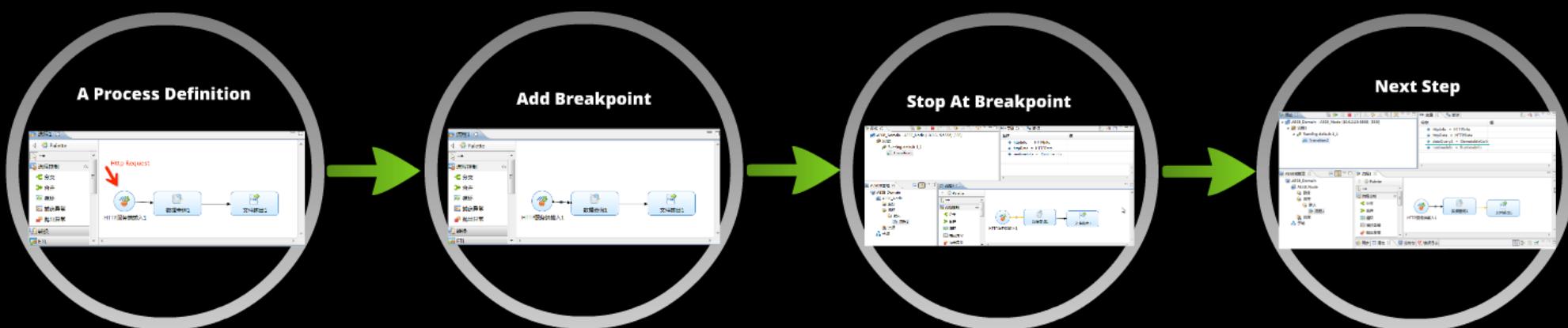


Demo Of Debug A Process



Implementing debug functionality for process editor

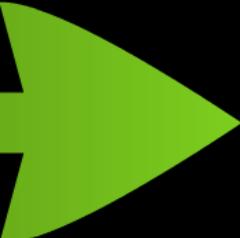
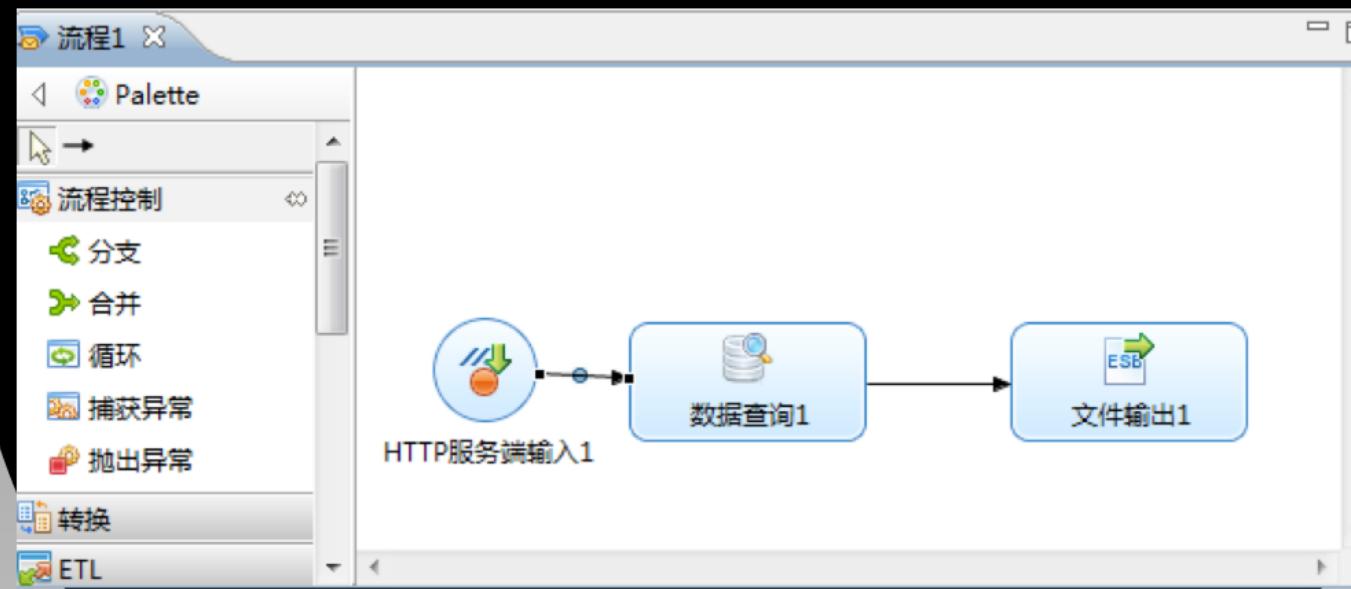
Demo Of Debug A Process



