

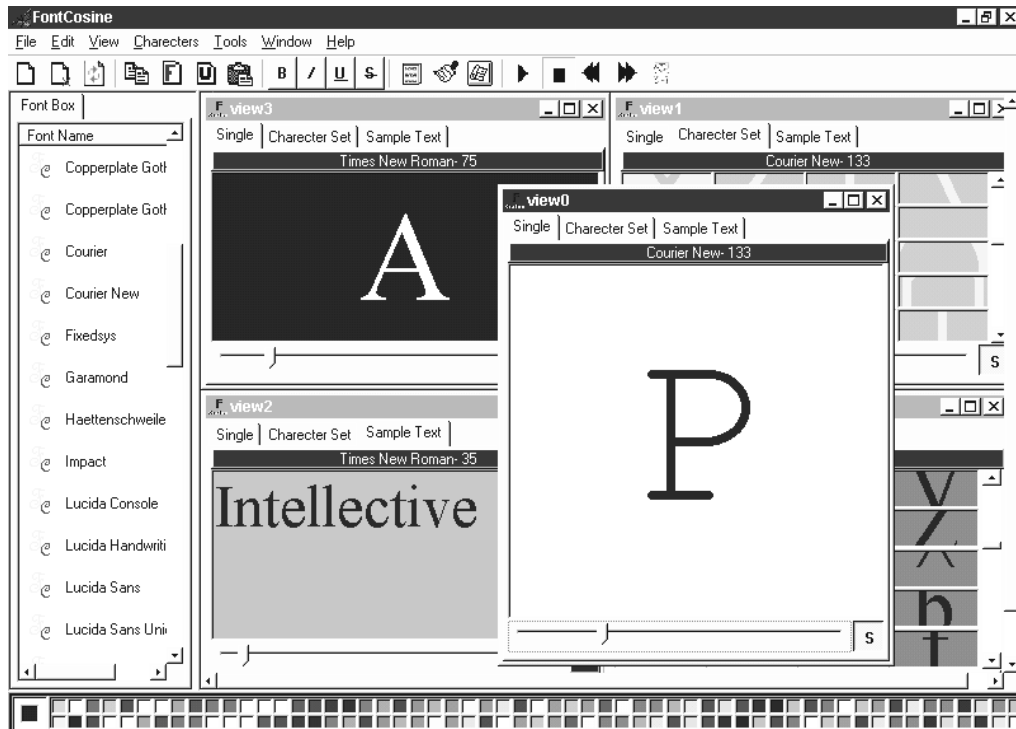
FontCosine 1.0

---

# Project Report

---





Which Font Do You Wanna To See Today?

**Guided By:**

*Mr. Vishal Khasgiwala*  
*Mr. Ashish Jain*  
*Mr. Atul Jain*

**Developed By:**

*Vaibhav Jain,*  
*Student, B.C.A V semester*

**Head Office**

**182, Jaora Compound,  
 Indore.**



**Swati Jain Academy (SJA)**

**Institute Campus**

**33, Sampat Farms,  
 Bicholi Mardana,  
 Indore**



**Swati Jain Institute  
 of  
 Management Studies (SJIMS)**



# CERTIFICATE

*This is to certify that **Vaibhav Jain** an enrollee of Bachelor of Computer Application and a student **Swati Jain Institute of Management Studies**, has worked on the project "**FontCosine 1.0**". He has put sincere effort in the project and has performed tasks related to the project in the Computer Lab of **Swati Jain Institute of Management Studies**. This project may be considered as a partial fulfillment for the examinations conducted by **Devi Ahilya Vishva Vidyalaya, Indore**.*

**Date:**

Mr. Vishal Khasgiwala

External

Mr. Ashish Jain

Mr. Atul Jain:

# Acknowledgments

No man is born complete and I am no exception. When the times were tensed and it seemed like I should put the whole bunch of code into the recycle bin all that could sustain me was the support of friends and elders alike. I was lucky enough to be surrounded by friends and elders who are helpful and supportive. Without their help FontCosine would have probably being released on my 75<sup>th</sup> birthday.

I am also greatly thankful to the member faculty at Swati Jain Institute Of Management Studies as well as Swati Jain Madam who were there when their support and that wonderful sense of humor was badly and eagerly needed.

Special Thanks to Mr. Vishal Khasgiwala who beared with my late project submissions and odd replies during the pre viva. I promise not to be late again.

Thanks also goes Mr. Ashish Jain whose wonderful ideas and inspirations make for some very great challenges for me to face.

Project FontCosine was a one of the most complex and profound undertaking that I have ever taken in my life. For the first I was able to see how complex it is to either develop a truly commercial software and how to launch it as an commercial product. All this has stimulated a new sense of respect for those of my friends who are into marketing and commerce.

And, finally a word of gratitude to my Parents and brother ronak, who were always there with their support and encouragement, even though I'm a little crazy at times.

**Vaibhav Jain**

vaibhav@genesiconvent.com  
57, Shiv Shahkti Nagar,  
Kanadia Road,  
Indore.  
**November 2003.**

# Table of Contents

<b>1. Introduction</b>	<b>1</b>
1.1 What is FontCosine	3
1.2 Product Features & System Requirements	4
1.3 Why FontCosine?	5
1.4 System Design Goals/Design Specifications	6
<b>2. The Project</b>	<b>7</b>
2.1 Software Engineering Paradigm	9
2.2 Software Tools	11
2.3 Visual Basic: An Overview	12
2.4 The Software Process Model	15
2.5 Project Outline.	18
<b>3. The Evaluation Component</b>	<b>21</b>
3.1 Introduction	23
3.2 Architectural Design	24
3.3 Authentication File Format	26
3.4 Class Module Eva	28
3.5 Code Module Validation	29
3.6 Code Module Serials	30
3.7 Code Module FileCrypt	31
3.8 Code Module Ellreg	32
3.9 Code Module Windows	34
3.10 Form Module Form1	35
3.11 Form Module frmReg	36
<b>4. Project FontCosine 1.0</b>	<b>37</b>
4.1 Introduction	39
4.2 Architectural Design	40
4.3 Control Charset	42
4.4 Control Color_Palette	44
4.5 Class Module MyApp	45
4.6 Code Module Startup	46
4.7 Code Module Ellreg	47
4.8 Form Module frmAboute	49
4.9 Form Module frmSplash	50
4.10 Form Module frmSetting	51
4.11 Form Module frmShow	52

**5. The Product Rollout 59**

5.1 Introduction	61
5.2 Licensing	63
5.3 Distribution & Marketing	64
5.4 Product Pricing & Upgrade Policy	67
5.5 Order Processing	68
5.6 Fighting Software Piracy	70
5.7 Technical Support to End Users	71

**6. Miscellaneous 73**

6.1 Future Enhancements: FontCosine 2.0	75
6.2 Bibliography & References	76
6.3 Screen Shots	77
6.4 Project Statistics	79
6.5 Known Bugs/Issues	80
6.6 License Agreement.	81

## INTRODUCTION



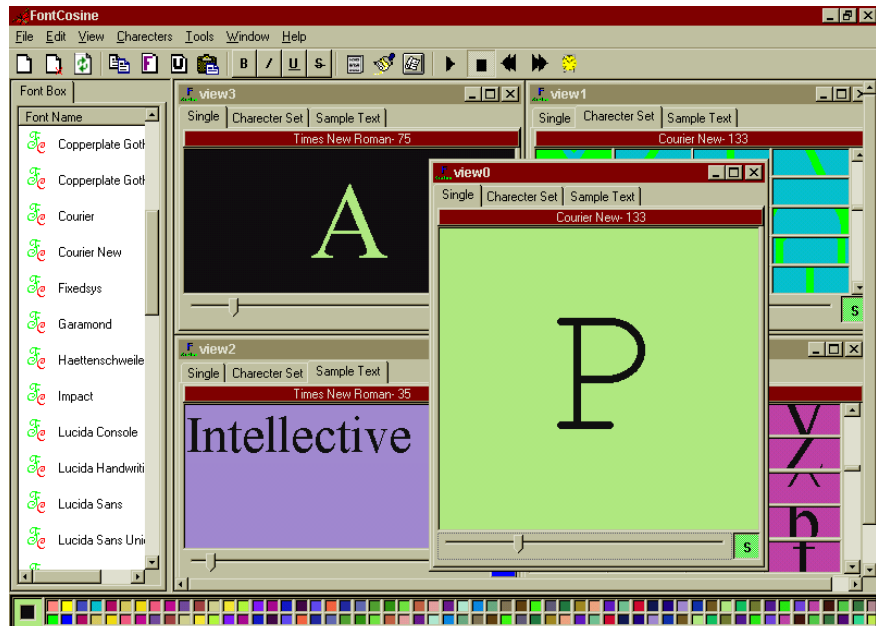
## TOPICS

***What is  
FontCosine?***

***Product  
Features &  
System  
Requirements***

***FontCosine?***

***System Design  
Goals***





## What is Font Cosine?

**F**ontCosine 1.0.12 is a 32-bit font comparing and previewing utility compatible with Microsoft Windows 9x/Me/Xp/2000 Operating Systems. Its aim is to provide the users ability for the first time, compare various fonts available on the system. Using FontCosine users can quickly and effortlessly find the font most appropriate to their document layout. Using FontCosine, users can compare fonts with their glyph styles as well as in various colors with any choosen foreground or background color. The built in font show utility lets the users to automatically iterate to each and every system font quickly and easily.

Revolution in computer technology has enabled P.C users to do a variety of jobs using a computer. Since windows 3.1, PC users have had access to the high quality Adobe and TrueType font engines. Thousands of typefaces are available from both large firms and independent design houses. TrueType fonts can now be found on many commercial online services and throughout the Internet. Traditional font houses such as such as Adobe, Bitstream, Émigré etc continue to release fresh and exciting typefaces.

Unfortunately this wide availability of fonts has created another headache for the designers. Now they have to choose their design font from am overwhelming number of fonts each one having a different look. This usually leads to frustration when a designer is trying to design the layout document and has to hundreds of fonts to choose from. In such a case he would have to spend quit some time going through each and every font on the system to see whether it is suitable for the layout. This same process had to be repeated for each and every element in the layout. Thus it can be easily seen that the designer is wasting a large subset of their time is searching for a suitable font for their document.

This situation becomes worst with application which do not provide users user friendly tools to preview their text before apply a style on it. For example tools like Adobe Photoshop do not provide a tools to assist the user in selecting a font. All that is provided is a combo box containing box filled with entries of all the system font and the users typically selects each item one by one to see the preview of the text.

Thus what really is needed is a general tool to preview fonts. This tool should provide assistance to the user in selecting the font which best matches their design. This is where FontCosine comes to the rescue.

Font Cosine is a tool for previewing these fonts on Windows 9x/Me systems. It works with both TrueType and Adobe Type 1 (ATM) fonts. It's aimed at users who deal in fonts regularly. As system can have a large number of installed fonts & going through all of them and to choose which one fits best to a particular scene can be a very cumbersome processes, especially when the number of fonts installed is very large. Present Font previewing utilities do not provide much features to make this job easier. Aiming this need in mind, we wrote FontCosine; a complete Font Previewing Utility. FontCosine is made with the users in mind and is aimed at making working with font a lot more easier and faster. We are assured that FontCosine will make the job of finding fonts joy and fun instead of a nightmare.



## Product Features & System Requirements

### **Features**

FontCosine includes many features to make working with fonts easier and fun.

#### **Font Viewing & Comparisons**

- ✓ A familiar windows interface to make it easier to learn.
- ✓ A Multiple Document Interface to enable users to compare any number of fonts with each other
- ✓ A very large Color palette helps user to view fonts is any available color changing either its foreground of its background.
- ✓ A view sync which can sync any number of views to each other so that the user can have great flexibility while comparing fonts
- ✓ Font can be either viewed in single character view, Character-set or Sample Text View for greater flexibility.
- ✓ Each view can has its own unique Font, Font-Style, Font-Size, Foreground Color & Background Color.
- ✓ Font Show to view any view or Synced views in all the available system fonts in one after another.
- ✓ “My chars” menu to help you quickly paste to any particular custom character or word to a view.
- ✓ Formatted text of a view can be easily pasted in any word processor that supports Rich Text Format with any loss in format.
- ✓ Innovative features like View-Cloner and View-Painter to make working with views easier.

#### **Extensible Environment**

- ✓ A Custom "Launch Pad" menu to launch you custom application from Fontcosine `s interface itself in a fly.
- ✓ A custom “My Chars” enables the user switch to a custom characters or phrases on a fly.
- ✓ Fast and optimized code ensures you get the best out of you system and Fontcosine.

#### **Great Value**

- ✓ FREE ONLINE UPGRADE to FontCosine 2.0 for all the registered users of FontCosine version 1.0

### **System Requirements**

- ✓ Microsoft Windows9x / Me, Windows 2000/NT and Windows XP.
- ✓ Pentium II 200MHz or Higher.
- ✓ 16 M.B System RAM
- ✓ 10 M.B Free Hard Disk Space.



## Why FontCosine?

Revolution in computer technology has made Desktop Publishing possible. Tools for publishing have reached a level where anyone even an armature can create great looking work with very little effort. These publishing tools which include text-processing tools like Microsoft Word and Image Processing Tools like Adobe Photoshop etc. provide the user all sort of functionality to create marvelous looking designs for page layout.

A truly great page layout design is one which places proper elements of information at their proper place. This elements should be designed such that they stress on what is important and instead on what is not. This makes it mandatory to choose a proper font for the given element. If the selection of font is not proper it is the design that will ultimately suffer.

This constraint of choosing the best possible font for every element in the page layout makes the jobs of designers more like a hell. Any system dedicated to DTP usually has hundreds of fonts installed and the designer/artist has to go through all the fonts one by one to select the best possible match of fonts.

This task is usually made further difficulty due to the fact that many publishing tools aren't well equipped with tools to make this iteration over fonts easier. Thus an artist usually spends hours in selection of fonts instead of concentrating on its design.

One more caveat that is usually faced by many computers users who are just using a text processing tool. Many a times a tool does not provides its user the font style they need in their document. In that case users are forced to choose an alternate style of text. For example the font selection interface of WordPad does not provide way to write rotated text over the screen. If I wanted to incorporate rotated text into my document then I was simply out of luck would had to change my mind.

Thus a tool that can make up for the inadequacy of tool used by the users should exist that simply integrates with the existing tools and provide services that they don't.

- ❖ A tool that assists its users in find the best possible font for their design.
- ❖ A tool to select the best possible color for a given font.
- ❖ A general tool that has a uniform interface such that users can reuse the same tool and interface even when working with many different DTP tools.
- ❖ And a tool that make the dream of comparison possible.

FontCosine was built with above requirements in mind. From the very start we were convinced that in order to solve all the problems being faced by the users one has to provide a general tool to manage, compare and preview fonts available on the system

The name 'FontCosine' derived from the geometric term 'Direction Cosine' which is a set numeric values that provides the sense of direction of a font. In other word they tell us numerically the direction where the vector is headed. FontCosine in the very same way provides the sense of relevance of a given font to a provided design context.

## [todo]System Design Goals

Below are the guiding design goals/ specifications for the development the FontCosine:-

- ✓ *FontCosine must act as general tool for font management.*
- ✓ *It must supplement as well as complement services that are provided by the publishing tools in choosing or viewing a particular font.*
- ✓ *Since iteration through each and every system font is a common activity the interface should provide the user with a tool to iterate over all the fonts quickly.*
- ✓ *Since searching for a given symbol in a given font is also a common activity tools to view the character-set of a font should be present moreover it must be possible to print this character set as a hard copy on paper.*
- ✓ *In order to make font comparison possible the system should provide the users a way to view number of fonts at a same time. This means that the interface needs to be based on Multiple Document Interface.*
- ✓ *The System should expose all the details about the fonts to the users. This also includes various styles options like 'bold', 'italics' etc faces should also viewable to the user.*
- ✓ *Font comparison must also be available in case of variations of font color in different foreground or background color schemes.*
- ✓ *The system will be distributed via Shareware. In this case the system should incorporate a scheme by which the system automatically locks itself after 30 days of installation of the computer. The scheme should be robust enough to able to withstand attacks of sundry users as well as crackers.*
- ✓ *The system must provide the users a way by which they can unlock the product by purchasing a valid end user license from **intellective solutions group**.*
- ✓ *The project must provide all necessary documentation to the end users.*
- ✓ *The project should be well planned and should follow most of the formal methods of Software Engineering. It should be completed with 2-3 month of time.*

## THE PROJECT

todo] illustration of working men

## **TOPICS**

***Software  
Engineering  
Paradigm***

***Software Tools***

***Visual Basic An  
Overview***

***The Software  
Process Model  
Project Outline***





## Software Engineering Paradigm

Software systems come and go through a series of passages that account for their inception, initial development, productive operation, upkeep, and retirement from one generation to another. This section explains the methods and model using which FontCosine 1.0 was developed. It begins an introduction to the explanation of how Software Engineering Principals were applied to the development process and in selection of tools to produce the final product that meets the specification of mentioned in the previous section. The section ends with the final design draft for coding the project.

