

3.3 Context Launching Enhancements

Author: Darin Wright, Michael Rennie, Wayne Ho - Last modified: Tuesday, May 8, 2007.

Introduction

This document proposes enhancements to the user's application launch experience in Eclipse. The document first reviews the usability concerns with the existing launch experience and then proposes enhancements to address the concerns. This document does not propose to change the way external tools are launched.

To maintain existing behavior (for those that like it), there should be a user preference to use the existing launch behavior (i.e. launch most recent again).

Existing Usability Issues

The current launch framework in Eclipse provides a flexible and powerful way to launch applications within and external to the Eclipse workbench/IDE. The framework supports the following features.

- Allows ISVs to easily contribute launchers that are tightly integrated with the platform.
- Breaks the launcher/resource dependency. Previously, launchers only worked on workspace resources.
- Centralizes the collection of launch-related attributes.
- Allows users to save sets of launching attributes for reuse.

While the framework provides the advantages described above, there are a number of usability issues with user interface. The Launch Configuration dialog and the task flow around the dialog are fairly complex. This is a particular problem for less technical users of Eclipse (e.g., Web Designers, Testers, etc.). While the launch shortcuts (Run, Debug, Profile cascading context-menus) have provided some improvements, many users continue to struggle with learning and using launch configurations. A number of usability issues have been identified:

- Users expect the Run toolbar button to be context-sensitive. Users wish to select the resource to run in the Navigator (or in the Editor) and simply click on the toolbar and have it run.
- When using the context menu and the cascading menu that contains launch shortcuts, users often are compelled to choose the "Run..." option as opposed to the easier launch shortcut. This brings the user to the more advanced Launch Configuration dialog where less technical users tend to struggle.
- The Launch Configuration Dialog is complex: The list of configurations tends to be long and often displays a list that includes irrelevant (to the selected context) configuration types. For example, if a user selects a test within the Navigator and chooses to "Run->Run..." the user will continue to see Java-related configuration types.
- Users do not easily discover the need to create a new configuration. Users often do not discover the "New" button (or the double-click action to create a new configuration) as they do not realize it is a necessary step. They simply wish to run something not create something (in this case a configuration).

Proposed Enhancements

This document proposes the following enhancements. The goal is to streamline and simplify the basic run scenario for the less technical and novice users while maintaining power and flexibility for advanced users. In all of the scenarios, the toolbar launch buttons (run, debug, profile, etc.), become context sensitive.

Displaying launch configurations as resource properties ("Run/Debug Settings")

A property page (named "Run/Debug Settings") will be used to display launch configurations associated with a resource based on launch configuration resource mappings (introduced as API in 3.2). For example, when the "Run/Debug Settings" for a project are displayed, all configurations mapped to a resource within that project would be displayed. When settings for a file are displayed, only configurations mapped to that file would be displayed. This provides view of launch configurations scoped to the selected resource.

Users will be able to create, edit, delete, and copy launch configurations from the "Run/Debug Settings" property page. When a users chooses to create a new configuration they will be prompted to select the type of launch configuration they want to create. The choices will be limited to relevant configuration types. In 3.2, there was no API to determine relevant configuration types for a given resource. To support this feature a new

attribute will be added to the launch shortcut extension allowing a shortcut to specify the types of configurations it corresponds to.

All edits made on the property page will support the cancel operation available on the property page.

Not all configuration types maintain resource mappings. Configurations that do not support or maintain resource mappings will only appear for applicable resources. If a launch shortcut specifies that the Eclipse Workbench launch configuration type is applicable to a resource, then Eclipse Workbench launch configuration instances will appear in the property page for that resource.

