



NaturalONE Productivity Study

Application Development & Maintenance Tools under Eclipse to Increase Developer Productivity

Karlheinz Kronauer
Software AG

November 2010

CONTENTS

OVERVIEW	3
GENERATE A CONTINUED RETURN ON YOUR NATURAL ASSETS	3
INCREASING APPLICATION DEVELOPMENT PRODUCTIVITY	4
INCREASING APPLICATION MAINTENANCE PRODUCTIVITY WITH NATURAL ENGINEER	11
ADDITIONAL APPLICATIONS OF NATURAL ENGINEER	18
CONCLUSION	22

TABLES & FIGURES

TABLE 1: QUALITATIVE BENEFITS OF NATURALONE APPLICATION DEVELOPMENT TOOLS	7
TABLE 2: POTENTIAL PRODUCTIVITY GAINS BY USING NATURALONE APPLICATION DEVELOPMENT TOOLS	8
FIGURE 1: CODE COMPLETION	10
FIGURE 2: DATA BROWSER	10
FIGURE 3: DEBUGGER	11
FIGURE 4: SAMPLE EXECUTIVE SUMMARY REPORT	14
TABLE 3: POTENTIAL PRODUCTIVITY GAINS BY USING NATURAL ENGINEER – SELECTED ITEMS	20
TABLE 4: SUMMARY OF ADVANTAGES AND BENEFITS OF NATURAL ENGINEER	21

OVERVIEW

- New technologies.
- New business practices.
- New application requirements.
- New standards.
- New development environments and hardware platforms.
- New legal requirements.

All of these changes frequently force modifications or adjustments to your production applications and staffing requirements. With this barrage of constant change, how do you effectively enhance and maintain your strategic Natural applications?

These long-lived Natural applications are the backbone for thousands of organizations because they execute business-critical transactions and manage the corresponding data. This makes it even more important that you effectively manage the challenges that arise from constant change.

GENERATE A CONTINUED RETURN ON YOUR NATURAL ASSETS

Successfully managing change by improving productivity and reducing development costs is important to providing your organization with a continued return on assets. You need the ability to:

- Fully leverage prior investments in mission critical Natural applications, for example, by leveraging the functionality of existing Natural systems within new, composite applications or extending Natural applications to Windows-based and Browser-based environments or implementing Web services successfully within Natural applications
- Respond more quickly to technological developments by incorporating new technologies and standards into your existing business-critical Natural systems
- Develop, deploy and maintain applications on a variety of platforms
- Improve the productivity of your application developers
- Manage fluctuating staffing requirements
- Improve the management of evolving application requirements
- Provide ongoing maintenance of mature systems that lack complete documentation

To help you achieve these goals, Software AG introduces the complete new application development platform NaturalONE and the productivity package, NaturalONE Standard Edition. Bundled together and packaged as a single offering, these tools offer you the means of effortlessly increasing your application development and maintenance productivity. NaturalONE and the standard edition package help you develop and maintain your Natural applications under Eclipse from a Windows or Linux environment, generating a positive return on investment and increasing your developer productivity by as much as 50 percent.

Besides the base product NaturalONE, which already integrates all relevant tools required for application development, we introduced a packaged edition of NaturalONE which includes additional development-oriented products. These two editions address different levels of needs:

The NaturalONE Productivity Study includes metrics for Natural Engineer along with NaturalONE.



Use NaturalONE to generate a positive return on investment and increase your developer productivity by as much as 50%.

NaturalONE – All Application Development Tools

NaturalONE includes a full Eclipse-based Natural IDE for developing and maintaining Natural applications. NaturalONE is available for Windows and Linux. It includes tools for web service generation, web application development, debugging and testing, documentation and deployment. With the Natural Development Server, developers perform remote development and maintenance of Natural applications on any platform - UNIX, Linux, OpenVMS Itanium and even the mainframe - directly from today's standard development environment - the Eclipse platform on the PC desktop.

NaturalONE Standard Edition Package – All Application Development Tools including remote Development

This edition includes NaturalONE, the Natural Development Server, Adabas for Windows and Entire Net-work Client. This allows for remote development for the most common server platforms, like the mainframe. Tools developers can quickly and securely maintain, enhance and re-engineer very large and long-lived business applications.

This paper will discuss the specific features and functionality of NaturalONE and the NaturalONE Standard Edition Package that will help you to:

- Boost developer productivity while increasing the quality of your Natural code
- Support undocumented, unstructured code in a production environment
- Automate maintenance tasks
- Apply standards to your application code
- Enable non-mainframe programmers to develop and maintain mainframe applications through a modern Eclipse-based environment
- Efficiently support the conversion of existing applications to new AJAX-based Web applications

INCREASING APPLICATION DEVELOPMENT PRODUCTIVITY

Improve developer productivity with all of your Natural applications – regardless of the platform they run on – through a single Eclipse-based development environment. By centralizing development for multiple platforms, integrating various tools under the same environment, and providing simple, familiar, graphical user interfaces to your application developers, you not only accrue productivity gains but extensive cost savings.

The ability to develop for mixed platforms from a centralized environment, for example, helps control costs by allowing everyone to work under Eclipse on Windows or Linux machines, while deploying and maintaining applications on the most appropriate platform(s).

In addition, the NaturalONE Application Development Tools help you easily attract and train new developers who expect to work within the familiar Eclipse environment and with familiar tools and menus and functional capabilities. These productivity, training and retention benefits all translate to more satisfied application developers and higher-quality code.

Ultimately, the costs related to software development, maintenance and administration are significantly reduced by all these factors.

Key Features of the NaturalONE Application Development Tools

Both NaturalONE and the NaturalONE Standard Edition provide a state-of-the-art development platform for building, testing and maintaining all Natural applications throughout their lifecycle and across all supported platforms. Key features of the product that help drive development productivity gains include:

NaturalONE's Eclipse-based IDE

Even for experienced Natural developers, NaturalONE can help accelerate delivery of new Web-based applications by an estimated 30 percent. NaturalONE includes automatic, interactive testing and debugging tools and allows developers to write, test and execute programs locally. Using the outline and the dependency view, the developer can ramp up the development speed especially for large programs. NaturalONE also provides a team-support capability; developers have all tools in a single local environment for working independently while sharing/securing project source code in a team repository. At the same time, NaturalONE retains Natural's core strength of simplicity in coding and rapid development of applications.

