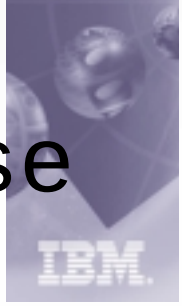
Introduction to Eclipse



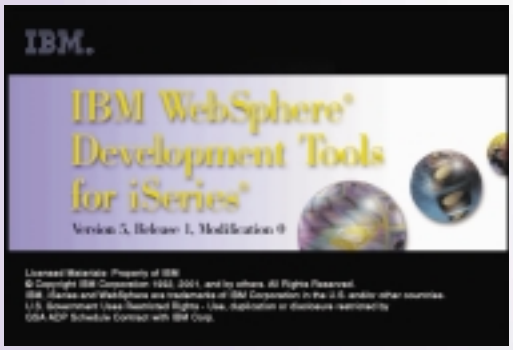
By Phil Coulthard and George N. Farr

Have you heard of Eclipse? Probably you have. If not, you will. This article, the first in a series, introduces the new IBM® Eclipse technology, and touches on its implications for you as an iSeries® programmer, although future articles will have more to say on this topic.

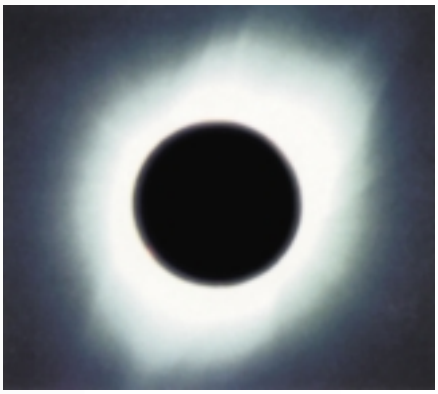
What Is Eclipse?

In November 2001, IBM announced and released Eclipse to the open source community. Eclipse is a Java-based integrated development environment (IDE), which is fully extendable by users, customers and vendors. The code base is written in Java and runs initially on Windows and Linux. To extend it, programmers write Java® code in the form of "plugins." Eclipse, both in binary form and the source code itself, can be downloaded from the Eclipse Web site www.eclipse.org.

Think of Eclipse as a slate from which many great tools can be created. Out of the box, it has tools for Java development, already built and ready for use. That is, you can use Eclipse as an effective Java development environment, both for the purpose of writing Java applications and for writing plugins for Eclipse. The power and potential of Eclipse is not the tooling supplied with it, though, it is the tooling that IBM and others will write on top of it. This tooling could be for anything from Web applications to RPG applications.



Some of this tooling will be offered back to the Eclipse code base, as open source contributions. Other tooling will be packaged and sold as part of a product in which Eclipse is embedded. The open source agreement for Eclipse allows vendors to do just that, at no cost to them. Further, as IBM and others start selling popular products built on Eclipse technology, vendors will start just selling their tools as plugins, that pre-req any of these Eclipse-based tools.



At the end of November, 2001, IBM announced the Eclipse.org Consortium. This group of companies will oversee the open source initiative, and each of them will be offering Eclipse-based products. The initial companies on the consortium board of directors are: Merant, IBM, Borland, QNX Software Systems, Rational Software, RedHat, SuSE and TogetherSoft. This is not a final list; the current list of companies on the board can be found at the Eclipse Web site.

Beyond the consortium, thousands more companies are already in the process of building Eclipse-based products. Since the November 7 announcement of Eclipse, there have been over 4000 downloads a day of Eclipse. Hundreds of companies have been working closely with IBM's Partner World to develop their Eclipse skills and products.

Eclipse offers a number of features and technologies that make it a compelling starting point for tools. It is a project-based IDE, where users create projects that can be shared by a team. There is a growing choice of server-side change management repositories that support Eclipse, including Rational ClearCase and CVS (Concurrent Versions System), the open source change management tool (www.cvshome.org). Projects contain folders and files (it is a file-system based architecture). Tool writers immediately inherit a full function IDE that allows the users to create, manipulate and work with team-oriented projects. Tool writers can use the supplied project types or create their own project types. The tool writer then focuses on writing tightly integrated tools that are launched from projects or files of a particular type. End users will not be able to tell where Eclipse ends and a plugged-in tool begins, so seamless is the integration. There are many specific areas of Eclipse than can be customized or extended by tool writers, including the run, debug and build actions for projects, the built-in source editor for language-sensitive editing, and every aspect of the user interface.

Why Eclipse?

So why did IBM create Eclipse and donate it to the open source community? The reason is to spur on a new world of consistent application development tooling in which each vendor can focus on their specific tooling without constantly reinventing the basic infrastructure that all application development tools need, such as project and team support, source editors and debuggers. Think of Eclipse as the building of a new shopping mall. Eclipse is the building itself, while the vendors who write tools will be the individual stores. The more stores there are, and the more successful they are, the better the results will be for everyone, especially the shoppers.

Of course tools in and of themselves are not important. It is all about the applications that the tools enable you, the user of the tools, to build. From a rich, robust, modern, interlocking and consistent set of tools, modern e-business applications will flow. Applications are the end; compelling tooling provides the means.

