ENTERPRISE ARCHITECT

用户指南系列

记录

Author: Sparx Systems

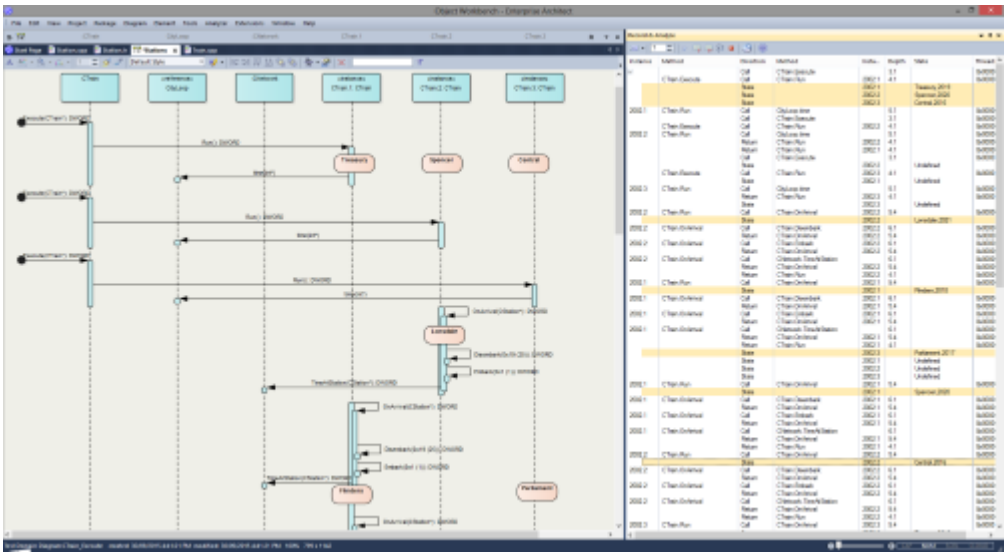
Date: 20/06/2023

Version: 16.1

创建于 ENTERPRISE ARCHITECT

目录

记录



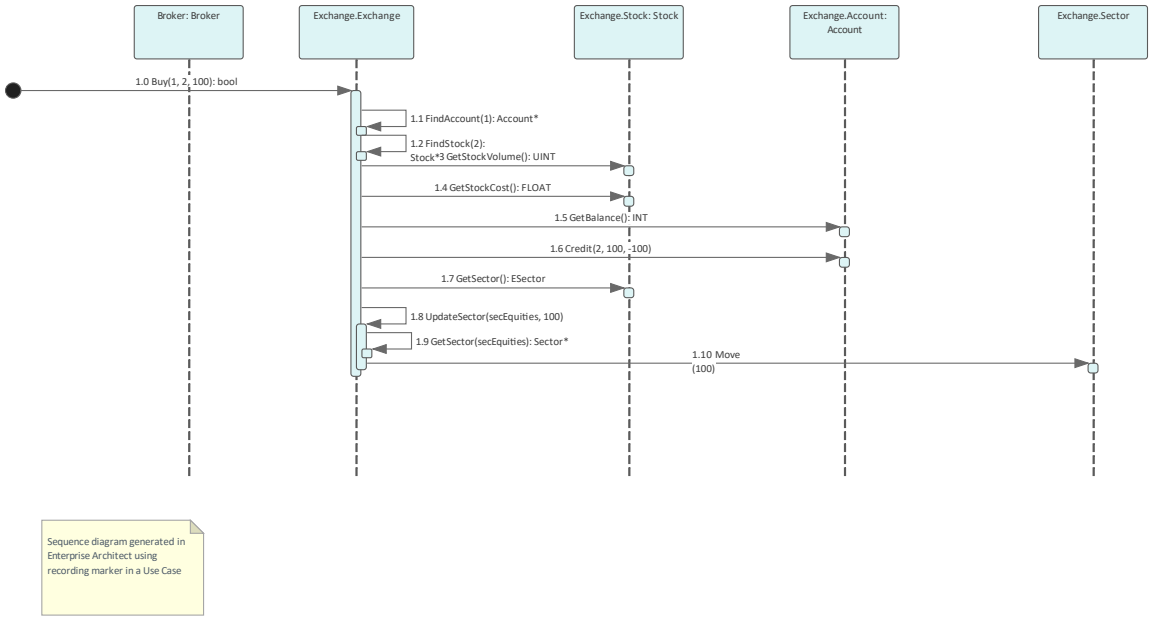
序列图是理解行为的极好帮助。类协作图也很有帮助。除了这些，有时调用图正是我们所需要的。再说一次，如果你有这些信息，你可以用它来记录一个用例，为什么不建立一个测试域呢？ Enterprise Architect分析器可以为您生成所有这些，并从单个记录中生成。它通过记录一个正在运行的程序来做到这一点，它适用于所有最流行的平台。

访问

功能区	执行 > 工具 > 记录器

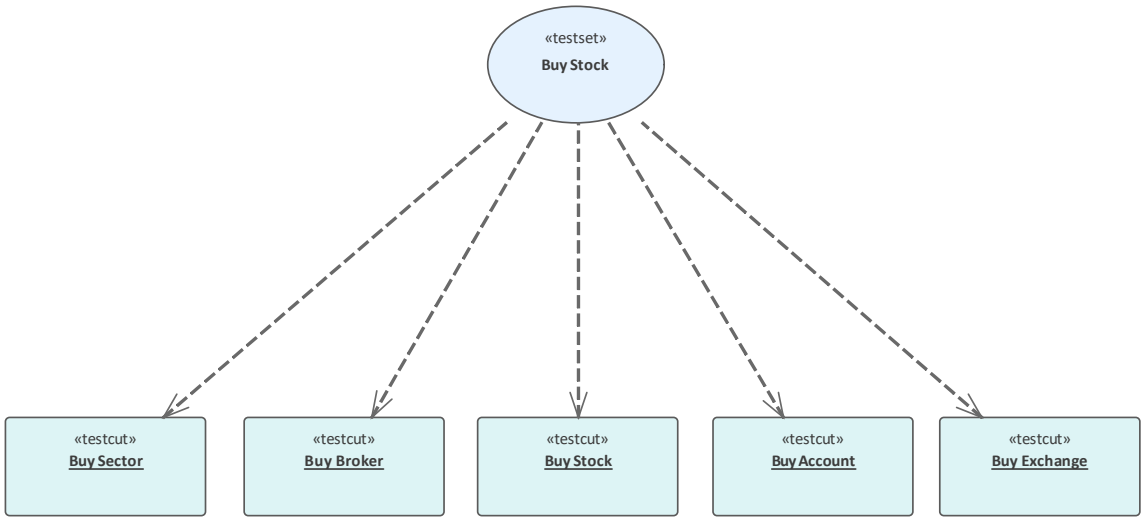
概述

在最简单的情况下，即使使用全新的模型，只需很少的步骤即可生成序列图。你甚至不必配置分析器脚本打开Enterprise Architect代码编辑器 (Ctrl+Shift+O)，在您选择的函数中放置一个记录标记，然后将Enterprise Architect调试器附加到运行该代码的程序中。每当调用该函数时，都会捕获其行为以形成记录历史。从这段历史可以很容易地创建这些图表。



示例模型记录中的序列图。

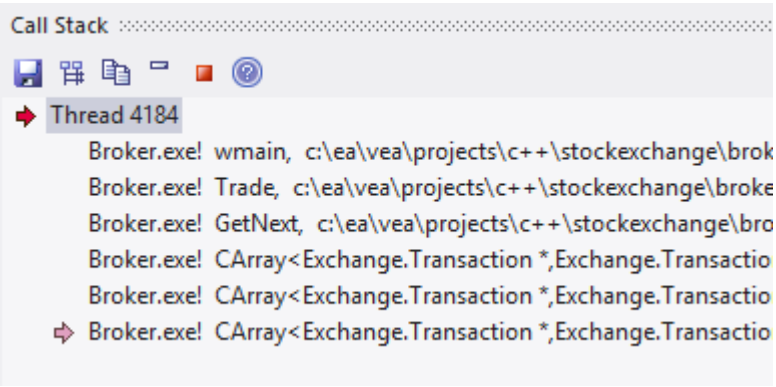
来自同一录音的类协作图。



来自同一记录的测试域图。

当然，分析器脚本是最好的想法，并且开辟了极其丰富的开发环境，但值得注意的是，没有分析器也可以获得显着的效果。 Enterprise Architect调试器和 Profiler 工具也是如此。