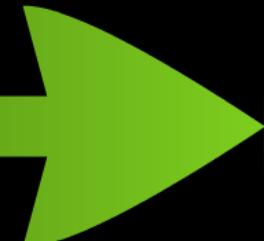
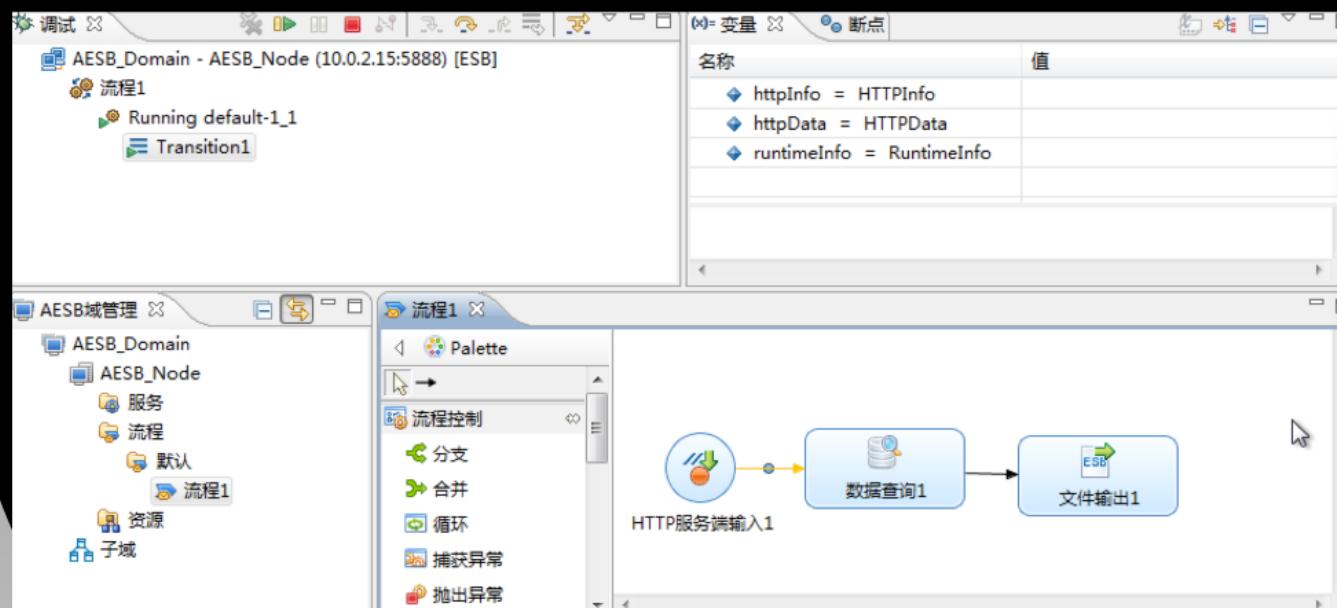
A Process Definition