Context aided filtering

To further aid novice users with the selection of applicable launch types we could implement another resource mapping. In this case though we will map launch shortcuts to the applicable type(s) that they can launch. This mapping then can be leveraged:

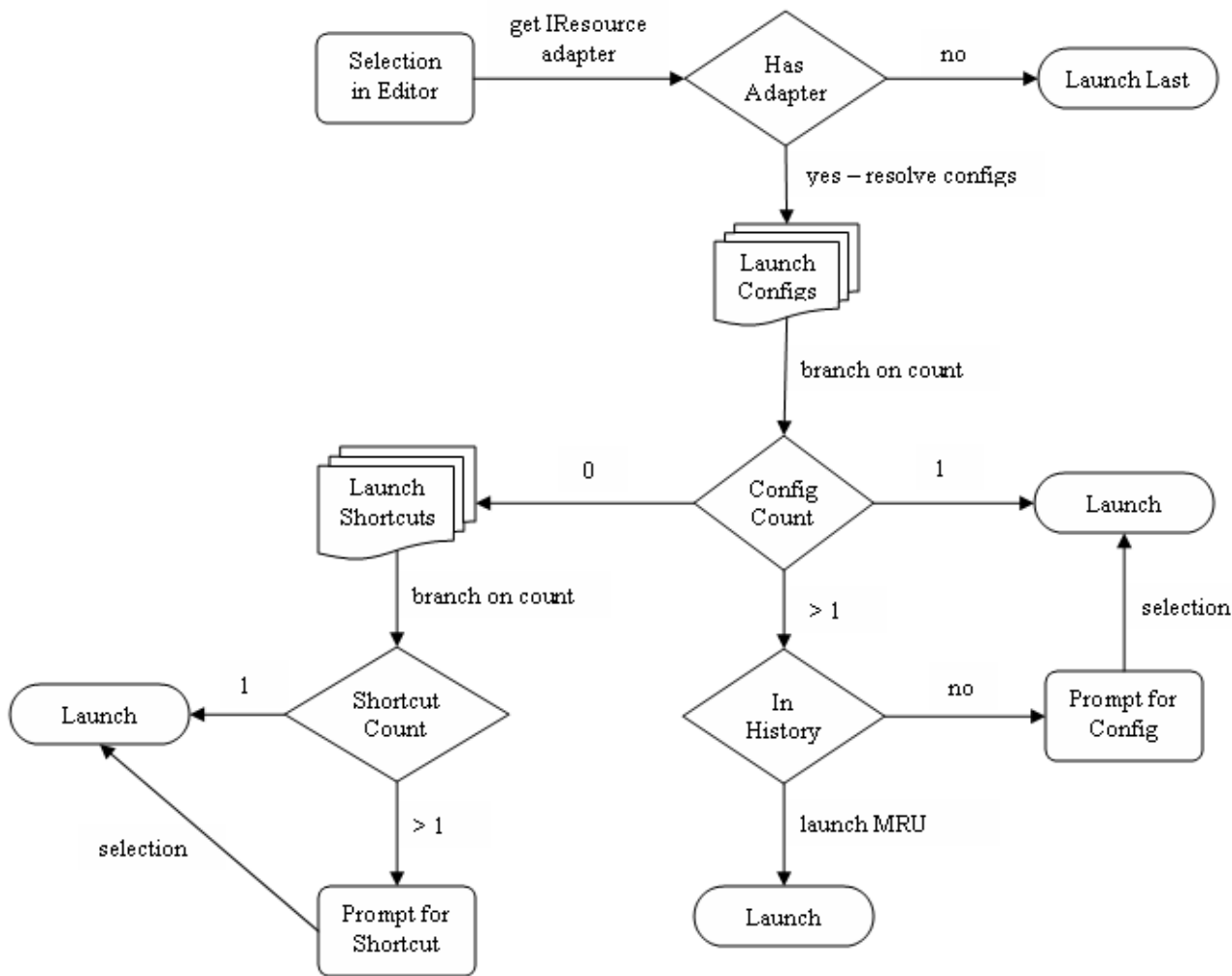
- In the launch dialog we can filter the launch types able to launch the currently selected context, which would be useful in the terminal context launching story when the dialog must be presented to the user.
- Launch histories can take advantage of this mapping to further filter in the case when a history item must be selected to launch but a matching one is not immediately found
- We can use this default mapping in project creation wizards as a template and allow it to be customized
- We can build on top of the default mapping with customized selections from the user from property and/or preference pages

In terms of usability we would require the ability to disable this feature via a preference page or the like.