兴趣点：您可以A记录线程时查看线程的行为。在录制过程中显示调用堆栈堆栈的更新将实时显示，就像动画一样。这是一个很好的反馈工具，在某些情况下它可能就是所需要的。

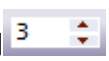


特征一目了然

图表生成

- 序列图
- 类协作图
- 测试域图
- 状态转移
  捕获
- 调用图

控件

- 支持多线程和单线程模型
- 支持堆栈深度控制 

- 支持过滤器限制捕获
- 过滤器通配符支持
- 实时堆栈更新

集成

- 类模型
- 测试域
- 状态机
- 可执行状态机
- 单元测试

平台

- 微软.NET
- 微软原生
- Java
- PHP
- 广发银行
- 安卓

需求

- 记录可供Enterprise Architect所有版本的用户使用

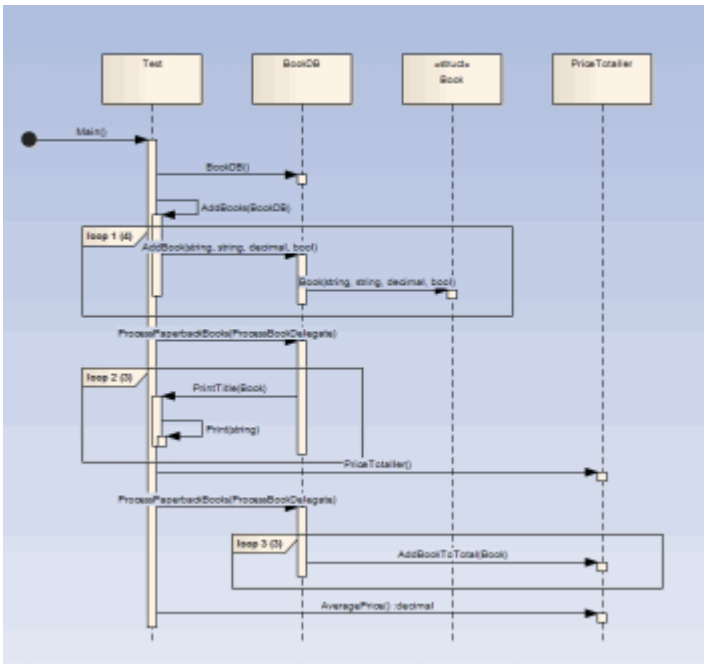
注记

- Oracle 的Java服务器平台 特征"不支持可视化执行分析器的调试和记录特性

这个怎么运作

本主题介绍了可视化执行分析器如何生成序列图。

解释

积分	细节
用途	可视化执行分析器使您能够从应用程序实时执行的记录中生成序列图。随着应用程序的运行，每个线程的历史都会被记录下来。此历史可用于生成序列图。 这是一个计算书籍价格的程序生成的序列图：  录音机如何知道要记录什么？ • 录音机通过录音笔工作；这些由您放置在感兴趣的功能中 调用Java中的堆栈可以延伸到肉眼所能看到的范围之外。我们如何将录制限制为仅十帧？ • 记录器由记录器工具栏上设置的深度或与存储在模型中的标记集相关联的深度控制
它是真实的	录制时，不修改目标应用程序；根本没有任何图像或模块的检测。使用程序的 发布”版本制作A录音是程序所做工作的可靠文档。
你从哪里开始	我们有一个非常大的服务器应用程序；那么我们从哪里开始呢？如果您对要录制的程序知之甚少或根本不了解，并且很少或根本没有模型可以说出来，那么您最好从 Profiler 开始。在以特定方式使用程序的同时运行 Profiler 可以从所呈现的入口点和调用图快速识别使用案例。拥有这些知识可以使您聚焦于未发现的区域并记录这些功能。 如果你源代码，你需要做的就是在你感兴趣的函数中放置一个记录标记。我们建议不要同时在多个函数中放置多个记录标记。在实践中，这已被证明不太有用。你在哪里放置录音标记？对于 Windows UI 程序，以及与某些业务用

	例相关的情况，您可能首先在事件处理程序中放置一个似乎最相关的消息。如果您正在调查实用程序函数，只需在开始处或附近设置一个方法记录标记。 对于服务、守护进程和批处理，您可能希望针对每个感兴趣的行为对程序进行一次概要分析，并使用报告来探索那些未发现的区域。
小费	最好在调试前快速浏览一下断点和标记窗口，并检查此处列出的标记是否符合您的预期。
场景	• Microsoft 本机 C 和 C++、VB（窗口程序、窗口服务、控制台程序、COM 服务器、IIS ISAPI 模块、旧版） • 微软.NET（ASP.NET、窗口Foundation (WPF)、窗口、工作流服务、设备、模拟器） • Java（应用程序、小程序、Servlet、Beans） • 安卓（对设备和模拟器使用 Android 调试桥） • PHP（网络网站脚本） • 广发银行（窗口/Linux 互操作性）

记录历史

当应用程序的执行分析遇到用户定义的记录标记时，记录的所有信息都保存在 Record & Analyze 窗口中。

访问

功能区	执行 > 工具 > 记录器 > 打开记录器

功能

功能	信息/选项
信息显示	记录和分析窗口中的列是： • - 唯一序列序列 • Threads - 操作系统线程 ID • Delta - 自序列开始以来经过的线程 CPU 时间 • 方法 - 有两个方法列：第一个显示调用者，如果返回，则显示当前帧；第二个显示调用的函数或返回的函数 • Direction - Stack Frame Movement，可以是调用、返回、状态、Breakpoint或Escape（Escape在内部制作序列图时使用，用来标记一次迭代的结束） • Depth - 调用时的堆栈深度；用于生成序列图 • -状态-状态之间的状态 • 源- 有两个源列：第一个显示调用者的源文件名和行号，如果是返回，则显示当前帧；第二个显示函数调用或返回函数的源文件名和行号 • Instance - 有两个Instance列，只有当产生的序列图包含状态Transitions时才有值；这些值由以逗号分隔的两项组成 - 第一项是捕获的类实例的唯一编号，第二项是类的实际实例 例如：假设一个类'类'的内部值为4567，程序创建了该类的两个实例；这些值可能是： - 4567, 1 - 4567,2 第一个条目显示类的第一个实例，第二个条目显示第二个实例
操作信息	Record & Analyze 窗口工具栏提供了功能用于控制记录分析器脚本执行的功能。 录制完成后，您可以使用 录制和分析”窗口上下文菜单对录制结果执行完成操作。

注记

- 每个操作的复选框用于控制此调用是否可用于从该历史记录中创建序列、测试域类或协作类图

- 除了使用复选框启用或禁用调用之外，您还可以使用上下文菜单选项来启用或禁用整个调用、对给定方法的所有调用或对给定类的所有调用

图表特征

当您生成序列图时，它包括以下特征：

特征

特征	细节
参考	当可视化执行分析器无法匹配模型中的操作的函数调用时，它仍然会创建序列，但也会创建任何它无法定位的类的引用。 它适用于所有语言。
片段	片段图中显示的序列代表一段代码的循环或迭代。 可视化执行分析器尝试将函数范围与方法调用相匹配，以尽可能准确地直观地表示执行。
状态	如果在录制过程中使用了状态机，则任何状态转换都会在导致转换发生的方法调用之后呈现。 状态是根据每个方法返回给它的调用者来计算的。

记录设置

本节说明如何准备记录应用程序的执行。

脚步

节
先决条件 - 要设置记录序列图的环境，您必须： • 已完成编译和调试的基本设置并为包创建了执行分析脚本 • 能够成功调试应用程序
通过应用过滤器缩小记录的聚焦。
通过调整堆栈深度来控制控件的细节。

控件深度

在应用程序中记录特别高级的点时，堆栈帧计数可能会导致收集大量信息；为了获得更快更清晰的图片，最好限制工具栏上的堆栈深度：

- 断点和标记窗口或
- 记录和分析窗口

访问

功能区	执行 > 工具 > 记录器 > 打开记录器

设置录音堆栈深度

您可以在断点和标记窗口或记录和分析窗口的工具栏上的数字字段中设置记录堆栈深度：

默认情况下，堆栈深度设置为三帧。可以输入的最大深度为 30 帧。

深度与遇到记录标记的堆栈帧相关；因此，在开始录制时，如果堆栈帧为 6，并且堆栈深度设置为 3，则调试器会记录第 6 到第 8 帧。

对于堆栈非常大的情况，建议您首先使用 2 或 3 的低堆栈深度。从那里您可以逐渐增加堆栈记录深度并插入额外的记录标记以扩展图片，直到所有必要的信息都显示。

放置录制标记

本节说明如何放置记录标记，使您能够在两点之间静默记录代码执行。该记录可用于生成序列图。
由于此过程记录了多个线程的执行，因此它在捕获事件驱动的序列（例如鼠标和计时器事件）时特别有用。

