

Eclipse EJB 3.0 Object/Relational Mapping Project Requirements

Features and use cases for early project milestones

Revision History

Date	Description	Author(s)	Comments
June 1, 2005	First Draft	Neil Hauge	Covers just the features relating to the creation of a basic Entity.

Note: See project proposal for high-level project use cases. This document describes features and use cases for early project milestones. It is meant to describe current work on the project to solicit feedback from the community. Content is subject to change.

Feedback

Comments on this document should be posted to the Eclipse EJB ORM project newsgroup (<news://news.eclipse.org/eclipse.technology.ejb-orm>).

1. Features

- Support for defining EJB3 entities through annotation and/or the XML deployment descriptor
 - Works with user or tool defined annotations, being as helpful or unhelpful as the user wishes
- Content-assistance for annotation completion
 - Local Attribute and Entity name values for mapping configuration
 - Table/Field values from the Eclipse based RDB tools (WST -> DTP)
- EJB3 Persistence Outline for a persistence based view on a class
 - Displays default values for undeclared settings
 - Aware of EJB3 persistence annotations and EJB3 XML Deployment Descriptor (provides embedded merged view to determine overrides)
- EJB3 Persistence Properties View for configuring persistence settings on a class
 - Rich UI for configuring class persistence
 - Context based choices for mapping selections (drop downs and list choosers vs text entry)
 - Based on cursor position in the Java file or Persistence Outline View selection
 - Works for annotation based settings and XML deployment descriptor settings
- EJB3 Persistence Outline for a persistence based view on the XML based EJB 3.0 Deployment Descriptor
 - Allows user to work in the XML file editor, with the Persistence Outline and Persistence Properties View provided for assistance
- EJB3 Persistence Perspective for working with EJB 3.0 Entities (See **Figure 1**)
 - Houses Persistence Outline and Persistence Properties views + DTP Explorer view when available
- Entity validation through Problems reporting
- JST integration for EJB packaging and application deployment

2. Use Cases

- Define an Entity and its properties
 - For New Class: New Entity Wizard extends the New Class Wizard
 - “New Entity” action available when the EJB3.0 Persistence plug-in is loaded (launches wizard – See **Figure 2,3**)
 - Provides class-level configuration of entity (Entity Name, Table Name, Access Type, etc) (See **Figure 4**)
 - Upon completion, should ask the user if they wish to switch to "Persistence Perspective" if not already in this mode
 - For existing class: Right-click on an existing Java Class in the Package Explorer and select “Make Entity” action to mark a basic Class as an Entity. This would be done through annotation insertion or XML descriptor creation (preference based). (See **Figure 5**)
 - Class is marked as an Entity
 - Optional Entity Wizard to walk through mapping the class
 - The affected java source file should be opened as a result if not already open
 - Should ask the user if they wish to switch to "Persistence Perspective" if not already in this mode

3. Screen Shots

Note: Perspective, Outline View and Property View names are likely to change to be more specific

EJB 3.0 Persistence Perspective – (Default View)

