.
  </Description>
  <Documentation>
    <![CDATA[Open the Web page for %a1.]]>
  </Documentation>
```

```
<Argument Name="AuthorName"
  IsMandatory="true"
  Direction="In"
  DynamicListOfValues="true">
  <Type VariantType="String"/>
  <Description>The author.</Description>
</Argument>
</Operation>
<Operation ExposureLevel="CommonUsed"
  Name="GoToUsedBooksPage"
  PropertyType="Method">
  <Description>
    Opens the UsedBooks page.
  </Description>
  <Documentation>
    <![CDATA[Open the %l UsedBooks page.]]>
  </Documentation>
</Operation>
</TypeInfo>
<IdentificationProperties>
  <IdentificationProperty ForDefaultVerification="true"
    ForVerification="true"
    ForDescription="true"
    Name="title"/>
  <IdentificationProperty ForDefaultVerification="true"
    ForVerification="true"
    ForDescription="true"
    Name="authors"/>
  <IdentificationProperty ForDefaultVerification="true"
    ForVerification="true"
    ForDescription="false"
    Name="price"/>
  <IdentificationProperty ForDefaultVerification="true"
    ForVerification="true"
    ForDescription="false"
    Name="min_used_price"/>
</IdentificationProperties>
</ClassInfo>
</TypeInfoInformation>
```

第 3 阶段：使 UFT 标识、侦测和识别 Book 控件

要支持自定义控件，UFT 必须能够标识表示给定控件的测试对象类。因此，Web 插件扩展性最基本的元素是在工具包配置文件的每个 **Control** 元素内定义的 **Identification** 元素。每个 **Control** 元素定义一个测试对象类。**Identification** 元素指定该测试对象类表示的控件。

当 UFT 需要识别所测试应用程序中的控件时，它将检查为每个测试对象类定义的 **Identification** 元素。**Identification** 定义与控件匹配的测试对象类将用于表示此控件。

如计划对 [Web 插件扩展性 Book 示例工具包的支持 \(第 64 页\)](#) 中所述, `tagName` 属性是 `table` 并且 `className` 属性是 `Book` 的任何控件都通过 `WebExtBook` 测试对象表示。这可以在工具包配置文件中轻松定义, 无需使用 JavaScript 函数。

定义 `WebExtBook` 测试对象类的标识规则:

请使用以下文本替换 `WebExtSample.xml` 文件中的文本:

```
<?xml version="1.0" encoding="UTF-8"?>
<Controls>
  <Control TestObjectClass="WebExtBook">
    <Identification>
      <Browser name="*">
        <Conditions type="IdentifyIfPropMatch" logic="and">
          <Condition prop_name="tagName" expected_value="TABLE"/>
          <Condition prop_name="className" expected_value="Book"/>
        </Conditions>
      </Browser>
    </Identification>
  </Control>
</Controls>
```

这段代码将 `Identification` 元素添加到定义 `WebExtBook` 测试对象类的 `Control` 元素。标识元素包括一个含有两个条件的 `Conditions` 元素, 控件必须满足这两个条件才能成为 `WebExtBook`。`Conditions` 元素中的每个 `Condition` 元素指定一个条件。在每个条件中, 控件指定 HTML 属性的值必须与指定的期望值匹配 (比较时不区分大小写)。

此教程使用上面的定义说明在 `Conditions` 元素中使用多个 `Condition` 元素的情况。但是, 如果您使用的应用程序页面上有许多控件, 或具有大型 DOM 架构, 则最好如下定义这些标识规则:

```
<Identification>
  <Browser name="*">
    <HTMLTags>
      <Tag name="TABLE"/>
    </HTMLTags>
    <Conditions type="IdentifyIfPropMatch">
      <Condition prop_name="className" expected_value="Book"/>
    </Conditions>
  </Browser>
</Identification>
```

此定义可以提高性能, 因为它指示 UFT 仅标识 `TABLE` 元素。

有关定义测试对象类 `Identification` 元素的详细信息, 请参见[使 UFT 标识用于自定义 Web 控件的测试对象类 \(第 40 页\)](#)和“UFT Java Add-in Extensibility Toolkit Configuration Schema Help” (可通过“Web 插件扩展性 帮助”获得)。

部署和测试工具包支持集（第 3 阶段）

定义测试对象配置文件中的 **WebExtBook** 测试对象类，和工具包配置文件中此测试对象类的标识规则后，可使用 UFT 测试使用工具包支持集的效果。

备注: 开发自己的工具包支持集时，如果要修改测试对象配置文件中 **Identification Property** 元素的属性，则应保持 **TypeInformation** 元素的 **DevelopmentMode** 属性在自定义工具包支持的设计阶段设置为 **true**。在部署自定义工具包支持集进行常规使用之前，请确保删除此属性（或将其设置为 **false**）。执行本教程时，这并非必需操作。有关详细信息，请参见 [修改测试对象配置文件中的 Identification Property 属性 \(第 60 页\)](#)。

部署工具包支持集


1. 将 **WebExtSampleTestObjects.xml** 文件复制到 **<UFT 安装文件夹>\dat\Extensibility\Web** 中。
2. 在 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits** 文件夹中，创建名为 **WebExtSample** 的文件夹。
3. 将 **WebExtSample.xml** 文件复制到 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 文件夹中。
4. 在 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 文件夹中，创建名为 **Res** 的文件夹。
5. 将 **WebBook.ico** 文件放到 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample\Res** 文件夹中。

测试工具包支持集

1. 部署工具包支持集后，打开 UFT。


备注: UFT 将在打开后读取工具包支持文件。因此，如果 UFT 已打开，则必须关闭 UFT，然后再次将其打开。

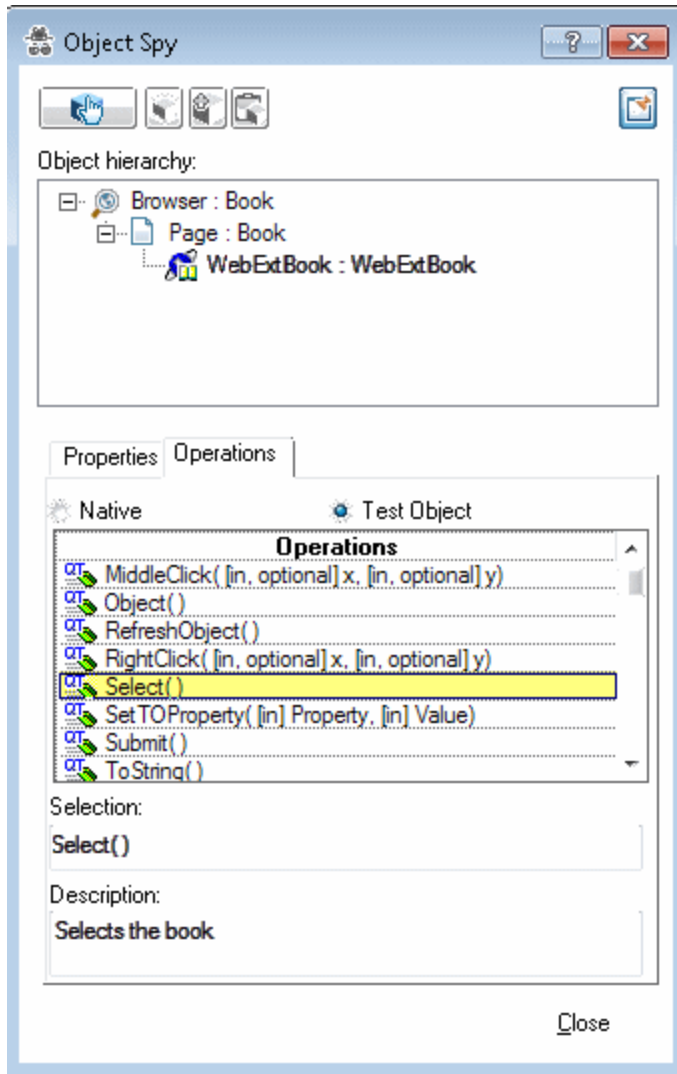
“Add-in Manager”对话框在可用插件列表中将 **WebExtSample** 显示为 Web 环境的子项。（如果“Add-in Manager”对话框未打开，请参见《HP Unified Functional Testing 插件指南》，了解相关说明。）


2. 选中 **WebExtSample** 的复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 使用“Object Repository”对话框中的 **Define New Test Object**  按钮，打开“Define New Test Object”对话框。WebExtSample 环境显示在 **Environment** 列表中。从列表中选择 WebExtSample 环境时，您在测试对象配置文件中定义的 **WebExtBook** 测试对象类将显示在 **Class** 列表中，带有在 **WebExtSampleTestObjects.xml** 文件中指定的图标。
4. 通过打开 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** 文件运行示例控件。

备注: UFT 会在应用程序打开时与其建立连接。因此，如果 Book 控件已打开，则必须将其关闭然后再次运行。

在 UFT 中，在 Book 控件上执行以下活动，以查看 UFT 识别控件的方式。（有关使用 UFT 的详细信息，请参见《HP Unified Functional Testing 用户指南》。）

- 使用 Object Spy  查看 Book 控件支持的标识属性和测试对象操作：
 - 为 Book 控件创建的 **WebExtBook** 测试对象将以其测试对象类的名称命名。稍后在此课程中，您可以通过提供更具体的名称对工具包支持集进行自定义。
 - 测试对象操作列表包含从 **WebElement** 基类继承的所有操作（方法和属性），以及在 **WebExtSampleTestObjects.xml** 测试对象配置文件中定义的所有方法。



- 标识属性列表包含在 **WebExtSampleTestObjects.xml** 测试对象配置文件中定义的所有属性。属性值未显示，因为您尚未实现从应用程序返回属性值的方法（**WebElement** 基类不支持这些属性。）稍后您将在此课程中实现此方法。
- 使用“Object Repository”对话框中的 **Add Objects to Local**  按钮识别 Book 控件。确保使用正确的图标表示对象存储库中的测试对象。
- 在“Keyword View”中，创建从 **Item** 列中的对象存储库选择 **WebExtBook** 对象的测试步骤。

- **Operation** 列中的可用操作列表反映了测试对象配置文件中的定义。
 - 选择操作后, UFT 在 **Value** 单元格的工具提示中显示操作参数的“Name”属性。如果定义具有多个参数的操作, 则选择操作后, **Value** 单元格将根据操作的参数个数进行分区。
 - 您在测试对象配置文件中为测试对象方法定义的描述和文档字符串分别显示在工具提示和 **Documentation** 列中。
- 在编辑器中, 创建带有 WebExtBook 测试对象的测试步骤。语句完成功能将基于测试对象配置文件中的定义显示测试对象可用的所有操作。

如果为操作参数定义了可能的值 (在 ListOfValues 元素中), 则这些值也将显示在语句完成列表中。稍后在此课程中, 您将学习如何启用控件以动态提供可能的参数值列表。您将开发支持以提供可用于 **GoToAuthorPage** 操作的 **AuthorName** 参数的作者列表。

5. 使用在 WebExtBook 测试对象上执行新测试对象方法的步骤运行测试。UFT 搜索将在控件上运行测试对象方法的 JavaScript 函数。因为尚未实现运行测试对象方法的支持, 所以将出现运行时错误。在 **next** 部分, 实现此支持。

第 4 阶段: 实现对 WebExtBook 测试对象方法的支持

使 UFT 能够识别自定义控件后, 必须提供运行测试对象方法的支持。对于在测试对象配置文件中定义的每个测试对象方法, 都必须编写 JavaScript 函数, 供 UFT 在控件上运行以执行步骤。

在工具包配置文件中, 您需要指定 UFT 将在其中查找 JavaScript 函数的 JavaScript 文件, 还可以选择指定每个测试对象方法使用的函数名称。

在此部分中, 您将提供对 WebExtBook 测试对象方法的支持: **Select**、**GoToAuthorPage(AuthorName)** 和 **GoToUsedBooksPage**。

可以为工具包配置文件中的每个测试对象方法指定 JavaScript 文件和函数。但在此课程中, 您将使用最简单的方法开发对运行测试对象方法的支持。在 **Control** 元素级别, 定义 UFT 针对所有在此元素中定义的测试对象方法默认使用的 JavaScript 文件。对于 JavaScript 函数名, 默认情况下, 在指定文件中搜索与测试对象方法具有相同名称的 JavaScript 函数。因此, 您无需在工具包配置文件中指定函数名, 只需创建具有正确名称的 JavaScript 函数即可。

开发对 WebExtBook 测试对象方法的支持:

1. 在 **WebExtSample.xml** 文件中, 在为 **WebExtBook** 测试对象类定义的 **Control** 元素内, 向 UFT 添加以下 **Settings** 元素:

