



**ENTERPRISE ARCHITECT**

用户指南系列

# 测试点

Author: Sparx Systems

Date: 20/06/2023

Version: 16.1

创建于  **ENTERPRISE  
ARCHITECT**

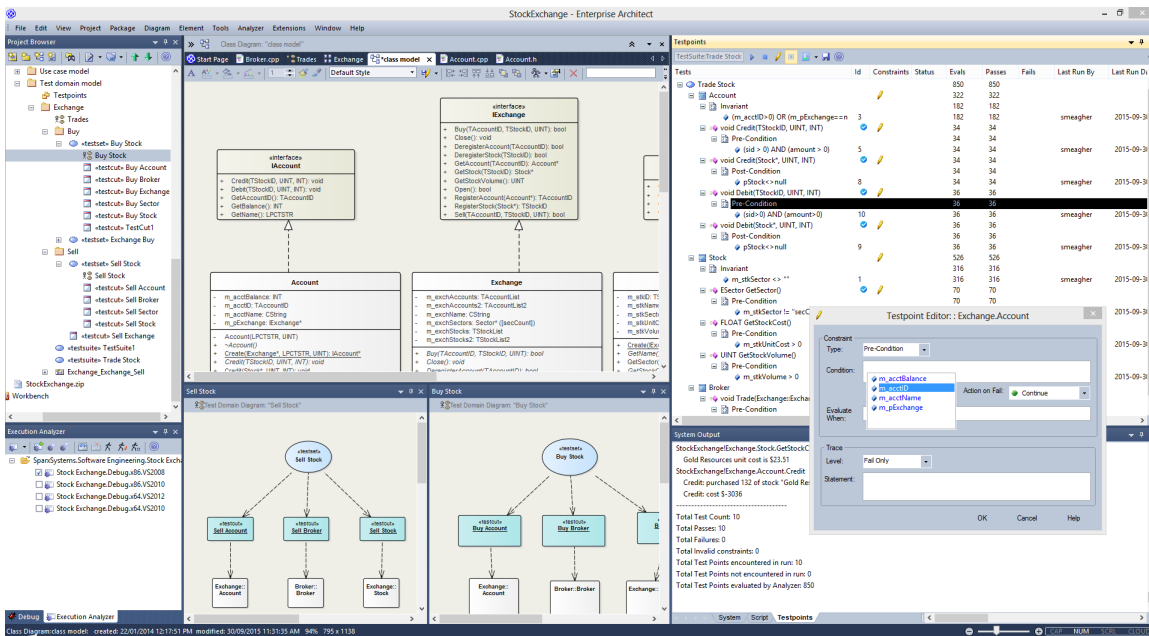
# 目录

测试点	3
测试域图表	6
测试接口	8
测试集	9
测试套件	10
测试点窗口	11
测试点工具栏	13
测试点编辑	15
测试点约束	17

# 测试点

测试点提供了一种方案，通过该方案可以从模型中获取控制对象行为的约束和规则，并将其应用于一个或多个应用程序。像这样的方案提供的优点是对代码更改的容忍度——从函数中添加和减去行对管理它的约束没有影响。另一个优点是行为规则的改变不需要对源代码进行相应的改变；意味着什么都不需要重新编译！

此外，使用单个测试域验证多个应用程序的能力是一件简单的事情，而不是繁重的事情。测试域既是逻辑模型又是关系模型；类中的约束可以用测试接口模型口进行划分。这些可以使用连接器简单地聚合到测试集和测试套件中。由于测试域与代码库的解耦，可以简单地选择按钮来正常运行程序，或者运行特定的测试域运行它。该系统还提供了实际的好处，因为根本不需要任何仪器。在运行期间，测试结果会在程序运行时实时显示在报告窗口中。这些结果可以保留，并随时在“测试细节”对话框中使用Enterprise Architect的文档特征进行审查。



