INTEGRATION WITH OPTIMALJ

user's guide

No Magic, Inc.

January, 2007

CONTENTS

INTEGRATION WITH OPTIMALJ

MAGICDRAW UML OPTIMALJ EDITION INSTALLATION

Editions requirements

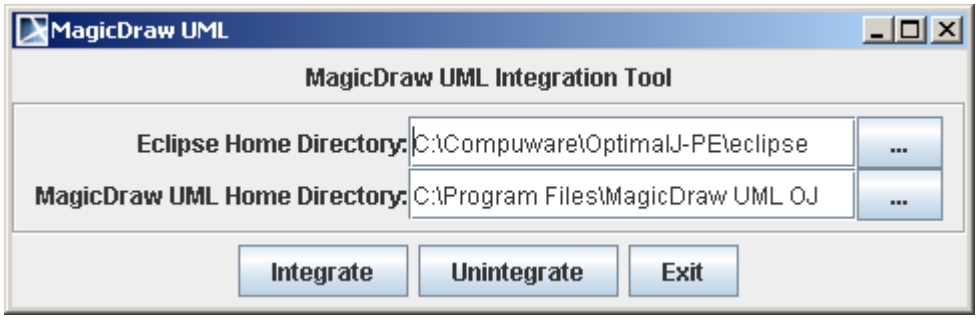
One of these OptimalJ 4.3 editions installed:

- Professional Edition build on Eclipse;
- Architecture Edition build on Eclipse.

MagicDraw and OptimalJ integration

If OptimalJ Professional Edition build on Eclipse is installed, **MagicDraw UML Integration Tool** starts when MagicDraw installation is completed.

Click the **Integrate** button for MagicDraw and OptimalJ integration in the **MagicDraw UML Integration Tool** dialog box.



For more information about MagicDraw and Eclipse integration, see **MagicDraw Integrations UserGuide.pdf**, *Integration with Eclipse/WSAD* section.

NOTE: In Eclipse, MagicDraw by default becomes part of the Java perspective. Also separate MagicDraw perspective is added.

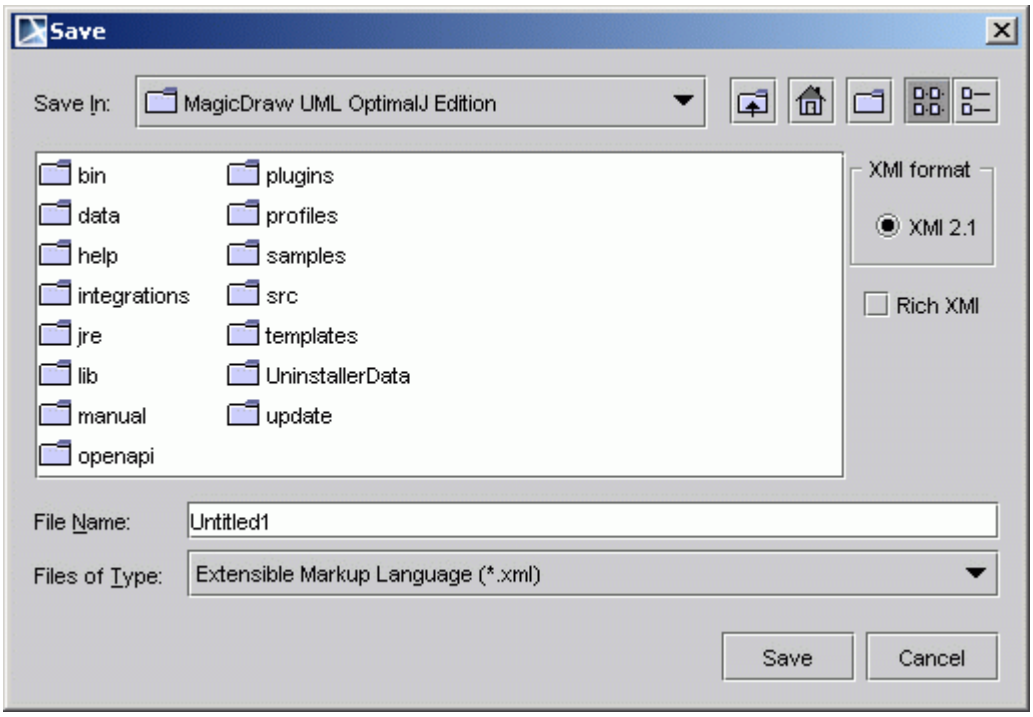
MODEL EXCHANGE BETWEEN OPTIMALJ AND MAGICDRAW

Model (UML model) exchange between MagicDraw and OptimalJ is done through XMI (model saved to XML). XMI is the native MagicDraw project file format.