Implementation of Context Launching

The following steps describe in detail how the platform will determine what to launch when the run (debug, etc.) button is pressed. The approach attempts to resolve an existing launch configuration with the workbench selection or active part. Failing that, the platform resolves default launch settings associated with the selection (i.e. default way to launch something), and failing that the platform determines what launch shortcuts are applicable to the selection. When all else fails the user is notified that there is nothing associated with their current context (selection, part, etc.), to launch. They could be given the choice to launch the last configuration at this point, or open the launch dialog. See Figure 1 for a corresponding state chart.

Context Launching simplified.



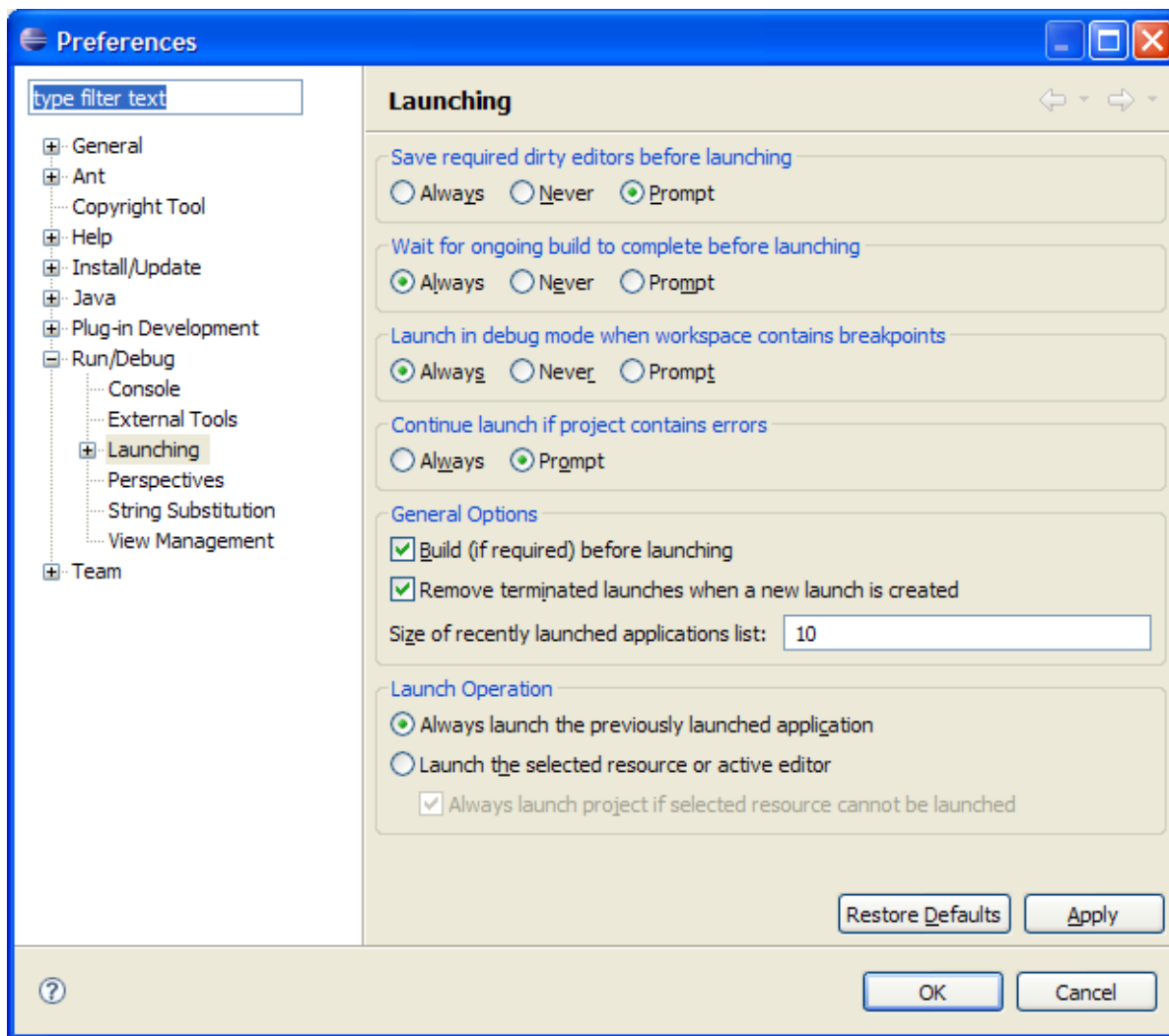
1. Resolve if the context is a shared launch configuration, or shared launch configuration editor input, if it is launch **[done]**
2. Ask the SelectedResourceManager for a corresponding IResource for the current context.
3. If a resource is resolved, attempt to resolve a launch configuration for the resource as follows (else, proceed to the next step)
 1. Resolve the launch configurations for the IResource, if there is only one launch configuration, launch it **[done]**
 2. Else if there are no launch configurations resolve the applicable launch shortcuts for the IResource
 1. If there is one shortcut, use it to launch **[done]**
 2. Else if there is greater than one shortcut prompt the user for the one to select and use it to launch **[done]**
 3. Else there are no launch shortcuts available, prompt the user to check the project and start over from sub-step 1 above, OR this is the project and there is no way to launch it, prompt the user that the current context cannot be launched **[done]**
 3. Else there are more than one applicable launch configuration, so we check the launch history
 1. If there is no entry in the launch history for one of the applicable launch configurations, prompt the user to select one and launch it **[done]**
 2. Else we have found a most recently launched configuration, launch it **[done]**
4. Else if there is a last launched configuration run it **[done]**
5. Else if there is no last launched configuration tell the user that the selected resource cannot be launched **[done]**