## 特征

特征	细节
测试点组合	<p>测试点组合是使用 Testpoints 窗口执行的。Testpoints 窗口是上下文相关的，并在浏览器窗口或图表中显示所选元素的测试域。选择单个类将显示类结构。对具有现有约束的类和方法显示A 铅笔”图标。</p> <p>当您选择测试接口、Set 或 Suite测试时，Testpoints 窗口会显示整个域结构，包括构成域的所有类。笔记：您可以使用右侧的 导航”窗格导航域层次结构。测试点使用类成员的变量名组成表达式。智能感知快捷方式 Ctrl+Space 可在编辑器中找到这些内容。计算结果为True的表达式被视为通过。返回False意味着失败。</p>

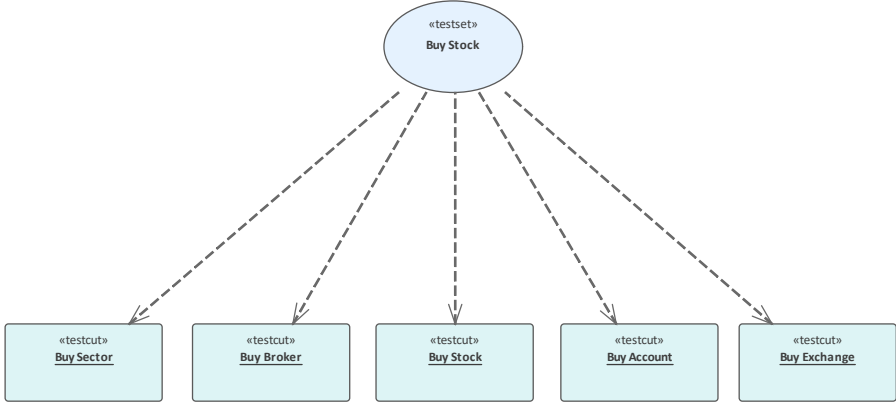
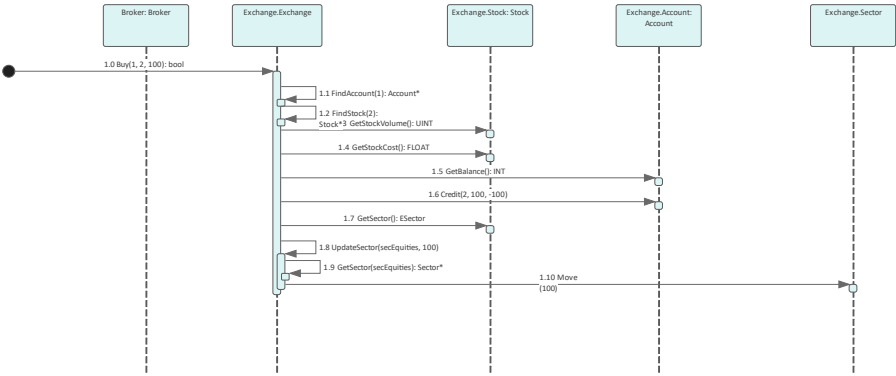
	 <p>您可以通过双击类来添加或编辑现有的不变量。 您可以通过双击方法类似地添加或编辑现有的前置条件或后置条件。 如果有源代码，双击测试点会自动显示源代码。 最好使用代码编辑器的快捷菜单从代码编辑器中添加行条件。 此图像是测试域中的先决条件。</p>
测试点跟踪陈述	每个测试点都可以有自己的跟踪语句。跟踪语句是一个动态消息，可以在其 object 或本地范围内引用变量。它们在测试评估期间输出。可以将它们配置为在每次评估约束时输出，或者更常见的是在测试失败时输出。可以将跟踪语句定向到系统输出窗口的“测试点”选项卡或外部文件。您可以在任何分析器脚本配置它。
测试域组合	测试域图是一个动态媒介，其中测试点被组装以测试使用案例。测试域图中的使用案例以三种不同的原型提供：测试接口、测试集和测试套件。域的管理就像在任何图表上建模一样简单。工具箱快捷菜单提供对任何测试域和工件的访问。简而言之，来自多个类的测试点被聚合成测试集。然后将测试集链接到形成测试套件。测试接口和测试集都是可重复使用的资产。将同一个测试集链接到一个或多个测试套件是绘制连接器的问题。