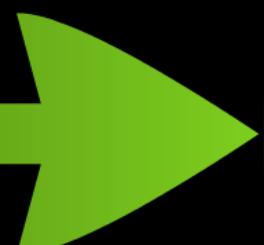
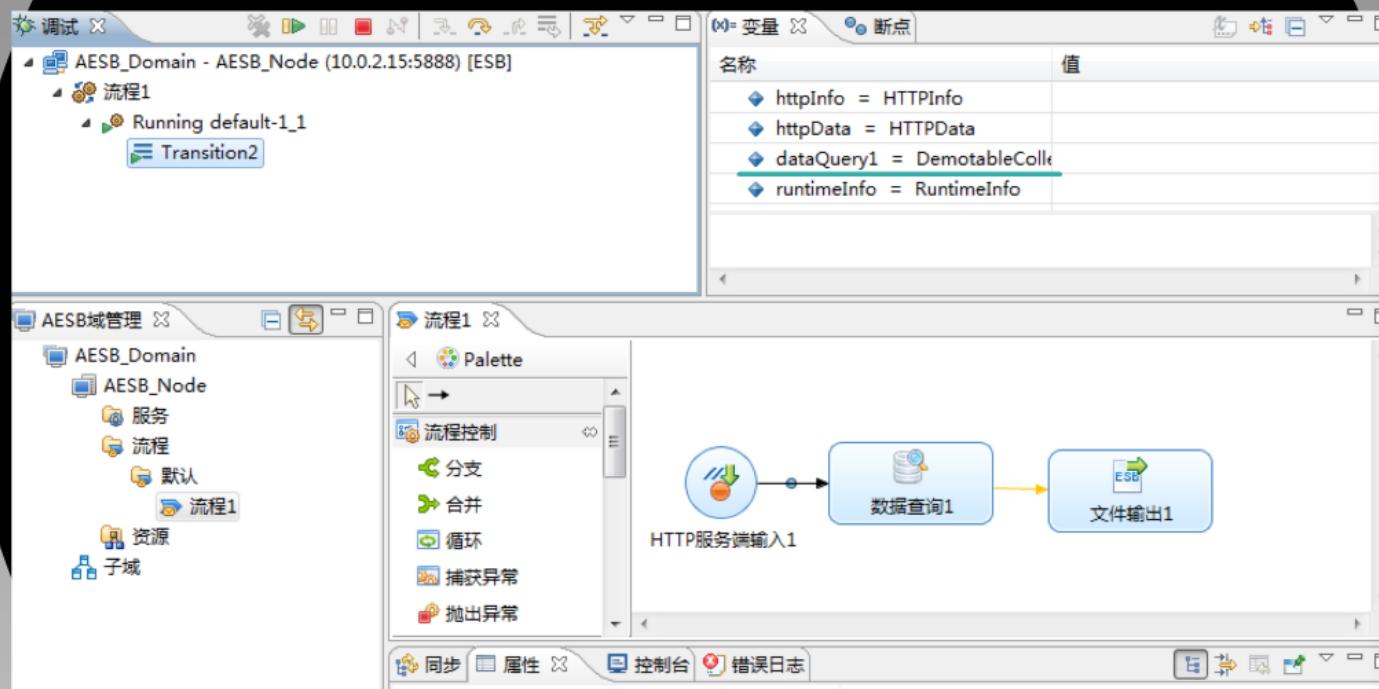
Add Breakpoint