```
<Settings>
  <Variable name="default_imp_file" value="WebExtBook.js"/>
</Settings>
```

这段代码指示 UFT 在 **WebExtBook.js** 文件 (位于 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 文件夹) 中搜索 JavaScript 函数。

备注: 您可以修改工具包支持集文件夹中的 **WebExtSample.xml** 文件, 并在稍后将其部署到 UFT 进行测试, 或者直接修改 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample.xml**。

2. 在 **第 1 阶段: 创建工具包支持集 (第 69 页)** 中创建的 **WebExtBook.js** 文件中, 粘贴下方文本为每个测试对象方法创建 JavaScript 函数: **Select**、**GoToAuthorPage(AuthorName)** 和 **GoToUsedBooksPage**。

备注: `_elem` 对象是 UFT 用于参考当前正在处理的 HTML 控件的保留对象。

```
// Run implementation
// ~~~~~
// This section contains the functions that carry out the
// test object methods.
function Select()
{ // Click the link in the second cell of the first row.
  _elem.rows[0].cells[1].children[0].click();
}
function GoToAuthorPage(AuthorName)
{ // Look for the specified author name among the children
  // of the first cell in the second row and click it.
  var bWasFound = false;
  for( var i = 0 ; i <&lt; _elem.rows[1].cells[0].children.length ; ++i )
  {
    if( _elem.rows[1].cells[0].children[i].innerText == AuthorName )
    {
      _elem.rows[1].cells[0].children[i].click();
      bWasFound = true;
      break;
    }
  }
  if( bWasFound == false )
    throw ("Author name not found !");
}
function GoToUsedBooksPage()
{ // Click the link in the first cell of the third row.
  _elem.rows[3].cells[0].children[1].click();
}
```

部署和测试工具包支持集 (第 4 阶段)

开发对运行测试对象方法的支持后, 将更新后的工具包支持集部署到 UFT 并进行测试。

测试对运行测试对象方法的支持:

1. 要将更新后的工具包支持集部署到 UFT 上, 请将 **WebExtBook.js** 文件 (必要时和 **WebExtSample.xml** 文件一起) 复制到 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框, 然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行示例控件。
4. 打开 GUI 测试并使用“Object Repository”对话框中的 **Add Objects to Local**



按钮识别 Book 控件。

5. 创建带有以下步骤的测试，然后运行此测试：

```
Browser("Book").Page("Book").WebExtBook("WebExtBook").GoToAuthorPage "Jane Doe"
```

备注: 如果运行的 **GoToAuthorPage** 测试对象方法带有控件中不存在的作者姓名，则 JavaScript 函数抛出异常，UFT 显示运行时错误消息，测试步骤失败。

创建并运行类似测试以测试 **Select** 和 **GoToUsedBooksPage** 测试对象方法。

第 5 阶段：实现对 WebExtBook 标识属性的支持

在此部分中，您将实现在测试运行期间检索标识属性值的支持。UFT 在不同的标准测试对象方法（如 **GetROProperty**）中使用标识属性运行时值。各种基本功能（如创建检查点和输出值）也需要使用标识属性运行时值。

要支持检索标识属性的运行时值，需要实现接受 **PropertyName** 参数并返回 UFT 请求的任意属性值的 JavaScript 函数。必须实现此方法以返回在测试对象配置文件中定义的每个标识属性的值。

在工具包配置文件中，您可以指定实现检索属性值的 JavaScript 函数的 JavaScript 文件。还可以指定为此实现的函数的名称。但是，如果未指定函数名称，UFT 将调用 **get_property_value (PropertyName)**，此为必须实现的函数。如果未指定文件名，UFT 将从您在 **Control\Settings** 元素中指定的 JavaScript 文件中调用函数。因此，在此课程，您将在 **WebExtBook.js** 文件中创建 **get_property_value** 函数。

支持检索 WebExtBook 标识属性的运行时值：

将以下行添加到 **WebExtBook.js** 文件中：

```
// Property retrieval implementation
// ~~~~~
// The function provides values for all of the identification
// properties defined in the test object configuration XML file.
function get_property_value(prop)
{
    if ( prop == "title" )
        // For the "title" identification property,
        // Return the inner text of the second cell in the first row
        {
            return _elem.rows[0].cells[1].innerText;
        }
    if ( prop == "authors" )
        // To return a list of all the authors, look for all the
        // children of the first cell in the second row.
        {
            var BookAuthors = "";
            var AuthorsCount = 0;
            for( var i = 0 ; i < _elem.rows[1].cells[0].children.length ; ++i )
```

```
{
  if( _elem.rows[1].cells[0].children[i].tagName == "A" )
  {
    if( AuthorsCount > 0 )
      BookAuthors += ",";
    BookAuthors += _elem.rows[1].cells[0].children[i].innerText;
    AuthorsCount++;
  }
}
return BookAuthors;
}
if ( prop == "price" )
// To return the price of the book, return the innerText
// property of the first cell in the fourth row.
{
  return _elem.rows[3].cells[0].children[0].innerText;
}
if ( prop == "min_used_price" )
// To return the lowest price available for a used copy of the
// book, return the innerText property of the second child of
// the first cell in the fourth row.
{
  if( _elem.rows[3].cells[0].children.length > 2 )
    return _elem.rows[3].cells[0].children[2].innerText;
}
}
```

备注: 您可以修改工具包支持集文件夹中的 **WebExtBook.js** 文件，并在稍后将其部署到 UFT 进行测试，或者直接修改 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample\WebExtBook.js**。

部署和测试工具包支持集（第 5 阶段）

开发对检索标识属性运行时值的支持后，将更新后的工具包支持集部署到 UFT 并进行测试。

测试对检索标识属性运行时值的支持：

1. 确保 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行示例控件。
4. 新建 GUI 测试，将 **WebExtBook** 测试对象添加到对象存储库，然后使用此测试对象创建测试步骤。右键单击该对象，然后选择 **Insert Standard Checkpoint**。将打开“Checkpoint Properties”对话框。请确保在测试对象配置文件中定义的标识属性（**title**、**authors**、**price** 和 **min_used_price**）包含在属性列表中并已选中。

5. 创建并运行测试，检索每个标识属性并检查属性的值，或在消息框中显示该属性。例如，可以运行以下测试：

Item	Operation	Value
▼ Action1		
▼ Book		
▼ Book		
WebExtBook	CheckProperty	"authors","Jane Doe,John Doe"
WebExtBook	Check	CheckPoint("WebExtBook")
Function Call	MsgBox	Browser("Book").Page("Book").WebExtBook("WebExtBook").GetROProperty("title")

第一步先检查“authors”属性的值，在第二步中检查点检查在检查点内选中的属性（这种情况下为“price”和“min_used_price”），第三步在消息框中显示书籍标题。

6. 单击 **OK** 关闭消息框。测试运行完成，将显示运行结果。展开运行结果树查看步骤详细信息。

第 6 阶段：更改测试对象的名称

在此部分中，您将按照自己的计划（[计划对 Web 插件扩展性 Book 示例工具包的支持 \(第 64 页\)](#)），将工具包支持集修改为指示 UFT 根据其标题对 WebExtBook 测试对象进行命名。

当 UFT 创建表示控件的测试对象时，将调用具有参数 **logical_name** 的 **get_property_value** 函数（用于检索测试对象标识属性值）以决定测试对象的名称。您可以相应地实现 **get_property_value** 函数，自定义 UFT 为测试对象提供的名称。如果 **get_property_value** 函数不支持 **logical_name** 属性，则测试对象将以测试对象类命名（如果同一页上有多个测试对象类相同的对象，则后跟索引）。

自定义测试对象的名称：

在 **WebExtBook.js** 文件的 **get_property_value** 函数中，将以下行：

