



Eclipse IDE project resources

Project information for Eclipse developers

[Highlights](#) [Start here](#) [Downloads](#) [Community](#)

Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software. This article gives you links to the latest version of Eclipse, information on IBM's involvement with Eclipse, and a guide to some of the most interesting Eclipse IDE projects. Learn what Eclipse is good for, why it is important, how you can get started, and where to learn more about it.

What is Eclipse?

The short story is that Eclipse is an open source community focused on developing a universal platform of frameworks and exemplary tools that make it easy and cost-effective to build and deploy software.

There is a large consortium of major software vendors, solution providers, corporations, educational and research institutions, and individuals working cohesively to create an ecosystem that enhances and cultivates the Eclipse Platform with complementary products, capabilities, and services.

Eclipse provides value to three constituencies.

Users of Eclipse IDE-based offerings benefit from:

- Providing access to research and knowledge from the entire Eclipse ecosystem
- Higher-quality software that comes under scrutiny from the eyes of the open source community
- The ability to reuse skills because of the consistent Eclipse IDE interface

Java™ technology developers using Eclipse benefit from:

- A world-class Java IDE
- Native look and feel across platforms
- Easy extensions to Java tooling

Developers of Eclipse IDE tools benefit from:

- A portable and customizable platform
- Seamless tool integration
- An end-to-end solution

[↑ Back to top](#)

What is Eclipse good for?

This is a difficult question because the answer depends on the person inquiring. From the researcher's standpoint, Eclipse provides a platform to quickly prototype, collaborate, and share ideas built on a common architecture. From a tool developer's standpoint, you have access to a powerful and extensible platform, which makes it easy to develop higher-quality tools quickly and efficiently. If the whole platform is too heavyweight to use, Eclipse offers the Rich Client Platform (RCP), which is for applications that don't require a common resource model or some of the other features of the platform. The [RCP FAQ](#) is a great resource for understanding the capabilities of RCP. Look at the notable projects and popular applications listed in this FAQ to give you an idea of the capabilities of Eclipse IDE projects.

[↑ Back to top](#)

Why is Eclipse important?

The Eclipse IDE Platform builds confidence and trust by providing the source code for the platform. Software developers are tired of integrating tools and trying to deconstruct how to make tools work together in an environment. Making the Eclipse IDE Platform an open source initiative enables tool developers to do the same and to not only contribute new plug-ins but to also help improve the existing platform. In the end, the importance of Eclipse lies in the fact that everyone -- tool developers and users -- benefits from full disclosure on how to develop tooling at an industry level and, ultimately, benefits the end users.

[↑ Back to top](#)

What does IBM have to do with Eclipse IDE projects?

IBM® is the progenitor of the Eclipse Platform. The best way to explain IBM's involvement with Eclipse is to provide a short history of the three most important phases of Eclipse's climb to success:

The beginning

The platform began development by Object Technology International in 1998 (a subsidiary of IBM purchased in 1996, now known as the IBM Ottawa Lab) to address the problems raised by customers that dealt with the cohesiveness of IBM software tooling. Customers complained that IBM's tooling looked like it came from different companies and didn't work together. IBM took this to heart and listened.

The gift of open source

In 2001, IBM established the Eclipse consortium and gave the gift of Eclipse to the open source community. The goal was to let the open source community control the code and let the consortium deal with commercial relations. There were nine initial members of the consortium, which included IBM partners and competitors. IBM continued to nurture the evolution of the platform by funding various programs like Eclipse innovation grants and sponsoring Eclipse code camps.

The platform was developed using an open source model through an open source license where anyone is welcome to participate.

Independence

IBM wanted more serious commitment from vendors, but vendors perceived the Eclipse consortium as IBM-controlled and were reluctant to make a strategic commitment while IBM was in control. To resolve these problems, IBM relinquished any control. With the support of many companies, the Eclipse Foundation was formed in 2004 as a not-for-profit organization with a dedicated professional staff.

Today, IBM is committed to Eclipse IDE projects more than ever and takes an active part in the Eclipse Foundation as a strategic member. Furthermore, IBM has more developers contributing to Eclipse IDE projects than any other vendor.

[↑ Back to top](#)

What are the Eclipse communities?

Essential to the success of the Eclipse Platform are three intertwined communities:

Committers

- An open, active, inclusive community of committers responsible for developing official Eclipse IDE tooling.

- An example group of committers is the [Eclipse Web Tools Platform](#) project team.

Plug-in developers

- A community that exists outside the committer community that extends the platform to create useful tooling.
- [Eclipse Plugin Central](#) contains a large sampling of plug-in developers.

Users

- A community composed of people who use the tooling developed by committers and plug-in developers.

[↑ Back to top](#)

Why should I contribute to Eclipse?

