

How to Debug Linnor Applications

1	Introduction.....	1
2	Enable Debugging Log	1
3	Record Performer States	3

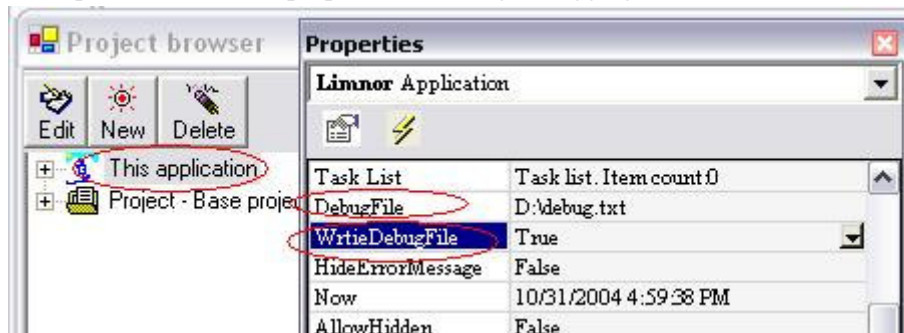
1 Introduction

When you are testing your Linnor applications sometimes you may not get the results you expected. In such cases you are facing a task of debugging: find out what are going on and detect the causes of the problems.

Linnor may record all actions your Linnor application executes and performer states when an action is executing. You may examine the records and find out which parts surprise you and thus detect the causes of problems.

2 Enable Debugging Log

The Application performer has two properties for doing debugging:



DebugFile – You need to give a file name for this property. All debug logging information will be appended to this file.

WriteDebugFile – You need to set this property to True to enable the logging of debugging information

Below are parts of debug log file from one application:

```
Started===== - 2004-10-28 09:02:30Z
Task begin: OnAppStart fired by: Linnor - OnStart - 2004-10-28 09:02:33Z
  Action 1: tmLock.Stop - 2004-10-28 09:02:33Z
    tmLock.Stop
  Action 2: tmUnlock.Stop - 2004-10-28 09:02:33Z
    tmUnlock.Stop
```

```
Action 3: Application.SetLocked - 2004-10-28 09:02:33Z
  Limnor.SetVariable
  p0:[Locked]=Locked
  p1:[True]=True
Action 4: Application.ClearCanStart - 2004-10-28 09:02:33Z
  Limnor.SetVariable
  p0:[CanStart]=CanStart
  p1:[False]=False
Action 5: Application.ReadTerminalID - 2004-10-28 09:02:33Z
  Limnor.SetVariable
  p0:[TerminalID]=TerminalID
  p1:[Config1.SYS(TerminalID)=1]=1
Action 6: dqTerminal.SetTerminalID - 2004-10-28 09:02:33Z
  dqTerminal.SetParameter
  p0:[@TerminalID]=@TerminalID
  p1:[Config1.SYS(TerminalID)=1]=1
Action 7: duSetSessionID.SetTerminalID - 2004-10-28 09:02:33Z
  duSetSessionID.SetParameter
  p0:[@TerminalID]=@TerminalID
  p1:[Config1.SYS(TerminalID)=1]=1
Action Condition:Limnor - Exiting
Action 8: tmUnlock.Start - 2004-10-28 09:02:33Z
  tmUnlock.Start
Task end: OnAppStart - 2004-10-28 09:02:33Z
Task begin: ShowMainPage fired by: PageHome - AfterShow - 2004-10-28 09:02:33Z
  Disabled - 2004-10-28 09:02:33Z
  Task Condition:Limnor - Variables, Locked
Task end: ShowMainPage - 2004-10-28 09:02:33Z
```

In the log file for every event it records following information:

1. The task or action executed at the event
2. The event name
3. Time

For example,

```
Task begin: OnAppStart fired by: Limnor - OnStart - 2004-10-28 09:02:33Z
```

It shows the task executed is named OnAppStart. The event name is the OnStart. The event owner is Limnor, that is, the Application performer. The time of the event is 2004-10-28 09:02:33.

For each action executed, it records action name, action owner and method name, parameter values.

For example:

```
Action 5: Application.ReadTerminalID - 2004-10-28 09:02:33Z
  Limnor.SetVariable
```

```
p0:[TerminalID]=TerminalID  
p1:[Config1.SYS(TerminalID)=1]=1
```

It shows the action name is Application.ReadTerminalID; the action owner is Limnor, that is, the Application performer; the action method is SetVariable; the first parameter value is TerminalID; the second parameter value is 1; it also shows that the value of the second parameter is from performer Config1 which represents an INI file, the section name is SYS and the item name is TerminalID. Because we know that for SetVariable method the first parameter is variable name and the second parameter is variable value, we know that this action sets the variable TerminalID to 1.