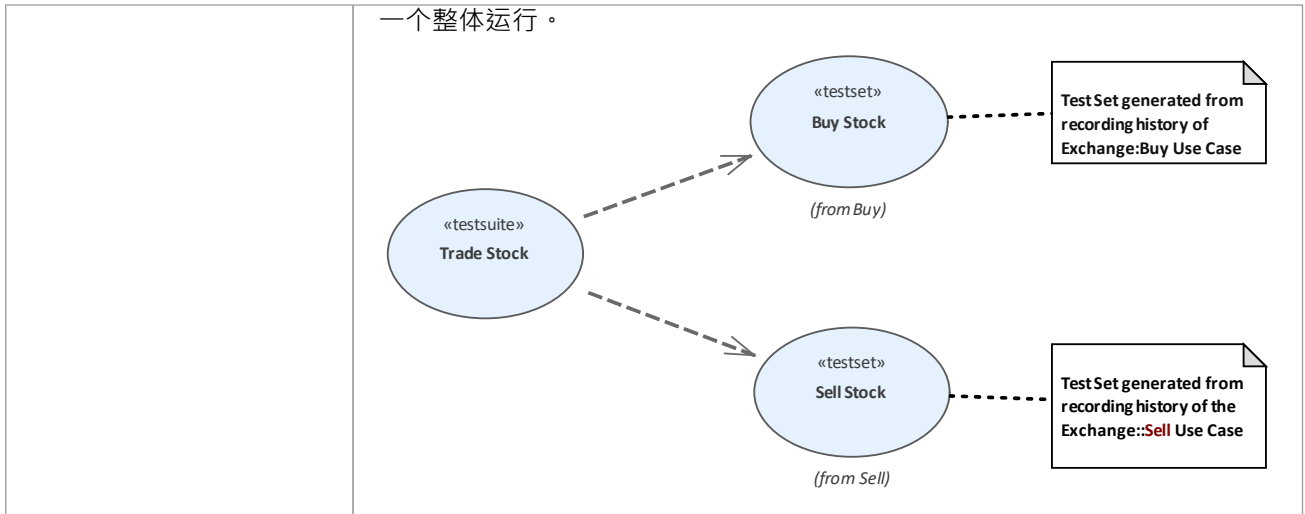
<p>测试域和类模型</p>	<p>一个用例很少会涉及单个类的所有方法。它很可能是使用协作类中的各种方法来实现的。我们称此方法为“剪切口”的子工件，是我们用来进行测试截口的工具。测试点窗口将根据时间进行上下文，以适应测试域领域或类元素。此图显示了选择测试截口时的测试截口窗口。请注记复选框，这些复选框仅对测试截口可见。它们表示对测试集有贡献的方法（测试截口）。在这个例子中，测试域是由执行分析器生成的，它为我们完成了方法识别工作。</p>
<p>测试点评估</p>	<p>Testpoints 窗口用于评估测试域。该窗口有一个用于启动或附加到目标应用程序的工具栏。要测试的域总是由具有时间的元素上下文，因此如果您选择一个类，则窗口将仅显示该类的类结构和测试类。如果您选择一个测试套件，该窗口将显示整个域层次结构以及其中包含的所有测试点。单击运行按钮将加载执行分析器中的测试点域，然后在使用案例通过或失败时评估、收集和更新报告窗口。每个约束类型的确切细节以及该约束捕获的时间和方式是：</p> <ul style="list-style-type: none"> <li>• 类对类类型的object调用的任何方法完成时，分析器都会评估A不变量；不变量用于测试符合object的状态是否已知且允许</li> <li>• 在调用操作之前立即评估前置条件</li> <li>• 方法完成时评估后置条件（同时作为类不变量）</li> <li>• 如果以及何时在程序执行期间它们的特定代码行进入范围，则评估行条件</li> </ul>