访问

功能区	执行>窗口>断点

行动

行动
可以使用不同的记录标记来记录执行流程；有关这些标记的属性和用法的信息，请参见相关链接。
在断点和标记窗口中管理断点。
激活和停用标记。
使用标记集 - 当您创建断点或标记时，它会自动添加到标记集，默认集或您为特定目的创建的集。

注记

- 断点和标记管理主题（软件工程）描述了不同的视角

设置记录标记

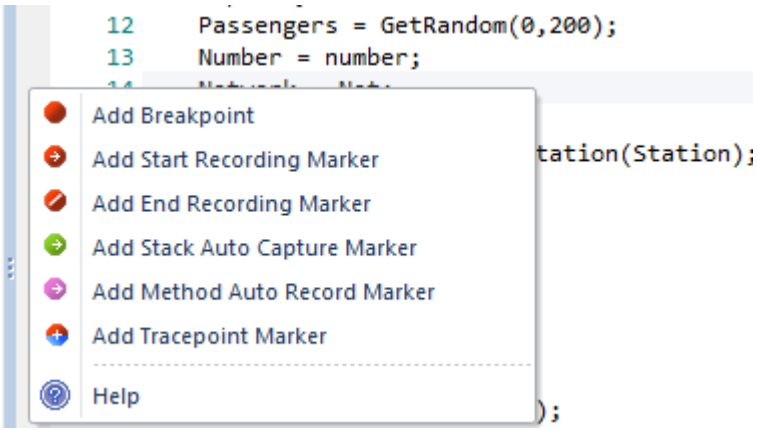
标记在源代码编辑器中设置。它们被放置在一行代码中；当该行代码执行时，执行分析器执行适合标记的记录动作。

访问

使用此处概述的方法之一，显示代码编辑器窗口并加载与所选类或类元素关联的源代码。

功能区	执行 >源> 编辑 > 编辑元素源 执行 >源> 编辑 > 打开源文件
键盘快捷键	在元素上按 Ctrl+E 或 F12 要打开 打开源文件”浏览器，请按 Ctrl+Alt+O

设置录音标记

节	行动
1	在集成源代码编辑器中打开源代码进行调试。
2	找到合适的行，在左边（Breakpoint）空白处右击，调出断点/标记上下文菜单；选择所需的标记类型： 
3	如果已设置开始记录标记，则还必须设置结束记录标记。

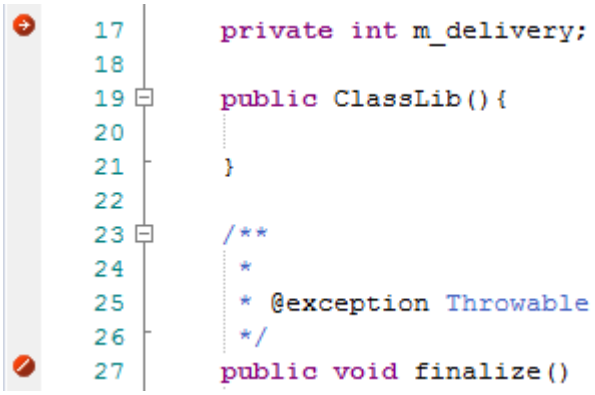
标记类型

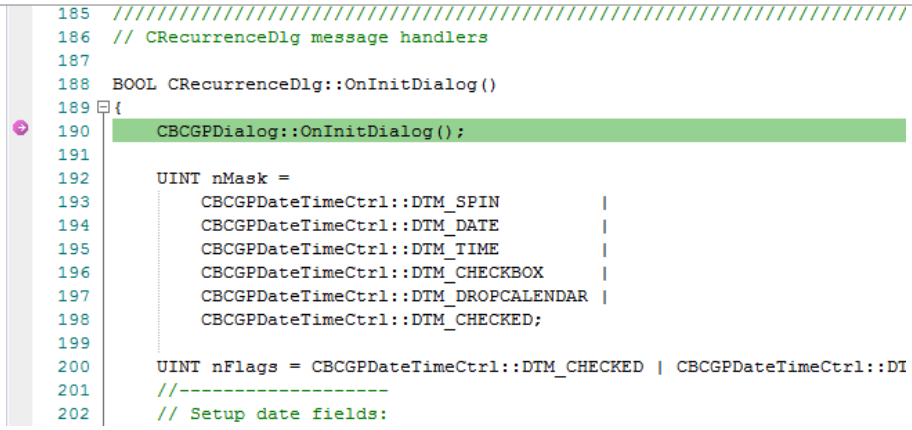
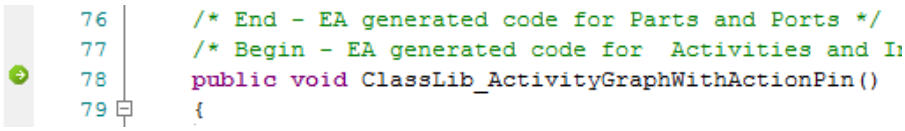
标记真的很棒。不同寻常的是，它们在小心使用时占用空间非常小，它们对正在录制的节目性能的影响可以忽略不计。标记有几种口味（实际上颜色很好），而且总是会添加更多。它们被放置并且在编辑器的左边距可见，所以你需要有一些源代码。

使用到

- 记录一个函数
- 记录函数的一部分
- 使用跨越多个功能的案例
- 记录调用堆栈
- 生成序列图
- 生成测试域图
- 生成类协作图

参考

标记	细节
开始/结束记录标记	将标记放置在要记录的代码的开始行和结束行。这些不必在同一个函数中。  17 private int m_delivery; 18 19 public ClassLib(){ 20 21 } 22 23 /** 24 * 25 * @exception Throwable 26 */ 27 public void finalize() 当程序遇到开始录制标记时，将启动新的录制（相机开始滚动！）。当遇到结束标记时，当前录制结束（这是一个 *take* ）。如何使用这些标记取决于您和您对您所关心的系统的了解。 **高级东西（嵌套标记）：** 如果在录制过程中遇到开始录制标记，但在使用中的堆栈深度值禁止捕获的情况下，将启动单独的录制。每个记录都保存在一个堆栈上。当一个结束时，它被删除。这种技术可以在Enterprise Architect中用于在非常复杂的系统中记录和渲染场景。它类似于拼接视频中的短场景以创建预告片。如果你只想记录一个函数，你应该使用自动记录标记。
方法自动记录标记	A自动记录标记使您能够记录特定的函数。当函数完成时，调试器会自动结束录制。这很好，因为录制是一项密集的操作。 函数标记将开始记录标记和结束记录标记合二为一，因此记录在标记点之后执行，并且总是在该函数退出时停止。

	 记录标记可以嵌套。当录制时点击新的自动标记时，要录制的堆栈深度将扩展为包括当前方法和该函数所需的深度。
堆栈自动捕获标记	 堆栈标记使您能够捕获应用程序中某个点出现的任何唯一堆栈跟踪；它们提供了从何处调用应用程序中某个点的快速且有用的图片。 要在代码中的所需点插入标记，请右键单击该行并选择 添加堆栈自动捕获标记”选项。 每次调试器遇到标记时，它都会执行堆栈跟踪；如果堆栈跟踪不在记录历史记录中，则将其复制，然后应用程序继续运行。
限制记录深度	您可以使用记录器和断点工具栏上的堆栈深度控件来限制任何记录中的帧深度。

