



*CHANGEMINER WHITE PAPER*

# **Improve your application team's productivity by 30~75% with application visibility!**



Copyright © 2015 GTOne Corp. All Rights Reserved.

Copyright in this document is vested in GTOne Corp. The contents of the document (wholly or in part) must not be reproduced, distributed used or disclosed without the prior written permission of GTOne Corp.

## Overview

---

Most of organizations' application governance practices are not agile enough for today's rapid business change. Change costs a lot and contains high risks. Furthermore, application management team is required to do more with less under scarce resources and budget. Application managers, developers, analysts, and QA need productivity gains for a large amount of daily activities.

Enterprise application mining or analytics tool provides application professionals with deep insight into applications, helping them solve wide range of application issues. According to Forrester Research, "application mining tools increase productivity by as much as 30%".<sup>[1]</sup> According to GTONE's customer cases research, up to 75% of productivity improvement has been reported.

In this paper, we will explore the common issues of application and how enterprise application analytics tool provides solution.

---

[1] "Attack Weak Application Maintenance Processes That Stifle IT Productivity", Phil Murphy, Forrester Research, 2007. 2.

## Application Issues - high costs but poor productivity

---



### Rapid and frequent change bring risk

Reliability of IT service is an essential part of today's business but frequent business change is compelling to make a lot of changes to their applications. Unwary application change most likely results in IT system failures which increase the risk of a catastrophic business disruption.

When a developer makes a change to his or her source code or database object without impact analysis or deep understanding of overall relationships, the probability of system failure is high.

Frequent application change also affects build and test phase of application lifecycle management. Without application visibility, exact test scoping is not so easy which gives build/test staffs heavy burden.



### Increasing complexity and size result in application quality and compliance challenges

IT organizations are running and maintaining incredibly complex applications for decades. Their application logic touches many different databases and application components across multiple technologies such as COBOL, 4GL, Java, .NET etc.

Application size is also constantly increasing rapidly. In this context, application maintenance team can not have visibility and control of applications, which forces them to face quality and compliance challenges.



## Understanding of legacy application requires high cost

"Applications become legacy a few short year after the initial production implementation as staff move on to other assignments and knowledge of how applications function "under the covers" fades. By that definition, legacy extends to our newest development languages."<sup>[2]</sup>

When applications become legacy, SDLC (Software Development Lifecycle) tasks such as analysis, change and testing take longer time. Furthermore more errors occur in every test, and migration and integration points are missed on every project.

Fast and correct understanding of existing application is one of the key success factors of application modernization projects and maintenance. It is difficult to complete those kinds of projects on time and budget if the organization can not gain application visibility.

---

[2] "Strategies To Cut Application Costs And Increase Productivity Using Application Mining Tools" Phil Murphy, Forrester Research, 2009. 4.



## Applications are poorly documented and understood

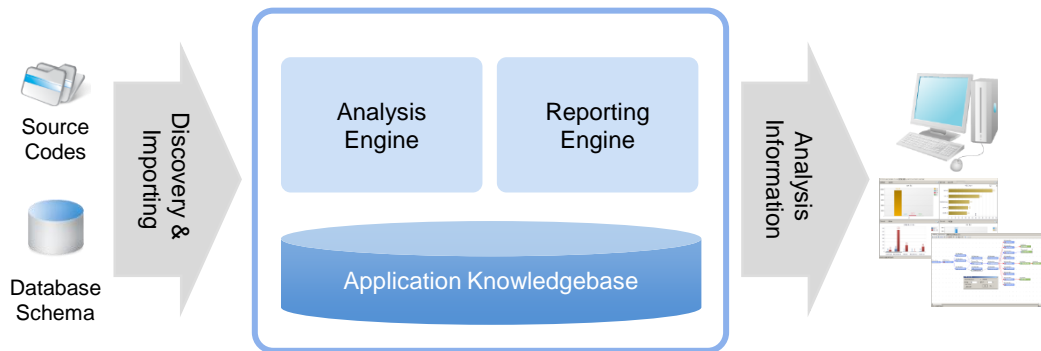
The real problem of legacy application is knowledge loss. In a short time after production, developers move on or forget documentation against changes. Eventually, lack of experienced expert and inaccurate documentation should escalate costs and lower productivity. Inexperienced developers and analysts don't know how the application works. They can not know what is affected when changing source code or database. It is hard to for QA staff to catch all affected objects and test scope in their test scenario.

