Modeling and analysis of cost-efficient Web advertisement

By

Karl Rylander, f96-kry@f.kth.se
Jens Jonsson, d96-jou@d.kth.se

Department of Microelectronics and Information Technology

2001-10-09

Examiner:	Prof. Rassul Ayani Department of Microelectronics and Information Technology Royal Institute of Technology
Advisor:	Ass. Prof. Vladimir Vlassov Department of Microelectronics and Information Technology Royal Institute of Technology
Industrial Advisor:	Mikael Eriksson EPO.com

Abstract

The object of this thesis is to find a way to monitor and optimize advertisement efforts on the Internet so that money is spent on the right kind of advertisement alternatives. In order to do so it is necessary to develop a tool that can perform traffic surveillance on the web server and a mathematical model that can predict and optimize the effect of the money spent on advertising. The two applications are stand-alone but they interact with each other when confirming if the predicted results from the mathematical model were accurate or not.

Due to the large amount of uncertainties in the mathematical model no precise indication can be given whether or not the model works. Instead the ambition has been to develop and explain a first model from which more advanced models can be derived. The mathematical model is also very dependent on how statistical data is gathered and therefore the monitoring of the effect of the advertisement is an essential part so that data can be adjusted to fit reality.

There are many traffic analyzing tools available on the market today. The reason for creating a new tool is because the existing tools cannot deliver enough detailed visitor information, to serve as meaning full input parameters to the mathematical model. The monitoring tool also suffers from errors, which are then transferred to the optimizing program causing erroneous output.

The conclusion of this thesis is that monitoring visitors and knowing what they do on the site is valuable information for any Internet based company despite the errors involved. The optimization of a campaign is also doable but maybe not from a strictly mathematical point of view. The uncertainties in such a model are too great to be discarded and the solution is therefore inaccurate.

Summary of contents

Table of contents

Table of figures

1 Introduction

1.1 Background

EPO.com is an Internet based company located in Stockholm. EPO stands for electronic public offering and serves as a distributor of initial public offerings. EPO.com was founded in 1998, today EPO.com is a part of EO.net and has sites in five European countries.
On the EPO site people can get the latest news from the economical sector and also take part in new initial offerings. There is also a possibility to subscribe to a weekly newsletter. Today EPO.com has over 10,000 registered members that can buy shares through EPO.com and around 70,000 newsletter subscribers.

1.2 Introduction to the problem

EPO.com is dependent on advertisement to attract people in becoming members. By having a large member database there is a greater possibility for the issuer of the shares to choose EPO.com as a distributor. In other words attracting people to the site is essential for success. Advertisement is the only way to achieve these results and EPO.com has run several campaigns both off and on line.
Since a considerable amount of money is spent on advertisement it is important to know the effect of this money and to optimize it if possible. The goal of this thesis is to create a model that maximizes the effect and minimizes the cost of the advertisement. A web traffic-monitoring device will be constructed and serve as an input source providing data necessary for the optimizing algorithm.

1.3 Expected results

The outcome of the thesis is to provide EPO.com with a better understanding on how to maximize their advertisement efforts with a minimal cost. Figure1 shows the interaction and the information flow of the suggested solution in this thesis. The information flow goes as follows: (1) a marketing director supplies the optimization model with initial data about the campaign. (2) the model then gives a suggested solution that may include several sites and different time periods. When visitors are referred (3) from the advertisement sites to the EPO.com site the web server writes (4) information about the client to the log file. (5) the Website Analyzer can then generate a detailed analysis of the web site traffic and thereby give the marketing director a good view of the outcome (7) of the campaign. The marketing director can then fine-tune the input parameters so that the optimization model can give a more correct solution. The Website Analyzer can also serve as a direct feedback channel (6) to the advertising sites showing amount of visitors, page hits etc.

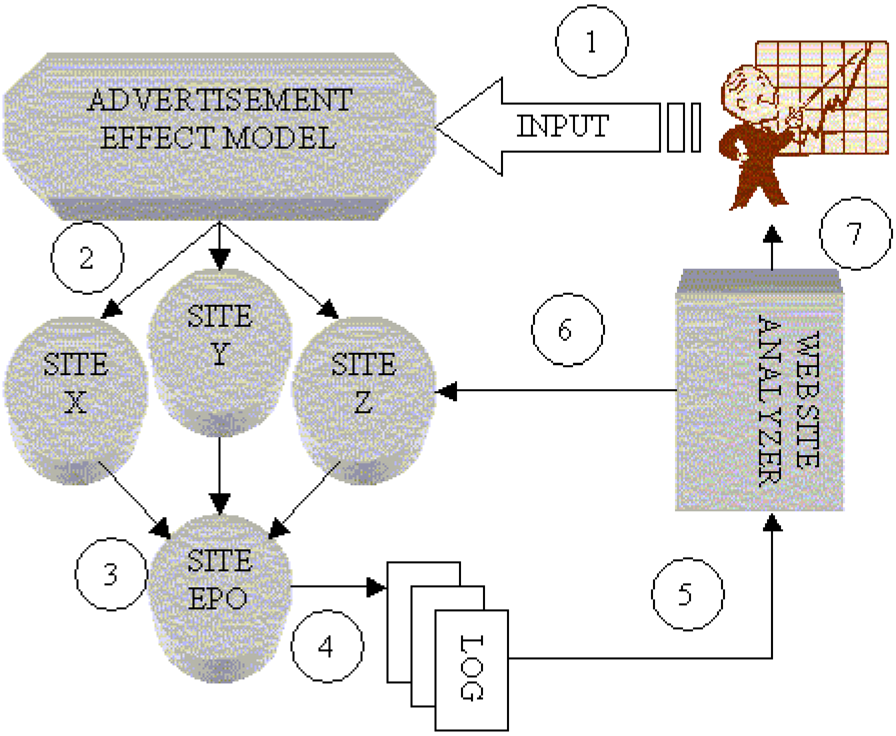


Figure1 The interaction and the suggested solution

1.4 Structure of the thesis

There are two main parts in this project. The first part, Website traffic optimization , written by Karl Rylander, is an optimizing tool that predicts and gives guidelines on how to setup an optimal campaign. The other part, Website analyzing tool, written by Jens Jonsson, is a program for measuring the web site traffic and also the effect of the accuracy of the optimization model.

The thesis is divided into five parts, part one, two and five have been co written, part three is written by Karl Rylander and part four is written by Jens Jonsson.

Part one of the thesis gives a short introduction to EPO.com and the problem at hand. The second part describes the language chosen for implementation and also the architectural structure of the system at EPO.com.

The third part describes the website traffic optimizer. It begins with an explanation of how Internet advertising works and then moves on to previous work. The construction of the mathematical model is described and the information about the members at EPO.com is gathered. Then the implementation is described and the section ends with conclusions and future work.
In part four it is described how the traffic analyzing tool works, it also describes previous work and existing software, implementation and testing follows, as well as conclusions and future work.
Part five describes the interaction between the optimization tool and the traffic-monitoring tool. It describes the limitations and problems with this type of work; it also covers the expected results and future work.

2 *Architectural overview*

This section gives a short overview of the current system design at EPO.com. The implementation style and languages used by EPO.com and in this thesis are also shortly explained.

2.1 Current system design

The EPO.com site is constructed using a so-called three-tier solution as shown in Figure2. Tier 1 is the client domain. The client makes requests to the server in tier 2 where ASP (Active Server Pages) code is processed; the result of the code is then displayed in the client browser.

Tier 2 is as previously mentioned the server domain. Here the web server processes the client requests and executes logical functions written in VB (Visual Basic). The VB layer sends queries to the database in tier 3.

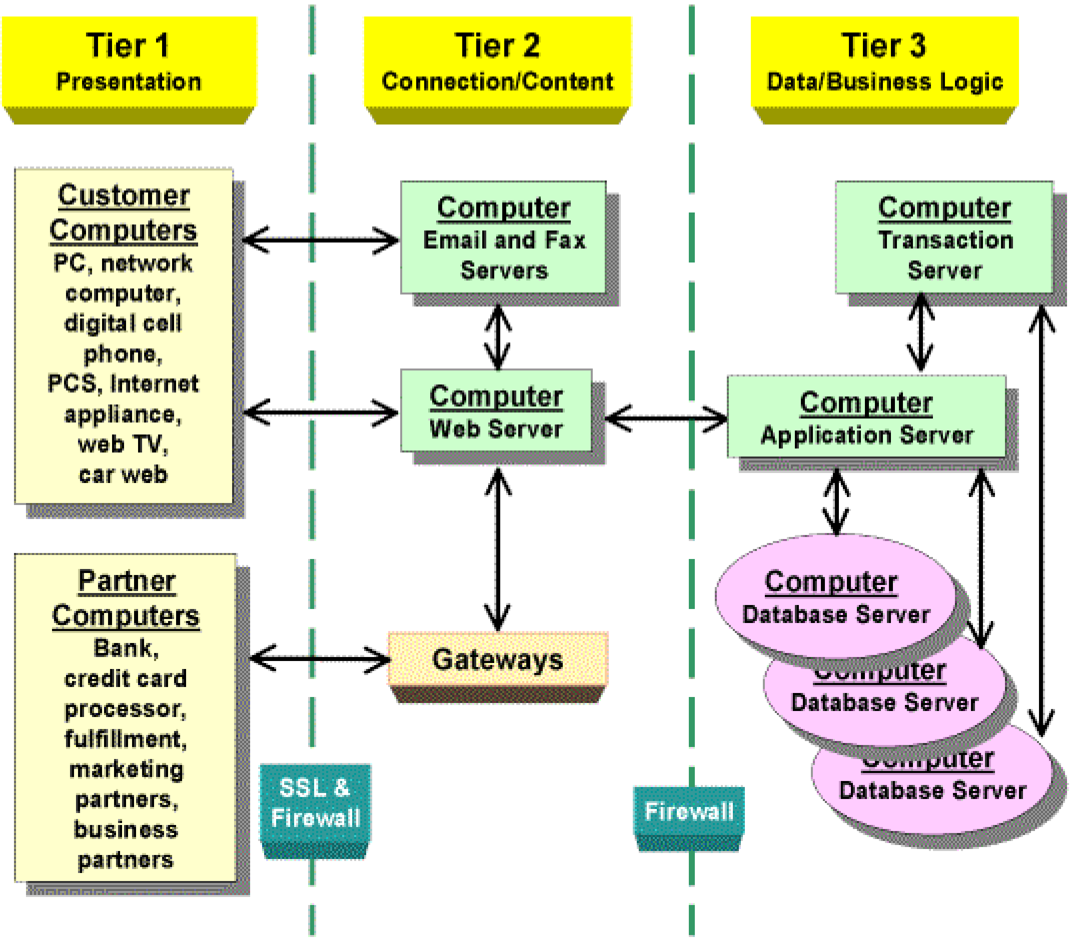


Figure2 A three-tier solution.

The EPO site could be divided in two parts, the **Front end** and the **Back end.** The front end is what the visitors see when they reach the EPO site. Here they can get

information and take part in new investment opportunities. It contains five country specific sites that both share and have country dependent ASP pages. They also have a set of shared and country specific VB-functions. These functions are called from the ASP layer and perform the more time consuming and complex tasks. A large SQL-database with user and system data is used by the functions in the VB-layer.
The back end works the same way as the front end but this is the interface that is used by the people working on the system. It is used to view and modify investor data and for controlling the contents of the front end. The back end is protected by login control.

2.1.1 VB

Visual Basic[20] is a high level programming language evolved from the earlier DOS version called BASIC. BASIC means **B**eginners' **A**llpurpose **S**ymbolic **I**nstruction **C**ode. Visual Basic is a visual and event driven programming language. These are the main divergence from the old BASIC. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In Visual Basic, programming is done in a graphical environment. Because users may click on a certain object randomly, so each object has to be programmed independently to be able to response to those actions(events). Therefore, a Visual Basic program is made up of many subprograms, each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another.

2.1.2 ASP

Active Server Pages (ASP) [21] is a server-side design environment that makes it possible to create engaging Web applications. An ASP page is an HTML page that contains server-side scripts that are processed by the Web server before being sent to the user's browser. Unlike conventional Common Gateway Interface (CGI) applications, which are difficult to create, ASP is designed to greatly simplify the process of developing Web applications. With just a few lines of script you can add database connectivity or advanced customization features to the Web pages. In the past, it was necessary to use PERL or C to add such functionality, but with ASP it is possible to use ordinary Web scripting languages such as Microsoft JScript, Microsoft Visual Basic (VBScript), or any COM compliant scripting language, including JavaScript, PERL, and others.
Beyond ordinary scripting tasks, ASP can be used to extend scripts into COM components. These reusable, programmatic modules make it possible to scale scripts into full-fledged applications that perform complex tasks such as transaction processing for electronic commerce.

2.1.3 Database

In the project a MS SQL database server [22] is used. This is the database server used on the EPO website as well. The MS SQL Server is an easy to use database handler that can execute SQL statements written from either a MS Query tool or from within VB or ASP code. Microsoft SQL Server permits client applications to control the information retrieved from the server using several specialized tools and techniques. These include options such as stored procedures, server-enforced rules, and triggers that permit processing to be done on the server automatically. You don't have to offload all processing to the server, of course. You still can do appropriate information processing as needed on the client workstation.

3 Website traffic optimization

3.1 Introduction

Complex mathematics has had great wins in areas that were before empirical and based on experience. The economic sector is today dependent on advanced mathematical models to predict the future and give adequate information about the present. The problem in this thesis is to guide advertisers so that correct decisions can be made regarding advertisement options before money has been spent and before the campaign has been launched. In order to more formally describe this, and thereby formulating a theory and building a model, mathematics needs to enter the picture in this area too. This section, Website traffic optimization , will try to develop and implement such a model that explains and reflects the advertisement world on the Internet.

Almost every company tries to somehow measure the effect of their advertisement. It has always been difficult to see the real effect of a campaign and therefore information has not existed to check whether or not the money spent gave the desired result. Well, does not this problem still exist and why is it possible to create a model today and not before? Traditional advertisement and the effects of it are still hard to measure and it is difficult to construct a model simulating this. Traditional advertisement in this case is represented by television, radio, i.e. advertisement that has no interaction. Together with the Internet another form of advertisement has evolved – banner ads. For the manager concerned with reaching business objectives, the Web promises the precise quantification of the effectiveness of a particular advertising campaign in terms of those objectives – for the first time in the history of media. It is this quantification that makes this thesis feasible.

3.2 Internet advertising

Internet advertising is rapidly emerging as a key strategic tool in the battle for online customers. Internet advertising revenues exceeded $3 billion dollars in 1999 and are expected to hit nearly $13 billion by 2003 [6], [4]. There is little doubt that the Internet will continue to grow and therefore also Internet advertising. A common problem is that managers have no idea how their advertisement is going and what the effect of it is. If managers knew a little more about the result they would invest more money in advertisement. To quote an anonymous market director regarding advertisement on the Internet: “We have the first truly accountable advertising medium! We can literally count each customer that responds to the ad banner with a click through to the advertiser’s Web site.” [9] Again this is a breakthrough for market directors and with a clever theory or model advertisement cannot only be measured but perhaps also be predicated in terms of effect or impact. If the effectiveness can be measured before the campaign has been launched market directors can decide whether or not the campaign is worth running and based on that result, reconstruct the campaign so it reaches the desired results. The first task in achieving this is to understand how marketing on the Internet works and how companies choose to advertise themselves and what techniques they prefer.

3.2.1 Advertising techniques

In the early days of the Internet advertisers used exposure based pricing models, where the cost of an ad is a function of the total number of impressions delivered by the advertising site. Page impressions or page views refer to the number of times a web page has been requested by the server. That impression has no special target group and every page view, in terms of spending money, cost an equal amount for the advertiser. Although exposing a broad demographic target to a commercial message can satisfy awareness and branding objectives, smaller, more targeted segments with whom the firm can interact are actually worth much more if they are exactly the customers most likely to give the desired market response. Thereby ending up paying for people that are more likely to use the site and not for the others. Today web advertising has evolved since the early day of the Internet, both from the surfers and the advertisers point of view. It is interesting to see that Internet advertising efforts still favor banner ads that are based on exposure based CPM or click-through rates, as in the dawn of the Internet. CPM is a metric from the print days of advertising, meaning "Cost Per Thousand," using the Roman numeral "M" to stand for one thousand. A price of $15 CPM means, $15 for every thousand times a banner is displayed. A click through comes from the word click, when a visitor clicks his or her mouse on a banner ad, he or she is transferred to the advertiser's site. The number of responses to a banner ad is sometimes refereed to as the number of "clicks." Click throughs are therefore commonly used to count the number of visitors who click on the banner and are transferred to the advertiser's site [13]. The reason why CPM and click throughs are used is that such ways of advertising are widely understood by traditional media executives and that they are easy to implement.

Another form of advertising, that does not use traditional ways, is the pay-for-performance model. “In pay-for-performance ad strategies, the Web provider displaying the ad shares more in the risks and rewards of advertising placement than traditionally has been the case. The revenue gained by achieving a particular market response, such as a product sale, is shared between the Web advertiser and the web provider”[1].

The three pricing models used are:

- CPMs
- Click throughs
- Pay-for-performance

The model most frequently used is the CPM, ending up paying for impressions. The click through rate is unfortunately not used very often. Later it will be explained why this pricing model would generate a better and more stable mathematic model. According to [1] pay-for-performance is the most “webby” form of advertising and for many Internet based companies it is the most effective way to sell a product or be introduced on the market. It seems like smaller companies lean towards the pay-for-performance strategy, more interested with the actual profit involved in a campaign. The bigger players seem to favor other models due to interest in not only the measurable profit but also in branding process. While cost per sale relates to direct marketing objectives, another way of looking at banner ads is as "branding" tools. They create brand awareness, and a brand image in the viewer's mind, whether or not the viewer clicks on the ad. Branding is very difficult to measure, but can be very powerful. Typically, only the larger and better-established companies have the budget to pursue branding consistently.

3.2.2 Banner ads

Consumers visit publishers Web sites according to their individual interests and tastes. Once there, consumers have the potential to be exposed to advertisements in the form of banner ads placed throughout the site. Banner ads come in many sizes and colors, the Internet Advertising Bureau (IAB) specifies eight different standard banner sizes, they can be viewed in the Appendix A . The initial goal of such web advertising efforts is to attract customers to the advertiser's web site [4]. This ability depends greatly on the advertisers target group. Banner ads are no good if the people that are supposed to see them are no frequent Internet users. Then the campaign would do no good.

The question is of course whether measuring click through is an appropriate form of valuing Internet advertising. According to [2] it depends on the goals of the advertiser. If the goal is to attract visitors to a web site then the effectiveness should be evaluated by measuring the ability to generate the desired response. If the goal is to build brand awareness or brand attitude then it is better to use measures of ad-related or brand-related responses to measure the effectiveness of the advertising.

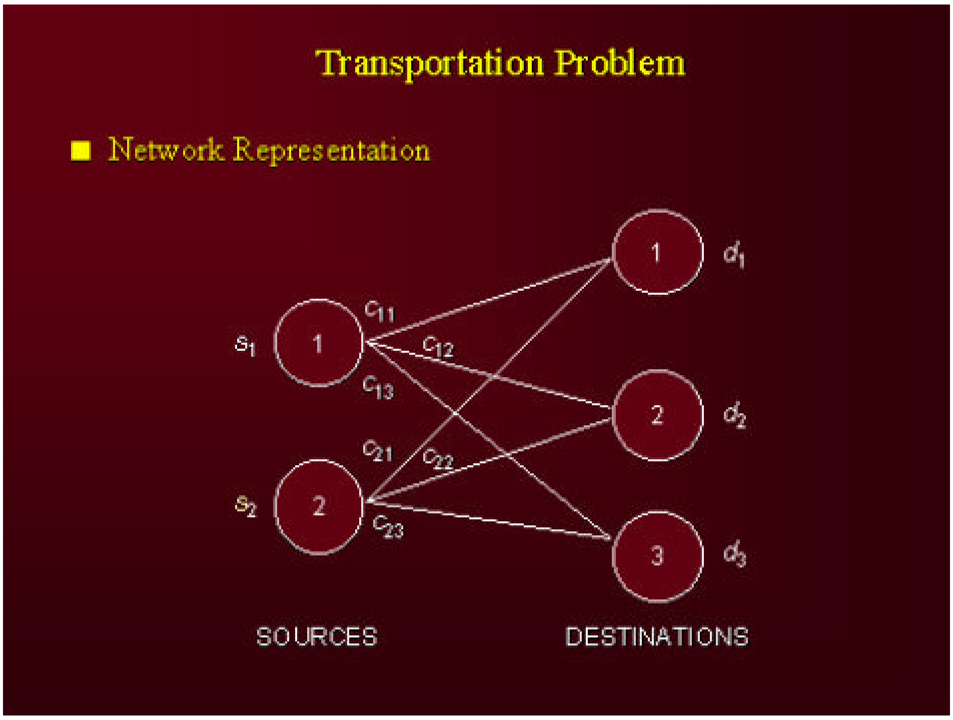
For this thesis banner ads must be used since it is almost the only way to measure traffic. It is not the most effective way and it certainly has its limitations. Banner ads will probably never disappear from the Internet, [4] and [1], but they will definitely loose their strong position to some other from of advertising that is more cost efficient and so that both parties involved in the advertisement profits from it, like the pay-for-performance alternative.

3.3 Theory and related work

Relevant articles covering Internet advertising and the economical background involved with it are quite easy to find. It has been more difficult to find interesting technical articles. The reason for this is probably because this way of measuring advertisement success in a quantified way and then being able to optimize it is a new way of thinking, and been impossible to apply until now.

3.3.1 Target function

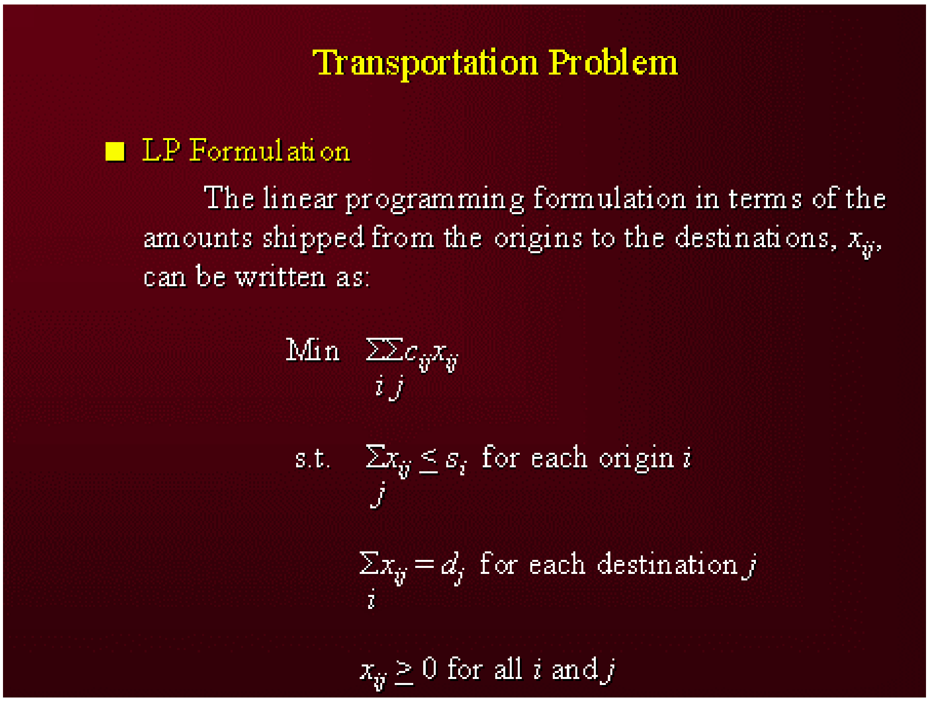
As mentioned technical information was hard to come by. Two important articles were found and they proved to be very valuable [5], [11]. The goal in these papers is to develop a new technique of adapting online advertisement to a user's short-term interest in a non-intrusive way. The system relies only on search keywords supplied by a user to a search engine. Based on the user's current interests the system dynamically selects a best matching advertisement. By only relying on one or more keywords, no user specific data is collected and the solution is therefore non-intrusive. It is not so much the idea but the theory behind it that is important. One paper, [11], uses the same kind of transportation problem that will be used in this thesis, a transportation problem is always nice to deal with because there will always be feasible solution to the problem, if constructed the right way [12]. Figure3 displays the cost coefficients $c_{ij}, i = 1,2\ j = 1,2,3$ involved in a typical transportation problem when “shipping” a quantity between the nodes.


Transportation Problem

- Network Representation

Figure3 The “shipping” involved in a transportation problem

The idea of the transportation problem is that the amount shipped cannot exceed the supply. Figure4 displays this relation. The first constraint says that the amount shipped, x_{ij} cannot exceed the supply s_i. The second constraint says that the amount shipped to a node d_j must satisfy the nodes demand. The target function $\sum_i \sum_j c_{ij} x_{ij}$ is the function that will be minimized with respect to x_{ij}.



Transportation Problem

- LP Formulation

The linear programming formulation in terms of the amounts shipped from the origins to the destinations, x_{ij}, can be written as:

$$\text{Min} \quad \sum_{i}\sum_{j} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j} x_{ij} \le s_i \text{ for each origin } i$$

$$\sum_{i} x_{ij} = d_j \text{ for each destination } j$$

$$x_{ij} \ge 0 \text{ for all } i \text{ and } j$$

Figure4 The entire transportation problem.

Instead of using a transportation problem, which sets the target function to be linear, it is possible to use a non-linear target function [5]. This paper deals with the trivial solution that can appear in an ordinary LP optimization problem when parameters are not known with perfect precision; this problem makes itself heard in this thesis too. This problem is easily explained with an example: Say that x_1, x_2 stands for the money to be invested and that the profit made is explained by the profit coefficients c_1, c_2, the budget is 100.

$$\begin{gathered} \textit{Maxímize} \;\; c_1x_1 + c_2x_2 \\ \textit{subject to} \;\; x_1 + x_2 = 100 \\ x_i \ge 0 \end{gathered}$$

Equation 1

For instance if $c_1 = 51$ and $c_2 = 49$ then the optimal solution to this problem is of course $x_1 = 100$, $x_2 = 0$. Spend everything on alternative number one and nothing on number two. Is this solution realistic or even wanted? Well that depends, if the profit coefficients were known with perfect precision, then the solution above would be correct. In the real world there is no such thing as perfect precision and data like click through probabilities are uncertain. If the uncertainty in the click through probabilities is only 5%, then in the worst case the actual profit coefficient might be 46 for x_1 and 54 for x_2 and the optimal solution would have completely changed. The optimum must be more robust if the model is to hold, therefore [5] proposes the following solution:

$$\begin{gathered} \textit{Maxímize}\ \ c_1x_1 + c_2x_2 - 0.5x_1\ln(x_1) - 0.5x_2\ln(x_2) \\ \textit{subject to}\ \ x_1 + x_2 = 100 \\ x_i \geq 0 \end{gathered}$$

Equation 2

Equation 2 has the same interpretation, as Equation 1 the difference is that the non-linear terms make the solution more robust. The non-linear terms serve as a penalizing function that prohibits the target function to change drastically when the profit coefficients are slightly changed. If $c_1 = 51$ and $c_2 = 49$ the solution to the NLP (Non linear problem) is $x_1 = 51.00$, $x_2 = 49.00$. This solution is worse than the LP solution, 50 to 51 but on the other hand it is much more stable. This method can be used instead of using statistical uncertainties.

3.3.2 Visitor value

In terms of profit there are different ways to put a revenue gain on visitors, click through, click streams, time spent on the site and pages accessed on the site. The following parameters are used when valuing a surfer and what he does on the site, [10]:

- Number of absolute accesses per page,
- Number of relative accesses per page,
- Mean page time, how much time a user spends on a specific page,
- Mean user time, how much time a user spends on the server every session,
- Mean number of pages, how many different pages a user visits every session.

To create a value function with these parameters and also to track the visitor on the site so that these parameters are given a value is very hard work. This advanced technique in valuing surfers will not be used in this thesis. Another form of valuing will be applied. Nevertheless it is interesting to see that the valuing function can be made so large that only solving that problem can take an incredible amount of time.

3.4 Model construction

Now it is time to formulate a theory that explains the world of Internet advertising from a mathematical point of view. The pay-for-performance model is very hard to implement because of its complexity, no evident structure can be found in that type of advertisement. The pricing models that remain are represented by either an approach where every customer is unique and handled by a binary value; this way to go is represented by the pricing model click through, or an approach where the amount spent on a certain form of advertisement generates a stochastic amount of click-throughs; this is represented by the pricing model CPM. Since sites today still favor CPM it is obvious that this pricing model must be used in the mathematical model.

When the advertising sites charge by impressions, a click-through probability must be estimated in order to convert impressions to click throughs. The ability to predict an optimal campaign depends greatly on the expected arrival intensity meaning the click through rate of the page views. The reason why click throughs are better than CPM is because the conversion of impressions to click throughs is an error source and it is clear that if one uncertainty is removed from the model, it will generate a more precise answer. Therefore also mentioned in section 3.2.1 click throughs is better to use than CPM.

3.4.1 Profit - Expense

Since the costs involved in the model are known, the money being spent on advertising, it is vital to find a relevant profit function. The parameter that stands for the real money in the model is the profit of an event. These events will be explained in section 3.7.4 but to give a short overview they are:

- A click through that turns into a page view,
- A click through that turns into a newsletter subscriber,
- A click through that turns into a member.

The expected revenue of these events enters the model as a profit. The profit estimation involved with each event is as mentioned explained more thoroughly in section 3.7. It is a difficult task tracking a visitor from the advertising site and then knowing what he or she does on the site. The tracking tool being built simultaneously, explained in section 4, will give information about this. The tracking tool should be able to see who were attracted to the site by what banner and then follow their actions on the site.