断点和标记窗口

使用断点和标记窗口，您可以在记录执行以生成序列图时对可视执行分析进行控制；例如，您可以：

- 启用、禁用和删除标记
- 将标记作为集进行管理
- 组织标记的显示方式，无论是在列表视图中还是按文件或类分组

访问

功能区	执行>窗口>断点

使用标记集

标记集使您能够将标记创建为命名组，您可以将其重新应用于代码文件以用于特定目的。

您可以单独从断点和标记窗口执行某些操作，但要理解和使用标记和标记集，您还应该在 源代码查看器”中显示适当的代码文件（单击类元素并按 F12）。

访问

功能区	执行>窗口>断点：  工具栏图标

使用标记集

行动	细节
使用示例	您可以创建一组自动标记来记录代码中各种函数的操作，并创建一组 Stack Capture 标记来记录导致调用这些函数的调用序列。 然后，您可以根据每个集合下的记录创建序列图。
创建标记集	要从断点和标记窗口创建标记集，请单击  图标上的下拉箭头并选择 新建集”选项。 显示 新断点标记集”对话框；在 输入新集名称”字段中，输入集的名称，然后单击 保存”按钮。 设置名称显示在 设置选项”图标左侧的文本字段中。 或者，您可以从 设置选项”下拉列表中选择 另存为设置”选项，以制作当前所选设置的精确副本，然后您可以对其进行编辑。
访问集	要访问标记集，请单击 设置选项”图标左侧文本字段上的下拉箭头，然后从列表中选择所需的集。 集合中的标记列在断点和标记窗口中。 您通常会在要捕获动作的点之前加载标记集。 例如，要记录涉及特定对话框的序列，当您开始调试时，您将在调用对话框之前加载该集合；一旦您在应用程序中打开对话框，您标记的操作就会被记录下来。
添加标记以设置	要将标记添加到标记集，请将每个必需的标记添加到 源代码查看器”中相应的代码行。 标记会立即添加到断点和标记窗口中当前列出的任何集合中。 对话框中列出的每个标记在 已启用”列中都有一个复选框；新添加的标记会自动启用，但您可以在检查代码时快速禁用和重新启用标记。
储存	当您创建标记集时，它会立即保存在模型中；任何使用该模型的用户都可以访问该集合。 但是，模型始终存在的默认集是个人工作区，不共享，而是存储在模型外部。

从集合中删除标记	右键单击标记并选择 删除断点”选项。
删除一组	如果您不再需要标记集，请在断点和标记窗口中访问它，然后从 设置选项 ”下拉列表中选择 删除选定集”选项。 您还可以通过选择 删除所有集”选项来清除所有用户定义的标记集；将显示一个提示以确认删除。

注记

- 标记集非常简单和灵活，但是由于模型的任何用户都可以使用它们，因此它们很容易损坏；考虑以下准则：
  - 命名一套时，请在名称中使用您的首字母并尽量表明其用途，以便其他模型用户可以识别它的所有者和目的
  - 使用默认以外的集合时，避免过度实验，以免添加集合中的许多临时标记
  - 确保您知道断点和标记窗口中暴露了哪些标记集因为您可以很容易地在不经意间向集合中添加与代码文件无关的标记集是为
  - 在任何集合中，如果您添加了不必保留的标记，请删除它们以保持集合的目的；对于默认集尤其如此，它可以快速累积冗余的临时标记

控制录制过程

记录和分析窗口使您能够控制记录会话。该控件有一个工具栏和一个历史记录窗口，该窗口在捕获时显示记录历史记录。此窗口中的每个条目代表一个由一个或多个函数调用组成的调用序列。

访问

使用此处概述的方法之一打开 记录和分析"窗口。

您还必须打开执行分析器窗口（ 执行>分析器|分析器脚本"），其中列出了模型中的所有脚本；您必须为录制选择并激活相应的脚本。

功能区	执行 > 工具 > 记录器 > 打开记录器

记录器工具栏

您可以通过 Record &功能工具栏访问用于启动、停止和调节执行分析记录会话的功能。

访问

功能区	执行 > 工具 > 记录器 > 打开记录器 探索 >门户> 显示工具栏 > 记录

纽扣

按钮	描述
	显示用于定义录制会话操作的选项菜单。 • Attach to进程- 启用即使没有分析器脚本，该选项会显示一个对话框，您可以通过该对话框选择要记录的进程和要使用的调试平台；您还可以选择在录制期间使用记录标记集和/或状态机 • 生成序列图表记录- 从执行分析器跟踪生成序列/图形状态图 • 从历史生成测试点图表- 从执行分析器跟踪生成测试域图，可以与测试点功能一起使用 • 从历史生成类类图表- 从执行分析器协作生成一个类图，仅描述记录的动作中涉及的那些类和操作（用例） • 生成调用图from History - 从记录历史生成动态调用图，如您在VEA配置文件工作区执行结构分析布局；这在识别所涉及的唯一调用堆栈方面可能比序列图更有用 • 全部生成——从执行分析器trace中生成序列、测试点和协作类图 • 工件- 在当前选择的包中创建一个工件，在当前元素的浏览器窗口中如果您随后将这个元素工件并双击它，则记录在分析类图上的历史记录工件被复制回 • 从文件加载序列历史 - 选择一个 XML 文件，从中恢复以前保存的记录历史 • 将序列历史保存到文件 - 将记录历史保存到 XML 文件
	选择标记集的记录堆栈深度；也就是说，从录制开始的点开始的帧数。
	启动并记录脚本中描述的应用程序；您可以选择在记录期间使用记录标记集和/或状态机。 该图标在活动分析器脚本为调试时启用。
	在调试会话期间执行当前线程的临时手动记录。 使用调试器的 步骤”按钮使用此函数；由于 step 命令而调用的每个函数都会记录到历史记录窗口。 如果没有进行记录并且您当前处于断点（即调试），则启用该图标。

	在调试会话期间执行临时自动记录。 当您单击此图标时，分析器开始记录并且在程序结束、停止调试器或单击停止图标之前不会停止。 如果没有进行记录并且您当前处于断点（即调试），则启用此图标。
	节一个函数，在 History 窗口中记录函数调用，然后退出。 仅对手动录制启用。
	停止记录执行跟踪。
	显示 同步模型”对话框，您可以通过该对话框将模型与记录配置文件操作期间生成的代码文件同步。

使用记录历史

您可以使用 记录和分析”窗口上下文菜单对或从记录会话的结果执行许多操作。

选项

选项	行动
来电显示源	在源源代码查看器中显示调用序列的方法的源代码。
显示源for Callee	在源源代码查看器中显示被序列调用的方法的源代码。
生成图表序列的图形	为记录历史中选择的单个序列生成序列图。
图表生成序列	生成包含记录历史中所有序列的序列图。
清除	清除当前显示在 记录和分析”窗口中的记录历史。
将记录历史保存到文件	将记录历史保存到 XML 文件。 将显示A浏览器窗口，您可以在其中指定 XML 文件的文件路径和名称。
从文件加载记录历史	从 XML 文件加载以前保存的记录历史。 将显示A浏览器窗口，您可以在其中指定要加载的 XML 文件的文件路径和名称。
禁用所有呼叫	禁用 记录和分析”窗口中列出的每个呼叫。
禁用此调用	禁用选定的呼叫。
禁用此方法	禁用所选方法。
禁用这个类	禁用选定的类。
禁用此调用之外的所有调用	禁用 记录和分析”窗口中列出的每个呼叫，所选呼叫除外。
启用所有呼叫	启用 记录和分析”窗口中列出的每个呼叫。
启用这个调用	启用选定的呼叫。
启用此方法	启用所选方法。
启用类	启用选定的类。
帮助	显示记录和分析窗口的帮助主题。

开始记录

当您将执行流程记录为序列图时，您可以通过选择 记录和分析"窗口工具栏上的 记录"图标来开始记录。 录制"对话框显示，录制选项设置为默认值；即当前的Breakpoint and Markers Set，当前分析器中定义的脚本，recording mode as basic。

访问

功能区	执行 > 工具 > 记录器 > 打开记录器： 

记录对话框选项

字段/按钮	细节
记录集	记录标记确定记录的内容。 如果您有要使用的记录集，请单击下拉箭头并选择它。
附加过滤器	调试器使用过滤器从记录历史中排除匹配的函数调用。记录过滤器在分析器脚本定义。 在 附加过滤器"字段中，您可以为此特定运行添加其他过滤器。如果您指定多个过滤器，请用分号分隔它们。
基本记录模式	在基本模式下，调试器在遇到适当的记录标记时记录程序进行的函数调用的历史记录。
跟踪命名类的实例	在跟踪实例模式下，调试器还捕获您指定的类实例的创建。然后它将该信息包含在历史记录中。然后，生成的序列图可以显示该类的每个实例的类，在适当的情况下，函数调用链接到生命线。
跟踪状态Transitions	记录还可以使用指定的状态机图捕获状态变化。状态机图必须作为类的子级存在。 执行分析器捕获类的实例，并在当前记录序列中的函数返回时计算每个实例的状态。
确定	单击此按钮以启动调试器。

节通过函数调用

可以通过单击 记录和分析”窗口工具栏上的 节通过”按钮来执行 节通过”命令。
或者，按 Shift+F6 或选择 执行 >运行>节输入”功能区选项。
'节Through' 命令导致执行'节Into' 命令；如果检测到任何函数，则该函数调用会记录在 历史记录”窗口中。
然后调试器退出，并且可以重复该过程。
此按钮使您无需实际进入函数即可录制通话；该按钮仅在断点和手动记录模式下启用。

