# Eclipse IDE

## Papyrus

**URL:** <https://marketplace.eclipse.org/content/papyrus-software-designer>

**Description:** Papyrus Software Designer is a project of the Eclipse Papyrus’s galaxy. It enables code generation and reverse engineering for C++ (C soon to come) and Java. Code generation from state-charts is available via model-to-model transformations that are executed prior to code generation (on a temporary model).

Software designer also supports component based modeling via the new OMG standard UCM (unified component model).

## UML Lab

**URL**: <https://marketplace.eclipse.org/content/uml-lab-modeling-ide>

**Description:** UML Lab is the first Modeling IDE to seamlessly combine modeling and programming with an intuitive UML diagram editor and next-gen round-trip engineering. Import your software project and get a nice overview via UML within seconds. Experience truly agile modeling and coding within your Eclipse IDE.

## AgileJ

**URL:** <https://marketplace.eclipse.org/content/agilej-structureviews>

**Description:** Anyone inheriting someone else’s Java code will be familiar with the problem of understanding how it’s been put together, how well structured it is and how best to take it forwards. Stepping through the code helps break the ice, but sometimes it’s good to take a step back from the detail and browse a picture of how the components relate to each other. This is where this tool can help. AgileJ StructureViews analyses all the code in your workspace and pulls out interesting looking groups of classes, presenting them as class diagrams in a web browser. This is a far quicker way to become familiar with an object oriented architecture than reading the source text.

Classically, reverse engineered UML diagrams have been labor-intensive to put together, the end results are cluttered and they cannot be trusted as they fall out of step with the code they are a representation of. However, code structure visualization has a more viable future due to the features we have incorporated in this tool.

1. We include Java element filtering which identifies design patterns, project-specific structures and the usual conventions which Java programmers are familiar with. These filters control how class diagrams are populated and reduce the noise in the information which is presented.

2. The batching engine is designed to be run repeatedly once it is set up, giving you fresh diagrams to browse. The results are not static images. Instead they leverage the GWT platform to present a navigable, filterable view geared towards encouraging engagement with the design.

Version 1.9.0 of AgileJ StructureViews is mostly a re-write of the batching engine. You will notice an order of magnitude in the speed of running batches. The size of war file is now a fraction of the size of that of version 1.8.3, which means less time in IO during batching, faster upload times, and faster loading of class diagram information on the webserver.

## MaintainJ

**URL:** <https://marketplace.eclipse.org/content/maintainj>

**Description:** MaintainJ generates the runtime sequence diagrams for a use case. MaintainJ generated diagrams are dynamic, easy to explore and help Java developers to understand, debug, document and performance tune Java applications.

MaintainJ is the winner of the Best Modeling Product award at Eclipse Community Awards 2012Please check the demo video (<http://maintainj.com/userGuide.jsp?param=overviewDemo>, 3 minutes).

## ModelGoon UML4Java

URL: <https://marketplace.eclipse.org/content/modelgoon-uml4java>

Description: ModelGoon brings new points of view of a Java project. Thanks to its tight connection and interaction with the Eclipse Java Development Tools JDT. ModelGoon provides also round-trip features on Class Diagrams actually as beta.

ModelGoon provides actually four diagrams which are built and synchronized directly from the sources of a Java Project:

* Package Dependencies Diagram : shows the dependencies between packages and highlights circular dependencies. Each dependency line can also display the involved classes.
* Class Diagram : shows a structural view of user selected classes and allows the user to navigate from the diagram's elements to the corresponding Java element in the source code. Semantics, such as composition or aggregation can also be added to associations.
* Interaction Diagram : based on the UML communication diagram, shows overall interactions between objects in order to check responsibilities affectations.
* Sequence Diagram : A high detailed dynamic diagram which is built from a user selected method. Handles most of the statements available in the Java language : if, while, switch/case, for, enhanced for loop, exceptions throwing / catching…

## G9 Database Import

**URL:** https://marketplace.eclipse.org/content/g9-database-import

**Description:** Reverse engineer your database structure into java JPA classes. Create Java classes, or an Xcore or Ecore representation of your exisiting database. g9 Database Model Import is a free g9 Eclipse plugin producing models based on structure information read from databases.

Given a connection to an existing database, the g9 Database Import will read tables and columns in your database (using the database schema) and create the corresponding model either in Java, Xcore or Ecore depending on the project type of your Eclipse project.

Install g9 Database Import, restart Eclipse and create a Java or Xcore project. Select File > Import... > g9 > Import Database Model to start the import wizard (see the Screenshots).

Complete documentation is available from Help > Help Contents. Select g9 Documentation > g9 Database Model Import.

Choose

* which tables to include: by default, all tables are included and a class name based on the table names is suggested
* which columns to include: by default, all columns are included and an attribute name based on the column names is suggested
* the data type mappings for your domain model based on the JDBC types - or use default values

An import of your database to a Java Project will create Java classes, an import to an Xcore Project will create an Xcore model, all other Eclipse Project types will result in an Ecore model. The Java classes may be enriched with JPA annotations. If you work with Xcore or Ecore models you should use Eclipse for Java and DSL Developers.

The g9 Database Import connects to a database using JDBC drivers. The built-in drivers are MySQL, MS SQL Server (jTDS) and Sybase ASE (jTDS). Use the function [Maintain JDBC Drivers] in the g9 Import feature to connect to other databases, e.g Oracle, PostgreSQL, Firebird or a DBMS of your choice. You may download the complete g9 product and use the Database import feature in a g9 Eclipse Project. To be able to exploit the other features in g9, you should create a g9 UI Modeling Project and import your database to an Ecore, Xcore model or Java source with g9 annotations. Please see the User guide for more information about the g9 Database Import feature. If you get Cannot complete the install because one or more required items could not be found, you may have disabled the eclipse update sites.

# IntelliJ IDE

* In general: <https://www.visual-paradigm.com/support/documents/vpuserguide/2381/2385/66580_reverseengin.html>
* Java from UML: <https://www.visual-paradigm.com/support/documents/vpuserguide/2381/2385/66581_codegenerati.html>

# NetBeans

* <https://circle.visual-paradigm.com/docs/ide-integration/netbeans/how-to-reverse-engineer-uml-model-in-netbeans/>
* <https://netbeans.org/download/magazine/02/nb02-part7-uml.pdf>