The greatest uncertainty in the model will be the values corresponding to the probabilities that a page view at the advertisement site will turn into for instance a newsletter subscriber. The visitor analysis tool will provide information about how customers behave on the site, which will be used to verify and update these probabilities.

Another difficulty is that it is quite dangerous to compare the money expected to be generated to the actual money being spent. The expected revenue is built on assumptions and if they are wrong the solution will fail. It is therefore important to see to it that the solution is robust and does not change dramatically when then expected profit of an event is slightly disturbed, another problem is the click through probability and the statistical error in them. As mentioned, [5] gives a solution to this problem with a non-linear target function, which makes the solution more stable to perturbations in the click-through probabilities. This can also be applied in this thesis but it is also possible to let each and every one of the coefficient pass through an error function before a solution is generated. That would simulate the possible statistical error in the coefficients. The solution would be different each time as in the real world.

3.4.2 Mathematical formulation

With the information presented above it is time to create a model that simulates the generated amount of click throughs from different sites and what that gives in terms of profit.

The target function or profit function is created the following way:

Target function:

$$\min \sum_i \sum_j \sum_t x_{ijt} - \sum_i \sum_j \sum_t \sum_k f(x_{ijt}(\frac{d_k p_k}{c_{ij}}),t)$$

x_{ijt} : The amount spent on advertising for site i, banner ad j under day t
$(i = 1,..,n),(j = 1,..,m),(t = 1,..,p),(i,j,t) \in N$

d_k : The value of a click through turning into a virtual profit value k,
$(k = 1,..,q), k \in N$

p_k : The probability that a click through turns into a virtual profit value k,
$(k = 1,..,q), k \in N$

c_{ij} : The weighting coefficient that describes how many page views that turn into click throughs on site i and banner ad j, $(i = 1,..,n),(j = 1,..,m),(i,j) \in N$.

f : The function that describes the relation of the virtual profit function and the x_{ijt}:s. It is a description saying if the x_{ijt}:s follow a linear, logarithmic or any other relation.

The time t is essential in the model. This parameter sees to it that the effect of the advertising is not the same in terms of how many days the banner has been up on the advertising site. The decay of the advertisement effect is not known and therefore it is difficult to estimate the decaying function with the parameter t. The constants in the model are d_k , p_k and c_{ij} . They will be estimated each time the model is used. d_k is expressed in SEK / click through, p_k is a probability $\sum_{k=1}^{q} p_k = 1$ and c_{ij} is expressed in SEK / click through. These constants will be fine tuned if it turns out that they have been incorrectly estimated. More details on that subject will be explained in section 3.7 where they will either be given a concrete value or an explanation on how to calculate them.

The unknowns are the x_{ijt}:s, and the function f explains under what relation the x_{ijt}:s go under. As seen f is not explicitly defined, unfortunately there are implementation restrictions on how the function f can be constructed, for now f stands for a relation of some kind.

For a clearer understanding of the model it might be a good idea to explain in words what is meant to happen. To use the simplest case let $i = j = t = 1$. Let also $k = 3$ which means that there are three different values that the click-throughs can turn into, see section 3.4.1. The following target function is constructed:

$$x_{111} - f(x_{111}(\frac{d_1 p_1}{c_{11}}),1) - f(x_{111}(\frac{d_2 p_2}{c_{11}}),1) - f(x_{111}(\frac{d_3 p_3}{c_{11}}),1)$$

x_{111} is the cost, and the function f stands for the profit made by the this expense. The model tries to minimize the cost of the advertisement with respect to the x_{ijt}:s. The quotient $\frac{x_{111}}{c_{11}}$ shows how many click-throughs that will be generated when the amount

x_{111} is spent. If $x_{111} = 10000$ SEK, and c_{11}= 50 SEK/click through then the amount of generated click throughs will be 10000 SEK/50 SEK/click through = 200 click throughs. These will then turn into the virtual values d_k depending on the conversion rate probability p_k, thereby generating a profit. If $d_1 = 1$ SEK, $d_2 = 10$ SEK, $d_3 = 100$ SEK and $p_1 = 0.7$, $p_2 = 0.2$, $p_3 = 0.1$ the target function will have the following appearance:$10000 - f(140,1) - f(400,1) - f(2000,1)$.

The limiting conditions will be difficult to formulate, they serve as restrictions to the solution. Unfortunately enough interesting limiting conditions have not been found. That is mainly because the solution is quite restriction free. An experienced advertiser could probably formulate conditions that have to hold for the solution to be interesting. There is no use in seeking for restrictions, if they are found they could easily be implemented. Using the formulation of the transportation problem explained in section 3.3.1 the "supply" is described by Equation 3. This condition describes that the amount of money spent on the different sites cannot exceed the advertisement budget D.

$$\sum_i \sum_j x_{ij} \le D$$

Equation 3

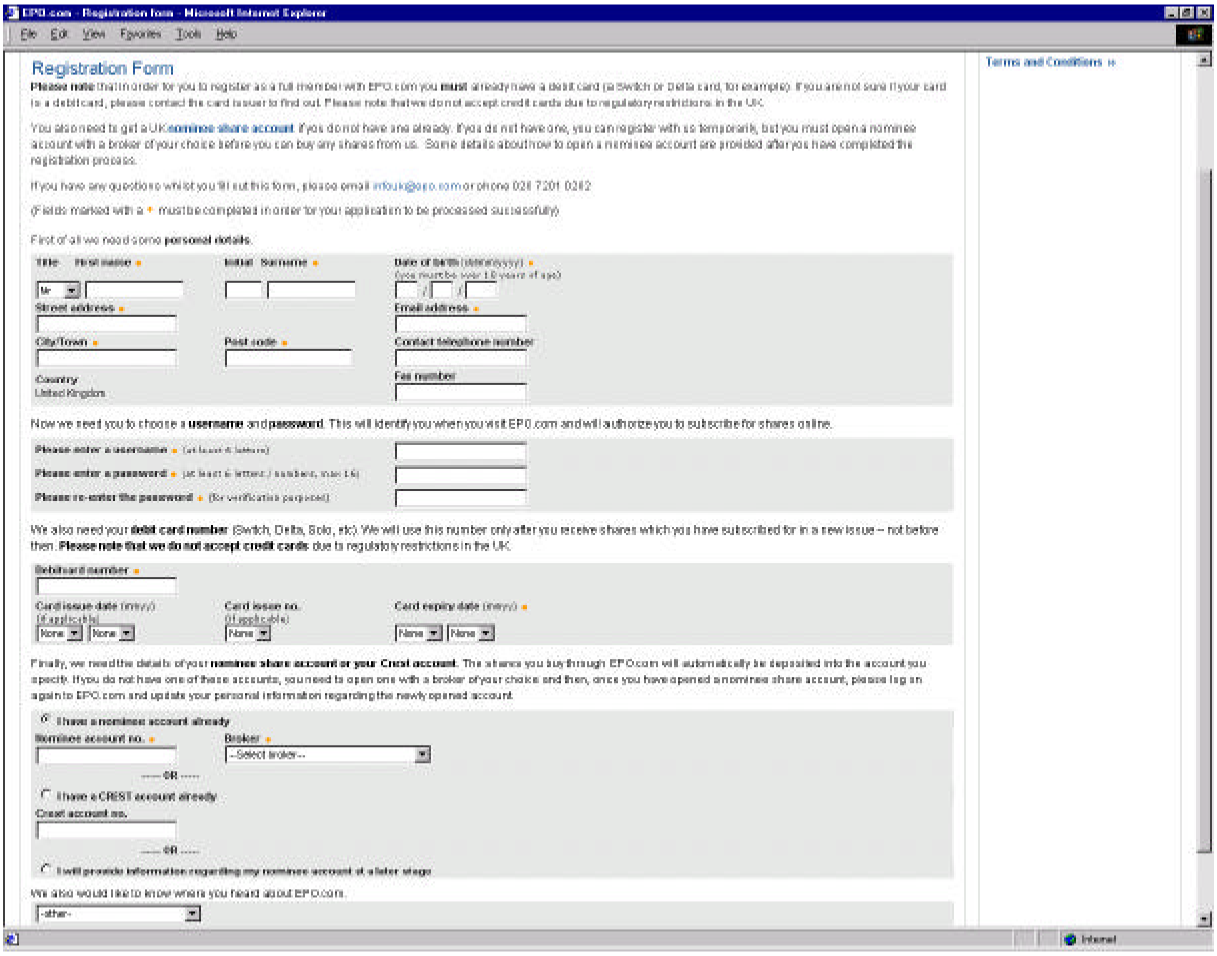
In terms of the transportation problem, the constraints that remain to be found are the ones describing the "demand". They can be added at any time, the problem is solvable without them but if interesting restrictions are found that limits the solution they can as mentioned be easily implemented.

3.5 Detailed task specification

The theoretical model has been created and explained, what remains to be done as far as the model goes is to describe more intimate the input parameters and also to implement the model. In order to be able to estimate the model parameters in section 3.4.2, it is important to understand the member distribution at EPO.com, which is done in section 3.6. The reason for this is that it is more cost efficient to advertise where more click throughs are generated. Advertising on sites that have roughly the same target group as EPO.com achieves this. After it is clear what members that use the services of EPO.com it is time to estimate the value of the click throughs, and how much money that is expected to be made on those click throughs, this is done in section 3.7. The model stands and falls with these estimations, if they are wrong the solution will be incorrect.

3.6 Member Information

In order for EPO.com to understand its members and their behavior it is desired to perform some information gathering regarding the profile of these members. The definition of members in this case is people who have typed in personal information like social security number, bank account number and provider account. In Figure5 the registration procedure is shown.


EPO.com - Registration form - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Terms and Conditions »
Internet

Registration Form

Please note that in order for you to register as a full member with EPO.com you **must** already have a debit card (a Switch or Delta card, for example). If you are not sure if your card is a debit card, please contact the card issuer to find out. Please note that we do not accept credit cards due to regulatory restrictions in the UK.

You also need to get a UK **nominee share account** if you do not have one already. If you do not have one, you can register with us temporarily, but you must open a nominee account with a broker of your choice before you can buy any shares from us. Some details about how to open a nominee account are provided after you have completed the registration process.

If you have any questions whilst you fill out this form, please email infouk@epo.com or phone 020 7201 0202

(Fields marked with a * must be completed in order for your application to be processed successfully)

First of all we need some **personal details**.

Title Mr **First name** * **Initial** **Surname** * **Date of birth** (dd/mm/yyyy) * (you must be over 18 years of age) / /

Street address * **Email address** *

City/Town * **Post code** * **Contact telephone number**

Country United Kingdom **Fax number**

Now we need you to choose a **username** and **password**. This will identify you when you visit EPO.com and will authorize you to subscribe for shares online.

Please enter a username * (at least 4 letters)

Please enter a password * (at least 6 letters / numbers, max 16)

Please re-enter the password * (for verification purposes)

We also need your **debit card number** (Switch, Delta, Solo, etc). We will use this number only after you receive shares which you have subscribed for in a new issue – not before then. **Please note that we do not accept credit cards** due to regulatory restrictions in the UK.

Debitcard number *

Card issue date (mm/yy) (if applicable) None None Card issue no. (if applicable) None Card expiry date (mm/yy) * None None

Finally, we need the details of your **nominee share account or your Crest account**. The shares you buy through EPO.com will automatically be deposited into the account you specify. If you do not have one of these accounts, you need to open one with a broker of your choice and then, once you have opened a nominee share account, please log on again to EPO.com and update your personal information regarding the newly opened account.

● **I have a nominee account already**

Nominee account no. * Broker * --Select broker--

---- OR ----

○ **I have a CREST account already**

Crest account no.

---- OR ----

○ **I will provide information regarding my nominee account at a later stage**

We also would like to know where you heard about EPO.com.

-other-

Figure5 The registration page at EPO.com.

Everything entered in the registration page is stored in the database. When a person is a member at EPO.com he or she can participate in IPOs, Initial Public Offerings, which means that he or she can by new issues. Members also get the newsletter once a week, which contains financial news. The large database at EPO.com serves an excellent starting point in building this target group profile. The information that can be gathered from the database is as mentioned facts that are typed in by the user and also the deal related facts like subscriptions, allocations and price paid for shares. It is up to the company to chose how much personal information that is needed from the surfer in order for he or she to become a member.

3.6.1 Motivation

There are many reasons why a solid understanding of customers is important for a company like EPO.com. In the thesis this information will be used to optimize advertising in the sense that banner ads are put on sites where they give the most effect, meaning sites that have roughly the same target group as EPO.com. From a more technical aspect the member information will be used to give different click through rates to people coming from different sites, the highest click through rate will be given to the site that has the most similar target group compared to the target group of EPO.com.

The member information will, as mentioned, be retrieved from the database at EPO.com. There are limitations on how much can be found out through this procedure, a survey is needed to get more detailed information, but the database queering is a cheep way to get accurate information. The information that will be collected follows below:

- Age distribution
- Gender distribution
- Deal participation

3.6.2 Age distribution

The age distribution of the members at EPO.com is important in order to target the right site since many sites are quite age specific. The age distribution is displayed in Figure6, the curve plotted is the moving average.

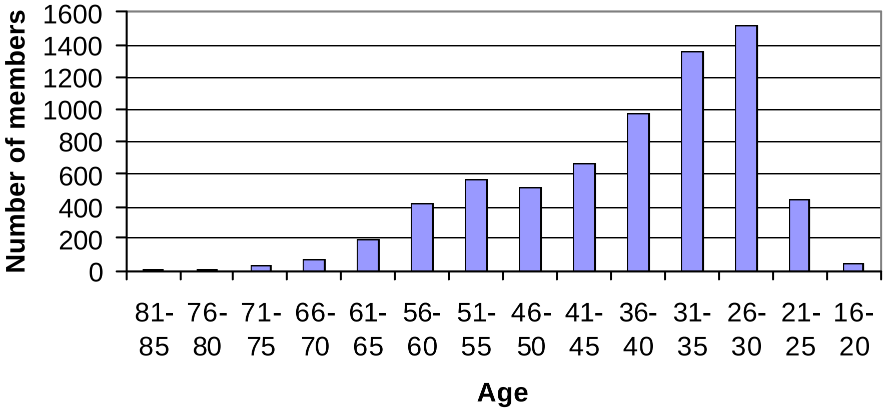


Figure6 The age distribution of the members at EPO.com.

The most frequent age group is 1970 – 1975. Many people in this age category have a good education, have good salaries and are well accustomed to using the Internet. Although it might not be the optimal target group it is not surprising that this age group is the most frequent.

3.6.3 Gender distribution

Gender is a selection parameter that is important, although it might not be as important as age or income it gives valuable information about the customers, especially since the usage of Internet today is quite equal regarding gender. The gender distribution is shown in Figure7. It is evident that males dominate.

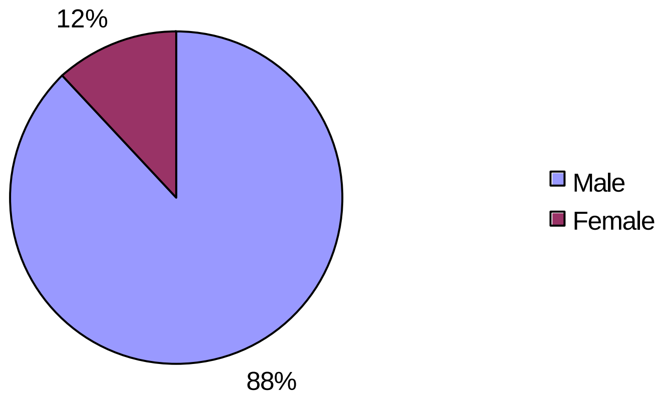


Figure7 The gender distribution at EPO.com.

As pointed out earlier it is interesting to notice that there is such a large shift towards males despite the increase of female Internet users.

3.6.4 Deal participation

Deal participation gives information on what kind of spending habits or deal interest the members at EPO.com have. In Figure8 information about deal participation is given. The majority of members have chosen to invest in one deal and as seen the moving average does not include the first staple.

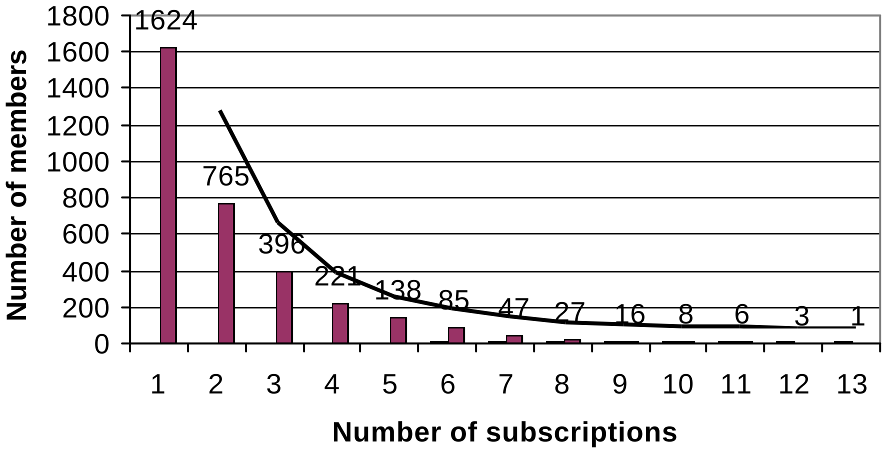


Figure8 The number/numbers of subscriptions different members have made.

Most likely they have been particularly interested in one company and bought shares in that specific company. When people chose to invest in one deal and get allocation the chance that they invest in another deal might grow, or it might not. It is nevertheless interesting to see how many subscriptions that have turned into allocations; this is displayed in Figure9.

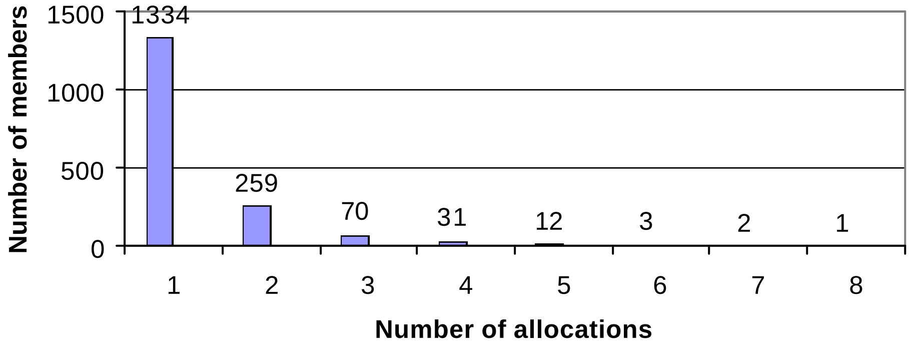


Figure9 The number/numbers of allocations different members have received.

A comparison can be made between Figure8 and Figure9. The chance of receiving allocation for a member the first time subscribing is substantially higher than if he or she has participated in a couple of deals. This can be the result of one or two very large deals where people have registered themselves just for these specific deals and where the allocation factor has been high.

Since one-time investors seem to dominate it is interesting to see if they dominate in terms of capital. A possible scenario could be that a small group of investors stand for most of the capital. Figure10 shows the one-time investors and so on compared to how much these investor groups have generated in terms of capital.

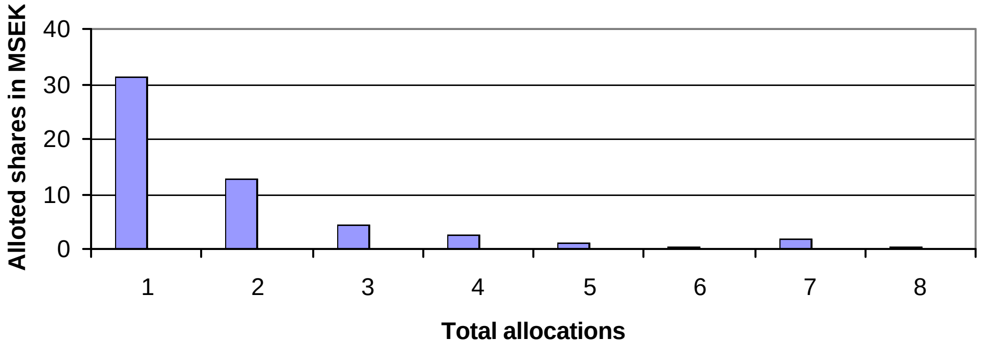


Figure10 Number of times investors have received allocation and how much capital they have generated.

The one-time investors dominate heavily; there is no large capital from a small group of investors. This makes it even clearer that the current member groups are attracted to EPO.com by a specific deal.

3.6.5 EPO.com Survey

Although a lot of information of the members can be gathered from the database there is some information that is not accessible. EPO.com has done a statistical survey in March 2000 to cover these gaps. The survey was a sort of an interactive study where 500 newsletter subscribers were chosen randomly from the database. The subscribes then received an email with the following text:

"Dear Mr, Ms or Mrs,

As a registered subscriber to the EPO.com newsletter, we would like to thank you for your interest and ask if you would kindly consider visiting the site in order to complete our customer survey form? This should take no more than five minutes of your time and would greatly aid us in improving the EPO.com experience. Completion of our survey also entitles you to be entered in our subscriber prize draw, which could win you £100/SKr1,000!

The survey is being conducted as an online questionnaire, and to reach it, you just need to click on the address below. To ensure that only those who are invited fill in the survey, you have been allocated a password for access, which is embedded, in your personal link below: <link>"

A reward was offered to the people that filled out the questionnaire which very often increase the participation rate quite drastically. The questionnaire used can be viewed in the EPO.com Customer Survey in Appendix C. The survey covers quite a lot of information, and for this thesis some facts are more important than others. They are:

- First learn of EPO.com,
- Annual spend on investments,
- Household income.

First learn of EPO.com tells whether or not advertising gives result. According to the survey it is obvious that banner ads have given a measurable result. It is the single most important form of advertising.

Annual spend on investments and household income is interesting because it shows how much money the average investor has and their willingness to invest this money, in this case especially in IPOs. The most common household income is between £30000 - £49999 and the spend on IPOs in the next 12 months is less than £5000. There are indeed an infinite amount of parameters that can be taken into account when trying to map a certain target group. The question is if the amount of work put into statistical information gathering pays of in the sense that it refines the member profile enough to be worth the effort.

3.6.6 Source of errors

It is evident that a survey from March 2000 does not reflect the current situation in terms of economic climate today. For instance the annual spend on investments might change drastically when the economic climate changes. Another issue is that the study was conducted with newsletter subscribers and not members. The people that are members will be newsletter subscribers but since the amount of newsletter subscribers is substantially higher than the amount of members, the members will be greatly outnumbered since the survey selection process was random. It is important to have this in mind when using the survey to map members. Non-the less the survey can give valuable hints that can be used to pinpoint people that are interested in the site and the service that it provides.

3.6.7 Compilation of member information

The information from the database says that the average member is 25-30 years old, has only participated in two deals, got allocated one time and the gender is male. The survey says that the household income is around 500000 SEK and that the annual spend on investments is less than 70000 SEK. It is quite easy to see what kind of people who are members at EPO.com. A good guess is young people with good jobs, who are well accustomed to using the Internet. EPO.com must ask themselves if this is the desired member or if another group of people is “better” in terms of creating revenue. The highest click-through rate will anyhow be generated when choosing sites like the target group specified above.

3.7 Parameter estimation

Under the time period December 1999 – July 2000 EPO.com has been advertising online for an amount of 1,2 MSEK. Under this time period the company acquired 20168 newsletter readers and 2729 active members. It is very hard to give an estimate on how many of these members actually was a result from advertising. The amount of real money that EPO.com has made on advertising is also very hard to estimate because it has been very hard to measure the real effect of the money spent. As mentioned earlier the economic climate was better during the time studied and therefore people were more likely to spend money on investments, and therefore more likely to be attracted to services like the one EPO.com provides. It is safe to assume that the advertisement gave a better result during the winter of 2000 than it would deliver today and the numbers on how many newsletter subscribers and members that were acquired are not very reliable today.

The main function of this section is to estimate the value of an acquired page view, member, or newsletter subscriber. In May 2000 an acquired newsletter subscriber at EPO.com was valued to 100 SEK and an actual member was valued to 1000 SEK. These values have served as guideline on how much money a marketing director is allowed to spend. Today these generated values mean nothing, they are completely wrong and later it will be shown how these values can be regenerated.

3.7.1 Defining parameters

The model created depends greatly on the input values. If the estimated values put into the algorithm are wrong, then so will the result be. The parameters that will have to be estimated are the following:

- Click through probability
- Transition probability
- Member, Newsletter and Page view value
- Site Member value

It is important to understand that these parameters might be inaccurate. There are many error sources that can ruin the correctness of these estimated parameters. With the web site analyzing tool that will be explained in section 4, more accurate values can be produced.

3.7.2 Click through probability

The chance of a web surfer seeing and clicking on a banner at a site that is advertising something depends on a number of parameters. Not only is it the message the banner gives but also the design and banner placement that decides whether or not it is a "good" banner [9]. Banners are all over the Internet and they are proven to be quite effective, at least in the early days when banners had a large click through rate, today surfers seem to be pickier about which banner they click on and therefore the click through probability has decreased.

In order to target the right site it is vital to know the target group of that specific site, and that has been done in section 3.6. It is not the goal of this thesis to map the target groups of all the specific sites that might be appropriate to advertise on. The member profile serves as a base in case EPO.com decides to advertise on other sites than it already has. During December 1999 – July 2000 EPO.com has received information from the advertising sites on how many impressions the banners have delivered and how many that clicked on the banner. From this information it is possible to estimate the click through rate with acceptable precision. These parameters can then be fine tuned with the web-analyzing tool, section 4. The results of the different banner campaigns can be viewed in the Appendix B under banner campaign results.

Click through rates are all subject to fluctuations and if they enter the model with out any margin of error the answer will be trivial. Given that the implementation language supports the usage of statistical functions and distributions it is a good idea to somehow include this in the click through rates. The solution would be more robust if the click-through coefficients changed values subjected to some statistical error function.

3.7.3 Transition probability

The transition probability describes the chance that a page view either remains a page view or turns into a newsletter subscriber or a member. Again the visitor-analyzing tool will give this information after a couple of campaigns have been run. The transition coefficients are uncertain just like the click-through coefficients and the same statistical problem naturally arises.

3.7.4 Member, Newsletter and Page view value

In order to estimate the value of an acquired member it is important to look at the past, and also decide on what is expected from the future. The past represents the actual profit that one customer has created. The future represents the goals that the company has and what the company hopes for in terms of profit from the members. EPO.com sells banner placements to companies that are interested in advertising on the site. The target group is well defined, just like the sites EPO.com is looking for. There is therefore an interest in advertising at EPO.com. An acquired member generates profit because he uses the service and thereby sees the banner ads on the site.

To be able to estimate the expected profit and the expected banner profit the life length of a customer must be found, otherwise it will be impossible to come to any conclusions what so ever regarding the profit. Due to the sparse information about how long people remain members at a company that has only existed from a couple of years, no distribution function can be found. This is of course a draw back because the profit that a customer generates cannot be sufficiently estimated. If enough information could be gathered the following relation would be used:

$$P(x) = \int_0^x f_{life\_length}(y)dy$$

Equation 4

$P(x)$ is the probability that a person remains a member for x number of years, $f_{life\_length}(y)$ is the distribution function for the life length of a member. It is also possible to use the mean value, or the variance together with the mean value. Until a reasonable distribution is found good guesses will have to be made on how long a person will remain an active investor at EPO.com. If Equation 4 is used it is possible to run different scenarios and see when and where the campaign becomes profitable. With the expected life length two simple relations can be derived which finds the profit corresponding to the future and to advertising.

The advertisement profit:

$Pageviews/Year$: An estimate on how many page views an investor creates during one year.

$Advertisement\_\Pr ofit / Pageview$: The advertisement profit per page view.

$Advertisement\_\Pr ofit / Member$: The total advertisement profit for a member.

E : The mean value of the life length of a customer.

With these parameters the following relation can be found:

$$\frac{Advertisement\_\Pr ofit}{Member} = \frac{Pageviews}{Year} x \frac{\Pr ofit}{Pageview} xE$$

The future profit:

$T/Year$: Expected turnover of one customer per year.

$Future\_Profit/Member$: The expected profit in share sales for a member.

$P\%$: The expected profit coefficient ($TxP\% = Profit$)

E : The mean value of the life length of a customer.

$$\frac{Furure\_Profit}{Member} = \frac{T}{Year} xP\% xE$$

Both the newsletter and page view value profit comes from advertising, but in different forms. The single page view comes from a surfer clicking on a banner, being transferred to the advertising site, seeing the banners that the advertising company has posted on their site, an impression, and then decides to leave the site without doing anything. This event has of course the lowest associated profit, but it is never the less an income. The other ad related event is when a surfer becomes a newsletter member. A newsletter itself contains ads so every week, the newsletter at EPO.com is sent once a week, the subscriber is subjected to ads from the newsletter. These ads have a different value than the ads posted on the site. This leads to the conclusion that an acquired newsletter member has two incomes, the first page view together with the ads in the newsletter. It is very likely that a newsletter subscriber also uses the site and thereby delivering more impressions to the banners posted on the site.

The mean life length for a newsletter subscriber is probably different than the life length for a member. Statistics for this is also needed to find a corresponding distribution. If the mean value is found the same reasoning can be used here as with the advertisement profit model.

The newsletter subscriber profit:

$Number\_Of\_Newsletters/year$: The Number of newsletters that are sent over a year to one subscriber.

$Advertisement\_Profit/Subscriber$: The total profit involved in newsletter advertisement for a subscriber.

$Advertisement\_Profit/Newsletter$: The profit per page view.

E : The mean value of the life length of a customer.

With these parameters the following relation can be found:

$$\frac{Advertisement\_Profit}{Subscriber} = \frac{Number\_Of\_Newsletters}{Year} x \frac{Advertisment\_Profit}{Newsletter} xE$$

3.7.5 Site member value

When a surfer is attracted to a site from another that surfer might be more worth depending on which site he or she comes from. The users of a specific site can have spending habits or other specific properties that are wanted or perhaps preferred, although it is not the actual target group. Then it might be appropriate to give this specific click through a higher value if he or she becomes a member. How this weighting should be done is of course total subjective. This parameter can be used when the model is to be fine-tuned.