STEP #1 SAVE THE MAGICDRAW PROJECT

In MagicDraw:

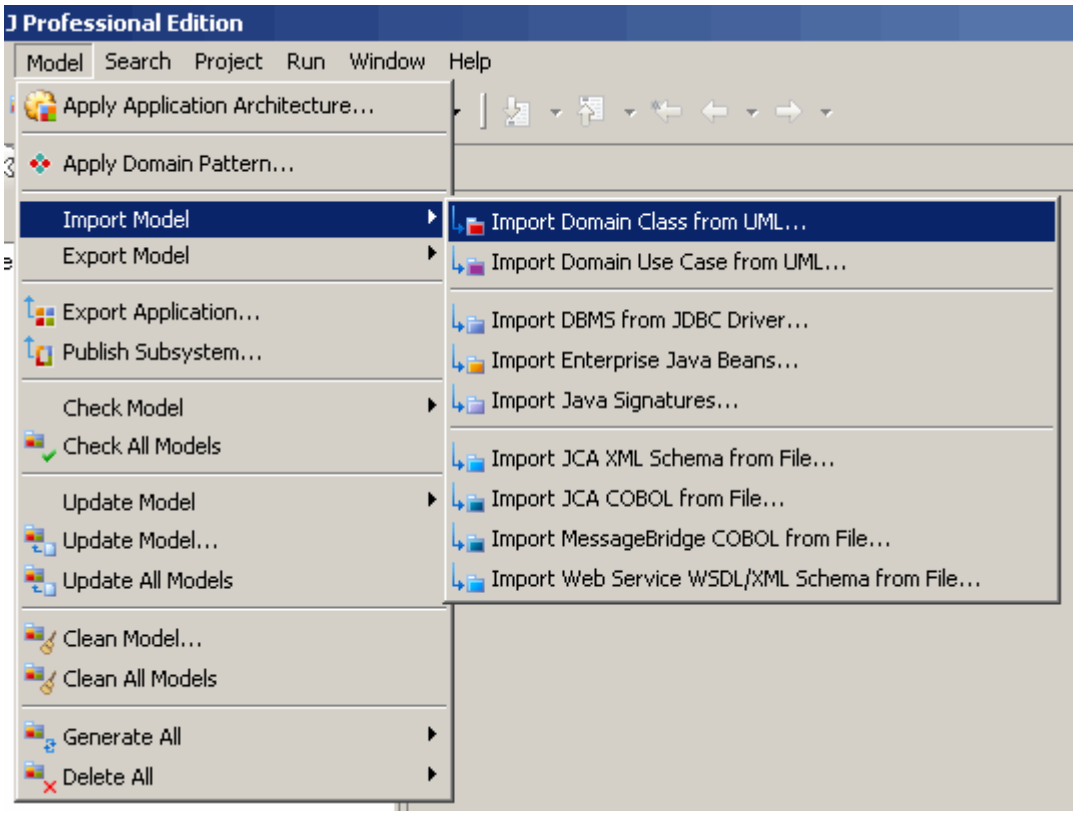
1. From the **File** menu, choose the **Save Project** or **Save Project As** command, or click the **Save** button on the main toolbar, or press shortcut keys CTRL+S. The **Save** dialog box opens.
2. Select the destination directory (where you wish to save the project) and type the file name for it.
3. Choose the **Extensible Markup Language (*.xml)** format for saving.