In the case of application knowledge loss, people can not help but rely on their peers and documents. However, it is a reality that every organization suffers from poorly documented applications. Because documentations might not be correct against current production system, most of them are useless. Maintaining poorly understood applications must be a sort of nightmare.

## Insight from the tool solves the problems

### What is enterprise application analytics tool?

Enterprise application analytics tool automatically collects source code and database catalog and parse them to extract overall relationships among parsed objects and useful metrics. The information gained from the tool is stored in repository to implement application knowledgebase.

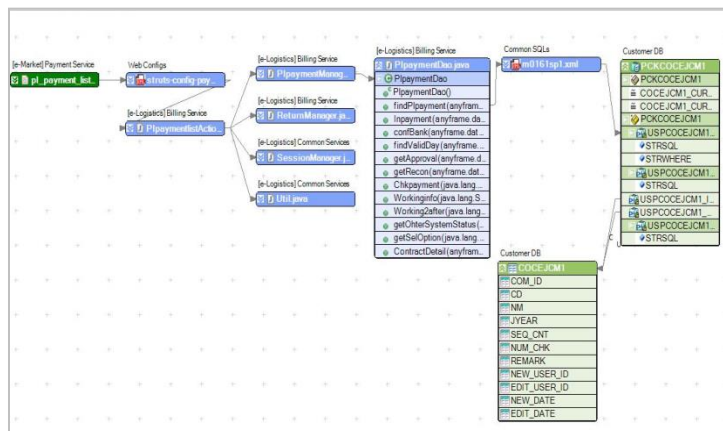


[Figure 1] The tool's basic concept

The users don't have to manually map the relationships or dependencies among the parsed objects. Rather, the tool automatically conducts everything based on fundamental user input configurations.

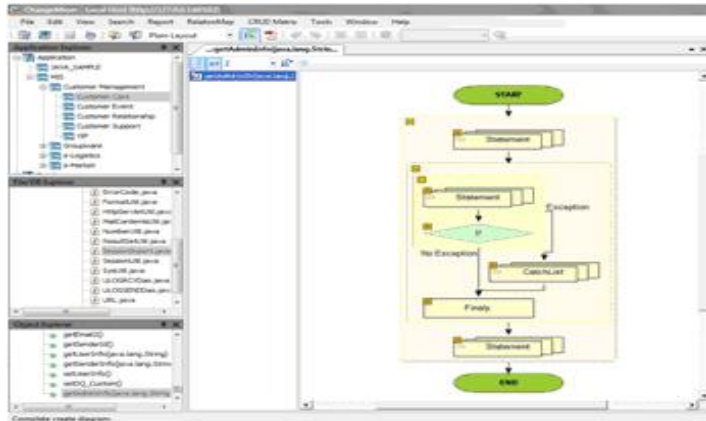
The users may want to know what kind of information is available from the knowledgebase. The users can get information like below to enhance productivity:

- Calling chain – objects or methods call other ones from client to database.
- Visual map – graphical displays of source files and methods invocation to support developers in understanding an unfamiliar application.



[Figure 2] Graphical display example

- Overall references – objects or variables are referenced through whole system and where reference to database fields.
- Source code execution flow – all possible paths by condition statements are provided in a flow chart style.



[Figure 3] Flow diagram example

- CRUD matrix – Create, Read, Update, and Delete information between programs and database tables/columns is provided to aid developers and DBA.

