Модуль 4: Мониторинг и отладка приложений

Обзор

- Управление системным Event Log
- Работа с Application Processes
- Managing Application Performance
- Отладка приложений
- Трассировка приложений
- Embedding Management Information and Events

Урок 1. Управление системным Event Log

- Что такое системный Event Log?
- Как писать и читать записи из Event Log
- Обсуждение: Выясняем как управлять Event Log.

Что такое системный Event Log?

Event log хранит записи о различных событиях и действиях возникающих в приложении. Типы существующих системных Event Log:

- Система
- Безопасность
- Приложение

Как читать и писать данные в Event Log

Свойство	Описание
Log	Данное свойство указывает имя лога из которого производится чтение
MachineName	Данное свойство указывает имя компьютера на котором лог существует
Entries	Данное свойство предоставляет доступ к содержанию лога
Source	Данное свойство настраивает имя источника, сопоставленного с event log
WriteEvent	Данный метод позволяет записать «локализированные» данные в event log
WriteEntry	Данный метод позволяет записать данные в event log

Обсуждение: Выясняем как управлять Event Log

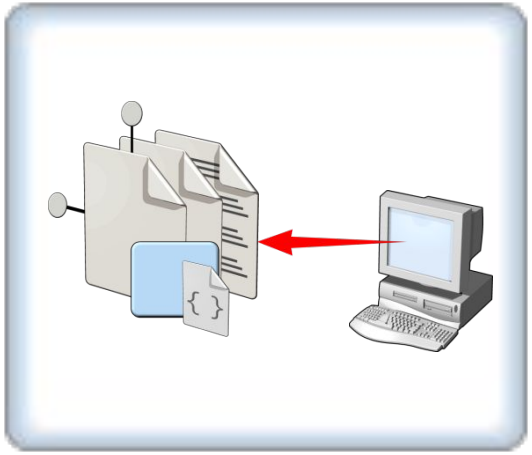
- Что такое event log?
- Зачем использовать event log?
- Как считывать данные из event log?
- Как вносить данные в event log?
- Что такое event source?
- Что такое event entry?

Урок 2. Работа с процессами приложений

- Как получить список процессов
- Как получить информацию о текущем процессе
- Как получить список модулей процесса
- Как запустить и остановить процесс
- Обсуждение: Определить методы работы с процессами приложений

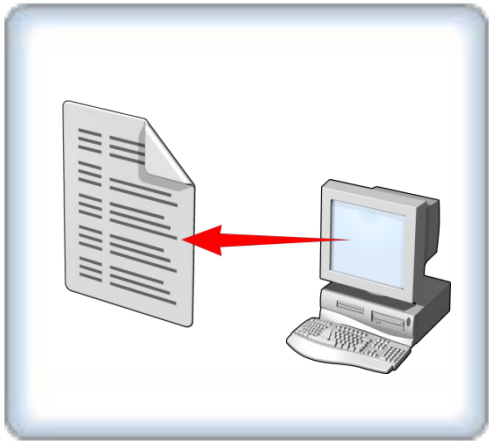
Как получить список процессов?

GetProcess Method	Принадлежит классу Process и получает список всех процессов запущенных на компьютере.



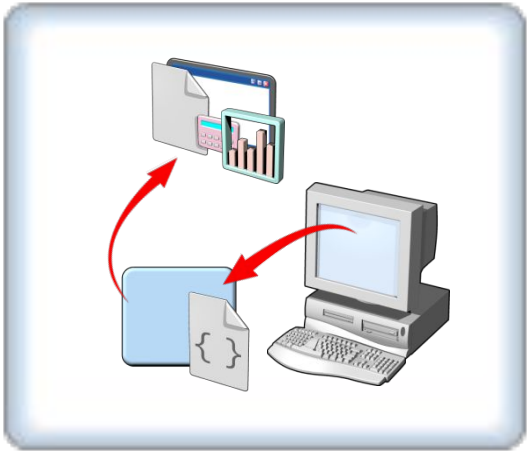
Как получить информацию о текущем процессе?

GetCurrentProcess Method	Получает информацию о текущем процессе. Информация может быть как о длительности выполнения процесса, так и задействованных ресурсах.



