

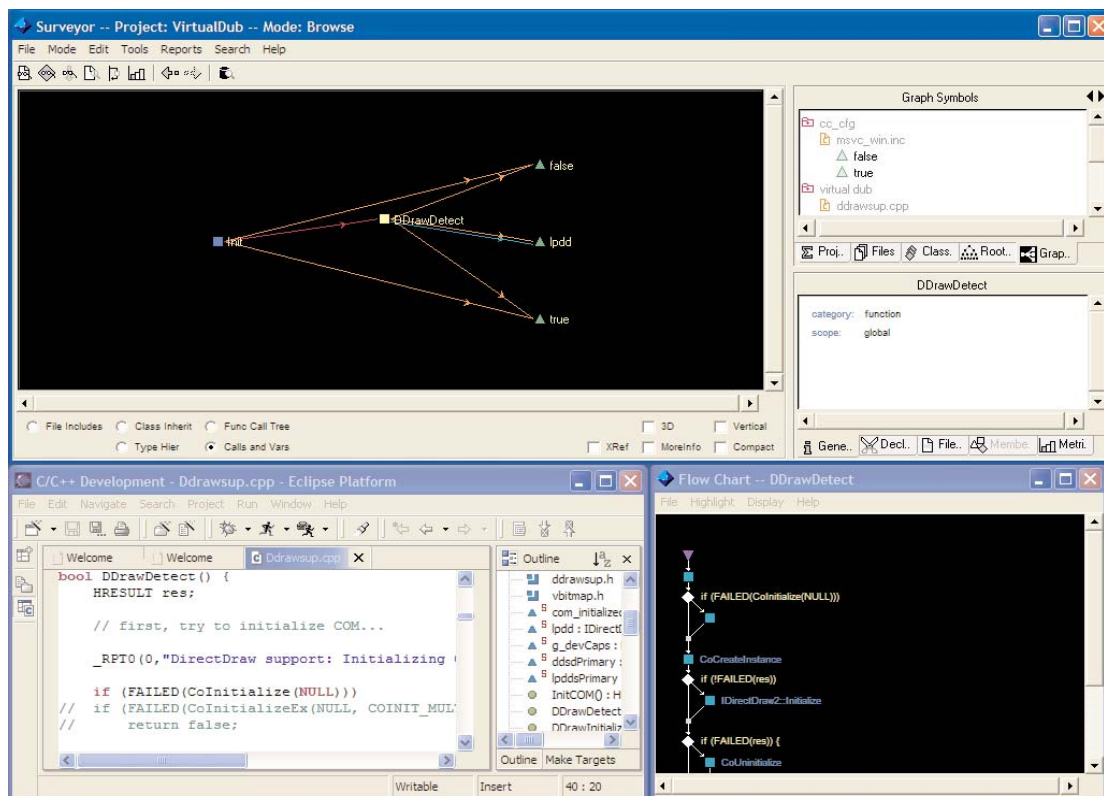
Using Surveyor with Eclipse

Eclipse and Surveyor

The Eclipse Platform marks a new breed of IDEs. Designed to be platform, tool, and language independent, it provides tools with an architecture organized by mechanisms and rules that lead to seamlessly integrated tools. In a certain way, Eclipse can be viewed as an empty box, waiting to be filled with a rich diversity of tools that will provide significant value to users.

Lexient's Surveyor tool is a rich, contextual interface and analysis tool for source code. It has been designed to be platform and language independent, able to work with a variety of IDEs. Surveyor's strength is in providing users and managers with a high comfort level in the code that they've developed by providing development teams with a better means of analyzing, visualizing and interacting with source code. Surveyor is the type of tool that Eclipse was developed for.

Introduction to Surveyor



What is Surveyor?

Surveyor is a tool for software development teams that lowers the overall costs associated with application extension and maintenance activities. It does this by streamlining the development cycle, automating the code analysis process, and unifying development teams.

Teams enjoy these benefits by using Surveyor to:

- Examine and interact with code elements' structure, interactions, and relationships
- Visually navigate code elements just as one can browse web pages
- Review code metrics and other analytical information
- Automatically document code

As the above screenshot of Surveyor being used within the Eclipse Platform shows, Surveyor provides a visual, interactive interface to source code, enabling developers to see the 'big picture' of the structure, interactions, and relationships between various applications or within a single application, between inter-related source code elements. Surveyor also allows developers to narrow their focus to fully analyze particular statements, variables, and interactions within the source. Surveyor also provides fully automated documentation and code quality metrics.

How does Surveyor Work?