Experience Unprecedented Ease-of-use

Within one week of NaturalONE training, skilled developers with no prior Natural experience will be able to modernize Natural applications in two ways. First, they will be able to use NaturalONE to intuitively expose existing Natural programs (or objects) as Web services for consumption in an SOA. Second, they will be able to use NaturalONE to create attractive AJAX-based user interfaces for applications that currently are accessed through "green screens."

NaturalONE will appeal to Java programmers because its Eclipse tools and development concepts are very similar. NaturalONE's standards-based platform also makes it possible to efficiently cross-train personnel and to deploy resources where they are most needed and at a moment's notice.

Simplify Service Development for SOA and BPM

These days, many companies have mandated that all new development must support SOA and/or BPM. Businesses must become more agile, minimize costs and increase IT asset effectiveness through speed and re-use. With NaturalONE, you'll achieve these aims with one-stop Web service development and a direct interface to an SOA governance tool—CentraSite—for ease of control.

NaturalONE is also integrated with Software AG's webMethods suite for easy development, support and deployment in a comprehensive SOA or BPM environment. You can easily leverage existing Natural business logic to create Web services. Additionally, any service available inside or outside your organization can be consumed as a NaturalONE object.

Easily Deploy Rich Web Interfaces

Organizations still using "green screen" applications are looking to quickly convert and deploy Web applications. That's because today's users demand browser-based interfaces and Internet access from applications. NaturalONE makes it easy to convert existing Natural screens into rich Web interfaces using AJAX-based technology. Wizard-driven Web-interface creation tools are integrated in Eclipse (just like a Natural editor) to help you quickly deliver new Web-based applications. Additionally, a single development team/developer can write user interface and back-end business logic. Count on NaturalONE for a straightforward approach to transforming existing map-based Natural applications into rich Internet applications with the benefit of increased end-user acceptance and productivity. Best of all, no changes to Natural applications are required!



NaturalONE uses AJAX to easily create powerful Web-based applications.

Standardize on a Single Platform

Looking to streamline application development? With NaturalONE, you can enhance existing Natural applications—faster—to meet changing business requirements and reduce application costs. NaturalONE offers the advantage of an IDE based on Eclipse. This facilitates collaboration between Natural and Java developers and expands the base of in-house Natural skills. New developers can easily learn Natural using Eclipse.

Speed up Natural Application Maintenance

If you have very large Natural applications to maintain, your development staff likely spends a lot of time on maintenance and not enough on new development. With NaturalONE, a Java developer familiar with Eclipse can easily learn Natural and step in to fill the gap. NaturalONE provides all maintenance and development tools in a single environment including automatic, interactive testing, debugging and restructuring. You can reduce the time you spend maintaining and enhancing Natural applications by up to 50 percent and spend 70 percent less time managing the application lifecycle.

Development in Teams

NaturalONE supports a variety of methodologies working in development teams. The recommended approach is to use a source code repository, like CVS or Subversion. This repository can be used to synchronize changes to the source code in a time or for many teams. Once completed with development tasks, the changed source code can be deployed to any Natural Server using the NaturalONE deployment tools for testing or production. For testing, NaturalONE provides features whereby developers in a team can test their own code on a remote test server without conflicting with each other.

Productivity Advantages with NaturalONE

- Single view under Eclipse to all objects required for application development – source code, data structures, debugging information, cross-references, and documentation
- Instant parser and error handling
- Editor productivity tools like content assist, searching and object linking
- Simplified development and maintenance
- More organized Natural development steps
- Familiar Eclipse IDE look and feel
- Tools for Web (AJAX) development included
- One Click generation tools for Web services
- Subprogram and Unit test tools included
- Common versioning of all development assets in standard versioning tool (CVS, Subversion)
- Active support of collaborative team work
- Work area for offshore and remote development

Benefits of NaturalONE

- Reduced training costs
- Access to larger pool of development resources because of more familiar environment and tools
- Lower costs for software development, maintenance, and administration
- Greater acceptance and job satisfaction for application developers



*Download the FREE NaturalONE
Community edition at:
www.naturalone.com.*

Components of NaturalONE

- Natural IDE for Eclipse, combined with a Natural Runtime for program execution
- EntireX developer tools for service generation
- Natural for AJAX developer tools for developing and debugging Natural Web applications
- Natural debugger
- Data Browser for browsing data files in Adabas and RDBMs
- CRUD (Create, Read, Update and Delete code generation tool)
- Optional remote Plug-Ins for Natural Engineer (code conversion to and AJAX application) and Predict (documentation and generation)

In addition the NaturalONE Standard Edition Package includes:

- The Natural Development Server for the target platform – Mainframe, UNIX, Linux, Windows or OpenVMS Itanium
- Adabas for Windows or Linux for Local testing database access
- Entire Net-work Client, for Remote Adabas database access testing

Productivity Gains You May Experience from NaturalONE & the NaturalONE Standard Edition

Please refer to Table 1 for a summary of the qualitative benefits provided by Software AG's NaturalONE. For a summary of quantitative benefits, please refer to Table 2. The information provided should be used as a guideline only; your specific experiences may vary. In addition, all of these improved development and maintenance tasks may result in an average 20-50 percent savings of CPU time on the mainframe

Table 1: Qualitative Benefits of NaturalONE Application Development Tools

Key Features	Key Advantages	Key Benefits/Results
<ul style="list-style-type: none"> • Standardized, centralized development environment on Eclipse • Supports collaborative development in teams • Familiar look and feel and ease of use • Attract developers coming from college • Support of off- shore development activities 	<ul style="list-style-type: none"> • Keep track of all development objects in separate views • Keep track of complete source code (expand, collapse, coloring of source code, content assist) • Get documentation close to development • Review cross references to optimize application structure • Gain better debugging capabilities • Obtain better organization capabilities • Better synchronization of team work • Gain higher developer retention • Get access to a larger pool of development skills • Provide an attractive modern development environment 	<ul style="list-style-type: none"> • Better structured source code • Better documented applications • Improved application design • Fewer errors in source code • Higher overall quality in your development code

Table 2: Potential Productivity Gains by using NaturalONE Application Development Tools

*Note: These productivity measurements resulted from comparing steps required on the mainframe with steps required under NaturalONE, as well as considering experiences from existing customers and Software AG Global Consulting Services projects.