Program	C	R	U	D	W	Q	TABLE1	TABLE2	TABLE3	TABLE4	TABLE5	TABLE6	TABLE7	TABLE8	TABLE9	TABLE10	TABLE11	TABLE12	TABLE13	TABLE14	TABLE15	TABLE16	TABLE17	TABLE18	TABLE19	TABLE20
POKCOCEXTM.LUSPCOCEXTM1_INSERT			CR																							
POKCOCEXTM.LUSPCOCEXTM1_UPDATE			U																							
POKCOCEXHM.LUSPCOCEXHM1_INSERT					C																					
POKCOCEXHM.LUSPCOCEXHM1_UPDATE					U																					
POKCOCEFRM.LUSPCOCEFRM1_INSERT						C																				
POKCOCEFRM.LUSPCOCEFRM1_UPDATE						U																				
POKCOCEHSM.LUSPCOCEHSM1_INSERT							C																			
POKCOCEHSM.LUSPCOCEHSM1_UPDATE							U																			
POKCOCEJCM.LUSPCOCEJCM1_INSERT								C																		
POKCOCEJCM.LUSPCOCEJCM1_UPDATE								U																		

```

1 PACKAGE pkgCOCEXHM AS
2 TYPE COCEXHM1_Cur#1 IS REF CURSOR;
3 TYPE COCEXHM1_Cur#2 IS REF CURSOR;
4
5 -- 1. 한글관리 MASTER(공통-위탁) 목록 가져오기 ...
6 PROCEDURE uspCOCEXHM1_LIST(pCur OUT NOCOPY COCEXHM1_Cur#1,
7                             pSearchField IN VARCHAR2,
8                             pSearchList IN VARCHAR2);
9

```

[Figure 4] CRUD matrix along with source code

- Measurement metrics – industry standard complexity and size measures, source code never called, data never referenced, orphaned uses etc.

## Why do you need it?

---

One of the main reasons to introduce enterprise application analytics tool is to improve overall application related productivity. It allows IT organizations to do more with less.

### **Impact analysis**

The tool shows its user all objects where a component or code is changed so that he or she can detect the real scope of application and database changes. It also ensures that the users limit testing to only the change affected areas of the application. Developers using impact analysis will not miss artifacts impacted by a change, increasing accuracy and reliability. By impact analysis, organizations are able to reduce the risks from system failures due to frequent change.

### **Rapid ramp-up time – application understanding**

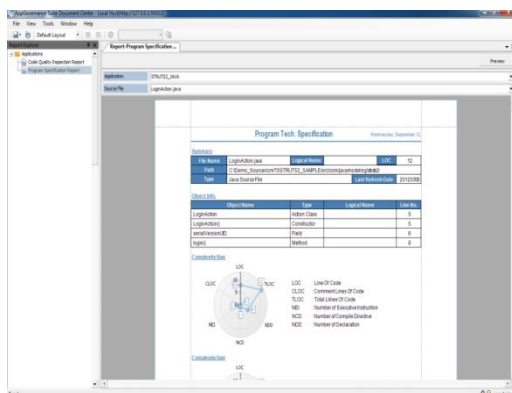
The tool offers a way to reduce the effort and costs of application understanding. Its repository serves as an on-demand reference for all staffs and permits them to easily understand the application structure and relation. Manual labor, especially in migration or modernization projects, is replaced by automation, ensuring on-time and on-budget. Without visibility of applications, developers who are not familiar with an application must rely on unproductive methods of analysis and learning: reading not-up-to-date documentation or source code, and asking the knowledge of peers who have prior experience with the application. This kind of work obviously increases project costs.

### **Better collaboration and knowledge transfer**

Without an application knowledge repository, knowledge begins to disappear as application changes remove it gained during the initial design and implementation phases. The tool built a graphical representation of the application in repository for analysts and programmers to navigate, and enabled them to drill down into various data and code structure. With enterprise application analytics tool, IT organization is able to rapidly transfer application knowledge to new developers when it hires them. Enterprise can also achieve better collaboration among analysts, developers, and QA staffs using visualized and transparent application knowledge from the tool.

### **Improving system documentation**

Documentation typically means static paper documents that are obsolete shortly after a system enters production. Static documentation usually loses its sync-points with reality as changes occur to source code but not documentation. Because of the low confidence of inaccurate static documentations, they are not used in application team's daily activities. Enterprise application analytics tools continuously updates application repository and supports automated documentation. Application team can always gain up-to-date dynamic documentations.