### Introduction

Software engineering is the computer science discipline concerned with developing large applications. Software engineering covers not only the technical aspects of building software systems, but also management issues, such as directing programming teams, scheduling, and budgeting. The Institute Of Electrical and Electronic Engineers -IEEE describes Software Engineering as:

*“Software engineering. (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1)”*

- IEEE Std 610-1990.

Today any person who is developing software for an individual or an organization must follow the principals of Software Engineering in order to produce software quickly and efficiently. Ignorance or Negligence of these principles mostly results in project failure or software that does not meet the user expectations.

According to the principals of Software Engineering all Software Project should pass through some phases during their development cycle. This pursuance of a phases that act as a milestones to the project development process is known as the Software Life Cycle Model. The classic software life cycle models usually include some version or subset of the following activities:

*System Initiation/Planning:* where do systems come from? In most situations, new feasible systems replace or supplement existing information processing mechanisms whether they were previously automated, manual, or informal.

*Requirement Analysis and Specification:* identifies the problems a new software system is suppose to solve, its operational capabilities, its desired performance characteristics, and the resource infrastructure needed to support system operation and maintenance.

*Functional Specification or Prototyping:* identifies and potentially formalizes the objects of computation, their attributes and relationships, the operations that transform these objects, the constraints that restrict system behavior, and so forth.



*Partition and Selection* (Build vs. Buy vs. Reuse): given requirements and functional specifications, divide the system into manageable pieces that denote logical subsystems, then determine whether new, existing, or reusable software systems correspond to the needed pieces.

*Architectural Design and Configuration Specification*: defines the interconnection and resource interfaces between system subsystems, components, and modules in ways suitable for their detailed design and overall configuration management.

*Detailed Component Design Specification*: defines the procedural methods through which the data resources within the modules of a component are transformed from required inputs into provided outputs.

*Component Implementation and Debugging*: codifies the preceding specifications into operational source code implementations and validates their basic operation.

*Software Integration and Testing*: affirms and sustains the overall integrity of the software system architectural configuration through verifying the consistency and completeness of implemented modules, verifying the resource interfaces and interconnections against their specifications, and validating the performance of the system and subsystems against their requirements.

*Documentation Revision and System Delivery*: packaging and rationalizing recorded system development descriptions into systematic documents and user guides, all in a form suitable for dissemination and system support.

*Deployment and Installation*: providing directions for installing the delivered software into the local computing environment, configuring operating systems parameters and user access privileges, and running diagnostic test cases to assure the viability of basic system operation.

*Training and Use*: providing system users with instructional aids and guidance for understanding the system's capabilities and limits in order to effectively use the system.

*Software Maintenance*: sustaining the useful operation of a system in its host/target environment by providing requested functional enhancements, repairs, performance improvements, and conversions.

All these phases were well taken into account during the development of FontCosine 1.0. Each iteration through the above mentioned phases integrated new set of features with the application.



## Software Tools

FontCosine 1.0 is a 32bit program, which runs on Microsoft Windows compatible operating system. Following tools were used in development of FontCosine.



### **Microsoft Visual Basic**

Microsoft Visual Basic is the fastest and easiest way to create applications for Microsoft Windows®. Whether one is an experienced professional or brand new to Windows programming, Visual Basic provides a complete set of tools to simplify rapid application development. Since most persons of our development team were new to windows programming, Visual Basic provided us excellent tool to enter into the world of windows programming. Please see the next section for a detailed overview of Visual Basic.



### **Microsoft Word 2000**

Microsoft Word 2000 ,a member of Microsoft Office 2000 suit is one of the most popular and comprehensive word processing tool available in the market. It provides almost every thing that every user expects from a word processor. It can both be used for day-to-day work like writing a letter as well as for designing and publishing commercial documents. Microsoft Word 2000 was used to write product documentation files, which were then compiled by the help workshop to produce the final, help files



### **Microsoft Help Workshop**

Help Workshop is a program that is used to create Help (.hlp) files, edit project and contents files, and test and report on help files. Help Workshop takes the information in the project (.hproj) file to combine the topic (.rtf) files, bitmaps, and other sources into one Help file that can be viewed using the Microsoft Windows Help program. The compiled help files can also be linked to from an application to provide context sensitive help for dialog boxes. Entire online documentation of FontCosine 1.0 is compiled using The Help Workshop.



### **Microsoft Paint**

Microsoft Paint is one of the most basic drawing tool which is provided free with Microsoft Windows. It sports an easy to use interface and very small requirements for system resources. It was used to draw various shapes like circles, lines etc as these tools were missing from Adobe Photoshop.



### **Adobe Photoshop**

In world of Desktop Publishing Adobe Photoshop is to image processing what Microsoft Word is for text processing .One of the most comprehensive image manipulation tool till date providing tools right from basic tonal balance of a photograph to sophisticated features to incorporate alpha blending with in the photos. This tool has become an industry standard to judge the other image manipulation tools.



## Visual Basic An Overview

Microsoft Visual Basic, the fastest and easiest way to create applications for Microsoft Windows®. Whether one is an experienced professional or brand new to Windows programming, Visual Basic provides a complete set of tools to simplify rapid application development.

In Visual Basic the "Visual" part refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, you simply drag and drop pre-built objects into place on screen. If you've ever used a drawing program such as Paint, you already have most of the skills necessary to create an effective user interface.

The "Basic" part refers to the BASIC (Beginners All-Purpose Symbolic Instruction Code) language, a language used by more programmers than any other language in the history of computing. Visual Basic has evolved from the original BASIC language and now contains several hundred statements, functions, and keywords, many of which relate directly to the Windows GUI. Beginners can create useful applications by learning just a few of the keywords, yet the power of the language allows professionals to accomplish anything that can be accomplished using any other Windows programming language.

The Visual Basic programming language is not unique only to Visual Basic. The Visual Basic programming system, Applications Edition included in Microsoft Excel, Microsoft Access, and many other Windows applications uses the same language. The Visual Basic programming system, Scripting Edition (VBScript) for Internet programming is a subset of the Visual Basic language. Thus the investment in learning Visual Basic will carry over to these other areas as well. Infact all the tools from Microsoft that support scripting supports Visual Basic code as well.

### ***The Evolution of Visual Basic.***

**Visual Basic 1.0:** Visual Basic for Windows was first released on May 20, 1991 at the Windows World convention in Atlanta Georgia. In September 1992, Microsoft announced Microsoft Visual Basic for MS-DOS in Standard and Professional editions. Like Visual Basic for Windows, this version combined the ease of graphical design with the power and versatility of traditional programming. Developers simply drew the user interface and attached code that responded to events. However, following the release of Windows 3.1 in March 1992 it became apparent that the DOS environment had come to the end of its useful life. The last version of MS-DOS, 6.22, was released in 1994.

**Visual Basic 2.0:** VB 2.0 for Windows (November 1992) was faster, more powerful and easier to use than version 1. VB 2 was also available in a freeware student release called the Primer edition. Visual Basic 3.0 (1993) added tools to access and control databases and Object Linking and Embedding (OLE) version 2. It came in Standard and Professional versions. A superset of VB, called Visual Basic for Applications, was released as part of Microsoft Excel 5 and Microsoft Project 4 in 1993. It has since become the internal

programming language of the Microsoft Office family of products, and is available for license by other software companies.

**Visual Basic 4:** was released in 1995 and supported the new Windows 95 family of 32-bit operating systems. The Professional Edition could also compile code to run on the older 16-bit Windows 3.x systems. Visual Basic Scripting Edition (VBScript) was also announced in 1995. VBScript is used to write embedded code for inclusion in web pages, although not all web browsers will run VBScript.

**Visual Basic 5:** With the introduction of Visual Basic version 5 in early 1997, 16-bit systems were no longer supported. Between versions 4 and 5, significant changes were made in the user interface. Visual Basic 5 added, among other things, the ability to create true executables and to create your own custom controls. It also supported Microsoft's Active-X technology. Visual Basic 5 was available in Standard (Learning), Professional and Enterprise Editions. A free edition, called Control Creation Edition, could be downloaded from [www.microsoft.com](http://www.microsoft.com), and was included with many textbooks. Visual Basic 5 was also included as part of a package known as Visual Studio 97.

**Visual Basic 6:** VB6 was introduced in 1998 and was included as part of a package known as Visual Studio 6.0. VB6 added new capabilities in the areas of data access, Internet features, controls, component creation, language features and wizards. To quote Microsoft's web site, "Visual Basic 6.0 features provide graphical, integrated data access to any ODBC or OLE DB data source, and additional database-design tools for Oracle and Microsoft SQL Server™-based databases. New Web development features bring the easy-to-use, component-based programming model of Visual Basic to the creation of HTML- and Dynamic HTML (DHTML)-based applications." Many organizations are still using this version today.

**Visual Basic .NET:** The newest version of Visual Basic, sometimes referred to as VB7 or Visual Basic .NET, was released in February 2002. This product is a part of Microsoft's .NET software initiative, designed to produce XML-based applications for the Microsoft Internet environment. A 2002 Microsoft Web article said, "At first glance, it may appear to you that Visual Basic .NET is so radically different from what you know that you will have to learn it all over again."

## ***Features Of Visual Basic***

Whether the goal is to create a small utility for an individual himself or for a work group, a large enterprise-wide system, or even distributed applications spanning the globe via the Internet, Visual Basic has the tools a programmer needs.

- ✓ Data access features allow to create databases and front-end applications for most popular database formats, including Microsoft SQL Server, Oracle and other enterprise-level databases.
- ✓ ActiveX™ technologies allow use of functionality provided by other applications, such as Microsoft Word word processor, Microsoft Excel spreadsheet, and other Windows applications. One can even automate applications and objects created using the Professional or Enterprise editions of Visual Basic.
- ✓ Internet capabilities make it easy to provide access to documents and applications across the Internet from within the application.
- ✓ The compiled application is a true .exe file that uses a run-time dynamic-link library (DLL) that can be freely distributed.

## ***Drawbacks of Visual Basic***

Although VB is one of the most popular programming tool currently begin used in the industry , it has some major short comings.

- ✗ Compiled VB code do not show kind of speed and scalability of code produced by C++ or other languages. Moreover all executables produced from VB required Microsoft Visual Basic Virtual Machine which is dll of form MSVBVMXX.dll residing in the system directory of the computer. This dll needs to be delegates all the call from the application to underlying Operating System.
- ✗ Application written in VB cannot be in any way ported to other operating systems or machine architectures unless Microsoft has ported Visual Basic to that environment.
- ✗ Although VB exposes many features of ActiveX/COM through its intuitive interface yet it does not exposes all the features and capabilities of COM. In order to fully utilize the awesome power of COM one has to still head toward the realm of C/C++.
- ✗ Most of the development in visual basic centers around third party components. Although custom controls can be build in VB but they the speed and usability of components created using C/C++. Infact most of the third party components are created using Visual C++.
- ✗ Visual basic code is hard to reuse unless it is wrapped in an ActiveX Component.

## ***Why Visual Basic Is Used For FontCosine 1.0?***

Although VB has some major drawbacks but still it is one of the easiest tool to create windows apps. Since we didn't have any pervious of windows programming, we went for VB it is very easy to learn and can be mastered in a very short time. However since the resulting code was bulky we decided to switch to a more powerful environment of Visual C++ later on. Thus FontCosine 1.0 was built using Visual Basic but FontCosine 2.0 will be build using Visual C++.

## ***Visual Basic Editions***

Visual Basic is available in three versions, each geared to meet a specific set of development requirements.

- The Visual Basic Learning edition allows programmers to easily create powerful applications for Microsoft Windows®. It includes all intrinsic controls, plus grid, tab, and data-bound controls.
- The Professional edition provides computer professionals with a full-featured set of tools for developing solutions for others. It includes all the features of the Learning edition, plus additional ActiveX controls, including Internet controls, and the Crystal Report Writer.
- The Enterprise edition allows professionals to create robust distributed applications in a team setting. It includes all the features of the Professional edition, plus the Automation Manager, Component Manager, database management tools, the Microsoft Visual SourceSafe™ project-oriented version control system, and more.

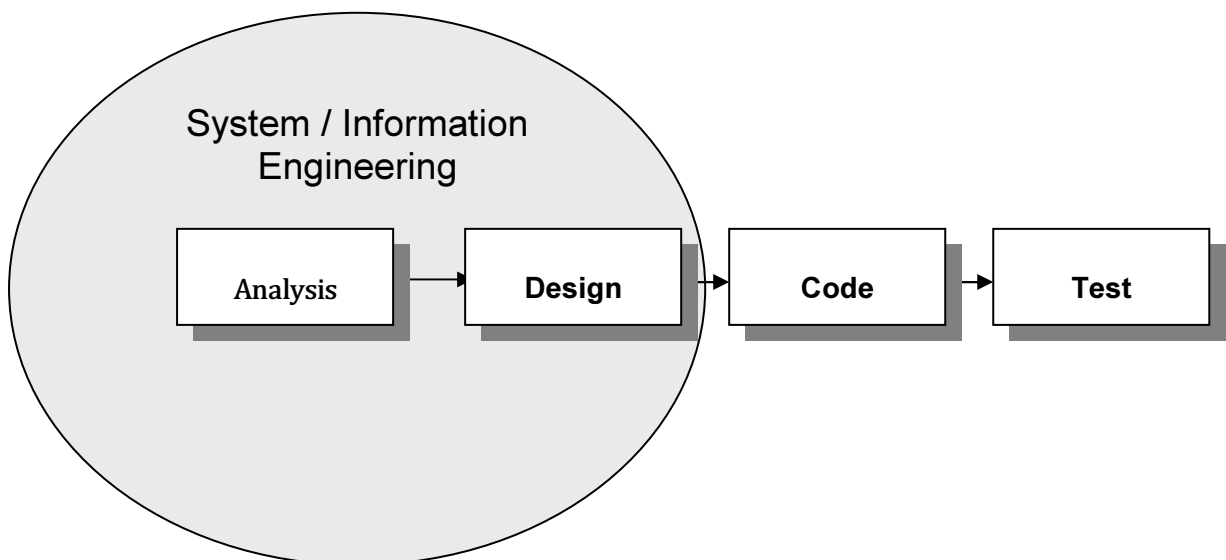


## The Software Process Model

Software process models often represent a networked sequence of activities, objects, transformations, and events that embody strategies for accomplishing software evolution. Such models can be used to develop more precise and formalized descriptions of software life cycle activities. Their power emerges from their utilization of a sufficiently rich notation, syntax, or semantics, often suitable for computational processing. These model encapsulate all the necessary sub processes & tools to produce the final product.

Software process models are networks that can be viewed as representing multiple interconnected task chains. Task chains represent a non-linear sequence of actions that structure and transform available computational objects (resources) into intermediate or finished products. Non-linearity implies that the sequence of actions may be non-deterministic, iterative, accommodate multiple/parallel alternatives, as well as partially ordered to account for incremental progress. Task actions in turn can be viewed a non-linear sequences of primitive actions which denote atomic units of computing work, such as a user's selection of a command or menu entry using a mouse or keyboard.

Traditional Software model is what is known as the waterfall model.