If an action or a task is disabled the log file will show “disabled” and show the property which disables the action or task. For example:

```
Task begin: ShowMainPage fired by: PageHome - AfterShow - 2004-10-28 09:02:33Z  
  Disabled - 2004-10-28 09:02:33Z  
  Task Condition:Limnor - Variables, Locked  
Task end: ShowMainPage - 2004-10-28 09:02:33Z
```

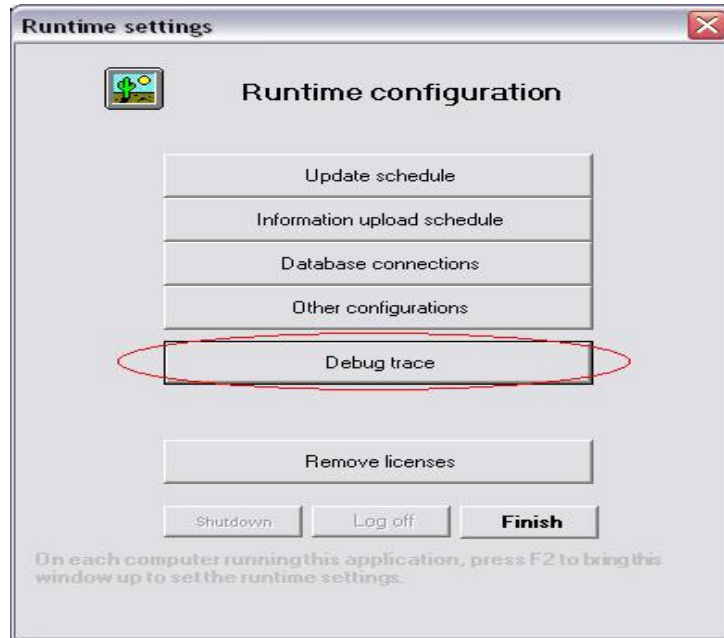
Here it shows the task ShowMainPage was called at event AfterShow of PageHome, but it was disabled. It was disabled by Variables property of Limnor (the Application performer), and the variable name is Locked.

3 Record Performer States

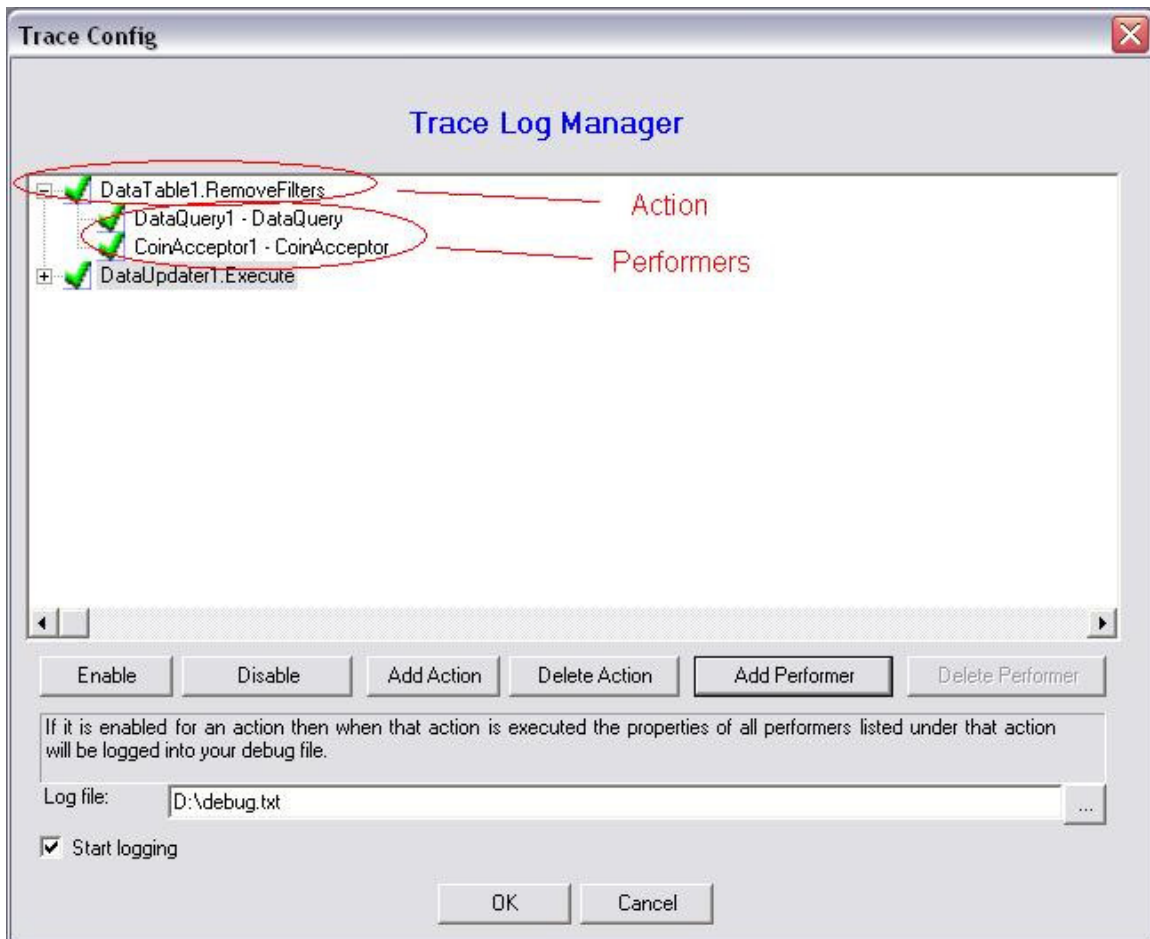
You may record performer states when an action is executed. You may specify which performers you want to trace when an action is executed.