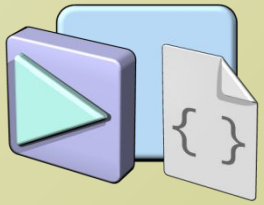
Как получить список модулей процесса?

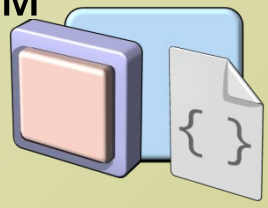
Modules Method	«Получает» список всех модулей и информацию о конкретном модуле, загруженном в процесс.



Как запустить и остановить процесс?

Start Method	Запускает и ассоциирует с компонентом процесс
Kill Method	Останавливает и высвобождает все ассоциированные с процессом ресурсы

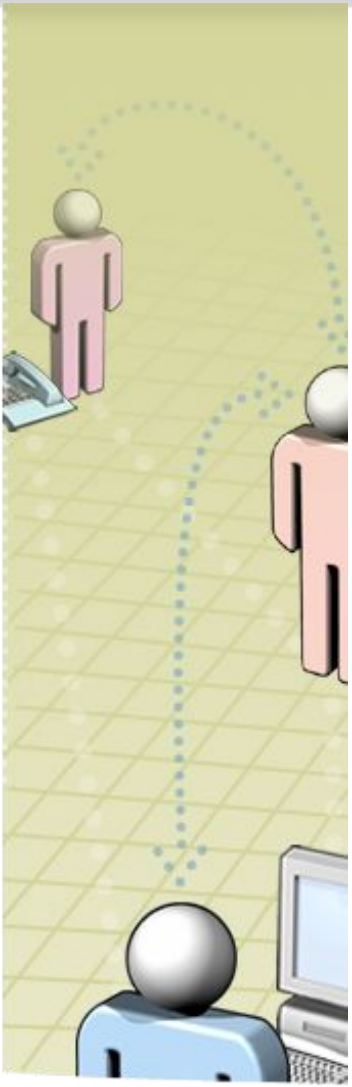




Практика

- Написать приложение, отображающее список процессов, запущенных на локальном компьютере.

Обсуждение



- Что такое процесс?
- Как запустить и остановить процесс?
- Как получить информацию о текущем процессом?
- Как получить информацию о запущенных в системе процессах?

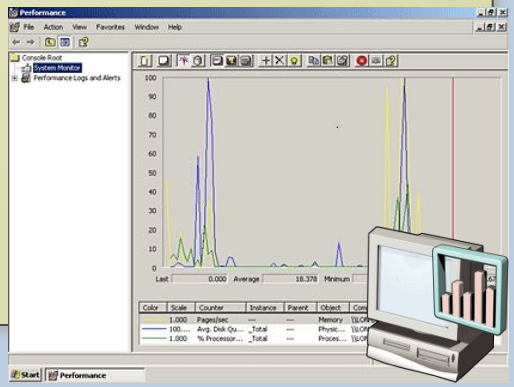
Урок 3. Производительность приложений

- Мониторинг производительности приложений используя Performance Monitor
- Как информация о производительности может кастомизироваться используя Performance Counter классы.
- Обсуждение: Определяем возможности для мониторинга производительности приложений.

Мониторинг производительности приложений используя Performance Monitor

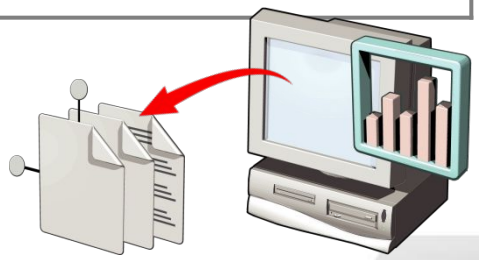
Мониторинг производительности может быть осуществлен с помощью соответствующих утилит и методов. Подопытными в нашем виртуальном «Большом Брате» (Дом2, кому как больше нравится) являются:

- CPU
- Hard disk drive
- Memory
- Processes and Threads

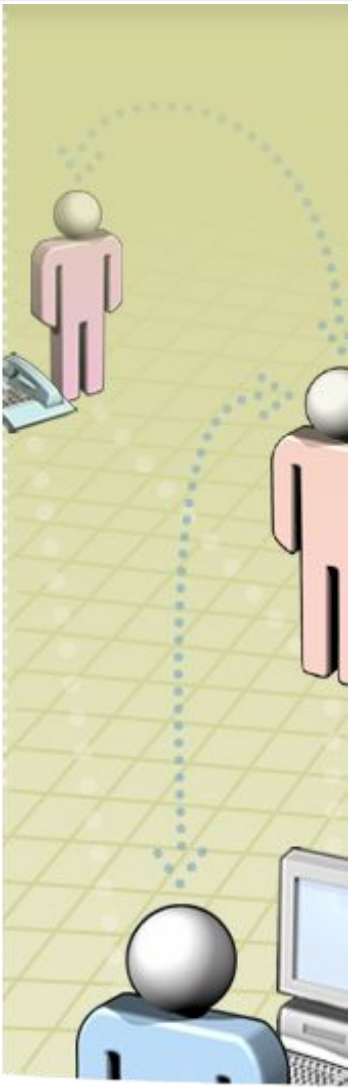


Как информация о производительности может customized используя Performance Counter

Classes	Description
PerformanceCounter	Управляет данными полученными от счётчиков производительности
PerformanceCounterCategory	Управляет категориями
CounterCreationData	Создаёт счётчики производительности и т.д.



Обсуждение



- Ну и как же Вы будите мониторить производительности приложений, используя стандартные механизмы, предоставляемые платформой Microsoft .NET Framework, для операционных систем семейства Windows?

Урок 4. Отладка приложений

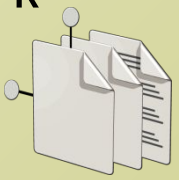
- Демонстрация: Просмотр сообщений об ошибке используя VS Debugger
- Как Debugger класс используется для программной отладки
- Как Debug класс используется для программной отладки
- Debugger Attributes, зачем они?
- Управление стэком используя StackFrame and StackTrace Classes
- Обсуждение:Возможности отладки

Демонстрация: Просмотр сообщений об ошибке используя VS Debugger

- Демо… если получится на данном EeePc.

Как Debugger класс используется для программной отладки

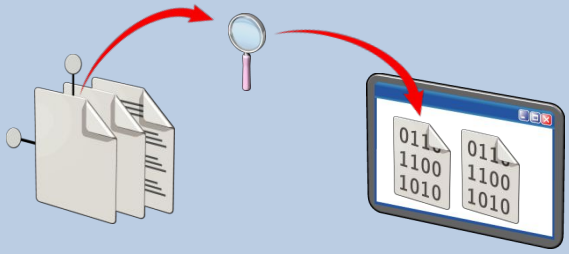
Debugging Process / Процесс отладки	Процесс поиска и исправления ошибок в коде. В общем случае.
Debugger Class	Запускает отладчик из самого приложения и присоединяет его к запущенному процессу.



Как Debug класс используется для программной отладки

Debug содержит методы и свойства, которые помогают в отладке нужных блоков кода или методов. Основные свойства Debug:

- Assert
- WriteLine
- WriteLineIf

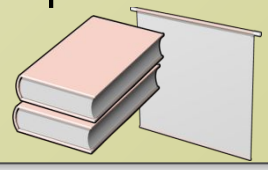


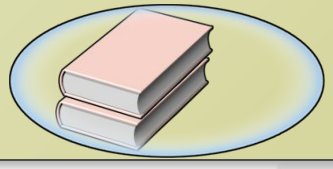
Debugger Attributes, зачем они?

Debugger Attributes	Используются для конфигурирования пользовательских типов, влияет на отладку – ну кто бы мог подумать!