Surveyor captures, analyzes, and builds an enterprise-wide library of meta-data for a development project. Once built, team members can use Surveyor's intuitive interface to examine, traverse, and manipulate source code in order to help them work together to accomplish design, analysis and coding tasks.

First, Surveyor captures, or parses, source code just as a compiler does. In fact the parsing method used is designed to closely emulate the compiler used by a particular development organization and language. Like a compiler Surveyor's parse conducts a full semantic parse of 100% of the source files, analyzing all symbols present in the code.

Today's development organization tends to support applications written in multiple languages. Acknowledging this situation, Surveyor was designed to be language independent, and will be extended to support all commonly used languages. Currently, Surveyor supports:

- C/C++
- Java
- Javascript/Jscript
- Visual Basic
- VB Script
- PL/SQL
- Tcl
- Python
- numerous versions of COBOL
- FORTRAN
- HTML/XML
- PowerPC Assembler

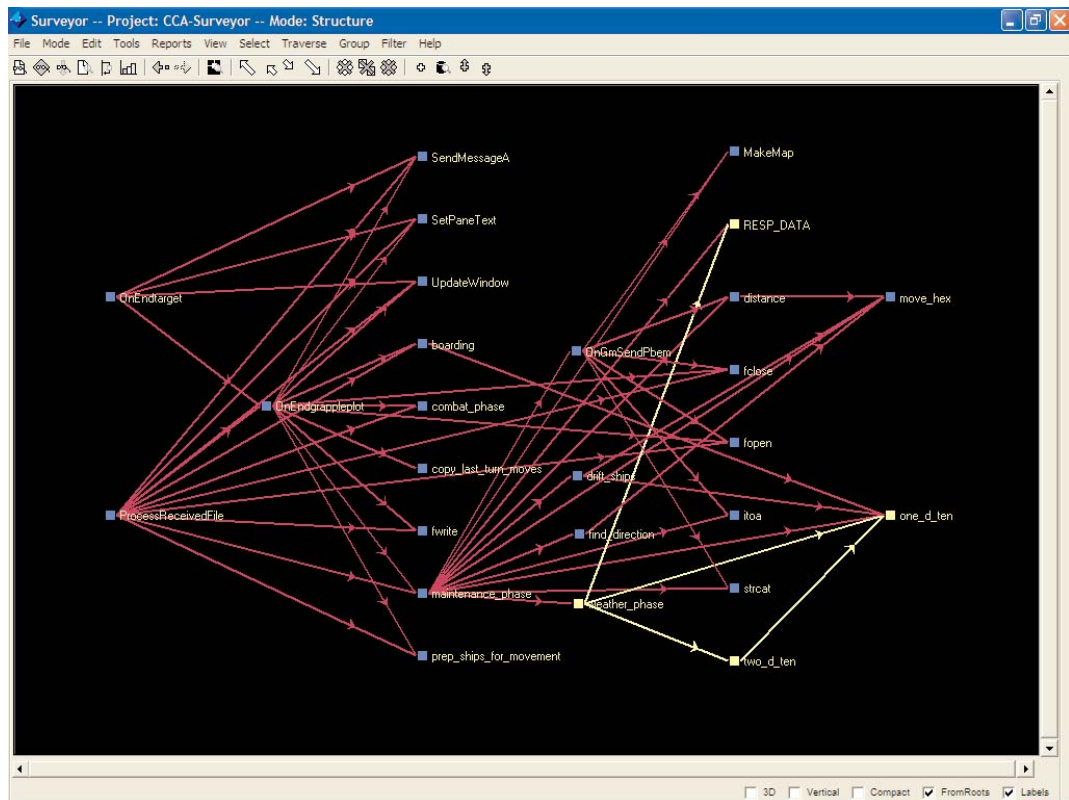
Developers use Surveyor to make important decisions regarding the direction of their future development efforts. Since this decision-making is dependant upon a thorough analysis of 100% of an enterprise's source code, extensive work has been done to ensure the quality of Surveyor's parsing engine. The robustness of Surveyor's C/C++ parser, for example, can be shown by examining the elements supported, including:

- Non-ANSI C
- Inline Assembler
- Preprocessor Directives

After capture, a thorough analysis is conducted on the resulting 'marked up' source code. This analysis builds a set of meta-data on the source, containing metric, structure, interaction, and relationship infor-

mation for all of the analyzed symbols in the development project. The goal of the analysis process is to enable users to:

- Clearly and easily understand complex existing source code
- Determine the intent of the application's originators
- Receive clear impact analysis information before, during and after code changes
- Provide overview metrics and documentation to automate code reviews and inspections



Surveyor is used to build an understanding of the complicated interactions between various source code elements

Once code is fully read in and analyzed, an object-oriented entity-relationship-attribute database is created. The XML compliant database contains information on all of the symbols in the project, ranging from primitive data types, such as `int`, all the way up to files and directories.