```
if ( prop == "title" )
// For the "title" identification property,
```

替换为：

```
if ( prop == "logical_name" || prop == "title" )
// For the "title" identification property, as well as
// the "logical_name" property
```

备注：您可以修改工具包支持集文件夹中的 **WebExtBook.js** 文件，并在稍后将其部署到 UFT 进行测试，或者直接修改 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample\WebExtBook.js**。

get_property_value 函数现在将针对 UFT 用于命名测试对象的 **logical_name** 属性返回相同的文本，正如对 **title** 标识属性一样。（相应地修改解释此函数的注释。在注释末尾，添加以下内容：as well as the hard coded "logical_name" property that UFT uses to name the test object）。

部署和测试工具包支持集（第 6 阶段）

开发对表示控件的测试对象进行命名的支持后，将更新后的工具包支持集部署到 UFT 并进行测试。

测试对命名 WebExtBook 测试对象的支持:

1. 确保 <UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行示例控件。
4. 打开 GUI 测试，打开“Object Repository”，然后使用“Add Object to Local”按钮识别 Book 控件。确保将 UFT 创建的测试对象命名为 **The History of QuickTest Professional**。

第 7 阶段：实现防止识别子对象的筛选

指示 UFT 识别网页时，将打开“Define Object Filter”对话框，用于确定应识别哪些网页的子网页。选择 **All object types** 会指示 UFT 识别 WebExtBook 控件及其父网页，并将 WebExtBook 控件包含的所有控件识别为该网页的子网页（以及 WebExtBook 控件的同级控件）。

对于 Book 控件，无需为其所有子控件创建测试对象，如 [计划对 Web 插件扩展性 Book 示例工具包的支持 \(第 64 页\)](#) 中所述。

要防止 UFT 识别 Web 插件扩展性支持的控件的所有子控件，可以定义识别筛选。复杂筛选可使用 JavaScript 函数实现，这样就可以在工具包配置文件中指定函数的位置和名称。简单筛选可直接在工具包配置文件中实现，而无需使用 JavaScript 函数。

要防止识别 Book 控件中包含的控件，使用简单筛选即可。实现此筛选前，识别包含 Book 控件及其所有子控件的网页以查看是否也识别了所有 Book 子控件。要执行此操作，请执行 [部署和测试工具包支持集 \(第 7 阶段\) \(第 81 页\)](#) 中所述的步骤。

防止识别 Book 控件包含的控件:

在 **WebExtSample.xml** 文件中为 **WebExtBook** 测试对象类定义的 **Control** 元素内，添加以下 **Filter** 元素:

```
<Filter>
  <Learn learn_control="Yes" learn_children="No"/>
</Filter>
```

这段代码指示 UFT 在识别 WebExtBook 测试对象的父网页时识别这些对象，但不识别其包含的子控件。


备注: 您可以修改工具包支持集文件夹中的 **WebExtSample.xml** 文件，并在稍后将其部署到 UFT 进行测试，或者直接修改 <UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample.xml。

部署和测试工具包支持集 (第 7 阶段)

定义筛选以防止识别子项后，您可以将更新后的工具包支持集部署到 UFT 并进行测试。

测试对识别无子对象的 WebExtBook 测试对象的支持:

1. 确保 <UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。

3. 关闭并重新运行示例控件。
4. 打开 GUI 测试并打开“Object Repository”。使用“Object Repository”对话框中的 **Add Objects to Local**  按钮识别包含 Book 控件的网页。将打开“Define Object Filter”对话框。
5. 选择 **All object types** 并单击 **OK**。名为 **The History of QuickTest Professional** 的 WebExtBook 对象将添加到对象存储库中，但不添加它所包含的控件。

第 8 阶段：实现在 Book 控件上进行录制的支持

在此教程中，目前为止您的工具包支持集已启用完整的 UFT 功能。UFT 可识别 Book 控件并在该控件上运行测试。

另外，在 UFT 中创建测试的可选方法是通过录制用户在应用程序上执行的操作。您可以在[计划对 Web 插件扩展性 Book 示例工具包的支持 \(第 64 页\)](#)中看到，默认情况下，UFT 将录制 Book 控件中各种 Web 链接和图像对象的普通 **Click** 操作。相对于响应这些相同的单击，录制 Book 控件自身上的 **Select**、**GoToAuthorPage** 和 **GoToUsedBooksPage** 操作可能会更有帮助。

要支持在控件上进行自定义录制，必须指示 UFT 侦听相关事件并通知 UFT 针对每个事件录制哪些测试步骤。

为此，您需要编写两种类型的 JavaScript 函数：

- 一个 JavaScript 函数，它使用 UFT 在 Web 插件扩展性 SDK 中公开的 **_util** 实用程序对象中的 **RegisterForEvent** 函数注册，以在正确的元素上侦听正确的事件。此函数的参数也决定了在每个事件发生时 UFT 将调用的 JavaScript 函数。
在工具包配置文件中，指定此 JavaScript 函数的名称和（可选）其位置。
- 一个或多个处理事件的 JavaScript 函数通过调用 **_util** 实用程序对象中的 **Record** 函数通知 UFT 关于添加到测试中的步骤。

备注: **Record** 函数，以及其他实用程序对象函数需要 **SafeArray** 类型参数。要将数组转换为 **SafeArray**，可使用 Web 插件扩展性提供的 **toSafeArray (array)** 函数。此函数在 **<Extensibility Accelerator 安装文件夹>\bin\PackagesToLoad\common.js** 中定义。（此文件也位于 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits folder** 中。）

有关实用程序对象函数的语法信息，请参见“UFT Java Add-in Extensibility API Reference”中的 **_util** 部分（可通过“Web 插件扩展性帮助”获得。）

开发在 Book 控件上进行录制的支持：

备注: 您可以修改工具包支持集文件夹中的 **WebExtSample.xml** 和 **WebExtBook.js** 文件，并在稍后将其部署到 UFT 进行测试，或者直接修改 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample**。

1. 在工具包配置文件中的 **Control** 元素内添加以下 **Record\EventListening** 元素：

```
<Record>
  <EventListening use_default_event_handling_for_children="false"
    use_default_event_handling="false"
    type="javascript">
```

```
function="ListenToEvents"/>  
</Record>
```

这段代码指示 UFT 在 Book 控件及其子控件上不使用默认“Web Event Configuration”处理事件，而是调用 JavaScript 函数 **ListenToEvents**。因为您未指定 JavaScript 文件，所以 UFT 将在您在 WebExtBook 测试对象类的 **Control** 级别指定的 **WebExtBook.js** 文件中查找 JavaScript 函数。

2. 在 **WebExtBook.js** 文件中，添加以下 **ListenToEvents** 函数：

```
function ListenToEvents( elem )  
{  
    // Connect to the "Select" event:When the book name or the  
    // book icon is clicked, call OnSelectClicked.  
    _util.RegisterForEvent( _elem.rows[0].cells[0].children[0], "onclick", "OnSelectClicked");  
    _util.RegisterForEvent( _elem.rows[0].cells[1].children[0], "onclick", "OnSelectClicked");  
    // Connect to the "Author" event:When an author name is  
    // clicked, call OnAuthorClicked.  
    for( var i = 0 ; i < _elem.rows[1].cells[0].children.length ; ++i )  
    {  
        if( _elem.rows[1].cells[0].children[i].tagName == "A" )  
        {  
            _util.RegisterForEvent( _elem.rows[1].cells[0].children[i], "onclick", "OnAuthorClicked" );  
        }  
    }  
    // Connect to the "UsedBooks" event:When "Used" is  
    // clicked, call OnUsedBooksClicked.  
    if( _elem.rows[3].cells[0].children.length > 1 )  
        _util.RegisterForEvent( _elem.rows[3].cells[0].children[1], "onclick", "OnUsedBooksClicked"  
    );  
    return true;  
}
```

此函数注册 UFT 以侦听书籍标题、图像和作者以及 **Used** 链接上的单击事件。注册事件时，此函数将指定 UFT 在事件发生时必须调用的 JavaScript 函数。

3. 在 **WebExtBook.js** 文件中, 添加以下事件处理程序 JavaScript 函数:

```
function OnSelectClicked( handlerParam , eventObj )
{
    // Record the "Select" step
    var arr = new Array();
    _util.Record( "Select", toSafeArray(arr) , 0 );
    return true;
}
```

```
function OnAuthorClicked( handlerParam , eventObj )
{
    // Record the "GoToAuthorPage" step
    var arr = new Array();
    arr[0] = eventObj.srcElement.innerText;
    _util.Record( "GoToAuthorPage", toSafeArray(arr) , 0 );
    return true;
}
```