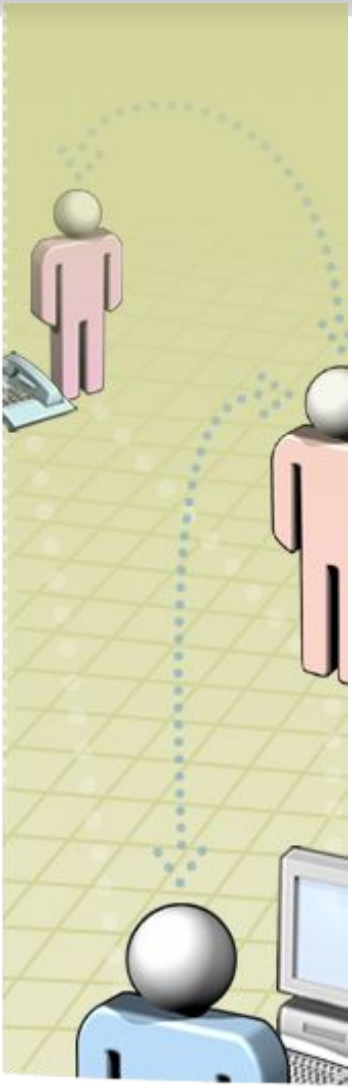
Управление стэком используя StackFrame and StackTrace Classes

StackFrame	Предоставляет информацию о фрейме стэка, что является представлением вызова функции в стэке текущего потока.
StackTrace	Управляет двумя и более фрэймами.





Обсуждение:Возможности отладки



- Что есть отладка
- Debug class?
- Debugger class?
- Что есть the call stack?

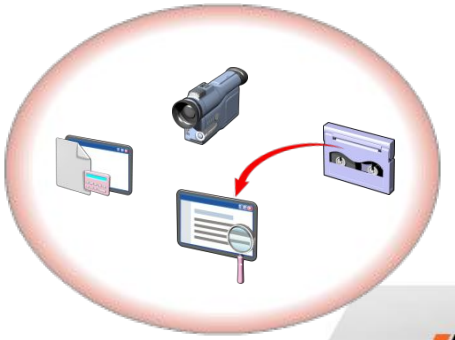
Трассировка приложений

- Что такое трассировка?
- Как использовать программную трассировку в приложениях с помощью the Trace Class
- Как идентифицировать источник трассировки используя TraceSource
- Как информация трассировки настраивается используя Trace Switch
- Как информация трассировки направляется используя Trace Listener
- Как информация трассировки категоризируется используя CorrelationManager
- Обсуждение: Возможности трассировки

Что такое трассировка?

Трассировка приложений	Процесс монитоинга и записи событий приложения.

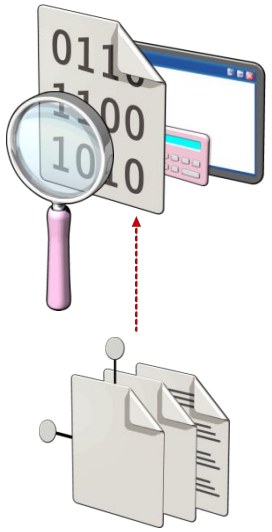
Суперская картинка



Как использовать программную трассировку в приложениях с помощью the Trace Class

Trace класс отслеживает выполнение приложения, основные методы:

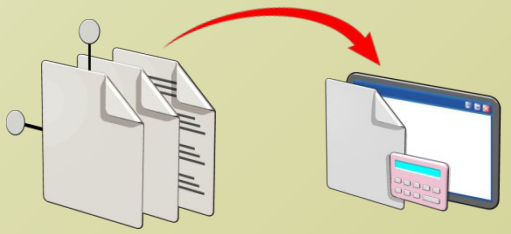
- Assert
- WriteIf
- Fail
- Write
- WriteLine
- WriteLineIf



Как идентифицировать источник трассировки используя TraceSource

TraceSource класс помогает в трассировке исполняемого кода и ассоциировании сообщений трассировки с источником. Основные методы и св-ва:

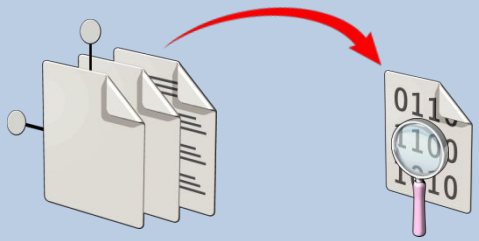
- TraceEvent
- TraceData
- TraceInformation
- Name
- Switch
- Listeners