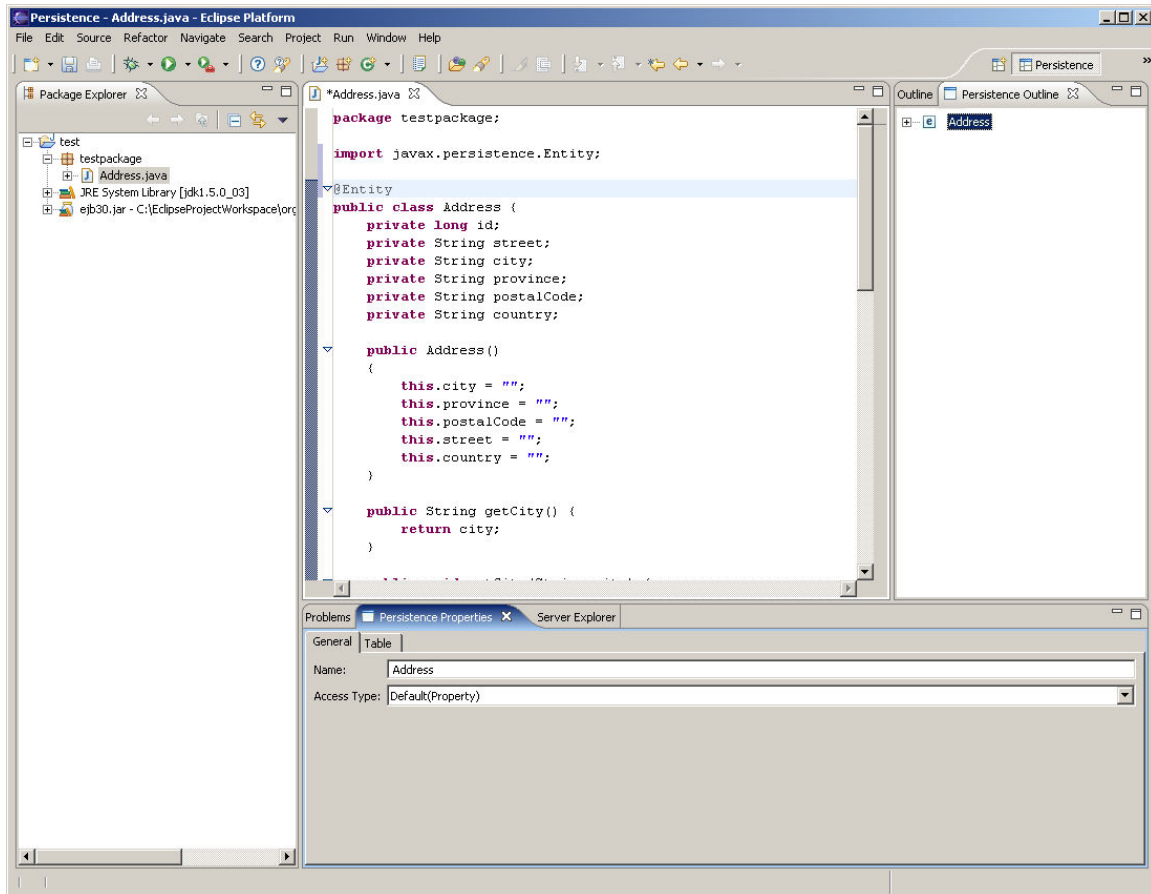


Figure 1

- An example of what the default EJB3 Persistence Perspective would look like
 - Includes the “Persistence Outline” and the “Persistence Properties” view
 - Would also include a “Problems” view and a “DTP Explorer” view when available