Feature	Functionality	Average Productivity Gain %**
Navigator view	<ul style="list-style-type: none"> · Structured information in with object type extension instead of flat library view · Decorators showing object status (e.g. source changed) · Library can be structured according to object type · Libraries can be structured in folders, sub-folders · All assets – also non Natural assets – in one view · Eclipse terms are used (->project) · All libraries – one view · Direct access to each single object · Copy and paste (vs. Sysmain) · Personal library views using filter · Commands with right hand mouse click · Multiple selection of objects possible · Direct check-in (commit), check-out to repository (CVS, Subversion) 	75%
Program Editor	<ul style="list-style-type: none"> · Parser working instantly and reports errors directly (vs. Check command) · Copy and paste instead of “.Commands” · Scrolling through source text · Comment, “De-Comment” of source lines · Syntax coloring · Content assist – Code completion · Code templates · Open external object by selection/linking in source code · Follow references of variable through the source code 	40%
Layout Painter	<ul style="list-style-type: none"> · Visual design of Web pages · Many controls available (Grids, tables) · Preview of web page · Direct inclusion of PDF generation or printing 	25%
Map Editor	<ul style="list-style-type: none"> · Visual map design · No “.and .. Commands”, just click, drag and drop · Move fields on the map using mouse · Easy and fast definition of data elements · Fast access of field via Outline view 	30%
Documentation Tools	<ul style="list-style-type: none"> · NatDoc similar to javaDoc · Generates documentation · Layout customizable using templates 	75%
Data Browser Tools	<ul style="list-style-type: none"> · Browsing Adabas and SQL databases · Sorting Columns for different criteria · Save results in files (e.g., CSV) 	50%
Testing Tools	<ul style="list-style-type: none"> · Direct test of subprograms · Regression tests/unit tests can be defined · Unit tests can be executed via Ant script 	75%

Feature	Functionality	Average Productivity Gain %*
Cross References	<ul style="list-style-type: none"> • Separated in Dependency view • Object can be read in editor by clicking on dependency • Dependency view directly updated if code change (vs. Catalog/Stow) 	50%
Offshore Development Support	<ul style="list-style-type: none"> • Natural Runtime Integrated • Programs can be executed, tested and debugged without remote access • Remote access for deployment and consolidation of source code 	30%
Predict Documentation	<ul style="list-style-type: none"> • Wizard leads through generation process • Generation process simplified – not all steps necessary • Fast selection of files/fields – scrolling instead of PF Keys / + - • No Sysdic Logon 	30%
Predict Generation	<ul style="list-style-type: none"> • Visual map design • No “.and .. Commands”, just click, drag and drop • Move fields on the map using mouse • Easy and fast definition of data elements • Fast access of field via Outline view 	20%
Debugger	<ul style="list-style-type: none"> • Comprehensive functionality; easy and fast to use • Interactive walk-through programs • All information on one screen • Interactive run-time changes of values • Debugging of “Map” and “Web” applications possible 	50%
Utilities	<ul style="list-style-type: none"> • Easy and fast selection of data using browse buttons • User friendly handling • Error handling included in Navigator view 	20%
Find and Search Functions	<ul style="list-style-type: none"> • Search does look in all objects, not just in Natural code • Many options for Search available : Pattern search, regular expressions, wild card and exclusions 	Between 50-90%