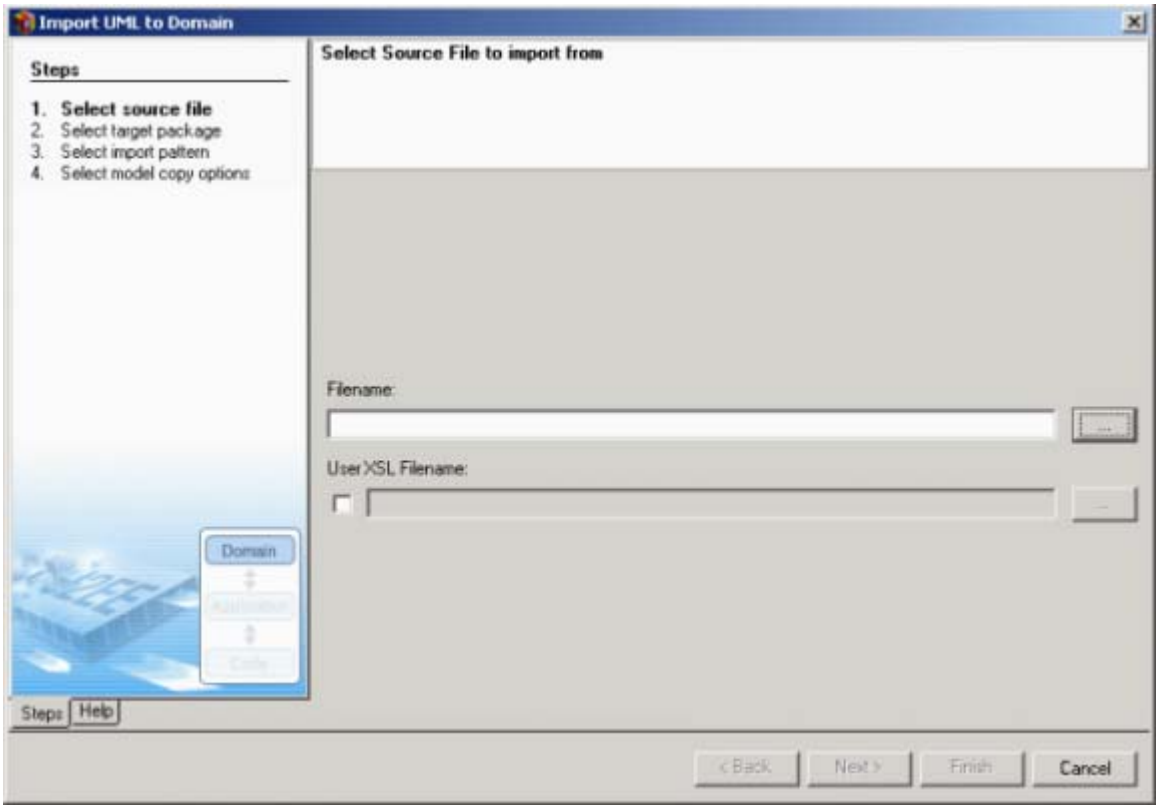
STEP #2 IMPORT MAGICDRAW UML MODEL INTO OPTIMALJ

1. From the **Model** menu, choose **Import Model** and then **Import Domain Class from UML** (or **Import Domain Use Case from UML**.)

NOTE: Using OptimalJ build on Eclipse, switch to Model or Application perspective.



2 Select MagicDraw saved project file and in the **Import UML do Domain** dialog box, follow the import instructions.



STEP #3 GENERATE CODE IN OPTIMALJ

For the details how to process the imported model in OptimalJ, see OptimalJ documentation.

SEQUENCE DIAGRAM CREATION FOR OPTIMALJ GENERATED CODE

MagicDraw UML can create Sequence Diagram for any Java source code. If you did not integrate MagicDraw with OptimalJ, using the MagicDraw UML Integration tool, you can follow the steps, specified in **MagicDraw Tutorials.pdf**, *Java Reverse to Sequence Diagram* section.

Also you may watch **Java Reverse to Sequence Diagram** demo in MagicDraw webpage www.magicdraw.com/viewlets about how to perform Java code reverse from the existing source files to Sequence Diagram.

If you did run the MagicDraw UML integration tool, follow the steps in the Section "Sequence Diagram creation with OptimalJ build on Eclipse" on page 6.