[Figure 5] Dynamic documentation example



## Solving other problems

Enterprise application analytics tool can help application team in wide range of area:

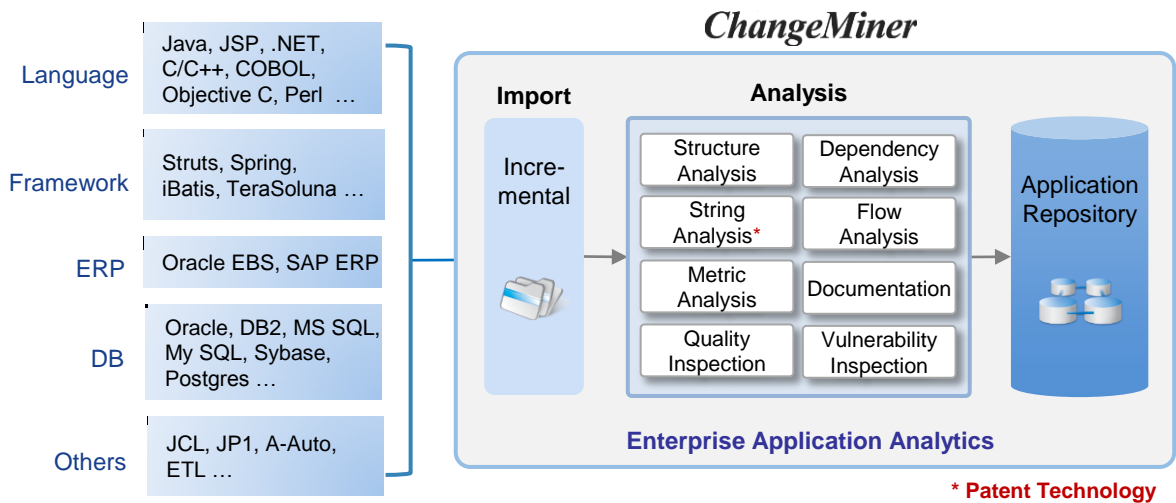
- Assessing application complexity and size - the tool automatically collects and calculates application complexity and size using industry standard measures.
- Ensuring technical standards compliance – the tool can help inspect source code for technical standards compliance. It automatically locates the source code lines that violate pre-defined technical standards. Programmers are able to analyze the quality of the source code and locate source codes that were poorly designed and causing fatal errors.
- Evaluating outsourced development – the tool can be applied to evaluate the quality and compliance of outsourced development.
- Better estimating the cost to change software –the tool provides all the relationships between source codes to identify all objects affected by a proposed change. With an accurate affection list, IT organizations can significantly increase the accuracy of the change cost estimation. It also greatly improves the success rate of the changes from a completeness and accuracy standpoint.
- Eliminating unused wastes – the tool can identify unused objects in user’s overall source files and databases. IT organization can reduce maintenance efforts for unused wastes by cleaning them based on the tool’s list.
- Managing application and database asset portfolio – IT managers can gain on-demand information on application and database asset. They can get instance answers about the following questions. How many source files and code lines are consisting each application? Which applications have complex modules? Which applications and databases are frequently changed?

Developers, analysts, QA staffs, and IT managers are getting help from the enterprise application analytics tool.

## Why ChangeMiner?

### What is ChangeMiner?

ChangeMiner of GTOne Co., Ltd. is a full featured enterprise application analytics tool that provides comprehensive insight into your mission critical applications and databases. The following figure shows its architecture. ChangeMiner consists of 3 components; server, repository and client.



[Figure 6] ChangeMiner architecture

The first component, ChangeMiner server is based on J2EE technology for platform independency and stability. There are 2 kinds of servers; analysis server and view server. The analysis server is responsible for the following tasks:

- Collects source files and database catalog from remote machine (using FTP) or local machine.
- Parses and analyzes collected source files and database catalog to extract both structure and relationship information.
- Stores analysis results to central repository.

The view server, J2EE web application, takes care of the followings:

- Processes requests from ChangeMiner client to show analysis results.
- Processes requests from ChangeMiner Admin Console to conduct administration works.

