



A whirlwind tour of Eclipse Europa

Eclipse V3.3 contains significant changes to 21 projects; learn how Europa will boost your productivity

Chris Aniszczyk, Software Engineer, IBM

Summary: With the simultaneous release of 21 projects this year, the Eclipse Europa release marks a significant improvement over last year's release train of 10 projects. Get a brief overview of each of the projects associated with the Europa release.

Date: 29 Jun 2007

Level: Intermediate

Activity: 1615 views

Comments: 0 ([Add comments](#))

★★★★☆ Average rating (based on 22 votes)

Simply put, Europa is the simultaneous release of 21 major Eclipse projects.

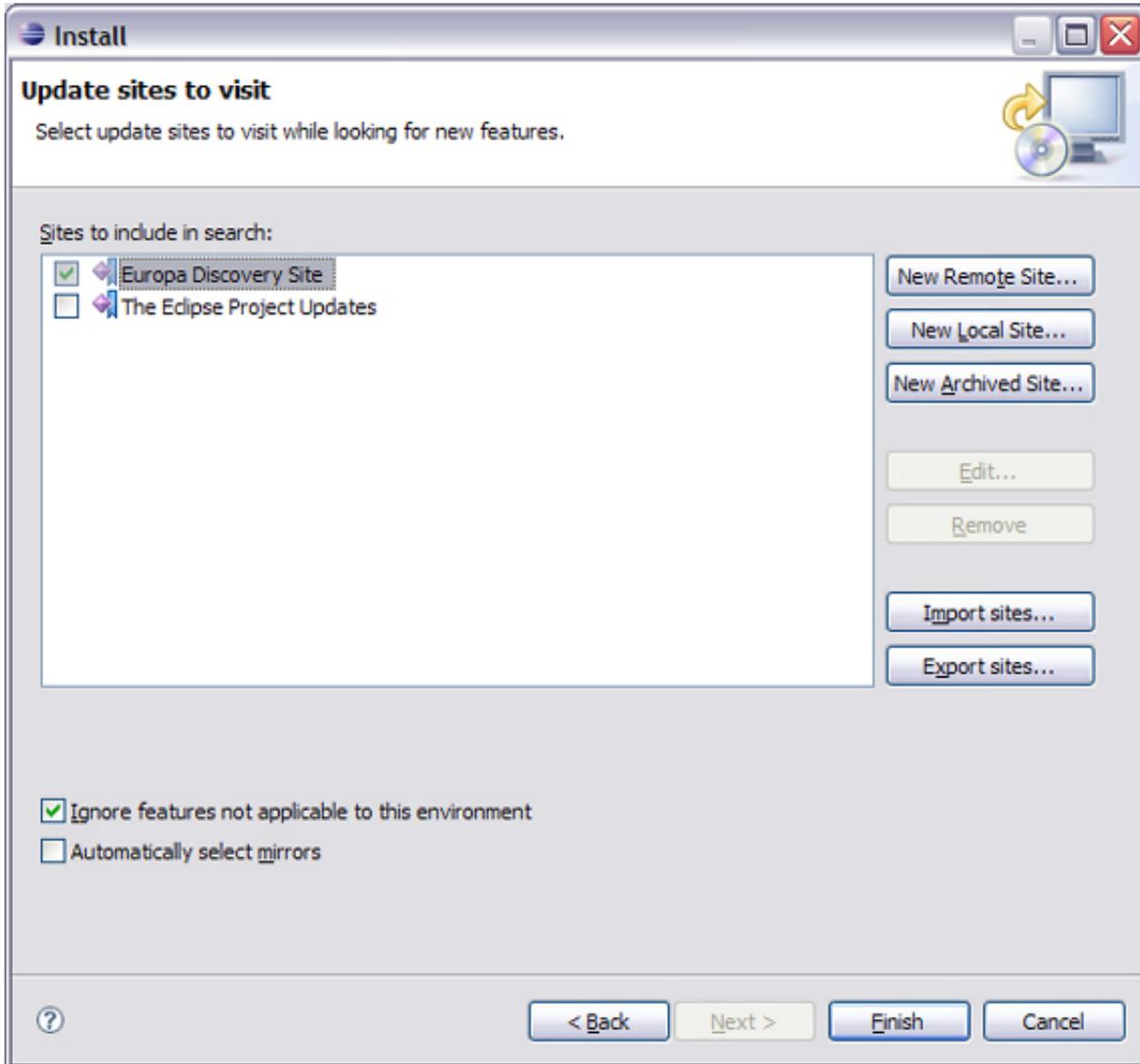
The important thing to note about Europa and Eclipse release trains is that even though it's a simultaneous release, it doesn't mean these projects are unified at all. Each project remains a *separate* open source project, operating with its own project leadership, its own committers, and its own development plan. In the end, Europa is about improving the productivity of developers working on top of Eclipse projects by providing a more transparent and predictable development cycle. It's easier to ship software when you know it will always be available on a certain date.