Sequence Diagram creation with OptimalJ build on Eclipse

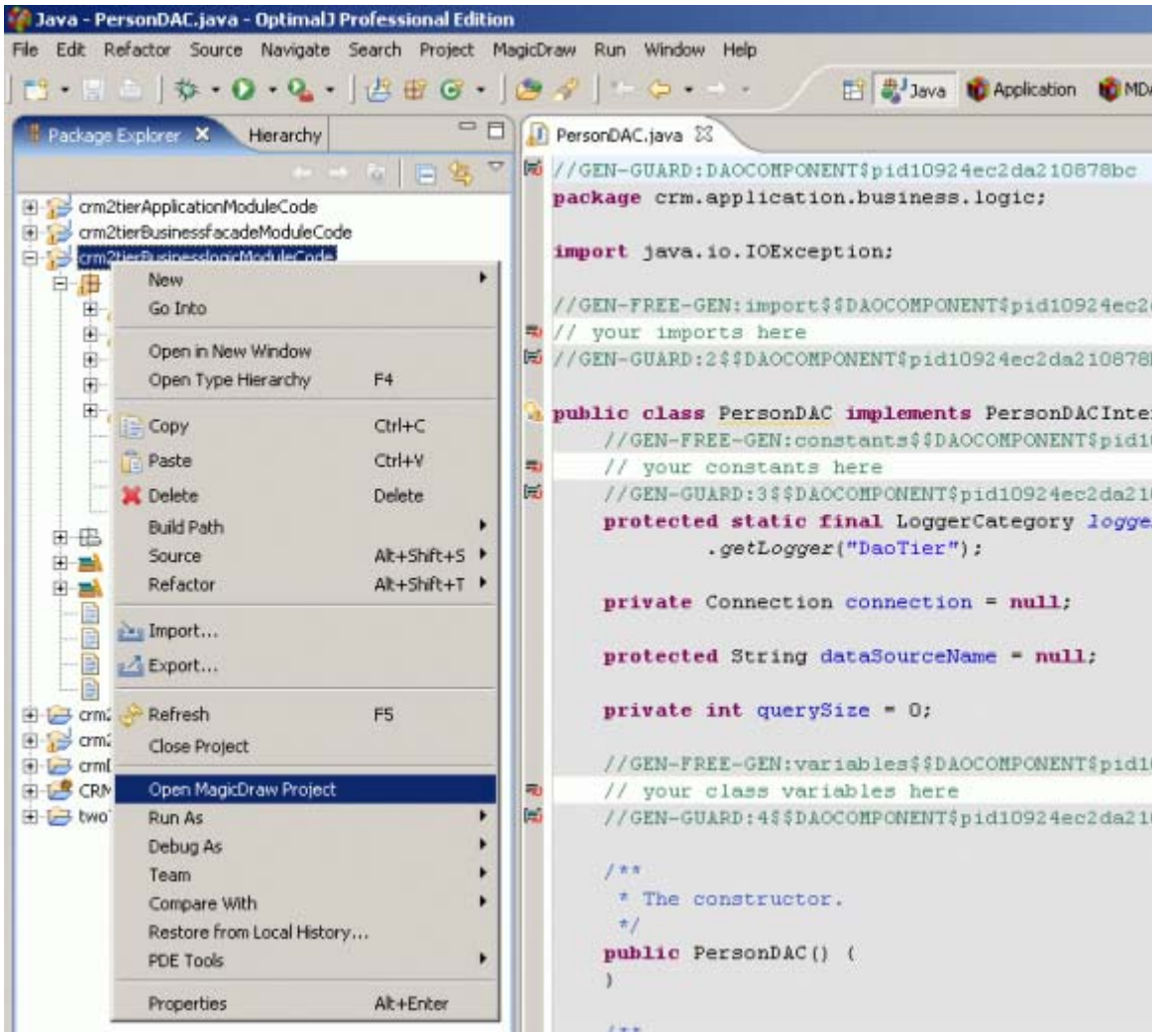
This option is available only if MagicDraw is integrated with OptimalJ (see Section "MagicDraw and OptimalJ integration" on page 3).

If integration was successful, the **MagicDraw** menu appears in the main OptimalJ menu and MagicDraw perspective is added.

NOTE: For more information how to work with MagicDraw and Eclipse integration, see **MagicDraw Integrations UserGuide.pdf**, *Integration with Eclipse/WSAD* section.