New Entity Wizard – Page 1

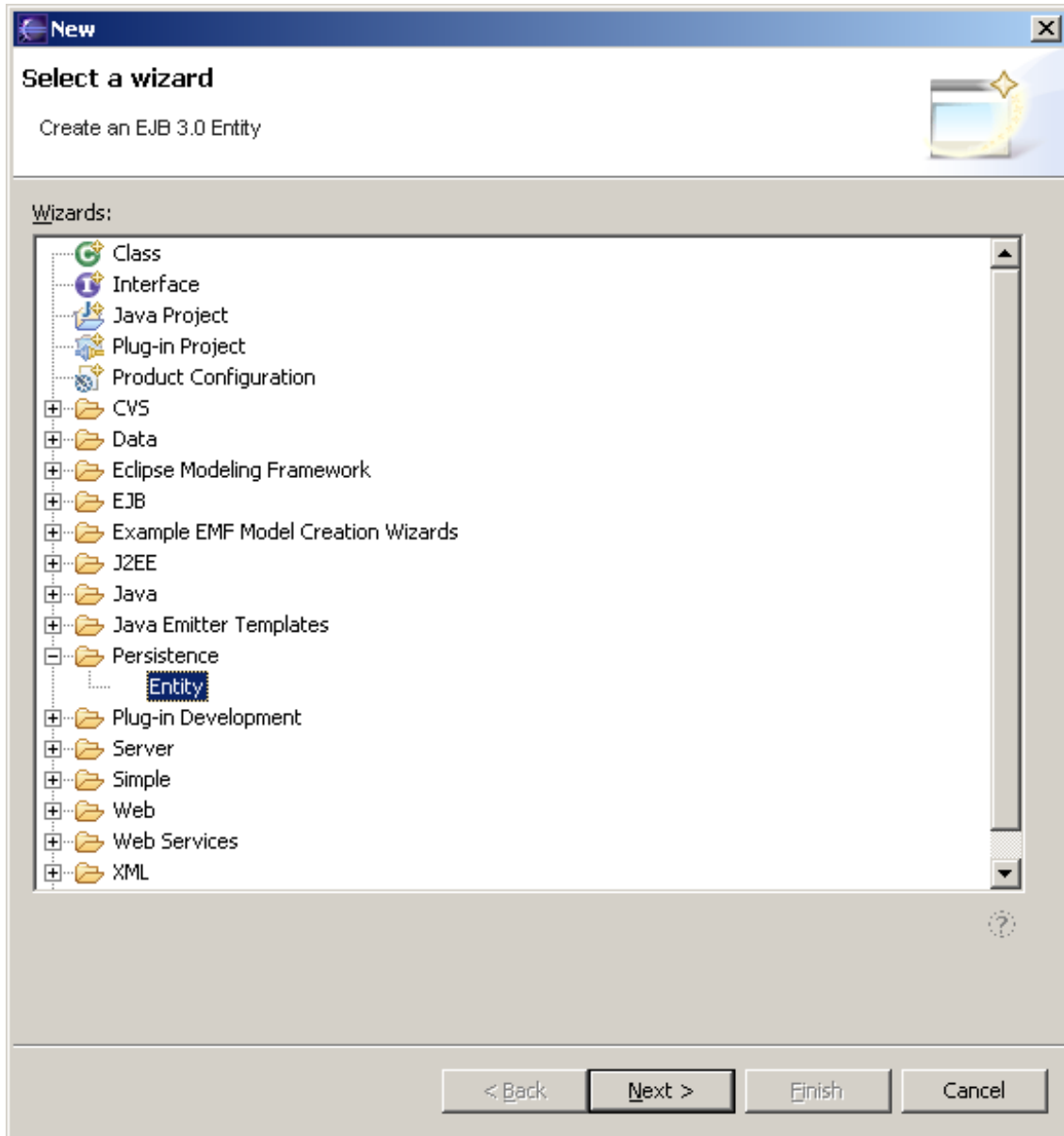


Figure 2

- Select "Entity" in the "New" Wizard to begin defining a base EJB3 Entity

New Entity Wizard – Page 2

The screenshot shows the 'Create New EJB 3.0 Entity' wizard, specifically the 'Java Class' step. The window title is 'Create New EJB 3.0 Entity'. The main heading is 'Java Class' with the instruction 'Create a new Java class.' Below this, there are several input fields and buttons:

- Source folder:** A text box containing 'test' and a 'Browse...' button.
- Package:** A text box containing 'test' and a 'Browse...' button.
- Enclosing type:** A checkbox labeled 'Enclosing type:' followed by an empty text box and a 'Browse...' button.
- Name:** A text box containing 'Employee'.
- Modifiers:** Radio buttons for 'public' (selected), 'default', 'private', and 'protected'. Below them are checkboxes for 'abstract', 'final', and 'static'.
- Superclass:** A text box containing 'java.lang.Object' and a 'Browse...' button.
- Interfaces:** An empty text box with 'Add...' and 'Remove' buttons.
- Which method stubs would you like to create?:** A list of checkboxes: 'public static void main(String[] args)' (unchecked), 'Constructors from superclass' (unchecked), and 'Inherited abstract methods' (checked).
- Do you want to add comments as configured in the properties of the current project?:** A checkbox labeled 'Generate comments' (unchecked).