Как информация трассировки настраивается используя Trace Switch

Trace switchesмогут включать, выключать и фильтровать трассировочную информацию. Типы переключателей:

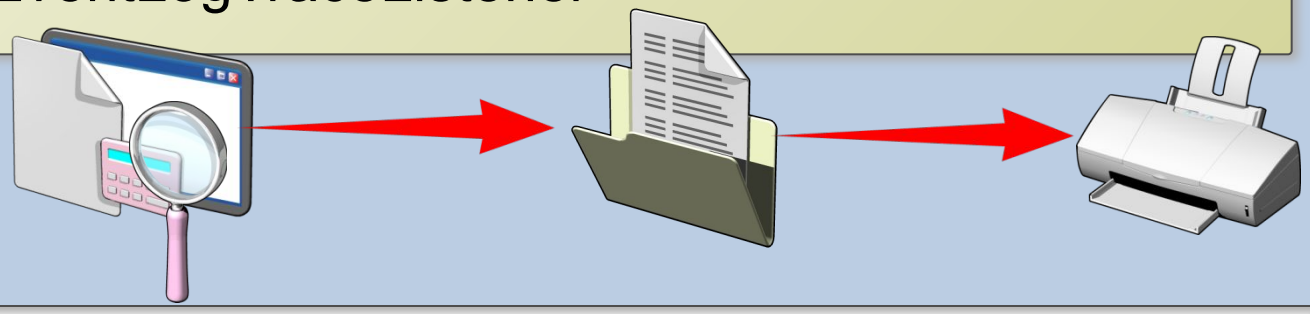
- TraceSwitch
- BooleanSwitch



Как информация трассировки направляется используя Trace Listener

Trace listener классы направляют информацию о трассировки туда, где не светит солнце, т.е. сохраняют. Существующие классы :

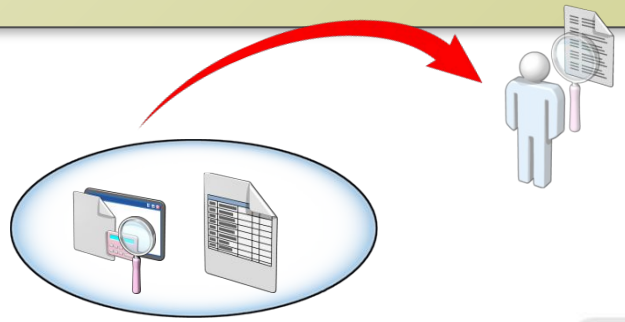
- TraceListener
- XmlWriterTraceListener
- DelimitedListTraceListener
- EventLogTraceListener



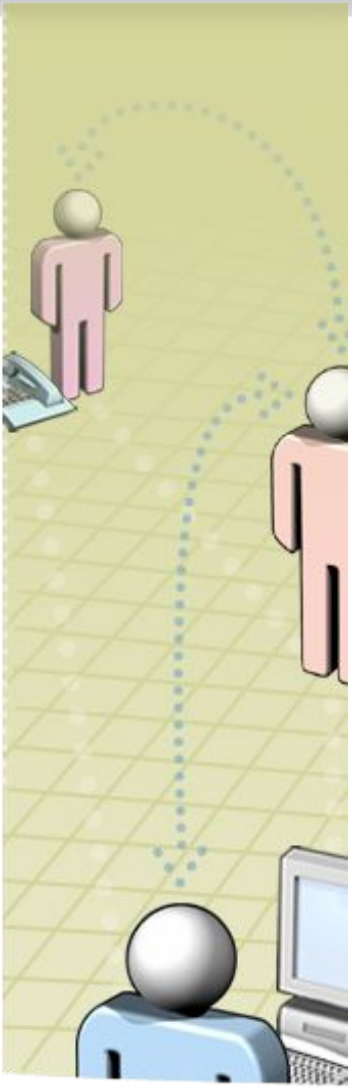
Как информация трассировки cetegorized используя CorrelationManager

CorrelationManager Class

Группировка и классификация информации трассировки для последующего анализа – вот его жизненное кредо.



Обсуждение: Возможности трассировки



- Что есть трассировка?
- Trace class?
- Trace source?
- CorrelationManager class?

That’s all folks.