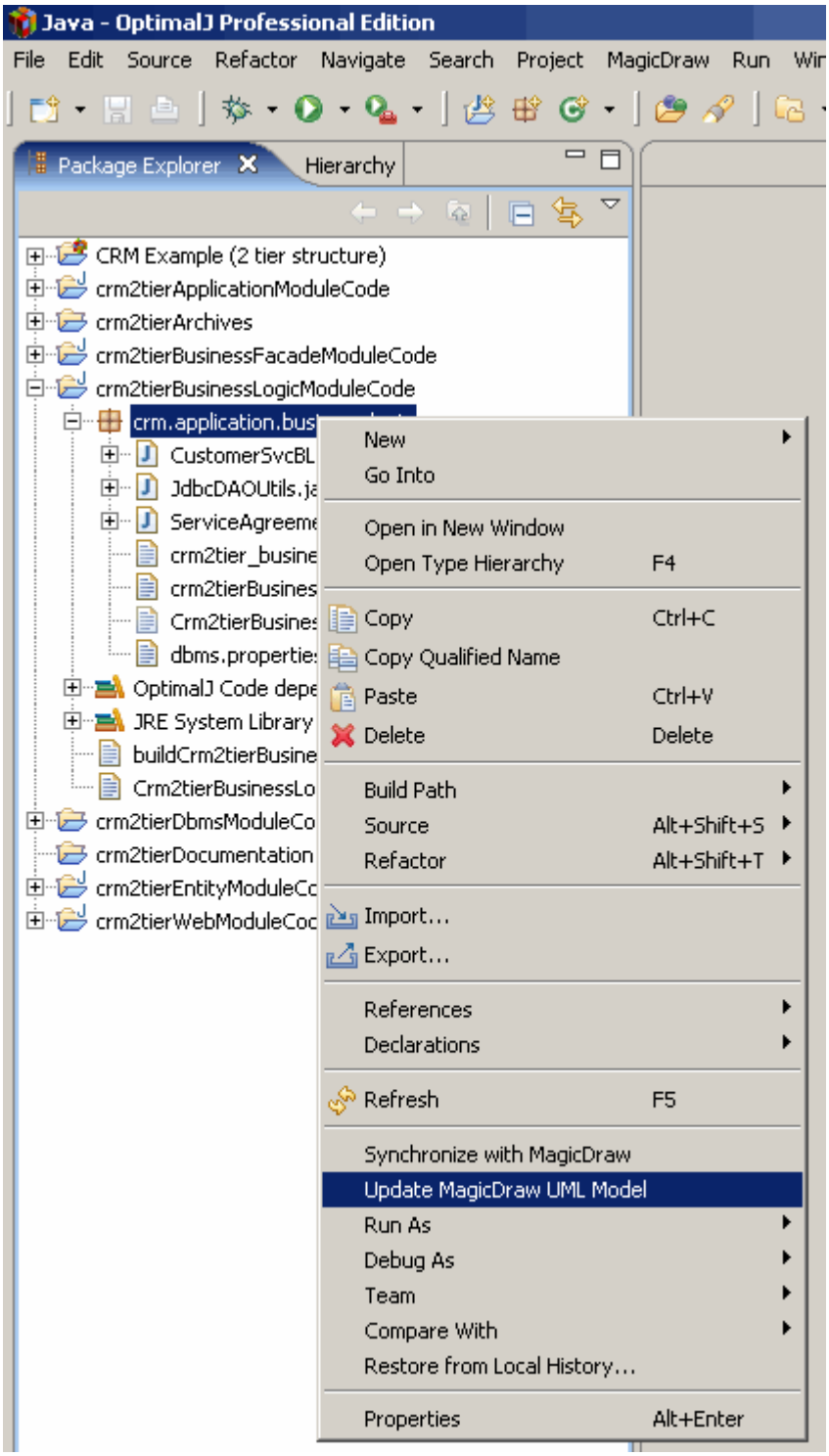
STEP #1 OPEN/CREATE MAGICDRAW UML PROJECT

In the browser (Package Explorer in Java perspective or Code Model browser in Application perspective), select project and from the shortcut menu, choose **Open MagicDraw Project**.