```
function OnUsedBooksClicked( handlerParam , eventObj )
{
    // Record the "GoToUsedBooksPage" step
    var arr = new Array();
    _util.Record( "GoToUsedBooksPage", toSafeArray(arr) , 0 );
    return true;
}
```

这些函数将按计划对 [Web 插件扩展性 Book 示例工具包的支持 \(第 64 页\)](#) 中的计划录制 WebExtBook 测试对象上的 **Select**、**GoToAuthorPage** 和 **GoToUsedBooksPage**。

部署和测试工具包支持集 (第 8 阶段)

开发在 Book 控件上进行录制的支持后, 将更新后的工具包支持集部署到 UFT 并进行测试。

测试在 Book 控件上执行的录制操作的支持:

1. 确保 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框, 然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行示例控件。
4. 打开 GUI 测试并单击 **Record** 按钮或选择 **Record > Record**。单击 Book 控件中不同的链接 (每次单击后必须返回到上一页以返回到 Book 控件): 书籍标题、控件中的图像、作者姓名和 **Used** 链接。

每次单击都会在测试中添加一个新步骤:

Item	Operation	Value	Documentation
▼ Action1			
▶ Book			
▶ Book			
▶ The History of QuickTest	Select		Select the "The History of QuickTest" book.
▶ Not Implemented	Sync		Wait for the Web page to synchronize before continuing.
▶ Book	Back		Navigate back to the previous page of the browser.
▶ Book			
▶ The History of QuickTest	Select		Select the "The History of QuickTest" book.
▶ Not Implemented	Sync		Wait for the Web page to synchronize before continuing.
▶ Book	Back		Navigate back to the previous page of the browser.
▶ Book			
▶ The History of QuickTest	GoToAuthorPage	"Jane Doe"	Open the Web page for "Jane Doe".
▶ Not Implemented_2	Sync		Wait for the Web page to synchronize before continuing.
▶ Book	Back		Navigate back to the previous page of the browser.
▶ Book			
▶ The History of QuickTest	GoToUsedBooksPage		Open the "The History of QuickTest" used books page.

单击 **Stop** 按钮或选择 **Record > Stop** 以结束会话录制。

第 9 阶段: 实现对 AuthorName 值动态列表的支持

使用 Web 插件扩展性, 您可以根据特定控件的运行时值为 UFT 用户提供一列可用于测试对象方法参数的值。例如, **WebExtBook** 测试对象类的 **GoToAuthorPage** 测试对象方法将接收 **AuthorName** 参数。如果 UFT 用户可从可能值的列表选择作者姓名而非输入作者姓名, 则操作会更容易一些。但是, 每个 **WebExtBook** 控件的此列表各不相同。

在测试对象配置文件中, 将 **AuthorName** 参数的 **DynamicListOfValues** 属性定义为 **true**, 指示 UFT 在创建测试步骤时从控件请求可能值的列表。

在工具包配置文件中, 您可以指定文件名和 UFT 必须调用以检索值列表的 JavaScript 函数的函数名。默认情况下, UFT 通过从在 **WebExtBook** 测试对象类的 **Control** 级别指定的 **WebExtBook.js** 文件中调用 **get_list_of_values** JavaScript 函数来请求值列表。UFT 为每个其 **DynamicListOfValues** 属性在测试对象配置文件中被设置为 **true** 的参数调用 JavaScript 函数。为此函数提供的参数指示请求值的测试对象方法和参数。

在此部分, 您将实现 **get_list_of_values** JavaScript 函数, 从 **Book** 控件返回作者姓名。

在 GoToAuthorPage 测试对象方法中提供 AuthorName 参数值的动态列表:

在 **WebExtBook.js** 文件中, 添加 JavaScript 函数:

```
// Dynamic list of values implementation
// ~~~~~
function get_list_of_values( method, argIndex )
{
    // When creating a step with the GoToAuthorPage test
    // object method, provide a list of the authors of this book
    // that can be used for the method's argument.
    if (method == "GoToAuthorPage")
    {
```