Getting Europa

Before we get into the details about each of the 21 projects, let's complete a quick, hands-on exercise that will result in the installation of Europa on your machine.

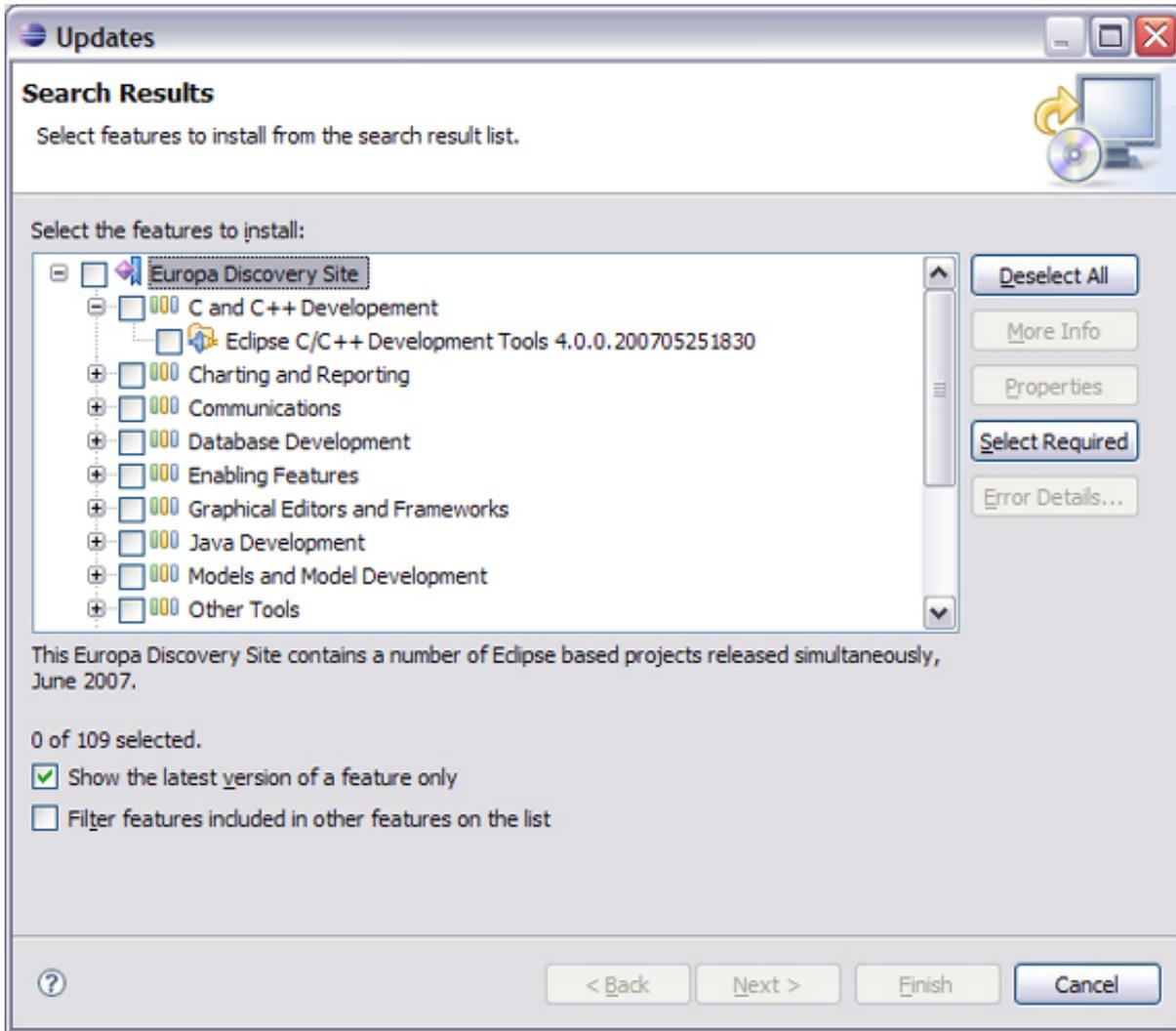
The first step in acquiring Europa is to download the [Eclipse V3.3 SDK](#). Next, launch Eclipse and access the update manager via **Help > Software Updates > Find and Install** (see Figure 1). Enter the proper Europa update site information for your locale (see [Resources](#)) if it isn't available at the Europa discovery site by the time you read this.