The second component, ChangeMiner repository is based on relational database. It contains all pre-analyzed data and serves as an application knowledgebase that provides useful information for application managers and developers.

The third component, ChangeMiner client is based on Microsoft .NET technology for rich user interface. It supports zero configuration installation and automatic version upgrade for user convenience. Users just need the internet connection to use it. ChangeMiner users can gain a variety of information from application repository through this client program such as relationship map, CRUD matrix, various diagrams, function points and reports. Since it also supports dynamic documentation feature, users can improve their productivity of creating document artifacts.

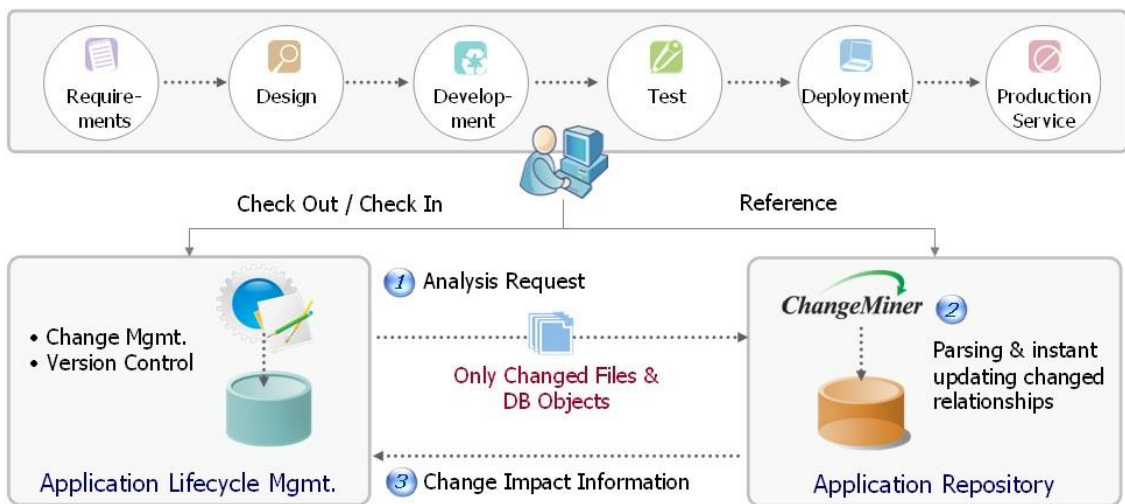


**ChangeMiner’s uniqueness – On-demand analysis**

Unlike most of enterprise application analytics tools in the market, ChangeMiner supports on-demand analysis for changed source files only. It discovers and collects the changed source files directly from remote file system or application lifecycle management repository through their APIs. Then it instantly captures changed relationships among objects by parsing only the changed source files.

Usually, other tools repeatedly parse and scan all source files to capture changed relationships. It is the reason why they analyze source files periodically, which results in a little bit outdated information. We call it a batch analysis.

ChangeMiner, however, can support both on-demand and batch analysis to improve user flexibility and provide up-to-date information. ChangeMiner’s advanced technology makes near real-time integration with ALM tools possible.



[Figure 7] On-demand integration with ALM

**ChangeMiner uniqueness – Dynamic string analysis, Patent technology**

To carry out a correlation analysis of program source code and database information, ChangeMiner uses not only the program grammar parsing method which is widely used for this purpose, but also a unique patent technology called USA (Universal String Analyzer). USA was developed when we discovered that inherent correlation data within source code could not be analyzed through lexical parsing alone.