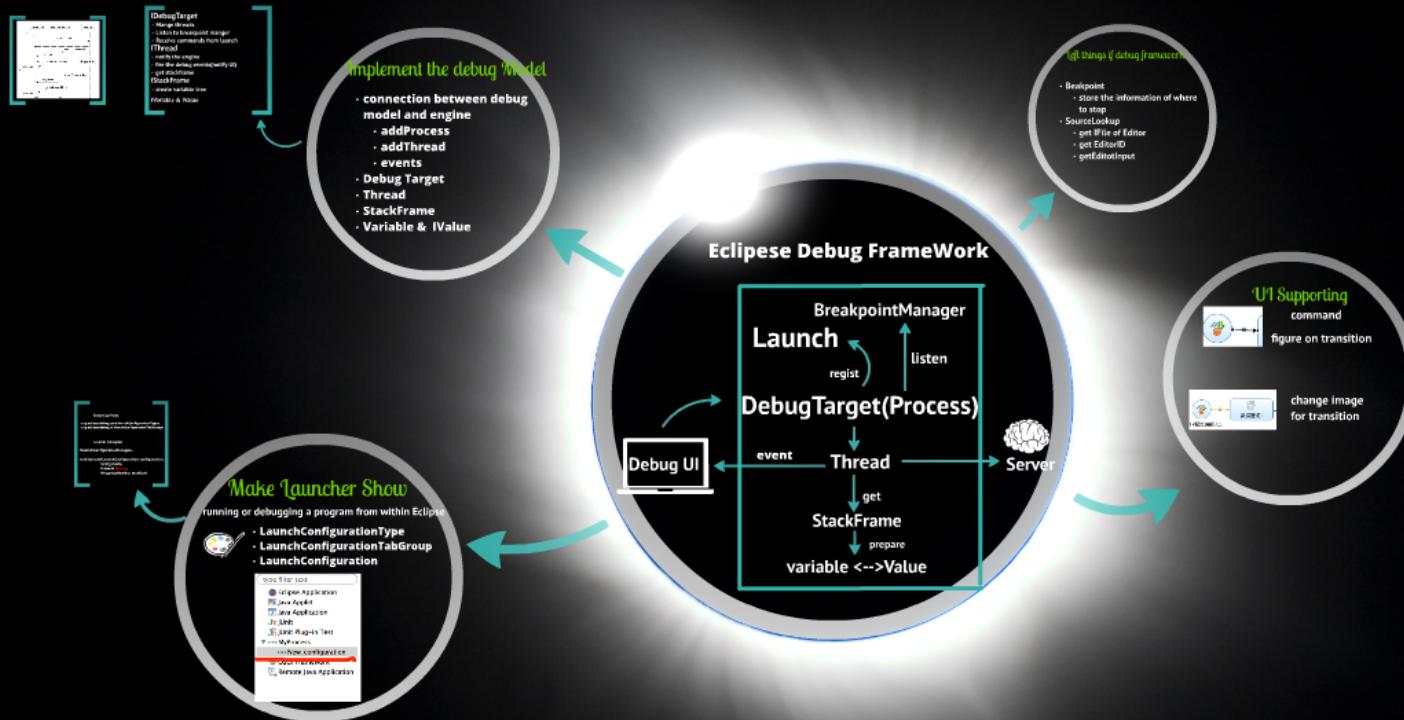


Stop At Breakpoint



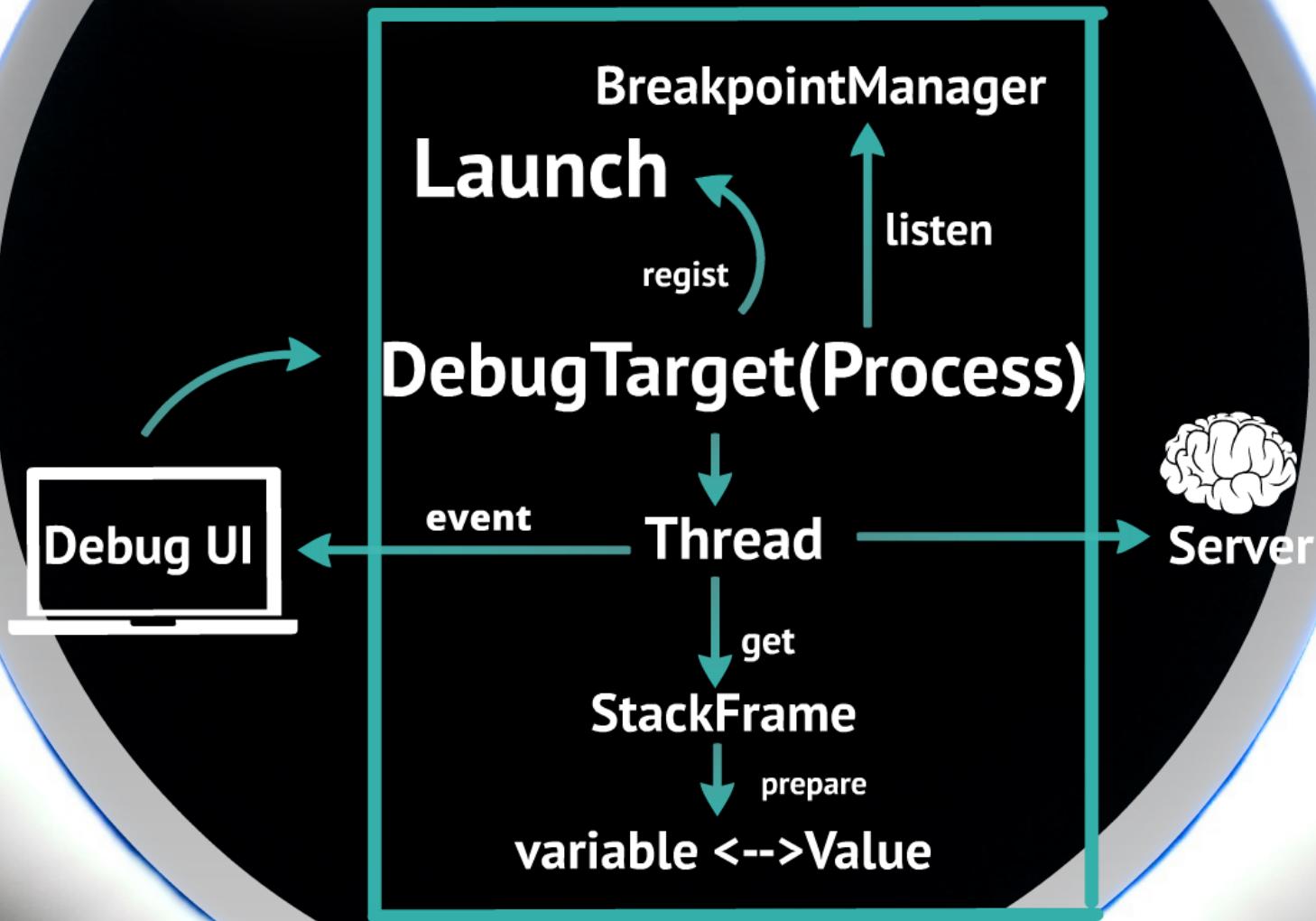
Next Step





Implementing debug functionality for process editor

Eclipsese Debug FrameWork



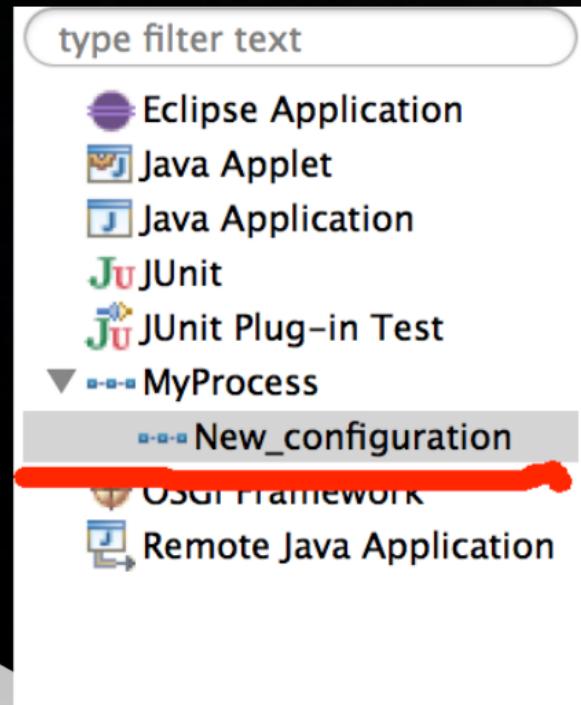
gate
uration configuration,
,
tor monitor)

Make Launcher Show

running or debugging a program from within Eclipse



- **LaunchConfigurationType**
- **LaunchConfigurationTabGroup**
- **LaunchConfiguration**



ExtentionPoint

- **org.eclipse.debug.core.launchConfigurationTypes**
- **org.eclipse.debug.ui.launchConfigurationTabGroups**

Launch Delegate

ILaunchConfigurationDelegate

```
void launch(ILaunchConfiguration configuration,  
String mode,  
ILaunch launch,  
IProgressMonitor monitor)
```