As iSeries developers, you understand this truth perhaps better than most people do. Since the first days of AS/400, you have enjoyed a common, productive and integrated set of tools for developing AS/400 and now iSeries traditional applications. These tools include PDM, SEU, SDA, RLU, DFU and the system debugger. These are the core supplied tools, but there are also many additional tools built and sold by business partners. These tools at minimum mimic the look and feel of the IBM tools, and often go beyond that and offer actual integration with PDM though user-defined actions. Indeed, you can consider Eclipse to be the modern-day PDM. It is the integration point for various tools from various vendors. It is also where you will spend your day, as a developer. Further, like PDM, this toolset is extendible by you to suit your needs. Indeed, Eclipse is the very ultimate extendible platform.

As a developer, Eclipse ultimately will mean you will have a plethora of products and tools to choose from, all of which snap into your Eclipse IDE and offer a consistent development and team experience. The benefits of this include a reduced learning curve and reduced footprint and, at long last, tool integration! That is, output from one tool can be used as input to another tool. Also, having a common base on which vendors can quickly and easily build tools that have a large market will mean more competition and hence more tools to choose from for any given task. Indeed, as the open source aspect takes hold, many of these tools, or at least some version of them, may be offered free.

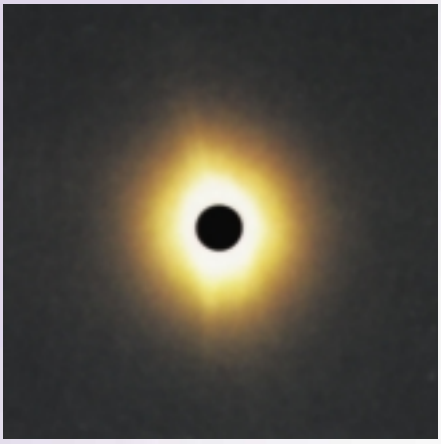
Beyond Eclipse

IBM is doing more than just offering Eclipse to the open source community. It is rebuilding its entire application development product set on top of Eclipse, as well as building many exciting new tools on top of Eclipse. The first major offering from IBM is WebSphere® Studio Application Developer (WSAD). This product is the follow-on to both VisualAge® for Java Enterprise Edition, and WebSphere Studio Advanced Edition. WSAD is built on a snapshot of Eclipse. On top of this base, WSAD adds the following tools:

- Java tools (most of which are inherited from Eclipse)
- EJB and J2EE tools
- Web tools (follow-on to WebSphere Studio)
- Server tools (WebSphere Application Server is built in for local testing of Web applications. TomCat, the Apache-based open source application server, is not built in, but it is also supported.)
- Remote Publishing tools (for publishing Web and J2EE applications to a remote WebSphere Application Server)
- Web Services tools
- XML tools
- Database tools
- Profiling tools (for profiling performance, execution and memory aspects of Java logic)

WSAD is a comprehensive and compelling product, purchasable from IBM. While it incorporates most of the functionality of its two predecessor products and adds significant new functionality, its price is still less than the combined price of its predecessor products. WSAD has already generated considerable excitement, not only for its rich capabilities but also because it is the first tangible instantiation of Eclipse in product form. Many vendors are already busy writing plugin tooling for WSAD.

Eventually there will be a trimmed-down version of WSAD, possibly named WebSphere Studio Site Developer (WSSD), but of course this name is subject to change. It will offer a subset of the rich WSAD function at a subset of the price. There will be other Eclipse-based offerings from IBM, and they will come soon.



It is said that necessity is the mother of invention, and IBM needed Eclipse. With its diverse application development product line, developers were finding it difficult to ensure all the tools worked well together. Further, many tools offered overlapping functionality -- majoring in one area but still minoring in many others. Without a common design for reuse, it is not possible to share components between diverse products. As a result, many products often had to manually duplicate another product's functionality, or pre-req that other product. This was not only inefficient for IBM, but for customers too. With Eclipse, IBM now has a common base for the core infrastructure all AD products need, as well as the means to create plugins that can be used by multiple products, eliminating redundancy.

Eclipse and iSeries

To iSeries developers, Eclipse itself is perhaps not that interesting. Rather, it is the products built on Eclipse that are interesting. For example, WebSphere Studio Application Developer is a product that is directly useful to iSeries developers for writing Java code, XML languages or parsers, Web Services, Web applications, or full-blown J2EE applications. It is applicable because all of these are completely portable, so the output from using the tooling can be run on any operating system with an appropriate run-time environment. The capability to directly publish Web and J2EE applications to a remote WebSphere Application Server (4.0 or higher) does not yet support iSeries, but an OS/400 PTF will soon provide this capability. In the meantime, you simply export the application to your local drive and use the WebSphere Administrative Console to import it.

The beta currently available for WebSphere Studio Site Developer is also directly applicable to iSeries customers. Indeed, the exciting part of the new generation of applications is their portability, which benefits programmers by allowing them their choice of tools on their choice of client operating systems. We in the iSeries application development team in Toronto have the responsibility to ensure the tools do work with iSeries and to fix them if they don't.