Figure 1: Code Completion

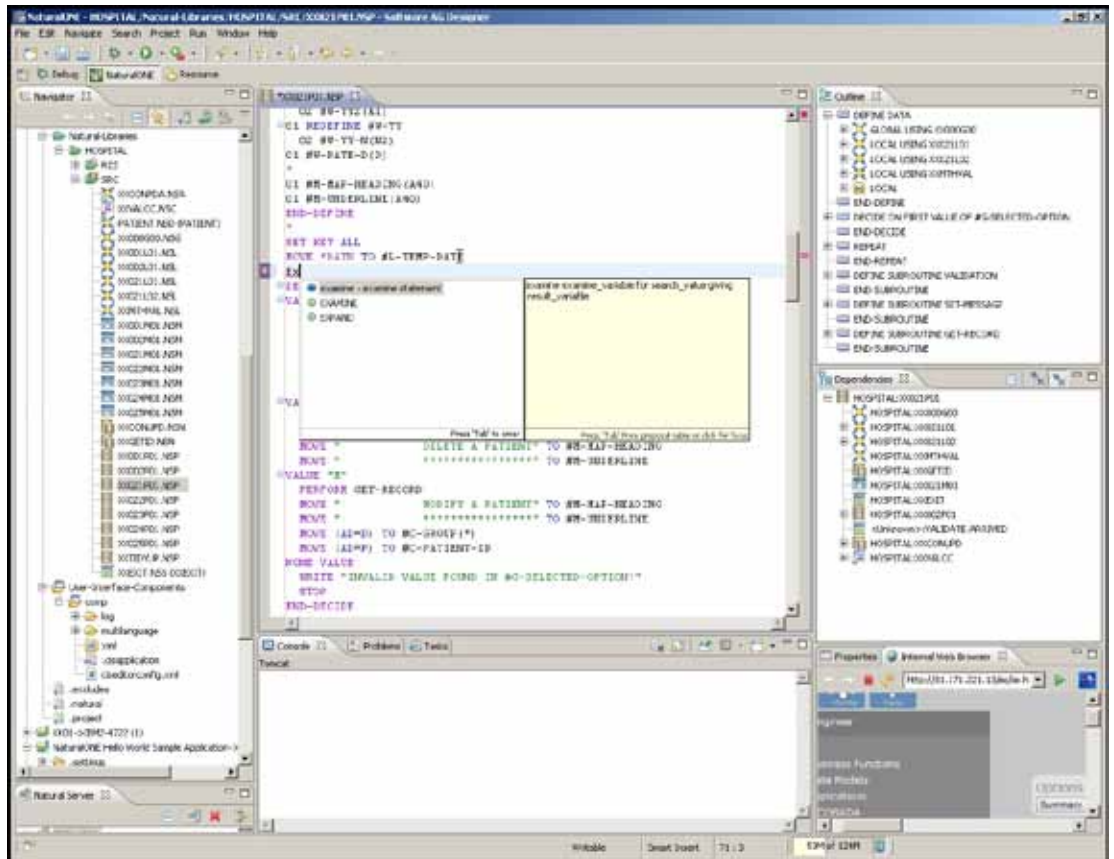


Figure 2: Data Browser

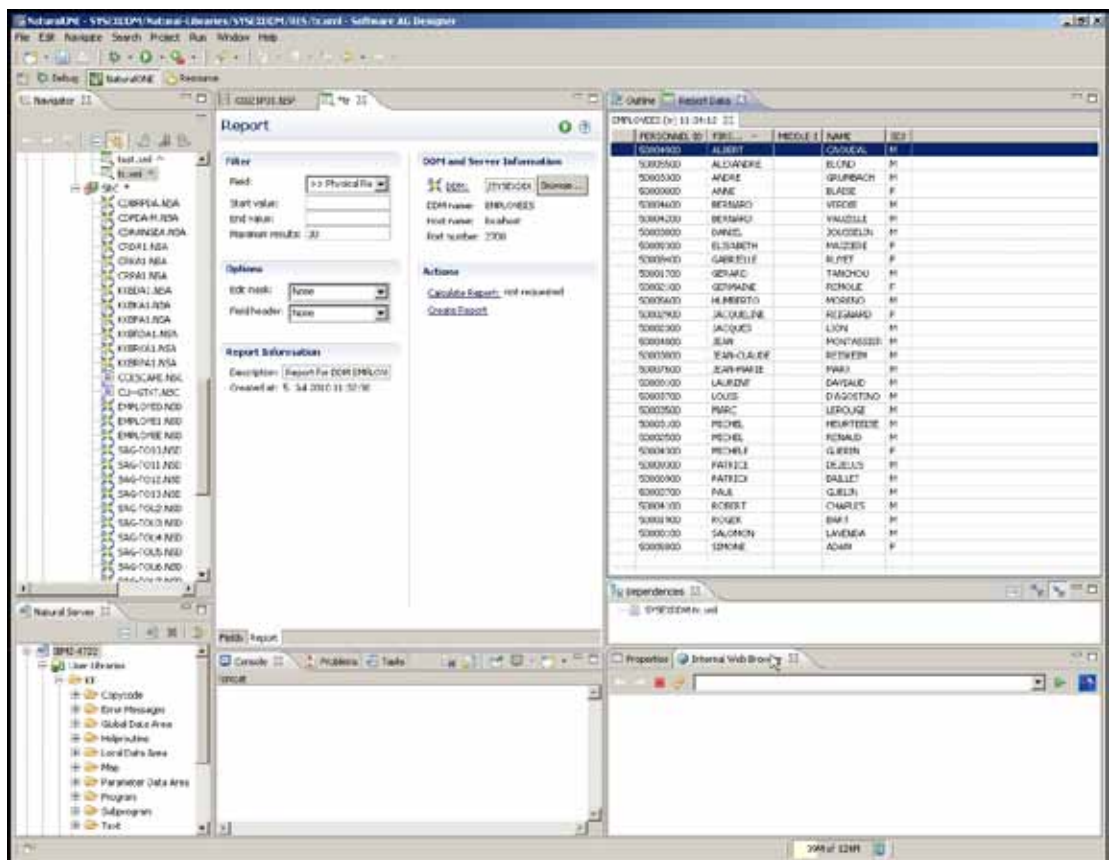
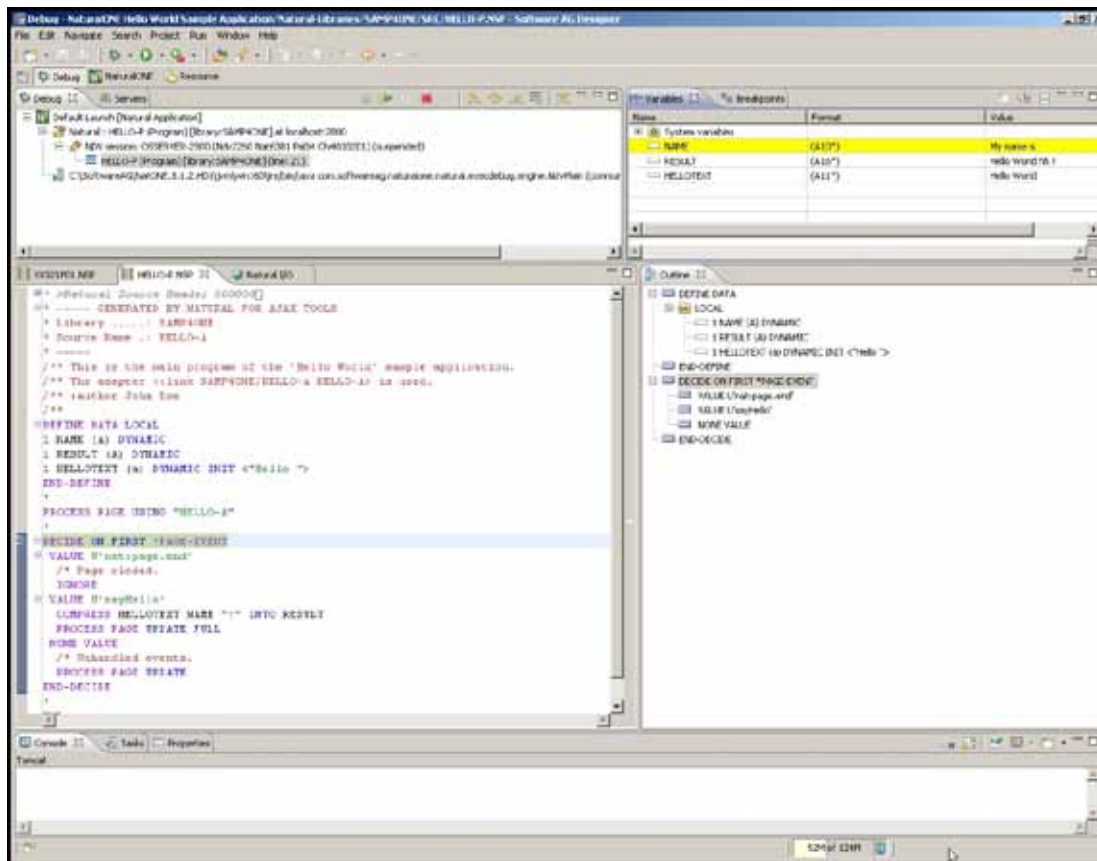


Figure 3: Debugger



INCREASING APPLICATION MAINTENANCE PRODUCTIVITY WITH NATURAL ENGINEER

Ongoing software maintenance is a fixed cost in most IT budgets today. In fact, maintenance costs often absorb more funding than the development of new systems. And, it is extremely costly and difficult to estimate the impact of required maintenance or reengineering changes to an application, especially if documentation of the application source code is flawed, not up-to-date or non-existent. Therefore, it is very important to provide your development teams with the ability to more easily maintain or re-engineer your Natural applications to improve productivity and decrease maintenance costs.