At the bottom of the wizard, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Figure 3

- Create a Java Class as a part of Entity creation

Create New EJB 3.0 Entity

EJB 3.0 Entity

Specify entity information for the Java class.

Name: Default (Employee)

Access Type: Default(Property)

Table

Name:

Catalog:

Schema:

< Back Next > Finish Cancel

Figure 4

- Specify Entity information for the class
- EJB 3.0 defaults are provided as default values for your review during creation
 - Defaults could be typed over or changed with the drop-down to user specified selections
 - User specified selections would actually be entered as annotations or deployment XML entries, as opposed to EJB 3.0 default values (default values will be visible in the Persistence Outline)

Make Entity For Existing Class Action

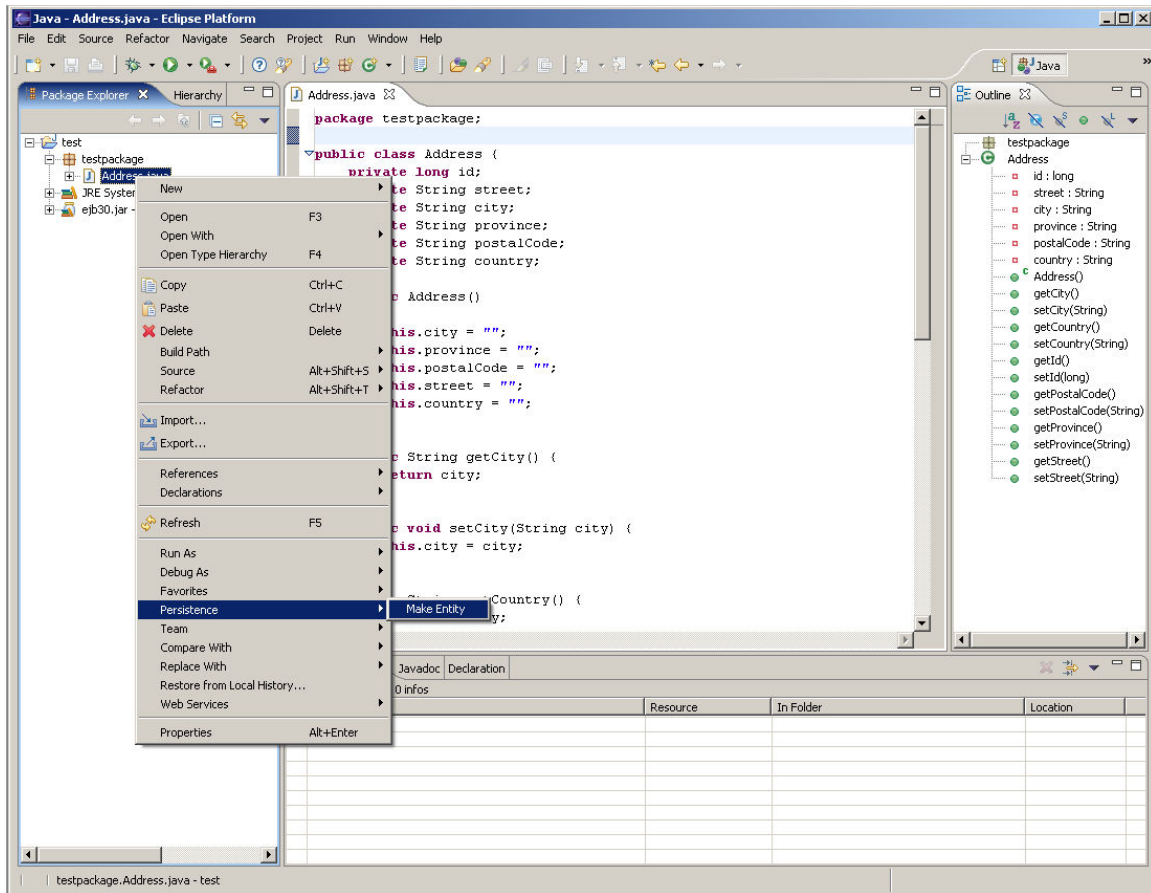


Figure 5

- Make an existing Java Class an Entity by right-clicking on the class and selecting “Persistence -> Make Entity”
- This will annotate the class with `@Entity` or create an entry in the ORM XML descriptor (based on preferences)
- In future iterations, this action would invoke a wizard which would lead you through the configuration of all the base mapping values