# 测试域图表

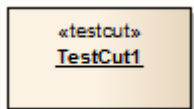
测试域图是您为特定领域组装和分组测试用例的媒介。测试域的一个示例可能是“顾客”。您组装的域的广度和深度取决于您。您可能为“添加顾客”和“删除顾客”设置了不同的域，这完全取决于您认为平衡域层次结构的最佳方式。工具图表快捷菜单提供了一个数字工具箱工件模型该领域。因为媒体是动态的，允许您重新访问和构建测试域之间的关系，所以该系统是一个很好的模型，用于向组织交付可重用资产，其开销低，并与UML世界视图和软件工程集成日常生活的基本要素。

## 功能

功能	细节
<p>测试域生成</p>	<p>如果您认为编写测试域的过程很复杂，可以，但帮助就在眼前！执行分析器可以为您生成测试域图。它不能为您编写测试，但它可以做一些腿部工作。它可以识别类并仅挑选出参与用例的那些方法。这不是猜测。分析器测试域是从一个正在运行的程序中获得的。此图显示了执行分析器通过记录示例模型程序生成的测试域。</p>  <p>这就是生成测试域的记录本身（作为序列图）。</p>  <p>Sequence diagram generated in Enterprise Architect using recording marker in a Use Case</p>
<p>测试域组合</p>	<p>测试域图的第一个任务是创建使用案例（测试集）。这些定义了这个特定域的职责。图表和快捷菜单提供工具箱工件您实现这一目标。这些元素中的第一个是测试接口，用于下一步；从类模型中识别出您认为参与使用示例的用例方法。测试接口工件测试，因为它允许划分一个类，选择那些相关的方法。测试接口可以运行或链接一个或多个。测试集反过来可以链接到一个或设置更多测试套件。在任何情况下，测试域树的任何元素都可以运行或作为</p>



## 测试截图



### 描述

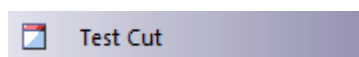
测试截图元素是A原型的物件元素，用于Enterprise Architect内部，用于使用测试点代码测试功能定义测试集。

诸如“打印”之类A任务可能涉及对不同类的操作。为了创建“打印”测试，您可能只想包含这些类的“打印”操作并排除任何其他操作。

测试截图使您能够仅捕获表示为单个类定义的行为（在本例中为“打印”）A操作。然后，您可以将几个类中的每个类中的测试截图作为测试集放入一个任务中。

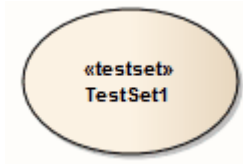
当您将测试截图元素拖到测试域图上时，您将创建一个与所需类元素的依赖关系。因此，当您在Testpoints窗口中选择测试截图元素时，该类的操作会在窗口中列出，每个类都带有一个复选框。然后选中每个类操作的复选框以包含在测试截口中。

### 工具箱icon





## 测试集

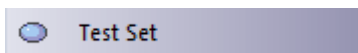


### 描述

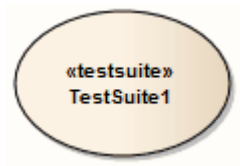
测试集元素是A典型的用例元素，用于将可能跨越多个类的一组或多组方法（测试接口）聚合到单个任务中。测试集也可以聚合到测试套件中。

您使用依赖连接器将测试接口元素链接到测试集。

### 工具箱icon



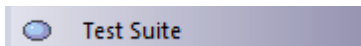
# 测试套件



## 描述

测试套件元素是A原型用例元素，用于聚合一组或多组任务（测试集）。您使用依赖连接器将测试集元素链接到测试套件套件。

## 工具箱icon



## 测试点窗口

测试点窗口是组成测试域约束的中心。它也是让您验证程序上的特定测试域的控件。该程序可能已经在运行，或者可以使用控件的工具栏启动。在这里，您还可以实时查看测试结果。此控件是上下文相关的，响应浏览器窗口或图表中元素的选择。根据选择，可以对单个类、一个用例（测试集）或一组使用案例（A测试套件）进行测试。