With versatile code analysis and restructuring tools, developers quickly and securely maintain and re-engineer very large and long-lived mission-critical business applications. These kinds of tools help you accurately identify the full impact of modifications. And developers are able to modularize the code in order to expose your Natural application as a service to Web interfaces or other applications.



Use Natural Engineer to speed up and simplify software maintenance. You can reduce maintenance costs and free up resources to develop new applications.

A key Add-On for Software AG's NaturalONE is Natural Engineer. It is a multi-purpose, multi-functional maintenance and re-engineering tool for Natural applications. It comprises powerful core functions with a range of options for special purposes, such as Natural version migration or application restructuring. With Natural Engineer you:

- More effectively manage maintenance and re-engineering projects
- Restructure Natural applications more easily to meet new user requirements or delivery channels
- Migrate more quickly to a new version of Natural or a different platform
- Enhance software maintenance productivity
- Quantify the cost of major maintenance projects
- Assess the impact of proposed changes across all applications

With Natural Engineer, your IT management team establishes better control of all aspects of application change and associated costs.

Utilizing Natural Engineer, you speed up and simplify your software maintenance process, cutting costs and releasing scarce resources for the development of new applications.

In conjunction with Natural for Windows, Natural Engineer helps you exploit the power of the PC. Natural Engineer is invoked from the Natural Studio where graphical features and local processing power provide maximum effect while Natural source code is directly processed from the target environment. This allows developers to leverage an incremental approach to assessing the impact of required changes to Natural applications. Developers can "drill down" through application hierarchies and objects to identify the precise location and reason for a particular impact. They are no longer dependent on batch processes and the use of limited scanning and editing tools.

Following are additional ways you improve maintenance productivity using Natural Engineer:

Application of Coding Standards

The benefits of adopting strict standards for the design and coding of application systems are well-known. The objective is to display application code with a consistent appearance and use of constructs offered by the programming language. This ensures that even a developer who is new to the application can quickly understand the logic and processes used.

However, although standards manuals and similar documents abound, it is often difficult to enforce standards rigorously across the whole organization. Natural Engineer solves this problem by applying standards definitions to application source code automatically.

Typical specifications could include:

- Controls on program syntax
- Specification of minimum percentages of comments statements
- Fully-qualified database field names
- Use of reserved words
- Rules in maps

Integrated Structure Diagramming

An important way in which Natural Engineer enhances maintenance productivity is its integrated structure diagramming function. This offers a graphical representation of Natural objects and their relationships, making it exceptionally easy to understand the structure of an application.

Icons are used to identify different types of Natural objects in a structure diagram at a glance. Where a map object forms part of the structure, a special preview function is available which, at a single mouse click, displays the map as an application that the end user would see.

When working at a PC, it is easy to navigate through the structure of an application graphically and then click the mouse to bring up a window showing the source code of the selected object. The ability to switch rapidly between diagrams and the source will generate significant productivity improvements when analyzing code for required changes. And using a PC, it is easy to drill down through the application structure very rapidly, to whatever level of detail is required.

More Automated Approach

Natural Engineer provides powerful facilities for the automated modification of Natural source code. Changes are made in place of, before or after identified impacts. Solutions models are easily created in the form of Natural text members. Once a solution model has been created and tested, the change is automatically applied across the whole application. Where large volumes are involved, the Natural Engineer Scheduler feature is used to apply the modifications as a batch process.

Users can also easily customize Natural Engineer to their own environment. For example, on the PC, they view output using their browser of choice.

Identifying Complex Code

A major obstacle to maintenance productivity often not appreciated is complex program code. The complexity and effect of code in specific areas of an application can be difficult to understand and, therefore, modify. This is particularly true if application knowledge is limited, or contract development staffs are being used for maintenance projects.

Natural Engineer's Metrics feature solves this problem by applying standard industry complexity measurements, such as Halstead or McCabe, to Natural source code. The results are presented in report form or as a series of easily understood and compared graphics. With the information gained from the Metrics, developers having relevant knowledge concentrate on the most complex part of the application, leaving less experienced staff to handle the more routine areas of the system.

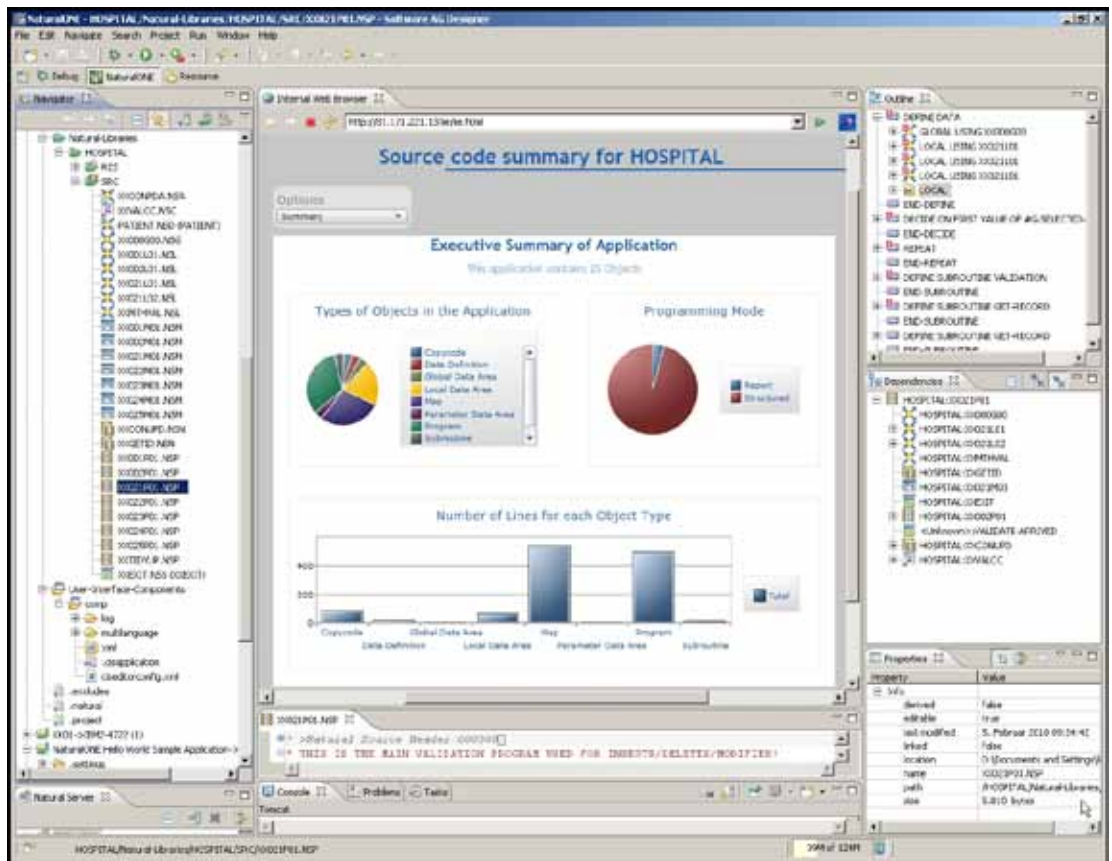
Complexity of program code becomes a significant issue when application maintenance outsourcing is under consideration. Having clearly documented details of program complexity helps you negotiate better contractual terms and lower costs in such situations.