3.8 Implementation of model

From an implementing point of view it is desired to find a programming language that can use and handle every mathematical aspect in the theoretical model. There are several language options available, some better than others. To give a clear picture of these languages they will be explained and analyzed, after that a decision will be made which programming language or mathematical tool to be used. The languages are:

- GAMS
- MATLAB
- MatrixVB

3.8.1 GAMS

This text is taken from the home page www.GAMS.com: "The General Algebraic Modeling System (GAMS) is a high-level modeling system for mathematical programming problems. It consists of a language compiler and a stable of integrated high-performance solvers. GAMS is tailored for complex, large scale modeling applications, and allows you to build large maintainable models that can be adapted quickly to new situations."

As the text suggests GAMS can handle a large variety of mathematical optimization problems. The implementation design is high level and very close to how the problem is written on plain paper. The advantage of GAMS is that almost every part of the theoretical model can be implemented. The drawbacks are that it is not user-friendly and that the license is very expensive. A student license exists but it cannot be used after this thesis has been completed. GAMS is of course the best solution from a mathematical point of view, unfortunately the price and the hands on experience required to change parameters in the model makes it inappropriate for this thesis. An interface would also be quite time consuming to build which would also limit the usability of the program.

3.8.2 MATLAB

MATLAB is used throughout the entire technical spectrum. It is quite easy to use and since the possibility to build C programs and link them to MATLAB exists large-scale problem building is no problem. Though MATLAB is filled with predefined mathematical functions it still lacks some basic optimization tools. Linear mathematical problems of the form are easy to solve:

$$\min_x \; c^T x$$
$$Ax \leq b$$

Matlab also supports QP (quadratic programming) problems. They have the following structure:

$$\min_x \frac{1}{2} x' q x + c^T x$$
$$Ax \leq b$$

If more advanced constraints than the linear ones are added to the LP or QP problem MATLAB will not be able to cope with these, nor can it deal with more complex non-linear optimization in a satisfactory way. This will have implications on the model, the ability to use a non-linear target function and constraints will not exist, but on the other hand it might not make a difference because the linear approximation might be good enough. It is difficult to estimate the different sources of error in the model and what impact transforming a complex problem into a less complex problem does to the solution. The restrictions in this case are that MATLAB only supports the problem structure mentioned above. Linear problems can be easily solved but as soon the problem enters the non-linear spectrum MATLAB has a hard time. The decision to be made is whether or not these restrictions make it impossible to implement the model or if the linear approximation is good enough.

From an interface implementation point of view it would be great if MATLAB could be linked to Visual Basic because EPO.com uses the Visual Studio package and also because it is very easy to create user friendly interfaces with Visual Basic. If MATLAB were to be used it would be preferable to find a tool or a package that can be linked to for example Visual Basic and that has optimization tools that MATLAB has.

3.8.3 MatrixVB

MatrixVB is a COM library, a collection of functions that enhance Visual Basic's built-in functionality by allowing users easy access to many powerful computational algorithms. MatrixVB carries many of the same functions as MATLAB, as well as the optimization tools. Since MatrixVB can be accessed through COM via Visual Basic it is very easy to construct user-friendly interfaces.

3.8.4 Language decision

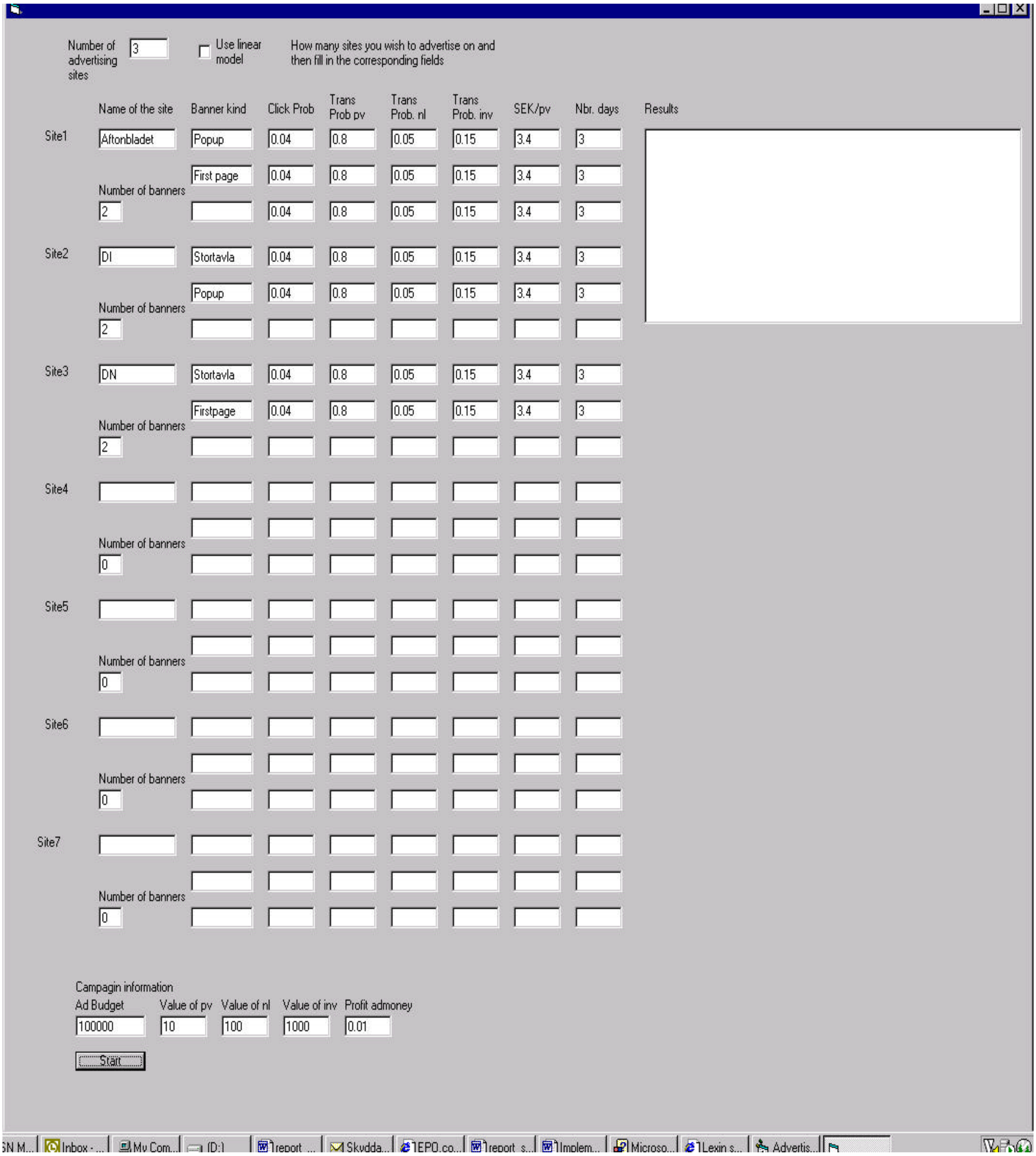
It is clear that from an academic point of view GAMS is the optimal choice and that from a user point of view Visual Basic with the package MatrixVB is the best choice. As mentioned earlier it depends on whether or not the linear or the quadratic model is sufficient or not. For EPO.com it is not an option to buy the GAMS license. Two options remain, MATLAB or MatrixVB. Since MatrixVB holds the same functions as MATLAB with the ability to create user-friendly interfaces, MatrixVB will be the language used.

3.8.5 User Interface

If a program or an application is to be used by a person with limited computer knowledge it is important to make the program or application robust and easily understood. Especially important is that the interface of the program is comprehensible and that the functionality of the program is implemented so that all adjustments to parameters can be done by just clicking and typing. In order for this to be possible a graphical interface will be used. Visual Basic is perfect when designing interfaces, especially graphical ones, but like any other language it has its limitations, which means that sometimes a detour has to be taken in order to land at the desired result.

The static parameters that have been explained in section 3.4.2 are the ones that should be typed in by the user. The detour mentioned was that it is hard and difficult to make the form dynamic. A form in Visual Basic is the actual interface where textboxes, radio buttons or any other predefined object are displayed and implemented. To explain this problem an example will be used: Say there are 10 sites that have been chosen for advertisement by the marketing director. Each one of these sites has different prices, click through probabilities and other parameters that are specific for that site. On one site more banners are placed than on others and the time period for these ads to be posted on the advertising site might be longer or shorter than on the other sites. It would be nice to have some sort of dynamic input for each site and banner alternative, instead of having an interface like the one that has been created, see Figure11. When the dynamic fields have been typed in and it is time to move on to the next site the previous values are saved or stored in some sort of array or dictionary. It would greatly reduce the amount of redundant code and also make the entire program generic in the sense that the program would handle an infinite amount of sites and banner alternatives. As mentioned this is not available in Visual Basic, instead the entire interface is hard coded into one form. The number of sites and banner alternatives has an upper bound. It is important to clarify that it is merely the graphical interface that set the bounds. There are certainly ways around the problem with the dynamic forms, but to investigate this would be a waste of time. An experienced Visual Basic programmer solves this problem quite fast.

The user interface is shown in Figure11and how the application looks when the program is started. The values are typed in the corresponding fields, the maximum number of sites is limited to seven and each site can have three different banner alternatives. The number of days the banner should be put on the site is not limited to any value. The result field shows the optimized results with the site name, banner kind, money that should be invested that day and how many visitors this brings to the site. When the program is to be used the user specifies the name of the site and the definitions of the different banner alternatives that will be used on that corresponding site. After that the click through probability and transition probability for a page view, newsletter subscriber and a member is put in. The cost of an impression i.e. how much the advertising site is charging for an impression and also the amount of days the banner should be posted in the site is entered. When all the sites for the advertisement campaign have been entered the user pushes the Start button and the optimal solution is given in the results window. That was a short explanation of typical usage of the program.



Number of advertising sites: 3

□ Use linear model

How many sites you wish to advertise on and then fill in the corresponding fields

	Name of the site	Number of banners	Banner kind	Click Prob	Trans Prob pv	Trans Prob. nl	Trans Prob. inv	SEK/pv	Nbr. days
Site1	Aftonbladet	2	Popup	0.04	0.8	0.05	0.15	3.4	3
			First page	0.04	0.8	0.05	0.15	3.4	3
				0.04	0.8	0.05	0.15	3.4	3
Site2	DI	2	Stortavla	0.04	0.8	0.05	0.15	3.4	3
			Popup	0.04	0.8	0.05	0.15	3.4	3
Site3	DN	2	Stortavla	0.04	0.8	0.05	0.15	3.4	3
			Firstpage	0.04	0.8	0.05	0.15	3.4	3
Site4		0							
Site5		0							
Site6		0							
Site7		0							

Results

Campagin information

Ad Budget	Value of pv	Value of nl	Value of inv	Profit admoney
100000	10	100	1000	0.01

Figure11 The user interface for the advertising campaign optimizer.

3.8.6 LP/QP – implementation

The interface described in section 3.8.5 serves as the donor to the optimization algorithm. It supplies the parameters to the model so that the target function can be created. The theoretical model in section 3.4.2 will now be implemented. MatrixVB supports two optimization algorithms, the first one is an LP model (linear programming) and the other one is a QP-model (quadratic programming). As mentioned in section 3.4.2 the function f has restrictions in terms of how it explains the relation of the x_{ijt}:s. The function f can give a linear or quadratic approach when using MatrixVB. If a more sophisticated language would have been used then f could have been more complex. MatrixVB uses a matrix representation when solving problems. If the linear approach is used the current structure of the problem can be transformed to the following matrix representation.

$$\min_x c^T x$$
$$Ax \le b$$

If on the other hand the problem is transformed to a QP representation it will have the following form:

$$\min_x \frac{1}{2} x^{'} qx + c^T x$$
$$Ax \le b$$

The advantage with this description is that there is a quadratic term present that takes away a little bit of the uncertainties in the cost coefficients. The correct q matrix must be found so that it penalizes the cost coefficients the right way. It is also desirable if the q matrix is positive semi definite because any local optimizer is global, so it finds the global optimum.[12] The problem is to find this q matrix. The idea behind this quadratic matrix is best explained by an example; the simple example in section 3.4.2 will be reused. If q is an eye matrix, ones in the diagonal, the quadratic term will have the following form: $\frac{1}{2} x_{111}^2 + x_{211}^2 + x_{311}^2$. This term will be minimized together with the linear term $c^T x$. The quadratic term grows much faster than the linear term, if the solution is to be minimized the money spent must be evened out over the advertisement alternatives even though site one and banner alternative one is the best. If the solution had been completely linear as much money as possible, the supply being the limitation, would have been spent.

In order to implement the model the so-called cost coefficients must be transformed to matrix form. Equation 5 displays the cost coefficients.

$$C_{ijt} = (1 - \frac{P_{ij}T_{ij}^{P}V_i^{P}}{W_{ij}} - \frac{P_{ij}T_{ij}^{N}V_i^{N}}{W_{ij}} - \frac{P_{ij}T_{ij}^{I}V_i^{I}}{W_{ij}})t^{-1/2}$$

Equation 5

It might be appropriate to explain the time dependency in Equation 5. For now there is really no way of knowing how the click through intensity behaves under the time period the banner is on the advertising site. Again the tracking tool will provide valuable information about how this intensity varies over time. When enough statistical data has been collected a function can be constructed from that data. The relation $t^{-1/2}$ is chosen because the decay of the intensity over time is reasonable, but this approximation is not built on any mathematical background more it is more of a common practice form the world of Internet [4]. The linear part of target function will have the following appearance in matrix form:

$$(C_{111}, C_{112}, \dots, C_{11t}, C_{121}, \dots, C_{12t}, \dots, C_{1jt}, \dots, C_{ijt}) \begin{pmatrix} X_{111} \\ X_{112} \\ \vdots \\ X_{11t} \\ X_{121} \\ \vdots \\ X_{12t} \\ \vdots \\ X_{1jt} \\ \vdots \\ X_{ijt} \end{pmatrix}$$

The quadratic term is represented by the matrix representation:

$$\begin{pmatrix} X_{111} \\ X_{112} \\ \vdots \\ X_{11t} \\ X_{121} \\ \vdots \\ X_{12t} \\ \vdots \\ X_{1jt} \\ \vdots \\ X_{ijt} \end{pmatrix} \begin{pmatrix} X_{111} & X_{112} & \cdots & X_{11t} & X_{121} & \cdots & X_{12t} & \cdots & X_{1jt} & \cdots & X_{ijt} \\ X_{112} & X_{112} & & & & & & & & & \\ \vdots & & \ddots & & & & & & & & \\ X_{11t} & & & X_{11t} & & & & & & & \\ X_{121} & & & & X_{121} & & & & & & \\ \vdots & & & & & \ddots & & & & & \\ X_{12t} & & & & & & X_{12t} & & & & \\ \vdots & & & & & & & \ddots & & & \\ X_{1jt} & & & & & & & & X_{1jt} & & \\ \vdots & & & & & & & & & \ddots & \\ X_{ijt} & & & & & & & & & & X_{ijt} \end{pmatrix} (X_{111}, X_{112}, \cdots, X_{11t}, X_{121}, \cdots, X_{12t}, \cdots, X_{1jt}, \cdots, X_{ijt})$$

The limiting conditions will also be represented in matrix from. Since the budget is the only restriction as of now this condition will be easily formulated. If more restrictions are needed it is quite easy to fill up the A matrix with these.

$$(1{,}1,\ldots,1{,}1,\ldots,1,\ldots,1\ldots,1)\begin{pmatrix} X_{111} \\ X_{112} \\ \vdots \\ X_{11t} \\ X_{121} \\ \vdots \\ X_{12t} \\ \vdots \\ X_{1jt} \\ \vdots \\ X_{ijt} \end{pmatrix} \leq B$$

The problem can now, with the clear matrix representation easily be implemented, see Appendix D for the code.

3.8.7 MatrixVB and Visual Basic integration

To gain access to the functions from MatrixVB the DLL that contains the mathematical library must be linked to the Visual Basic. The DLL is a COM component, which means that it can be linked to all visual basic projects it does not matter what language is used as long as it is COM compatible.

When the DLL has been linked the optimization functions are now available. The input parameters, which are the matrices from section 3.8.6 can now enter the predefined functions from MatrixVB. The function that will be used is the LP – function.

`x=lp(c,a,b,lb,ub)` returns the best solution to the linear programming (LP) problem,

$$\min_x \; c^T x$$
$$Ax \leq b$$

where *lb* is the lower bound for *x* and *ub* is the upper bound. The x vector is the result vector, which is one-dimensional. This vector has to be transformed into a three-dimensional result matrix before the final results can be displayed in the result window. When some banner alternatives have a longer time span than others or when a site has more banner alternatives than others there will always be execs parameters. This means in order to still have the matrix structure some values that do not exist will have to given the value zero. These nonsense values have to be filtered out and creating the result matrix does that. A problem is that the corresponding x_{ijt} values do not assume the value zero. This is a result of the numerical truncations in the LP method.

When solving the quadratic programming problem the function `x=qp(q,c,a,b,lb,ub)` is used where the matrices c,a,b and the vectors lb and ub have the same meaning as above and the q is the matrix describing the quadratic relations.

3.9 Validation and verification

If the proposed solution of the optimal advertisement campaign planner is to be used so that marketing directors can trust and depend on the results, the model must be validated. The linear or quadratic path might for instance be insufficient when trying to simulate the arrival intensity. An even worse scenario is that the complexity of the interaction between banner ads and surfers makes it too hard to construct a dependable relation.

The proper way to go about when validating the model is for the marketing director to create an advertisement campaign without using the optimization program and then let the web site analyzing tool monitor the campaign. When the results of the campaign are known information about the conversion rates, for example a click through to a newsletter conversion, can be found. When the input and the expected output the advertisement campaign planner is known the validness of the model can be tested. The way to go about is basically filling in the fields in the interface explained in section 3.8.5 and pressing start. Since all the information exists about the campaign an optimal solution will be generated. It is not likely that the proposed optimal solution will correspond exactly to the actual real result of the completed campaign. Non-the less the solution should indicate which sites gave a good result in terms of click throughs and profit and the ones that did not. The profit is, as explained in, 3.7.4, measured in conversions, for example a page view that turns in to a member is an event that generates a profit in the model. Basically the verification lies in if the constructed model can in broad terms simulate the result of the already completed campaign i.e. when the first campaign has been run and then rerun by the advertisement campaign planner it will be known whether or not the predication made by the program was right or if the model needs to be refined. This is the first step in the validation. The procedure explained above only tests if the mathematical algorithm used can roughly simulate reality. There is another important part of the validation and that is to check the robustness of the model. After the first test has been done it is time to tweak the conversion and transition rates to see how much the optimal solution changes. If the solution changes drastically when small changes are done to the mentioned parameters the model is unstable and can therefore not be trusted. To be able to run this test the mathematical model must first be verified. Otherwise it would be like running stability checks on a non-valid mathematical theory, which is nonsense.

When it is known that the model works and it is proven that it is robust enough, it is time to start estimating the conversion rates for the next campaign. The conversion rates are not fixed for one site, they will constantly change, hence the stability verification. It must be remembered that too large deviations will always make the solution invalid and estimating these conversion rates is a great part in achieving a trust worthy solution. The more statistical data about the conversion rates the better will the coefficient estimation be i.e. running many campaigns improves the estimation of the conversion rates. When enough data is gathered, a distribution for the conversion rates can found and from this distribution it will be possible to somehow create confidence intervals indicating how probable large deviations from the value serving as the input to the model is. Different scenarios can be constructed so that the marketing director knows when the solution falls and how probable that event is, then he or she can decide on whether it is worth running the campaign proposed by the advertisement campaign planner.

Unfortunately there is no way, at the moment, to test the advertisement campaign planner and thereby verifying the implemented mathematical algorithm. The results presented in Appendix B cannot be used in tests because they lack the conversion rates, which the model relies heavily on. Further more no campaigns have been made since the work with this thesis began and no campaigns are planned in the near future.

A way to test the model without having to spend money on advertisement is to simulate a campaign. The problem is that it is impossible to estimate how the page views change over time. A decay of $t^{-1/2}$ is used at the moment. If the tests follow this relation with different conversion rates the program will generate a valid solution. It is likely that when setting up the simulation the results will only verify what was intended to be verified because the simulation is so colored by the actual problem. What this in fact means is that it is too difficult to simulate the real world because there are to many unknowns in this problem.

3.10 Conclusions

The estimated parameters are the most crucial factors in the model. As mentioned many times before in the thesis a correct solution cannot be generated without accurate input parameters. It is the marketing directors job to gather as much information as possible about these parameters before running the program. From an advertisement point of view EPO.com should always aim to advertise when shares can be bought at the site, one-time investors dominate and they are often interested in one deal.

The mathematical model that explains the flow of click throughs is a transportation problem, which is very well explained with a network representation. The problem is that the target function and constraints of a standard transportation problem is linear and in this case the function explaining the flow of visitors to the site is not known but from other similar papers [5] [6] the visitor flow is assumed to be linear. This has not been verified and due to the lack of robustness in the linear model another functional relation has been sought. Therefore another model, a QP (quadratic programming) representation has been created that makes the target function non-linear. For this representation to hold it is important to interpolate this target function to the data of recent campaigns to land at a correctly estimated quadratic matrix. The quadratic matrix describes all the quadratic combinations possible between the variables that will be minimized, in this case the money spent. Since no efficient monitoring tool has been used earlier together with campaigns there is no reliable data and this interpolation cannot be preformed.

If the linear model could be made more robust, meaning less sensitive to perturbations in the estimated parameters especially the click through rate, it could hold as a good approximation. Unfortunately this cannot be done within the time span of this thesis. Instead the QP model is used which evens out the money invested over more advertisement alternatives making them dependent of each other, in other words it penalizes large amounts spent on one advertisement alternative. In the real world this is probably the case when advertising to the same target group on different sites.

The restrictions in this thesis have been the implementation language. MatrixVB is not typically designed for solving advanced optimization problems and it lacks some tools that would have been very useful, especially the ability to create a non-linear target function without having restrictions due to the implementation language.

It is hard to estimate the click stream because there are an infinite amount of parameters that it depends on. The solution generated will always relate to the past and therefore the solution will hold if the difference between the past and the present in terms of economic climate is similar. If it is not the uncertainty in the model is too great and the program solution will not be valid.
If a more sophisticated model would be implemented the first step would be to make it stochastic linear problem. The uncertainties in the click throughs would then be included and a more trustworthy solution would be generated.
All in all, the implemented program is very hard to measure in terms of accuracy because enough statistical data is not available. Due to the nature of banner ads it is possible to explain, with the proposed mathematical theory, the flow of visitors. What relation this flow goes is difficult to estimate. A linear and a non-linear program have been introduced and it will be known after enough campaigns have been run whether or not these descriptions hold.

3.11 Future work

There are two ways to go if the model is to be improved, a better linear model or a totally new non-linear target function. A new non-linear target function would explain the advanced relations in terms of click throughs compared to time the banner has been up on the site and penalizing functions that restricts better banner alternatives to grow unrealistically large, thereby simulating statistical uncertainties. This methods is the most time consuming and not as interesting as improving the linear model. As mentioned in section 3.10 a stochastic linear programming problem can be created that incorporates the drawbacks that the normal linear problem has.

3.11.1 Stochastic programming

Stochastic programs are mathematical programs where some of the data incorporated into the objective or constraints is uncertain. Uncertainty is usually characterized by a probability distribution on the parameters. In this thesis it is typically the click through probability. Although the uncertainty is rigorously defined, in practice it can range in detail from a few scenarios (possible outcomes of the data) to specific and precise joint probability distributions. The outcomes are generally described in terms of elements w of a set W. W can be, for example, the set of possible click through probabilities over the next few days or over different banners.

When some of the data is random, then solutions and the optimal objective value to the optimization problem are themselves random. A distribution of optimal decisions is generally unimplementable. Ideally, generating one decision and one optimal objective value would is wanted.

One logical way to pose the problem is to require that we make one decision now and minimize the expected costs of the consequences of that decision. This is called the *recourse* model. Suppose x is a vector of decisions that we must take, and y(w) is a vector of decisions that represent new actions or consequences of x. Note that a different set of y's will be chosen for each possible outcome w. The *Two-Stage* formulation is:

$$\min\ f_1(x) + E\left[f_2(y(w), w)\right]$$

$$\text{Subject to: } \begin{array}{l} g_1(x) \leq 0, \cdots, g_m(x) \leq 0 \\ h_1(x, y(w)) \leq 0, \forall w \in W \\ \vdots \\ h_k(x, y(w)) \leq 0, \forall w \in W \\ x \in X, y(w) \in Y \end{array}$$

The set of constraints h1 ... hk describe the links between the first stage decisions x and the second stage decisions y(w). Note that we require that each constraint hold with probability 1, or for each possible w in W. The functions f2 are quite frequently themselves the solutions of mathematical problems. We don't want to make an arbitrary correction to the first stage decision; we want to make the best such correction.

Recourse models can be extended in a number of ways. One of the most common is to include more stages. With a multistage problem, we in effect make one decision now, wait for some uncertainty to be resolved, and then make another decision based on what's happened. The objective is to minimize the expected costs of all decisions taken.

The problem in this thesis is to find these scenarios and see how they affect each other. That it is not an easy task. Like any other parameter they will have to be found out through statistical data. When the different scenarios have been created containing the possible click through probabilities the solution will be more robust and the simplicity with the linear problem has been kept without neglecting precision.

4 Website analyzing tool

4.1 Introduction

The concept of analyzing website traffic is not new and there are plenty of tools available on the market that can deliver simple statistical visitor data from analyzing the log files. To be able to know more detailed information about the visitors like how many times a particular visitor visited the site before becoming a member etc. one has to solve the problem of how to recognize a visitor. This cannot be done by just analyzing the log files like the existing analyzing tools do. The idea is to create a tool that can serve as a necessary complement to one of the existing tools and thereby be able to give a detailed statistical report about the visitors and their actions on the website. This part of the thesis will present some work that has been done by others, the Website analyzing tool that was developed in this thesis project and also some future work that can be done to make the tool better.

4.1.1 Goal

The goal of the project is to deliver a program that can give a more detailed analysis of the visitors. This is no attempt to make a better log analysis program than the ones available on the market today. Probably there will be a need for a complementary program that handles the regular analyzing of log files. The Log File Analyzer will concentrate on tracking visitors and give information about who has been here before and what they have done on the site. It will also produce statistical data that can be used by a marketing director to fine-tune the input data to the optimization model described in part three of the thesis.

4.2 Theory and related work

This chapter concentrates on some of the concepts and techniques that are common within the log file analyzing area. It also contains a small survey on some existing software available for website analyzing.

4.2.1 Web servers and Log files

A web server is typically just a normal computer equipped with special server software. If the site is large the administrator often use more than one web server and traffic is divided between the servers using a so-called load balancing software [19]. That means that traffic is distributed between the servers depending on the current traffic load. EPO.com uses two Compaq multi processor computers with load balancing and Microsoft Internet Information Server (IIS) software.

In order to get any statistics about the intensity of the web server traffic it is needed to keep a log of the web server activities. Even though the Internet is anonymous by nature it is still possible to learn something from the visitor of the site.
When a client makes a request for some data the web server logs information about the client and what file the client requested. Below is a list of some of the things that’s possible to learn from the visitor [14][18].

Date and time of the hit	Visitor's IP address
Name of the host	Path of the file served
Request	Bytes transferred
Visitor's login name (if the user is authenticated)	Visitor's host (if the visitor's IP address can be translated)
Referrer (if user was linked from another site)	Cookies sent by the visitor
Visitor's user agent	

From this information it is then possible to make some conclusions like:

Number of requests made ("hits")	Browsers and versions making the requests.
Number of unique visitors during a period	URLs from which user came to the site (referring pages)
Number of requests by HTTP status codes (successful, failed, redirected, informational)	Totals and averages by specific time periods (hours, days, weeks, months, years)
Number of requests for specific files or directories	

There is a lot of uncertainty in these data and more of the problems regarding the correctness of the log files will be discussed in chapter 4.2.6.

4.2.2 Traffic analysis

There are basically two techniques used when analyzing the web site traffic:

- Analyzing the server access log files
- Sniffing the network traffic

Programs that use the server access log are the most commonly used. Every web server keeps a log of all requests that it gets. There are basically two types of log file formats [15]: CLF (Common Log file Format) and ELF (Extended Log file Format). There are also some other platform dependent formats like the Microsoft IIS log file format.

The other technique of website analyzing is the one that sniffs the packets on the network traffic. The packet sniffing technology has recently been introduced into the traffic analysis market and eliminates the need to collect and store data in large log-files. It works by opening the TCP/IP packets that go in and out of the web server and look at the content information. This is done in real-time which gives the user an instant snapshot of the website traffic. This solution demands that the sniffing program must be installed on a computer on the same network as the web server, which makes it unsuitable for all systems.

4.2.3 Existing tools for analyzing traffic

There are piles of different log file analyzing software available on the market today. Some are very simple and others more complex. Below are just three of the programs available:

- WebTrends Log Analyzer from Webtrends corp (www.webtrends.com).
- Sawmill 5.0 from Flowerfire (www.flowerfire.com).
- Basic Traffic Reporter 02.00.00 from HouseholdVentures.com (www.householdventures.com).

Below is a small evaluation of these three analyzers.

WebTrends LogAnalyzer 6.0

This is an almost complete tool for analyzing the website traffic. It gives the user a wide variety of statistical data to choose from. The user interface is very nice and easy to understand. The user can decide in what format the report should be in (html, ASCII, excel, word, delimited) and get graphs and diagrams that show the statistics in a very nice way.
The program delivers statistics about:

General statistics, no. of hits, page views etc.	**Activity statistics**, when and for how long visits are etc.
Resources accessed, requested pages, top entry/exit pages etc.	**Technical statistics**, client and server errors etc.
Advertising, views and clicks.	**Referrers & Keywords**, referrers, top search engines with keywords etc.
Visitors & Demographics, new/old users, top geographic regions.	**Browsers & Platforms**, info about the visitors platforms.