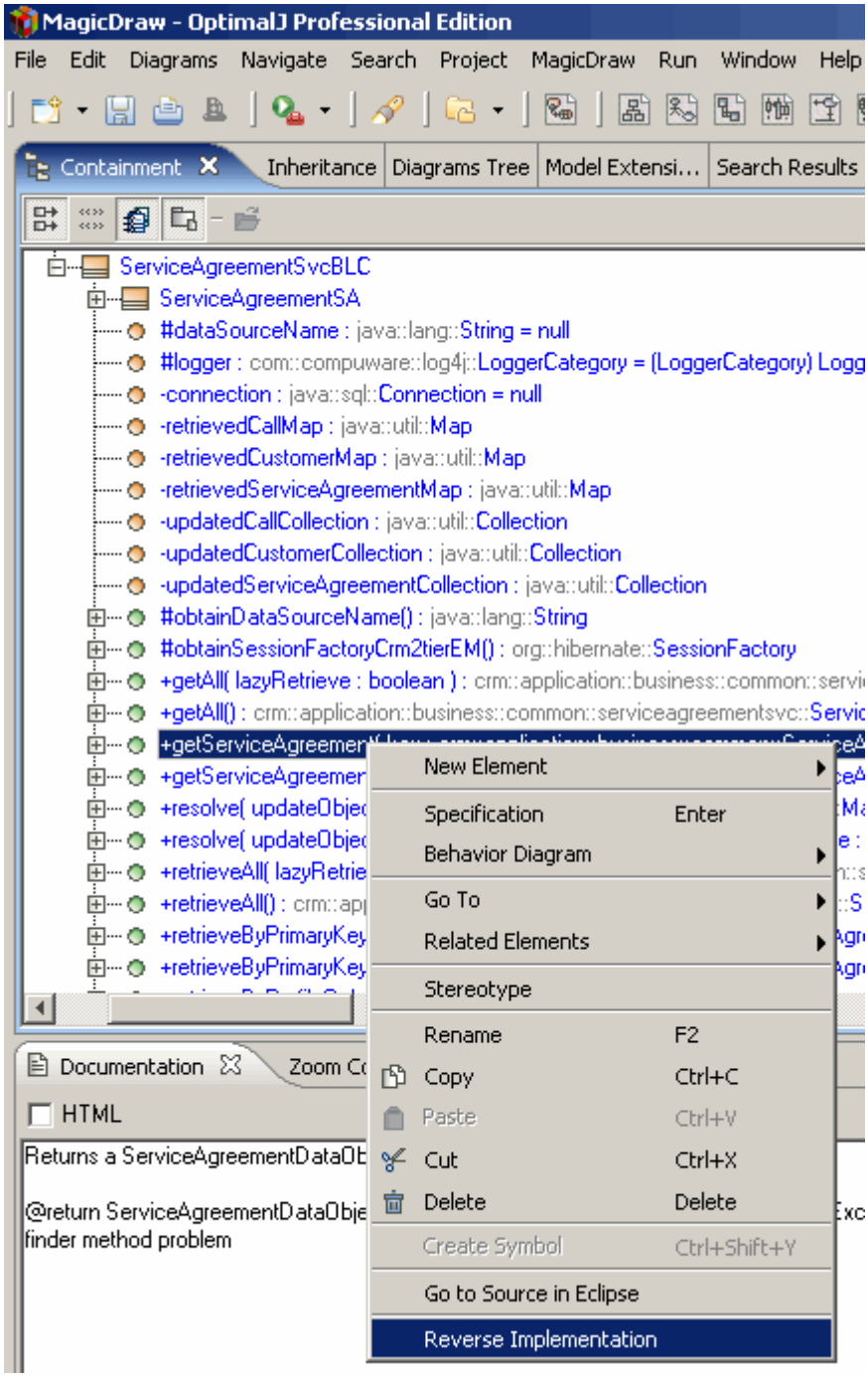
STEP #2 UPDATE MAGICDRAW UML MODEL FROM OPTIMALJ

Select java package or file element in the Browser and from the shortcut menu, choose **Update MagicDraw UML Model**. The UML Model representing the java source code is created/updated in MagicDraw.



STEP #3 CREATE SEQUENCE DIAGRAM IN MAGICDRAW

1. Switch to MagicDraw perspective.
2. In the Containment Tree Browser tab, select desirable operation and from the shortcut menu, choose **Reverse Implementation**.



3. The **Sequence Diagram from Java Source Wizard** dialog box is opened. For more information about this wizard, see **MagicDraw UserGuide.pdf**, the *Sequence Diagram from Java Source Wizard* section, page 10-465.