Enhanced Documentation

Another important factor in the improvement of application maintenance and re-engineering quality is the availability of better and more comprehensible system documentation. Natural Engineer provides this at a number of different levels. Even at the basic level more than 30 standard reports are available. And, with Natural Engineer, all documentation is always available online and fully up-to-date from the repository.

Easily customize Natural Engineer to your environment. For example, you can view output using your browser of choice.

Figure 4: Sample Executive Summary Report



An important way to improve documentation is to introduce consistency in the way source code is presented. For this purpose, Natural Engineer lets you define templates to be applied to program source code at an installation, application or individual level.

It is obvious that the older a production systems, the more complex, and frequent the requirements for re-engineering maintenance work. It is also very common that the development staff that created the original system is no longer available and the modification work is carried out by staff (or potentially external contractors) not familiar with the system, The consistent presentation of program source code made possible by Natural Engineer is a considerable productivity boost in these situations.

Examples of standards that can be applied include:

- Level numbering
- Indentation spacing
- Use of acronyms instead of symbols for logical operators and vice versa
- Statement spacing
- Line joining and splitting

Natural Engineer also provides a Web interface using Microsoft Silverlight technology. Diagrams of application flow and structure or the internal flow of a program can be visualized leveraging the powerful Silverlight advantages like zooming. With flow diagrams, application developers not only navigate interactively through the diagram, they also view the corresponding source code at the same time.

Flow-charting capabilities offer a system-wide, graphical view of an application, an excellent complement to the detailed structure and code information delivered by Natural Engineer's structured diagramming feature.

Natural Engineer Features/Advantages/Benefits

Natural Engineer's core technology is comprised of a repository, a search engine, automatic modification facilities and auditing support. All additional functions, such as extended documentation, are based on this foundation.

Additionally, Natural Engineer is a Natural application and utilizes Adabas for its repository. As a result, it runs on a variety of platforms, including z/OS, VSE/ESA and BS2000/OSD and UNIX environments and with the latest Microsoft Windows releases on PCs. For code conversion from existing maps to web pages a natural Engineer Plug-In for NaturalONE is available. This operational flexibility means that you can use Natural Engineer in a totally mainframe environment, in a standalone PC configuration, or in a combination of the two, also using the Plug-In for NaturalONE.

Repository

The repository - a single, comprehensive source of all application information in the enterprise - is the basis of all Natural Engineer functions.

The initial step of creating the Natural Engineer repository immediately provides you with a number of benefits. An accurate, fully documented description of all current application code is available with complete, online cross-reference data. Also, the powerful Natural Engineer parser adds value to the basic program information stored in the repository, facilitating later impact analysis and modification.

Natural Engineer caters for the automatic re-synchronization of the repository data after code modification. Based on date/time stamp information in saved Natural source code, a re-synchronization process can be initiated at user-defined intervals. The information stored in the repository is then updated automatically based on differences detected in the date/time stamp.

Extract and Load

The Extract and Load feature retrieves source code from operational libraries and loads relevant information into the repository. A powerful parser verifies the existence of every object referenced and identifies redundant and missing objects. This ensures a complete inventory of all application code.

Where objects are flexibly linked, for instance when a subprogram is called using a variable with the sub-program name, a feature known as Soft Links allows the user to specify a list of valid object names for the call. The use of Soft Links means that the integrity of the cross-reference information is always maintained.

The Extract and Load processes can be run separately, if required, and for large volumes of code, the repository can be built up over time with a series of incremental load executions.

A single repository offers a comprehensive source of all application information in the enterprise. It's the basis of all Natural Engineer functions.



Choose from more than 30 standard reports. This allows you to retrieve precisely the required details and reduce redundant information.

Reporting Facilities

Comprehensive reporting facilities are provided. Natural Engineer Impact Analysis generates reports at a summary, object and detail level, and impacted source code is always highlighted. More than 30 standard reports are available, allowing you to retrieve precisely the required details and reducing redundant information to the absolute minimum.

Search Engine

The powerful Natural Engineer search engine enables you to quickly locate specific fields, files, statement types, etc., and report their frequency and location across all applications. It tracks impact chains across program and application boundaries to ensure that all potential impacts resulting from a required change are identified.

Impact Analysis

Natural Engineer provides a number of standard Impact Analyses with pre-defined search criteria, including:

- Identifying impacted code when planning a Natural conversion
- Application restructuring with the Object Builder feature
- Identifying Natural formats, constructs, etc. that could have a negative impact on performance
- Identifying candidate text and fields for internationalization

Natural Engineer Impact Analysis also allows detailed search criteria for a particular impact analysis to be created, and refined, very flexibly. A full range of partial and wild card functions can be added to the search criteria. Results from a specific impact analysis can be stored under different version numbers. Incremental Impact allows these results to be refined with additional criteria to produce just the information set required by your developers. Incremental Impact is carried out on the whole application, previously impacted objects or previously impacted statements. This gives maximum flexibility in precise identification of impacted code that can later be the subject of modification.