Figure 1. Update site configuration



Once you are connected to the Europa update site, you should see the list of available features that are part of the Europa release train (see Figure 2). It's as simple as that. Once connected to the update site, you simply choose the features you want to install into your copy of Eclipse and start using them. In these two simple steps, you are primed to take advantage of the Europa release.

Figure 2. Europa update site



The release train

A brief overview of each project will be given, followed by a quote from an Eclipse committer about what's exciting about the project in the Europa release. The quotes are from e-mail conversations I had with each project committer. Consider this a dialog between developers — no fluff or marketing hand-waving was expected or allowed.

Eclipse Platform

The Eclipse Platform project is the heart of Eclipse. The project actually consists of many "subprojects," but in my view, the platform project consists of four main pieces:

Plug-in Development Environment (PDE)

"In what has been described in the [Planet Eclipse blog](#) as a 'Plug-in Developer's Christmas,' the Plug-in Development Environment (PDE) brings to Europa a

major plug-in manifest editor facelift, complete with rich-source editing features that include the highly anticipated content assist," said Wassim Melhem, Eclipse PDE lead and IBM® employee. "Other notable features include cheatsheet authoring, target provisioning, and enhanced RCP and OSGi tooling. Last but not least, with nearly 800 enhancement and bug reports fixed this past year, robustness is certainly a key highlight of the PDE 3.3 release."

Core

The core of the Eclipse Platform mostly comes from the [Equinox](#) project. Equinox serves as the reference implementation for [OSGi](#), which can be considered as forming the building blocks of Eclipse. Eclipse's plug-in model and notions of extensions are housed within Equinox, along with a few other technologies.

User interface (UI)

All things related to the [workbench](#) live here.

Java™ Development Tools (JDT)

This is the first component most developers come across when they initially use Eclipse. Eclipse ships with a world-class Java development environment, known as the JDT.

Aspect Java Development Tools (AJDT)

The [AJDT](#) project provides Eclipse Platform-based tool support for aspect-oriented software development with AspectJ. The AJDT makes the life of an aspect developer easier, similar to the way the JDT made the life of a Java developer easier.

"The AspectJ Development Tools project (AJDT) joins the Eclipse release train for the first time this year. Last year, the project moved from being a technology project to become a tools project, reflecting the project's level of maturity and widespread usage. The project provides everything required for aspect-oriented programming: the AspectJ compiler and weaver, an editor for AspectJ source code, and additional views and other functionality designed to allow developers to get the maximum benefit from AOP," said Matt Chapman, AJDT lead and IBM employee. "Since last year's 1.4 release, there have been a number of improvements, including:

- "Refactoring participant to update aspects when Java types are renamed
- "Better support for binary weaving across projects using the aspect-path setting
- "Making the aspect-path and in-path settings more consistent with the regular Java classpath setting
- "Improvements to the Crosscutting Comparison view
- "New functionality to highlight changes to the crosscutting relationships in a project"

Business Intelligence and Reporting Tools (BIRT)

The **BIRT** project strives to bring an Eclipse-based reporting system that integrates with your application to produce compelling reports for Web and PDF. BIRT provides core reporting features such as a graphical report designer, data access, and scripting support.