- **connection between debug model and engine**
 - **addProcess**
 - **addThread**
 - **events**
- **Debug Target**
- **Thread**
- **StackFrame**
- **Variable & IValue**

IDebugTarget

- Mange threads
- Listen to breakpoint manger
- Receive commands from launch

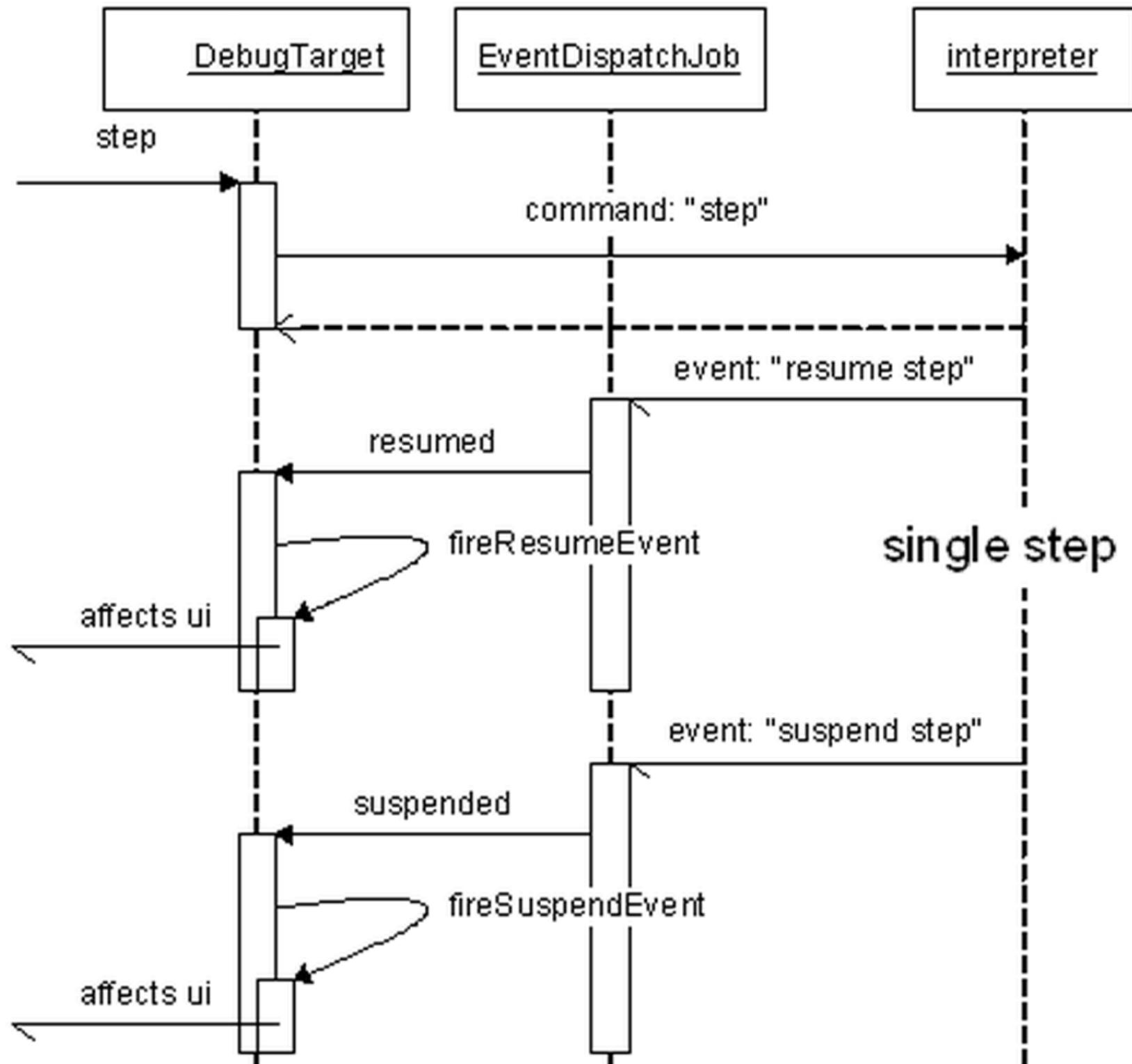
IThread

- notify the engine
- fire the debug events(notify UI)
- get stackframe

IStackFrame

- create variable tree