At design time, you may use menu “Project | Runtime settings” to bring up configuration options. At runtime, you may press F2 key to bring up configuration options.

Click “Debug trace” button to specify actions and performers:



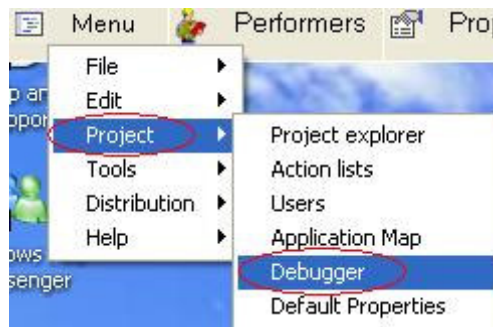
The Trace Log Manager dialog box appears:




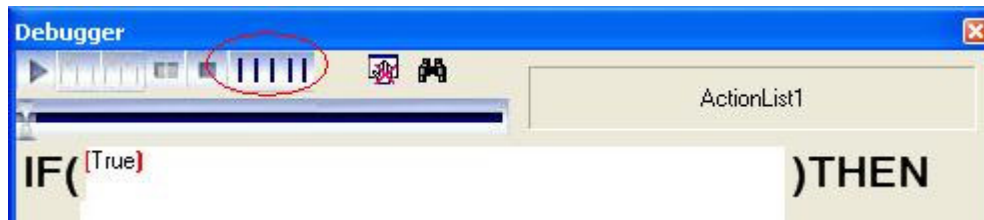
Using this dialogue box, you may specify which actions you want to monitor and for each action which performers you want to trace.

4 Debug Action List

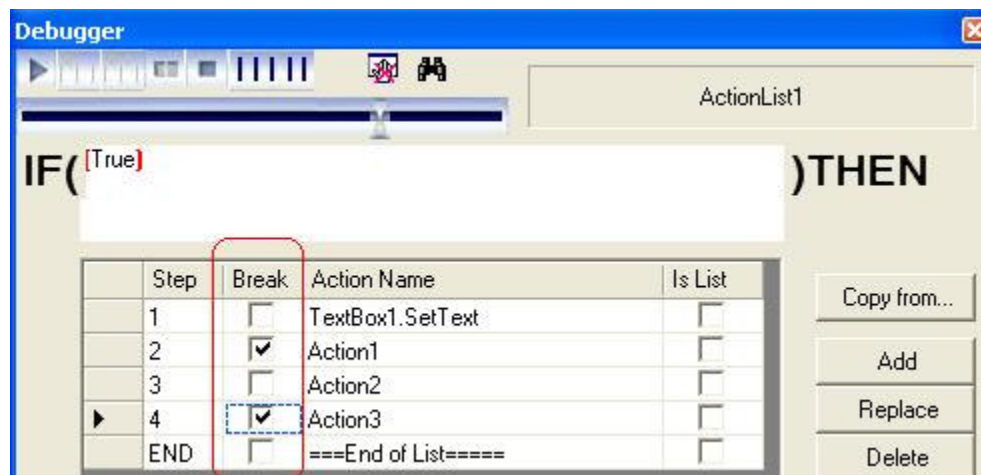
To execute an action list action by action, open debugger by menu “Project | Debugger”:




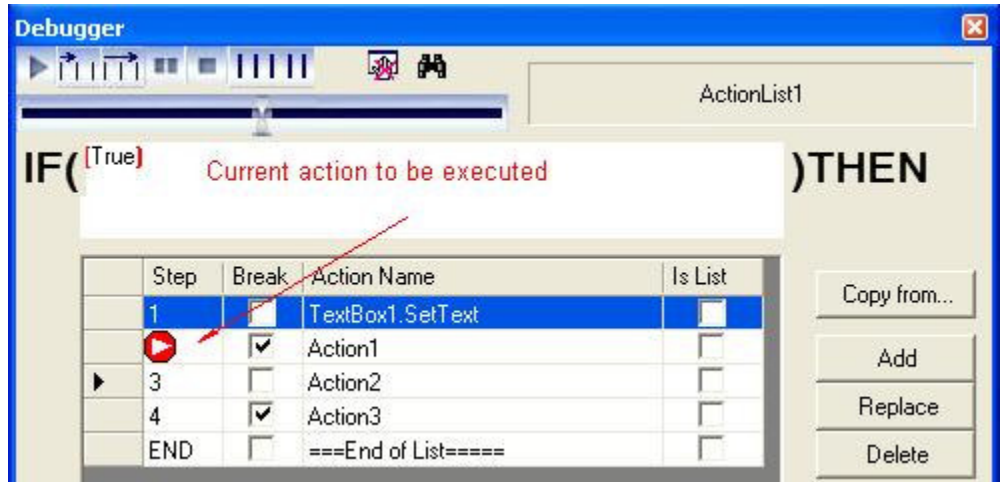
Click  to load the action list:



Before clicking  to execute the action list set break points on desired actions:

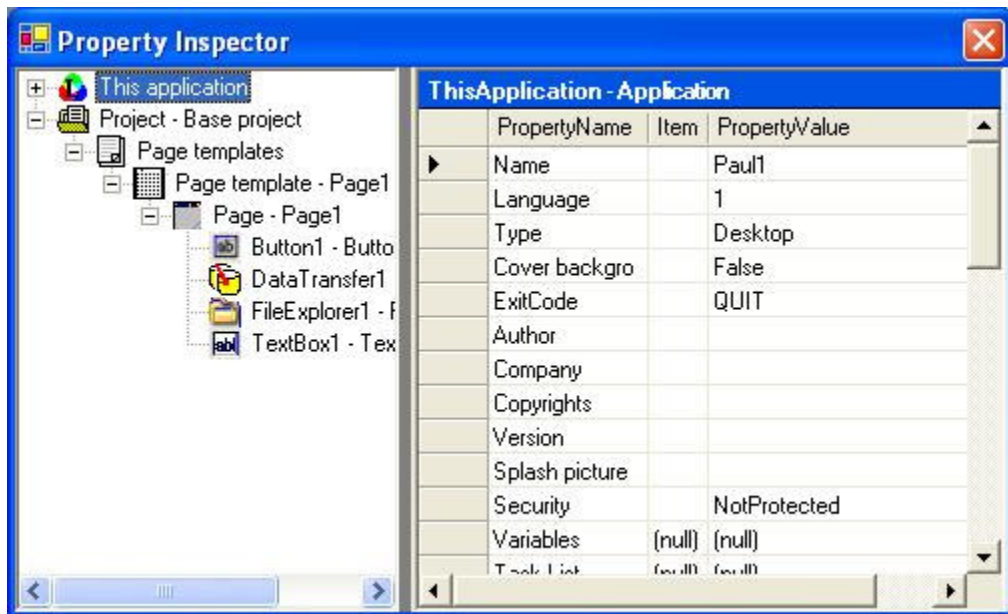


Click , the execution will stop at the first break point:



Click to execute the next action. Click to execute the following actions and stop at the next break point.

When the execution stops at an action, click to view the properties of performers:



You may execute action by action and watch the properties to see if they change in the ways as you expected.