In the dialog used to prompt users in the event there is no way to launch the selected resource, there should be an option for users to have their choice remembered

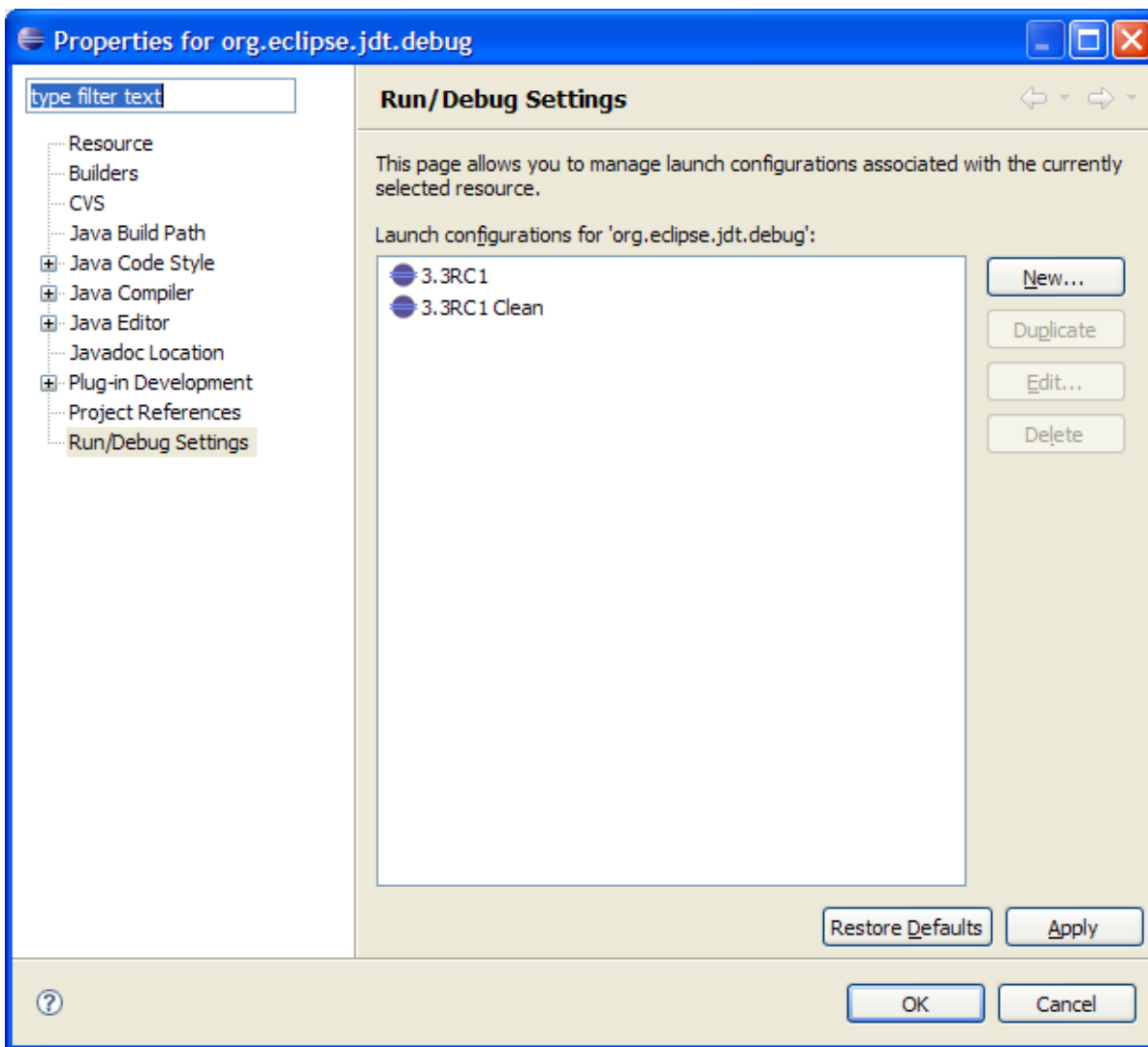
- OPTION: Make this the default way to launch things in [project]

