

# Building perspectives, views and plug-ins



## Agenda

- Creating your own perspective
- Creating a custom view
- Creating a simple plug-in

# Creating your own perspective



 eclipse

© 2004 IBM Corporation.

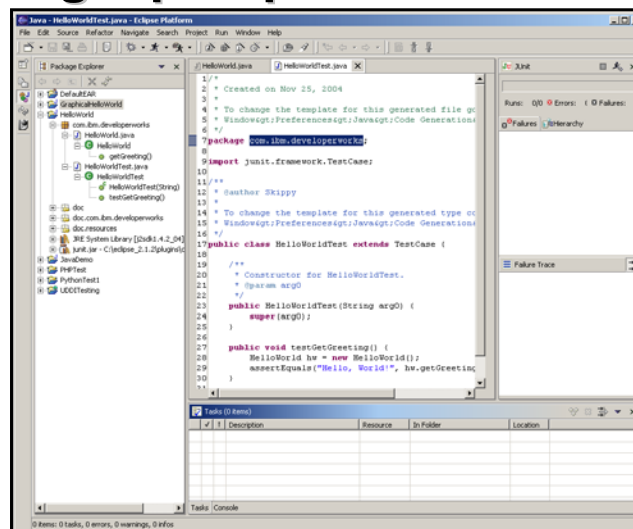
## Creating a perspective

- Creating a perspective is relatively easy; you arrange a group of views the way you want them, then save your workspace as a perspective.

## Creating a perspective

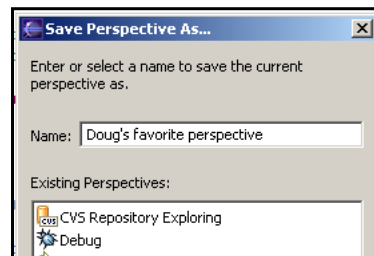
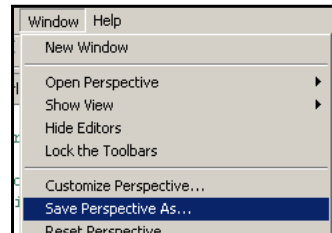
- For my Java development work, I've decided that I like to use the following views:
  - Package Explorer
  - Java editor
  - JUnit
  - Tasks and Console stacked together

## Creating a perspective

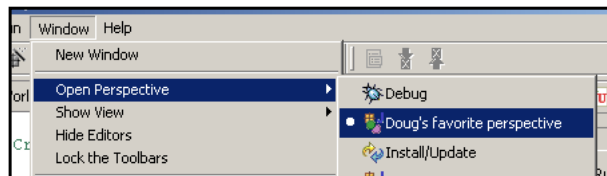


## Creating a perspective

- To save the perspective, select **Window→Save Perspective As...** and give your new perspective a name.



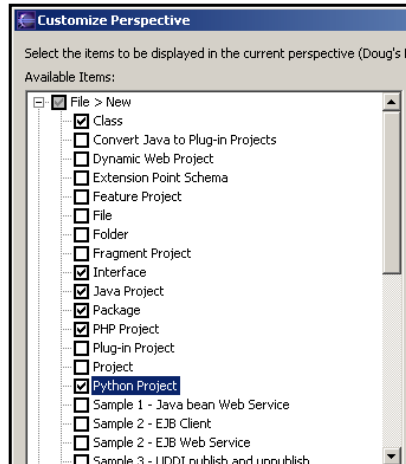
## Creating a perspective



- Now your perspective appears along with the other installed perspectives.

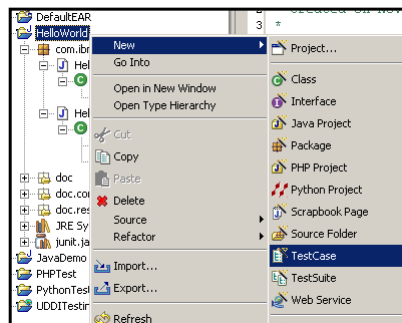
## Customizing your perspective

- Select **Window** → **Customize Perspective...** to change your perspective's settings.



## Customizing your perspective

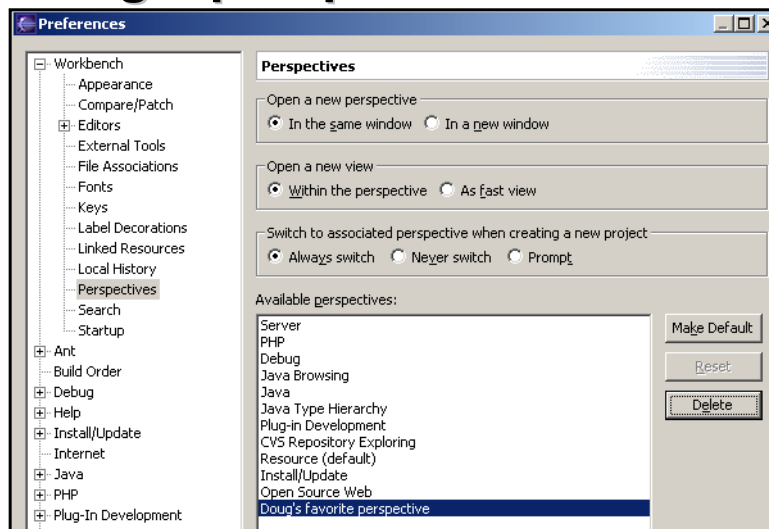
- You can change which items appear on menus when you customize your perspective.



## Deleting a perspective

- You can use the Preferences dialog to delete your perspective whenever you want.
  - You can't delete one of the pre-installed perspectives.

## Deleting a perspective



# Creating a view



eclipse

© 2004 IBM Corporation.

## Creating a view

- Creating a view is more complicated.  
You need to decide what your view will display, then figure out how to draw it in the view window.

## MVC

- Any view you create should follow the model-view-controller paradigm.
  - The model of your resources (whatever they are) should not control the view.
  - The view should not modify your resources.
- By using the controller to keep the view and the model in sync, you can change one and the other reflects those changes.

## Creating a plug-in





## Creating a plug-in

- As you'd expect, this is the most difficult task of the three we're discussing here.
- There are a number of interfaces you need to learn if you're going to build a commercial-quality plug-in.
  - To keep things simple, we'll use the wizard facility here.

## Review



## Review

- We've covered the basics of plug-in development with Eclipse.
- There's lots more we could cover (maybe next time...)