"BIRT 2.2, which is part of the Europa release, offers many new features to assist in report development, ease of framework integration, and exporting of content. These include a new dynamic cross-tab element, which supports hierarchal reporting complete with level aggregation, filtering and sorting, new emitters for XLS, Doc, PPT, and postscript formats, a new Web service data access driver, and many improvements to existing features," said Jason Weathersby, BIRT evangelist and Actuate employee. "Additionally, BIRT provides several new project types, such as the Web project, which allows the BIRT Viewer and new tag libraries to be deployed using WTP mechanisms and new ODA projects that facilitate building your own data drivers."

Buckminster

The **Buckminster** project consists of frameworks and tools for automating build, assemble, and deploy development processes in complex or distributed component-based development. Buckminster allows development organizations to define fine-grain "production lines" for the various stages in software production and to automate the execution of corresponding processes. A common example of such a process is a developer joining a new team and being expected to set up the development environment from scratch. Buckminster allows new developers to materialize their development workspaces and quickly become productive team members.

"Europa will be the first formal release of Buckminster, which has been expanded significantly in scope and production quality. Buckminster now supports fine-grain control and greater variability in the configuration resolution process through dynamic component queries and a two-phase resolution model. We have also introduced an action framework, which supports orchestration of assembly processes involving native Eclipse (PDE) and external build mechanisms. Headless execution capabilities have also been expanded. Extensibility is greatly improved through formal XML schemas for all artifacts, and usability is enhanced with an initial set of graphical editors," said Thomas Hallgren, Buckminster committer and Cloudsmith employee.

C/C++ Development Tooling (CDT)

The **CDT** project aims to bring a fully functional C and C++ development environment to the Eclipse Platform, similar to what JDT did for Java development.

"The CDT is becoming the de-facto standard C/C++ IDE among commercial vendors targeting Linux® and embedded. The CDT enjoys a large community of users

working with the Windows® GNU compilers, as well. The breadth of the CDTs user base — from tiny barebones embedded systems to high-performance computing applications running on massive clusters — really highlights the flexibility of the frameworks and features that it provides," said Doug Schaefer, CDT lead and QNX Software Systems employee.

"CDT 4.0 is our largest release, with over 20 contributors working in it and one we hope will bring the CDT to the next level of adoption. The user experience, especially when creating new projects has been greatly simplified with a streamlined New Project wizard that hides some of the details of the underlying build systems. It also offers the ability to create new projects based on templates that provide skeletal code and project settings," Schaefer said. "A lot of work has gone into the CDT editor, which offers many more identifier coloring options and a faster content assist. Three new views give users greater insight into the structure of their code, including an Includes Browser, Call Hierarchy View, and a new Type Hierarchy View. The set of build and debug tool chain integrations grows with a cleaner MinGW integration, a GDB hardware debugging integration, and a new integration for IBM's xLC compiler. And, finally, the path to CDT's future is laid out with the introduction of new mechanisms for handling additional programming languages that includes a new experimental C parser and a parser for UPC, Unified Parallel C, using the LPG parser generator."

Dynamic Languages Toolkit (DLTK)

The [DLTK](#) is a tool for vendors, researchers, and end users who rely on dynamic languages. DLTK comprises a set of extensible frameworks designed to reduce the complexity of building full-featured development environments for dynamic languages, such as PHP and Perl. Furthermore, besides a set of frameworks, DLTK provides exemplary TCL, Ruby, and Python development environments ready to use out of the box.

"For the Europa release, the DLTK provides TCL and Ruby components with a set of features you may find in Java Development Tools. Starting from code editor with code assistance and code outline, code navigation features like Open Type and Go to declaration, powerful search features like find references up to interactive console and integrated debugger, these features bring Europa a powerful development environment for Ruby and TCL programming languages," said Andrei Sobolev, DLTK project co-lead and XORED employee.

DSDP Device Debugging (DSDP-DD) and Target Management (DSDP-TM)

The [DSDP-DD](#) project creates new frameworks and extensions to the existing platform debug framework to support the three phases of device software development: hardware bring-up, platform software development, and application software development. The Device Debugging project will build enhanced debug models, APIs, and views to provide greater visibility into and control over device

software targets. The [DSDP-TM](#) project creates data models and frameworks to configure and manage remote systems, their connections, and their services.