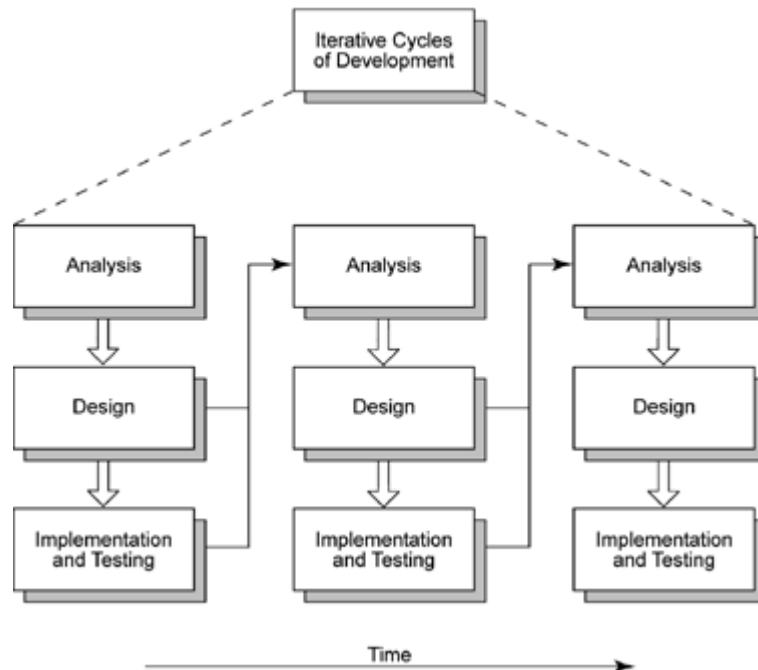


**TRADITIONAL WATER FALL/CLASSIC MODEL**

The classic software life cycle is often represented as a simple prescriptive waterfall software phase model, where software evolution proceeds through an orderly sequence of transitions from one phase to the next in order. Such models resemble finite state machine descriptions of software evolution. However, these models have been perhaps most useful in helping to structure, staff, and manage large software development projects in complex organizational settings, which was one of the primary purposes. Alternatively, these classic models have been widely characterized as both poor descriptive and prescriptive models of how software development "in-the-small" or "in-the-large" can or should occur. Figure above provides a common view of the waterfall model for software development attributed to Mr. Roger S Pressman.

## ***The Incremental Process Model***

The incremental model combines elements of the linear sequential model with the iterative philosophy of prototyping. The incremental model applies linear sequences in orderly fashion as the time progresses. Each sequence delivers a deliverable product.



## ITERATIVE DEVELOPMENT CYCLE APPROACH

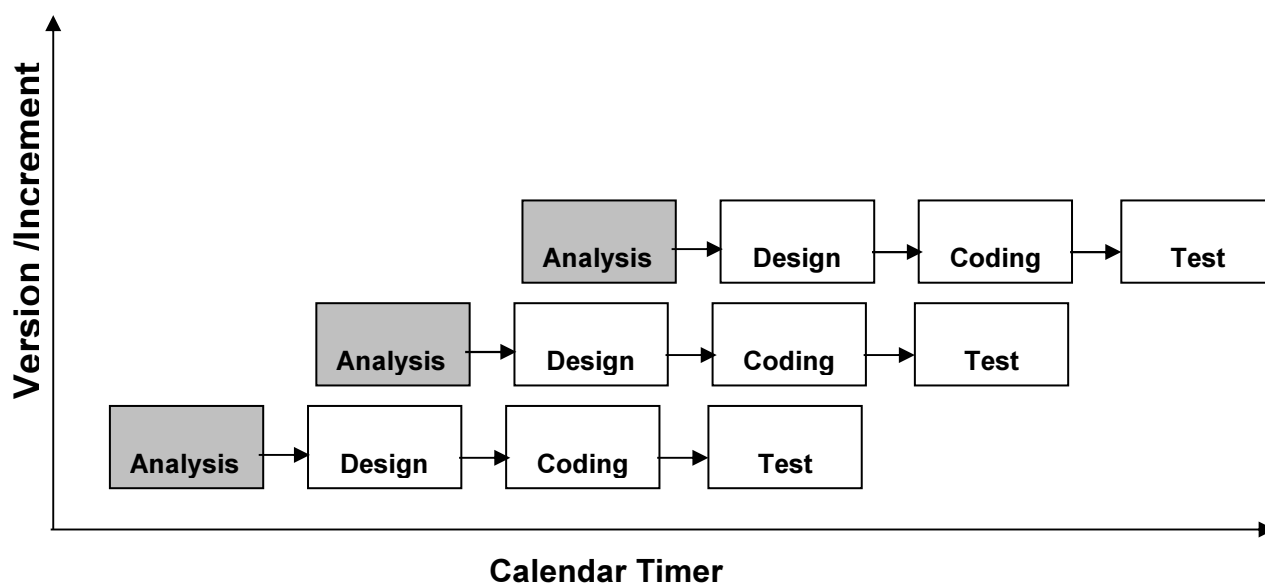
When an Incremental model is used the first increment is often a core product. That is, basic requirements are addressed but many subsequent features (known or unknown) are left undelivered. This core product is delivered and goes under a detailed review after which a plan for the next iteration is sketched out. This plan takes into consideration the changes requested to the previous version by the customer, and features that aren't yet addressed. The whole process is repeated in this way until the final product is delivered. It should be noted that each iteration can also follow the prototyping model.

Incremental model focuses primarily on delivery of a working product as early as possible in the development process. The working product produces after each iteration reorients the development team towards the achievement of the final product. Moreover any changes to the product specification made during the development process can easily be adapted to as compared to other process models. Thus incremental is a perfect model for projects having attributes to being dynamic. Moreover the final product usually achieves Higher Quality Rating as almost all of the customer requirements are addressed in the final release.

A drawback of the Incremental model is that it usually takes longer to develop the final product as compared to other processes. This can lead to serious problems in projects having a very tight time window. Moreover the iterative method makes no sense if the specs of the final product are known and understood thoroughly.

In development of FontCosine this model was adopted, as the development team was new to Visual Basic Environment and the model made it possible for them to create the application as they learn Visual Basic.

### PROCESS MODEL FOR FONTCOSINE



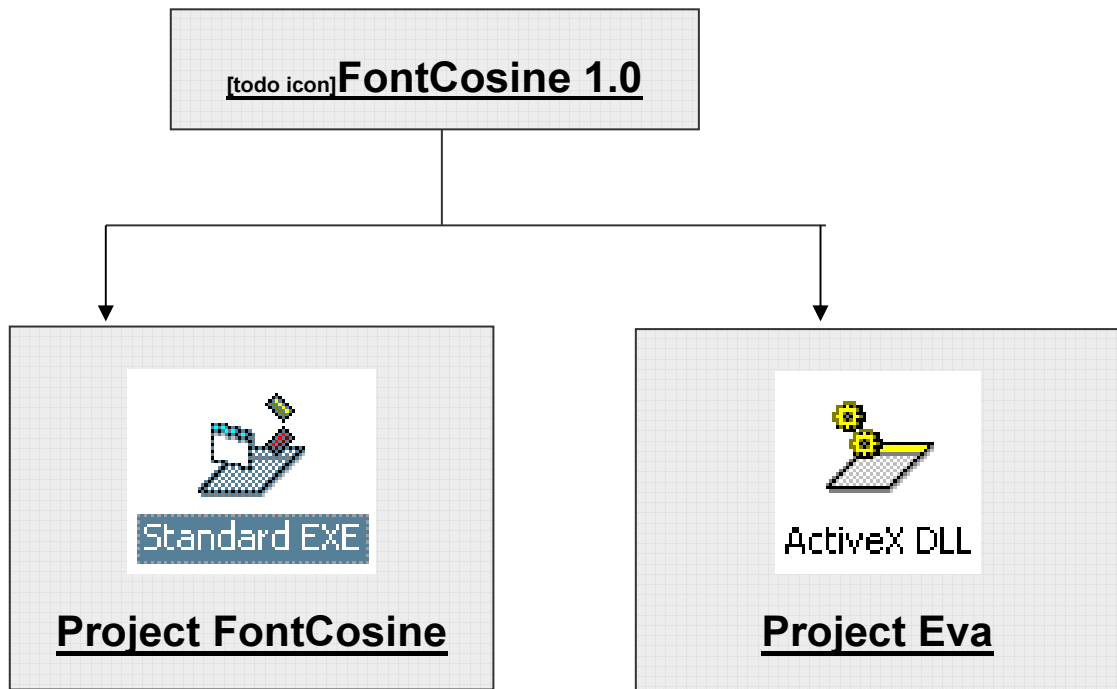
Since there was no time frame set for the project completion we were able freely producing product increments. Also the model made it possible to implements valuable suggestions provided by friends and colleagues. This model made it possible for us to achieve almost 90% of our design goals. However this accuracy in attaining the design goals could not be carried forward to the estimate time taken for the project. The project required 200% more time of what was the estimate.





## Project Outline

After finally deciding the tools, process methods and design goals of FontCosine we arrived at the Design of FontCosine. Here we have distributed the project specs provided in section 1 to two sub projects.



### ***Project Eva***

Project Eva aims at an ActiveX dll component that confirms that the User has a valid serial number to run the application. If he does not has the than provide him access till 30 days past the installation of the program. The component should remind the user each time the application starts about the evaluation period and should provide him with necessary info about the registration The component should also be able to withstand following situations:

- ✓ Change of system date in attempt to fool the software in providing more number of days to evaluate
- ✓ Reinstallation of the component over the existing installation
- ✓ Attacks on component data files by users or crackers
- ✓ Should make it difficult for the user/crackers to bypass the security measures up to maximum possible extent.
- ✓ The component should attain specifications laid for it up to maximum extent.

- ✓ In case the user has a valid license either in form of registration license or Evaluation license it should inform the controlling parent by an event.
- ✓ The component should provide its controlling parent with detailed information about the current license.
- ✓ In all other cases the component should destroy itself and should notify its parent about the same.

## ***Project FontCosine***

Project FontCosine will build an the main application program that should provide the users with an interface that enables him to:

- ✓ Simultaneously compare two or more number of fonts available on the system, each in its own unique size, color and style. Preferably the application should provide an Multi-Document Interface with child windows and each window should have its own font and color setting.
- ✓ Provide a method by which users can iterate through the whole font list quickly and easily.
- ✓ A color grid should be provided through which the user can select both the foreground as well as the background color of then shown text in a given font.
- ✓ Application should provide user with a way through which they can see all the symbols present in the font.
- ✓ Provide the users way to launch their favorite application right from the main interface.
- ✓ Provide a detailed help subsystem to the user.
- ✓ Since the interface is MDI – based the application should provide a way to lock two or more view panes in such a way that any change any one of the locked pane is reflected in all the other locked panes.

## ***Explanations for Dropped Specifications***

Since we are programming in Visual Basic we cannot give much importance to application speed and performance since it's a race that is bound to be lost by us.

Since our knowledge about Visual Basic is limited we have to drop the spec of providing the user about the information about the font file like the author's name etc.

For the very same reason as said above we are dropping the feature of printing the character set of a font on the printer.

=====



## THE EVALUATION COMPONENT

## **TOPICS**

***Introduction***

***Design & Architecture***

***Authentication File  
Format***

***Class Module Eva***

***Code Module  
Validation***

***Code Module Serials***

***Code Module  
FileCrypt***

***Code Module EllReg***

***Code Module  
Windows***

***Form Form1***

***Form FrmReg***





## The Evaluation Component

### — | **Project Objective**

Create an ActiveX dll component that provides confirms that the User has a valid serial number to run the application. If he does not has the than provide him access till 30 days past the installation of the program. The component should remind the user each time the application starts about the evaluation period and should provide him with necessary info about the registration The component should also be able to withstand following situations:

- ✓ Change of system date in attempt to fool the software in providing more number of days to evaluate
- ✓ Reinstallation of the component over the existing installation
- ✓ Attacks on component data files by users or crackers
- ✓ Should make it difficult for the user/crackers to bypass the security measures up to maximum possible extent.
- ✓ The component should attain specifications laid for it up to maximum extent.

[todo text needs rework]

### — | **ANALYSIS**

It is impossible to create security measures that can circumvent all possible attacks. The components should not in any case hinder operations done by a legitimate user. Also the component should have minimum overhead on the application.

To provide maximum protection against casual copying and crackers the security measures should be distributed i.e. the security should not at all depend on the state of a single indicator and should infact depend on multiple indicators. All the indicators should be aligned with one another for application to run. If any one indicator fails than the whole authentication process must fail. The above text can be restated as

"There should be a single point of failure for the whole authentication process"

Another measure against intrusion is incorporation of encryption for protection of authentication data. Major encryption algorithms are copy righted and require licenses, so a custom encryption algorithm should be devised. The Algorithm should be such that encryption should have negligible overhead on the authentication process. Since this encryption will serve as the basis for storing critical registration info on the user computer ; it should not be publicized.



## Design & Architecture

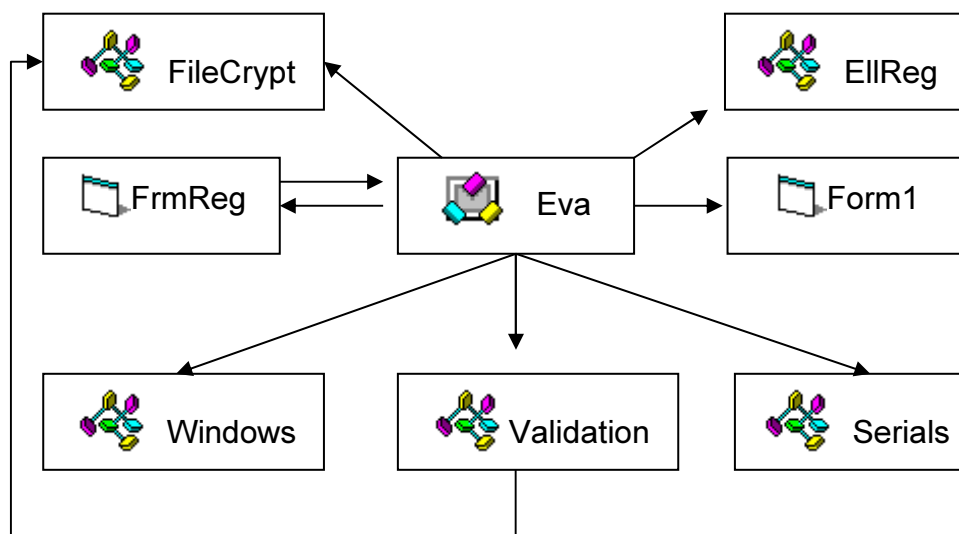
Above specs can be met by using a combination of encryption and windows registry database. Windows registry database is a repository where application can store configuration data. A typical user usually does not access the registry database. Also, the registry is huge. Thus we can easily hide a secret data inside the registry.

The registration should be stored in an encrypted file placed at secret location.

the decryption key of the should be stored in the system registry. Whenever the component needs to retrieve user information from the file it should first retrieve the decryption key from the system registry and should use that key to decrypt this information from the file. The key advantage of this scheme is that the protection mechanism is distributed. i.e. the key resides in the registry and the actual data resides in the file. if either one is attacked and tempered with the other one will automatically fail. Also to make mapping attacks very very difficult the authentication file should be re encrypted each time accessed.

for example if reinstallation of FontCosine is done than existing key in registry is replaced with the default key making the authentication file undecipherable.

The project Eva has the following Architecture. Figure below depicts the various sub components of Eva and their mutual interaction.



[todo : text needs rework dfds ]



## Authentication File Format

The Authentication File is an important file needed for the proper functioning of the Evaluation Component. It stores all the necessary and critical registration information by which the component decides whether the user can use the program or not.

### Record Format

Each record in the authentication is a fixed length record having length of 15 characters. All the information to be stored needs to be serialized to a record. Data requiring more length can use more than one record.

**Record Structure:** {char [15];}

### File Layout

The file layout should be like shown below. It contains some redundant records to make job of cracker difficult. The File should not have fixed format instead it should constantly change positions of critical record in order to prevent mapping attacks. Thus each file should contain a record specifying the starting record number of the critical data. This record also known as STA(Start At) record is to be positioned at record number 8 of the file. All the other record should than be filled with random redundant data. Below is the list of valid records in the file all the other records contain garbage text:

**Record Name: Encryption Header**

**Location: Record no 2**

**Data: text"2SinCos"**

**Explanation:** the second record should be a redundant record in order to check the integrity of the file. if the second record is decrypted to text "2SinCos" *than the decryption key is valid.*

**Record Name: Date Accessed**

**Location: Record no 6**

**Data: date**

**Explanation:** *the second record should contain the date the file was last accessed. this information is stored in order to prevent lapses by system clock changes.*

**Record Name: Time Accessed**

**Location: Record no 4**

**Data: date**

**Explanation:** *the second record should contain the time the file was last accessed. this information is stored in order to prevent lapses by system clock changes.*

**Record Name: StartAt Record**

**Location: Record no 8**

**Data: number**

**Explanation:** *the StartAt Record should contain the starting record number of the user information. this record is used to change the file layout*



**Record Name: Username Record****Location:** STA+0 - STA+1**Data:** char

**Explanation:** *the Username Record should contain the name of the registered user. the name is of maximum length 30 chars thus two record are required to store the data. this data is then used to calculate valid serial number for the user.*

**Record Name: Company Record****Location:** STA+2 - STA+4**Data:** char

**Explanation:** *the Company Record should contain the name of the registered organization. the name is of maximum length 30 chars thus two record are required to store the data.*

**Record Name: Serial Record****Location:** STA+5 - STA+6**Data:** char

**Explanation:** *the Serial Record should contain the registration no. of the user. At run time this data is compared to the code generated by Username record to confirm the validity of evaluation license the name is of maximum length 30 chars thus two record are required to store the data.*

***File Name & Location***

The authentication file is named as '**regt.txi**' and is created in the windows system directory.