Eclipse is about many things, but one of those things is the Eclipse IDE ecosystem and the pursuit of profit. Eclipse IDE contributors are building products on top of extensible frameworks that provide value for everyone. The main reasons to contribute are:

1. **Product dependency**
The ability to help direct a project in ways to help your commercial offering.
2. **Branding**
The ability to associate your company and product with the Eclipse brand is favorable from a publicity standpoint.
3. **Commoditization**
The ability to stay aware of the current trends is a powerful advantage over competitors.

There are other reasons that stretch into the development space, including better developer morale and increased product quality by participating in the community process.

[↑ Back to top](#)

How can I become an Eclipse IDE project committer?

The Eclipse Foundation created the [Eclipse Development Process](#), which governs how Eclipse IDE projects are proposed and led. Eclipse is a meritocracy, which means that the more you contribute to Eclipse the more respect you will earn in the committer community. There are currently three ways to become an Eclipse IDE project committer (from easiest to hardest):

- Your employer commits you to an Eclipse IDE project on a full-time basis (working full time on the project allows you to quickly gain the respect of your peers and become a committer).
- Starting a new Eclipse IDE project allows you to be a committer on that project. However, the process of starting a project is fairly intensive and is outlined in the Eclipse Development Process. As the project grows in reliability, predictability, and results, your reputation in the community improves.
- Contributing on a part-time basis or working on a particular aspect of a project. This is the hardest avenue to become a committer due to projects having large full-time populations of committers. The projects evolve rapidly, making it hard for a part-time developer to keep up.

[↑ Back to top](#)

What should I know about using or contributing to Eclipse?

First, all of the content released by the Eclipse Foundation is governed under the [Eclipse Public License \(EPL\)](#). In May

2004, the EPL was approved by the Open Source Initiative (OSI) which makes the EPL an official open source license. The Eclipse Foundation provides several resources to help with licensing issues:

- [Eclipse.org legal information](#)
- [EPL FAQ](#)
- [Eclipse IP Policy](#)
- [Eclipse.org guide to legal documents](#)

The Eclipse Foundation follows a development process based on open source methodologies. The process is called the Eclipse Development Process and dictates how all development should happen under the Eclipse umbrella. To better help people understand this process, there is a set of [Eclipse Development Process Guidelines](#).

↑ [Back to top](#)

What are some of Eclipse's interesting projects?

Here are some of the many projects that demonstrate the flexibility and coolness of the platform:

- **[Business Intelligent and Reporting Tools \(BIRT\)](#)**
BIRT is an open source reporting system for Web applications. BIRT has two main components: a graphical report designer based on Eclipse and a run-time component you can deploy to your application server. Put simply, BIRT lets you add a variety of reports to your applications with the familiar comfort of a drag-and-drop GUI within Eclipse.
- **[Web Tools Platform \(WTP\)](#)**
The WTP project contributes tools for developing J2EE Web applications. The WTP project includes tools to edit HTML, JavaScript, CSS, JSP (and so on), and also provides database access and query tools. The scope of the project is large, and a good resource illustrating the capabilities of the project is available.
- **[Graphical Editing Framework \(GEF\)](#)**
The GEF project allows developers to take an existing application model and quickly build a rich graphical editor. The project includes examples that range from a graphical circuit designer to a WYSIWYG text editor to help you get started.
- **[Visual Editor \(VE\) Project](#)**
The VE Project is a framework for developing GUI editors within Eclipse. By default, it contains support for Swing, SWT, and even RCP-based GUIs. There is a good Flash demo that demonstrates the capabilities of the VE project.
- **[C/C++ Development Tools \(CDT\)](#)**
The CDT project provides a functional C/C++ IDE for the Eclipse Platform.
- **[Mylyn](#)**
If you have used an IDE in the past and noticed the problem of information overload when there are too many projects or objects on the screen, Mylyn is for you. The project, according to its definition, aims to avoid information blindness when staring at Eclipse.
- **[Eclipse Communications Framework \(ECF\) Project](#)**
The ECF provides APIs that simplify the creation of reliable distributed applications. The project is still in its infancy, but it has already demonstrated that the Eclipse Platform can be used for communications. Sample work from the ECF project includes the ability to use Jabber from within Eclipse and share editors visually.

Visit [Eclipse.org](#) for more interesting [projects](#).

↑ [Back to top](#)

Who wrote this guide?

Chris Aniszczyk is a software engineer at IBM (Tivoli® Security) and a graduate of IBM's Extreme Blue internship program. He is an open source enthusiast at heart, working on the Gentoo Linux (<http://www.gentoo.org>) distribution, and is a committer on the Eclipse Modeling Framework Technology (EMFT) project.

[↑ Back to top](#)