Changes to the source code can be inserted into the database in a variety of ways, depending upon an organization's needs. Changes to source code at the local and enterprise level can also be incorporated, in order maintain an appropriate level of synchronicity between the source code in the project, and the Surveyor Library.

Once the Surveyor Library is built and on-line, information about the source code in a project is available through a rich and intuitive interface. The interface is designed to accommodate the variety of different ways in which developers are accustomed to interacting with source code within their own particular tools. The display can be fully customized to suit a particular perspective and mode that developers are most comfortable using.

Source code elements of interest within the library can be selected and viewed in a number of different ways:

- Structural, interactions, relationships
- Flow charts of functions, procedures, & programs
- Metric information
- Source code text (via integration with Eclipse)

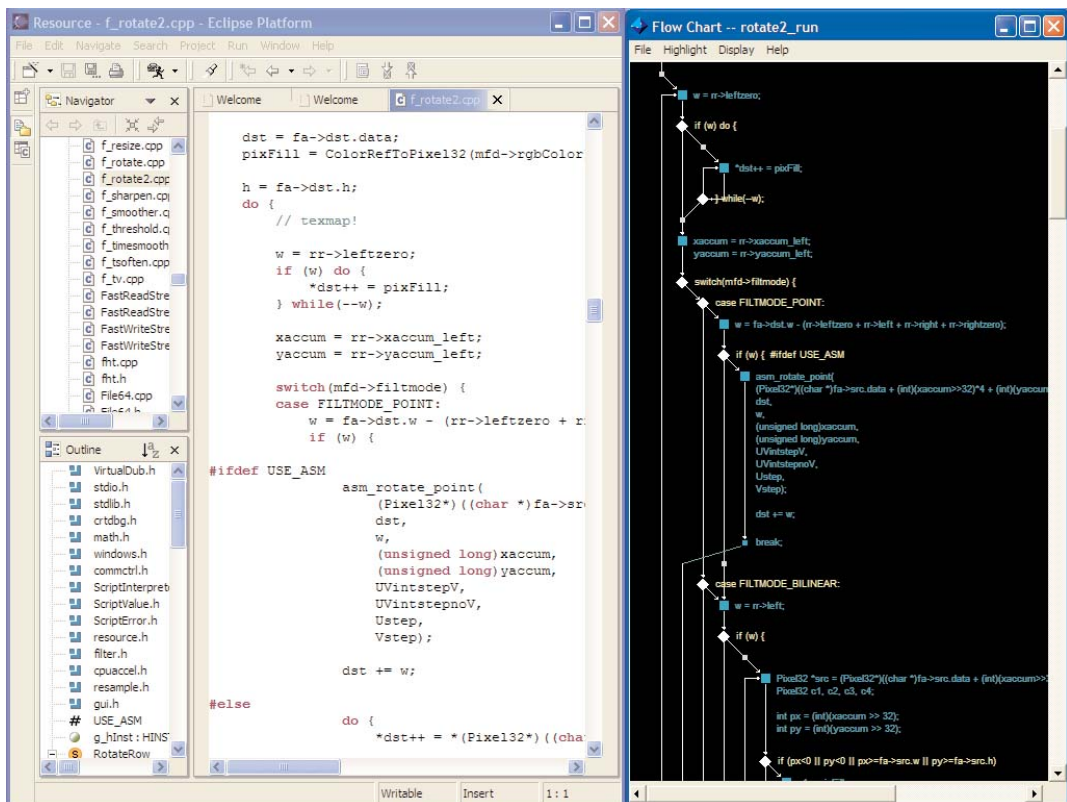
In order to organize and better manage the natural complexity present in source code, each of the symbols present in the library (such as files, classes, functions, and variables), may be interacted with in many ways, for example:

- Grouped in any way
- Filtered through structured queries, configurations, or completely customized
- Bookmarked

The depth and breadth of the ways in which users can visually interact with their source code elements provides developers with a new level of understanding of an existing body of source code. The richness of the interaction in turn facilitates comprehension and speeds development activities.