## Class Module Eva

'Eva the main class module manages the registration and authentication of the user .It provides is the only public class in the component. It processes the checks the users data to see whether the evaluation period has expired or it has a valid serial number.



**Filename :** *Serials.bas*



### **Events**

Public Event DD() *'Invalid User Signaled*  
 Public Event validate(name As String, company As String, serial As String)  
     *' the user has successfully passed the security*



### **Function & Subroutines**

Private Sub Class\_Initialize() *' Decides whether or not to show the reminder screen to the user by using the authentication data*  
 Sub filereset() *' resets the file to allow the application to continue after the expiration period*  
 Private Sub Command2\_Click() *' event handler for command button to unload the reminder form*  
 Private Sub frreg\_validate(name As String, com As String, code As String)  
     *'event handler for registration form*  
 Private Sub Command1\_Click() *'event handler for command button to unload the reminder form*



### **Data Members**

Private filenum As Long *'stores the handle to the opened authentication file*  
 Private sta As Long *'stores the starting address of the user data*  
 Private key As String *'stores the current Encryption/Decryption key*  
 Private username As String *'stores the user name*  
 Private company As String *'stores the user name*  
 Private serial As String *'stores the user name*  
 Private ins\_dat As Date *'stores the Installation date*  
 unload\_me\_not As Boolean *'whether to unload the registration form or not*  
 do\_not\_mutate\_me As Boolean  
     *'whether to reencrypt the authentication file*  
 WithEvents Timer1 As Timer *'Event dispatcher for the event timer*  
 WithEvents frmmain As Form *'Event dispatcher for the reminder form*  
 WithEvents Command2 As CommandButton  
     *'Event dispatcher for the reminder form command button*  
 WithEvents Command1 As CommandButton  
     *'Event dispatcher for the reminder form command button*  
 WithEvents frreg As Frmreg *'Event dispatcher for the registration form*



## Code Module Validation

Validation code module manages the retrieval and checking of the authentication data from authentication file. It also provides services to check the integrity of file data and the file itself.



**Filename :** *Serials.bas*



### **Function & Subroutines**

```
Public do_not_mutate As Boolean 'do not change the file
Function check_header() As Boolean
    ' Checks the key validity by decrypting the file header for text '2SinCos'
    ' return true on failure and false on success
Function date_check() As Boolean
    'Checks whether the current date is greater than or equal to the date appli
    'cation was opened returns true on success and false on failure
Function file_integrety_check() As Boolean
    'checks the integrety of the athenticatio file by first checking the file
    header'then checking the date to detect any tempering with system clock
Function get_key() As String
    'return the encryption key retrived from the system registry
    'returns the default key if fails to find one
Sub save_key(keyy As String)
    'saves the encryption key to the system registry
Function get_date_last() As Date
    'returns the last date the application was opened 'if an error occurs than
    returns curdate+100
Function get_company() As String
    'retrives the company name from record number startat+2 ,+3
Function get_username() As String
    'retrives the username name from record number startat +0 ,+1
Function get_serialnumber() As String
    'get the product serial(registration number) from record number stat
    ' +4,+5
Function get_insdte() As Date 'get installation date from record number stat+6'return
curdate-100 on failure
Sub mutate() 'mutates(changes) the file as per the key generated key
Function validate_user(username As String, companyname As String, serial As String)
As Boolean 'checks wheather the username company name and serial number match
```



### **Data Members**

```
Public key_changed As Boolean 'whether the key has being changed
Public filenum As Long, 'the handle to the file
Public sta As Long 'the starting record
Public key As String ' encryption key and
```



## Code Module Serials

Serials Code module provides routines to generate Valid Registration numbers out of the provided string. This generated code is then compared to the Registration code provided by the user.



**Filename :** *Serials.bas*



### **Function & Subroutines**

Function get\_serial(name As String) As String

*'generates the serial number from the given string*

Private Function clocker(valu As Currency) As Long

*'normalizes the given number to be less 257 by repeted devision by 10*



### **'Algorithms**

#### **Generation Of Registration Numbers**

*let nm be a string containg name as charecter array*

*let sr be a string containg the generated serial as charecter array*

*let mat be an array[30] of numbers*

1. *resize nm to size 30 by padding or tuncuating*
2. *store the ascii value of each charecter in corrsponding element in mat*
3. *store the last digit of ascii value of last charecter of nn in i*
4. *repeat i times*
  - 4.1 *repeat 30 times*
    - 4.1.1 *multiply each element of mat by 2 and add 1 to it*
    - 4.2 *repeat 29 times*
      - 4.2.1 *add each element to its succedin element except element 1*
      - 4.3 *normalize each charector in mat*
  - 5 *repeat 30 times*
    - 5.1 *if the current element in mat is a valid english alphabet then concate it to sr else concate the numeric value of the element to sr*
6. *resize sr to size 30 by padding or tuncuating*



## Code Module FileCrypt

Manages the encryption & decryption of user data and procedures to store this information inside the file. It contains routines to encrypt, decrypt data record types and to store or retrieve these records to the authentication data file.



**Filename :** *File\_Crypt.bas*



### Data Members

Public Dexter As Boolean

data record type (key As String \* 15, data As String \* 15)



### Function & Subroutines

Function get\_record(filename As Long, rec\_number As Long, Optional key As String) As String *'retrieves the record number from the given file handle and decrypts it either with the provided key or with the default key*

Sub put\_record(filename As Long, data As String, Optional rec\_no As Long, Optional key As String) *'puts the given string data in the authentication file at the specified record number if 0 record number is given then data is appended to the end of the file*

Function genkey() As String *'generates a random encryption key*

Function crypt(data As String, Optional key As String) As record *'encrypts the given data with provided or with a generated key and returns record type*

Function uncrypt(data As record) As String *'uncrypts(decrypts) the given record data with stored encryption key*



### 'Algorithms

#### **Encryption of data**

*let 'data' be data to be encrypted as String*

*let key be the encryption key as String*

*let encdat be the encrypted data*

*1.resize data to 15 by truncating and padding*

*2.Randomize the rand generator*

*3.repeat i from 1 to 15*

*3.1 store the ith character in k*

*3.2 store the ascii value of ith key character in ii*

*3.3 add the numeric value of ii to character value of k*

*3.4 concatenate k to the encdat*

#### **Decryption of data**

*let 'data' be data to be decrypted as String*

*let key be the decryption key as String*

*let decdat be the decrypted data*

*1.repeat i from 1 to 15*

*1.1 store the ith character in k*

*1.2 store the ascii value of ith key character in ii*

*1.3 subtract the numeric value of ii to character value of k*

*1.4 concatenate k to the decdat*



## Code Module EllReg

Manages the interface to the system registry. It contains routines to retrieve, store, delete information from the registry.



**Filename :** *eva\_reg.bas*



### Data Members

```
'Security Mask constants
Public Const READ_CONTROL = &H20000
Public Const SYNCHRONIZE = &H100000
Public Const STANDARD_RIGHTS_ALL = &H1F0000
Public Const STANDARD_RIGHTS_READ = READ_CONTROL
Public Const STANDARD_RIGHTS_WRITE = READ_CONTROL
Public Const KEY_QUERY_VALUE = &H1
Public Const KEY_SET_VALUE = &H2
Public Const KEY_CREATE_SUB_KEY = &H4
Public Const KEY_ENUMERATE_SUB_KEYS = &H8
Public Const KEY_NOTIFY = &H10
Public Const KEY_CREATE_LINK = &H20
Public Const KEY_ALL_ACCESS = ((STANDARD_RIGHTS_ALL Or
KEY_QUERY_VALUE Or
KEY_SET_VALUE Or KEY_CREATE_SUB_KEY Or
KEY_ENUMERATE_SUB_KEYS Or KEY_NOTIFY Or
KEY_CREATE_LINK) And (Not SYNCHRONIZE))
Public Const KEY_READ = ((STANDARD_RIGHTS_READ Or
KEY_QUERY_VALUE Or
KEY_ENUMERATE_SUB_KEYS Or KEY_NOTIFY) And (Not SYNCHRONIZE))
Public Const KEY_EXECUTE = ((KEY_READ) And (Not SYNCHRONIZE))
Public Const KEY_WRITE = ((STANDARD_RIGHTS_WRITE Or KEY_SET_VALUE
Or KEY_CREATE_SUB_KEY) And (Not SYNCHRONIZE))
' Possible registry data types
Public Enum InTypes
ValNull = 0
ValString = 1
ValXString = 2
ValBinary = 3
ValDWord = 4
ValLink = 6
ValMultiString = 7
ValResList = 8
End Enum
' Registry value type definitions
Public Const REG_NONE As Long = 0
Public Const REG_SZ As Long = 1
Public Const REG_EXPAND_SZ As Long = 2
Public Const REG_BINARY As Long = 3
```

```

Public Const REG_DWORD As Long = 4
Public Const REG_LINK As Long = 6
Public Const REG_MULTI_SZ As Long = 7
Public Const REG_RESOURCE_LIST As Long = 8
    ' Registry section definitions
Public Const HKEY_CLASSES_ROOT = &H80000000
Public Const HKEY_CURRENT_USER = &H80000001
Public Const HKEY_LOCAL_MACHINE = &H80000002
Public Const HKEY_USERS = &H80000003
Public Const HKEY_PERFORMANCE_DATA = &H80000004
Public Const HKEY_CURRENT_CONFIG = &H80000005
Public Const HKEY_DYN_DATA = &H80000006
    ' Codes returned by Reg API calls
Private Const ERROR_NONE = 0
Private Const ERROR_BADDB = 1
Private Const ERROR_BADKEY = 2
Private Const ERROR_CANTOPEN = 3
Private Const ERROR_CANTREAD = 4
Private Const ERROR_CANTWRITE = 5
Private Const ERROR_OUTOFMEMORY = 6
Private Const ERROR_INVALID_PARAMETER = 7
Private Const ERROR_ACCESS_DENIED = 8
Private Const ERROR_INVALID_PARAMETERS = 87
Private Const ERROR_NO_MORE_ITEMS = 259

```



## **Function & Subroutines**

```

Public Sub WriteRegistry(ByVal Group As Long, ByVal Section As String,
    ByVal key As String, ByVal ValType As InTypes, ByVal Value As Variant)
    ' This routine allows you to get values from anywhere in the Registry

```

```

Public Sub WriteRegistry(ByVal Group As Long, ByVal Section As String, ByVal key
    As String, ByVal ValType As InTypes, ByVal Value As Variant)
    ' This routine allows you to write values into the entire Registry, it currently only
    handles string and double word values.

```

```

Public Function DeleteValue(ByVal Group As Long, ByVal Section As String, ByVal
    key As String) As String
    ' This routine deletes a specified value from below a specified subkey.
    ' Be very careful using this function.

```

```

Public Function ReadRegistryGetAll(ByVal Group As Long, ByVal Section As String,
    Idx As Long) As Variant
    ' This routine allows you to get all the values from anywhere in the Registry under
    any given subkey, it currently only returns string and double word values.

```

```

Public Function DeleteSubkey(ByVal Group As Long, ByVal Section As String) As
    String
    ' This routine deletes a specified key (and all its subkeys and values if on Win95)
    from the registry.

```



## Code Module Windows

Contains routines to retrieve the windows System directory and windows directory.



**Filename :** Window.bas



### **Function & Subroutines**

Public Function GetWinPath() as string

*'return the path to windows directory through GetWindowsDirectory API*

Public Function GetSystemPath()

*'return the path to system directory through GetSystemDirectory API*

.....

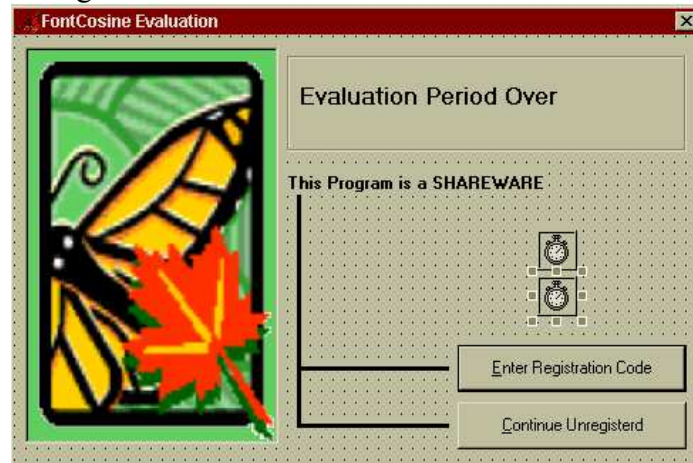




## Form Form1

### Form Form1 (frm\_ev4.frm)

Reminds the user about the evaluation period and optionally allows him to register the product or continue it unregistered.



**Filename :** *frm\_ev4.frm*



### **Member Controls**

Frame Frame1  
 Label Label1  
 CommandButton Command1  
 Timer Timer1  
 CommandButton Command2  
 PictureBox Picture1  
 Line Line1  
 Line Line1  
 Line Line1  
 Label Label3



### **Function & Subroutines**

```
Private Sub Timer2_Timer()  
    'handler for the timer object
```



## Code Module Serials

Provides user with an interface to provide registration to the Eva class and also validates the entered data.



**Filename :** *frmreg.frm*



**Events** validate(name As String, com As String, code As String)



**Data Members & Properties**

Public fila As Long



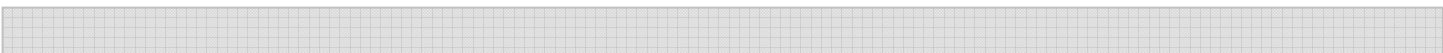
**Members Controls**

Frame Frame1  
 TextBox Text1  
 TextBox Text1  
 TextBox Text1  
 Label Label1  
 Label Label1  
 Label Label1  
 CommandButton Command2  
 CommandButton Command1  
 Label Label2



**Function & Subroutines**

Private Sub Command1\_Click() 'handler for Cancel button  
 Private Sub Command2\_Click() 'handler for Ok button  
 Private Sub filereset() 'Reset the authentication file  
 Private Sub kl() 'Check the user information entered



**PROJECT FONTCOSINE**

## **TOPICS**

*Introduction*

*Design & Architecture*

*Control Charset*

*Control Color\_pallette*

*Class MyApp*

*Code Module Startup*

*Code Module EllReg*

*Form frm\_About*

*Form frmSplash*

*Form frmShow*

*Form mdifrmMain*





## Introduction: Project FontCosine 1.0

[todo todo: change the whole text]

### — | **Project Objective**

Create an ActiveX dll component that provides confirms that the User has a valid serial number to run the application. If he does not has the than provide him access till 30 days past the installation of the program. The component should remind the user each time the application starts about the evaluation period and should provide him with necessary info about the registration The component should also be able to withstand following situations:

- ✓ Change of system date in attempt to fool the software in providing more number of days to evaluate
- ✓ Reinstallation of the component over the existing installation
- ✓ Attacks on component data files by users or crackers
- ✓ Should make it difficult for the user/crackers to bypass the security measures up to maximum possible extent.
- ✓ The component should attain specifications laid for it up to maximum extent.

[todo text needs rework]

### — | **ANALYSIS**

It is impossible to create security measures that can circumvent all possible attacks. The components should not in any case hinder operations done by a legitimate user. Also the component should have minimum overhead on the application.

To provide maximum protection against casual copying and crackers the security measures should be distributed i.e. the security should not at all depend on the state of a single indicator and should infact depend on multiple indicators. All the indicators should be aligned with one another for application to run. If any one indicator fails than the whole authentication process must fail. The above text can be restated as

"There should be a single point of failure for the whole authentication process"

Another measure against intrusion is incorporation of encryption for protection of authentication data. Major encryption algorithms are copy righted and require licenses, so a custom encryption algorithm should be devised. The Algorithm should be such that encryption should have negligible overhead on the authentication process. Since this encryption will serve as the basis for storing critical registration info on the user computer ; it should not be publicized.