```
    return get_GoToAuthorPage_list_of_values(argIndex);
  }
  return null;
}
function get_GoToAuthorPage_list_of_values(argIndex)
{
  var arr = new Array();
  if( argIndex > 1 )
    return toSafeArray(arr);
  // Retrieve all authors
  var AuthorsCount = 0;
  for( var i = 0 ; i < _elem.rows[1].cells[0].children.length ; ++i )
  {
    if( _elem.rows[1].cells[0].children[i].tagName == "A" )
    {
      arr[AuthorsCount]="\""+_elem.rows[1].cells[0].children[i].innerText+"\"";
      AuthorsCount++;
    }
  }
  return toSafeArray(arr);
}
```

此函数将返回书籍作者的列表，每个作者均放在引号中。

根据您的决定的有关计划自定义支持的时间规范，现在已完全支持 Book 自定义控件。

备注: 您可以修改工具包支持集文件夹中的 **WebExtSample.xml** 文件，并在稍后将其部署到 UFT 进行测试，或者直接修改 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample.xml**。

部署和测试工具包支持集（第 9 阶段）

实现 **get_list_of_values** JavaScript 函数后，将更新后的工具包支持集部署到 UFT 并测试作者姓名的动态列表是否正确提供。

测试在 Book 控件上执行的录制操作的支持：

1. 确保 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行示例控件。
4. 在编辑器中打开 GUI 测试，使用 **WebExtBook** 测试对象和 **GoToAuthorPage** 测试对象方法创建一个步骤。在方法名称后输入空格，将在下拉列表中显示 Book 控件上列出的、要用于方法参数的作者姓名。

课程总结

在此课程，您创建了新测试对象类 `WebExtBook`，定义了其标识属性和测试对象方法。创建了对 `Book` 控件的支持，支持 UFT 将其识别为 `WebExtBook` 测试对象。

- 您已学习了解测试对象配置文件。
- 学习并理解了工具包配置文件。
- 学习了如何支持新标识属性和测试对象方法。
- 学习了如何创建防止识别子控件的筛选。
- 学习了如何支持录制和使用 **Record** 和 **RegisterForEvent** 实用程序方法。
- 学习了如何为测试对象参数提供动态值列表。

接下来要执行的操作

有关工具包支持集的结构和内容的详细信息，请参见[开发工具包支持 \(第 25 页\)](#)。

有关测试对象配置文件的结构和内容的详细信息，请参见“UFT Test Object Schema Help（可通过“Web 插件扩展性帮助”获得）”。

有关工具包配置文件的结构和内容的详细信息，请参见“UFT Java Add-in Extensibility Toolkit Configuration Schema Help”（可通过“Web 插件扩展性帮助”获得）。

有关 `_util` 实用程序对象和全局 JavaScript 方法的详细信息，请参见“UFT Java Add-in Extensibility API Reference”（可通过“Web 插件扩展性帮助”获得）。

在接下来的课程中，您将学习如何创建对 `UsedBooks` 自定义控件的支持。表示 `UsedBooks` 控件的测试对象类将扩展现有的 `WebTable` 测试对象类。在为此控件开发 UFT 支持时，您将学习使用 Web 插件扩展性提供的一些更高级的选项。

第 6 章: 学习为复杂的自定义 Web 控件创建 UFT 支持

在此课程，您将在随 Extensibility Accelerator for HP Functional Testing 安装的 Web 插件扩展性 Book 示例工具包中创建对 UsedBooks 控件的支持。表示 UsedBooks 控件的测试对象类将扩展现有的 WebTable 测试对象类。“创建对 UsedBooks 控件的支持”将教您如何使用 Web 插件扩展性的一些更高级选项。

在[学习为简单的自定义 Web 控件创建 UFT 支持 \(第 63 页\)](#)此课程中，您将学会创建对简单自定义控件的支持。现在，您已熟悉 Web 插件扩展性的基本知识，因此本课程仅介绍更高级的信息。

%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample 文件夹包含此示例的完整工具包支持集，学习此课程时可供参考。JavaScript 代码与您创建的代码不同，因为示例支持集设计用于在 Firefox 和 Internet Explorer 上运行，并使用 **jQuery** JavaScript 库。

此课程包括：

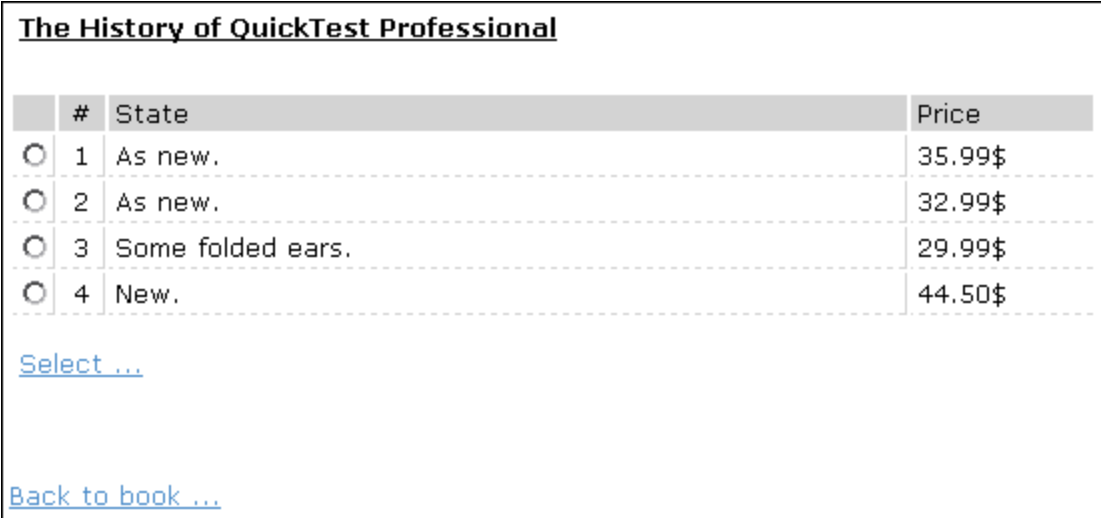
- [本课程的准备工作](#) 89
- [计划对 Web 插件扩展性 示例 UsedBooks 控件的支持](#) 89
- [开发工具包支持集](#) 94
- [课程总结](#) 105

本课程的准备工作

扩展 UFT 对自定义控件的支持之前，必须能够访问其源文件。您无需修改自定义控件的任何源，即可在 UFT 中支持该控件，但您需要熟悉这些源。确保您知道组成控件的元素和属性、此控件上可能发生的事件等。设计支持时请使用此信息。

UsedBooks 控件的源文件位于 `%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\UsedBooks.htm`。

要运行示例应用程序，请打开 `%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm` 文件。此时 Book 控件将打开。在 Book 控件中，单击 **Used** 运行 UsedBooks 控件。



#	State	Price
<input type="radio"/> 1	As new.	35.99\$
<input type="radio"/> 2	As new.	32.99\$
<input type="radio"/> 3	Some folded ears.	29.99\$
<input type="radio"/> 4	New.	44.50\$

[Select ...](#)

[Back to book ...](#)

运行控件，打开其源文件，研究控件的行为和实现。

UsedBooks 控件以 `div` 元素的形式实现，由 Web 表组成，Web 表包含有关此手册的可用旧版和单选按钮的信息，以及用于从列表选择手册的 **Select** 链接（表外元素）的信息。选择手册并打开关于选定手册的页面（不在此示例中实现）需要选择表中相关行的单选按钮，然后单击 **Select**。

计划对 Web 插件扩展性 示例 UsedBooks 控件的支持

在此部分，您可以根据所掌握的控件知识，分析 UFT 当前识别 UsedBooks 控件的方式与识别控件应采用的方式。接下来，确定了解 [Web 插件扩展性 计划清单 \(第 22 页\)](#) 中问题的答案，并相应地填写在 [Web 插件扩展性 计划清单 \(第 23 页\)](#) 中。

执行此操作的最佳方式是一方面使用“Object Spy”、“Keyword View”和“Record”选项从 UFT 角度分析 UsedBooks 控件，另一方面考虑控件的实现方式和控件的使用目的。

1. 打开 UFT 并运行 UsedBooks 控件。

打开 UFT 并加载 Web 插件。

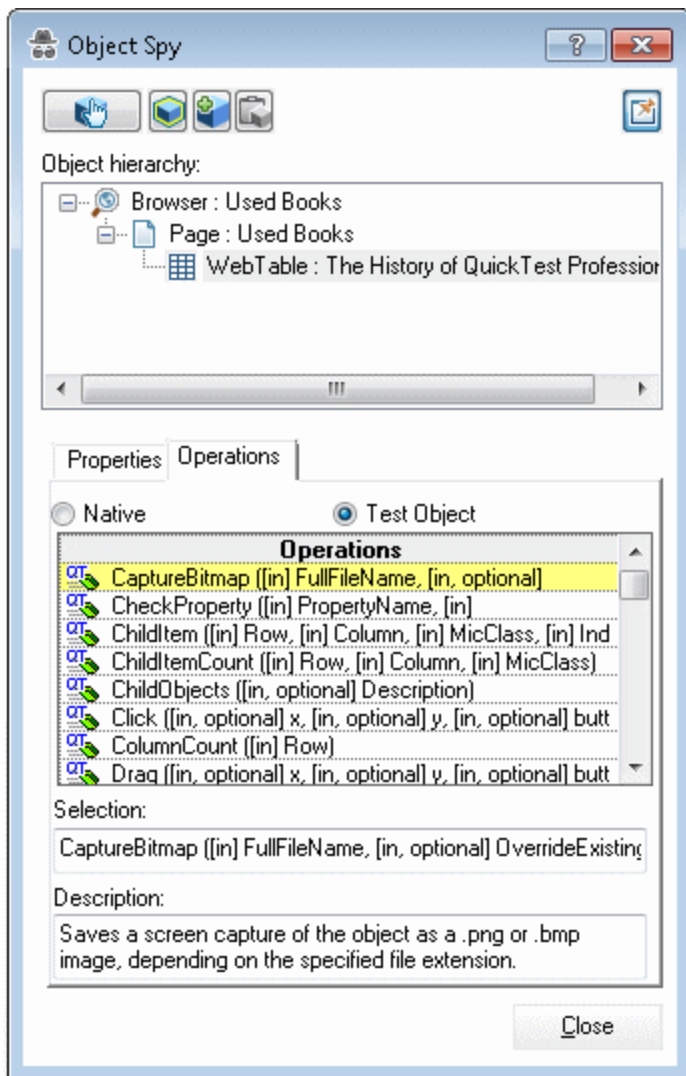
关闭 UsedBooks 控件的任何已打开实例，然后通过打开 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** 文件并在打开的 Book 控件中单击 **Used** 将其打开。

2. 使用 Object Spy 查看 UsedBooks 测试对象操作。

在 UFT 中，打开 GUI 测试并选择 **Tools > Object Spy**，或单击 **Object Spy** 工具栏按钮  以打开“Object Spy”对话框。单击 **Operations** 选项卡并选择 **Test Object Operations**。

在“Object Spy”对话框中，单击指向手 ，然后单击“UsedBooks”表。

UsedBooks 控件包含 Web 表，因为内嵌有 UFT 支持，所以该控件将识别为 **WebTable**，并根据表的标题命名。测试对象使用的图标是标准 WebTable 类图标。UFT 将忽略 **div** 元素，实际上该元素是 UsedBooks 控件的根。



关闭 Object Spy。

3. 在 UsedBooks 控件上录制操作。

在 UFT 中, 选择 **Run > Run Settings** 或 **Record > Record Settings** 打开 “Record and Run Settings” 对话框。在 “Web” 选项卡中, 选择 **Record and run test on any open browser**。单击 **OK**。

单击 **Record** 按钮, 或选择 **Record > Record**。在 “UsedBooks” 表中, 选择某个单选按钮, 然后单击 **Select**。

每次单击都会在测试中添加一个新步骤:

Item	Operation	Value	Documentation
<ul style="list-style-type: none"> ▼ Action1 <ul style="list-style-type: none"> Used Books <ul style="list-style-type: none"> Used Books <ul style="list-style-type: none"> SelectedBook Select ... 	Select	"#0"	Select the "#0" radio button in the "SelUsedBook" radio button group.
	Click		Click the "Select ..." link.

单击 **Stop** 按钮或选择 **Record > Stop** 以结束会话录制。

录制的步骤分别反映单选按钮的选择和链接的单击, 并且不会将这些操作识别为与 UsedBooks 控件相关。

4. 确定 UsedBooks 控件所属的自定义工具包。

为控件扩展 UFT 支持时, 您始终在工具包的上下文中执行此操作。在此教程中, 将对两个自定义 Web 控件进行分组, 组成名为 WebExtSample 的自定义工具包: “Book” 和 “UsedBooks”。

在上一课中已为此工具包创建了工具包支持集。在此课程中, 将对 UsedBooks 控件的支持添加到 WebExtSample 工具包支持集中。

5. 完成自定义控件支持计划清单。

此部分描述了在计划 UsedBooks 控件的支持时需要做出的决定, 然后总结了支持计划清单中的信息。

a. 选择表示自定义控件的测试对象类:

由于 UsedBooks 控件包含的信息类型, UsedBooks 控件的内部内容将实现为 Web 表控件。如果要在 UsedBooks 控件上执行测试并检查该控件所包含的信息, 则 UFT 应将此控件识别为表。但是, 为了实现对 UsedBooks 控件的最佳支持, 表示该控件的测试对象必须支持 **SelectBook** 测试对象方法, 该测试对象方法通过选择表中正确行内的单选按钮并单击 **Select** 从表中选择手册。

另外, 因为 “UsedBooks” 表的第一行含有列名称, 所以这有助于替换 (或覆盖) WebTable 对象支持的 **RowCount** 测试对象方法, 从而减少行数并返回此手册的已用副本数。要支持 **SelectBook** 测试对象方法并覆盖 **RowCount** 的实现, 应新建名为 **WebExtUsedBooks** 并扩展 **WebTable** 的测试对象类。然后使 UFT 将其标识为表示 UsedBooks 控件的测试对象类。

b. 定义 UFT 如何标识用于表示控件的测试对象类:

如果满足以下条件, 则使用 **WebExtUsedBooks** 测试对象表示控件:

- 控件的 **tagName** 属性是 **div**。
- 控件第一个子控件的 **tagName** 属性是 **table**。
- 控件第一个子控件的 **className** 属性是 **UsedBooks**。

c. 确定新测试对象类的详细信息:

- 新测试对象类由以下图标文件表示: **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample\Res\WebBookList.ico**
- 未提供任何帮助文件。
- **WebExtUsedBooks** 测试对象类需要支持 **title** 标识属性 (用于唯一地标识控件, 在检查点属性对话框中默认选中, 但不用于智能标识)。
- 测试对象自身的名称应与其 **title** 标识属性相同。

d. 决定自定义控件支持的测试对象方法:

WebExtUsedBooks 测试对象类需要支持 **WebTable** 测试对象类支持的所有测试对象操作。另外, 它需要支持 **SelectBook** 测试对象方法。

- 所有从基类、WebTable 继承的操作均可受 UsedBooks 控件中包含的 **table** 元素支持。但是, 因为 **table** 元素不是 UsedBooks 控件的根元素, 所以 UFT 不会将此元素识别为基本元素。必须实现将 **table** 元素作为 UsedBooks 控件的基本元素返回的 JavaScript 函数。此函数指示 UFT 使用 **table** 元素支持从基本 WebTable 测试对象类继承的操作。
- **SelectBook** 测试对象方法模拟选择指定手册的单选按钮并单击 **Select**。
- 为了返回表中实际的手册数而非行数, 需要覆盖 WebTable 测试对象方法 **RowCount**。

e. 定义 UFT 识别控件时应识别的子控件:

在此教程中, 当将 **WebExtUsedBooks** 测试对象识别为网页的一部分时, 也应识别其中的单选按钮。

f. 决定 Object Spy 是否应显示 **WebExtUsedBooks** 测试对象: 是。

g. 决定是否支持录制, 以及录制的事件:

侦听 **Select** 链接上发生的鼠标单击。当选中单选按钮并单击此链接时, 记录选择手册的测试步骤 (该手册的单选按钮已选中)。

h. 决定需要在工具包配置文件中设计的支持部分以及需要 JavaScript 函数的部分:

- 对于 UsedBooks 控件, 测试对象类标识由工具包配置文件中指定的 JavaScript 函数执行。为了避免对 JavaScript 函数不必要的调用, 在工具包配置文件中定义了条件元素, 用于指示 UFT 仅在控件定义为 **div** 元素时调用 JavaScript 函数。
- **table** 基本元素必须由工具包配置文件中指定的 JavaScript 函数返回。
- 测试对象标识属性可受使用默认命名约定的 JavaScript 函数支持, 因此无需在工具包配置文件中进行任何更改。
- WebTable 的 **RowCount** 测试对象方法被名为 **BookCount** 的 JavaScript 函数提供的新实现覆盖。因此, 需要在工具包配置文件中指定该函数的名称。
- 筛选与 UsedBooks 控件一起识别的子控件通过调用 JavaScript 函数来完成, 该函数需要在工具包配置文件中指定。
- 要支持录制, 请将工具包配置文件修改为关闭默认 “Web Event Configuration”, 并指定注册 UFT 以侦听正确事件的 JavaScript 函数。另外, 设计一个处理事件注册的 JavaScript 函数, 以及指示 UFT 在事件发生时录制相关步骤的其他 JavaScript 函数。

下面, 您可以查看根据上述信息完成的[清单](#)。

Web 插件扩展性计划清单

<input checked="" type="checkbox"/>	自定义控件支持计划清单	是否在工具包 XML 中指定?	JavaScript 函数是否支持?
<input checked="" type="checkbox"/>	此自定义控件的源位于: %ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\ WebExtSample\Application\UsedBooks.htm	N/A	N/A
<input checked="" type="checkbox"/>	指定新测试对象类扩展的 Web 测试对象基类: (默认值 —— WebElement) WebTable	N/A	N/A
<input checked="" type="checkbox"/>	基本测试对象类是否为 WebElement? 否 如果为“否”, 则是否为基本元素(匹配基本测试对象类的元素)? 是 如果是基本元素, 是否需要 JavaScript 函数来重新运行该元素? 是	是	是
<input checked="" type="checkbox"/>	指定新 Web 测试对象类的详细信息: • 测试对象类名称: WebExtUsedBooks • 图标文件位置(可选): <UFT 安装文件夹>\dat\Extensibility\ Web\Toolkits\WebExtSample\Res\WebBookList.ico • 描述的标识属性: title • 默认的测试对象方法: SelectBook • 帮助文件位置: N/A	N/A	N/A
<input checked="" type="checkbox"/>	指定标识控件所使用的测试对象类的基础: tagName = div tagName of 1st child = table className of 1st child = UsedBooks.	是	是
<input checked="" type="checkbox"/>	指定测试对象命名基础: Use the UsedBooks table title	N/A	是
<input checked="" type="checkbox"/>	列出支持的标识属性, 并标记哪些应可用于检查点(哪些默认选中)以及哪些(如有)应用于智能标识: title (available for checkpoints and selected by default, not used for Smart Identification)	否	是
<input checked="" type="checkbox"/>	列出要支持的测试对象方法(如果需要, 包括参数、返回值、帮助文件位置和帮助 ID): SelectBook (BookIndex) RowCount	是	是
<input checked="" type="checkbox"/>	是否提供所有测试对象方法参数值的动态列表? 否 如果是, 列出参数。	N/A	否
<input checked="" type="checkbox"/>	指定 UFT 应使用控件识别的子类型: Radio buttons	是	是
<input checked="" type="checkbox"/>	是否在 Object Spy 中显示此类的测试对象? 是	否	N/A

<input checked="" type="checkbox"/>	自定义控件支持计划清单	是否在工具包 XML 中指定?	JavaScript 函数是否支持?
<input checked="" type="checkbox"/>	是否提供录制支持? 是 如果提供, 则列出可触发录制的事件: Click on Select	是	是

开发工具包支持集

按照此部分的步骤为 WebExtSample 工具包开发工具包支持集并深入了解 Web 插件扩展性。开发此工具包支持集包括以下阶段:

- 第 1 阶段: 扩展工具包支持集以支持其他控件, 如第 94 页所述
- 第 2 阶段: 使 UFT 标识、侦测和识别 UsedBooks 控件, 如第 95 页所述
- 第 3 阶段: 实现对 WebExtUsedBooks 测试对象方法的支持, 如第 99 页所述
- 第 4 阶段: 实现对 WebExtUsedBooks 标识属性和测试对象名称的支持, 如第 101 页所述
- 第 5 阶段: 实现防止识别子对象的筛选, 如第 102 页所述
- 第 6 阶段: 实现在 UsedBooks 控件上进行录制的支持, 如第 103 页所述

第 1 阶段: 扩展工具包支持集以支持其他控件

要添加对 UsedBooks 控件的支持, 首先将 **WebExtUsedBooks** 测试对象类的定义添加到 **WebExtSampleTestObjects.xml** 文件。

扩展工具包支持集以支持 UsedBooks 控件:

1. 将 UsedBooks 控件的图标文件 **WebBookList.ico** 从 **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Res** 复制到 **<工具包支持集文件夹>\Toolkits\WebExtSample\Res** 文件夹中。
2. 将以下 **WebExtUsedBooks** 测试对象类定义添加到 **WebExtSampleTestObjects.xml** 文件中 (在 **TypeInformation** 元素内, WebExtBook 测试对象类的 ClassInfo 元素后):