Capturing Original Intent

Surveyor puts the original author's knowledge in the hands of an entire development organization quickly and easily. The original intent of what the author created is embodied in the source code that they



Surveyor being used to show a side by side view of source in text and flow-chart mode

wrote. This intent is captured through the meta-data creation process, and is communicated back to the user in a variety of ways, including the following:

- Flow-charting of functions, procedures & programs shows the logic paths that the original author took in the code
- Containment of member functions and other elements within classes, files and machines
- Call-diagramming shows the critical interactions between various source code elements
- Viewing the author's comments as a whole, or integrated within the code and the flow-chart views provides a level of hyper-understanding than was previously unavailable

Surveyor's functionality highlights the original intent of the author in a much faster and more accessible manner than was previously available to development teams. Making this information available for any project 'on demand' after Surveyor's automated analysis process has been shown to lead directly to reducing overall development expenses.

Unifying the Development Team

Surveyor provides a development team with a common rich interface to their code resources. The interface has been constructed in such a way that both designers and coders, senior and junior members of a team can interact with code resources in a way which facilitates collaboration between team members. Marathon 'white-boarding' sessions to achieve coordination between stake-holders can be eliminated with Surveyor. No matter how complicated the spaghetti code, Surveyor's capture and analysis facility automatically creates whiteboard diagrams for teams in an interactive, realtime environment, right on everyone's desktops. Coordinating activities between team members is eased since all team members are able to seamlessly share code visualizations and organizations. Partitioning, containment, review, and assignment activities are simplified due to the common visualization between various constituents.

Streamlining the Development Process

Most development organizations respond to changing business requirements by initiating an iterative cycle of analysis, design, coding and testing. Surveyor enables several of these steps to be conducted at the same time, and by the same personnel, which speeds both the quality of and the rate at which this code is able to be produced. Thus, designers and coders can work more closely together, thereby preventing miscommunications, improper implementations, and inefficient executions of designs. By providing a forum which enables the analysis, design, and coding process to be streamlined, team members are able to work more closely together, speeding accomplishment of project milestones. Integration with Current Tools Surveyor was designed with an open structure to allow for integration with other team-critical development tools. Not only can developers use Eclipse (or the IDE or editor of their choice), which is synchronized with Surveyor's various presentations, but Surveyor also provides a punch-out of information to major Source Code Management tools. This ensures that teams can get the benefits of Surveyor while easily giving management the progress information they need within the prevailing processes and tools. Lexient is committed to constantly integrating additional tools across the entire developer toolset currently available in the marketplace.

Perspective and Focus

One of the greatest challenges facing the typical developer is to maintain a sense of perspective. They must be able to see the big picture, while paying due attention to little details. In a single interface, Surveyor provides this needed sense of perspective. While maintaining a sense of the relationship to the whole structure of an application, developers can use Surveyor to narrow their focus down to examine the text or logic paths within a particular function, or broaden their focus to examine global interactions and relationships. This facilitates quicker understanding of complex code, and eases design and coding tasks.

Multi-Language Development

The typical development organization, team, and developer uses multiple languages to write source

code. The most typical reason given is related to performing particular tasks best suited in diverse languages - performance and development timelines may be adversely impacted by choosing a sub-optimal language. As a result, the typical team uses half a dozen languages.

Given that most IDEs support a single language, until the advent of Eclipse, developers were using many different editors to complete projects. Eclipse, coupled with Surveyor, enables developers to rapidly and thoroughly complete project deliverables.

Software Development Done Better

Surveyor adds a powerful tool to Eclipse, and an organization's development team by enhancing communication between team members, providing an automated analysis process, and streamlining the various iterative steps in the development cycle. By becoming a part of an organization's current development process, Surveyor delivers on the goal of increasing the speed and quality of completed software, and decreases overall software development costs.

About Lexient

Lexient is a creator of graphical software development tools. Lexient's Surveyor product provides the software development community with a means of significantly reducing the costs associated with software development activities, by providing automated analysis, audit, and documentation capabilities. Surveyor paves the way for developers to write higher quality software faster. Surveyor supports source code written in many of the most frequently used programming languages, including Java, COBOL, and C++. Visit <http://www.LexientCorp.com> for more information.

About Eclipse

Eclipse is an open source ecosystem of royalty-free technology and a universal platform for tools integration. Eclipse-based tools give developers freedom of choice in a multi-language, multi-platform, multi-vendor supported environment. Eclipse delivers a plug-in based framework that makes it easier to create, integrate and use software tools, saving time and money. By collaborating and sharing core integration technology, tool producers can concentrate on their areas of expertise and the creation of new development technology. The Eclipse Platform is written in the Java™ language, and comes with extensive plug-in construction toolkits and examples. It has already been deployed on a range of development workstations including Linux®, MacOS®, QNX® and Windows® based systems. Full details of the Eclipse community and white papers documenting the design of the Eclipse Platform are available at <http://www.eclipse.org>.