In all, WebTrends LogAnalyzer is a very complete log file-analyzing tool. Apart from the fact that it can not make any deeper analysis the only drawback is that it feels like the program is generating almost too much statistics making it hard to understand what is really being measured. Otherwise, it is the best program of the ones that have been tested in this thesis. The price of this program is currently $499.

Sawmill 5.0

This is a simple log file analyzer from Flowerfire that produces log file reports as html pages. To use the program one has to start a server on the computer that can be reached from your ordinary web browser. From there you can import a log file and analyze it and produce a report in html. The program has all the basic statistics and displays them with graphs and colorful diagrams. It is not as detailed and "good looking" as the log analyzer from WebTrends but it serves a purpose for the not so demanding web site manager. The full version retail price is $200.

Basic Traffic Report 02.00.00

This program is not good. Most log file analyzers recognize the format of the log file and adjust to the type. Something goes wrong when this program tries to process the log file that was used in the test, a good guess is that it can’t recognize the format of the log file and therefore it generates an error. This is not the only example of not so good log file analyzers. There are more examples of programs that has the same kind of problem. The only excuse is that they most often are free of charge.

4.2.4 Problems when analyzing log files

Analyzing log data seems simple enough but there are some problems. Some request to the web server never get to the log file and others should not be in the log file. Some of the problems [17] are:

Problem	Cause
Events are not registered in the log file.	If a web page already exists in the **cache** on a computer the browser will retrieve the page from the cache instead of making a request to the web server and thereby the event will never be registered in the log file. **Proxy servers** also use caching and will check to see if a page is in the cache before sending the request for a page on to the server.
Different clients have the same IP address.	Computers behind a **firewall** all look the same to the web server (same IP address) and this can lead to some problems when looking at IP addresses because of the fact that IP addresses may not be unique. An IP address is not a unique identifier for an individual visitor. A lot of people use a modem connected to an account at their ISP (Internet Service Provider) to access the Internet. Most ISPs use a dynamic distribution of such IP addresses. This means that the next time a user connects to the ISP it will most likely receive another IP address. This makes it impossible to use IP addresses to make a reliable identification of a certain visitor.
Data that should not appear in the log file.	Search engines on the web have programs so-called **Robots** or **Spiders** that search the Internet for information. These are programs and should not be considered as visitors but they do appear in the log file and therefore must be disregarded when making the analysis.

4.3 Task specification

Originally the idea was to create a tool for learning some basic visitor statistics that would serve as input and feedback to the optimization tool that is described in part three of the thesis. After brainstorming, it became clear that it would be more interesting to have a tool that could give a more detailed picture of the visitors and what they did on the site. Especially to know what went on before they decided to become investors at EPO.com.

4.3.1 Idea of solution

Figure 1 in section 1.3 shows a figure describing how the advertisement-effect model and the web site analysis tool work together. The web site analyzer gets input from the log and returns output that can be used to adjust the advertisement-effect model and also give direct feedback to the partner sites.
One important aspect in this project is that the analyzing tool must not disturb the daily traffic on the site. That is why the decision was made to use an *event-driven model*. That means that the analyzing program is not active all the time but that there are some events that could trigger the program to perform its tasks. For example when some one decides to become a member on the site, the last stage in the sign-up process triggers a database call that links the investor to a visitor in the database.
Below are three events and a description of what happens when they are triggered.

Event 1. “Visitor request”
The visitor sends a request for an asp-file to the EPO web server. Along with the request it also sends browser data such as browser version, cookies, referrers etc.
The web server gives the visitor a SessionID and checks the visitor for any cookies. If no CookieID is found, the visitor is given one. This CookieID is used for identifying the visitor so that a trace of the visitor’s activities later on is possible.
The web server then sends the requested file to the client and writes to the log. The CookieID of the visitor can then be traced from the cookie-field of the log file.
The checking and setting of CookieID’s is only done once per session and does not affect the system in any negative way.

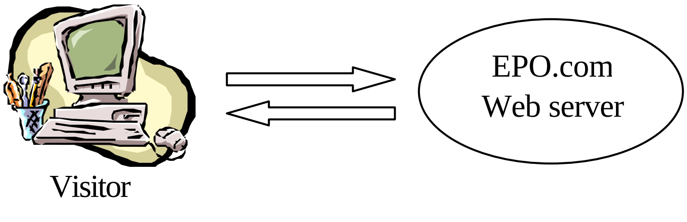


Figure12 The first time a visitor comes to the EPO site it will be marked with a CookieID making it possible to trace the visitors activities from the log file.

Event 2. “The visitor becomes a member”
Now the visitor has decided to become either a real member of EPO.com or a newsletter member. The last stage in the sign-up process triggers a call to a Stored Procedure in the SQL-database that links the visitors CookieID to the new InvestorID.

Now we can identify the visitor both as an investor in the system and as a visitor in the log file.

Event 3. "Analyzing the data"

Before any analyzing of the data can be done the log files are reduced from redundant and non interesting data and is then stored into a database as shown in the figure 13. The program called the Log File Stripper extracts only the interesting parts from the log-file and creates a new log file containing the stripped data. The program then puts the data into an SQL database that is indexed on the visitors cookieIDs. Storing the logged data in the database will add flexibility to the analyzing program and makes it easier to search for data on a certain date, CookieID, referrer etc.

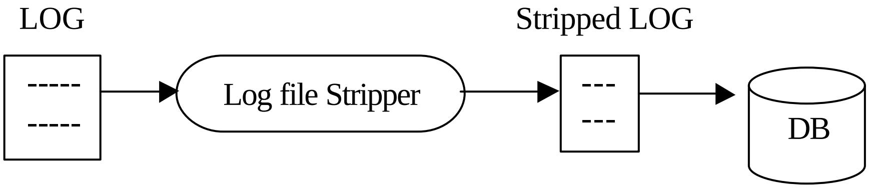


Figure13 Stripping the log-files and storing into the database.

When the log files are processed and stored in the database its possible to start analyzing the data. As shown in figure 14 the user can specify between what date and time he/she would like to analyze. The user could either make a general analysis that looks for all visitors during the chosen period or make a more detailed search for a particular type of visitor. The program then makes a SQL-call to the database and asks it to return all data between the specified dates. The program will give information about the visitors such as: number of visitors, referrers, search engines, new investors, and also detailed investor information etc.

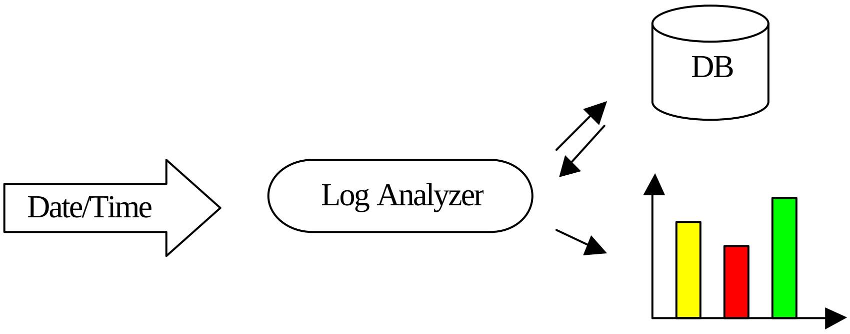


Figure14 Log Analyzer workflow

A more detailed description over the Log File Stripper and the Log Analyzer is given later on in this chapter.

4.4 Implementation

This section describes the implementation of the Log File Analyzer and the Log File Stripper and some of the problems and solutions that occurred during the project. The source code is found in the appendix.

4.4.1 Cookies

One big problem that has to be faced when trying to keep track of visitors that have not yet been authenticated through a log-in process is how to recognize them when they revisit the site. One approach could be to keep track of the IP-address and recognize the visitors in that way. This is as previously described not reliable because two different individuals could have the same IP-address and thus cause erroneous conclusions to be made in the analyzing process.

Cookies [16] were developed to help site builders overcome the anonymous nature of the Web. The technology enables developers to stash a user ID, a SessionID, or some other bit of identifying data on the user's machine. That makes it possible for developers to get a sense of whom they're dealing with and what path the user is taking through the site. Cookies could be either long-term or short-term. SessionIDs are short term. They are often only valid for about 15 min. The CookieID that is used in the project is long-term and makes it possible to recognize a visitor on its return to the EPO site. This is possible to do as long as the visitor does not remove any cookies from the browser.

So when the user reaches the EPO site a function looks for a cookie on the client browser and if it can’t find any it sets one. This so-called CookieID is a combination of the SessionID and the current date and time.

4.4.2 The Log File Stripper

A problem with regular log file analysis software is the time it takes to analyze large log files. At EPO.com, a log file could be as large as 15 Mbytes for just one day. Imagine the amount of log data that has to be gone through to analyze log files for the last month. Also since the EPO server produces one log file per day it would be difficult to analyze over a longer period. That is why the decision was made to store the log data into a database. This will give the Log Analyzer software a flexible and fast access to the log data and better possibilities to search over large amounts of data and still be very fast.

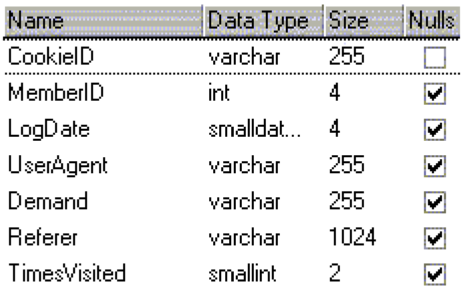
Every hit the web server gets is stored into the log file. This has the consequence that the log file is loaded with info about requested images, style sheets and other “non interesting” requests. This information could be useful in some applications but not in this and that is why it is needed to strip the log file from unnecessary data.

The program is called the Log File Stripper and is implemented in Visual Basic. It is surprisingly fast and it reduces the size of the log file to about 200Kbytes instead of the 15 Mbytes it was before stripping it.

So the idea is to only store one row from each unique visitor. Then you face the first big problem: How do you find a unique visitor? You could check for IP address but this is as previously mentioned not a reliable way because of how the ISPs work. The easiest way is to use the method of SessionIDs that the IIS gives to each visitor. This

is actually very simple. When a visitor surf to the site the server checks the client for any site-related cookies. If it can't find any SessionID it plants a cookie called SessionID on the client browser. This cookie contains a number that no other user has. The cookie is only valid for 15 min if the visitor does not update it by requesting another page from the server. This cookie enables the system programmers to separate the visitors from each other and thereby store important user-specific variables on the server. When the server writes info to the log file the SessionID is then written to the cookies field.
One very important piece of information is the referrer-field, i.e. the field that tells where the user came from before he/she visited the EPO site. The problem is that when the visitor clicks on links on the EPO site the referrer is changed to the EPO domain. Since the program only extracts one row from every unique visitor it is important to find the row that says where the visitor came from originally.

The figure below shows the database table and the entries that are put into the database for each row.



Name	Data Type	Size	Nulls
CookieID	varchar	255	□
MemberID	int	4	☑
LogDate	smalldat...	4	☑
UserAgent	varchar	255	☑
Demand	varchar	255	☑
Referer	varchar	1024	☑
TimesVisited	smallint	2	☑

Figure15 Database table

CookieID: This is the identification number that every visitor gets when he/she first visits the site. This data retrieved from the Cookies field in the log file.
MemberID: This is the unique identifier every investor has in the back-office system. This field is currently not used.
LogDate: The date of the first visit. Retrieved from the date-field in the log file.
Demand: The requested file. At this time not used. Retrieved from the request-field in the log file.
Referrer: The referring page. Retrieved from the referrer-field in the log file.
TimesVisited: Shows how many times the visitor has visited the site since the first logged time. This is incremented every time the row is updated.

The idea is to only have one row for each visitor in the database and update this row when the client returns. Therefore is it needed to first do a search through the database to see if the CookieID exist. To do this in a fast way is it appropriate to create a so called Stored Procedure that first checks the database for any occurrences of the CookieID and updates the entry if it finds it or else creates a new entry for that visitor. A Stored Procedure is just a function in the database that can receive parameters and execute SQL statements. The big advantage is that it is easy to implement logical statements such as IF-statements and so on.

4.4.3 Log File Stripper implementation details

The Log File Stripper Algorithm works as follows:

```
1. While Not EndOfFile (log file)
        a. Read line from log file
        b. Parse out the SessionID from the line read
        c. Check if SessionID exists in collection (if already collected)
                i. If Yes, RETURN to a.
               ii. If NO, Insert into collection and RETURN to a
2. End Of While
3. Insert Collection into database
```

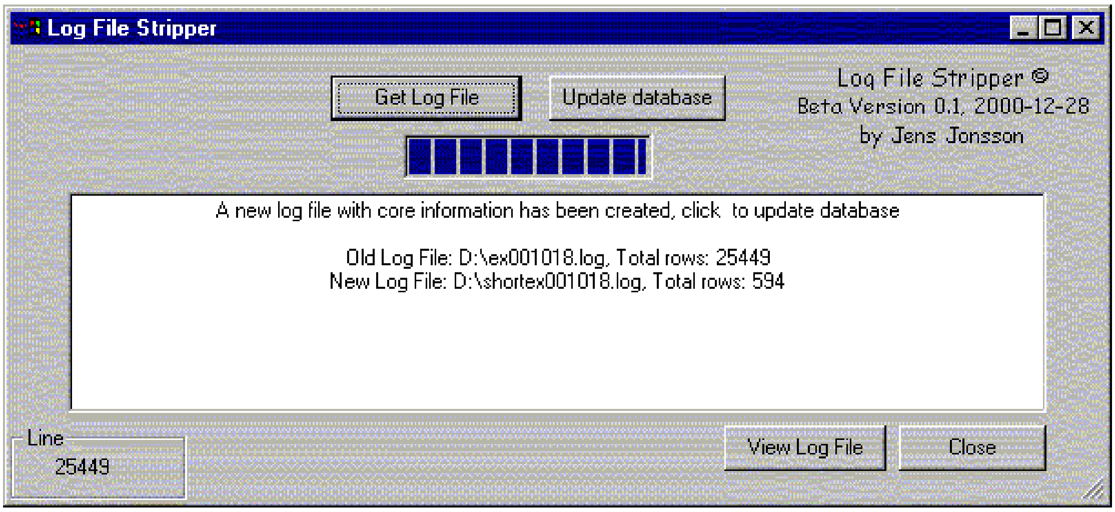
Below are the main VB-functions with a description of their input and what they do.

Function	Description
GetLogData (String)	This function reads from the log file and then calls the search function to make sure only unique visitors get printed in the new log file that is created. The input is a string with the file-path to the original log file. The function also handles some display functions such as the progress bar etc.
Search (String, Long, Dictionary) **returns** Boolean	This function searches through a Dictionary of unique visitors. It returns true if the line already exist in the Dictionary. The first input is a string containing a line from the original log file. The second input is a long containing the number of the row in the log file. The third input a reference to the Dictionary of unique visitors.
DataInsert ()	This function reads from the new log file and insert data into the database by performing a call to a stored procedure on the SQL-database.

The source code for the Log File Stripper can be found in Appendix E in the back of the thesis.

4.4.4 Log File Stripper GUI

The Log File Stripper user interface is very simple. The main form has buttons for opening a new log file, updating the database with new log data and a progress bar that gives an approximation of the time left to process. Figure 15 below shows the main frame after a successful run.



Log File Stripper ©
Beta Version 0.1, 2000-12-28
by Jens Jonsson

A new log file with core information has been created, click to update database

Old Log File: D:\ex001018.log, Total rows: 25449
New Log File: D:\shortex001018.log, Total rows: 594

Line
25449

Figure16 Log File Stripper

Figure 16 shows the Load Log File frame. All files with the extension .log will be shown. When a log file is chosen and the "Analyze" button is pressed the analyzing process is started.

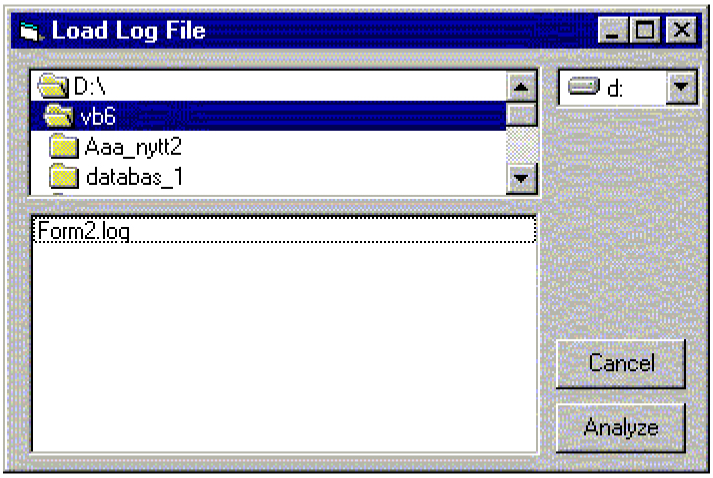


Figure17 Load Log File Form

4.4.5 The Log Analyzer

The Log Analyzer program is the main implementation part of this thesis. The purpose of it is to give general statistical data about the visitors of the EPO site and also a more detailed analysis that can serve as input to the "advertisement-effect model".

The idea isn't that it should compete with the existing log file analyzing software that is available but rather be a complement that gives information that no ordinary analysis tool could ever give.

4.4.6 Log Analyzer implementation details

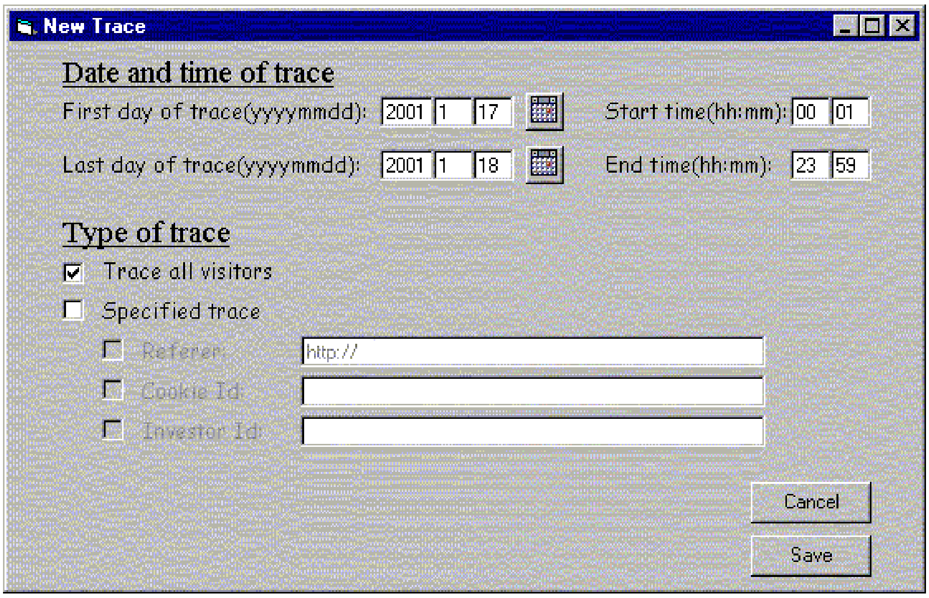
The Log Analyzer program is implemented in Visual Basic and consists of several different Forms with their own functions. The Forms and their main functions are listed below.

Form name	Function and description
Analyzer Form – Main form	**LoadFile** (String) Reads from the log file and stores the data in a dictionary. A call is then made to the **LoadProject** (Dictionary) function in the TraceForm form.
Calendar Form 1-4 – Shows a calendar	**MonthView_DateClick** (Date) This function is called when a date is clicked in the calendar. When this is done it calls the **SetStartDate** (Date) and **SetEndDate** (Date) functions in the Calling forms.
NewTrace Form – Trace information	**SetStartDate** (Date) Fills in the date boxes with Year/Month/Day information in the Date input parameter. **SetEndDate** (Date) Fills in the date boxes with Year/Month/Day information in the Date input parameter. **Event handling functions** Functions that handles events like when a check box is pressed etc.
Save Form – Stores the Newtrace form information in a file	**Event handling functions** Different functions that handles events when buttons are pressed, ex. Saves the trace information in a chosen drive/directory/file.
OpenFile Form – Open a saved trace project	**Drive_Change** () Function that sets the chosen drive **Dir_Change** () Function that sets the chosen directory **File_dblClick** () Makes a function call to the **LoadFile** (String) function in the Analyzer form with the chosen drive/dir/file.
Referrer Form – Known referrers listing	**Form_Load** () Function that prints all known referring

	sites from file and adds new referrers to the file as they are added.
SearchEngines Form – Known search engines listing	**Form_Load** () Function that prints all known search engine sites from file and adds new search engine sites to the file as they are added.
TraceForm Form – Analysis result listing	**LoadProject** (Dictionary) This function reads from the LAT-file and fills in the project data box. It also makes a call to the **GetLogData**(String, String) function. **GetLogData** (String, String, Opt. String, Opt. String, Opt. String) This function handles almost all of the analysing of the log data. It makes connections to the database and produces all kinds of statistical data that is displayed in the program. **GetInvestorData** (String, String, Dictionary, Dictionary) **return** Bool. Function that retrieves all visitors that has become investors during the analysing period. **Event handling functions** All the functions that handle events, such as button handling and check boxes etc.

4.4.7 The Log Analyzer GUI

When creating a new trace-project in the Log File Analyzer, a so-called LAT-file is created. LAT stands for Log Analyzer Trace and the file specifies the range and type of the analysis. The advantage of creating such a file is that it will be easier to do follow-ups on a special trace. The lat-files are just plain text files with the .lat extension. Figure 18 shows the “New Trace” question box.



New Trace

Date and time of trace

First day of trace(yyyymmdd): 2001 1 17 Start time(hh:mm): 00 01

Last day of trace(yyyymmdd): 2001 1 18 End time(hh:mm): 23 59

Type of trace

☑ Trace all visitors

□ Specified trace

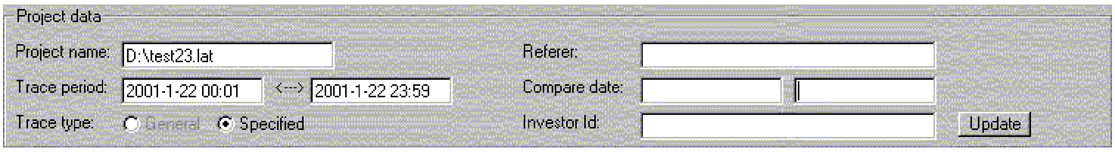
□ Referer: http://

□ Cookie Id:

□ Investor Id:

Figure18 The New Trace form

After opening a trace project a box called “Project data” appears (see figure 19). This box contains the information in the LAT-file. It’s possible to change the information here and pressing the update button will resave the changes to the LAT-file.



Project data

Project name: D:\test23.lat

Trace period: 2001-1-22 00:01 <--> 2001-1-22 23:59

Trace type: ○ General ● Specified

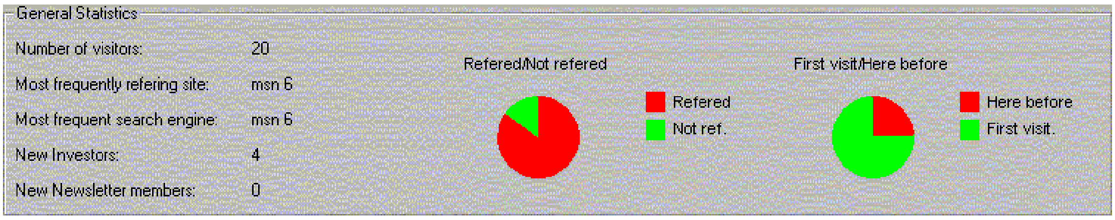
Referer:

Compare date:

Investor Id:

Figure19 Project data form

Below the “Project data” is the “General Statistics” box. This box gives some basic statistic data such as: number of unique visitors, most frequent referring site, number of new investors etc. It also shows two graphs. The first tells how many visitors came from another site versus how many that came directly to the EPO site. The second shows the ratio between visitors who have been at the site before and visitors that came for the first time.

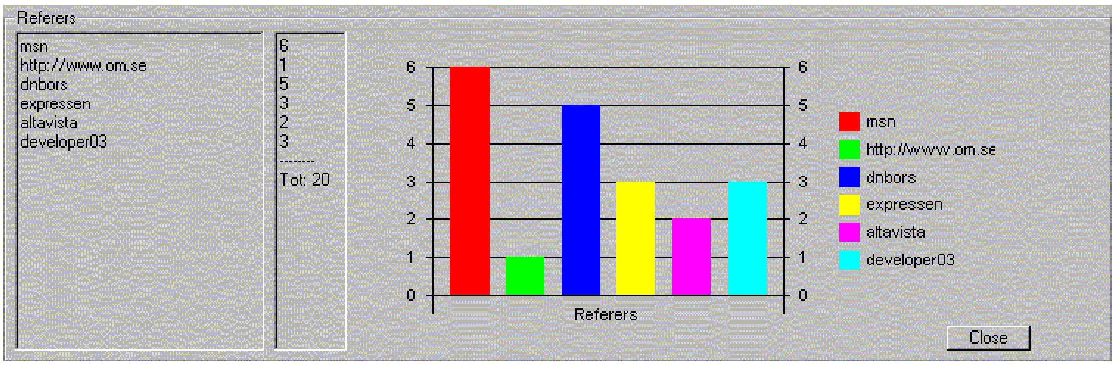


General Statistics

Number of visitors:	20
Most frequently refering site:	msn 6
Most frequent search engine:	msn 6
New Investors:	4
New Newsletter members:	0

Figure20 General Statistics form

Figure21 Figure 21 shows the "Referrers" info box. This box shows what referring sites that the visitors came from and how many that came from the referring sites. To make the analyzing more reliable, the referring sites have to be listed in a special file used by the program when determining from where the visitor was referred.

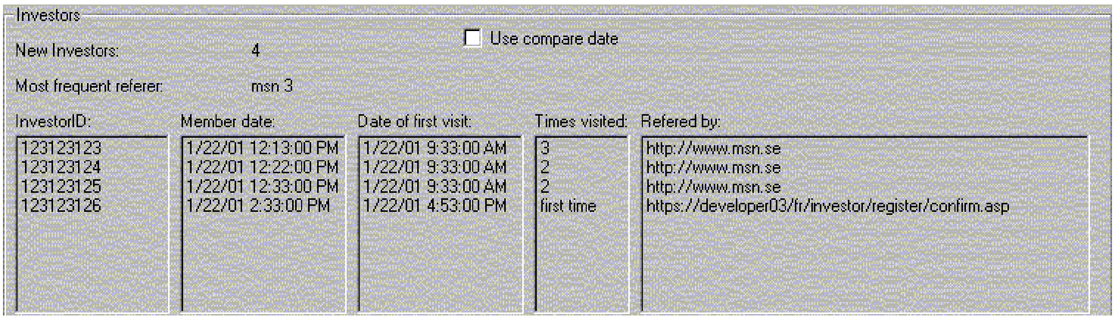


Referers

msn	6
http://www.om.se	1
dnbors	5
expressen	3
altavista	2
developer03	3
	------- Tot: 20

Figure22 Referrers form

The "Investors" box contains information about what and how many visitors became investors during the analyzed period. It also shows the original referrer, date of visit and the number of times the user visited the site before becoming a full member.



Investors

New Investors: 4

Most frequent referer: msn 3

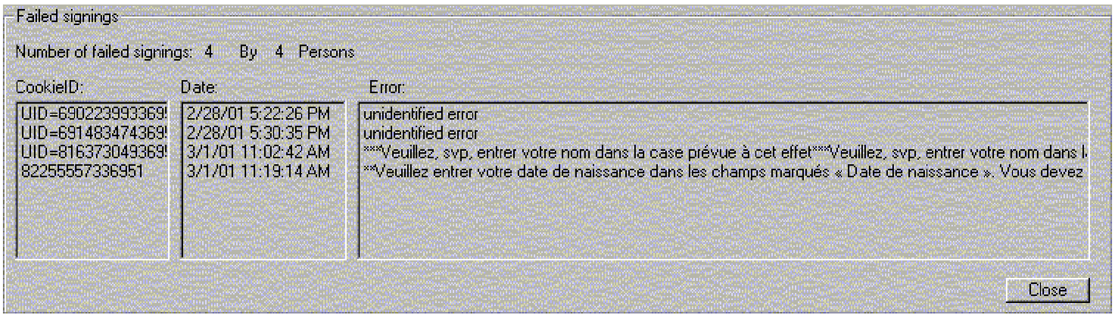
□ Use compare date

InvestorID:	Member date:	Date of first visit:	Times visited:	Refered by:
123123123	1/22/01 12:13:00 PM	1/22/01 9:33:00 AM	3	http://www.msn.se
123123124	1/22/01 12:22:00 PM	1/22/01 9:33:00 AM	2	http://www.msn.se
123123125	1/22/01 12:33:00 PM	1/22/01 9:33:00 AM	2	http://www.msn.se
123123126	1/22/01 2:33:00 PM	1/22/01 4:53:00 PM	first time	https://developer03/fr/investor/register/confirm.asp

Figure23 Investors form

When a visitor becomes a member there are some information the visitor has to submit such as name, address, account numbers etc. This information is then processed by the system and if the input is correct the visitor gets a InvestorID and get access to an account on the EPO server. If the submitted data is erroneous in some way the visitor will receive an error message telling the visitor what was wrong. This information is valuable to know because it tells the system designers something about how difficult the infrastructure is to understand. Figure 23 shows the "Failed

Signings” form. This form shows the visitors that failed to become members, date and the type of error.