## Design & Architecture

[todo todo]

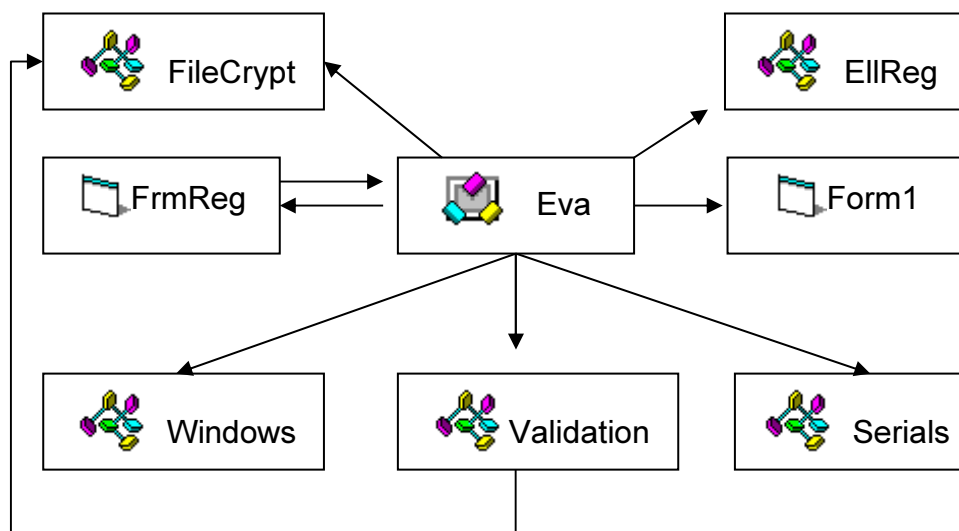
Above specs can be met by using a combination of encryption and windows registry database. Windows registry database is a repository where application can store configuration data. A typical user usually does not access the registry database. Also, the registry is huge. Thus we can easily hide a secret data inside the registry.

The registration should be stored in an encrypted file placed at secret location.

the decryption key of the should be stored in the system registry. Whenever the component needs to retrieve user information from the file it should first retrieve the decryption key from the system registry and should use that key to decrypt this information from the file. The key advantage of this scheme is that the protection mechanism is distributed. i.e. the key resides in the registry and the actual data resides in the file. if either one is attacked and tempered with the other one will automatically fail. Also to make mapping attacks very very difficult the authentication file should be re encrypted each time accessed.

for example if reinstallation of FontCosine is done than existing key in registry is replaced with the default key making the authentication file undecipherable.

The project Eva has the following Architecture. Figure below depicts the various sub components of Eva and their mutual interaction.

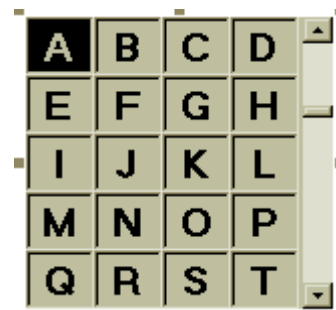


[todo : text needs rework dfds ]



## Control Charset

The Charset control provides the user with a rectangular area. Containing character set from the ASCII character set. A scroll bar too is present to allow user to scroll through all the characters of the character set. The control container can control the font the control uses to draw the characters and also the foreground and background colors of each box in control. The control can also reshape itself to have specified number of rows and columns.



**Filename:** frmreg.frm



### Events

```
Public Event MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    'The mouse button has being pressed
Public Event Dblclick()
    ' the user has Double clicked the mouse
Public Event click()
    ' the user has clicked the mouse
```



### Functions & Subroutines

```
Public Sub refresh()
    ' Redraw all the boxes in the grid.
Private Sub resi()
    ' Resizes the grid creates/unloads objects
Private Sub relo()
    'locate all labels to appropriate locations
Private Sub label1_DblClick(Index As Integer)
    'Event handler for label1 event
Private Sub Label1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
    'Event handler fo label1 event
Private Sub Label1_Click(Index As Integer)
    'Event handler fo label1 event
Private Sub Picture1_Click(Index As Integer)
    'Event handler for Picture1 event
Private Sub Picture1_DblClick(Index As Integer)
    'Event handler for Picture1 event
Private Sub Picture1_KeyPress(Index As Integer, KeyAscii As Integer)
    'Event handler for Picture1 event
```



```

Private Sub Picture1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
    'Event handler for Picture1 event
Private Sub UserControl_Initialize()
    'Control Initilizer/Constructor
Private Sub populate(FR As Integer)
    'populate all labels with captions
Private Sub UserControl_KeyPress(KeyAscii As Integer)
    'handle control keypress event
Private Sub UserControl_resize()
    'handle control resize event
Private Sub VScroll1_Change()
    'handle scroll control change event
Private Sub VScroll1_Scroll()
    'handle scroll control change event
Private Sub aloalo(who As Integer, siza As Long)

```



### **Class Object Properties**

```

Public Property Let forecolor(new_color As OLE_COLOR)
Public Property Let backcolor(new_color As OLE_COLOR)
Public Property Let fontsize(New_Font As Long)
Public Property Let fontname(New_Font As String)
Public Property Let fontitalic(New_Font As Boolean)
Public Property Let fontunderline(New_Font As Boolean)
Public Property Let fontstrikethru(New_Font As Boolean)
Public Property Let fontbold(New_Font As Boolean)
Public Property Get forecolor() As OLE_COLOR
Public Property Get backcolor() As OLE_COLOR
Public Property Get fontsize() As Long
Public Property Get fontname() As String
Public Property Get fontitalic() As Boolean
Public Property Get fontunderline() As Boolean
Public Property Get fontstrikethru() As Boolean
Public Property Get fontbold() As Boolean
Public Property Get text() As String
Public Property Let text(new_text As String)
Public Property Get Font() As Font

```

## Control Color\_pallete



The Color\_pallete control provides the user with a grid of colors in which he can choose a color either by clicking or by right clicking. Clicking changes the foreground color and right click changes the background color. Color elements can be changed by double clicking on an color element and than choosing an appropriate color for the element via a Color Dialog Box. The control has properties like Rows, Columns to get/change the dimensions of the color grid.



**Filename** :color\_pallete.ctl



### Events

Public Event click(what As String)

*'click event with what thing has changed as 'what' parameter*



### Functions & Subroutines

Private Sub label1\_DblClick(Index As Integer)

*'Event Handler for label1*

Private Sub Label1\_MouseDown(Index As Integer, Button As Integer, Shift Private Sub lbl\_fore\_Click(Index As Integer) *'Event Handler for label1*

Private Sub get\_colours() *'load colors from system registry*

Private Sub UserControl\_Initialize() *'Control Initilizer/Constructor*

Sub lo()

*'load controls required into memory*

Private Sub UserControl\_resize()

*'Control resize event handler*

Private Sub aling()

*'relocate the color elements*

Public Sub dimentions(n\_rows As Long, n\_cols As Long)

*'set dimentions of the grid*

Private Sub UserControl\_Terminate()

*'handle control instance termination*



### Class Object Properties

Public Property Let forecolor(new\_color As OLE\_COLOR)

Public Property Get forecolor() As OLE\_COLOR

Public Property Let backcolor(new\_color As OLE\_COLOR)

Public Property Get backcolor() As OLE\_COLOR

Public Property Let Rows(new\_rows As Long)

Public Property Let columns(new\_col As Long)

Public Property Get Rows() As Long

Public Property Get columns() As Long

Public Property Let enabled(new\_val As Boolean)

Public Property Get enabled() As Boolean



Class MyApp is the application class, which does all the initializations necessary to start the application. It first creates an object of Eva class and checks whether the user has a valid legal copy of the application and if the Eva class generates an Validate Event than the class initializes the main window and other controls.



**Filename\_:** myapp.cls



### **Functions & Subroutines**

Private Sub Class\_Initialize()

*'initialize the class and creates an Eva Object*

Private Sub ev\_validate(name As String, company As String, serial As String)

*'the user is valid continue loading the main window*

Private Sub ev\_DD()

*'the user in invalid destroy all the objects and windows.*



## Code Module Startup

Module Startup is startup module of the project. It contains the Sub Main procedure that initializes an MyApp Object. Other sub-routines include one to open a web browser window to a specific URL.



**Filename** : Startup.bas



### **Functions & Subroutines**

Function ShellToBrowser(Frm As Form, ByVal URL\$, ByVal WindowStyle%)

*'Open a Web Browser window to the specified url*

Sub Main()

*'the startup module of the application creates and MyApp Class Object.*



## Code Module Ellreg

Manages the interface to the system registry. It contains routines to retrieve, store, delete information from the registry.



**Filename** : reg2.bas



### Data Members

'Security Mask constants

Public Const READ\_CONTROL = &H20000

Public Const SYNCHRONIZE = &H100000

Public Const STANDARD\_RIGHTS\_ALL = &H1F0000

Public Const STANDARD\_RIGHTS\_READ = READ\_CONTROL

Public Const STANDARD\_RIGHTS\_WRITE = READ\_CONTROL

Public Const KEY\_QUERY\_VALUE = &H1

Public Const KEY\_SET\_VALUE = &H2

Public Const KEY\_CREATE\_SUB\_KEY = &H4

Public Const KEY\_ENUMERATE\_SUB\_KEYS = &H8

Public Const KEY\_NOTIFY = &H10

Public Const KEY\_CREATE\_LINK = &H20

Public Const KEY\_ALL\_ACCESS = ((STANDARD\_RIGHTS\_ALL Or

KEY\_QUERY\_VALUE Or

KEY\_SET\_VALUE Or KEY\_CREATE\_SUB\_KEY Or

KEY\_ENUMERATE\_SUB\_KEYS Or KEY\_NOTIFY Or

KEY\_CREATE\_LINK) And (Not SYNCHRONIZE))

Public Const KEY\_READ = ((STANDARD\_RIGHTS\_READ Or

KEY\_QUERY\_VALUE Or

KEY\_ENUMERATE\_SUB\_KEYS Or KEY\_NOTIFY) And (Not SYNCHRONIZE))

Public Const KEY\_EXECUTE = ((KEY\_READ) And (Not SYNCHRONIZE))

Public Const KEY\_WRITE = ((STANDARD\_RIGHTS\_WRITE Or KEY\_SET\_VALUE

Or KEY\_CREATE\_SUB\_KEY) And (Not SYNCHRONIZE))

*' Possible registry data types*

Public Enum InTypes

ValNull = 0: ValString = 1: ValXString = 2: ValBinary = 3: ValDWord = 4

ValLink = 6: ValMultiString = 7: ValResList = 8

End Enum

*' Registry value type definitions*

Public Const REG\_NONE As Long = 0

Public Const REG\_SZ As Long = 1

Public Const REG\_EXPAND\_SZ As Long = 2

Public Const REG\_BINARY As Long = 3

Public Const REG\_DWORD As Long = 4

Public Const REG\_LINK As Long = 6

Public Const REG\_MULTI\_SZ As Long = 7

```

Public Const REG_RESOURCE_LIST As Long = 8
    ' Registry section definitions
Public Const HKEY_CLASSES_ROOT = &H80000000
Public Const HKEY_CURRENT_USER = &H80000001
Public Const HKEY_LOCAL_MACHINE = &H80000002
Public Const HKEY_USERS = &H80000003
Public Const HKEY_PERFORMANCE_DATA = &H80000004
Public Const HKEY_CURRENT_CONFIG = &H80000005
Public Const HKEY_DYN_DATA = &H80000006
    ' Codes returned by Reg API calls
Private Const ERROR_NONE = 0
Private Const ERROR_BADDB = 1
Private Const ERROR_BADKEY = 2
Private Const ERROR_CANTOPEN = 3
Private Const ERROR_CANTREAD = 4
Private Const ERROR_CANTWRITE = 5
Private Const ERROR_OUTOFMEMORY = 6
Private Const ERROR_INVALID_PARAMETER = 7
Private Const ERROR_ACCESS_DENIED = 8
Private Const ERROR_INVALID_PARAMETERS = 87
Private Const ERROR_NO_MORE_ITEMS = 259

```



### **Function & Subroutines**

```

Public Sub WriteRegistry(ByVal Group As Long, ByVal Section As String, ByVal key
As String, ByVal ValType As InTypes, ByVal Value As Variant)

```

*' This routine allows you to get values from anywhere in the Registry*

```

Public Sub WriteRegistry(ByVal Group As Long, ByVal Section As String, ByVal key
As String, ByVal ValType As InTypes, ByVal Value As Variant)

```

*' This routine allows you to write values into the entire Registry, it currently only handles string and double word values.*

```

Public Function DeleteValue(ByVal Group As Long, ByVal Section As String, ByVal
key As String) As String

```

*' This routine deletes a specified value from below a specified subkey.*

*' Be very careful using this function.*

```

Public Function ReadRegistryGetAll(ByVal Group As Long, ByVal Section As String,
Idx As Long) As Variant

```

*' This routine allows you to get all the values from anywhere in the Registry under any given subkey, it currently only returns string and double word values.*

```

Public Function DeleteSubkey(ByVal Group As Long, ByVal Section As String) As
String

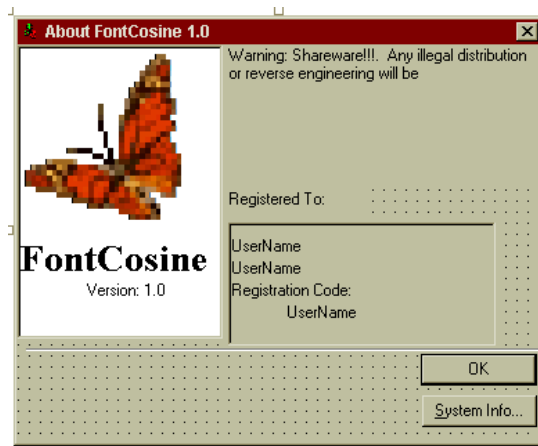
```

*' This routine deletes a specified key (and all its subkeys and values if on Win95) from the registry.*



## Form frmAbout

Form frmAbout displays the about dialog box of the application. It displays the 'END-USER LICENSE AGREEMENT' and the user name to which the application is registered. It can also display system information about the user computer.



**Filename** : frmAbout.ctl



### Functions & Subroutines

```
Private Sub cmdSysInfo_Click()  
    'Handler to the Sysinfo button event  
Private Sub cmdOK_Click()  
    'Handler to the OK button event  
Private Sub Form_Load()  
    'Loads the form and retrives userinfo and registration data  
Public Sub StartSysInfo()  
    'Start Sysinfo32.exe tool
```



## Form FrmSplash

Form frmSplash is the Splash Screen from the Application



**Filename:** frmSplash.frm



### **Member Controls**

PictureBox picture1  
 Label lblPlatform  
 Label lblVersion  
 Label lblCompany  
 Timer Timer1



### **Functions & Subroutines**

Private Sub Form\_KeyPress(KeyAscii As Integer)  
 Private Sub Form\_Load()  
 Private Sub Timer1\_Timer()





## Form frmSerrings

Form frmSettings is used to display a slider bar above the main window that provides control to the speed of FontShow feature.