嵌套记录标记

当第一次遇到记录标记时，记录从当前堆栈帧开始并继续直到帧弹出，记录额外的帧，直到记录工具栏上定义的深度。考虑这个调用序列：

A -> B -> C -> D -> E -> F -> G -> H -> I -> J -> K -> L -> M -> N -> O -> P -> Q - > R -> S -> T -> U -> V -> W -> X -> Y -> Z

如果您在K处设置一个录音标记并将录音深度设置为 3，这将记录通话序列：

K -> L -> M

如果您还想将调用 X、Y 和 Z 记录为序列图的一部分，您可以在 X 处放置另一个记录标记，分析器将记录：

K -> L -> M -> X -> Y -> Z

但是，当 XYZ 分量的记录结束时（弹出 X 帧），当重新进入 KLM序列的第M帧时，记录将恢复。使用这种技术可以帮助记录图表中的信息由于堆栈深度而被排除在外，并且它可以让您聚焦于要捕获的特定区域。

生成图表序列

本主题描述了您可以对执行分析会话的记录执行哪些操作。

访问

功能区	执行 > 工具 > 记录器 > 打开记录器

参考

行动	细节
生成图	在浏览器窗口中选择适当的包，用于存储序列图。 要从所有记录的序列创建图表，请执行以下任一操作： • 单击 记录和分析"窗口工具栏中的 记录菜单"图标 ()，然后选择 图表从记录生成序列"选项，或 • 右键单击窗口主体并选择 生成序列图表选项 要从单个序列创建图表，请执行以下任一操作： • 单击 记录和分析"窗口工具栏中的 记录菜单"图标 ()，然后选择 图表从记录生成序列"选项，或 • 右键单击序列并选择 从选定序列生成图表"选项
将记录的序列保存到 XML 文件	单击序列，单击 记录和分析"窗口工具栏中的 记录菜单"图标 ()，然后选择 将序列历史保存到文件"选项。
访问现有的序列XML 文件	任何一个： • 单击 Record & Analyze 窗口工具栏中的  ，然后选择 Load序列 from文件"选项，或 • 右键单击屏幕的空白区域，然后单击 加载序列从文件"选项 将显示 窗口"对话框，从中选择要打开的文件。

使用到

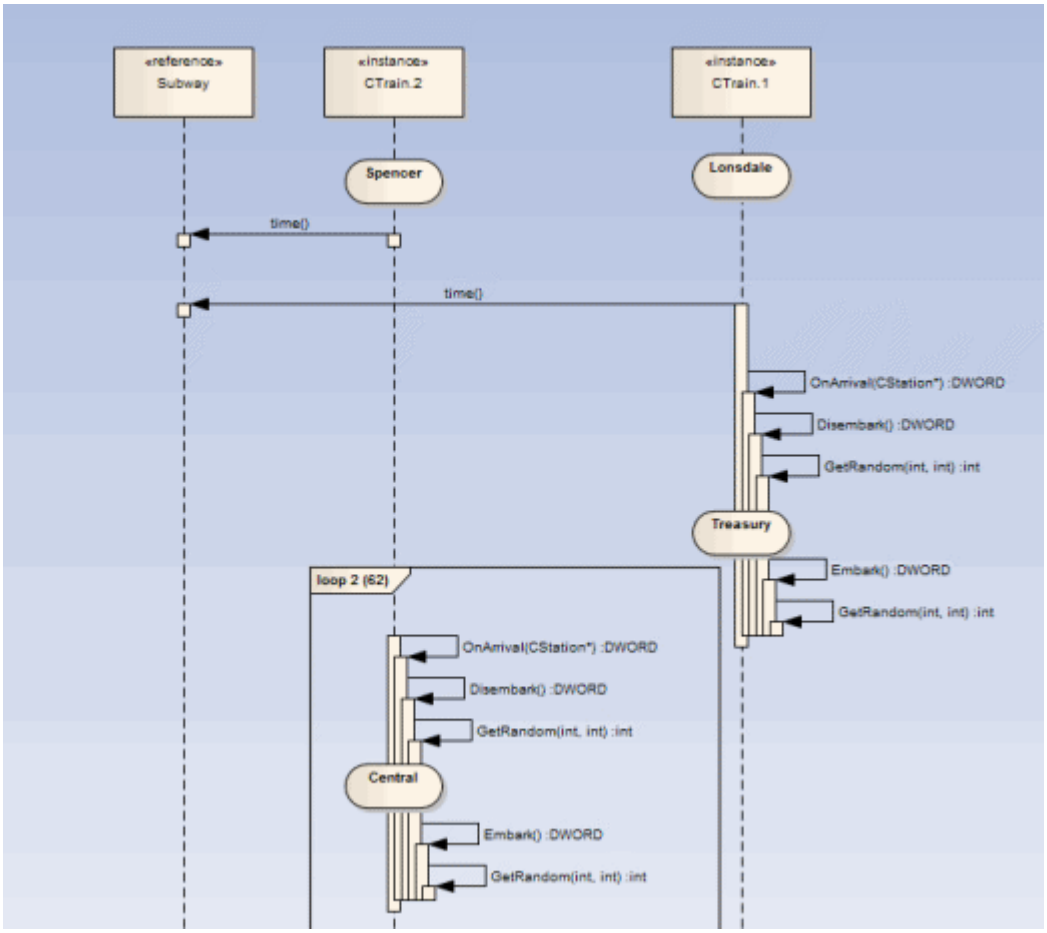
- 从记录的执行分析会话中生成序列图，用于：
- 所有记录的序列或
- 会话中的单个序列
- 将记录的序列保存到文件
- 检索保存的记录并将其加载到 记录和分析"窗口

报告状态Transitions

本节介绍如何生成显示程序执行时状态转换的序列图。

使用到

在程序执行时报告用户定义的状态转换的生成序列图（如生成的示例图所示）



话题
在要上报的类下创建一个状态机。
针对每个状态设置约束以定义要报告的状态变化。

上报状态机

执行分析器可以记录一个序列图，我们知道的。您可能不知道的是，它可以同时使用状态机来检测沿途可能发生的状态转换。这些状态在object生命线上的时间点表示。生命线上的转变也很明显。任何无效或非法的过渡都将用红色边框突出显示。看一看。

进程

首先你为适当的类元素模型一个状态机。

然后，您可以使用每个状态的 约束”选项卡来编写定义每个状态的状态。

这些简单的表达式是使用来自类模型和实际代码库的属性名称形成的。它们不是 OCL 语句。每个表达式应出现在单独的行上。

m_strColor == "蓝色"

然后使用 Recorder 窗口启动调试器。

记录器窗口运行按钮与其他调试器工具栏上的按钮不同。

如果您不知道状态机名称，记录器窗口将允许您浏览状态机机。 '状态转移

' 对话框显示整个模型的状态机列表，您可以在其中找到并选择适当的图表（参见示例）。

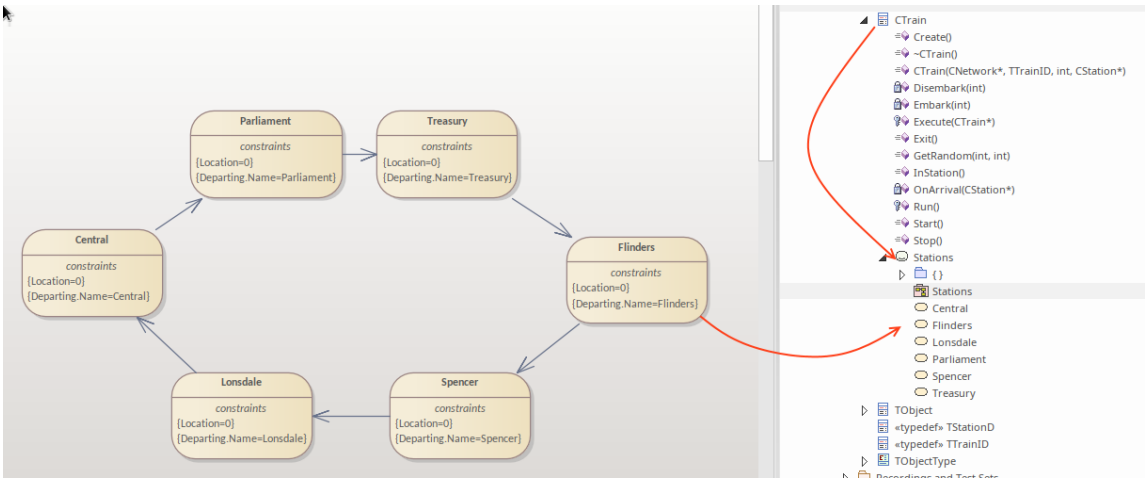
当你生成序列图时，它不仅描绘了序列，而且描绘了序列中各个点的状态变化；参与检测过程的每个类实例都有自己的生命线显示。