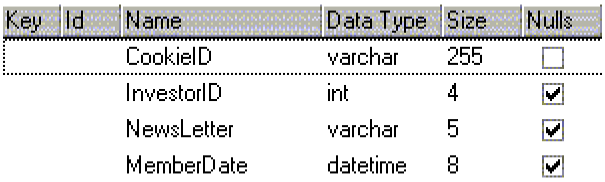
Failed signings

Number of failed signings: 4 By 4 Persons

CookieID:	Date:	Error:
UID=690223993369	2/28/01 5:22:26 PM	unidentified error
UID=691483474369	2/28/01 5:30:35 PM	unidentified error
UID=816373049369	3/1/01 11:02:42 AM	***Veuillez, svp, entrer votre nom dans la case prévue à cet effet***Veuillez, svp, entrer votre nom dans l
82255557336951	3/1/01 11:19:14 AM	**Veuillez entrer votre date de naissance dans les champs marqués « Date de naissance ». Vous devez

Figure24 Failed Signings

To know which visitor became a member a link has to be made between the InvestorID and the visitors CookieID. This is done with a database call in the last stage of the sign-up process when all the input processing is done. Below is a figure of the database table.



Key	Id	Name	Data Type	Size	Nulls
		CookieID	varchar	255	☐
		InvestorID	int	4	☑
		NewsLetter	varchar	5	☑
		MemberDate	datetime	8	☑

Figure25 Database table VisitorToMember

4.4.8 Database and Store procedures

The database used to store visitor and investor information is a Microsoft SQL server database [22]. Since this platform is used in the other parts of the system it was obvious to use it in this thesis also. Below are the tables and store procedures that is used by the Log File Stripper and Log File Analyzer listed along with a short description of what they do.

Table name	Fields	Data type
VisitorLog – Contains all the data that is extracted from the log file	CookieId MemberID LogDate UserAgent Demand Referrer TimesVisited	Varchar 255 Int 4 SmallDate 4 VarChar 255 Varchar 255 Varchar 255 SmallInt 2
VisitorToMember – Visitors that has become members at EPO.com.	CookieID InvestorID NewsLetter MemberDate	Varchar 255 Int 4 Varchar 5 DateTime 8
VisitorToMemberFail – Visitors that has attempted	CookieID FailDate,	Varchar 255 DateTime 8

to become members at EPO.com but failed.	Error,	Varchar 3999

Store Procedures:

Name	Description
VisitorLogGet **(StartDate, EndDate)**	Retrieves everything between the Start and EndDate.
VisitorLogGetInvestor **(CookieID)**	Retrieve everything from a visitor with the specified CookieID.
VisitorLogInsert **(CookieID, LogDate, UserAgent, Demand, Referrer)**	Inserts a new row into the VisitorLog table.
VisitorToInvestor **(CookieID, InvestorID, NewsLetter, MemberDate)**	This SP is called when someone becomes a member. Visitor data and member data is linked in the VisitorToMember table.
VisitorToInvestorGet **(StartDate, EndDate)**	Retrieves all that became members during Start to EndDate.
VisitorToInvestorFail **(CookieID, FailDate, Error)**	Inserts visitor data into VisitorToMemberFail table when an attempt to become a member fails.
VisitorToInvestorFailGet **(StartDate, EndDate)**	Retrieves all erroneous attempts to become members during Start to EndDate.

4.4.9 Problems and solutions

Below are some of the problems that arose during the development of the thesis project and how they were solved.

IP-addresses: One might think that IP-addresses are unique and that you could use them to pin point an individual user. This is of course not true. First of all it's impossible to know who is sitting behind a computer unless some kind of authentication (log-in) has been done. Second, many people are connected to the Internet using a modem calling their ISP (Internet Service Provider). The ISPs have a limited amount of IP-addresses and to use them as effectively as possible they are shared by all their clients. This means that several people can have the same IP-address. So instead of using IP-addresses as an identifier, the system uses cookies and the built-in identification method in the Microsoft web server (SessionID) to uniquely identify a user.

Cookies: Using cookies as a way of identification is very common on sites today. Cookies are data located on the client machine that is sent by the browser to the server when the client requests pages from the web server. The cookies are saved as text strings and do not execute anything on the client machine, they are only used by the system designers to store data about the client on the client machine.
The first problem is the same as with IP-addresses, it's still impossible to know who's sitting behind the computer without proper authentication (log-in), the cookie only says that someone using this computer has visited the site before. The second and most severe problem is that cookies can and probably will become erased from the client hard drive at some point. When this happens, the client will receive a new CookieID by the system when he/she revisits the site. This will lead to erroneous data in the analysis process hence the same person will become two independent visitors in the log file. The third problem is the fact that the client can choose not to accept cookies from the web server.

Unique visitors: With the CookieID that the system distributes to all new visitors it is easy to recognize a visitor on his/hers return to the site. When analyzing web site it is important to decide what a unique visitor is. The general opinion is that if a visitor is inactive for more than 15 minutes he/she will be regarded as a new visitor when revisiting the site. With the help of the SessionIDs that the Microsoft web server distributes to all visitors there is no problem determining if the user is unique. The SessionID is actually a cookie on the client machine. This cookie is only valid for 15 minutes and if the SessionID is missing or not valid the visitor will regarded as a new visitor. This way the visitor will be recognized in the log file by the Log File Stripper on all occasions during the same day.

4.5 Testing and typical usage

This chapter describes some basic testing that has been performed to show the usefulness of the analyzing tool and also give a better understanding of how the tool is used.

4.5.1 Problems with testing

To test the analyzer tool leads immediately to some problems that has to be solved. To make a live-test on the EPO web server, which would be the best way to get good and correct input data, one have to make some changes to the current web architecture. The Analyzer tool is as previously mentioned event driven. This means that some event has to trigger the analyzer to perform a task. Examples of such events are when new visitors arrive or new members sign-up. These events trigger a function in the VB-layer to store information about these visitors into the database. Since these functions include storing data on the client machine in the form of cookies it has been decided that more testing off-line must be done before any tests can be performed on the live web server.

The only way is therefore to test the program on a server that has an exact replica of the live content but that is located on the EPO intranet and not available for the public. A test script has to be developed that captures the main characteristics of an EPO visitor. What the test primarily should show is the usefulness of the Log File Analyzer and how it can help the people responsible for advertisement and web site development. The visitors in the test are all imaginary and one person will play all the parts. So the result regarding the behavior of the visitors must not be taken too seriously since it is all imaginary persons.

4.5.2 The test

To make the test as real as possible a test script has as previously mentioned been developed. In this document 15 different user profiles are described (see Appendix E). The Idea is that the person making the test should try to think and act in a way that the type of person described in the test script would when he/she visits the EPO site.

The person performing the tests has free hands to make the kind of decisions that he believes the persons described in the script would make when visiting the EPO site. The visitor could visit the site several times, become a member, participate in current offerings, be referred from different sites or none of these things.

The test script include these important factors:

Age
Gender
Yearly income
Total wealth
Previous internet experience
Personality description

When all 15 persons have visited the site and performed its actions the Log File Stripper will remove all redundant information in the log file and store the data in the database. When this is done the Log File Analyzer can perform an analysis of the visitor data and give some more detailed information about the visitors.

4.5.3 The test server

The test server used in the experiments is a Compaq Deskpro stationary computer, 500 MHz, 132 MB RAM running Windows NT Server. It has also been equipped with the Microsoft Internet Information Server (IIS), MS Certificate Server and MS Transaction Server. Apart from the standard software running on the server the whole EPO-web has also been copied and installed on the test server. This includes both ASP-pages and Visual Basic components
Some other web servers have also been created. These servers are meant to symbolize other referring sites and search engines. Their sole purpose is to redirect traffic to the test server. Figure 26 below gives an overview the test site and the different paths that the visitor can take to get to the test server.

1: The visitor
2: Ref. site www.spray.com
3: Ref. site www.barclays.co.uk
4: Ref. site www.dn.se
5: Search site. www.yahoo.com
6: Search site. www.altavista.com

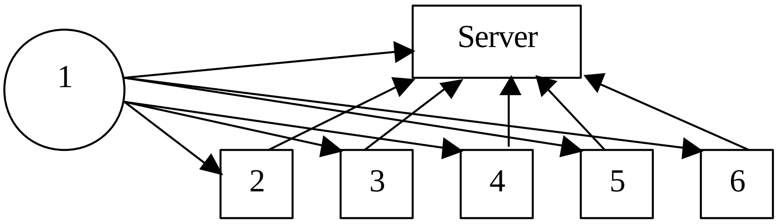


Figure26 The test site

As shown in figure 26 the visitor could either go directly to the test server or be redirected from a referring site or a search engine. If the visitor is redirected from another site it will show in the log file and the Log File Stripper will write the name of the referring site to the database.

4.5.4 Implementation and Results

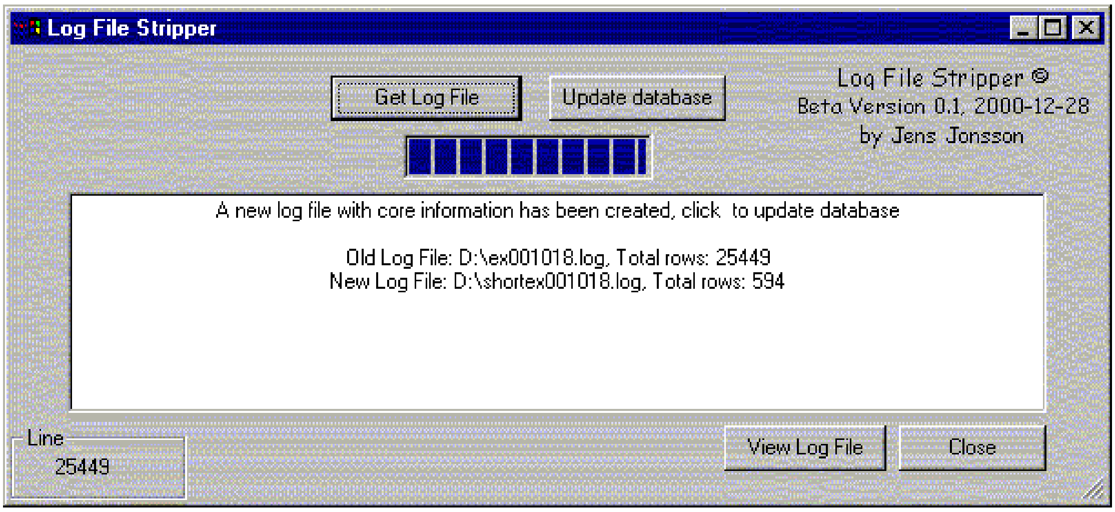
When the server is up and running it is possible to create some business opportunities[1] on the web site that the visitors can participate in. Three such investments opportunities were created and possible for the visitors to sign up on:

- HRC, High Risk Company.
- MRC, Medium Risk Company
- LRC, Low Risk Company

It is quite obvious that these opportunities will attract different kinds of investors since they are of different risk level. The idea is that the Log File Analyzer will be able to show a connection between the visitor's referrer and the visitors willingness and abilities to invest money as described in the test script.

During two days the visitors were exposed to the test server making choices that hopefully reflected something in their personality and their possibilities to participate in offerings. When all visitors had been exposed to the content of the web site it was possible to make an analysis of the log files to learn something more about the visitors.

The analysis process starts with the Log File Stripper. This program is used for removing all unnecessary and redundant information from the log file. By pressing the "Get log file" button (see figure 27) and by browsing the local directories to locate the correct log file a new log file is created that is stripped from all redundant data.



Log File Stripper ©
Beta Version 0.1, 2000-12-28
by Jens Jonsson

```
A new log file with core information has been created, click  to update database

Old Log File: D:\ex001018.log, Total rows: 25449
New Log File: D:\shortex001018.log, Total rows: 594
```

Figure27 Log File Stripper GUI

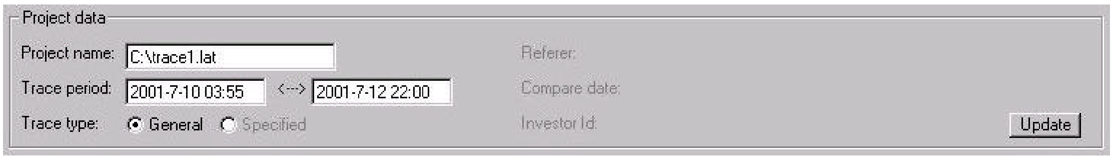
When the log file is stripped the new data is ready to be inserted into the database. This is done by clicking the "Update database" button. The screen in the middle of the program shows information about the size of the new and old log file.

[1] A business opportunity is meant as an Initial Public Offering (IPO).

Now that the database is updated it is possible to analyze the visitor data with the Log File Analyzer.

The first thing to do is to create a new trace file. A trace file contains all the preferences needed to make a trace in the database. There are possibilities to make general or specified traces. A specified trace looks for a certain visitor while a general trace looks for all visitors during a period. By clicking “New trace” under Archive a preference form appears (see figure 18). Filling out this form and pressing ok will create a Log Analyzer Trace-file (.lat). The next time the Log File Analyzer is started one can choose “Open trace file” and the program will use the preferences that was stated in that trace file. When the trace file is opened the Log File Analyzer processes the data and plots the result on the screen.

The results below were retrieved from the Log File Analyzer when doing an analysis of the visitor data during the period 2001-7-10 03:55 - 2001-7-12 22:00.
As shown in figure 2 the trace created is a general trace. This means that all visitors during this period will appear in the result set and not just those with a particular referrer or a particular investor. If one wants to change anything in the preferences, an update can be made to the trace file by clicking the update button. This also causes the Log File Analyzer to recalculate all results and resave the trace file. This makes it very easy to compare results from different dates or trace types.



Project data

Project name: C:\trace1.lat

Trace period: 2001-7-10 03:55 <--> 2001-7-12 22:00

Trace type: ● General ○ Specified

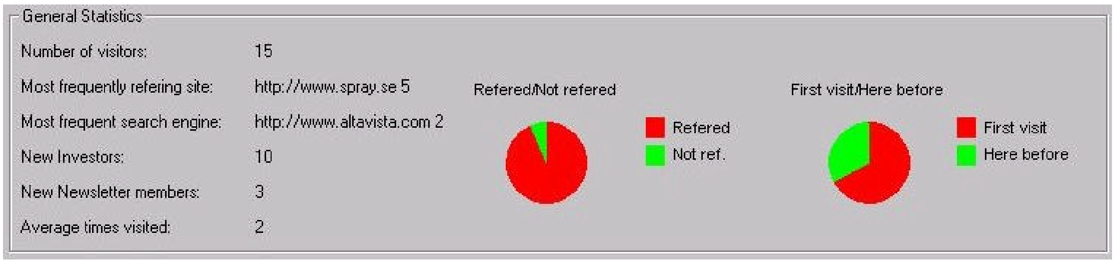
Referer:

Compare date:

Investor Id:

Figure28 Project preferences

The General statistics box (see figure 29) shows some basic results that were retrieved from the analysis. Here we can see that the total number of visitors during this period was as expected 15.
The most frequently referring site was www.spray.se with 5 visitors referred and the most frequent search engine was www.altavista.com with 2 visitors referred. The number of new investors[2] during the period was 10 and the number of new newsletter members was 3. The average number of times visited was 2.



General Statistics

Number of visitors:	15
Most frequently refering site:	http://www.spray.se 5
Most frequent search engine:	http://www.altavista.com 2
New Investors:	10
New Newsletter members:	3
Average times visited:	2

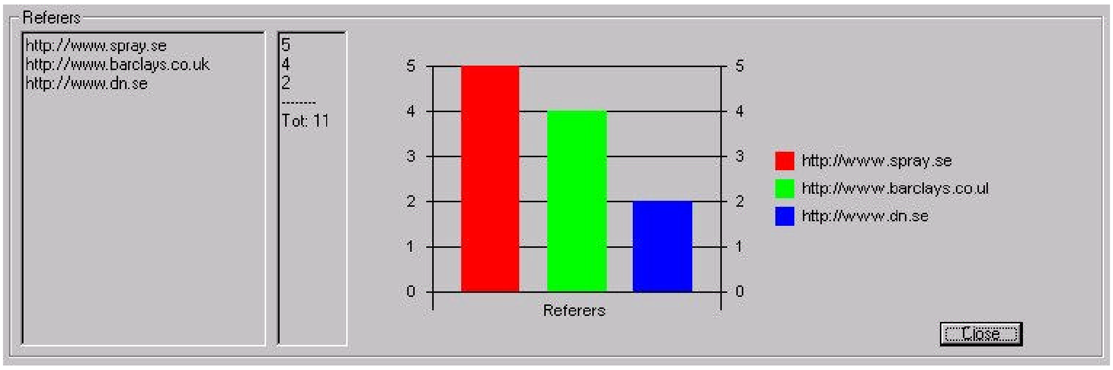
Figure29 General statistics

[2] An investor is a visitor at the EPO site that has become a member and can participate in offerings.

Referrers and Search engines

On the menu bar of the program there is a button that says “Referrers”. When this button is pressed the referrers-box (See fig 3.) appears. This box gives information about from where the visitors where redirected. The referrers-box shows that there were 3 sites in the log file that referred visitors to the EPO site. The sites were:

www.spray.se	5 visitors
www.barclays.co.uk	4 visitors
www.dn.se	2 visitors
	11 visitors


Referers

http://www.spray.se
http://www.barclays.co.uk
http://www.dn.se

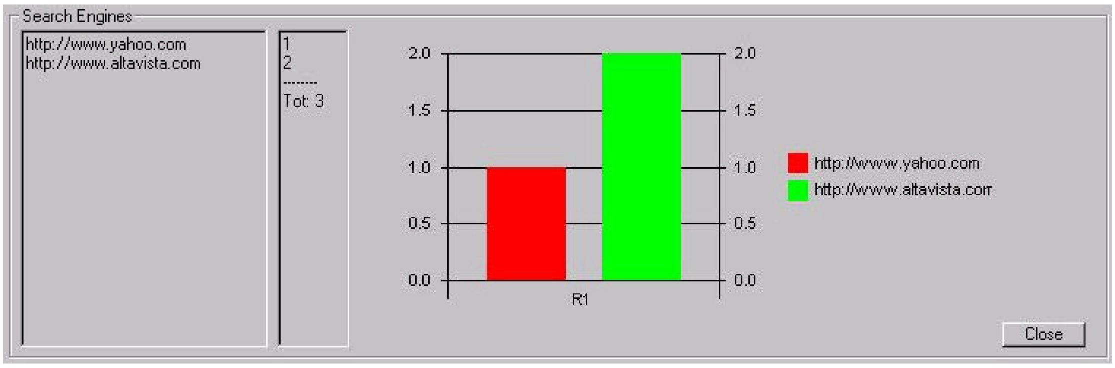
5
4
2

Tot: 11

Figure30 Referring sites

On the menu bar there is also a button called “search engines”. This box gives information about how many visitors that used a search engine to reach the EPO site and what search engine that was used. Figure 31 shows that there were two known search engines used and that three visitors used a search engine to find the EPO site. Those were:

www.altavista.com	2 visitors
www.yahoo.com	1 visitor
	3 visitors


Search Engines

http://www.yahoo.com
http://www.altavista.com

1
2

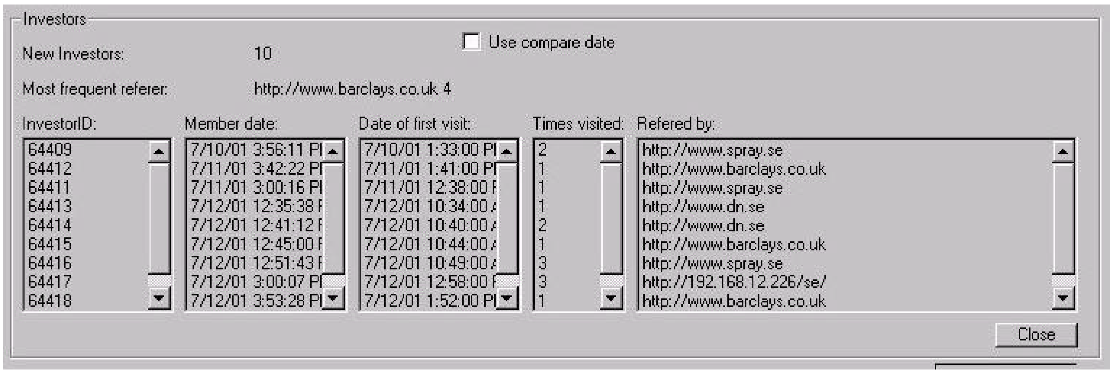
Tot: 3

Figure31 Search engines

In conclusion we can see that a total of 14 visitors were redirected from another site and only one visitor came directly to the EPO site. This information can be used directly to check the correctness of the hit rates provided by the different referrers.

New members

On the menu bar of the Log File Analyzer there is a button called “Investors”. The investors-box gives information about who became a member during the analyzed period. It also tells where the visitors first came from and how many times he/she has visited the site. Figure 32 is a printout of the investors-box. It shows that out of 15 visitors 10 became new members on the EPO site. The most frequently referred site was www.barclays.co.uk, which is a big English broker that has partner agreements with EPO.com.



Investors

New Investors: 10

☐ Use compare date

Most frequent referer: http://www.barclays.co.uk 4

InvestorID:	Member date:	Date of first visit:	Times visited:	Refered by:
64409	7/10/01 3:56:11 PI	7/10/01 1:33:00 PI	2	http://www.spray.se
64412	7/11/01 3:42:22 PI	7/11/01 1:41:00 PI	1	http://www.barclays.co.uk
64411	7/11/01 3:00:16 PI	7/11/01 12:38:00 F	1	http://www.spray.se
64413	7/12/01 12:35:38 F	7/12/01 10:34:00 /	1	http://www.dn.se
64414	7/12/01 12:41:12 F	7/12/01 10:40:00 /	2	http://www.dn.se
64415	7/12/01 12:45:00 F	7/12/01 10:44:00 /	1	http://www.barclays.co.uk
64416	7/12/01 12:51:43 F	7/12/01 10:49:00 /	3	http://www.spray.se
64417	7/12/01 3:00:07 PI	7/12/01 12:58:00 F	3	http://192.168.12.226/se/
64418	7/12/01 3:53:28 PI	7/12/01 1:52:00 PI	1	http://www.barclays.co.uk

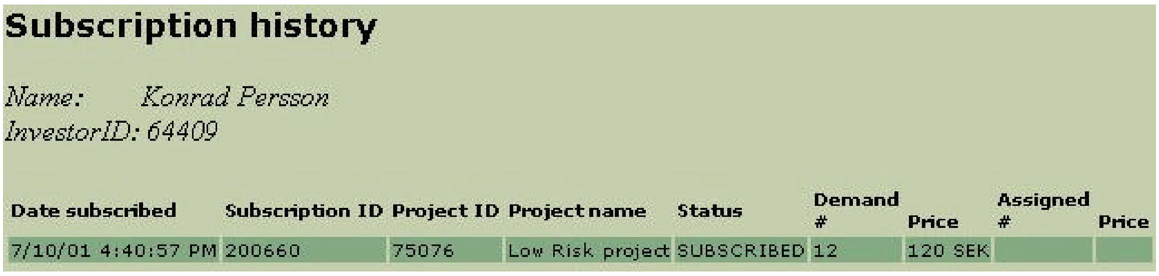
Figure32 New investors

By double-clicking on the investor ID-number one gets access to all personal information about this particular investor. From this we can see all the deals that the visitor has participated in and with what amount (see table below)

Referrer	**Name of Investor**	**Company**	**Amount invested (sek)**
Barclays	Sven Steen	HRC	250.000
Barclays	Glenn Patric	HRC	500.000
Barclays	Nalle Mann	MRC	5000
Barclays	Kenny Lagerfeldth	HRC	150.000
Spray	Konrad Persson	LRP	120
Spray	Liisa Nord	LRP	20.000
Spray	Sture Green	---	---
DN	Thorbjörn Dahl	MRC	5000
DN	Stina Swahn	---	---
---	Johnny Stark	MRC	5000

We can see that there seems to be a very strong connection between the referring sites and the type of investments that is done. Barclays seems to interest investors with more money and they are more willing to participate in high risk offerings whilst Spray attract people less willing to invest and if so in offerings with less risk.

Figures 33- 36 are some printouts of the subscription histories of some of the investors in the test script.

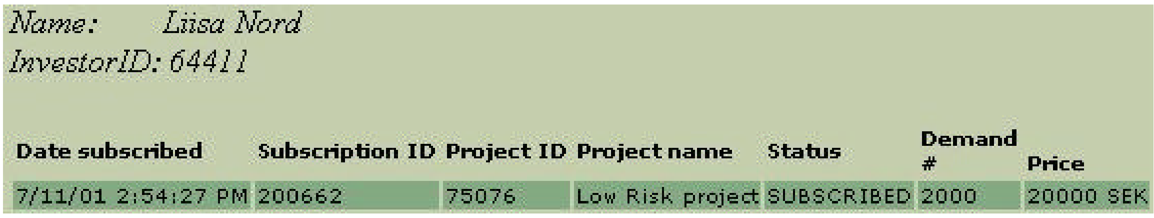


Subscription history

Name: Konrad Persson
InvestorID: 64409

Date subscribed	Subscription ID	Project ID	Project name	Status	Demand #	Price	Assigned #	Price
7/10/01 4:40:57 PM	200660	75076	Low Risk project	SUBSCRIBED	12	120 SEK		

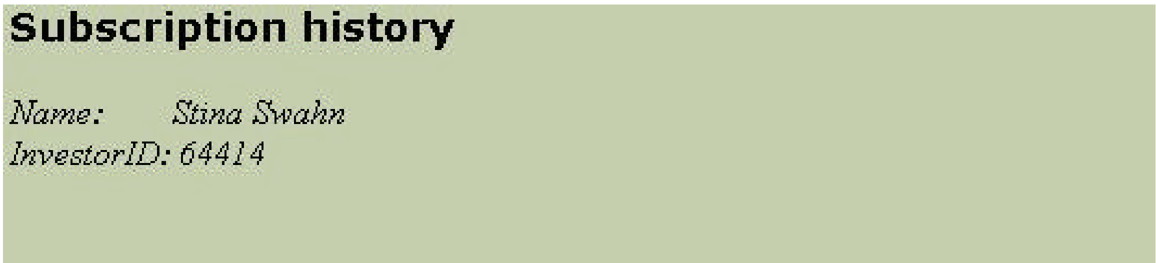
Figure33 Subscription history: Konrad Persson who invested 120 sek in LRC. See the test script for more personal information about this investor.



Name: Liisa Nord
InvestorID: 64411

Date subscribed	Subscription ID	Project ID	Project name	Status	Demand #	Price
7/11/01 2:54:27 PM	200662	75076	Low Risk project	SUBSCRIBED	2000	20000 SEK

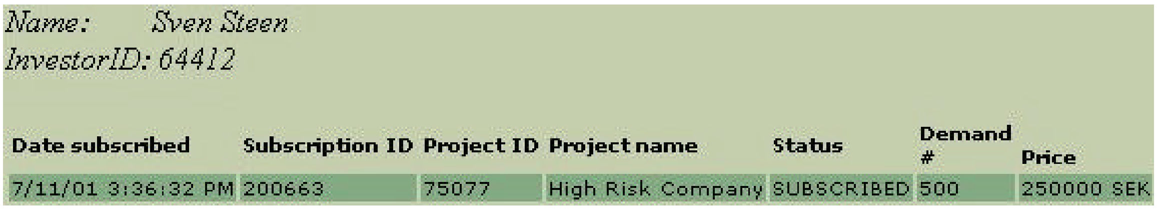
Figure34 Subscription history Liisa Nord who invested 2000 sek in LRC. See the test script for more personal information about this investor.



Subscription history

Name: Stina Swahn
InvestorID: 64414

Figure35 Subscription history Stina Swahn who did not invest anything. See the test script for more personal information about this investor.



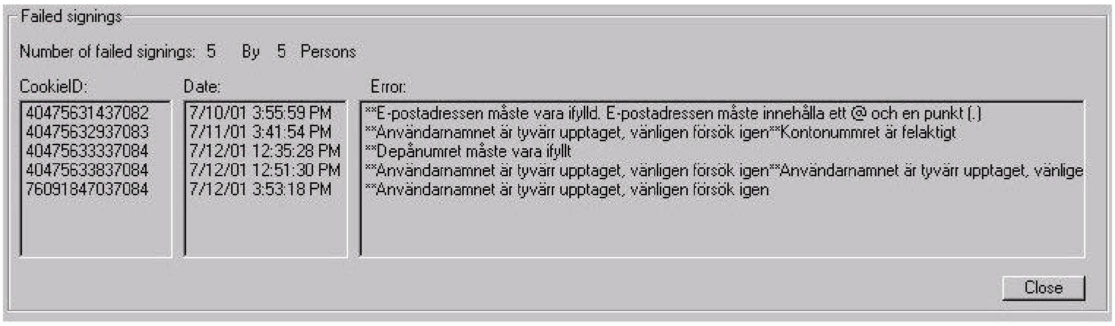
Name: Sven Steen
InvestorID: 64412

Date subscribed	Subscription ID	Project ID	Project name	Status	Demand #	Price
7/11/01 3:36:32 PM	200663	75077	High Risk Company	SUBSCRIBED	500	250000 SEK

Figure36 Subscription history Sven Steen who invested 250.000 sek in HRC. See the test script for more personal information about this investor.

Common Errors

To get an idea of how difficult it is to fill out the registration form for becoming an EPO member the program lists all errors made during the sign-up process (see figure37 below).



Failed signings

Number of failed signings: 5 By 5 Persons

CookieID:	Date:	Error:
40475631437082	7/10/01 3:55:59 PM	**E-postadressen måste vara ifylld. E-postadressen måste innehålla ett @ och en punkt (.)
40475632937083	7/11/01 3:41:54 PM	**Användarnamnet är tyvärr upptaget, vänligen försök igen**Kontonummret är felaktigt
40475633337084	7/12/01 12:35:28 PM	**Depånumret måste vara ifyllt
40475633837084	7/12/01 12:51:30 PM	**Användarnamnet är tyvärr upptaget, vänligen försök igen**Användarnamnet är tyvärr upptaget, vänlige
76091847037084	7/12/01 3:53:18 PM	**Användarnamnet är tyvärr upptaget, vänligen försök igen

Figure37 Failed sign-ups

This information can be useful for the developers of the site because the goal is of cause to make the process of becoming a member as simple as possible. The most common error seems to be involving the username.