**Filename : frmSettings.frm**



**Member Controls**

Slider Slider1



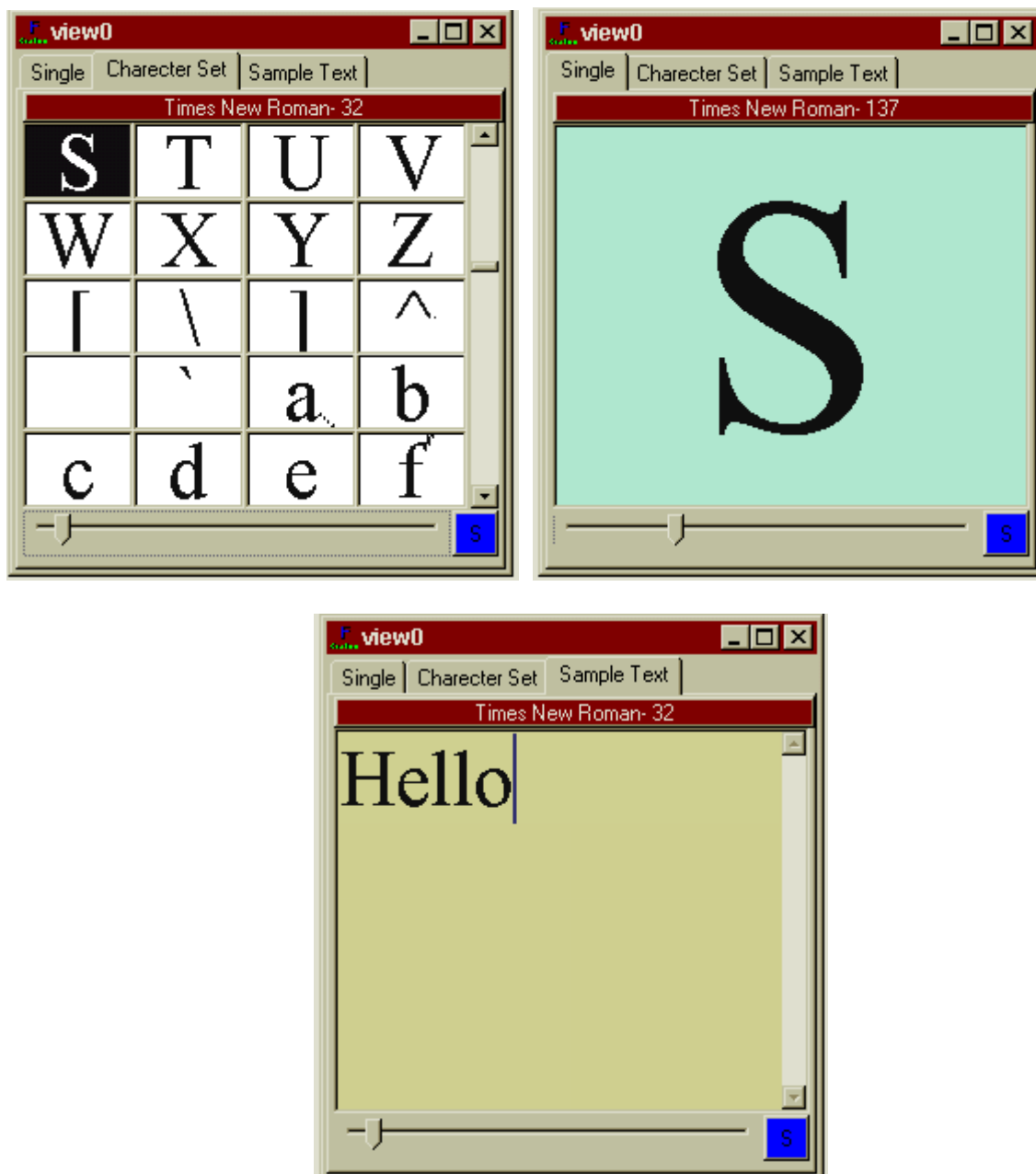
**Functions & Subroutines**

```
Private Sub Form_Load()  
Private Sub Slider1_LostFocus()  
Private Sub Slider1_Scroll()
```



## Form frmShow

Form frmShow implements the view plane in FontCosine 1.0. Each view plane possesses a single font, forecolor & backcolor. The main window can change these properties to reflect changes in the control of the view plane. Each view plane contains a



**Filename** : frmShow.frm



**Member Controls**

- ♦ Label Label1
- ♦ CheckBox lock1
- ♦ PictureBox Picture1
- ♦ RichTextBox RichTextBox1
- ♦ Slider Slider1
- ♦ Tabstrip Tabstrip1
- ♦ TextBox Text1



### **Functions & Subroutines**

```

Private Sub Charset1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
Private Sub Form_Activate()
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
Private Sub Form_KeyPress(KeyAscii As Integer)
Private Sub Label1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Private Sub Label1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Private Sub Charset1_DblClick()
Private Sub Check1_Click()
Private Sub Check1_GotFocus()
Private Sub Form_GotFocus()
Private Sub Form_Initialize()
Private Sub Form_Load()
Private Sub Form_Resize()
Private Sub lock1_Click()
Private Sub lock1_GotFocus()
Private Sub Picture1_GotFocus()
Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
Private Sub Slider1_Change()
Private Sub Slider1_GotFocus()
Private Sub Slider1_Scroll()
Private Sub TabStrip1_Click()
Private Sub TabStrip1_GotFocus()
Private Sub relocate(Optional i As Long = 0)
Public Sub refresh_() 'refresh the form and locked forms
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)

```



### **Data Members & Properties**

```

Private read_only As Boolean
Private con As New Collection

```

```

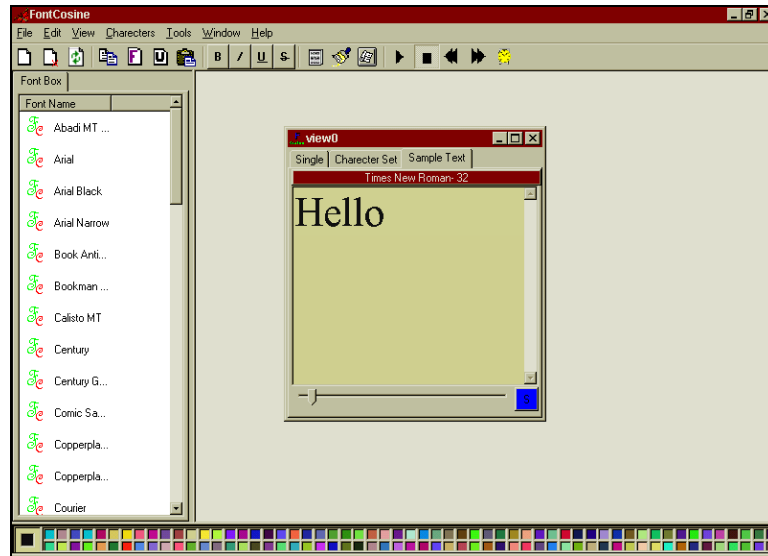
Private texta As String, _
cur_con As Integer
Public my_id As String
Private my_mdi As mdifrmmain
Private font_name As String, font_size As Integer
Private font_bold As Boolean, font_italics As Boolean
Private font_strikethru As Boolean, font_under As Boolean
Private my_fore_col As OLE_COLOR, my_back_col As OLE_COLOR
Dim moving As Boolean, xx As Long, yy As Long
Property Get utext() As String
Property Let readonly(new_va As Boolean)
Property Get readonly() As Boolean
Property Get text() As String
Property Let text(new_text As String)
Property Set mdi(md As mdifrmmain)
Property Get mdi() As mdifrmmain
Public Property Get locked() As Boolean
Public Sub refresh_()
Public Property Let fontsize_(ByVal new_size As Long)
Public Property Let fontbold_(ByVal new_val As Boolean)
Public Property Let fontitalic_(ByVal new_val As Boolean)
Public Property Let fontname_(ByVal New_Font As String)
Public Property Let fontunderline_(ByVal new_val As Boolean)
Public Property Let fontstrikethru_(ByVal New_Font As Boolean)
Public Property Get fontsize_() As Long
Public Property Get fontbold_() As Boolean
Public Property Get fontitalic_() As Boolean
Public Property Get fontname_() As String
Public Property Get fontunderline_() As Boolean
Public Property Get fontstrikethru_() As Boolean
Public Property Let forecolor_(ByVal new_col As OLE_COLOR)
Public Property Let backcolor_(ByVal new_col As OLE_COLOR)
Public Property Get forecolor_() As OLE_COLOR
Public Property Get backcolor_() As OLE_COLOR

```



## Form mdifrmMain

Form mdifrmMain is the application main windows. It spawns all the other form and manages their interaction with each other.



**Filename** : \_mdifrmMain.frm



### Member Controls

```
Menu mnu_fs ,mnuedit, mnuh(0)..muh(7)
  Mnuhelp,mnutool_pre(0),mnutools
  Mnuview,mnuwindow,mychars(0)
  prelaunch(0).. prelaunch(3),reset_view
  sysms(0..10), t_lpad, tw1,tw2,tw3
  v_box(0..2),v_refresh,v_w1, v_w2
  w_(0..3),wwtf_c_(0..5),ch_, e_(1,2,4,5,6)
  f_(0..2),f_exit,f_p(0..7),f_vaot,font_pop
  LaunchPad(0)
PictureBox picture1,picture2
TabStrip TabStrip1
Timer timer1,timer2,timer3
Toolbar t1
checkbox chk_style(0..3)
Color_pallette Color_pallette
Image image1
ImageList imagelist1,imagelist2
ListView ListView1(0)
```



## **Functions & Subroutines**

```

Sub reset(Frm As frmshow)
Sub refresh_view_settings()
Public Function get_my_id() As String
Private Sub refresh_fonts()
Private Sub c__Click(Index As Integer)
Private Sub Chk_style_Click(Index As Integer)
Private Sub Color_pallet1_click(what As String)
Private Sub f__Click(Index As Integer)
Function create_new_view() As frmshow
Private Sub do_check(Index As Integer)
Private Sub f_exit_Click()
Private Sub f_vaot_Click()
Private Sub Image1_Mousemove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Private Sub Image1_Mousedown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Private Sub Image1_MouseUp(Button As Integer, Shift As Integer, X As Single,
Y As Single)
Private Sub Launchpad_Click(Index As Integer)
Private Sub ListView1_DblClick(Index As Integer)
Private Sub ListView1_ItemClick(Index As Integer, ByVal Item As
MSComctlLib.ListItem)
Public Sub refresh_settings()
Private Sub MDIForm_Initialize()
Private Sub MDIForm_Resize()
Private Sub mnu_fs_Click()
Private Sub mnuh_Click(Index As Integer)
Private Sub mnutool_pre_Click(Index As Integer)
Private Sub mychars_Click(Index As Integer)
Private Sub reset_view_Click()
Private Sub syms_Click(Index As Integer)
Private Sub TabStrip1_GotFocus()
Private Sub Timer1_Timer()
Private Sub Timer3_Timer()
Private Sub v_box_Click(Index As Integer)
Private Sub v_refresh_Click()
Private Sub v_aot_Click()
Private Sub e__Click(Index As Integer)
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Private Sub w__Click(Index As Integer)
Private Sub MDIForm_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Private Sub MDIForm_Unload(Cancel As Integer)
Private Sub Timer2_Timer()

```



## **Data Members**

```

Const conHwndTopmost = -1
Const conHwndNoTopmost = -2

```

```
Const conSwpNoActivate = &H10
Const conSwpShowWindow = &H40
Public do_cloning As Boolean, do_copy As Boolean
Dim default_fore_color As OLE_COLOR, default_back_color As OLE_COLOR
Dim default_font_name As String, default_font_size As Long, default_char As String
Dim default_font_strike As Boolean, default_font_under As Boolean
Dim default_font_style As Long
Dim mbmoving As Boolean
Public clu As frmshow
Public class_col As New Collection, locked_clu As New Collection
```

## THE PRODUCT/ ROLLOUT



## **TOPICS**

***Introduction***

***Licensing***

***Marketing &  
Distribution***

***Product Pricing &  
Upgrade Policy***

***Payment processing***

***Fighting Software  
Piracy***

***Technical Support***



## [todo] Introduction

The engineering process described in the previous sections described how FontCosine 1.0 was built, however it does not describe ways that were used to launch FontCosine as a commercial product. As the sole purpose of any business group is to make profit by selling its products.

Here at intellectual we were very much concerned about the commercial success of our product as well as about the state of our group which was still budding and could not in any way bear a financial burden of any form. In other words we were not in a position of taking risks as the stakes were high. Thus in order to minimize investments as well as not compromising the needs of quality management and the expectations of customers.

In order to keep operating costs low we decided to launch our product as Shareware. Shareware is a licensing scheme, which is specifically suited to the small development groups like ours. This scheme is almost exclusively used by the software industry. No other industry can implement Shareware licensing scheme without suffering a huge loss. And for the icing on the cake, Shareware doubles up as a Distribution method. A whole subsection is dedicated to why the Shareware Method of product licensing was used, names as ‘Licensing’.

The first and foremost problem in launching a commercial product is to create a distribution channel through which users can obtain the product. Since our product is a Shareware we need to create two channels for the distribution of our product. One from where users can download our product and other from where the users can purchase a valid end user license in the form of an Registration Code. Since our pockets were not much deep we had to search and weight various options to do our business over the Internet. The chapter ‘Distribution Channels’ provides insight on how we were able to achieve our goal of product distribution that is transparent to both the users as well as us.

The success of a product also depends on how the product is marketed. How we can make the users aware of our product and its features. Also in order to attract them to the possible prospect of purchasing our product we had to make use of some perks. However as with almost all things in this world the problem was money. We were not in any way in a position to hire a super model to advertise our product on the ‘prime slot’ of major daily soaps. Neither we could advertise on the Internet to

One of the major challenges to be faced by us was of Software Piracy. Although ‘Project Eva’ has made our product much secure but still it cannot guard against the arrogant users who give away there Registration Codes to their friends, who in turn distribute them freely to their friends. Thus, in this way a single Registration Code lands up with many users instead of just one. All that amounts to a huge loss in revenue to the pity developer who had spent countless sleepless nights to build such a marvelous product.

Although nothing in this world is completely secure but still we can make things bit tougher for the users to freely distribute the Registration Code. **Mr. Vikas Chauhan** our Management Director had *A Eureka* moment when he found a way users can be discouraged to freely distribute their registration key. His idea is so marvelous that we expect the

shareware computer industry to soon follow the very same method to curb piracy. The chapter ‘Fighting Software piracy’ is dedicated to an idea by the genius.

According to the Classing Software Development Life Cycle Model one the most important phase that all software projects should incorporate is that of maintenance. Software does not wear out like cars or scooters who need periodic maintenance to account for this wear and tear. But software can crash or malfunction when being worked upon. Reasons for such unforeseen problems can be due to the end user or bugs present in the software itself.

Since FontCosine 1.0 has not gone through rigorous testing and checking we were sure that we might have left some bugs in the source code that will only reveal themselves to our customers. Thus we decided to provide technical support to our customers in form of a web site and email support system. The chapter ‘Technical support to end user’ describes the technical support options that were provided with FontCosine 1.0.



## Licensing

FontCosine is distributed as Shareware License. In Shareware License scheme the users are provided with a one time Evaluation License free of cost with each copy of the product. This Evaluation License is valid only for some number of days after which the License expires and the product ceases to function by locking itself. After the Evaluation Period is over the user either has to purchase the permanent License to continue using the product or has to remove the product from his computer.

Copyrighted software available for downloading on a free, limited trial basis; if you decide to use the software, you're expected to register and pay a small fee. By doing this, you become eligible for assistance and updates from the author. Contrast to public domain software which is not copyrighted or to freeware which is copyrighted but requires no usage fee.

Shareware scheme what in lay man's language is known as "Try Before You Buy" provides the users a chance to try software before buying it. If you try a shareware program and continue using it, you are expected to register (purchase) the program. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

In fact, this try-before-you-buy software has been discovered by traditional "shelfware" companies, and now, nearly every large software company provides some type of free trial version of their software. Some of those trial versions are shareware, and some aren't. It's more than a free trial; it's a free trial that you can share with your friends.

Shareware licensing provides an easy alternative to selling the products online. The client evaluated the product and if wants to continue using it pays a registration fees to the author via a web site or other means. In return the author provides the user now a customer with a Registration Code that will unlock the software usually via an email.

FontCosine 1.0 can be evaluated for 30 days after which the users need to get registered with Intellective Solutions by paying a us fee. Upon registration, users will receive a shareware unlock-key and customer code, which will be used to provide Technical Support and Upgrade.

### **Advantages of Shareware**

We at Intellective believe strongly in shareware. It's as good for us as it is for our customers. Shareware is great because:

- We waste no money on packaging, shipping, inventory, middlemen, or shelf space.
- We waste no time on those things either; all our time is spent making our software great.
- We have no unhappy customers. Everyone who purchases our products already thinks they're great -- if they didn't, they wouldn't become our customers!
- Customers get low prices and deals coupled with prompt access to technical support and other resources.



## Distribution & Marketing

*"A product is delivered when it reaches the hands of the customers"*

**-The Economic Times**

*"Marketing is the process of planning and executing the conception, pricing, promotion and distribution of ideas, goods, and services to create exchanges that satisfy individual and organizational objectives."*

**-American Marketing Association**

Above quotes from well-known authors signifies the importance of a good product distribution channel as well as good marketing strategy which play a key role in commercial success of any product. These aspects are most important factors that unfortunately shape the customer's mindset into buying the product. All the resources invested in making a good product are wasted when the customers either does not or cannot buy the product. Thus in order to produce a successful commercial product one needs to have a sound distribution channel supplemented with good product marketing with the goal of attracting the customers into buying the product.