### 访问

功能区	执行 > 工具 > 测试器 > 显示测试点窗口
-----	-------------------------

### 测试点窗口列

柱子	用途
测试	<p>显示选定测试点object的名称及其下对象的层次结构。</p> <p>选定的object可以是：</p> <ul style="list-style-type: none"> <li>• 类</li> <li>• 手术</li> <li>• 测试接口</li> <li>• 测试设置或</li> <li>• 测试套件</li> </ul>
ID	<p>对于一个操作，当分析器在目标应用程序中成功绑定该操作时，该列显示一个测试点标记图标 (  )。如果运行过程中该栏没有出现运行，则表示模型和代码库可能没有同步；也许函数的签名已经改变，或者操作是你正在处理的新方法，它存在于源代码中但尚未出现在你的模型中。</p> <p>对于测试点，此列显示生成的 ID 号。此 ID 号用于跟踪输出中以指示正在引用哪个约束。</p>
约束	<p>此列中A铅笔图标 (  ) 表示为类或操作定义了一个或多个约束。</p>
状态	<p>在测试运行期间，指示这些可能的状态：</p> <ul style="list-style-type: none"> <li>• (  ) 失败 - 约束被评估为false一次或多次。</li> <li>• (  ) Invalid Statement - 由于语法无效，约束解析失败。</li> <li>• (  ) 未找到变量 - A评估约束的位置未找到引用的变量名称。</li> </ul> <p>如果约束已通过，则不显示任何图标。</p>
评估	<p>在测试运行期间，指示执行分析器评估此约束的次数。</p>
通行证	<p>在测试运行期间，表示测试通过的次数。</p>
失败	<p>在测试运行期间，指示测试失败的次数。</p>

最后运行	显示最后一个运行此测试的人的用户名。（值来自“人物”对话框中的项目作者定义 - “设置 > 参考 > 模型类型 > 人物 > 项目作者”。）
上次运行日期	显示上次评估此测试的日期和时间。
上次运行结果	显示上次测试运行的结果。
父集合窗格	列出包含选定object作为其设计的一部分的所有父集合。 双击此集合使其成为左窗格中的选定object。 通过单击测试点窗口工具栏上的显示/隐藏父集合窗格按钮，可以隐藏父集合窗格。