"With the Europa release, DSDP introduces V2.0 of the Target Management project and V0.9 of the Device Debugging project. TM 2.0 focuses on several architectural improvements, including better separation between UI and non-UI components, cleanup of APIs, improved flexibility for system type and action contributions, and better usage of jobs. TM 2.0 also adds a shell processes subsystem, a Telnet subsystem and associated view, DNS-SD discovery, an FTP listing parser extension point, and an improved EFS provider. TM continues to experience excellent growth in its user community, with several new commercial products planning adoption. DD 0.9 is the DD project's first public release," said Doug Gaff, DSDP PMC and WindRiver employee.

"DD 0.9 introduces the Debugger Services Framework (DSF) — an extensible framework that leverages the new adaptable debug APIs introduced in Eclipse 3.2. DSF is geared toward commercial-embedded debug engine implementations in Eclipse. DD 0.9 also provides a memory rendering for the platform memory view that mimics the traditional memory view found in embedded debuggers. Finally, DD 0.9 provides a preview release of an XML editor that implements the SPIRIT consortium's IP-XACT V1.3 standard for System On Chip (SoC) hardware description. Other DSDP subprojects will also be releasing off-train this year," Gaff said.

DataTools Platform (DTP)

The [DTP](#) project aims to bring frameworks and tools to the domain of data management. Examples of frameworks include location and management of data source drivers, and configurations for access to particular data source instances. Once a connection is successfully made, the next task often is to explore the data source with tools. Some of these operations might be carried out by GUI actions, others directly through commands. For example, users — both developers and administrators — typically will create, edit, and test SQL for these commands. Assistance with editing SQL through code completion, formatting, and dialect specialization greatly enhances productivity. Further, the ability to execute or debug commands — both SQL and stored procedures — rounds out the rapid development process that Eclipse supports so well.

"DTP 1.5 builds on the foundation of the DTP 1.0 release last December to provide specialized database support for Sybase SQL Anywhere and HSQLDB, as well as an ODA Web service driver. NEC Soft, Japan, has also joined the DTP team in this release and has provided a wizard for creating database tables for DTP 1.5. As demonstrated by a strong showing at EclipseCon, including a sponsored talk about DTP integration by Ingres, the DTP ecosystem continues to expand at a healthy rate," said John Graham, DTP PMC chairman and Sybase employee.

Eclipse Communication Framework (ECF)

The [ECF](#) project consists of a framework for supporting the development of distributed Eclipse-based tools and applications. It can be used to create other plug-ins, tools, or full Eclipse RCP applications that require asynchronous point-to-point or publish-and-subscribe messaging.

"ECF brings communication capabilities to the Europa release with a uniform set of APIs to be consumed by developers. Support for transferring files through HTTP, BitTorrent, and messaging protocols like XMPP and IRC is provided by ECF out of the box. ISVs that wish to add their own protocols and communications layers can contribute them directly through the framework. The framework itself is stripped-down enough that it can even be used on embedded devices, allowing developers to easily write applications that require having Pocket PCs and notebooks talk to each other," said Remy Chi Jian Suen, ECF committer, and Scott Lewis, ECF committer.

Project Dash (Eclipse Monkey)

Project Dash (also known as Eclipse Monkey) makes it possible to script Eclipse similar to what [BeanShell](#) does for Java.

"Simply put, the [Eclipse Monkey](#) project 'script-enables' Eclipse. It enables developers to use JavaScript (extensibility points are provided to support other scripting languages, such as Ruby) to control any aspect of their Eclipse environment. Examples of what could be done include scripts to configure Eclipse consistently across installations, macros for performing a set of actions (including inserting/modifying code), and inserting code snippets," said Paul Colton, Eclipse Monkey lead and Aptana employee.

"The DOM available to the scripter can be fully extended to add support for any capability that is not already included. Another powerful feature of Eclipse Monkey is the ability to create new Eclipse views that contain Web content. That is, full HTML/CSS/JavaScript views can be created with two-way communications with Eclipse. For example, a view that displays real-time Google searches based on the keyword your cursor is on. Another example could be an RSS reader of code snippets: If a link is clicked, the snippet can be inserted from the RSS feed into your working document. Aptana (an open source, Eclipse-based Ajax development environment) fully leverages Eclipse Monkey in many of the ways described here," Colton said.