4.5.5 Conclusions

Some interesting conclusions were able be drawn from the test results. For example, most visitors were referred and did not come directly to the site. 10 out of 15 visitors became members at the EPO site. Barclays was the site where the most new investors came from. The most common error when signing-up as a new member is submitting an erroneous username.

Apart from this the Log File Analyzer gives the person responsible for advertisement a good idea of how good a referring site actually is. Spray for example gives a lot of page-views but not so many good investors, which is the actual goal of the advertisement. This information can be useful when deciding how much the referring site should be paid for each visitor.

Even though the test persons aren't real, the testing shows that the deeper information about the visitors and the possibility to analyze the results that the Log File Analyzer provides is necessary to make correct decisions about what referring sites to use in different advertisement campaigns. These results also serve as input for the advertisement-effect model (described in part three of the thesis) when deciding how much a visitor from a referring site is worth.

4.6 Conclusions

The area of analyzing web site traffic is huge and there are many ways of doing it.
In this part of the thesis project two tools were developed: First, a tool for stripping log files from unnecessary data and storing the data in a database (Log File Stripper).
Second, a tool that receives log data from the database and generates a detailed statistical analysis that is presented to the user in a user friendly way (Log File Analyzer).
The way of using a database to store the log data is the method used in this thesis.
This method makes the analyzing tool more flexible, it also improves speed and makes it easy to search over long periods of time. It does though demand quite a lot of space on the database server because of its rapid growth.
This type of analyzing is not 100% correct. Due to the anonymity of the Internet and the difficulty to control the visitors it is hard to make a completely reliable analysis.
Storing data on the client by using cookies is the only way to identify the visitors when they return. The problem is that this ID is in the hands of the client and can be removed at any time by the client.
This means that the user of the analysis tool has to be very careful and clear over the uncertainties in the output.
Testing shows that the Log File Analyzer can provide the person responsible for advertisement useful information about the visitors and their behavior on the site. It also gives necessary proof of how good/bad a referring partner-site actually is for EPO.com.

4.7 Future work

There is a lot of work that can be done to improve the usefulness of the Log Analyzer.
Below are some examples of functions that could be added to the application and the work needed to make them come true.

Future work	Work needed
Tracking a visitors path on the web site.	Enlargement of the database so that it can hold more than one line per visitor. The Log File Stripper must be modified so that it stores all lines from the original log file except the ones that the visitor are given without knowing it, i.e pictures and redirect pages. The Log Analyzer must be given functions to display and calculate the path taken by the visitors.
Tracking investments made by an investor.	The Log Analyzer must be modified to make a database call to the system database requesting all deals that the InvestorID has taken part in. Necessary functions for displaying this data must also be implemented.
List common search phrases when coming from a search engine.	Just a matter of parsing the referrer string in the database and extracting the phrases. A display function must be added to the Log Analyzer.
List all failed deal-signings made.	Works the same way as the “failed member signings” box. More or less a copy of that

	code is needed.

There are a lot more small features that can be added to the program in the future and that is actually the reason that no more than the basic functions has been included in the thesis.

5 Interaction of optimizer and traffic analyzer

5.1 Introduction

Part three and part four in the thesis describe the web site traffic optimization and the web site analyzing tool. As described earlier in the text the traffic optimization tool needs input data in order two generate a prediction about the expected outcome of an advertisement campaign. If the input data is accurate then so will the solution, better-input data is acquired by having a larger statistical basis. In this part of the thesis it will be explained how the optimization model and the web site analyzing tool interact, some limitations and also some further development.

5.2 Typical usage

Before investing in advertisement on a specific site the person responsible for advertisement will create a profile of the type of visitor that this campaign will attempt to draw to the site. When this is done these parameters will be inserted into the optimization model and a suggested solution will be presented. This solution could involve several different advertisement sites and time periods.
The website analyzer tool analyses the incoming traffic and gives the marketing director a good idea of how well the advertisement came out. This data also serves as input to the optimization model to modify it so that a better solution can be given the next time the optimization tool is used. It is common for advertisement sites to charge their clients for the number of visitors that came from the advertisement site. The analyzing tool is able to check so that these numbers are correct.

5.3 Limitations

The site selling the advertisement initially provides the input parameters such as visitor arrival intensity. When some campaigns have been run enough statistical data will have been gathered so that more accurate estimations of the input data can be made. Running campaigns is expensive and thereby statistical data is expensive. The intention is to have enough data from previous campaigns so that the information from the site selling the advertisement will not be needed. This is because the information provided is not specific enough and also it comes from and source which is biased. Before a large statistical basis has been generated the solution will be unstable and unpredictable because the estimation of the parameters will be crud.

5.4 Expected results

After the campaign optimizer has been used guidelines should have been generated indicating whether or not is it efficient to spend the intended advertisement money or if another campaign should be sought. It is important to remember that the results from the campaign optimizer are intended to aid the marketing director and not serve as a definite answer. The uncertainties are too great to blindly trust the solution.

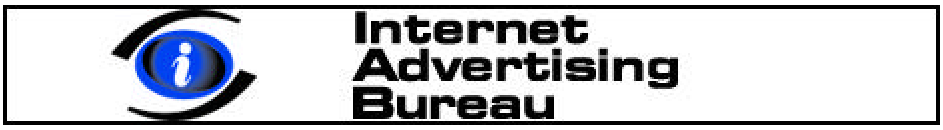
5.5 Further development

At the moment the web site traffic optimization and the web site analyzing -tool are not integrated into one functional program. The next step is to integrate these two separate modules so that the input estimation is done automatically. Creating a

database where all the statistical data from the different campaigns are stored can do this. When the user types in the campaign budget and customer profile the application will generate all the possible alternatives with respect to budget and user profile and then generate the optimal campaign.
One step even further would be to build a distributed database where companies share campaign statistics with each other. In this way costs are reduced and data is more frequently updated thereby making the suggested campaign more accurate.

Appendix A

Standard Banner sizes



view light IMU Utilizes JAVA
468 x 60 IMU
(Full Banner)

Internet
Advertising
Bureau

234 x 60 IMU
(Half Banner)

Internet
Advertising
Bureau

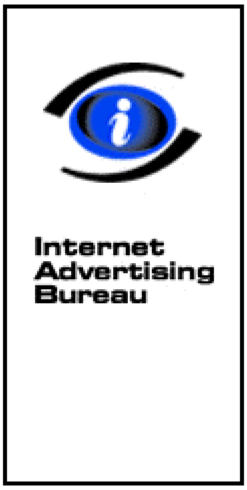
88 x 31 IMU
(Micro Bar)

Internet
Advertising
Bureau

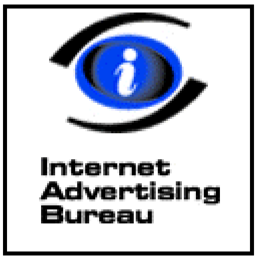
120 x 90 IMU
(Button 1)

Internet
Advertising
Bureau

120 x 60 IMU
(Button 2)



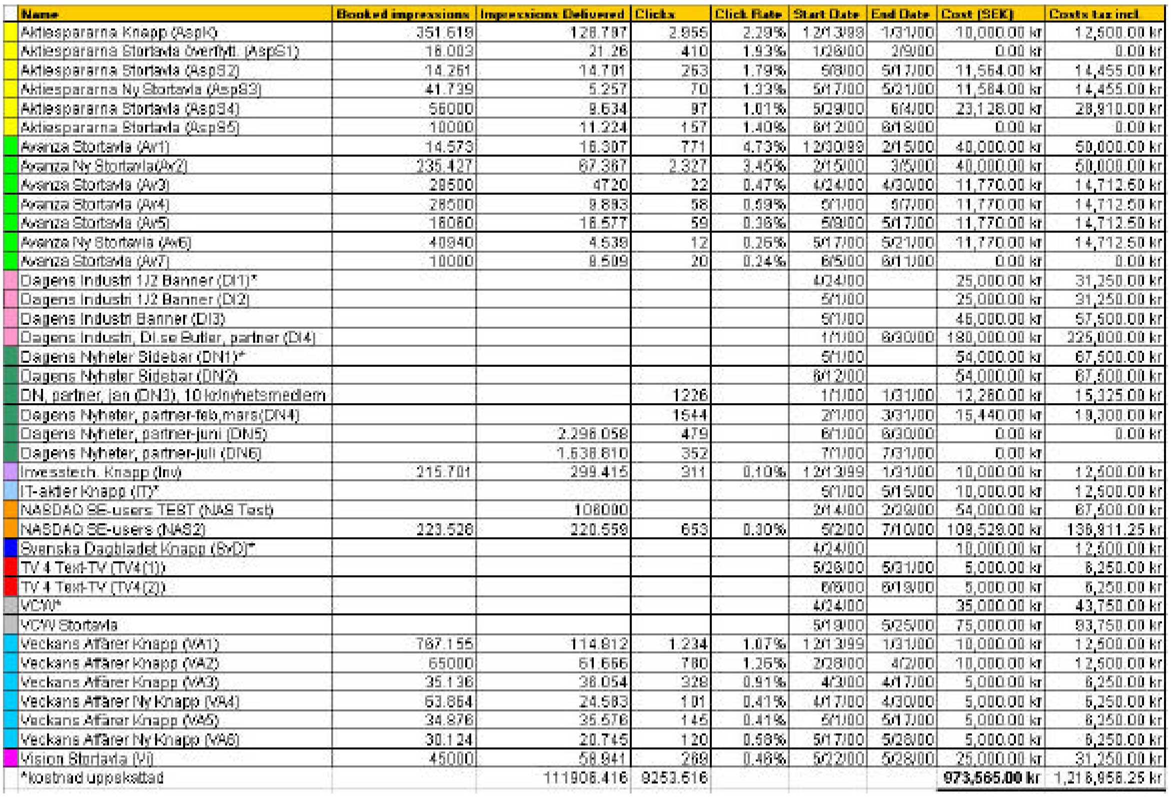
120 x 240 IMU
(Vertical Banner)



125 x 125 IMU
(Square Button)

Appendix B

Banner campaign results



Name	Booked impressions	Impressions Delivered	Clicks	Click Rate	Start Date	End Date	Cost (SEK)	Costs tax incl.
Aktiespararna Knapp (AspK)	351.619	128.787	2.955	2.29%	12/13/99	1/31/00	10,000.00 kr	12,500.00 kr
Aktiespararna Stortavla överflytt. (AspS1)	16.003	21.26	410	1.93%	1/26/00	2/9/00	0.00 kr	0.00 kr
Aktiespararna Stortavla (AspS2)	14.261	14.701	263	1.79%	5/8/00	5/17/00	11,564.00 kr	14,455.00 kr
Aktiespararna Ny Stortavla (AspS3)	41.739	5.257	70	1.33%	5/17/00	5/21/00	11,564.00 kr	14,455.00 kr
Aktiespararna Stortavla (AspS4)	56000	9.634	97	1.01%	5/29/00	6/4/00	23,128.00 kr	28,910.00 kr
Aktiespararna Stortavla (AspS5)	10000	11.224	157	1.40%	6/12/00	6/18/00	0.00 kr	0.00 kr
Avanza Stortavla (Av1)	14.573	16.307	771	4.73%	12/30/99	2/15/00	40,000.00 kr	50,000.00 kr
Avanza Ny Stortavla(Av2)	235.427	67.367	2.327	3.45%	2/15/00	3/5/00	40,000.00 kr	50,000.00 kr
Avanza Stortavla (Av3)	28500	4720	22	0.47%	4/24/00	4/30/00	11,770.00 kr	14,712.50 kr
Avanza Stortavla (Av4)	28500	9.893	58	0.59%	5/1/00	5/7/00	11,770.00 kr	14,712.50 kr
Avanza Stortavla (Av5)	16060	16.577	59	0.36%	5/8/00	5/17/00	11,770.00 kr	14,712.50 kr
Avanza Ny Stortavla (Av6)	40940	4.539	12	0.26%	5/17/00	5/21/00	11,770.00 kr	14,712.50 kr
Avanza Stortavla (Av7)	10000	8.509	20	0.24%	6/5/00	6/11/00	0.00 kr	0.00 kr
Dagens Industri 1/2 Banner (DI1)*					4/24/00		25,000.00 kr	31,250.00 kr
Dagens Industri 1/2 Banner (DI2)					5/1/00		25,000.00 kr	31,250.00 kr
Dagens Industri Banner (DI3)					5/1/00		46,000.00 kr	57,500.00 kr
Dagens Industri, DI.se Butler, partner (DI4)					1/1/00	6/30/00	180,000.00 kr	225,000.00 kr
Dagens Nyheter Sidebar (DN1)*					5/1/00		54,000.00 kr	67,500.00 kr
Dagens Nyheter Sidebar (DN2)					6/12/00		54,000.00 kr	67,500.00 kr
DN, partner, jan (DN3), 10 kr/nyhetsmedlem			1226		1/1/00	1/31/00	12,260.00 kr	15,325.00 kr
Dagens Nyheter, partner-feb,mars(DN4)			1544		2/1/00	3/31/00	15,440.00 kr	19,300.00 kr
Dagens Nyheter, partner-juni (DN5)		2.298.058	479		6/1/00	6/30/00	0.00 kr	0.00 kr
Dagens Nyheter, partner-juli (DN6)		1.638.610	352		7/1/00	7/31/00	0.00 kr	
Invesstech. Knapp (Inv)	215.701	299.415	311	0.10%	12/13/99	1/31/00	10,000.00 kr	12,500.00 kr
IT-aktier Knapp (IT)*					5/1/00	5/15/00	10,000.00 kr	12,500.00 kr
NASDAQ SE-users TEST (NAS Test)		106000			2/14/00	2/29/00	54,000.00 kr	67,500.00 kr
NASDAQ SE-users (NAS2)	223.526	220.559	653	0.30%	5/2/00	7/10/00	109,529.00 kr	136,911.25 kr
Svenska Dagbladet Knapp (SvD)*					4/24/00		10,000.00 kr	12,500.00 kr
TV 4 Text-TV (TV4(1))					5/26/00	5/31/00	5,000.00 kr	6,250.00 kr
TV 4 Text-TV (TV4(2))					6/6/00	6/19/00	5,000.00 kr	6,250.00 kr
VCW*					4/24/00		35,000.00 kr	43,750.00 kr
VCW Stortavla					5/19/00	5/25/00	75,000.00 kr	93,750.00 kr
Veckans Affärer Knapp (VA1)	767.155	114.812	1.234	1.07%	12/13/99	1/31/00	10,000.00 kr	12,500.00 kr
Veckans Affärer Knapp (VA2)	65000	61.666	780	1.26%	2/28/00	4/2/00	10,000.00 kr	12,500.00 kr
Veckans Affärer Knapp (VA3)	35.136	36.054	328	0.91%	4/3/00	4/17/00	5,000.00 kr	6,250.00 kr
Veckans Affärer Ny Knapp (VA4)	63.864	24.583	101	0.41%	4/17/00	4/30/00	5,000.00 kr	6,250.00 kr
Veckans Affärer Knapp (VA5)	34.876	35.576	145	0.41%	5/1/00	5/17/00	5,000.00 kr	6,250.00 kr
Veckans Affärer Ny Knapp (VA6)	30.124	20.745	120	0.58%	5/17/00	5/28/00	5,000.00 kr	6,250.00 kr
Vision Stortavla (Vi)	45000	58.941	269	0.46%	5/22/00	5/28/00	25,000.00 kr	31,250.00 kr
*kostnad uppskattad		111906.416	9253.616				**973,565.00 kr**	1,216,956.25 kr

Appendix C

EPO.com Customer Survey

INVESTMENTS

Q1. How did you first learn of EPO.com?

Banner ad on the Web
Internet financial information provider
Magazine advertisement
Search engine
Print news story or article
Recommended by a friend/word –of-mouth
Other

Q2. Which market sectors are you most interested in tracking or investing in?

Technology (general)
    Internet
    Dot com companies
    Hardware
    Software
    Networking
    Consulting
Manufacturing
Business Services
Distribution/Transport
Utilities
Retail/Wholesale
Health/Education
Other (specify) _______________

Q3. How frequently do you trade in shares or investments?

Daily, weekly, monthly, quarterly, annually

Q4. In how many IPOs have you invested in over the last 12 months?

None
1
2
3
4
5
More (Specify)____

Q5. Please choose the statement below which best describes your current investment interest in IPOs:

Actively in the market to purchase
In the market to purchase but undecided
Have postponed purchasing in the near future

Q6. Into which of the following bands does your annual spend on all investments fall?

Less than £5,000
£5,000 - £10,000
£11,000 - £20,000
£21,000 - £50,000
£51,000 - £99,000
£100K - £150K
£151K - £199K
£200K+

Q7. How much are you likely to spend on IPO's (Initial Public Offering) investments in the next 12 months?

Less than £5,000
£5,000 - £10,000
£11,000 - £20,000
£21,000 - £50,000
£51,000 - £99,000
£100K - £150K
£151K - £199K
£200K+

SWE ONLY

Q Overall, how do you compare the EPO.com process to the conventional IPO buying process?

The very best
Far better
Better
Somewhat better
No Better
Worse

DEMOGRAPHICS

Qi. On average, how many hours a week do you spend using the Internet at work and at home on financial related matters?

Work _______No. of hours per week

Home_______No. of hours per week

Qii. Into which of the following bands does your annual household income fall?

Less than £20,000
£21,000 - £30,000
£31,000 - £50,000
£51,000 - £75,000
£76,000 - £99,000
£100K - £150K
£151K - £199K
£200K+

Qiii

What age group
Male/Female
Marital Status
Employment (CEO/MD/Owner, Director, Manager, Dept Head, Executive, Other)

Qiv. Do you have an account with any of the following financial service providers?

Online banking
Online Broker
etc

SWE ONLY

Q Overall, how would you rate the EPO.com service?

Rate 1 - 5

And how useful have the following been:-

Rate 1 - 5
Newsletter information
Information about issuing firms
IPO transactions
Post IPO service
Customer Care/query service

Q. EPO.com believes that 'quality' is defined by our customers. Please tell us what it was about our service that MOST met your expectations, and also what LEAST met your expectations. Also, tell us what we could do for you that we are not doing now. (Open answers)

Q. Have you acted on information received through EPO.com?

Bought shares
Invested in IPO
etc

NEWSLETTER

Q. Could you state how frequently you read each of the following sections of the weekly EPO-Bulletin?

News
Always, Most of the time, Sometimes, Rarely

UK/SWE Hot IPOs
Always, Most of the time, Sometimes, Rarely

Spotlight Start-Up
Always, Most of the time, Sometimes, Rarely

Recently Floated
Always, Most of the time, Sometimes, Rarely

Q. What do you think of the length of the EPO-Bulletin newsletter?

OK, Too short, Too long, Don't know

Q. How often would you like the EPO-Bulletin newsletter to be published?

Daily, 2-3 times/week, As now (weekly), fortnightly, monthly, Don't know

Q. How interested would you be in the following services?
Please rate on a scale of 1 to 5, where 1 means not interested and 5 means very interested) [NEED TO SELECT 6 MAX!]

- Instant update on breaking news
- Overview of other's recommendations in offerings
- Research on interesting unlisted companies
- Information on European offerings
- Chat with other investors
- Follow-up information following a company IPO
- General information about IPOs
- General information about unlisted companies
- Company information
- Investor information on companies
- Video presentations
- Bookstore (with finance relate titles)

- IPO seminars

Q. Do you receive/subscribe to any other financial information services?

If so which ones (list)?

CHAT ROOM

Q. Do you know what a discussion forum/ chatroom/ bulletin board is?

Q. Can you name a few discussion forum that you know of?

Q. Regarding financial discussion forum, have you ever seen or used the one on the

iii website?
zyx website?
kjd website?
jkf website?

Q. Do you use any of these forum regularly?

Which one?
How often?
What do you think about this site?