```
<ClassInfo BaseClassInfoName="WebTable"
  GenericTypeID="Table"
  Name="WebExtUsedBooks"
  DefaultOperationName="SelectBook">
  <IconInfo IconFile="INSTALLDIR\dat\Extensibility\Web\Toolkits\
    WebExtSample\Res\WebBookList.ico"/>
  <TypeInfo>
    <Operation ExposureLevel="CommonUsed"
      Name="SelectBook"
      PropertyType="Method">
    <Description>
      Selects the radio button for the specified book and clicks Select.
    </Description>
```

```

<Documentation>
  <![CDATA[Select the radio button for the book with index %a1 and click Select.]]>
</Documentation>
<Argument Name="BookIndex"
  IsMandatory="true"
  Direction="In">
  <Type VariantType="Integer"/>
</Argument>
</Operation>
</TypeInfo>
<IdentificationProperties>
  <IdentificationProperty ForDefaultVerification="true"
    ForVerification="true"
    ForDescription="true"
    Name="title"/>
</IdentificationProperties>
</ClassInfo>

```

此文本将根据[Web 插件扩展性计划清单 \(第 93 页\)](#)中描述的详细信息定义 **WebExtUsedBooks** 测试对象类。

有关测试对象配置文件中的元素和属性的详细信息，请参见“UFT Test Object Schema Help”（可通过“Web 插件扩展性帮助”获得）。

第 2 阶段：使 UFT 标识、侦测和识别 UsedBooks 控件

定义新测试对象类之后，必须启用 UFT 以标识此测试对象类所使用的 Web 控件。

如[计划对 Web 插件扩展性 示例 UsedBooks 控件的支持 \(第 89 页\)](#)中所述，如果控件第一个子控件的 **tagName** 和 **className** 属性分别是 **table** 和 **UsedBooks**，则使用 **WebExtUsedBooks** 测试对象表示其 **tagName** 属性为 **div** 的控件。

对于 **WebExtUsedBooks** 测试对象类，标识操作由工具包配置文件和 JavaScript 函数中的 **Condition** 元素组合执行。

定义 WebExtUsedBooks 测试对象类的标识规则：

1. 在 **WebExtSample.xml** 文件中的 **Controls** 元素内，为此测试对象类型添加以下 **Control** 元素：

```

<Control TestObjectClass="WebExtUsedBooks">
  <Settings>
    <variable name="default_imp_file" value="WebExtUsedBooks.js"/>
  </Settings>
  <Identification type="javascript"
    function="IsWebExtUsedBooks">
  <Browser name="*">
    <Conditions type="CallIDFuncIfPropMatch" logic="and">
      <Condition prop_name="tagName" expected_value="div"/>

```

```
</Conditions>
</Browser">
</Identification>
</Control>
```

它定义 UFT 将在文件 **WebExtUsedBooks.js** 中查找 JavaScript 函数，除非指定了其他文件。**Identification** 元素包含一个 **Conditions** 元素，该元素指定所处理控件的 **tagName** 属性是否为 **div**（比较时不区分大小写），调用 JavaScript 函数 **IsWebExtUsedBooks** 以标识是否使用此测试对象类表示该控件。

此教程使用上述定义说明了如何使用 **Conditions** 元素 **Type** 属性的 **CallIDFuncIfPropMatch** 值。但是，如果您使用的应用程序页面上有许多控件，或具有大型 DOM 结构，则最好使用以下文本定义这些标识规则：

```
<Identification type="javascript"
  function="IsWebExtUsedBooks">
  <HTMLTags>
    <Tag name="div"/>
  </HTMLTags>
</Identification>
```

此文本提供相同的功能，指示 UFT 仅为“div”元素调用 **IsWebExtUsedBooks** 标识函数，但在识别自定义控件以及在自定义控件上运行步骤方面提供了更好的性能。

- 在工具包支持集文件夹中，在 **Toolkits\WebExtSample** 文件夹内创建一个名为 **WebExtUsedBooks.js** 的文件（这是设计用于支持 UsedBooks 控件的所有 JavaScript 函数的文件）。
- 在 **WebExtUsedBooks.js** 中，添加以下 JavaScript 函数：