Eclipse Modeling Framework (EMF)

[EMF](#) is a modeling framework and code-generation tool for building tools and other applications based on a structured model. EMF lets you build models quickly by taking advantage of EMF facilities. For example, one feature EMF provides is support for persisting models to XML and other formats.

"The Eclipse Modeling Framework provides powerful generative and runtime capabilities for applications based on structured data models. From a simple class diagram, XML Schema, or even Java interfaces, you can generate a complete Java implementation of the model, along with an editor for it, and take advantage of EMF's facilities for persistence, notification, validation, and change recording in your application," Marcelo Paternostro, EMF Committer and IBM employee.

"Europa includes EMF 2.3, which, for the first time, allows developers to exploit Java 5 concepts and features in their models. Generics are now fully supported in EMF's metamodel, Ecore, and in the EMF code generator," Paternostro said. "Other important Java 5 language features are also supported, including enumerated types, annotations, and the enhanced for loop. Developers still have the option to generate 1.4-compatible code, which can be run on the EMF 2.2 runtime.

"Other important and long-awaited features include Ecore model validation, Ant support for code generation based on annotated Java interfaces, and reference keys, which provide a more robust alternative to the default index-based reference serialization. Finally, EMF 2.3 will be delivered in fine-grain features, allowing for more flexible use in projects and products, without introducing unwanted code, user interface, or dependencies."

Eclipse Modeling Framework Technology (EMFT)

The [EMFT](#) project was initiated to incubate new technologies that extend or complement EMF. These technologies over time — if found useful — eventually graduate into EMF. Some example technologies found in EMFT include the ability to generically search, compare, and persist EMF models to a database.

Graphical Editing Framework (GEF)

The [GEF](#) project allows developers to take an existing application model and create a rich graphical editor quickly. GEF also contains a lightweight visualization layer, known as Draw2D, which is similar to Java2D.

"For the Europa release, GEF 3.3 is essentially a maintenance release in terms of the delivery of several small features and bug fixes. The GEF team did spend considerable effort making sure GEF aligned with the capabilities of Eclipse 3.3, such as testing on the new platforms like Microsoft® Vista, adopting new Eclipse 3.3 release engineering tools, and so on," said Anthony Hunter, GEF/GMF committer and IBM employee.

Graphical Modeling Framework (GMF)

The [GMF](#) project provides a generative component and runtime infrastructure for developing graphical editors based on EMF and GEF. The project aims to provide these components, in addition to exemplary tools for select domain models that

illustrate its capabilities.

"GMF remains an active and popular project within the Eclipse Modeling Project, and provides several enhancements with the 2.0 release to its runtime and tooling components to further simplify the task of building diagram-based editors for EMF models," said Richard Gronback, GMF lead and Borland employee. "GMF has also found two clients within the Eclipse ecosystem, with the introduction of the UML2 Tools component in MDT and the BPMN modeler within the SOA Tools project."

Model Development Tools (MDT)

The [MDT](#) project focuses on "big M" modeling within the Eclipse Modeling project. The project has two main purposes: Provide an implementation of industry standard metamodels; and provide exemplary tools for developing models based on those metamodels. For example, the MDT project contains implementations (as metamodels) of UML2, XSD, and OCL.

Model To Text (M2T)

The [M2T](#) project focuses on the generation of textual artifacts from models. Its purpose is threefold:

- To provide implementations of industry standard and de-facto Eclipse standard model-to-text engines
- To provide exemplary development tools for these languages
- To provide common infrastructure for this languages

For Europa, the M2T project will release a small component called JET.

"[Java Emitter Templates](#) (JET) is an JSP-inspired code generator designed to work in the Eclipse workspace environment. Instead of producing HTML Web pages in response to an HTTP request, JET produces Eclipse resources (files, folders, projects) when given an input model. JET can read a number of input model types, including simple XML documents and any Eclipse Modeling Framework (EMF) based document," said Paul Edler, JET lead and IBM employee.