While enabling these tools to iSeries is indeed necessary, it is not sufficient. The primary business logic languages on iSeries are RPG and COBOL, both for traditional applications and for many new e-business applications. Further, almost all applications on iSeries use varying amounts of CL and DDS. To fully realize the potential of Eclipse for iSeries development, there will need to be Eclipse-based tooling for these languages. As many of you know, there are highly productive Windows-based tools for these languages today in the CODE/400 toolset that is now part of WebSphere Development Tools for iSeries (WDT). WDT is the client-side part of the WebSphere Development Studio for iSeries (WDS) product, which is now the only product IBM sells for iSeries application development. See the previous article from Phil Coulthard on this subject (www.ignite400.org). Over time, you can expect all of the CODE/400 functionality to be redeveloped as Eclipse-based plugins. Further, since WDT also includes the "classic" versions of WebSphere Studio and VisualAge for Java, you can well imagine these will be replaced with their new Eclipse-based versions. Interestingly enough, the exciting new WebFacing tool inside WDT today is already Eclipse-based. It was one of the first tools from IBM to ship based on this new platform. WebFacing will be a big benefactor of a full Eclipse toolset, because it allows you to launch the new Java and Web tools for editing the WebFacing output directly from within the WebFacing project view. Today, you must exit WebFacing to use WebSphere Studio or VisualAge for Java for this task.

The next release of WDT will mark the first phase of this massive effort to recreate all the WDT tooling as Eclipse-based, and will include Eclipse as the core IDE into which the new tooling will be hosted. This is the phase IBM has been talking about since it announced WDS in June 2000. This implementation is the realization of the compelling vision of a single IDE, fully open, fully extendible and fully standards-compliant. As is true for PDM today, there will soon be another single, extendible, tool-integration point that all iSeries programmers will have.

Beyond the basic tooling for iSeries languages, there will also need to be support for team-oriented iSeries projects. Further, there will need to be support for the leading iSeries change-management products as options for the team repository for both iSeries projects and all other projects. IBM has engaged the vendors of these products and all are onboard to support Eclipse.

Eclipse and Business Partners

For iSeries business partners and independent software vendors, Eclipse offers many opportunities. It also offers challenges for those who do not buy in. From a technical point of view, by writing Eclipse plugins, vendors can avoid reinventing all the infrastructure and functions that application development tools commonly need. Not only will vendors be able to exploit the core Eclipse extension points for writing plugins; they will also have extension points that are unique to iSeries to make it easier to access OS/400 artifacts and APIs. Think of extension points as application programming interfaces (APIs) that can be exploited to add function to the Eclipse IDE. We will talk technical in a future article. This all means faster time to market and reduced cost of development. That market itself will be compelling too, as every iSeries programmer is destined to have Eclipse on their desktop. Consultants and educa-

tion partners will also find a new vibrant world of opportunity teaching and preaching Eclipse usage and development, as well the programming language for Eclipse plugins -- Java. Like the heady early days of AS/400, there will be much excitement and activity, all centered around a common base. This is a retreat from the recent years of vendors fragmenting the AD market by each creating their own full proprietary tool stacks. These world-unto-themselves tool stacks are costly to produce, hard to sell, and hard for customers to learn. Augmenting these tool stacks with other vendor's tools is difficult, as is migrating away from them should that become necessary. In IBM's vision, the world of completely separate, proprietary and non-integrating tools is nearing its end. It will be replaced by a world of plug and play tools. This is precisely what developers and software vendors alike have been requesting. Many iSeries tool vendors are already enabling themselves to work with Eclipse. Many more will follow. Before you buy your next AD tool, you would be wise to ask the vendor about their plans for Eclipse.

Business partners looking for information on Eclipse enablement can start at the IBM PartnerWorld Web site: http://www.developer.ibm/welcome/wstools/wstools/ready.mtml. This site will contain more information as time goes on, so keep checking it.

Summary

In time, as more and more third-party iSeries tools become available as Eclipse-based plugins, you, the developer, will be able to choose from a rich variety of tools. All of these tools will augment, rather than replace, IBM-supplied tools. Further, they will look and feel perfectly consistent with the IBM-supplied tools, offering a reduced learning curve. This truly will be the new 21st century equivalent to the phenomenally successful PDM days of last century. Eclipse provides the new single point of integration for all IBM and IBM business partner tools. Like PDM, it will also be extendible by you, either by simple user-defined actions or more complex, but tighter integrated, Java-authored plugins. However, unlike PDM, you will also be able to purchase and use all the tools from non-iSeries vendors as well, for your non-QSYS development activities. Another key benefit will be the ability to host all of your application source, be it traditional RPG business logic or new e-business Web applications, in a single change management repository of your choice, running on the server of your choice (including iSeries). In fact, you will also be able to choose that repository and server on an Eclipse project-by-project basis, with seamless integration in the client Eclipse IDE.

The future of application development on iSeries is looking very exciting. T G

George Farr *and* ***Phil Coulthard*** *are co-authors of the Midrange Computing books Java for RPG Programmers and Java for S/390 and AS/400 COBOL Programmers. George can be reached at farr@ca.ibm.com, and Phil can be reached at coulthar@ca.ibm.com.*