```
function IsWebExtUsedBooks()
{
  // Verify that the tagName property is "div" and the
  // className property of the first child (a TABLE element)
  // is "UsedBooks".
  var firstChild = _elem.children[0];
  if ( _elem.tagName == "DIV" &&
    firstChild.tagName == "TABLE" &&
    firstChild.className == "UsedBooks" )
    return true;
  return false;
}
```

此 JavaScript 函数检查该控件是否满足决定控件应由 **WebExtUsedBooks** 测试对象表示的条件。


部署和测试工具包支持集（第 2 阶段）

定义测试对象配置文件中的 **WebExtUsedBooks** 测试对象类，和工具包配置文件中此测试对象类的标识规则以及 JavaScript 函数后，可使用 UFT 测试使用工具包支持集的效果。

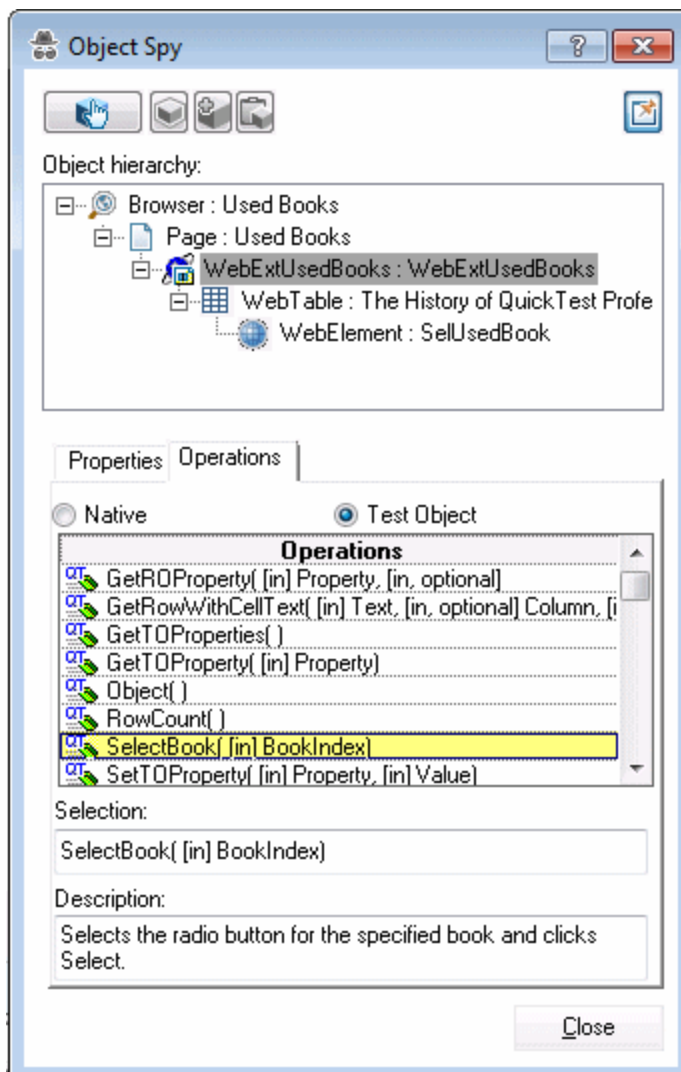
测试工具包支持集:


1. 将测试对象配置文件、工具包配置文件、图标文件和 JavaScript 文件部署到 UFT 安装文件夹内的正确位置上。
2. 打开 UFT 并加载 **WebExtSample** 支持（在“Add-in Manager”对话框中将其选中）。
3. 打开 GUI 测试，然后使用“Object Repository”对话框中的 **Define New Test Object**  按钮打开“Define New Test Object”对话框。通过从 **Environment** 列表中选择“WebExtSample”环境查看在测试对象配置文件中定义的 **WebExtUsedBooks** 测试对象类是否显示在 **Class** 列表中。
4. 通过打开 `%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm` 文件并单击 **Used** 运行此示例控件。

备注: UFT 会在应用程序打开时与其建立连接。因此，如果 UsedBooks 控件打开，则必须将其关闭然后再次运行。

5. 在 UFT 中，通过在 UsedBooks 控件上执行以下活动查看 UFT 识别此控件的方式。（有关使用 UFT 的详细信息，请参见《HP Unified Functional Testing 用户指南》。）
 - 使用 Object Spy  查看 Book 控件支持的标识属性和测试对象操作：**title** 属性未显示值，因为您尚未实现返回其值的 JavaScript 函数。
为 UsedBooks 控件创建的测试对象使用其测试对象类的名称，并使用您定义的自定义图标。稍后在此课程中，您可以通过提供更具体的名称对工具包支持集进行自定义。

WebExtUsedBooks 测试对象包括 WebTable 测试对象的所有测试对象操作，以及在测试对象配置文件中定义的 SelectBook 方法。



- 使用“Object Repository”对话框中的 **Add Objects to Local**  按钮识别 UsedBooks 控件。自定义图标用于表示对象存储库中的测试对象。
- 在 Keyword View 中，通过从对象存储库选择 **WebExtUsedBooks** 对象在 **Item** 列中创建测试步骤。
 - **Operation** 列中的可用操作列表反映了测试对象配置文件中的定义。WebTable 测试对象支持的所有测试对象操作都可用，因为在测试对象配置文件中，已定义 WebExtUsedBooks 测试对象扩展 WebTable 测试对象类（并因此从 WebTable 测试对象类进行继承）。
 - 选择操作后，**Value** 单元格将根据选定操作的参数数量进行分区。例如，使用操作 **SelectBook** 创建步骤时，值单元格将需要一个参数，并在工具提示中显示该参数的名称属性。

- 您在测试对象配置文件中为测试对象方法定义的描述和文档字符串分别显示在工具提示和 **Documentation** 列中。
 - 在编辑器中, 创建带有 WebExtBook 测试对象的测试步骤。语句完成功能显示测试对象可用的所有操作, 包括从 WebTable 继承的操作。
6. 使用在 WebExtUsedBooks 测试对象上执行 SelectBook 测试对象方法的步骤运行测试。UFT 搜索将在控件上运行测试对象方法的 JavaScript 函数。因为尚未实现运行测试对象方法的支持, 所以将出现运行时错误。在 [next](#) 部分, 实现此支持。

第 3 阶段: 实现对 WebExtUsedBooks 测试对象方法的支持

在测试对象配置文件中, 定义 WebExtUsedBooks 测试对象可用的测试对象方法。要使 UFT 运行这些测试对象方法, 必须实际实现这些方法。

必须为各种类型的测试对象方法提供实现:

- 从 WebTable 基本测试对象类继承的测试对象方法
- 为新测试对象类添加的测试对象方法
- 从需要以不同方式实现的基类继承的测试对象方法

实现从 WebTable 继承的测试对象方法

在测试对象配置文件中, 定义 WebExtUsedBooks 测试对象类扩展基类 WebTable。对于不希望替代的继承的 WebTable 测试对象方法, UFT 可通过与 UsedBooks 控件中定义的 **table** 基本元素进行交互来使用其内部实现。因为 **table** 元素不是 UsedBooks 控件的根级别元素, 所以必须通知 UFT **table** 元素是基本元素。为此, 您必须编写返回基本元素的 JavaScript 函数, 并在工具包配置文件中指定其名称。

要指示 UFT 将表 Web 元素用作基本元素:

1. 在 **WebExtSample.xml** 文件中, 在 **Settings** 元素 (在 **Control** 元素中为 WebExtUsedBooks 测试对象类定义) 内添加以下 **Variable** 元素:

```
<Control TestObjectClass="WebExtUsedBooks">
  <Settings>
    <variable name="func_to_get_base_elem"
      value="GetTableElem"/>
  </Settings>
</Control>
```

这段代码指示 UFT 调用名为 **GetTableElem** 的 JavaScript (在 **WebExtUsedBooks.js** 文件中) 以返回支持继承的 WebTable 测试对象方法的基本元素。

2. 在 **WebExtUsedBooks.js** 文件中, 添加以下 JavaScript 函数:

```
function GetTableElem()
{
  // Get the <table> element (the first child of the <div>
  // element which is the root of the UsedBooks control)
  return _elem.children[0];
}
```

```
}

```

此 JavaScript 函数返回 **table** 元素，该元素是定义 UsedBooks 控件的 **div** 元素内的第一个元素。此元素支持从 WebTable 继承的未被 WebExtUsedBooks 实现的测试对象方法。

在此文件中编写的其他 JavaScript 函数还可以使用 **GetTableElem()** 函数访问 UsedBooks 控件中的表元素。

实现新测试对象方法 SelectBook

要支持 **WebExtUsedBooks** 测试对象类的 **SelectBook** 测试对象方法，请在 **WebExtUsedBooks.js** 中编写 **SelectBook** JavaScript 函数。这是 UFT 调用以运行 **SelectBook** 测试对象方法的函数。它模拟选择指定手册的单选按钮，并单击 **Select**。

将以下 JavaScript 函数添加到 **WebExtUsedBooks.js** 文件中：

```
function SelectBook( BookIndex )
// Select the radio button for the specified index
// and clicks the "Select" link.
{
    if( BookIndex > BookCount() )
        throw "Book index is out of range !"
    //Select the radio button corresponding to the specified index
    GetTableElem().rows[1+BookIndex].cells[0].children[0].click();
    //Click the "Select" link (the 3rd child of the <div> element)
    _elem.children[2].click();
    //Add a log message to the event log to assist in debugging
    _util.LogLine("Book Selected",1);
}

```

替代继承的测试对象方法 RowCount 的实现

1. 在 **WebExtSample.xml** 文件中，在定义 **WebExtUsedBooks** 支持的 **Control** 元素内添加以下 **Run** 元素：

```
<Run>
  <Methods>
    <Method name="RowCount" type="javascript"
      function="BookCount" />
  </Methods>
</Run>

```

这段代码定义 **RowCount** 测试对象方法由 JavaScript 函数 **BookCount** 实现。

2. 在 **WebExtUsedBooks.js** 文件中，添加 **BookCount** JavaScript 函数，此函数将减小 UsedBooks 控件的行计数，并返回表中的手册数：