"In the Europa release, JET has moved from the Eclipse Modeling Framework Technologies (EMFT) project into the new Model to Text (M2T) Project. JET and M2T remain in the incubation phase as the M2T project continues to ramp up. Ultimately, M2T will include other Eclipse-based template languages (xPand from openArchitectureWare and MTL — an implementation of the OMG's MOF Model to Text specification), as well as common infrastructure to support both the implementation and consumption of model-to-text languages."

Eclipse Mylyn (formerly known as Mylar)

"Mylyn is a task-focused UI that reduces information overload and makes multitasking easy. It does this by making tasks a first-class part of Eclipse, and integrating rich and offline editing for repositories, such as Bugzilla, Trac, and JIRA. Once your tasks are integrated, Mylar monitors your work activity to identify information relevant to the task at hand, and uses this task context to focus the Eclipse UI on the interesting information, hide the uninteresting, and automatically find what's related," said Mik Kersten, Mylar lead and Tasktop employee.

"Mylyn 2.0 makes tasks as easy to browse and navigate as files and Java elements," Kersten said. "Tasks can be viewed with flexible presentations, focused to show the most relevant ones and organized with working sets. Just as the Eclipse JDT makes Java references easy to navigate, with Mylyn 2.0, pervasive hyperlinking tasks are only a click away whether referenced in source code or revision histories.

"Mylyn provides a Rich Internet Application (RIA) infrastructure and UI for working with task repositories. This means that issue tracking and project management tools, including Bugzilla, JIRA, Trac, Rally, CollabNet Enterprise Edition, and XPlanner, can all be used with Mylyn's rich editor, opened instantly with no roundtrips to the server, and accessed easily when offline. Mylyn's Task List can now be used as a single inbox for all your tasks, allowing you to unclutter your e-mail inbox by turning off e-mail notifications," Kersten said.

"The fundamental productivity benefits of Mylyn come from the Task-Focused UI, which reduces information overload and makes multitasking easy. Mylyn 2.0 facilitates working in a task-focused way by making recalling a task context almost instant. Additional features such as the task context pop-up dialog and task activation dialog it easier to adopt the Task-Focused UI without changing existing work practice," Kersten said. "In addition, context sharing has been streamlined and a context editor provided to better support teams."

SOA Tools Platform (STP)

There's no such thing as a single kind of SOA developer, so the [STP](#) project's goal is to bring together quality frameworks and tools to aid developers in constructing and deploying services using popular SOA solution technologies.

"SOA Tools is new addition to the mid-year big-bang Eclipse release train this year, and as part of it, we will be shipping new frameworks for Java Annotation-driven service development approaches, such as JAX-WS and SCA/Java, a BPMN Modeler framework, and a BPEL compiler," said Oisin Hurley, STP lead and IONA employee. "To help package and deliver your services, we've constructed a sophisticated deployment framework that enables the construction of multiple deployment profiles and multipackage deploy. This is integrated with the WTP Server framework to make sure that you can reuse any code you already have for deploying to single servers."

Eclipse Test & Performance Tools Platform (TPTP)

TPTP provides an open platform supplying powerful frameworks and services that allow software developers to build unique test and performance tools — both open source and commercial — that can be easily integrated with the platform and with other tools. The platform supports a broad spectrum of computing systems, including embedded, stand-alone, enterprise, and high-performance, and will continue to expand support to encompass the widest possible range of systems.

"TPTP provides a rich set of test, profiling, and monitoring tools. However its true value can only be realized by being part of a core typical user use case," said Harm Sluiman, TPTP committer, IBM employee. "The main new feature of TPTP (besides the normal maintainer and bug fixing) is that we have moved to JVMTI from JVMPI, which has given us better and more stable heap and thread analysis support, along with the ability to support Java 6 going forward."

Web Tools Platform (WTP)

The **WTP** project extends the Eclipse Platform with tools for developing J2EE Web applications. The WTP project includes source editors for HTML, JavaScript, CSS, JSP, SQL, XML, DTD, XSD, and WSDL; graphical editors for XSD and WSDL; J2EE project natures, builders, and models, and a J2EE navigator; and a Web service wizard and explorer. Generally speaking, if the task you need to complete is related to Web development, you will usually find the right tool in the WTP.