## 测试点工具栏

测试点窗口工具栏提供了对当前选择的测试点object执行配置的测试、停止当前正在进行的测试运行、过滤显示的项目以及保存已完成测试运行。

### 访问

功能区	执行 > 工具 > 测试器 > 显示测试点窗口
-----	-------------------------

### 测试点工具栏选项

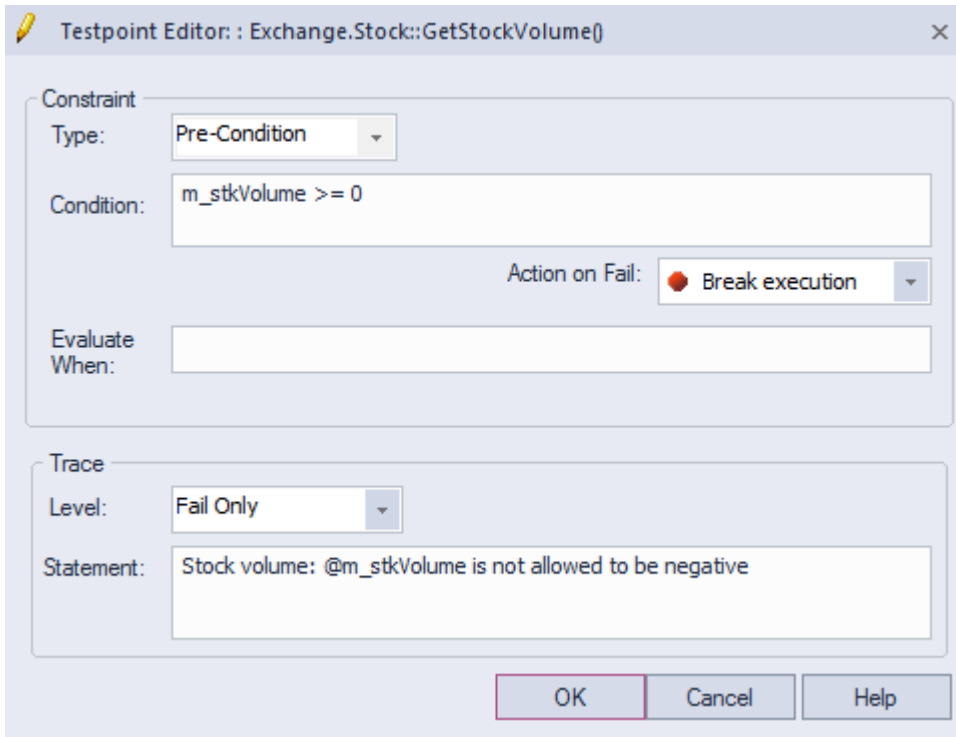
工具栏按钮	行动
	字段当前选中的测试点object的名称。
	执行测试运行。
	运行当前正在进行的测试。
	在显示所有项目和仅显示已定义约束的项目之间切换。
	在显示所有项目和仅显示已标记为包含在此测试截图中的操作之间切换；此按钮仅在选择测试截图object时启用。 当一个测试截图被选中时，其关联类的每一个操作都会显示一个复选框；您使用此复选框来标记适用于此测试截图的操作。
	单击此图标旁边的下拉箭头以显示“测试运行选项”菜单，提供以下选项： <ul style="list-style-type: none"> <li>'前缀跟踪输出With函数调用' - 前缀所有带有执行函数名的输出行</li> <li>'Enable Standard断点during testing' - 未勾选时，测试运行忽略当前断点集中的任何断点，并且在运行期间设置运行的任何尝试都将被忽略</li> <li>'视图跟踪输出' - 显示系统输出窗口的'测试点'选项卡</li> </ul>
	测试运行完成后点击此图标可将结果保存到当前object的测试项中。可以使用测试工作空间查看保存的测试。 显示A提示以选择测试类- Unit、集成、系统、Inspection、Acceptance 或 Scenario。选择合适的测试类并点击确定按钮。
	显示测试点管理帮助主题。
	显示或隐藏父集合窗格。



# 测试点编辑

测试点编辑器用于构成类和操作的约束。允许的约束类型取决于所选object。对于类，类型总是不变的。对于操作，类型可以是前置条件、后置条件或行条件。

当对选定类类型的object调用的任何方法完成时，分析器会评估不变量。每次调用指定操作开始时都会评估前置条件。在完成对指定操作的每次调用后评估后置条件。每次执行指定的代码行时都会评估行条件。



## 访问

功能区	<ol style="list-style-type: none"> <li>1. 执行 &gt; 工具 &gt; 测试器 &gt; 显示测试点窗口。</li> <li>2. 在 Testpoints 窗口中，双击类或 Operation 以显示 测试点编辑器“对话框”。</li> </ol>
-----	---

## 约束组字段

字段	用途
类型	所选类或操作的约束类型： <ul style="list-style-type: none"> <li>• 不变 - 在指定类上调用的任何方法完成后评估</li> <li>• 前置条件 - 在每次调用特定操作开始时进行评估</li> <li>• 后置条件 - 在完成对特定操作的每次调用后评估</li> <li>• 行条件 - 在操作中执行特定代码行时评估</li> </ul>
抵消	仅适用于 Line-Conditions，指定操作中的行号，用于评估约束。