USA has the ability to accurately extract dynamic strings without running program. Let's take look at a simple example. Assume that you have following source code:

```

72  ResultSet rs = null;
73
74  try {
75
76  con = DriverManager.getConnection();
77
78  StringBuffer sqlQuery = new StringBuffer();
79
80      sqlQuery.append("select count(*)      ") ;
81
82      if( kind == 1)
83          sqlQuery.append("   from action      ");
84      else
85          sqlQuery.append("   from behavior     ") ;
86
87      sqlQuery.append("where agent_id = '"+agent_id+"' ")
88      sqlQuery.append("   and action_ste = '00'      ") ;
89
90      LogManager.addDEVLog(3, "sung", this.getClass().getName(),
91      "getAccidentHistory()", sqlQuery.toString()+"", "selectQry");
92
93      pstmt = con.prepareStatement(sqlQuery.toString());
94

```

[Figure 8] Simple dynamic SQL example

USA can extract the expected dynamic strings when the source code would run:

- Select count(\*) from action where agent\_id=? and action\_ste='00';
- Select count(\*) from behavior where agent\_id=? and action\_ste='00';

From the above two dynamic SQL statements extracted from USA, ChangeMiner can exactly figure out the relationship between the program and database tables.

Because USA can be applied to any literals in source codes, ChangeMiner widely uses it to capture correlation data such as service name, Cobol program ID, key value for external SQL statements etc.

A benefit of USA is low rate of false positive. Without dynamic string analysis, users should conduct additional works for eliminating high rate of false positive data.



### It can analyze most languages and all relationships between them

Unlike other single language mining tools or IDE plug-ins, ChangeMiner can capture all relationships among applications written in heterogeneous languages such as COBOL, Java, .NET etc.

Although your business application has JSP or Java presentation layer, COBOL business logics and mainframe DB2 database, ChangeMiner can figure out the whole relationships and present them in a visual way. So the tool can identify enterprise impact and depth of quality analysis necessary for system changes. The following table shows the list of supported languages and database:

Language	Java, JSP, XML, Java Script, Objective C, C#.NET, VB.NET, ASP, PHP, Perl, Visual Basic, PowerBuilder, Delphi, ANSI C/C++, Pro*C, Amdocs C, PL/1, RPG, Visual Gen, COBOL: ZOS, MF, HITACHI, ILE
Database	Oracle, DB2, MS-SQL Server, Informix, Sybase ASE, AS IQ, My SQL, Postgres
Framework	J2EE Hibernate, iBATIS, Struts, Spring, Terasoluna, Ladybird, Microsoft WCF, WPF ...
ERP	Oracle EBS (Forms & PL/SQL), SAP (ABAP)
Job scheduler	JCL, JP1, A-AUTO
EAI	IBM MQ, Web Services, Tivoli, Shell for Informatica, DataStage, SAP BW, MSTR
Others	Calling patterns of non-popular languages not listed in this table can be captured by Universal Pattern Analyzer <sup>[3]</sup> and Handler <sup>[4]</sup> technology.

[3] Universal Pattern Analyzer is a name of mechanism for capturing coding patterns based on regular expressions and reference checking etc.

[4] Handler is a set of APIs for handling parsed objects internally to give more specific processing ability.

## Conclusion

Enterprise application analytics tool provides application professionals with deep insight into applications, helping them solve wide range of application issues.

Enterprise application analytics tool automatically collects source code and database catalog and parses them to extract overall relationships among parsed objects and useful metrics.

ChangeMiner is a full featured and strong enterprise application analytics tool that provides comprehensive insight into your mission critical application and database. Its uniqueness comes from:

- On-demand analysis for changed source files only
- Unique patent technology called USA (Universal String Analyzer) - accurately extracts dynamic strings without running program.
- Capturing all relationships among applications written in heterogeneous languages

Your application team is able to reduce the cost and complexity of application by solving several issues with enterprise application analytics tool. Source code version control or application lifecycle management alone is not enough. It's time to change your application governance style.



<http://www.gtonesoft.com>

<http://www.changeminer.com>

## **Restricted Rights Legend**

This software and documentation is subject to and made available only pursuant to the terms of the GTOne License Agreement and may be used only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronics medium or machine readable form without prior consent, in writing, from GTOne, Corp.

Information in this document is subject to change without notice and does not represent a commitment on the part of GTOne. The software and documentation are provided "as is" without warranty of any kind including without limitation, any warranty of merchantability or fitness for a particular purpose. Future, GTOne does not warrant, guarantee, or make any representations regarding the use, or the results of the use, of the software or written material in terms of correctness, accuracy, reliability, or otherwise.