Screen Shots

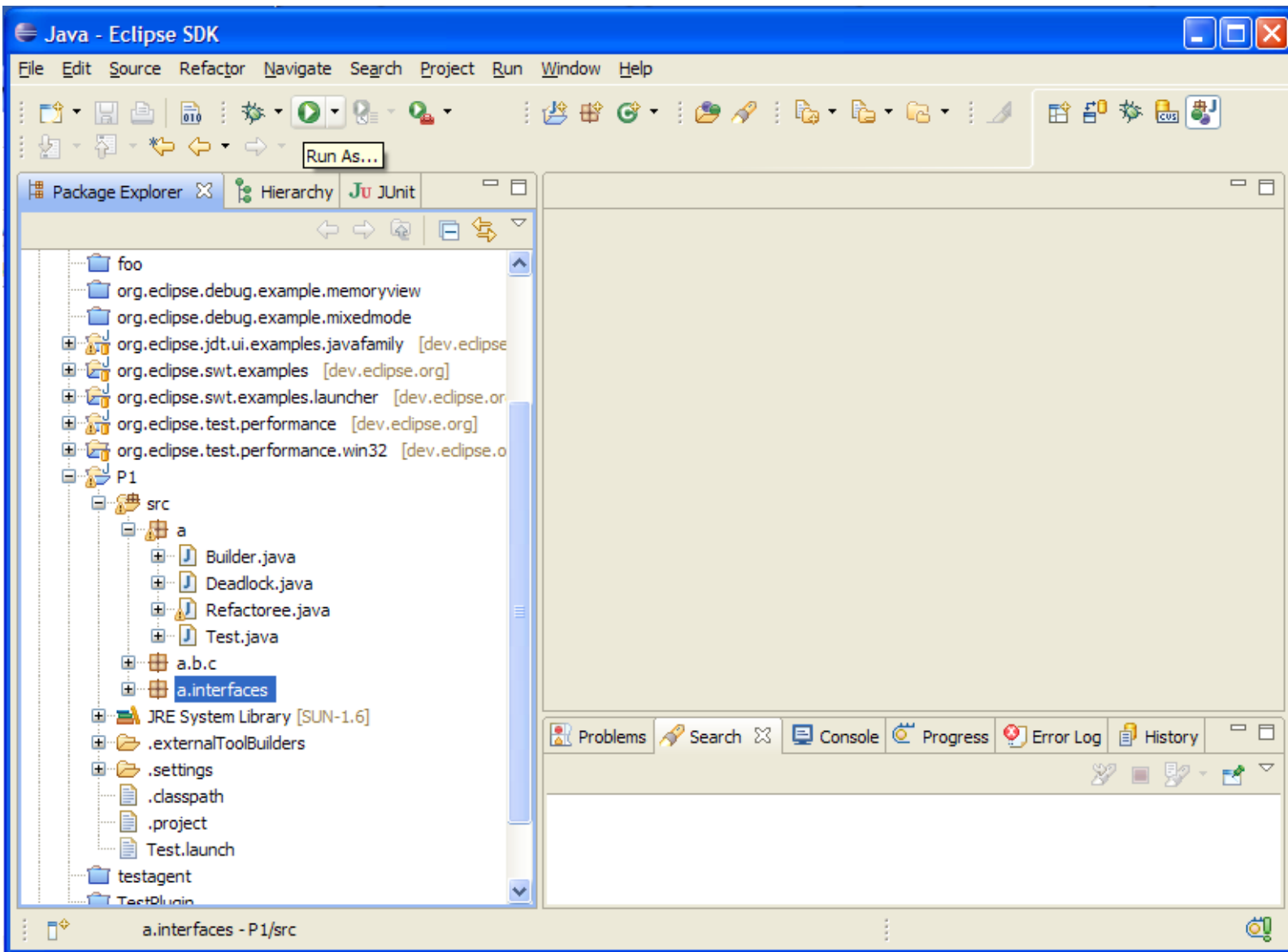
The Launching preference page, displaying the choices for launching style at the bottom of the page.