IVariable & IValue



Left things in debug framework

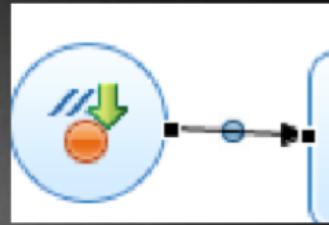
- **Breakpoint**
 - store the information of where to stop
- **SourceLookup**
 - get IFile of Editor
 - get EditorID
 - getEditotInput



UI Supporting

command

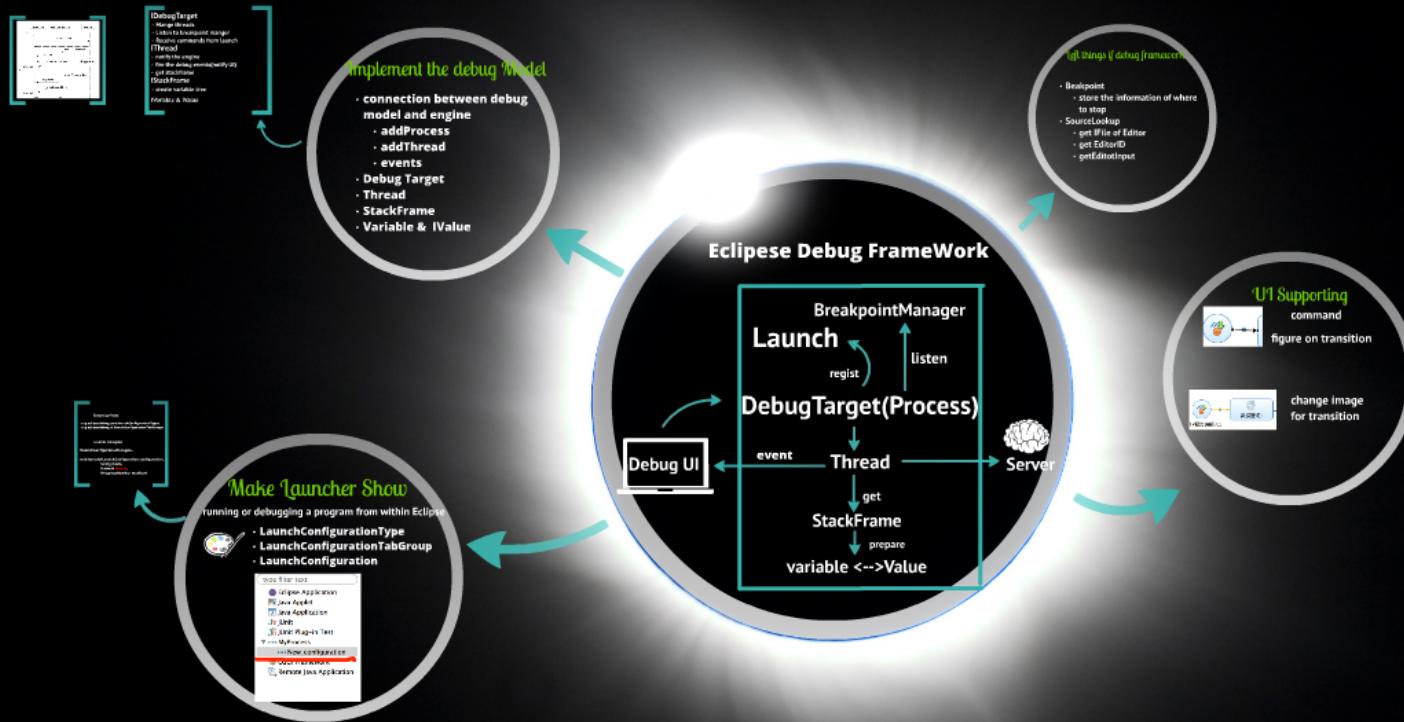
figure on transition



change image
for transition



TP服务端输入1



Implementing debug functionality for process editor