Q. What advantages do you see with discussion forum?

Q. What are the disadvantages?

Q. Do you think you would have visited an EPO.com discussion forum regularly?

```
'Karl Rylander
'Master thesis program
'Function describing the process when the interface data is retieved and processed

Private Sub Start_Click()

'Number of advertising sites

NumberOfPreDefSites = 7

ReDim TotAmountOfAds(1 To NumberOfPreDefSites)

TotAmountOfAds(1) = CLng(nb1.Text)
TotAmountOfAds(2) = CLng(nb2.Text)
TotAmountOfAds(3) = CLng(nb3.Text)
TotAmountOfAds(4) = CLng(nb4.Text)
TotAmountOfAds(5) = CLng(nb5.Text)
TotAmountOfAds(6) = CLng(nb6.Text)
TotAmountOfAds(7) = CLng(nb7.Text)

NumberTotalSites = CLng(NumberOfSites.Text)

ReDim AmountOfAds(1 To NumberTotalSites)

Banner_Max = 0
For j = 1 To NumberTotalSites
   If Banner_Max < TotAmountOfAds(j) Then
      Banner_Max = TotAmountOfAds(j)
   End If
Next



'The different time periods of the ads (i,j) where i is the sites and j the different ads


'Create arrays for the different calculations
'Since only three different bannerads are allowed on each site it is possible to hard
code the lentgh of j = 3


ReDim Transpv(1 To NumberOfSites, 1 To Banner_Max) As Double
ReDim Transnl(1 To NumberOfSites, 1 To Banner_Max) As Double
ReDim Transinv(1 To NumberOfSites, 1 To Banner_Max) As Double
ReDim Prob(1 To NumberOfSites, 1 To Banner_Max) As Double
ReDim Pageviews(1 To NumberOfSites, 1 To Banner_Max) As Double
ReDim TimeArray(1 To NumberOfSites, 1 To Banner_Max) As Long

ReDim ValuePageView(1 To NumberOfSites) As Double
ReDim ValueNewsLetter(1 To NumberOfSites) As Double
ReDim ValueInvestor(1 To NumberOfSites) As Double
ReDim SiteNameArray(1 To NumberOfSites) As String
ReDim BannerKind(1 To NumberTotalSites, 1 To Banner_Max)


'Fill the arrays with values depending on how many sites that are used

If NumberTotalSites = 2 Or NumberTotalSites = 3 Or NumberTotalSites = 4 Or
NumberTotalSites = 5 Or NumberTotalSites = 6 _
Or NumberTotalSites = 7 Or NumberTotalSites = 8 Or NumberTotalSites = 9 Or
NumberTotalSites = 10 Then

    'Name of the advertising sites
    SiteNameArray(1) = Site1.Text
    SiteNameArray(2) = Site2.Text

    'Fill the vectors depending on how large they are

    If BKind11 <> "" And BKind12 <> "" And BKind13 <> "" Then

        BannerKind(1, 1) = BKind11.Text
        BannerKind(1, 2) = BKind12.Text
        BannerKind(1, 3) = BKind13.Text

        TimeArray(1, 1) = CLng(Time11.Text)
        TimeArray(1, 2) = CLng(Time12.Text)
        TimeArray(1, 3) = CLng(Time13.Text)
        Transpv(1, 1) = CDbl(Transpv11.Text)
        Transpv(1, 2) = CDbl(Transpv12.Text)
        Transpv(1, 3) = CDbl(Transpv13.Text)
        Transnl(1, 1) = CDbl(Transnl11.Text)
        Transnl(1, 2) = CDbl(Transnl12.Text)
        Transnl(1, 3) = CDbl(Transnl13.Text)
        Transinv(1, 1) = CDbl(Transinv11.Text)
        Transinv(1, 2) = CDbl(Transinv12.Text)
        Transinv(1, 3) = CDbl(Transinv13.Text)
        Prob(1, 1) = CDbl(Prob11.Text)
        Prob(1, 2) = CDbl(Prob12.Text)
        Prob(1, 3) = CDbl(Prob13.Text)
        Pageviews(1, 1) = CDbl(Pagev11.Text)
        Pageviews(1, 2) = CDbl(Pagev12.Text)
        Pageviews(1, 3) = CDbl(Pagev13.Text)
    End If

    If BKind11 <> "" And BKind12 <> "" And BKind13 = "" Then
        BannerKind(1, 1) = BKind11.Text
        BannerKind(1, 2) = BKind12.Text

        TimeArray(1, 1) = CLng(Time11.Text)
        TimeArray(1, 2) = CLng(Time12.Text)
        Transpv(1, 1) = CDbl(Transpv11.Text)
        Transpv(1, 2) = CDbl(Transpv12.Text)
        Transnl(1, 1) = CDbl(Transnl11.Text)
        Transnl(1, 2) = CDbl(Transnl12.Text)
        Transinv(1, 1) = CDbl(Transinv11.Text)
        Transinv(1, 2) = CDbl(Transinv12.Text)
        Prob(1, 1) = CDbl(Prob11.Text)
        Prob(1, 2) = CDbl(Prob12.Text)
        Pageviews(1, 1) = CDbl(Pagev11.Text)
        Pageviews(1, 2) = CDbl(Pagev12.Text)
    End If

    If BKind11 <> "" And BKind12 = "" Then
        BannerKind(1, 1) = BKind11.Text
        TimeArray(1, 1) = CLng(Time11.Text)
        Transpv(1, 1) = CDbl(Transpv11.Text)
        Transnl(1, 1) = CDbl(Transnl11.Text)
        Transinv(1, 1) = CDbl(Transinv11.Text)
        Prob(1, 1) = CDbl(Prob11.Text)
        Pageviews(1, 1) = CDbl(Pagev11.Text)
    End If

    If BKind21 <> "" And BKind22 <> "" And BKind23 <> "" Then
        BannerKind(2, 1) = BKind21.Text
        BannerKind(2, 2) = BKind22.Text
        BannerKind(2, 3) = BKind23.Text

        TimeArray(2, 1) = CLng(Time21.Text)
        TimeArray(2, 2) = CLng(Time22.Text)
        TimeArray(3, 3) = CLng(Time23.Text)
        Transpv(2, 1) = CDbl(Transpv21.Text)
        Transpv(2, 2) = CDbl(Transpv22.Text)
        Transpv(2, 3) = CDbl(Transpv23.Text)
        Transnl(2, 1) = CDbl(Transnl21.Text)
        Transnl(2, 2) = CDbl(Transnl22.Text)
        Transnl(2, 3) = CDbl(Transnl23.Text)
        Transinv(2, 1) = CDbl(Transinv21.Text)
        Transinv(2, 2) = CDbl(Transinv22.Text)
        Transinv(2, 3) = CDbl(Transinv23.Text)
        Prob(2, 1) = CDbl(Prob21.Text)
        Prob(2, 2) = CDbl(Prob22.Text)
        Prob(2, 3) = CDbl(Prob23.Text)
        Pageviews(2, 1) = CDbl(Pagev21.Text)
        Pageviews(2, 2) = CDbl(Pagev22.Text)
        Pageviews(2, 3) = CDbl(Pagev23.Text)
    End If
    If BKind21 <> "" And BKind22 <> "" And BKind23 = "" Then
        BannerKind(2, 1) = BKind21.Text
        BannerKind(2, 2) = BKind22.Text

        TimeArray(2, 1) = CLng(Time21.Text)
        TimeArray(2, 2) = CLng(Time22.Text)
        Transpv(2, 1) = CDbl(Transpv21.Text)
        Transpv(2, 2) = CDbl(Transpv22.Text)
        Transnl(2, 1) = CDbl(Transnl21.Text)
        Transnl(2, 2) = CDbl(Transnl22.Text)
        Transinv(2, 1) = CDbl(Transinv21.Text)
        Transinv(2, 2) = CDbl(Transinv22.Text)
        Prob(2, 1) = CDbl(Prob21.Text)
        Prob(2, 2) = CDbl(Prob22.Text)
        Pageviews(2, 1) = CDbl(Pagev21.Text)
        Pageviews(2, 2) = CDbl(Pagev22.Text)
    End If

    If BKind21 <> "" And BKind22 = "" Then
        BannerKind(2, 1) = BKind21.Text
        TimeArray(2, 1) = CLng(Time21.Text)
        Transpv(2, 1) = CDbl(Transpv21.Text)
        Transnl(2, 1) = CDbl(Transnl21.Text)
        Transinv(2, 1) = CDbl(Transinv21.Text)
        Prob(2, 1) = CDbl(Prob21.Text)
        Pageviews(2, 1) = CDbl(Pagev21.Text)
    End If

End If

If NumberTotalSites = 3 Or NumberTotalSites = 4 Or NumberTotalSites = 5 Or
NumberTotalSites = 6 _
Or NumberTotalSites = 7 Or NumberTotalSites = 8 Or NumberTotalSites = 9 Or
NumberTotalSites = 10 Then

    SiteNameArray(3) = Site3.Text

    'Fill the vectors depending on how large they are

    If BKind31 <> "" And BKind32 <> "" And BKind33 <> "" Then
        BannerKind(3, 1) = BKind31.Text
        BannerKind(3, 2) = BKind32.Text
        BannerKind(3, 3) = BKind33.Text

        TimeArray(3, 1) = CLng(Time31.Text)
        TimeArray(3, 2) = CLng(Time32.Text)
        TimeArray(3, 3) = CLng(Time33.Text)
        Transpv(3, 1) = CDbl(Transpv31.Text)
        Transpv(3, 2) = CDbl(Transpv32.Text)
        Transpv(3, 3) = CDbl(Transpv33.Text)
        Transnl(3, 1) = CDbl(Transnl31.Text)
        Transnl(3, 2) = CDbl(Transnl32.Text)
        Transnl(3, 3) = CDbl(Transnl33.Text)
        Transinv(3, 1) = CDbl(Transinv31.Text)
        Transinv(3, 2) = CDbl(Transinv32.Text)
        Transinv(3, 3) = CDbl(Transinv33.Text)
        Prob(3, 1) = CDbl(Prob31.Text)
        Prob(3, 2) = CDbl(Prob32.Text)
        Prob(3, 3) = CDbl(Prob33.Text)
        Pageviews(3, 1) = CDbl(Pagev31.Text)
        Pageviews(3, 2) = CDbl(Pagev32.Text)
        Pageviews(3, 3) = CDbl(Pagev33.Text)
    End If

    If BKind31 <> "" And BKind32 <> "" And BKind33 = "" Then
        BannerKind(3, 1) = BKind31.Text
        BannerKind(3, 2) = BKind32.Text
        TimeArray(3, 1) = CLng(Time31.Text)
        TimeArray(3, 2) = CLng(Time32.Text)
        Transpv(3, 1) = CDbl(Transpv31.Text)
        Transpv(3, 2) = CDbl(Transpv32.Text)
        Transnl(3, 1) = CDbl(Transnl31.Text)
        Transnl(3, 2) = CDbl(Transnl32.Text)
        Transinv(3, 1) = CDbl(Transinv31.Text)
        Transinv(3, 2) = CDbl(Transinv32.Text)
        Prob(3, 1) = CDbl(Prob31.Text)
        Prob(3, 2) = CDbl(Prob32.Text)
        Pageviews(3, 1) = CDbl(Pagev31.Text)
        Pageviews(3, 2) = CDbl(Pagev32.Text)
    End If

    If BKind31 <> "" And BKind32 = "" Then
        BannerKind(3, 1) = BKind31.Text
        TimeArray(3, 1) = CLng(Time31.Text)
        Transpv(3, 1) = CDbl(Transpv31.Text)
        Transnl(3, 1) = CDbl(Transnl31.Text)
        Transinv(3, 1) = CDbl(Transinv31.Text)
        Prob(3, 1) = CDbl(Prob31.Text)
        Pageviews(3, 1) = CDbl(Pagev31.Text)
    End If
End If


If NumberTotalSites = 4 Or NumberTotalSites = 5 Or NumberTotalSites = 6 Or
NumberTotalSites = 7 Or NumberTotalSites = 8 Or NumberTotalSites = 9 Or
NumberTotalSites = 10 Then

    SiteNameArray(4) = Site4.Text
```

```
'Fill the vectors depending on how large they are

If BKind41 <> "" And BKind42 <> "" And BKind43 <> "" Then
    BannerKind(4, 1) = BKind41.Text
    BannerKind(4, 2) = BKind42.Text
    BannerKind(4, 3) = BKind43.Text

    TimeArray(4, 1) = CLng(Time41.Text)
    TimeArray(4, 2) = CLng(Time42.Text)
    TimeArray(4, 3) = CLng(Time43.Text)
    Transpv(4, 1) = CDbl(Transpv41.Text)
    Transpv(4, 2) = CDbl(Transpv42.Text)
    Transpv(4, 3) = CDbl(Transpv43.Text)
    Transnl(4, 1) = CDbl(Transnl41.Text)
    Transnl(4, 2) = CDbl(Transnl42.Text)
    Transnl(4, 3) = CDbl(Transnl43.Text)
    Transinv(4, 1) = CDbl(Transinv41.Text)
    Transinv(4, 2) = CDbl(Transinv42.Text)
    Transinv(4, 3) = CDbl(Transinv43.Text)
    Prob(4, 1) = CDbl(Prob41.Text)
    Prob(4, 2) = CDbl(Prob42.Text)
    Prob(4, 3) = CDbl(Prob43.Text)
    Pageviews(4, 1) = CDbl(Pagev41.Text)
    Pageviews(4, 2) = CDbl(Pagev42.Text)
    Pageviews(4, 3) = CDbl(Pagev43.Text)
End If

If BKind41 <> "" And BKind42 <> "" And BKind43 = "" Then
    BannerKind(4, 1) = BKind41.Text
    BannerKind(4, 2) = BKind42.Text

    TimeArray(4, 1) = CLng(Time41.Text)
    TimeArray(4, 2) = CLng(Time42.Text)
    Transpv(4, 1) = CDbl(Transpv41.Text)
    Transpv(4, 2) = CDbl(Transpv42.Text)
    Transnl(4, 1) = CDbl(Transnl41.Text)
    Transnl(4, 2) = CDbl(Transnl42.Text)
    Transinv(4, 1) = CDbl(Transinv41.Text)
    Transinv(4, 2) = CDbl(Transinv42.Text)
    Prob(4, 1) = CDbl(Prob41.Text)
    Prob(4, 2) = CDbl(Prob42.Text)
    Pageviews(4, 1) = CDbl(Pagev41.Text)
    Pageviews(4, 2) = CDbl(Pagev42.Text)
End If

If BKind41 <> "" And BKind42 = "" Then
    BannerKind(4, 1) = BKind41.Text
    TimeArray(4, 1) = CLng(Time41.Text)
    Transpv(4, 1) = CDbl(Transpv41.Text)
    Transnl(4, 1) = CDbl(Transnl41.Text)
```

```
        Transinv(4, 1) = CDbl(Transinv41.Text)
        Prob(4, 1) = CDbl(Prob41.Text)
        Pageviews(4, 1) = CDbl(Pagev41.Text)
    End If
End If

If NumberTotalSites = 5 Or NumberTotalSites = 6 Or NumberTotalSites = 7 Or
NumberTotalSites = 8 Or NumberTotalSites = 9 Or NumberTotalSites = 10 Then
    TimeArray(5) = Time_5
    SiteNameArray(5) = Site5.Text
    Transpv(5) = CDbl(Transpv5.Text)
    Transnl(5) = CDbl(Transnl5.Text)
    Transinv(5) = CDbl(Transinv5.Text)
    Prob(5) = CDbl(Prob5.Text)
    Pageviews(5) = CDbl(Pagev5.Text)
End If

If NumberTotalSites = 6 Or NumberTotalSites = 7 Or NumberTotalSites = 8 Or
NumberTotalSites = 9 Or NumberTotalSites = 10 Then
    TimeArray(6) = Time_6
    SiteNameArray(6) = Site6.Text
    Transpv(6) = CDbl(Transpv6.Text)
    Transnl(6) = CDbl(Transnl6.Text)
    Transinv(6) = CDbl(Transinv6.Text)
    Prob(6) = CDbl(Prob6.Text)
    Pageviews(6) = CDbl(Pagev6.Text)
End If

If NumberTotalSites = 7 Or NumberTotalSites = 8 Or NumberTotalSites = 9 Or
NumberTotalSites = 10 Then
    TimeArray(7) = Time_7
    SiteNameArray(7) = Site7.Text
    Transpv(7) = CDbl(Transpv7.Text)
    Transnl(7) = CDbl(Transnl7.Text)
    Transinv(7) = CDbl(Transinv7.Text)
    Prob(7) = CDbl(Prob7.Text)
    Pageviews(7) = CDbl(Pagev7.Text)
End If

'Find the largest time period

Time_Max = 0
For i = 1 To NumberTotalSites
    For j = 1 To Banner_Max
    If Time_Max < TimeArray(i, j) Then
        Time_Max = TimeArray(i, j)
    End If
    Next
Next
```

```
'Define the matrix that will be used for the coefficients


'Cannot work with arrays apperentl
'ReDim C_ijt(1 To NumberTotalSites, 1 To Banner_Max, 1 To Time_Max) As
Double

'Create the result array so that the coefficients are displayed in a (i,t) way


ReDim ResultArray(1 To NumberOfSites, 1 To Banner_Max, 1 To Time_Max) As
Double

'Fill the budget and the profit

Budget = CDbl(Budget.Text)
Profit = CDbl(Profit.Text)

'Fill the values of the different investor statuses

Valuepv = CDbl(Valuepv.Text)
Valuenl = CDbl(Valuenl.Text)
Valueinv = CDbl(Valueinv.Text)

For i = 1 To NumberOfSites
   ValuePageView(i) = Valuepv
   ValueNewsLetter(i) = Valuenl
   ValueInvestor(i) = Valueinv
Next

C_ijt = zeros(NumberTotalSites, Banner_Max, Time_Max)


'The coefficient calculation, this is the most importat part, except from the
optimization fucntions

For i = 1 To NumberTotalSites
   For j = 1 To TotAmountOfAds(i)
      For t = 1 To TimeArray(i, j)
         C_ijt(i, j, t) = (1 - Prob(i, j) * Transpv(i, j) * ValuePageView(i) / Pageviews(i,
j) - Prob(i, j) * Transnl(i, j) * ValueNewsLetter(i) / Pageviews(i, j) - Prob(i, j) *
Transinv(i, j) * ValueInvestor(i) / Pageviews(i, j)) * t ^ (1 / 2) / t
      Next
   Next
Next



'Bivillkor
```

```
'sum xij <= Budget

b1 = CDbl(Budget)
b = zeros(NumberTotalSites * Banner_Max, Time_Max)
For i = 1 To NumberTotalSites * Banner_Max
   For j = 1 To Time_Max
      b(i, j) = b1
   Next
Next

'creating the matrix

A = zeros(NumberTotalSites * Banner_Max * Time_Max, NumberTotalSites *
Banner_Max * Time_Max)

For i = 1 To NumberTotalSites * Banner_Max * Time_Max
   For j = 1 To NumberTotalSites * Banner_Max * Time_Max
      A(i, j) = 1
   Next
Next


'A = ones(Time_Max * NumberTotalSites, NumberTotalSites * Time_Max)
lb = zeros(NumberOfSites * Time_Max * Banner_Max, 1)
ub = zeros(NumberOfSites * Time_Max * Banner_Max, 1)

For i = 1 To NumberTotalSites * Time_Max * Banner_Max
   ub(i) = b1
Next

q = zeros(NumberOfSites * Time_Max, NumberOfSites * Time_Max)
For i = 1 To NumberOfSites * Banner_Max * Time_Max
   q(i, i) = 0.001
Next

'Dim C_ijt_t As Matrix
C_ijt_t = transpose(C_ijt)
'The call to MatrixVB,, using the linear or quadratic solver
If Linear = 1 Then
   x = lp(C_ijt, A, b, lb, ub)
Else
   x = qp(q, C_ijt_t, A, b, lb)


End If

'Build the result matrix

For i = 1 To NumberTotalSites
   For j = 1 To Banner_Max
```

```
        For t = 1 To Time_Max 'TimeArray(i, j)

            steps = steps + 1

            ResultArray(i, j, t) = Format((x.r1(steps)), "00.00")
        Next
    Next
Next


'Display the result in the result window

Results.Clear
Results.AddItem ("The optimal solution is the following:")
Results.AddItem ("Parameters")

For i = 1 To NumberTotalSites
    For j = 1 To TotAmountOfAds(i)
        For t = 1 To TimeArray(i, j)
            res1 = SiteNameArray(i) & " " & BannerKind(i, j) & ", day " & t & " = " &
ResultArray(i, j, t)
            res2 = res1 & " " & "The number of visitors each day: "  &
CLng(ResultArray(i, j, t) / Pageviews(i, j))
            Results.AddItem (res2)
        Next
    Next
Next


End Sub
```

Appendix E

Testing procedures for evaluation of the LogFile Analyzer and the LogFile Stripper:

Since testing the application on the real web server is not possible the testing will be performed on a
PC on the EPO intranet. The test machine will be equipped with the same software as the real web server and the server is connected to a replica of the live database.

The tests aim to show the correctness of the program and the result. The test visitors have different age, gender and economical possibilities. This is done to show essential differences between visitors and how valuable the output from the Log Analyzer is.

Since no real testpersons are available some imaginary people have been constructed.

****
Test visitor1.
****
Name: Konrad Persson
Age: 65
Occupation: Retired from public service
Yearly income: < 150.000
Wealth: < 200.000
Internet experience: Low but increasing since the last two years.
Short description: Has worked at the railways for the last 40 years but is now retired. Has a low income but have some money in store.

****
Test visitor2.
****
Name: Sven Steen
Age: 55
Occupation: Founder and president of a mediumsize electronics company
Yearly income: > 1 Million
Wealth: > 5 Millions
Internet experience: Fairly good. Has been using the internet on and off for the last 5 years. Has previously bought stock over internet.
Short description: Welthy man that has a good hand with money. Divorced.

****
Test visitor3.
****
Name: Liisa Nord
Age: 50
Occupation: University teacher
Yearly income: ~ 350.000
Wealth: < 400.000

Internet experience: Fairly good, mostly email. Uses an internet bank but has never purchased over the internet before.
Short description: Medium welthy with good economy with both cash and stock.

****
Test visitor4.
****
Name: Thorbjörn Dahl
Age: 39
Occupation: Television producer
Yearly income: ~ 600.000
Wealth: < 600.000
Internet experience: Fairly good, uses internet bank and has a Nordnet account.
Short description: Big spender that likes to test new things.

****
Test visitor5.
****
Name: Stina Svahn
Age: 40
Occupation: Polititian
Yearly income: ~ 600.000
Wealth: < 400.000
Internet experience: Recently dicovered the internet opportunities. Has a medium interest in stock.
Short description: Single mother.

****
Test visitor6.
****
Name: Glenn Patric
Age: 35
Occupation: Freelansing investor. Founder of a internet consultant company that he recently sold with good profit.
Yearly income: 500.000 < > 5 Million
Wealth: ~ 30 Million
Internet experience: High. Pioneer in the internet consulting branch.
Short description: Visionary and a high risk taker.

****
Test visitor7.
****
Name: Sture Gren
Age: 33
Occupation: Carpenter
Yearly income: < 300.000
Wealth: < 200.000

Internet experience: Low, recently descowered the advantages internet stovk exchange.
Short description: Family man with stabile economy

****
Test visitor8.
****
Name: Niclas Bergh
Age: 28
Occupation: Newly examined from KTH. Working at VM-data as a consultant.
Yearly income: ~ 300.000
Wealth: < 100.000
Internet experience: High. Has previously perchased stock over the internet
Short description: Single with loan on appartment and from studies.

****
Test visitor9.
****
Name: Lena Danielson
Age: 30
Occupation: Nurse
Yearly income: < 200.000
Wealth: < 50.000
Internet experience:Low, with a recently developed interest of stock.
Short description: Married and two children.

****
Test visitor10.
****
Name: Peter Karlsson
Age: 28
Occupation: Architect
Yearly income: < 350.000
Wealth: 2 Million (recent heritage)
Internet experience: Medium.
Short description: Recently inherited a large amount of money and needs to invest some of it.

****
Test visitor11.
****
Name: Kenny Lagerfeldth
Age: 35
Occupation: Lawyer
Yearly income: ~ 800.000
Wealth: ~ 1.5 Million
Internet experience: Medium, has a Nordnet account and has previouosly bought and sold stock over the internet.
Short description: Familyman with stabile economy.

****
Test visitor12
****
Name: Slejpner Nordman
Age: 25
Occupation: Plummer
Yearly income: ~ 200.000
Wealth: ~ 10.000
Internet experience: Uses the internet frequently, mostly for gaming.
Short description: Limited funds

****
Test visitor13
****
Name: Sahra Fine
Age: 23
Occupation: Actress
Yearly income: < 100.000
Wealth: ~ 30.000
Internet experience: Fairly good, has no previous stock experience.
Short description: Limited funds.

****
Test visitor14
****
Name: Nalle Mann
Age: 24
Occupation: Owns a successfull cleaning company
Yearly income: 600.000
Wealth: ~ 1.3 Million
Internet experience: Good, has used the internet for several years for his business.
Short description: medium-high risk taker, single

****
Test visitor15
****
Name: Johhny Stark
Age: 20
Occupation: Student
Yearly income: < 50.000
Wealth: ~ 600.000
Internet experience: High
Short description: Has made a small fortune on internet stock.

Appendix F

The following is the source code of the LogFile Stripper written by Jens Jonsson as part of thesis project 2000-2001. The main idea of the program is to parse out the important parts of the log file and store it in a database. The code comes from three different VB forms.

```
*****Start Form1*****
'Global Variables
Public oRefOld As String
Public newFilePath As String

Private Sub Command1_Click()
   Form2.Visible = True
End Sub
Private Sub Command2_Click()
   Form1.Text1.Text = vbNewLine & "Please wait while the file is beeing processed!"
   Form1.Refresh
   dataInsert
End Sub

'   This function gets data from the log files and inserts it into the database
'   Written by Jens Jonsson 2000-12-19
'
Function getLogData(ByVal filePath As String)
   'Read in log file
   If filePath <> "" Then
      newFilePath = Left(filePath, 3) & "short" & Right(filePath, Len(filePath) - 3)
   End If

   Form1.SetFocus
   Dim oFileSysObj As New FileSystemObject
   Dim oTextStream As TextStream
   Dim oOutTextStream As TextStream
   Dim oDict As Dictionary
   Set oDict = New Dictionary

   Dim numberOfRows As Integer
   Dim vKey As Variant
   Dim Linetext As String
   Dim Logtext As String
   Text = ""
   Logtext = ""
   Set oTextStream = oFileSysObj.OpenTextFile(filePath)
   Set oTextStream2 = oFileSysObj.OpenTextFile(filePath)
   Set oOutTextStream = oFileSysObj.CreateTextFile(newFilePath, True)

   Form1.Text1.Text = vbNewLine & "Please wait while the file is beeing processed!"

   Form1.Refresh
```

```
    While Not oTextStream2.AtEndOfStream
        oTextStream2.SkipLine
    Wend
    numberOfRows = oTextStream2.Line
    oTextStream2.Close
    Form1.Text2.Visible = False
    Form1.ProgressBar1.Visible = True
    Form1.Frame1.Visible = True

    For i = 0 To 4
        oTextStream.SkipLine
    Next

    While Not oTextStream.AtEndOfStream
        If Not Search(oTextStream.ReadLine, oTextStream.Line, oDict) Then
            'sessionid fanns inte redan!
        End If

        Form1.ProgressBar1.Value = (oTextStream.Line / numberOfRows) * 100
        Form1.Label4.Caption = oTextStream.Line
        Form1.Label4.Refresh
    Wend


    'Print the selected rows
    For Each vKey In oDict
        'Form1.Text1.Text = Form1.Text1 & oDict(vKey)
        oOutTextStream.WriteLine (oDict(vKey))
    Next
    Form1.Text1.Text = "A new log file with core information has been created, click
to update database" & vbNewLine & vbNewLine & "Old Log File: " & filePath & ",
Total rows: " & numberOfRows & vbNewLine & "New Log File: " & newFilePath &
", Total rows: " & oDict.Count
    Form1.Command2.Enabled = True
    Form1.Command4.Enabled = True
    Form1.Text1.Refresh
    Set oDict = Nothing
    oOutTextStream.Close
    oTextStream.Close
End Function

'   This function searches for individual SessionId
'   Written by Jens Jonsson 2000-12-20
'
Function Search(ByVal Line As String, ByVal number As Long, ByRef oDict As
Dictionary) As Boolean
    Dim index As Integer
    Dim ref As Integer
    Dim aspsessionid As String
    Dim stringArray As Variant

    Dim oDate As String
    Dim oTime As String
    Dim oRequest As String
    Dim oUA As String
    Dim oCookie As String
    Dim oRef As String

    If Left(Line, 1) = "#" Then
        Search = False
        Exit Function
    End If

    stringArray = Split(Line)
    oDate = stringArray(0)
    oTime = stringArray(1)
    oRequest = stringArray(6)
    oUA = stringArray(13)
    oCookie = stringArray(14)
    oRef = stringArray(15)

    'Get the sessionid from the log-row

    index = InStr(Line, "ASPSESSIONID")
    ref = InStrRev(Line, "http://www.eposcapital.se")

    If index <> 0 Then
        aspsessionid = Mid(Line, index, 45)
        'Check if sessionid exists allready and if not insert it
        If oDict.Exists(aspsessionid) Then
            If InStr(oRef, "eposcapital") Or InStr(oRef, "-") Then
                ' or the refferer is blank, dont replace old.
                If oRefOld <> "" And oRefOld <> "-" Then
                    oRef = oRefOld
                    oRefOld = ""
                    oDict.Remove (aspsessionid)
                    oDict.Add aspsessionid, oDate & " " & oTime & " " & oRequest & " " &
oUA & " " & oCookie & " " & oRef '& vbNewLine
                End If
            Else
                'refferer is not from our site and is not blank
                oDict.Remove (aspsessionid)
                'oDict.Add aspsessionid, line & vbNewLine
                oDict.Add aspsessionid, oDate & " " & oTime & " " & oRequest & " " &
oUA & " " & oCookie & " " & oRef '& vbNewLine

            End If
            Search = True
        Else
```

```
         If oRefOld <> "" And oRefOld <> "-" Then
            oRef = oRefOld
            oRefOld = ""
         End If
         oDict.Add aspsessionid, oDate & " " & oTime & " " & oRequest & " " & oUA
& " " & oCookie & " " & oRef '& vbNewLine
         'oDict.Add aspsessionid, aspsessionid & vbNewLine
         Search = False
      End If
   Else
      If Not InStr(oRef, "eposcapital") And Not InStr(oRef, "-") Then
         oRefOld = oRef
      End If
      Search = True
   End If
End Function


'This function reads from the new log file and inserts the data into the database
'Written by Jens Jonsson 2000-12-27

Function dataInsert()
Dim oFileSysObj As New FileSystemObject
Dim oTextStream As TextStream
Dim oTextStream2 As TextStream

Dim oCon As New ADODB.Connection
Dim oCmd As New ADODB.Command
Dim o_vRecSet As ADODB.Recordset

Dim numberOfRows As Integer
Dim Line As String
Dim stringArray As Variant
Dim oDate As String
Dim oTime As String
Dim oRequest As String
Dim oUA As String
Dim oCookie As String
Dim oRef As String

oCon.IsolationLevel = adXactReadCommitted


Set oTextStream = oFileSysObj.OpenTextFile(newFilePath)
Set oTextStream2 = oFileSysObj.OpenTextFile(newFilePath)

While Not oTextStream2.AtEndOfStream
   oTextStream2.SkipLine
Wend
numberOfRows = oTextStream2.Line
```

```
oTextStream2.Close

While Not oTextStream.AtEndOfStream
   Line = oTextStream.ReadLine
   stringArray = Split(Line)
   oDate = stringArray(0)
   oTime = stringArray(1)
   oDate = oDate & " " & oTime
   oRequest = stringArray(2)
   oUA = stringArray(3)
   oCookie = stringArray(4)
   oRef = stringArray(5)
   If InStr(oCookie, "User=UID=") Then
      Dim stringArray2 As Variant
      stringArray2 = Split(oCookie, ";")
      Dim i As Variant
      For Each i In stringArray2
         If InStr(i, "User=UID=") Then
            oCookie = Mid(i, 10, Len(i))
         End If
      Next
   End If

   oCon.Open ("DRIVER={SQL
Server};SERVER=192.168.12.109;UID=sa;PWD=;DATABASE=EPOT7_DEV20SE
;")
   With oCmd
      .ActiveConnection = oCon
      .CommandType = adCmdStoredProc
      .CommandText = "usp_VisitorLogInsert"
      .Parameters.Append .CreateParameter("@CookieID", adVarChar, adParamInput,
256, oCookie)
      .Parameters.Append .CreateParameter("@LogDate", adChar, adParamInput, 20,
oDate)
      .Parameters.Append .CreateParameter("@UserAgent", adVarChar,
adParamInput, 256, oUA)
      .Parameters.Append .CreateParameter("@Demand", adVarChar, adParamInput,
256, oRequest)
      .Parameters.Append .CreateParameter("@Referer", adVarChar, adParamInput,
1024, oRef)
      .Execute
   End With
   oCon.Close
   Set oCmd = Nothing
   Form1.ProgressBar1.Value = (oTextStream.Line / numberOfRows) * 100
   Form1.Label4.Caption = oTextStream.Line - 1
   Form1.Label4.Refresh
Wend

Set oCon = Nothing
```

```
Form1.Text1.Text = vbNewLine & "The Database is now updated with " &
Form1.Label4.Caption & " rows of new log data!"
' Kontrollera om posten redan finns i databasen

' Om den inte finns, lägg till den

End Function

Private Sub Command3_Click()
   Form1.Visible = False
End Sub

Private Sub Command4_Click()
   If newFilePath <> "" Then
      Form3.viewFile newFilePath
   End If
End Sub

Private Sub Form_Load()
   Form1.Label1.Caption = "Log File Stripper " & Chr(169)
   Form1.StatusBar1.SimpleText = "Line"
End Sub

******End of Form1******

******Start Form2******
Private Sub Command1_Click()
   Form2.Visible = False
   Form1.getLogData Form2.Dir1.path & Form2.File1.FileName
End Sub

Private Sub Command2_Click()
   Form2.Visible = False
End Sub

Private Sub Dir1_Change()
   Form2.File1.path = Form2.Dir1.path
End Sub

Private Sub Drive1_Change()
   Form2.Dir1.path = Form2.Drive1.Drive
   Form2.File1.path = Form2.Drive1.Drive
End Sub

Private Sub File1_DblClick()
   Form2.Visible = False
   Form1.getLogData Form2.Dir1.path & Form2.File1.FileName
End Sub
```

```
Private Sub File1_GotFocus()
   Form2.Command1.Enabled = True
End Sub

Private Sub File1_LostFocus()
   'Form2.Command1.Enabled = False
End Sub

******End Form2******

******Start Form3******
Function viewFile(ByVal path As String)
Dim MyAppID
   MyAppID = Shell("C:\Program Files\Windows NT\Accessories\wordpad.exe " &
path, 1)   ' Run Wordpad.
   AppActivate MyAppID
End Function
******End Form3******
```