```
function BookCount()


```

```
// This function overrides the RowCount test object method
// inherited from WebTable, so that it counts only book rows.
{
    var table = GetTableElem();
    if( table.rows.length < 2 )
        return 0;
    return table.rows.length - 2;
}
```

部署和测试工具包支持集（第 3 阶段）

开发对运行测试对象方法的支持后，将更新后的工具包支持集部署到 UFT 并进行测试。

测试对运行测试对象方法的支持：

1. 要将更新后的工具包支持集部署到 UFT，请将 **WebExtUsedBooks.js** 文件（必要时和 **WebExtSample.xml** 文件一起）复制到 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行 **UsedBooks** 示例控件。
4. 打开 GUI 测试并使用“Object Repository”对话框中的 **Add Objects to Local**  按钮识别 UsedBooks 控件。
5. 创建运行 **SelectBook** 和 **RowCount** 方法的测试并确保这些方法正确执行。还可以打开 Microsoft Windows 事件查看器查看 **SelectBook** 方法添加的日志消息（有关详细信息，请参见[使用 Microsoft Windows 事件日志 \(第 46 页\)](#)）。

第 4 阶段：实现对 WebExtUsedBooks 标识属性和测试对象名称的支持

在 **WebExtUsedBooks.js** 文件中，实现 **get_property_value** 如下：

```
function get_property_value(prop)
// The function provides values for all of the identification
// properties defined in the test object configuration XML file, as
// well as the hard coded "logical_name" property that UFT
// uses to name the test object.
{
    if ( prop == "logical_name" || prop == "title" )
        // For the "title" identification property, as well as the
        // "logical_name" property, return the inner text of the
        // first cell in the first row
        {
            return GetTableElem().rows[0].cells[0].innerText;
        }
}
```

```
}  
}
```

此函数返回 **title** 属性的对象标题，以及测试对象名称。

备注: 可以在工具包支持集文件夹中修改 **WebExtUsedBooks.js** 文件，然后将其部署到 UFT 进行测试，也可以直接修改 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample\WebExtUsedBooks.js**。

部署和测试工具包支持集（第 4 阶段）

开发对检索标识属性运行时值的支持后，将更新后的工具包支持集部署到 UFT 并进行测试。

测试对检索标识属性运行时值的支持：

1. 确保 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行 **UsedBooks** 示例控件。
4. 新建 GUI 测试，将 **WebExtUsedBooks** 测试对象添加到对象存储库，然后使用此测试对象创建测试步骤。确保测试对象的名称基于表的标题。右键单击该对象，然后选择 **Insert Standard Checkpoint**。将打开“Checkpoint Properties”对话框。确保在测试对象配置文件中定义的 **title** 标识属性包含在属性列表中并已选中。
5. 创建并运行测试，检索每个标识属性并检查属性的值，或在消息框中显示该属性。

第 5 阶段：实现防止识别子对象的筛选

在此部分，您可以创建筛选以防止 UFT 识别所有 **UsedBooks** 控件的子控件以及该控件。

在工具包配置文件和 JavaScript 文件中实现此操作。

筛选与 **UsedBooks** 控件一起识别的子控件：

1. 在 **WebExtSample.xml** 文件中为 **WebExtBook** 测试对象类定义的 **Control** 元素内，添加以下 **Filter** 元素：

```
<Filter>  
  <Learn learn_control="Yes" learn_children="CallFilterFunc"  
    type="javascript" function="GetChildrenToLearn" />  
</Filter>
```

此文本指示 UFT 在识别 **WebExtUsedBooks** 测试对象的父网页时对其进行识别，并调用 JavaScript 函数 **GetChildrenToLearn** 以确定要识别的子项。JavaScript 函数返回应与该控件一起识别的控件子对象的 **SafeArray**。

备注: 您可以修改工具包支持集文件夹中的 **WebExtSample.xml** 文件，并在稍后将其部署到 UFT 进行测试，或者直接修改 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample.xml**。

2. 在 **WebExtUsedBooks.js** 文件中, 添加以下函数:


```
// Learn filtering
// This function instructs UFT which child objects of a
// UsedBooksTable should be learned with the object is learned.
function GetChildrenToLearn()
{
    // Return all of the radio buttons in the UsedBooks table
    return toSafeArray(GetTableElem().getElementsByTagName("input"));
}
```

此函数可确保按照本课程开始的计划, 仅识别单选按钮。

部署和测试工具包支持集 (第 5 阶段)

定义自定义识别子对象的筛选后, 将更新后的工具包支持集部署到 UFT 并进行测试。

测试对识别无子对象的 WebExtUsedBooks 测试对象的支持:

1. 确保 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框, 然后单击 **OK**。将打开 UFT 并加载您设计的支持。
3. 关闭并重新运行 **UsedBooks** 示例控件。
4. 打开 GUI 测试并打开“Object Repository”。使用“Object Repository”对话框中的 **Add Objects to Local**  按钮识别包含 UsedBooks 控件的网页。将打开“Define Object Filter”对话框。
5. 选择 **All object types** 并单击 **OK**。名为 **The History of QuickTest Professional** 的 WebExtUsedBooks 对象将添加到对象存储库中, 因为它是 SelUsedBook 单选按钮组。但是控件中包含的所有其他元素将无法识别。

第 6 阶段: 实现在 UsedBooks 控件上进行录制的支持

在此部分, 您可以实现在 UsedBooks 控件上进行录制的支持。

1. 在 **WebExtSample.xml** 文件中, 在定义 WebExtUsedBooks 类的 **Control** 元素内添加以下 **Record** 元素:

```
<Record>
  <EventListening use_default_event_handling_for_children="false"
    use_default_event_handling="false"
    type="javascript" function="ListenToEvents"/>
</Record>
```

这段代码指示 UFT 在 UsedBooks 控件上不使用默认“Web Event Configuration”记录事件, 而是调用 **ListenToEvents** JavaScript 函数。

在 **WebExtUsedBooks.js** 文件中添加 **ListenToEvents** JavaScript 函数:

```
function ListenToEvents( elem )
{
    // Connect to the "Select" event:
    //When "Select" is clicked, call OnSelectUsedBooksClicked.
    _util.RegisterForEvent(_elem.children[2], "onclick",
        "OnSelectUsedBooksClicked" );
    return true;
}
```

此函数注册 UFT 以侦听“Select”链接上的单击，并在发生事件时调用相应的事件处理程序。

2. 在 **WebExtUsedBooks.js** 文件中添加 **OnSelectUsedBooksClicked** 事件处理 JavaScript 函数:

```
function OnSelectUsedBooksClicked( handlerParam , eventObj )
{
    var arr = new Array();
    var booksCount = BookCount();
    // Find the index of the selected radio button and record
    // a step that runs the SelectBook test object method
    // with that index.
    var BookIndex = -1;
    for( var i = 0 ; i < booksCount ; i++ )
    {
        if( _elem.rows[2+i].cells[0].children[0].status == true )
        {
            // This is the selected item
            arr[0] = i+1;
            _util.Record( "SelectBook", toSafeArray(arr) , 0 );
            _util.LogLine("SelectBook Recorded",1);
            break;
        }
    }
    return true;
}
```

此函数检查选定的手册单选按钮，并指示 UFT 记录选择该手册的步骤（将相关日志消息添加到事件日志）。

部署和测试工具包支持集（第 6 阶段）

开发在 UsedBooks 控件上进行录制的支持后，将更新后的工具包支持集部署到 UFT 并进行测试。

测试在 UsedBooks 控件上执行的录制操作的支持:

1. 确保 **<UFT 安装文件夹>\dat\Extensibility\Web\Toolkits\WebExtSample** 中有最新文件。
2. 关闭并重新打开 UFT。选中“Add-in Manager”对话框中的 **WebExtSample** 复选框，然后单击 **OK**。将打开 UFT 并加载您设计的支持。

3. 关闭并重新运行 **UsedBooks** 示例控件。
4. 打开 GUI 测试并单击 **Record** 按钮或选择 **Record > Record**。在 “UsedBooks” 表中，选择某个单选按钮，然后单击 **Select**。

仅在单击 **Select** 后才向测试添加新步骤：

Item	Operation	Value	Documentation
▼ Action1			
▼ Used Books			
▼ Used Books			
▼ The History of QuickTest	SelectBook	2	Select the radio button for the book with index 2 and click Select.

单击 **Stop** 按钮或选择 **Record > Stop** 以结束会话录制。

根据您的决定的有关计划自定义支持的时间规范，现在已完全支持 Book 自定义控件。

课程总结

在此课程中，您将：

- 创建扩展 WebTable 测试对象类的 WebExtUsedBooks 测试对象类。
- 创建了对 UsedBooks 控件的支持，支持 UFT 将其识别为 WebExtUsedBooks 测试对象。
- 学习了解工具包配置文件中的更多选项。
- 学习实现使用更复杂 JavaScript 函数并在工具包配置文件中指定函数位置的支持。

接下来要执行的操作

现在，您已学完本教程中的课程。下面，您可运用所学的 Web 插件扩展性概念和技能创建自己的自定义工具包支持。

有关工具包支持集的结构和内容的详细信息，请参见[开发工具包支持 \(第 25 页\)](#)。

有关测试对象配置文件的结构和内容的详细信息，请参见“UFT Test Object Schema Help（可通过“Web 插件扩展性帮助”获得）”。

有关工具包配置文件的结构和内容的详细信息，请参见“UFT Java Add-in Extensibility Toolkit Configuration Schema Help”（可通过“Web 插件扩展性帮助”获得）。

有关 `_util` 实用程序对象和全局 JavaScript 方法的详细信息，请参见“UFT Java Add-in Extensibility API Reference”（可通过“Web 插件扩展性帮助”获得）。

发送反馈



我们是否能够更完善开发人员指南？

请告诉我们怎样完善：sw-doc@hp.com