Alternate EJB3 Persistence Perspective (User Configured)

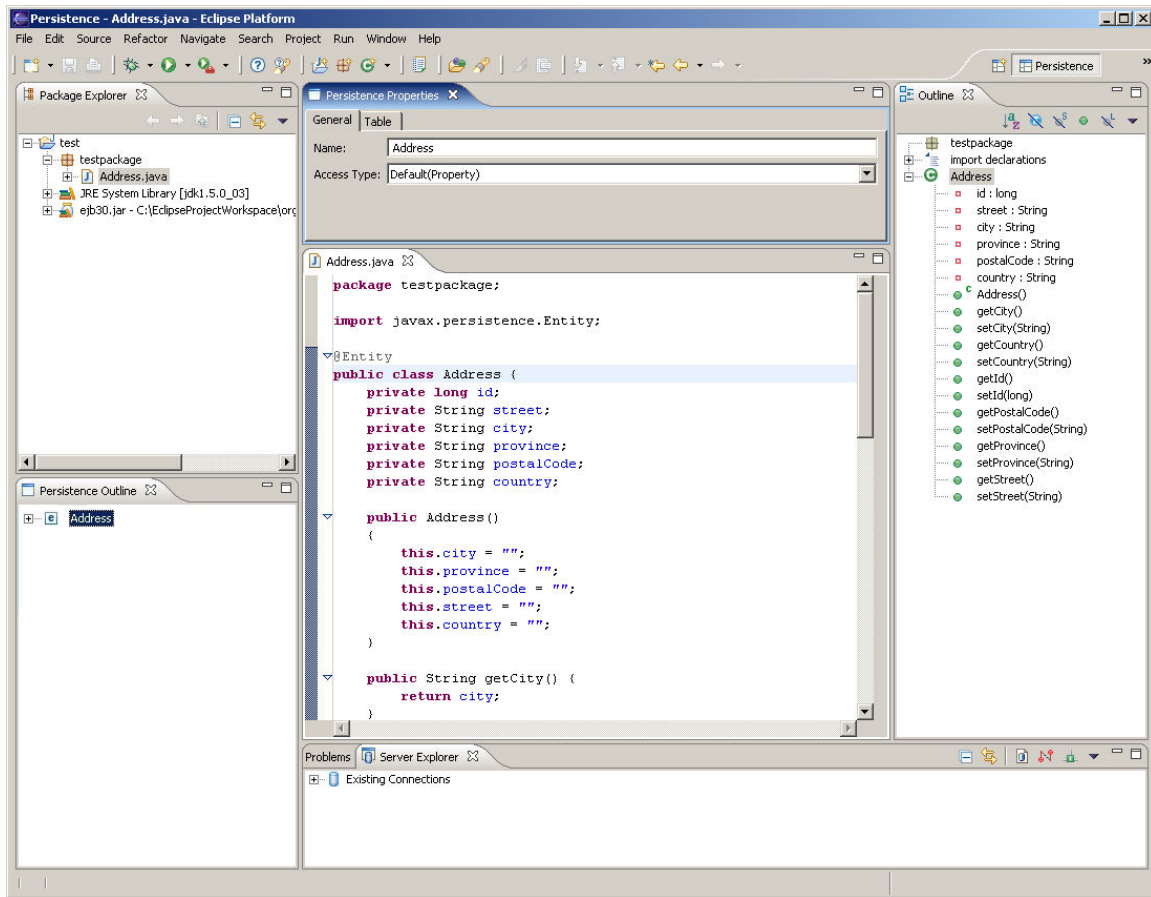


Figure 6

- This is simply a user configured environment showing the EJB3 Persistence Perspective
- The user could arrange the perspective, like any other perspective, based upon their preferred layout