The comprehensive impact analysis functions of Natural Engineer give a full picture of what needs to be modified, across all applications and objects. This is essential to allow accurate estimation of the total impact and cost of a proposed change. A key benefit of tool-based impact analysis is the elimination of missed impacts. This is often a problem in reengineering projects involving complex code or high volumes when manual techniques or simple code scanning products are used. The cost of missed impacts is significant, if they are not detected until late in the test phase or after the application has been returned to production.

Program Modification Facilities

Natural Engineer makes it easy to apply individual changes and pre-coded solutions automatically. The program modification facilities, including a flexible replace mechanism, are simple to use and significantly increase developer productivity.

After modification, Natural Engineer places all modified code into a user-defined, or default, Modification Library.

The automated approach to modification provided by Natural Engineer eliminates the need to make repetitive manual changes, a process that is error-prone. It also reduces implementation time, so scarce development expertise concentrates on areas requiring a unique, manual solution.

Audit Log

All changes made to the program information stored in the repository are logged, with full details of what was changed, by whom and when. This provides a comprehensive audit trail that will meet the requirements of EDP auditors. Changes are reviewed online or as hardcopy reports.

The standard change reports provided cover a wide range of typical audit requests. This makes it easy to answer requirements such as: "Show all changes made to the payroll system last year," or, "Show all changes made by developer x during the past 6 months."

As well as meeting auditing requirements, the change logging function of Natural Engineer is used to give project managers the detailed information necessary to effectively oversee re-engineering and maintenance projects.

Scheduler

An option is provided to allow developers to manage and execute the Natural Engineer Scheduler for unattended operation of the Environment Set-Up, Impact Analysis and Modification functions.

With the Natural Engineer Scheduler you are able to select required functions for execution and specify a start date and time. The Scheduler takes control of the Natural Engineer session and provides a dialog that controls starting and stopping of the Scheduler's execution.

The benefit of this feature is that long-running impact analyses or load processes are scheduled for overnight execution when PC equipment is not heavily utilized.

Support for Construct and Predict Case

Natural Engineer was designed to support the requirements of user-developed Natural code. There are, however, facilities to support the use of Natural generators, i.e., Construct and Predict Case.

Natural Engineer "understands" and processes code produced by Natural Construct and Predict Case.

There are Construct-specific reports produced by all Natural Engineer processes and the tool differentiates between Construct-generated code and code in user exits. During the Modification phase, Natural Engineer will only allow changes to the user exit code, not the Construct-generated code. If you modify the Construct model code, and it is therefore marked as no longer able to be regenerated by Construct, Natural Engineer treats the code as a normal Natural object.

Similarly, for Predict Case users, Natural Engineer produces specific reports relating identified impacts to Predict Case System Components.

ADDITIONAL APPLICATIONS OF NATURAL ENGINEER

Application Restructuring

For many re-engineering tasks today, it is not sufficient to apply individual code changes. It is increasingly necessary to restructure the existing application to enable the essential business processing logic to be used in radically different ways. Examples include creating components or objects from existing monolithic applications, layering the application to facilitate a change in the user interface or making the application ready for Web use.

The process of parsing a user's Natural source and storing the results in the repository means that Natural Engineer holds detailed information about the use of application code. This knowledge is utilized by Natural Engineer to provide automated support for application restructuring.

A basic requirement for restructuring applications is to the ability to identify blocks of processing code and isolate their logic as separate subprograms. Natural Engineer's Object Builder feature speeds this process significantly by automating the building of subprograms from single Natural programs.

Natural Engineer isolates a selected section of code from a monolithic main program and creates a Natural subprogram. The CALLNAT statement is placed in the main program with an automatically created Parameter Data Area (PDA), if appropriate. All necessary data areas are also built automatically. Code that is now redundant in the main program is commented out.

Having created the new Natural object, Natural Engineer is used to identify other occurrences of this program code elsewhere in the application portfolio and substitute calls to the new object for the existing in-line code.

Once a code block is selected for encapsulation, Natural Engineer automatically creates the object and all required linkages to the main program. This saves considerable time over manual techniques.

Restructuring is not the trivial task it might at first appear. Many factors have to be considered during the creation of each object, making restructuring decisions a time-consuming and error-prone task, particularly when many objects are being created from the original, single program. The use of Natural Engineer, even when only providing automated support for the generation process, can eliminate errors and significantly speed up application restructuring.

In addition to Object Builder's basic restructuring facilities, Natural Engineer also offers the automated conversion from Natural Report Mode to Natural Structured Mode.

Application Performance Analysis

When faced with a performance problem, Natural Engineer's powerful searching and automatic modification features are a valuable aid in the investigation process. All occurrences of statements and constructs that could be causing a problem are easily found and modified where required. In addition, the audit log provides an excellent tool for checking recent changes to an application system.

The complexity of code in specific areas of program code can be difficult to understand and, therefore, modify. This is particularly so if application knowledge is limited or contract development staff is used for maintenance projects. Complex application code is also often the source of performance problems in production systems.

The Metrics feature of Natural Engineer analyzes Natural source code and measures the relative complexity of the code based on industry standard measurements, such as those defined by Halstead and McCabe. Relative complexity between Natural objects is quickly displayed in report form, or as a series of graphics. This means that the areas of a system to be investigated during a performance analysis project are rapidly identified and prioritized.

With Natural Engineer, it is possible to execute an Impact Analysis, with a specific set of supplied search criteria, to identify Natural statements and definitions that could, potentially, have a negative effect on performance. The supplied search criteria are based on the detailed information on Natural internal execution provided by Software AG.

Natural Engineer makes it easy to identify exactly where these potential problems are located in the application source code. This frees up Natural developers to spend time on deciding the benefits of implementing changes based on this analysis, not on finding the occurrences.

By using Natural Engineer to find all occurrences of statements and constructs that could impact performance, it is possible to use the automatic modification features of the product to implement the code changes, if required. This powerful functionality should be used selectively and only when the impact of changes on application performance is clearly understood.

Natural Version Migration

Software AG always provides detailed information on what is new, or changed, in new releases of Natural. We also highlight the changes necessary to use Natural applications on different operating platforms.

Although the differences between versions and platforms are clearly documented, it is impossible to know immediately to what extent migration may affect your application portfolio. And so it is difficult to accurately estimate the time required for the migration project, and therefore the cost. With Natural Engineer, these problems are resolved quickly and accurately.