示例

Stations状态机显示了墨尔本地铁环路系统内的不同状态。

运行在地铁网络上A列车可以停靠在状态机上表示的任何一个车站。

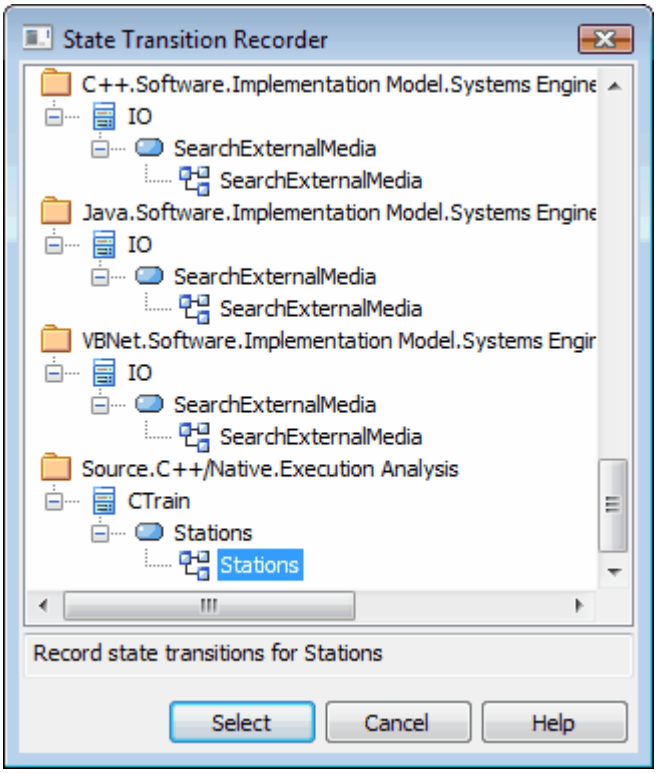
Stations状态机是CTrain类的子对象。



当您浏览 状态转移

”中的图表时转移

Recorder'对话框，层次结构只显示根包、父类和子SubMachine和图表；没有列出其他模型组件。


State Transition Recorder
C++.Software.Implementation Model.Systems Engine
IO
SearchExternalMedia
SearchExternalMedia
Java.Software.Implementation Model.Systems Engine
IO
SearchExternalMedia
SearchExternalMedia
VBNet.Software.Implementation Model.Systems Engir
IO
SearchExternalMedia
SearchExternalMedia
Source.C++/Native.Execution Analysis
CTrain
Stations
Stations
Record state transitions for Stations
Select
Cancel
Help

记录和映射状态修改

本主题讨论如何在一个类下的状态机中针对每个状态设置约束，来定义要记录的状态变化。

示例

这个 状态”对话框的示例是针对称为 属性”的状态； 约束”选项卡打开，显示状态如何与类状态关联。

状态可以A单个约束或多个约束来定义；在示例中，议会状态有两个约束：

约束的值只能针对元素、枚举和string类型进行比较

CXTrain类有一个名为 Location 的int类型的成员，以及一个名为名称类型 CString 的成员；此约束的含义是，在以下情况下，此状态被评估为True ：

- CXTrain类的一个实例存在并且
- 它的成员变量 Location 的值为 0 并且
- 成员变量名称的值为议会

约束的运营商

您可以在约束上使用两种类型的运算符来定义状态：

- 逻辑运算符 AND 和 OR 可用于组合约束
- 等价运算符 {= 和 !=} 可用于定义约束条件

除非另有说明，否则状态的所有约束均受 AND 运算；您可以改为对它们使用 OR 操作，因此您可以将示例中的约束重写为：

位置=0 或

位置=1 并且

Departing.Name!=中央

以下是使用等价运算符的一些示例：

Departing.Name!=Central AND

位置！=1

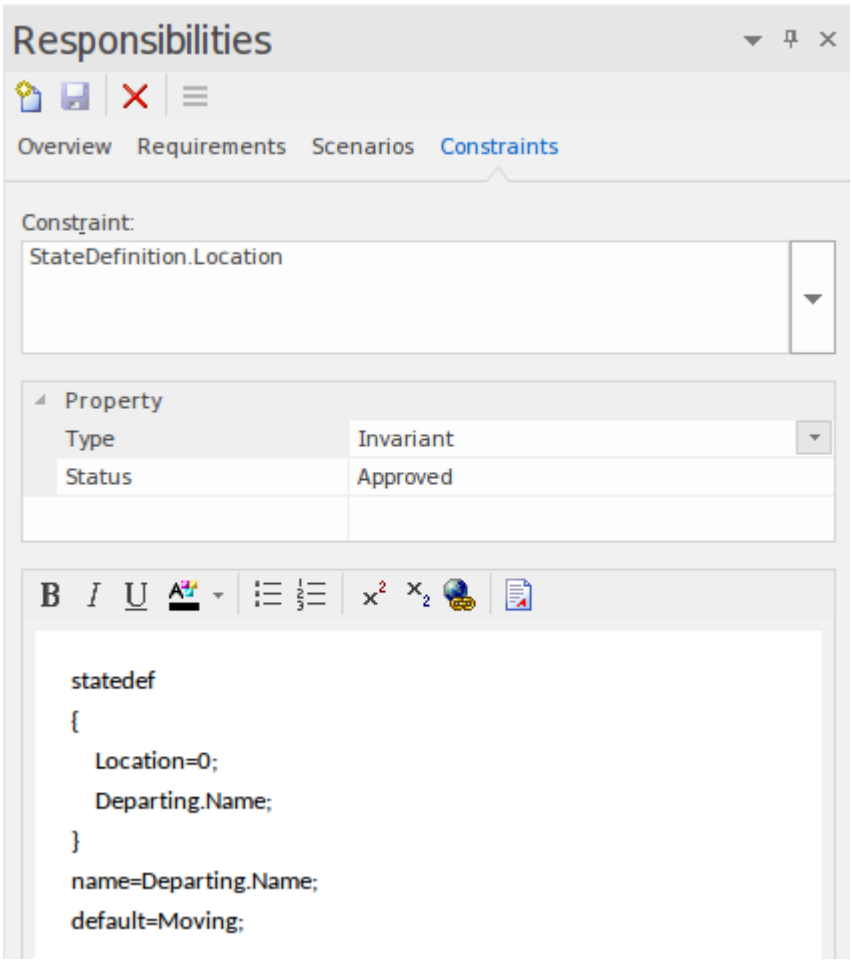
注记

- 字符串周围的引号是可选的；在确定约束的真实性时，字符串的比较始终区分大小写

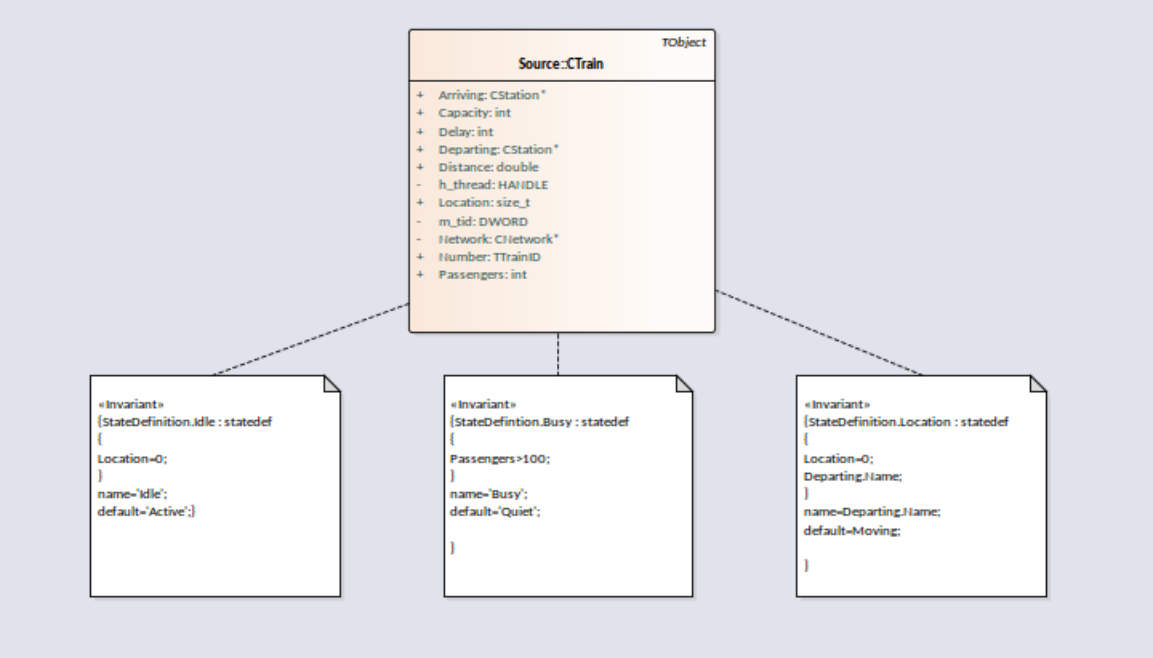
状态分析器

状态分析器是一个可以分析、检测和记录类实例状态的特征。特征通过组合状态定义（在类上定义为约束）和称为状态点的标记来工作。它适用于执行分析器支持的任何语言，包括 Microsoft.NET、Mono、 Java和原生 C++。