	如果测试点是使用代码编辑器上下文菜单创建的，则会自动设置偏移值。
条件	触发该测试点时要评估的约束。A该约束条件的计算结果为true or false，将记录通过或失败的状态。
行动失败	单击下拉箭头并从三个选项中进行选择： <ul style="list-style-type: none"> <li>• 'Continue' - 忽略此约束的失败并继续执行</li> <li>• 'Break execution' - 停止执行并显示堆栈跟踪</li> <li>• 'Disable on fail' - 失败一次后不再执行约束</li> </ul>
评估时间	(可选) 在评估主要测试点条件之前必须满足的附加约束，从而更好地控制测试覆盖率。

## 跟踪组字段

选项	行动
等级	指定何时输出跟踪语句(如果已定义)。可用选项有： <ul style="list-style-type: none"> <li>• 'Fail Only' - 仅在此测试点条件失败时输出跟踪语句</li> <li>• 'Always' - 每次评估此测试点时输出跟踪语句</li> </ul>
陈述	(可选) 评估此测试点时要输出A消息。 当前范围内的变量可以包含在跟踪语句输出中，方法是在变量名称前加上\$标记(用于string变量)或@标记(用于原始类型，例如“int”或“long”)。 监视状态的输出可以被定向到系统输出窗口的“测试点”选项卡，也可以定向到由跟踪分析器脚本包的外部文件。



## 测试点约束

约束通常由表达式中A局部变量和成员变量组成，由运算符分隔以定义必须满足的一个或多个特定标准。约束A评估为真才能被视为通过。如果约束评估为false，则将其视为失败。

约束内引用的任何变量都必须在评估测试点或断点的位置的范围内。

### 一般/算法算子

操作员	描述
+	添加 示例： $a + b > 0$
-	减去 示例： $a - b > 0$
/	划分 示例： $a / b == 2$
*	乘 示例： $a * b == c$
%	模数 示例： $a \% 2 == 1$
()	括号 - 用于定义复杂表达式中的优先级。 示例： $((a / b) * c) <= 100$
[]	方括号 - 用于访问数组。 示例： $Names[0].Surname == "Smith"$
.	点运算符 - 用于访问类的成员变量。 示例：示例 == "名称"
->	选择点运算符的表示法。 示例：站->名称 == 弗林德斯"

### 比较运算符

操作员	描述
=	等于 示例： $a = b$

==	等于 示例：a == b
!=	不等于 示例：a != b
<>	不等于 示例：a <> b
>	比...更棒 示例：a > b
>=	大于或等于 示例：a >= b
<	少于 示例：a < b
<=	小于或等于 示例：a <= b

## 逻辑运算符

操作员	描述
和	逻辑与 示例：(a >= 1) AND (a <= 10)
或者	逻辑或 示例：(a == 1) 或 (b == 1)

## 位运算符

操作员	描述
&	按位与 示例：(1 & 1) = 1 (1 & 0) = 0
	按位或 示例：(1   1) = 1 (1   0) = 1

^	按位异或 ( 异或 ) 示例：( 1 ^ 1 ) = 0 ( 1 ^ 0 ) = 1
---	--

## 其他示例

示例	描述
<code>((m_nValue &amp; 0xFFFF0000) == 0)</code>	使用十六进制值作为右操作数的位与运算符 (&) 来测试在变量的高位字节中没有设置任何位。
<code>((m_nValue &amp; 0x0000FFFF) == 0)</code>	使用以十六进制值作为右操作数的位与运算符 (&) 来测试在变量的低位字节中没有设置任何位。
<code>m_value[0][ 1 ] = 2</code>	访问多维数组
a 与 (b 或 c)	结合 AND 和 OR 运算符，使用括号确保优先级。在此示例中，变量 "a" 必须为真，"b" 或 "c" 必须为真。

## 注记

- 字符串比较区分大小写