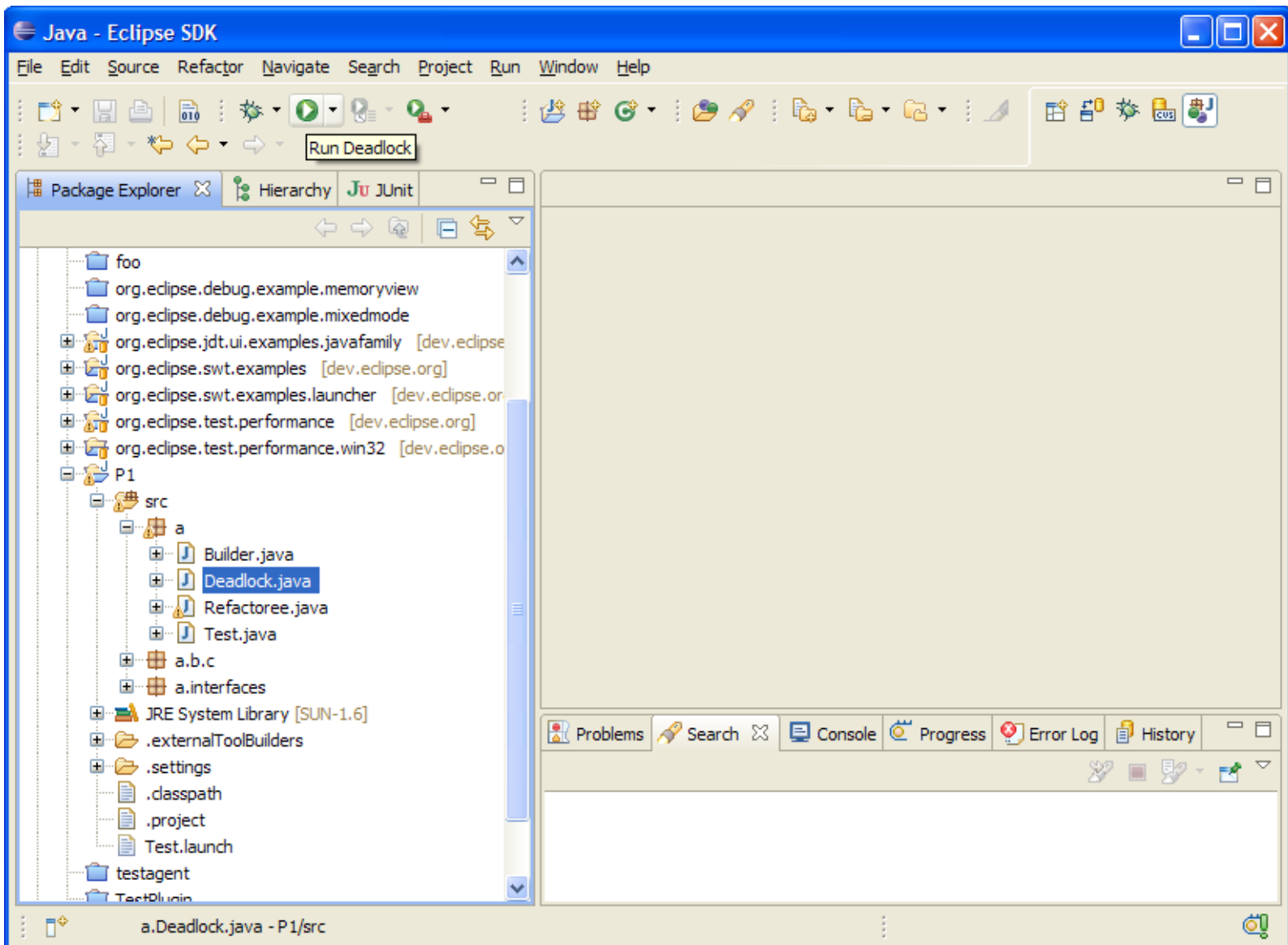
The Run/Debug properties page, which displays all of the launch configurations that apply to the currently selected IResource. Users can also add/duplicate/delete/edit configurations on this page.