"I'm excited about this Europa release of WTP 2.0. This is partially because we have better componetization, so users can install just our XML tools, from Europa's Discovery Site, or just the non-Java related Web tools (HTML, CSS, etc). And, for these smaller components, JDT is no longer even dragged in! These smaller installs have been much requested by the community and I am hoping they will attract a whole new set of users and integration with other Eclipse-based projects," said David Williams, WTP project management committee co-lead, and IBM employee.

"Additionally, the Java-related tools, for JEE Web Development, have gotten much richer this release. We are happy that the sub-projects for JSF and Dali (for Java Persistence Tooling) have graduated from incubation and provide some great tools for using these standards in JEE applications," Williams said. "With this release, we provide basic support for JEE 5 applications and servers, and have gained contributions for new standards, such as Axis2."

"While we have more to do, we have made progress in improving usability and 'out of the box' experience. One example is we have gotten a contribution for a much requested visual editor for editing HTML, JSP, and JSF, which should improve productivity while developers create Web pages," Williams said. "Additionally, we have a built-in 'preview' server for users' first, simple applications, as well as a new 'discovery' mechanisms for finding server adapters that are available to be installed after users get more advanced."

Go try Europa -- It's free

I have explained the Europa release train as efficiently as possible while remaining relevant technically. This was accomplished by giving a unique developer-to-developer perspective on each of the projects by gathering a relevant quote from project developers. From my perspective, the differences between the Callisto and Europa releases are validation that Eclipse is growing and will continue to be a relevant and viable platform for software development and deployment. So what are you waiting for? [Give Europa a try](#).

Resources

Learn

- Read "[Europa Simultaneous Release project](#)."
- Home of the [Europa release train](#).
- Documentation, articles, and downloads of Eclipse are available from the [Eclipse Foundation](#).
- Interested in what's happening in the Eclipse community? Check out [PlanetEclipse](#).
- Check out the available Eclipse plug-ins at [Eclipse Plug-in Central](#).
- Check out [EclipseLive](#) for webinars featuring various Eclipse technologies.
- Check out the "[Recommended Eclipse reading list](#)."
- Browse all the [Eclipse content](#) on developerWorks.
- Users new to Eclipse should check out [Eclipse project resources' Start Here](#).
- Expand your Eclipse skills by checking out IBM developerWorks' [Eclipse project resources](#).
- To listen to interesting interviews and discussions for software developers, check out [developerWorks podcasts](#).
- For an introduction to the Eclipse platform, see "[Getting started with the Eclipse Platform](#)."

- Stay current with developerWorks' [Technical events and webcasts](#).
- Watch and learn about IBM and open source technologies and product functions with the no-cost [developerWorks On demand demos](#).
- Check out upcoming conferences, trade shows, webcasts, and other [Events](#) around the world that are of interest to IBM open source developers.
- Visit the developerWorks [Open source zone](#) for extensive how-to information, tools, and project updates to help you develop with open source technologies and use them with IBM's products.

Get products and technologies

- Check out the latest [Eclipse technology downloads](#) at IBM [alphaWorks](#).
- Download [IBM product evaluation versions](#), and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.
- Innovate your next open source development project with [IBM trial software](#), available for download or on DVD.

Discuss

- Chat with other Eclipse developers and committers on [IRC](#).
- The [Eclipse Platform newsgroups](#) should be your first stop to discuss questions regarding Eclipse. (Selecting this will launch your default Usenet news reader application and open eclipse.platform.)
- The [Eclipse newsgroups](#) has many resources for people interested in using and extending Eclipse.
- Participate in [developerWorks blogs](#) and get involved in the developerWorks community.

About the author



Chris Aniszczyk is an Eclipse committer at IBM Lotus who works on OSGi-related development. His primary focus these days is improving Eclipse's Plug-in Development Environment (PDE) and spreading the Eclipse love inside of IBM's Lotus organization. He is an open source enthusiast at heart, specializing in open source evangelism. He evangelizes about Eclipse in his blog, and he's honored to represent the Eclipse committers on the Eclipse Foundation's board of directors. He's always available to discuss open source and Eclipse over a frosty beverage.

[Trademarks](#) | [My developerWorks terms and conditions](#)