我们首先选择一个类并编写我们的状态定义。



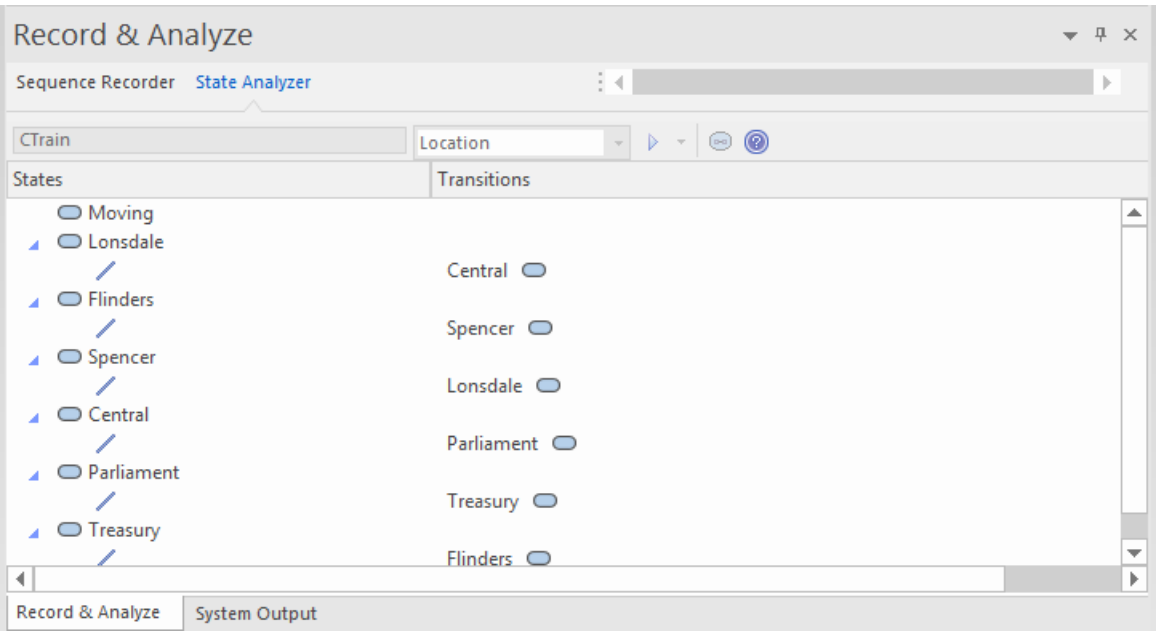
我们可以通过将类放在图表上并链接到本身链接到特定状态定义约束的类注记来获得我们定义的所有状态定义的图片。我们将在后面的部分中解释如何做到这一点。



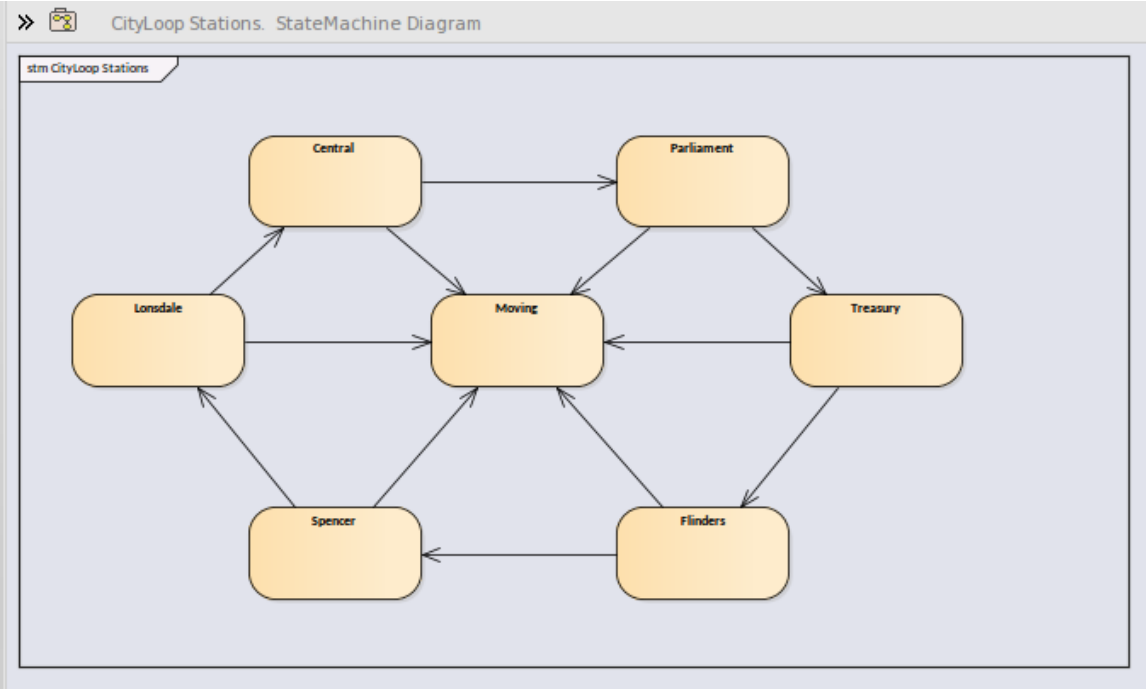
通过在相关源代码中放置一个或多个标记来设置状态点。

```
73 DWORD   CTrain::OnArrival( CStation* S)
74 {
75     Departing = S;
76     Location = 0;
77     Delay = (Disembark(GetRandom()) + Embark(GetRandom()));
78     DWORD ScheduleTime = Network->TimeAtStation(Departing);
79     if(Delay > (int)ScheduleTime)
80         return Delay;
81     return ScheduleTime;
82 }
```

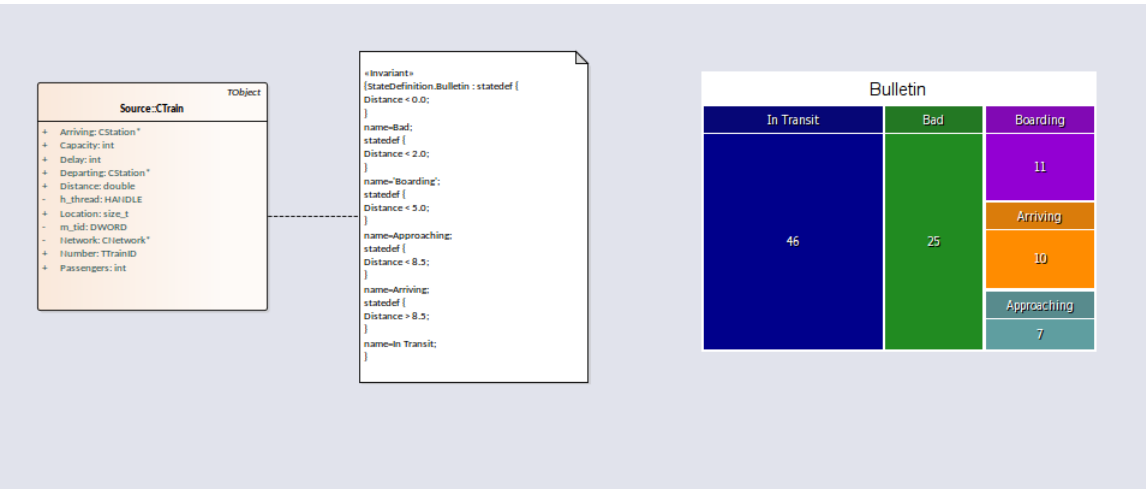
要分析的程序运行使用状态分析器控件运行。当执行分析器遇到任何状态点时，分析类的当前实例。在实例的值域与状态定义匹配的情况下，记录一个状态。每次实例变化时，都会检测到新的状态。控件列出发现的每个状态。在每个状态下，控件列出了由类的实例进行的到其他状态的离散转换集。