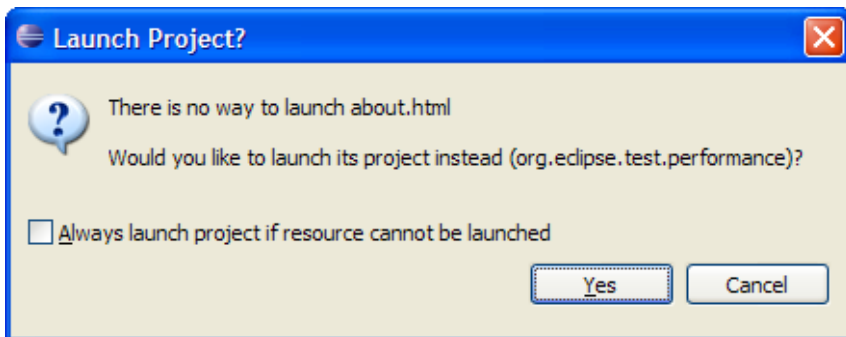
Adaptive tooltip based on the currently selected IResource. The following case demonstrates if no way to launch the selected resource has been found (uncertain launch context).



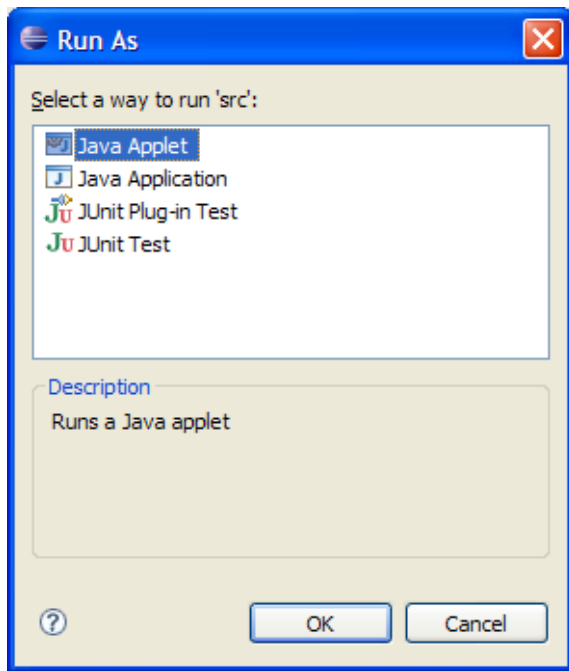
Adaptive tooltip based on the currently selected IResource. The following case demonstrates that we will launch the selected resource Main.java (launch context has been determined).



Prompt dialog used if the currently selected IResource is not launchable.



The 'way to launch' selection dialog, which is used if we have more than one 'way' to launch a context, and there are no existing launch configurations for the current context.



The select launch configuration dialog allows users to select a specific launch configuration to launch. This dialog is presented to the user in the event that more than one launch configuration for the current context exists, but none of the applicable configurations appear in the launch history.