### **Product Distribution**

Traditionally businesses have solely depended on a network of distributors and wholesalers for the distribution of their products. However this method was not feasible for distribution of FontCosine for following reasons:

1. This method of product distribution involves creating a network of dealers, distributors and wholesalers, which needs huge sums of money as well as time. Since we wanted the distribution process to be transparent to both our customers as well as ourselves we had to decide against this method.
2. Moreover this method of distribution involves a huge operating cost. With very little cash in hand this method of product distribution was out of the question.

In search of a more practical method of distribution for FontCosine. We turned our head to the Internet which traditionally has being a the channel of choice to distribute software products. However today the Internet is has become modern frontier of commerce. Businesses are moving to Internet based product distribution to minimize costs and to increase customer satisfaction as Internet provides a very fast as well as secure way to do electronic transaction.

Owing to the coverage of Internet no other distribution channel can even come closer to it in providing a truly global channel for product distribution. Starting a business on the Internet is easy all it takes is a web site dedicated to the specific products. Users can access this web site over the internet and will eventually know about your products and can download/order them.

This scheme of product distribution was perfect for us as it only required one time labor of building a web site from where the users can download the product. Also web space and Domain Name Registrations were beginning to get cheap those days. And as a god gift **Mr. Vikas Chauhan**, our team member already had a personal web space and also had a Domain Named as "Hamaramp.com" registered. Thus all that was needed was to quickly develop a web

site and upload it to the web with the final product. Thus without spending even a single penny we were distributing FontCosine 1.0 on the net.

The very same site then became the home page of intellectual solutions. It provides all the information the user needs right from product documentation to technical support in case of problems.

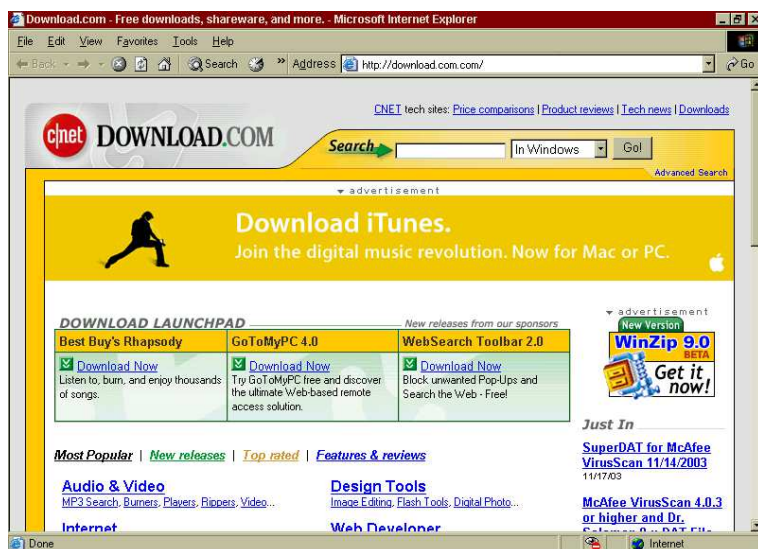
## Product Marketing

In plain and simple terms, marketing activities and strategies result in making products available that satisfy customers while making profits for the companies that offer those products. Marketing produces a "win-win" because:

- ♦ Customers have a product that meets their needs, and
- ♦ Healthy profits are achieved for the company. (These profits allow the company to continue to do business in order to meet the needs of future customers.)

Stated another way: A focus on what the customer wants is essential to successful marketing efforts. This customer-orientation must also be balanced with the company's objective of maintaining a profitable volume of sales in order for the company to continue to do business. Marketing is a creative, ever-changing orchestration of all the activities needed to accomplish both of these objectives.

We had realized this point right from the beginning that all the successful products are also well marketed. Thus marketing of FontCosine was of primary concern to us. Since we were incapable of any large investments into advertising we decided to make use of search engines and download sites available on the internet as the tools to market FontCosine. Also computer magazines that are published were sought after for inclusion of our product in their monthly CD ROMs. Below is a list of various download sites with which the FontCosine was registered.

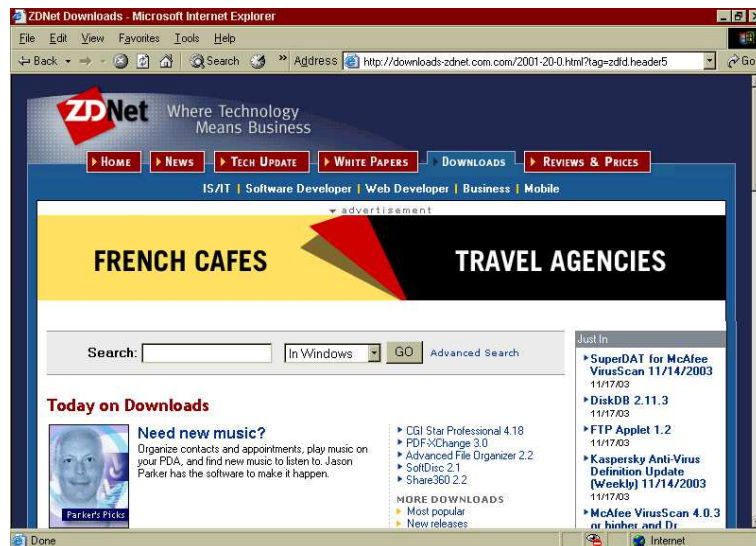


## Download.com

Download.com is one of the most popular download sites on the net. It has a huge collection of software that is neatly categorized into groups for easy access.

## HotDownloads.com

A fast growing download site which is a subsidiary of Regsoft our e-commerce service provider. All product registered with Regsoft are automatically listed on this site.

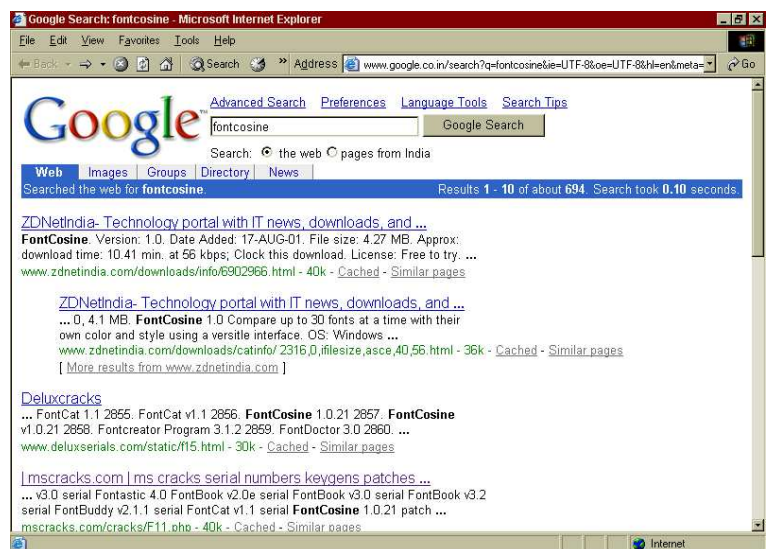


## ZDNet.com

Another popular destination for downloading the latest in software technology. Zdnet.com is also a complete tech news portal, providing information about the latest trend in the computer industry.

## Google.com

The most popular search engine on the web. Provide an easy way for displaying products to the web users. If the product page gets high ranks in the search list than there is a very good chance that the page will be view by the user.







## **[todo] Payment Processing**

### How to Buy FontCosine

For your convenience, we offer several payment options: -

1. Cheques/Demand Drafts/Money Orders: By mail, PAYING WITH DEMAND DRAFTS OR

MONEY ORDER. Please print the order form and mail it with drafts or checks payable to "Intellective Solutions"

#### Mailing Address

Intellective Solutions,

Genesis Convent School,

45 Shiv Shakti Nagar,

Kanadia Road,

Indore (M.P

INDIA

PIN CODE: 452016

2. Online via the Internet - If you have access to the Internet, you can place an order online by logging on to "RegSoft.Com" site. They have a secure payment facility at this site sending in your credit card information complete with your full name, profession, Company's name and its postal address. Once we receive your order, you will be emailed back a Registration Key and customer number.

#### Secure Order URL

<http://www.regsoft.net/purchase.php3?productid=39181>

#### Non-Secure Order URL

[http://www.regsoft.net/purchase\\_nonsecure.php3?productid=39181](http://www.regsoft.net/purchase_nonsecure.php3?productid=39181)

3. RegSoft.com Orderline - (Orders Only) Available 9am - 5pm:

U.S. and Canada: 1-877-REGSOFT (1-877-734-7638)

International: 1-770-319-2718

4. RegSoft.com Fax Line Available 24 hours: 1-770-497-9234



Once we receive your order from RegSoft/Mail you will be delivered your Registration key & Your Customer Number via email. Please allow period of 1 to 3 hours for the delivery. In case you don't get your code within this period, you can contact us at our Home Page.

In today's Internet Economy, accepting credit cards over the internet isn't an option, it is a necessity. It should be the vendor's goal to make it as easy and simple as possible for consumers to purchase a vendor's products. A vendor must choose whether to attempt processing transactions in-house, or whether to utilize a full service e-commerce provider such as RegSoft.com who can provide inclusive services such as toll-free ordering lines, billing support, multiple international credit card accounts, real-time vendor reporting and tracking, etc. The true costs of not having some of these features can often be the difference between success and failure. Below is a more comprehensive list of some additional services an e-commerce solutions provider offers to help vendors save time, increase revenues and keep costs down.

#### **Allows Easy Point of Purchase for Customers**

Vendor Payment Processing Services can provide software customers with the ability to electronically purchase, upgrade, register and download software. RegSoft.com can also help market and promote your product through its website.

#### **Provides Tractability**

Vendor Payment Processing Services can provide reliable, flexible, customizable systems that can allow vendors to better track and measure performance in today's competitive business environment, while customers experience smooth, specific and easy-to-follow purchasing procedures.

#### **Provides Organized, Monthly Statements**

Monthly statements include product name, customer information, tracking numbers, order date and sale price.

#### **Provides Fraud Screening**

Vendors who do not fully understand internet security and fail to properly secure their customers transactions receive a large percentage of chargebacks. Not only do the vendors lose the money from the sale, but many merchants apply "chargeback fees" ranging from \$25-\$75 on a lost sale. However, a benefit of RegSoft.com Vendor Payment Processing Services is that RegSoft.com absorbs all chargeback costs on behalf of its vendor customers.

#### **Provides Transaction Expertise and Convenience**

One of the greatest obstacles of success for many small businesses is that they lack the appropriate resources needed to properly orchestrate a full-time ordering operation. Setting up a secure Internet store, having a staff of operators ready to accept phone orders, and maintaining a dedicated fax line, etc. can often be costly and inefficient for one business alone. RegSoft Vendor Payment Processing Services can help solve this by correctly and effectively maximizing the purchasing opportunities of the vendor by handling the purchase and registration steps, and effectively resell the vendors' product.

### **Provides Avoidance of Dealing with Credit Card and Check Processing Issues**

Vendors are able to avoid dealing with the many difficult credit card and check processing issues such as fraud, denied transactions and refunds. Vendors can also reduce the number of checks cashed to as few as one per month. The time and effort it takes to process checks may occasionally exceed the value of checks, especially with foreign check transactions.

### **Eliminates Currency Conversion and TimeZone Issues**

In our global economy, customers come from all over the world. Vendors and customers from outside the U.S. can benefit from professional Vendor Payment Processing Services that accept international credit cards and offer international currency conversion 24 hours a day, seven days a week. This service eliminates all the legal currency conversion issues for the vendor such as how a customer in France would pay an author or vendor in the United Kingdom. Thus, the vendor/author gains more time to focus on their core competencies.

### **Provides Vendors With More Time to Focus on Core Competencies**

Vendors and authors are able to avoid the problems involved with the registration of user software, thus providing extra time to focus on core competencies, the development of future software products.

### **Provides Customers with Increased Ability to Obtain Products over the Internet**

Consumers can enjoy the security, convenience and speed of using credit cards to instantly download the software or electronic product of their choice. This equates to increased downloads, and thus, **additional revenues** for authors and vendors.

## [todo] Fighting Software Piracy

Software Piracy is the unauthorized duplication, distribution or use of computer software for example, making more copies of software than the license allows, or installing software licensed for one computer onto multiple computers or a server.

Copying software is an act of copyright infringement, and is subject to civil and criminal penalties. It's illegal whether you use pirated software yourself, give it away, or sell it. And aiding piracy by providing unauthorized access to software or to serial numbers used to register software can also be illegal.

In the larger picture, copying cheats the publisher and everyone who uses the software. It makes software more costly and denies the publisher the sales it needs (and earned) to improve software and finance new projects. In 1997, software piracy cost New York State more than \$860 million in lost wages, tax revenue and retail sales, according to a [Microsoft study](#).

Software piracy is a serious threat to all the organization who are into the business of software. Various methods have being proposed to curb piracy from time to time but all the no such method is completely effective in preventing piracy. Now the general attitude is that it is impossible to curb piracy. The failure of Microsoft Windows XP piracy protection scheme is an shining example of how all the efforts against this crime have proved to be futile. The only possible way to reduce piracy is to

## [todo] Product Pricing & Upgrade Policy

### **Prices**

FontCosine is priced taking into consideration our target customers. Also all the sharewares need to be priced low as they do not come with printed manual and the product CD. If the price is low customers will show more willingness in making their Evaluation License into a Permanent License. Product Pricing for FontCosine is as given below.

#### **FontCosine Single Copy: \$15 each**

**10 to 24 computers: \$12 each**

**25 to 49 computers: \$10 each**

**Prices guaranteed through July 2001.**

### ***Our Upgrade Policy***

Intellective Solutions provides minor revisions (1.0a to 1.0b) free of charge to registered users through our Web site or Emails. As we release maintenance versions, customers will be notified via email. Users can then download the new software and disable the "unregistered shareware" notices with your registration key.

Major Revision 2.0 will be provided free to the registered users of Version 1.0 and no upgrade fee will be charged. A new Registration Code will be provided via Email/Postal-Mail, which will disable all shareware notices.

To get the upgrade Users need to be registered with Intellective Solutions. They will need to provide their Registration Code and Customer Name. Only after verifying the information, users will be sent their new Customer ID and Registration Code. Users will be sent their Customer ID and Registration Code only once via Email/Postal Mail. In order to combat piracy, once the mail sent; you will not be sent this information again under any circumstances. Any requests containing the same Customer and Registration information will Not Be Entertained. So please keep your Customer and Registration Secret. Intellective Solutions will Not Be Responsible for the any loses during the delivery of the new Customer ID and the Registration Code.

## **[todo] Technical Support To End Users**

Special Product Support options are available to our Registered Users. You can reach us via email/Our Homepage/Telephone. Please include your Registration Code and name of the person whom the product is licensed to; with the query .You can avail these options through:

1. Email Support: Please mail to [intellective@sify.com](mailto:intellective@sify.com) with "FontCosine Technical Support" in the subject line without quotes.
2. Web Support: You can visit our homepage at [www.hamaramp.com](http://www.hamaramp.com) and can submit your query there.
3. Telephone Support: You can also contact 91-731-562093 to submit your request.

If you are an Unregistered user then you can mail at [intellective@hotmail.com](mailto:intellective@hotmail.com) for Support. Please enter "FontCosine Technical Support" without quotes as the subject text. Registered Users can also mail at this address with "FontCosine Technical Support Option" to ensure quick response to the query.

## MISCELLANEOUS

## **TOPICS**

***Future  
Enhancements:  
FontCosine 2.0***

***Bibliography &  
References***

***Screen Shots***

***Project Statistics***

***Code Listing***

***Known Issues***

***License Agreement***

***Other Products in the  
Pipeline***





## Future Enhancements: FontCosine 2.0

A lot has changed in the arena of computer typography since the release of FontCosine 1.0. When FontCosine 1.0 was developed the typography world considered true type as next generation font standard and Adobe's Type fonts were the industry standard for print work.

But now Adobe and Microsoft have jointly created the open type font standard which finally bridges the valley between computer rendering and print rendering. It has made possible WYSWYP (What You See is What You Print) tool for the commercial printing industry. Open Type fonts are supported in Windows 2000/XP.

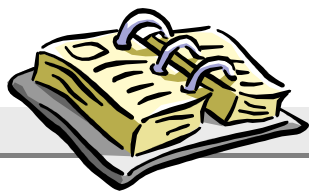
Also Microsoft has developed the Clear Type Font rendering standard. This system produces breath taking readability even on small resolutions screen like that of PDA's and Mobile phones. Clear-Type is currently supported only on Windows XP.