Built into Natural Engineer are a series of search criteria reflecting changes between Natural versions. Using these as the basis of an impact analysis allows you to quickly assess the volume and complexity of changes within your particular set of applications.

Once you identify the necessary changes, required source modification is applied automatically. Typically, solution models are created and tested. The model changes are then applied across the entire application portfolio.

In a function-rich programming language such as Natural, there is always the possibility that unique combinations of syntax will cause new problems during a migration project. Again, once a problem is identified, Natural Engineer rapidly finds all other similar occurrences in the application code and applies a suitable solution automatically.



Natural Engineer makes it easy to identify exactly where potential problems are located in the application source code.

Natural Engineer Expected Productivity Gains

With Natural Engineer you can expect to see productivity gains in the following areas of reengineering, migration and maintenance:

- Improved code maintenance productivity
- Improved code documentation
- Improved code efficiency
- Automated code re-structuring
- Improved version migration

Please refer to Table 3 for additional information.

Table 3: Potential Productivity Gains By Using Natural Engineer – Selected Items

*Note: These productivity measurements resulted from comparing steps required on the mainframe with manual work or using existing Natural utilities with steps using Natural Engineer for Windows. We also incorporated the experiences of our existing customers and of Software AG Global Consulting Services projects.

Feature	Functionality	Average Productivity Gain %**
Documentation	<ul style="list-style-type: none"> • Generation of documentation from source code to Microsoft Word • Documentation of application relationships • Graphical diagrams of relationships and program flow • Automatic inclusion of user comments in documentation <p>Compared to the legacy approach:</p> <ul style="list-style-type: none"> • Using Scan command to retrieve documentation information and references in source code • Manual documentation document creation, source code walk-through • Manual formatting of document 	90%
Analysis	<ul style="list-style-type: none"> • Predefined criteria and criteria groups • Incremental analysis (stepwise refinements) • Context sensitive analysis • Following data flow through whole application • Detailed documentation of analysis results • Interactive viewing of analysis results <p>Compared to the legacy approach:</p> <ul style="list-style-type: none"> • Using Scan command, not context-sensitive • Manual source code walk-through • Individual, hand-written utilities 	75%
Modification	<ul style="list-style-type: none"> • Automated modification in source code • Definition of modification rules • Pre-view of modifications in source code • Detailed audit log of each modification • Interactive Viewing of modification results • Applicable for single objects, sets or whole application <p>Compared to the legacy approach:</p> <ul style="list-style-type: none"> • Using Scan/Replace command, not context-sensitive • Manual source code walk-through with manual change • Individual, hand-written utilities 	25%

Feature	Functionality	Average Productivity Gain %*
Report to Structured Mode Conversion	<ul style="list-style-type: none"> • Incremental, automated approach • Structured mode Syntax Knowledgebase • Interactive viewing and reports results • Automatic change of source code <p>Compared to the legacy approach:</p> <ul style="list-style-type: none"> • Using Scan command for analysis • Manual source code walk-through and manual changes • Individual, hand-written utilities 	Between 75-95%
3-way Compare Function	<ul style="list-style-type: none"> • Comparing 2 Natural programs with 1 base program • Copy changes to base program's source code • Interactive walk-through source code differences <p>Compared to the legacy approach:</p> <ul style="list-style-type: none"> • Manual source code walk-through and manual changes 	75%
Browser-based User Interface for Diagramming and Navigation	<ul style="list-style-type: none"> • Presenting information from Natural Engineer repository (trees, relationships, flows) in a browser based on Microsoft Silverlight technology • Presenting results from Impact Analysis in Browser • Efficient navigation and zooming of information on browser screen <p>Compared to the legacy approach:</p> <ul style="list-style-type: none"> • Working with lists and green screen output 	50%

Table 4: Summary of Advantages and Benefits of Natural Engineer

Advantages	Benefits
<ul style="list-style-type: none"> • Accurate, fully documented description of all current application code • Complete picture of all impacts • Full cross-reference data always available online • Eliminates missed impacts • Eliminates need to make repetitive manual changes • Reduces implementation time • Provides full cost history of code modification activity 	<ul style="list-style-type: none"> • Improves comprehensive impact analysis and modification • Provides more accurate estimate of the total impact and cost of a proposed change or modification before you begin • Improves productivity and lowers costs • Eliminates errors • Provides ability to meet auditing requirements

CONCLUSION

Your Natural systems are the backbone of your mission critical business applications, while you consistently face evolving application requirements, new technologies and standards for integrating diverse environments. Your ability – to develop, deploy and maintain applications on a variety of platforms; to respond faster to technological developments; and to fully leverage prior investments in mission-critical Natural applications – is more critical to your competitive strategy and bottom-line than ever before.

This is why the Software AG's NaturalONE is so important – it is designed to help you improve your productivity for both application development and maintenance initiatives and generate a positive return for your organization while you overcome the many obstacles to change. NaturalONE helps you efficiently and successfully develop and maintain your Natural applications in order to leverage them in your current and future business processes and IT environments.

TO FIND THE SOFTWARE AG OFFICE NEAREST YOU,
PLEASE VISIT WWW.SOFTWAREAG.COM

Take the next step to get there – faster.

ABOUT SOFTWARE AG

Software AG is the global leader in Business Process Excellence. Our 40 years of innovation include the invention of the first high-performance transactional database, Adabas; the first business process analysis platform, ARIS; and the first B2B server and SOA-based integration platform, webMethods.

We offer our customers end-to-end Business Process Management (BPM) solutions delivering low Total-Cost-of-Ownership and high ease of use. Our industry-leading brands, ARIS, webMethods, Adabas, Natural, CentraSite and IDS Scheer Consulting, represent a unique portfolio encompassing: process strategy, design, integration and control; SOA-based integration and data management; process-driven SAP implementation; and strategic process consulting and services.

ABOUT IDS SCHEER

IDS Scheer is part of the Software AG Group and a leading provider of BPM software, solutions and services for corporations and public organizations worldwide. With the ARIS Platform for Process Excellence, the company offers an integrated, comprehensive solution portfolio for the strategy, design, implementation and controlling of business processes; allowing companies to continuously improve their overall business performance. Utilizing the ARIS Value Engineering (AVE) approach, IDS Scheer consultants bridge the gap between corporate strategy, business processes, IT solutions and process controlling.

Software AG – Get There Faster

© 2010 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.