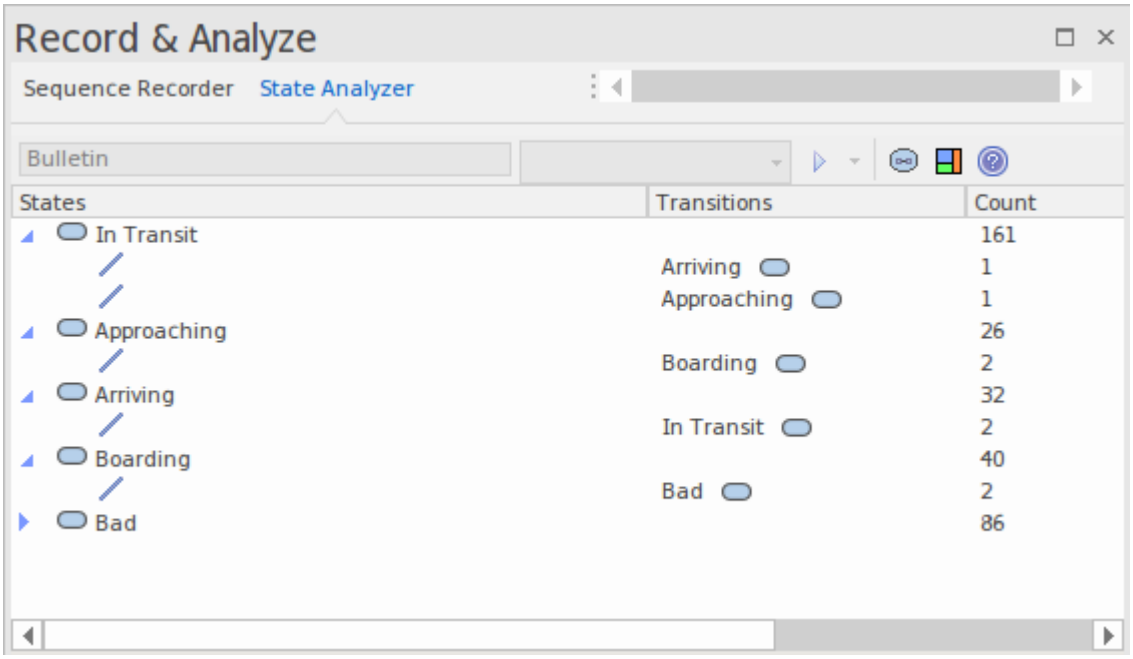
该信息可用于创建状态机。



使用相同的信息，我们可以轻松生成热图。这个例子展示了一个'Train'类，它的'Bulletin'状态定义（作为一个链接的注记），以及它产生的热图。地图中的数字是百分比。从地图中我们可以观察到火车在 46% 的时间处于 "在途"状态。



这是生成我们的热图的 "公告"状态定义的分析。



访问

功能区	执行 > 工具 > 记录器 > 打开记录器 >状态分析器 设计>元素>编辑>约束

状态定义

状态定义由类元素的约束属性组成。约束类型应命名为*StateDefinition.name* ，其中 'name' 是您选择的定义标题。 These titles are listed in the combo box of the状态分析器whenever a类is selected.在运行程序之前，您可以从此组合框中选择一个定义。我们示例中的状态定义名为 状态"。它根据 CTrain类实例的位置定义状态。

状态定义由一个或多个规范组成。每个状态规范都以关键字 "statedef"开头，然后是一个或多个语句。语句定义描述状态的约束，以及可选的变量，其值可用于命名状态。语句用大括号括起来，并以分号结尾，如下所示：

状态定义 {

位置=0；

出发。名称；

}

使用变量命名状态

在此示例中，"Location"是一个常量，"Departing.name"是一个变量。附加语句遵循约束并指示要从变量值分配的状态名称。这是命名指令的定义。

状态定义 {

位置=0；

出发。名称；

}

名称=出发。名称；

使用文字命名状态

在此示例中，状态定义仅包含常量，并且状态使用文字命名。

```
状态定义 {
位置=100；
}
名称='中央';
```

定义多个状态规范A单个状态定义。

```
状态定义 {
乘客> 100；
}
名称=忙；
状态定义 {
乘客>= 50；
}
名称=安静；
状态定义 {
乘客<50；
}
名称=非常安静；
状态定义 {
乘客= 0;
}
名称=空闲；
```

默认状态

状态定义可以指定A默认的 ć̋atch all”状态，当没有其他状态为真时，该状态将描述状态的状态。您使用类似于以下的语句为定义定义默认状态：

```
状态定义 {
位置=0；
出发。名称；
}
名称=出发。名称；
默认=移动；
```

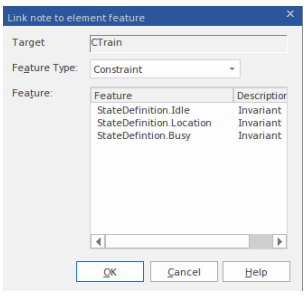
在此示例中，在执行过程中，检测到具有非零 位置”属性的任何实例都将被记录为处于 移动”状态。

您可以通过禁用状态分析器工具栏下拉菜单上的 包括默认状态”选项来选择排除默认状态的记录。这将排除到正在记录的任何 默认”状态的转换。

在显示状态的类元素上创建注记定义定义

本节介绍如何创建显示为类定义的所有状态定义的类图。

行动

显示类图	打开现有的类图或创建一个新的。
创建指向类元素的链接	将感兴趣的类作为链接拖到图表上。
创建一个注记元素	在图表上创建一个注记元素并将其链接到类。
将注记链接到状态定义	选择注记和类之间的链接，并使用其上下文菜单，选择 将注记链接到元素特征”选项。
选择要在注记上显示的定义 	在元素对话框中，从下拉组合中选择 约束”。将列出任何已定义的状态定义供您选择。
重复	对类上的任何其他状态定义重复过程。

同步

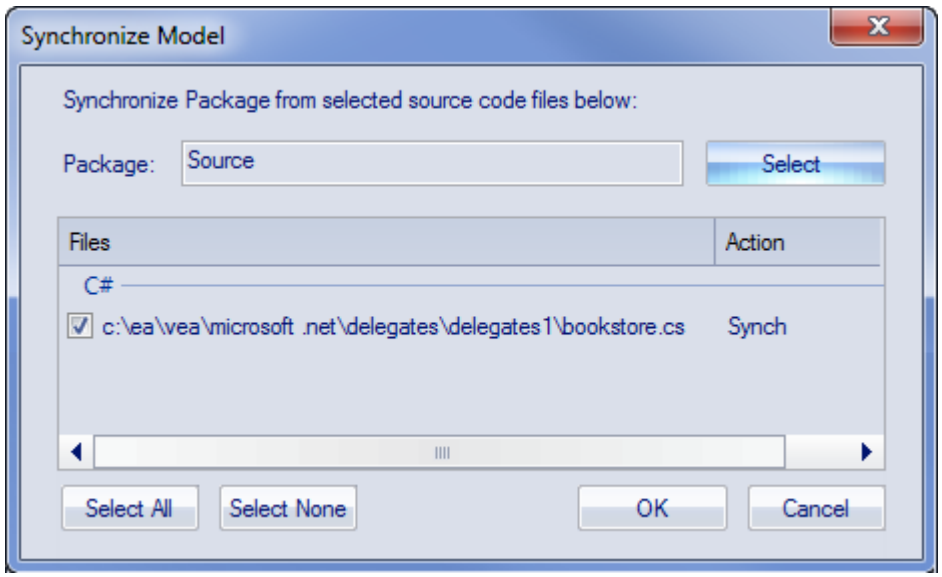
记录产生了许多资产，记录历史是主要的。记录还标识了一组源代码文件。该套件可用于制作类图和测试域图，也可用于同步您的模型。

同步模型A图表元素和类模型之间提供快速准确的导航。

访问

功能区	执行 > 工具 > 记录器 > 打开记录器 > 工具栏  按钮
上下文菜单	右键单击 记录和分析”窗口 |将模型与源代码同步

同步模型



字段/按钮	行动
包	单击 选择”按钮并选择要对代码文件进行逆向工程的目标包。
文件/行动	列出在一个或多个记录期间识别的文件。每个文件旁边都列出了相应的操作。
全选	单击此按钮以选中 文件”列表中每个文件的复选框。
选择无	单击此按钮可清除 文件”列表中每个文件的复选框。
确定	单击此按钮开始操作。将显示同步进度。
取消	单击此按钮可中止同步并关闭对话框。