Appendix G

The following is the source code of the LogFile Analyzer, written by Jens Jonsson as part of thesis project 2000-2001. The main idea of the program is to graphically give the user a statistical picture of the visitors of the website by analyzing log data that is retrieved from a database. The code contains functions from 11 different VB forms.

```
*****Start Analyzer Form*****
Private Sub menu1_1_Click()
   OpenFile.Visible = True
End Sub

Private Sub menu1_2_Click()
'This function creates a new trace file
   NewTrace.Visible = True
End Sub

Function loadFile(ByVal path As String)
Dim oFileSysObj As New FileSystemObject
Dim oInTextStream As TextStream
Dim filePath As String
Dim Index As Variant
Dim Line As String
Dim oDict As Dictionary
Set oDict = New Dictionary

Set oInTextStream = oFileSysObj.OpenTextFile(path)
   oDict.Add "Path", path
   While Not oInTextStream.AtEndOfStream
      Line = oInTextStream.ReadLine
      Index = InStr(Line, ":")
      oDict.Add Left(Line, Index - 1), Mid(Line, Index + 1)
   Wend
   oInTextStream.Close
   Analyzer.Refresh
   TraceForm.loadProject oDict
End Function


Private Sub menu3_2_Click()
   Referer.Visible = True
End Sub

Private Sub menu3_3_Click()
   SearchEngines.Visible = True
End Sub
*****End Form*****
```

```
*****Start Trace Form*****
Function loadProject(ByVal oDict As Dictionary)
Dim vKey As Variant
   Text2.Text = oDict("Path")
   Text3.Text = oDict("StartDate") & " " & oDict("StartTime")
   Text4.Text = oDict("EndDate") & " " & oDict("EndTime")
   Text7.Text = oDict("StartDate") & " " & oDict("StartTime")
   Text15.Text = oDict("EndDate") & " " & oDict("EndTime")
   If oDict("Type") <> "All" Then
      Option2 = True
      Text6.Text = oDict("referer")
      Text8.Text = oDict("Investor Id")
      getLogData Text3.Text, Text4.Text, oDict("referer")
   Else
      Option1 = True
      getLogData Text3.Text, Text4.Text
   End If
   TraceForm.Refresh
   TraceForm.Visible = True
End Function

Function getLogData(ByVal StartDate As String, ByVal EndDate As String, Optional
ByVal Referer, Optional ByVal CoookieID, Optional ByVal InvestorID)
Dim oCon As New ADODB.Connection
Dim oCmd As New ADODB.Command
Dim recSet As ADODB.Recordset
Dim oCon2 As New ADODB.Connection
Dim oCmd2 As New ADODB.Command
Dim recSet2 As ADODB.Recordset

Dim investorDict As Dictionary
Set investorDict = New Dictionary
Dim investorDateDict As Dictionary
Set investorDateDict = New Dictionary
Dim UniqueDict As Dictionary
Set UniqueDict = New Dictionary

Dim number As Integer
Dim refDict As New Dictionary
Dim ref2Dict As New Dictionary
Dim refDictSorted As New Dictionary
Dim searchDictSorted As New Dictionary
Dim searchDict As New Dictionary

Dim refLine As Variant
Dim ref2Line As Variant
Dim searchLine As Variant
Dim cookieLine As Variant
Dim vKey As Variant
Dim vKey2 As Variant
Dim vKey3 As Variant
Dim numberOfReferers As Variant
Dim FirstVisit As Variant
Dim AverageVisit As Variant

   oCon.IsolationLevel = adXactReadCommitted
   oCon.Open ("DRIVER={SQL
Server};SERVER=192.168.12.109;UID=sa;PWD=;DATABASE=EPOT7_DEV20SE
;")
   With oCmd
      .ActiveConnection = oCon
      .CommandType = adCmdStoredProc
      .CommandText = "usp_VisitorLogGet"
      .Parameters.Append .CreateParameter("@StartDate", adChar, adParamInput, 20,
StartDate)
      .Parameters.Append .CreateParameter("@EndDate", adChar, adParamInput, 20,
EndDate)
      Set recSet = .Execute
   End With
   Text1.Text = ""
   number = 0
   If recSet.EOF Then
      Text1.Text = "recordset is empty " & StartDate & " " & EndDate
   Else
      If Not IsMissing(Referer) Then

         While Not recSet.EOF
            If InStr(recSet("Referer"), Referer) Then
               Text1.Text = Text1.Text & recSet("Referer") & vbNewLine
               number = number + 1
            End If
            recSet.MoveNext
         Wend
      Else
      ' Ingen speciell referer
      'Hämta alla besökare som blivit investerare under perioden


      Dim invrefDict As New Dictionary

      If getInvestorData(StartDate, EndDate, investorDict, investorDateDict) Then
      'Hämta all besökar data från data basen som har cookieID som är samma som i
investorDict

         vKey2 = investorDict.Keys

         If investorDict.Exists(CStr(vKey2(0))) Then
            Dim tempTest As Variant
            tempTest = investorDict.Item(CStr(vKey2(0)))
         End If
```

```
            oCon2.IsolationLevel = adXactReadCommitted
            oCon2.Open ("DRIVER={SQL
Server};SERVER=192.168.12.109;UID=sa;PWD=;DATABASE=EPOT7_DEV20SE
;")
            For i = 0 To investorDict.Count - 1
                With oCmd2
                    .ActiveConnection = oCon2
                    .CommandType = adCmdStoredProc
                    .CommandText = "usp_VisitorLogGetInvestor"
                    .Parameters.Append .CreateParameter("@CookieID", adChar,
adParamInput, 255, CStr(vKey2(i)))
                    Set recSet2 = .Execute
                    .Parameters.Delete (0)
                End With

                'Add investorID,  to listbox1
                List1.AddItem (investorDict.Item(CStr(vKey2(i))))
                'Add MemberDate to Listbox2
                List2.AddItem (investorDateDict.Item(CStr(vKey2(i))))
                'Add FirstVisit to Listbox3
                List3.AddItem (recSet2("LogDate"))
                'Add referer to listbox5
                List5.AddItem (recSet2("referer"))
                'Add TimesVisited to listbox4
                If recSet2("TimesVisited") > 0 Then
                    List4.AddItem (recSet2("TimesVisited"))
                Else
                    List4.AddItem ("first time")
                End If
                If Not invrefDict.Exists(CStr(recSet2("referer"))) Then
                    'Put referer into a dictionary
                    invrefDict.Add CStr(recSet2("referer")), 1
                Else
                    invrefDict.Item(CStr(recSet2("referer"))) =
invrefDict.Item(CStr(recSet2("referer"))) + 1
                End If
            Next
            oCon2.Close

            Label14 = investorDict.Count
            Label21 = investorDict.Count
            numberOfReferers = 0
        End If

        While Not recSet.EOF
            'Text1.Text = Text1.Text & recSet("Referer") & vbNewLine
            refLine = recSet("Referer")
            cookieLine = recSet("CookieID")
```

```
    If refDict.Exists(refLine) Then             'If referer allready exists then up one
       refDict(refLine) = refDict(refLine) + 1
    Else
       refDict.Add refLine, 1                   'Else add and give value 1.
    End If

    'Count all referers that is not EPO
    If InStr(recSet("Referer"), "developer") = 0 Then
       numberOfReferers = numberOfReferers + 1
    End If

    'Count the visitors that are here for the first time
    If recSet("TimesVisited") >= 1 Then
       FirstVisit = FirstVisit + 1
       AverageVisit = recSet("TimesVisited") + AverageVisit
    End If

    number = number + 1
    recSet.MoveNext
 Wend
 'Put every referer in the referer file into a dictionary
 Dim oFileSysObj As New FileSystemObject
 Dim oInTextStream As TextStream
 Dim Index As Variant
 Dim Line As String

 Set oInTextStream = oFileSysObj.OpenTextFile("d:/referer.txt")

 'Läs in alla referers från filen och lägg dessa i ett dictionary
 While Not oInTextStream.AtEndOfStream
    ref2Line = oInTextStream.ReadLine
    ref2Dict.Add ref2Line, ref2Line
 Wend
 oInTextStream.Close
 Set oInTextStream = oFileSysObj.OpenTextFile("d:/engines.txt")

 'Läs in alla search engines från filen och lägg dessa i ett dictionary
 While Not oInTextStream.AtEndOfStream
    searchLine = oInTextStream.ReadLine
    searchDict.Add searchLine, searchLine
 Wend
 oInTextStream.Close

 'Check every referer in the dictionary with the usual referers in the referer file.
 Dim vArray As Variant
 Dim vArrayRef As Variant
 Dim vArraySearch As Variant

 vArray = refDict.Keys
 vArrayRef = ref2Dict.Keys
```

```
    vArraySearch = searchDict.Keys

    For i = 0 To refDict.Count - 1
      'For the referers
      For i2 = 0 To ref2Dict.Count - 1
        If InStr(vArray(i), ref2Dict(vArrayRef(i2))) Then
          If refDictSorted.Exists(ref2Dict(vArrayRef(i2))) Then
            refDictSorted.Item(ref2Dict(vArrayRef(i2))) =
refDictSorted.Item(ref2Dict(vArrayRef(i2))) + refDict(vArray(i))
          Else
            refDictSorted.Add ref2Dict(vArrayRef(i2)), refDict(vArray(i))
          End If
        End If
      Next
      'For the search engines
      For i3 = 0 To searchDict.Count - 1
        If InStr(vArray(i), searchDict(vArraySearch(i3))) Then
          If searchDictSorted.Exists(searchDict(vArraySearch(i3))) Then
            searchDictSorted.Item(searchDict(vArraySearch(i3))) =
searchDictSorted.Item(searchDict(vArraySearch(i3))) + refDict(vArray(i))
          Else
            searchDictSorted.Add searchDict(vArraySearch(i3)),
refDict(vArray(i))
          End If
        End If
      Next
    Next

    Dim vArrayInvRef As Variant
    vArrayInvRef = invrefDict.Keys
    Dim invrefSortedDict As New Dictionary

    For i4 = 0 To invrefDict.Count - 1
      For i5 = 0 To ref2Dict.Count - 1
        If InStr(vArrayInvRef(i4), ref2Dict(vArrayRef(i5))) Then
          If invrefSortedDict.Exists(ref2Dict(vArrayRef(i5))) Then
            invrefSortedDict.Item(ref2Dict(vArrayRef(i5))) =
invrefSortedDict.Item(ref2Dict(vArrayRef(i5))) + invrefDict(vArrayInvRef(i4))
          Else
            invrefSortedDict.Add ref2Dict(vArrayRef(i5)),
invrefDict(vArrayInvRef(i4))
          End If
        End If
      Next
    Next


    'Get the most frequently refering site and search engine
    Dim refHitrate As Variant
```

```
        refHitrate = 0
        Dim invrefHitrate As Variant
        invrefHitrate = 0
        Dim searchHitrate As Variant
        searchHitrate = 0
        Dim bestRef As Variant
        Dim bestinvRef As Variant
        Dim bestSearch As Variant
        Dim vArray2 As Variant
        vArray2 = refDictSorted.Keys
        Dim vArray3 As Variant
        vArray3 = invrefSortedDict.Keys
        Dim vArray4 As Variant
        vArray4 = invrefSortedDict.Keys
        Dim searchArray As Variant
        searchArray = searchDictSorted.Keys
        Dim totRefHits As Variant
        totRefHits = 0
        Dim totSearchHits As Variant
        totSearchHits = 0

        'Reset the text area
        Text9.Text = ""
        Text10.Text = ""
        Text11.Text = ""
        Text12.Text = ""

        If Not refDictSorted.Count = 0 Then
            MSChart1.ChartData = Array(numberOfReferers, number -
numberOfReferers)
            MSChart1.DataGrid.RowLabel(1, 1) = "Refered/Not refered"
            MSChart1.DataGrid.ColumnLabel(1, 1) = "Refered"
            MSChart1.DataGrid.ColumnLabel(2, 1) = "Not ref."

            MSChart2.ChartData = Array(FirstVisit, number - FirstVisit)
            MSChart2.DataGrid.RowLabel(1, 1) = "First visit/Here before"
            MSChart2.DataGrid.ColumnLabel(1, 1) = "Here before"
            MSChart2.DataGrid.ColumnLabel(2, 1) = "First visit."

            MSChart4.Visible = True
            MSChart5.Visible = True
            MSChart6.Visible = True
            MSChart4.ChartData = refDictSorted.Items
            MSChart4.ColumnCount = refDictSorted.Count
            MSChart4.DataGrid.RowLabel(1, 1) = "Referers"
            MSChart6.ChartData = invrefSortedDict.Items
            MSChart6.ColumnCount = invrefSortedDict.Count
            MSChart6.DataGrid.RowLabel(1, 1) = "Top Referers"

            If Not searchDictSorted.Count = 0 Then
```

```
   MSChart5.ChartData = searchDictSorted.Items
   MSChart5.ColumnCount = searchDictSorted.Count
End If

For i = 1 To refDictSorted.Count
   MSChart4.DataGrid.ColumnLabel(i, 1) = vArray2(i - 1)
Next

For i = 1 To invrefSortedDict.Count
   MSChart6.DataGrid.ColumnLabel(i, 1) = vArray3(i - 1)
Next

If Not searchDictSorted.Count = 0 Then
   For i = 1 To searchDictSorted.Count
      MSChart5.DataGrid.ColumnLabel(i, 1) = searchArray(i - 1)
   Next
   MSChart5.Plot.Sort = VtSortTypeDescending
End If
MSChart4.Plot.Sort = VtSortTypeDescending


For i = 0 To refDictSorted.Count - 1
   Text9.Text = Text9.Text & vArray2(i) & vbNewLine
   Text10.Text = Text10 & refDictSorted(vArray2(i)) & vbNewLine
   totRefHits = totRefHits + refDictSorted(vArray2(i))
   If refHitrate < refDictSorted(vArray2(i)) Then
      refHitrate = refDictSorted(vArray2(i))
      bestRef = vArray2(i)
   End If
Next

For i = 0 To searchDictSorted.Count - 1
   Text11.Text = Text11.Text & searchArray(i) & vbNewLine
   Text12.Text = Text12 & searchDictSorted(searchArray(i)) & vbNewLine
   totSearchHits = totSearchHits + searchDictSorted(searchArray(i))
   If searchHitrate < searchDictSorted(searchArray(i)) Then
      searchHitrate = searchDictSorted(searchArray(i))
      bestSearch = searchArray(i)
   End If
Next

For i = 0 To invrefSortedDict.Count - 1
   If invrefHitrate < invrefSortedDict(vArray4(i)) Then
      invrefHitrate = invrefSortedDict(vArray4(i))
      bestinvRef = vArray4(i)
   End If
Next

Label10.Caption = bestRef & " " & refHitrate
Text10.Text = Text10.Text & "--------" & vbNewLine & "Tot: " & totRefHits
```

```
            Label13.Caption = bestSearch & " " & searchHitrate
            Text12.Text = Text12.Text & "--------" & vbNewLine & "Tot: " &
totSearchHits
            Label23.Caption = bestinvRef & " " & invrefHitrate
        Else
            MSChart4.Enabled = False
            MSChart5.Enabled = False
            MSChart4.Visible = False
            MSChart5.Visible = False
            Label10.Caption = "non existent!"
            Text10.Text = "No referers!"
            Label13.Caption = "non existent!"
            Text12.Text = "No search engines!"
        End If
    End If
End If
oCon.Close
Set oCmd = Nothing
Set oCon = Nothing

Label9.Caption = number
If number <> 0 Then
    Label34.Caption = Round(AverageVisit / number)
End If

'Investor stuff
'Set recSet2 = New Recordset

' Get Failed signings

Dim Failcount As Variant
Failcount = 0
Dim UniqueFail As Variant
UniqueFail = 0

    oCon.Open ("DRIVER={SQL
Server};SERVER=192.168.12.109;UID=sa;PWD=;DATABASE=EPOT7_DEV20SE
;")
        With oCmd
            .ActiveConnection = oCon
            .CommandType = adCmdStoredProc
            .CommandText = "usp_VisitorToInvestorFailGet"
            .Parameters.Append .CreateParameter("@StartDate", adChar, adParamInput,
20, StartDate)
            .Parameters.Append .CreateParameter("@EndDate", adChar, adParamInput,
20, EndDate)
            Set recSet = .Execute
            .Parameters.Delete (0)
        End With
```

```
      While Not recSet.EOF
         List6.AddItem (recSet("CookieID"))
         List7.AddItem (recSet("Error"))
         List8.AddItem (recSet("FailDate"))

         'Count number of failures
         If Not UniqueDict.Exists(CStr(recSet("CookieID"))) Then
            UniqueDict.Add CStr(recSet("CookieID")), recSet("CookieID")
         End If
         Failcount = Failcount + 1
         recSet.MoveNext
      Wend
      Label38.Caption = Failcount
      Label41.Caption = UniqueDict.Count

   oCon.Close
   Set oCmd = Nothing
   Set oCon = Nothing
'End Failed Signings

End Function

Function getInvestorData(ByVal StartDate As String, ByVal EndDate As String,
ByVal investorDict2 As Dictionary, ByVal investorDateDict2 As Dictionary) As
Boolean
Dim oCon As New ADODB.Connection
Dim oCmd As New ADODB.Command
Dim recSet As ADODB.Recordset
Dim tempString As Variant
Dim NewsLetter As Variant
NewsLetter = 0

   oCon.IsolationLevel = adXactReadCommitted
   oCon.Open ("DRIVER={SQL
Server};SERVER=192.168.12.109;UID=sa;PWD=;DATABASE=EPOT7_DEV20SE
;")
   With oCmd
      .ActiveConnection = oCon
      .CommandType = adCmdStoredProc
      .CommandText = "usp_VisitorToInvestorGet"
      .Parameters.Append .CreateParameter("@StartDate", adChar, adParamInput, 20,
StartDate)
      .Parameters.Append .CreateParameter("@EndDate", adChar, adParamInput, 20,
EndDate)
      .Execute
   End With

   Set recSet = New ADODB.Recordset
   recSet.CursorLocation = adUseClient
   recSet.Open oCmd
```

```
    If recSet.EOF Then
        Text1.Text = "recordset is empty " & StartDate & " " & EndDate
        getInvestorData = False
    Else
        'Spara all investor data i ett dictionary
        While Not recSet.EOF
            'List1.AddItem recSet("InvestorID")
            'List2.AddItem recSet("MemberDate")
            If recSet("NewsLetter") = True Then
                NewsLetter = NewsLetter + 1
            End If
                investorDict2.Add CStr(recSet("CookieID")), CStr(recSet("InvestorID"))
                investorDateDict2.Add CStr(recSet("CookieID")),
CStr(recSet("MemberDate"))
        recSet.MoveNext
        Wend
        getInvestorData = True
    End If

    Dim vKey2 As Variant
    vKey2 = investorDict2.Keys
    For i = 0 To investorDict2.Count - 1
    Next

    Label16.Caption = NewsLetter
    'recSet = Nothing
    recSet.ActiveConnection = Nothing
    recSet.Close
    oCon.Close



End Function
Function setCompDate1(sdate As Date)
    Label29.Caption = sdate
    Label29.Visible = True
End Function

Function setCompDate2(sdate As Date)
    Label30.Caption = sdate
    Label30.Visible = True
End Function


Private Sub Check2_Click()


End Sub

Private Sub Command1_Click()
```

```
'Clear all fields
Label9.Caption = ""
Label10.Caption = ""
Label13.Caption = ""
Label14.Caption = ""
Label16.Caption = ""
Label34.Caption = ""
Label21.Caption = ""
Label23.Caption = ""
Label38.Caption = ""
Label41.Caption = ""

List1.Clear
List2.Clear
List3.Clear
List4.Clear
List5.Clear
List6.Clear
List7.Clear
List8.Clear

Text11.Text = ""
Text12.Text = ""
Text9.Text = ""
Text10.Text = ""

'saveFile
Dim oFileSysObj As New FileSystemObject
Dim oOutTextStream As TextStream

  Set oOutTextStream = oFileSysObj.CreateTextFile(Text2.Text, True)

  oOutTextStream.WriteLine "StartDate:" & Text3.Text
  oOutTextStream.WriteLine "EndDate:" & Text4.Text

  If Option1 Then
    'If all files are choosen
    oOutTextStream.WriteLine "Type:All"
    getLogData Text3.Text, Text4.Text
  Else
    oOutTextStream.WriteLine "Type:Specified"
    oOutTextStream.WriteLine "referer:" & Text6.Text
    oOutTextStream.WriteLine "Cookie Id:" & Text7.Text
    oOutTextStream.WriteLine "Investor Id:" & Text8.Text
    getLogData Text3.Text, Text4.Text, Text6.Text
  End If

  oOutTextStream.Close
  TraceForm.Refresh
End Sub
```

```
Private Sub Command2_Click()
    Frame3.Visible = False
    If Frame4.Visible Then
        Frame4.Top = 5040
    End If
End Sub

Private Sub Command3_Click()
    Frame4.Visible = False
    If Frame3.Visible Then
        Frame3.Top = 5040
    End If
End Sub

Private Sub Command4_Click()
    Frame5.Visible = False
    If Frame7.Visible Then
        Frame7.Top = 5040
    End If
End Sub

Private Sub Command5_Click()
    Frame6.Visible = False
    If Frame5.Visible Then
        Frame5.Top = 5040
    End If
End Sub


Private Sub Command6_Click()
    Frame7.Visible = False
    If Frame5.Visible Then
        Frame5.Top = 5040
    End If
End Sub

Private Sub List1_DblClick()
    Dim MyAppID
    Dim path
    path =
"https://www.epo.com/epoCommon/Manager/investor/man_inv_edit_account.asp?co
untry=SE&id=" & List1.Text
    MyAppID = Shell("C:\Program Files\Plus!\Microsoft Internet\iexplore.exe " &
path, 1)    ' Run Explorer.
    AppActivate MyAppID

End Sub
```

```
Private Sub menu1_Click()
   Frame5.Visible = False
   Frame7.Visible = False
   If Frame4.Visible And Frame3.Visible Then
   Else
      If Frame4.Visible Then
         Frame3.Top = Frame4.Top + Frame4.Height + 250
      Else
         Frame3.Top = 5040
      End If
      Frame3.Visible = True
   End If
End Sub

Private Sub menu2_Click()
   Frame5.Visible = False
   Frame7.Visible = False
   If Frame4.Visible And Frame3.Visible Then
   Else
      If Frame3.Visible Then
         Frame4.Top = Frame3.Top + Frame3.Height + 250
      Else
         Frame4.Top = 5040
      End If
      Frame4.Visible = True
   End If
End Sub

Private Sub menu3_Click()
   If Frame5.Visible And Frame7.Visible Then
   Else
      Frame6.Visible = False
      Frame3.Visible = False
      Frame4.Visible = False
      If Frame7.Visible Then
            Frame5.Top = Frame7.Top + Frame7.Height + 250
      ElseIf Frame5.Visible Then
         Frame7.Top = Frame5.Top + Frame5.Height + 250
      Else
         Frame5.Top = 5040
         Frame7.Top = 9000
      End If
      Frame7.Visible = True
      Frame5.Visible = True
   End If
End Sub

Private Sub menu4_Click()
   Frame5.Visible = False
```

```
    Frame6.Visible = True
End Sub

Private Sub Option1_Click()
    Option2 = False
    Label17.Enabled = True
    Label19.Enabled = False
    Label3.Enabled = False
    Label4.Enabled = False
    Label6.Enabled = False
    Text6.Visible = False
    Text7.Visible = False
    Text15.Visible = False
    Text8.Visible = False
End Sub

Private Sub Option2_Click()
    Option1 = False
    Label17.Enabled = False
    Label19.Enabled = True
    Label3.Enabled = True
    Label4.Enabled = True
    Label6.Enabled = True
    Text6.Visible = True
    Text7.Visible = True
    Text15.Visible = True
    Text8.Visible = True
End Sub
*****End Form*****
*****Start NewTrace Form*****
Private Sub Check1_Click()
    If Check1 Then
        Check2 = False
        NewTrace.Check3 = False
        NewTrace.Check4 = False
        NewTrace.Check5 = False
        NewTrace.Text11.Enabled = False
        NewTrace.Text12.Enabled = False
        NewTrace.Text13.Enabled = False
    End If
End Sub

Private Sub Check2_Click()
    If Check2 Then
        NewTrace.Check3.Enabled = True
        NewTrace.Check4.Enabled = True
        NewTrace.Check5.Enabled = True
        Check1 = False
    Else
        NewTrace.Check3.Enabled = False
        NewTrace.Check4.Enabled = False
        NewTrace.Check5.Enabled = False
    End If
End Sub

Private Sub Check3_Click()

    If Check3 Then
        NewTrace.Label3.Enabled = True
        NewTrace.Text11.Enabled = True
    Else
        NewTrace.Label3.Enabled = False
        NewTrace.Text11.Enabled = False
    End If
End Sub

Private Sub Check4_Click()
    If Check4 Then
        NewTrace.Label4.Enabled = True
        NewTrace.Text12.Enabled = True
    Else
        NewTrace.Label4.Enabled = False
        NewTrace.Text12.Enabled = False
    End If
End Sub

Private Sub Check5_Click()
    If Check5 Then
        NewTrace.Label5.Enabled = True
        NewTrace.Text13.Enabled = True
    Else
        NewTrace.Label5.Enabled = False
        NewTrace.Text13.Enabled = False
    End If
End Sub

Private Sub Command1_Click()
    Calendar1.Visible = True
End Sub

Private Sub Command2_Click()
     Calendar2.Visible = True
End Sub

Function setStartDate(sdate As Date)
    NewTrace.Text1.Text = Year(sdate)
    NewTrace.Text2.Text = Month(sdate)
    NewTrace.Text3.Text = Day(sdate)
    NewTrace.Refresh
End Function
```

```
Function setEndDate(edate As Date)
   NewTrace.Text4.Text = Year(edate)
   NewTrace.Text5.Text = Month(edate)
   NewTrace.Text6.Text = Day(edate)
   NewTrace.Refresh
End Function

Private Sub Command3_Click()
   Save.Visible = True
End Sub

Private Sub Command4_Click()
   NewTrace.Visible = False
End Sub
*****End Form*****

*****Start OpenFile Form*****
Private Sub Command1_Click()
   OpenFile.Visible = False
   Analyzer.loadFile OpenFile.Dir1.path & OpenFile.File1.FileName
End Sub

Private Sub Command2_Click()
   OpenFile.Visible = False
End Sub

Private Sub Dir1_Change()
   OpenFile.File1.path = OpenFile.Dir1.path
End Sub

Private Sub Drive1_Change()
   OpenFile.Dir1.path = OpenFile.Drive1.Drive
   OpenFile.File1.path = OpenFile.Drive1.Drive
End Sub

Private Sub File1_DblClick()
   OpenFile.Visible = False
   Analyzer.loadFile OpenFile.Dir1.path & OpenFile.File1.FileName
End Sub
*****End Form*****

*****Start Referrer Form*****
Private Sub Command1_Click()
Dim oFileSysObj As New FileSystemObject
Dim oOutTextStream As TextStream
Dim Index As Variant
Dim Line As String

   'Save on file
   Set oOutTextStream = oFileSysObj.CreateTextFile("d:/referer.txt")
   oOutTextStream.WriteLine (Text1.Text)
   oOutTextStream.Close
   Referer.Refresh
End Sub

Private Sub Command2_Click()
   Referer.Visible = False
End Sub

Private Sub Form_Load()
Dim oFileSysObj As New FileSystemObject
Dim oInTextStream As TextStream
Dim filePath As String
Dim Index As Variant
Dim Line As String

   Set oInTextStream = oFileSysObj.OpenTextFile("d:/referer.txt")
   While Not oInTextStream.AtEndOfStream
      Text1.Text = Text1.Text & oInTextStream.ReadLine & vbNewLine
   Wend
   oInTextStream.Close
   Referer.Refresh
End Sub
*****End Form*****

*****Start Save Form*****

Private Sub Command1_Click()
Dim oFileSysObj As New FileSystemObject
Dim oOutTextStream As TextStream
Dim filePath As String

   filePath = Save.Dir1.path & Save.Text1.Text
   Set oOutTextStream = oFileSysObj.CreateTextFile(filePath, True)

   oOutTextStream.WriteLine "StartDate:" & NewTrace.Text1 & "-" &
NewTrace.Text2 & "-" & NewTrace.Text3 & " " & NewTrace.Text7 & ":" &
NewTrace.Text8
   oOutTextStream.WriteLine "EndDate:" & NewTrace.Text4 & "-" &
NewTrace.Text5 & "-" & NewTrace.Text6 & " " & NewTrace.Text9 & ":" &
NewTrace.Text10

   If NewTrace.Check1 Then
      'If all files are choosen
      oOutTextStream.WriteLine "Type:All"
   Else
      oOutTextStream.WriteLine "Type:Specified"
   End If
   'If Specified type is choosen
   If NewTrace.Check3 Then
      oOutTextStream.WriteLine "referer:" & NewTrace.Text11.Text
   End If
   If NewTrace.Check4 Then
      oOutTextStream.WriteLine "Cookie Id:" & NewTrace.Text12.Text
   End If
   If NewTrace.Check5 Then
      oOutTextStream.WriteLine "Investor Id:" & NewTrace.Text13.Text
   End If

   oOutTextStream.Close

   Save.Visible = False
   NewTrace.Visible = False

End Sub

Private Sub Command2_Click()
   Save.Visible = False
End Sub

Private Sub Dir1_Change()
   Save.File1.path = Save.Dir1.path
End Sub

Private Sub Drive1_Change()
   Save.Dir1.path = Save.Drive1.Drive
   Save.File1.path = Save.Drive1.Drive
End Sub

Private Sub File1_DblClick()
   Save.Text1.Text = Save.File1.FileName
End Sub


Private Sub File1_GotFocus()
   NewTrace.Command1.Enabled = True
End Sub

Private Sub File1_LostFocus()
   'NewTrace.Command1.Enabled = False
End Sub
*****End Form*****

*****Start SearchEngine Form*****
Private Sub Command1_Click()
Dim oFileSysObj As New FileSystemObject
Dim oOutTextStream As TextStream
Dim Index As Variant
Dim Line As String

   'Save on file
   Set oOutTextStream = oFileSysObj.CreateTextFile("d:/engines.txt")
   oOutTextStream.WriteLine (Text1.Text)
   oOutTextStream.Close
   SearchEngines.Refresh
End Sub

Private Sub Command2_Click()
   SearchEngines.Visible = False
End Sub

Private Sub Form_Load()
Dim oFileSysObj As New FileSystemObject
Dim oInTextStream As TextStream
Dim filePath As String
Dim Index As Variant
Dim Line As String

   Set oInTextStream = oFileSysObj.OpenTextFile("d:/engines.txt")
   While Not oInTextStream.AtEndOfStream
      Text1.Text = Text1.Text & oInTextStream.ReadLine & vbNewLine
   Wend
   oInTextStream.Close
   SearchEngines.Refresh
End Sub
*****End Form*****

*****Start Calendar1 Form*****
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
   NewTrace.setStartDate DateClicked
   Calendar1.Visible = False
End Sub
*****End Form*****

*****Start Calendar 2 Form*****
Private Sub MonthView2_DateClick(ByVal DateClicked As Date)
   NewTrace.setEndDate DateClicked
   Calendar2.Visible = False
End Sub
*****End Form*****

*****Start Calendar 3 Form*****
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
   TraceForm.setCompDate1 DateClicked
   Calendar3.Visible = False
End Sub
*****End Form*****
```

```
*****Start Calendar 4 Form*****
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
   TraceForm.setCompDate2 DateClicked
   Calendar4.Visible = False
End Sub
*****End Form*****
```

References

[1] When Exposure-Based Web Advertising Stops Making Sense, Donna L. Hoffman and Thomas P. Novak

[2] Measuring Internet Advertising Effectiveness, Lars Berkvist and Jonas Melander

[3] IAB online advertising effectiveness study, Joint research effort of Internet advertising bureau and Millard Brown Interactive

[4] Modeling the Click stream: Implications for Web-Based Advertising Efforts, Patrali Chatterjee, Donna L. Hoffman and Thomas P. Novak

[5] An entropy approach to unintrusive targeted advertising an the web, Computer-Networks vol.33, no.1-6; June 2000; p.767-74, Tomlin J.A.

[6] A framework for the optimizing of WWW advertising, International IFIP/GI Working conference TREC'98, Aggarwal C.C. , Wolf J.L. and Yu P.S.

[7] New Metrics for New Media: Toward the Development of Web Measurement Standards, Thomas P. Novak and Donna L. Hoffman

[8] What Are the Internet Based Advertising Models? James R. Lussier and Marita Valdmanis

[9] MB Interactive: A Roadmap to Online Strategy, http://www.mbinteractive.com/site/roadmap.html

[10] Web site optimization, John Garofalakis, Panagiotis Kappas and Dimitris Mourloukos http://computer.org/internet

[11] Unintrusive customization techniques for web advertising, Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba and Yoshiyuki Koseki

[12] Introduction to operations research, Frederick S. Hillier and Gerald J. Lieberman

[13] Banner ads on the Internet, http://www.wilsonweb.com/articles/bannerad.html

[14] Tracking visitors by Bill Winett, http://webdesign.about.com/compute/webdesign/gi/dynamic/offsite.htm?site=http://hotwired.lycos.com/webmonkey/e%2Dbusiness/tracking/tutorials/tutorial2.html

[15] Demystify your log files by Olufemi Anthony, http://www.builder.com/Servers/LogFile/index.html

[16] Adding cookies to your site by Paul Bonner, http://webdesign.about.com/compute/webdesign/gi/dynamic/offsite.htm?site=http://www.builder.com/Programming/Cookies/splash.html

[17] Problems with analyzing log data by About.com, http://webdesign.about.com/compute/webdesign/library/weekly/aa021600d.htm

[18] Measuring web site usage: Log file analysis by Susan Haigh and Janette Megarity, 1998, http://www.nlc-bnc.ca/pubs/netnotes/notes57.htm

[19] Load balancing, http://www.webtechniques.com/archives/1998/05/engelschall/

[20] VB & VBA in a nutshell: The language, Paul Lomax , O'REILLY

[21] ASP in a nutshell, A.Keyton Weissinger, O'REILLY

[22] Inside Microsoft SQL Server 7.0, Ron Soukup, Kalen Delaney, Microsoft press.