These industry innovations have made it necessary to provide support for then in FontCosine. FontCosine 2.0, which is in the pipeline and is scheduled, to be released in mid-2004 will debut as the next generation tool that will reincarnate itself as a complete Font Management Tool. It will support all the new technologies plus a whole lot of new and nifty feature. Below are some of the main highlights of FontCosine 2.0:

- ❖ *Tools to search for a symbol by free hand drawing sample.*
- ❖ *Complete set of tools to Create and Manage Custom fonts.*
- ❖ *Brand new interface to make working with it even simpler and faster.*
- ❖ *Tools to compress fonts on to the local drive and to install/uninstall them selectively*
- ❖ *Features to provide font preview directly on the client site.*
- ❖ *Tools to provide font embedding support to the web developers.*



- ❖ Will be build in Visual C++ instead of Visual Basic for a high performance code which will make a very small memory footprint



## Bibliography & References

'A long journey to hell with a friends is more precious than a lone shot trip to Heaven' -**Ancient Chinese Saying**

'Books are the best friend a man can get and for girls there are diamonds'  
-**Me**

Above quotes illustrates the importance of good books in lives of man and of Diamonds in lives of Woman!!!. In the long journey of creating FontCosine I too made some very good friends out of some books. Without their help FontCosine could not have come out of the drawing board to our desktops. I would certainly like to express my sense of gratitude to these authors who have produced some of the best text I have ever seen in my current life. Below is just a short summury of that long list:



Microsoft Developers Library – Microsoft



Microsoft Developers Library Online –Microsoft



Visual Basic Books On Line–Microsoft



Mastering Visual Basic



Software Engineering A Practitioners Approach –  
Roger S Pressman



System Analysis and Design – Awad



Visual Basic New Group Hosted at Microsoft



[www.aboute.com](http://www.aboute.com)



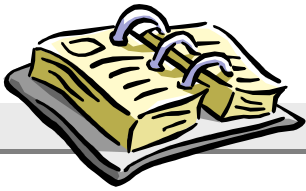
[www.microsoft.com](http://www.microsoft.com)



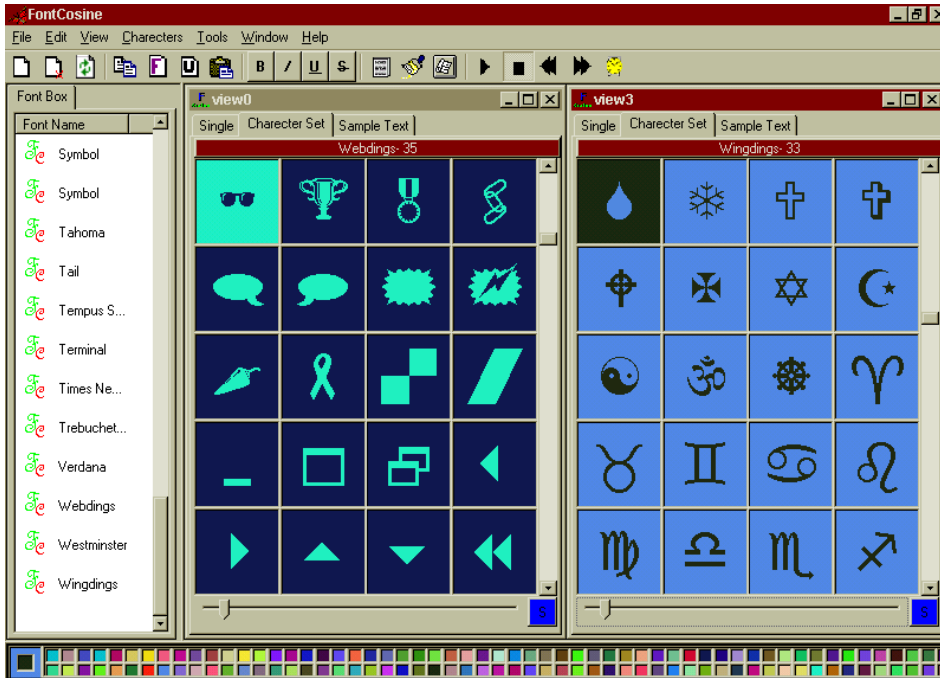
[www.vbworld.com](http://www.vbworld.com)



<http://www.hamaramp.com/>



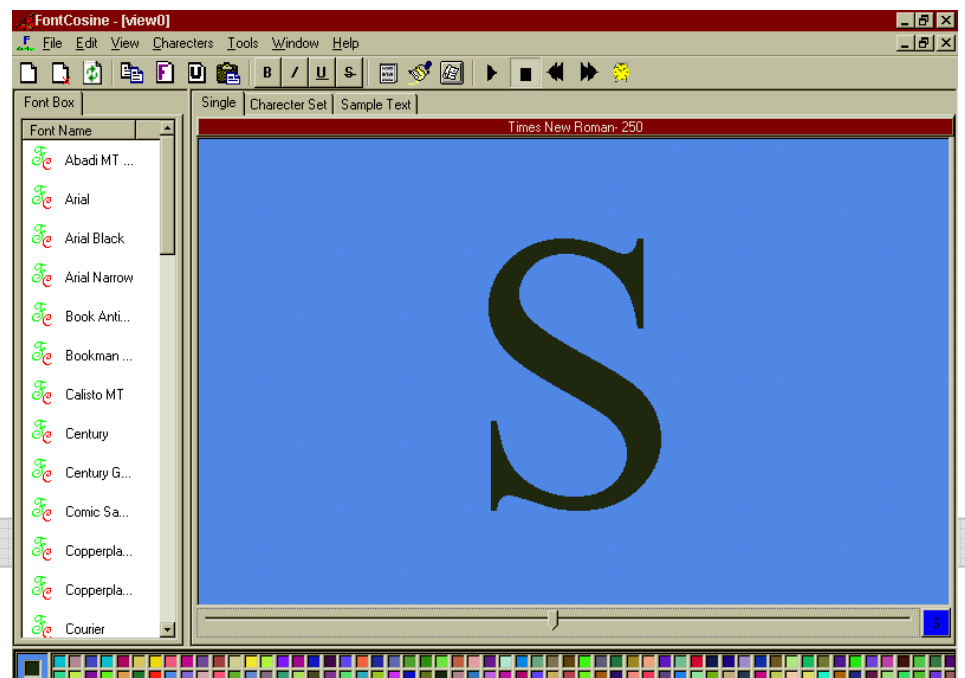
## [todo Icon of an Camera]Screen Shots

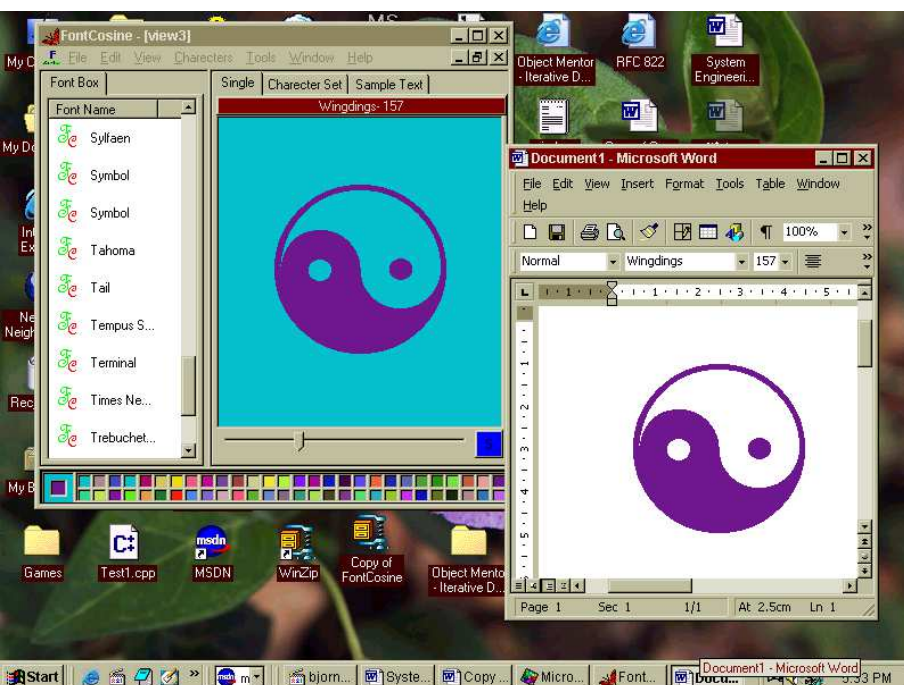


The Primary Interface of FontCosine1.0 which is based on the Multiple Document Interface. The main window is an MDI-Form which hosts the View Plane Windows.

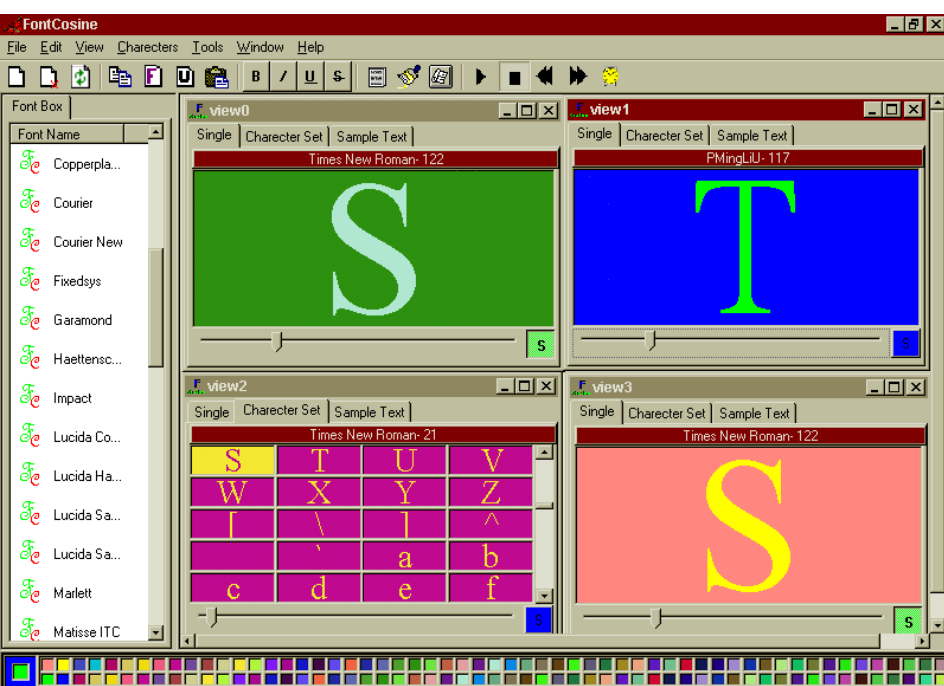
Each View Plane also has a character set viewer to view all the characters available in the font and text box to preview a custom text inside the view plane.

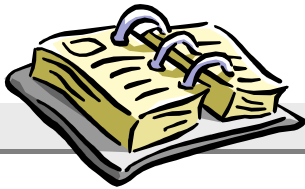
View Planes can be locked to each other to automatically synchronize other view when one of the locked changes. The 'S' Check Box on the bottom right locks the view plane to other planes.





Text copied from FontCosine lands inside any other document as it was seen in the view plane. Here the symbol is copied to the word document as it is.



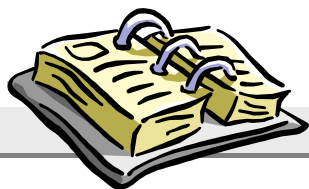


## [todo Icon of an Stats]Project Statistics

Project FontCosine	
Development Tool	Visual Basic
Total Time Taken	2 Months April 2001-June 2001
Total Developers Involved	1
Total Line of Code Produced	5000(approx)
Total Bytes of Documentation Produced	20 M.Bytes

Project Eva	
Development Tool	Microsoft Visual Basic
Total Time Taken	2 Weeks June 2001- July 2001
Total Developers Involved	2
Total Line of Code Produced	1000(approx)
Total Bytes of Documentation Produced	0 M.Bytes

Hamaramp.com	
Development Tool	Microsoft FrontPage 98
Total Time Taken	1 Week July 2001
Total Developers Involved	4
Total Line of Code Produced	1000(approx)
Total Bytes of Documentation Produced	550 K.Bytes + 4.5 M.Bytes



## [todo Icon of an bug]Known Issues

"To err is human , to find a bug, devine" –**Robert Dunn**

No software built is error free. Although every new software that come out are thoroughly tested for any possible issues (read Bugs) , some errors either logical or design error pass through undetected. They are known of only when the product maintenance department and Technical support request an additional telephone line. We too received complains about bugs after the release. Below is a list of known Issues discovered after the release.

- ❖ *Application refuses to start on some Windows 95 Installations.*
- ❖ *The contents option from the help menu does not open the content pane of winhelp instead it directly goes to the very first help page.*
- ❖ *The URL's coded were out of date when the product was released.*
- ❖ *Changing system date sometimes causes evaluation period over reminder to be displayed again.*
- ❖ *Application crashes on windows 2000 when executed from the user privilege account.*
- ❖ *Some times the view planes are not properly resized.*



## License Agreement

### Shareware Terms of Use

#### **General**

As part of our service, Intellective Group. agrees to provide freely downloadable evaluation versions of this software. Intellective Group. agrees to provide this service free of charge. Upon notice published by Intellective Group. Intellective may amend or modify these Terms of Service at any time. You agree to use the service in accordance with the Terms of Service.

#### **Copyright Notices**

No part of the product including the License Agreement may be reproduced, published, transmitted electronically, mechanically or otherwise, transcribed, stored in a retrieval system or translated into any language in any form, by any means, for any purpose other than the purchaser's personal use, without the express written permission of Intellective.

#### **End User License Agreement**

This license agreement is a legal agreement between you the *end user* (either as an individual or an entity) and The Intellective Group herein after referred to as *Intellective* . The software and any of its supporting documentation and associated hardware are herein after referred to as the *product* .

By installing or using the product, you indicate your complete and unconditional acceptance of these terms and conditions. If you do not agree to the terms and conditions of this agreement promptly remove the product from your computer and destroy associated documentation.

Should there be any conflict between the terms and conditions of this agreement and the terms and conditions of any other agreement between you and Intellective or their servants or agents in relation to the product the terms and conditions of this agreement shall apply.

If any provision of this agreement is found to be unlawful or void then that provision shall be severed from this agreement and will not affect the validity of the remaining provisions.

Information contained in the product is subject to change without notice and does not represent a commitment or contracted obligation on the part of Intellective.

#### **Evaluation License Agreement**

This Evaluation License Agreement grants you a non-exclusive License to use the product for an evaluation period of Thirty (30) days from the date you install the product. On the Thirty First (31) day after you installed the product you must either register the product by means of purchasing a commercial License Agreement from Intellective or destroy all copies of the product in your possession, and any related documentation.

You may make a single copy of the *software only* and no other part of the product in a machine readable form for backup use only.

This License Agreement (including the product) is not transferable. You may not lend, loan, lease, rent, sell or distribute the product (or copies) in any form.

### ***Limitations of Use***

You are expressly forbidden from making alterations or modifications to, merge, adapt, de-compile, disassemble, reverse engineer, or attempt to discover the source code without the expressed written permission from Intellective. You are expressly forbidden from using any part of the product in life saving or life critical applications without the expressed written permission from Intellective.

### ***Limitations of Liability***

Under no circumstances, including, but not limited to, negligence, shall Intellective Group. Its subsidiary and parent companies or affiliates be liable for any direct, indirect, incidental, special or consequential damages that result from the use of, or the inability to use, Intellective Group. services. You specifically acknowledge and agree that Intellective Group is not liable for any defamatory, offensive or illegal conduct of any user of the free forums created using the Intellective Group. service. If you are dissatisfied with any remotely hosted forum material, or with any of Intellective Group. terms and conditions, your sole and exclusive remedy is to discontinue using Intellective Group. services. In addition, you release Intellective Group and its affiliates from any damages that you incur, and agree not to assert any claims against them, arising from your use of its products or services.

### ***Privacy***

Unless the user informs us otherwise, Intellective Group. reserves the right to use and to disclose to third party vendors user information (e.g. user name and e-mail address) for purposes such as providing users with information about products and services.

### ***Arbitration***

Intellective Group. is operating in India at 45, Shiv Shakti Nagar, Kanadia Road, Indore –INDIA. If arbitration should occur, the court can only be governed by India Constitution.

### ***Disclaimer***

Intellective Group. has provided links and pointers to Internet sites maintained by third parties. Neither Intellective Group., its parent or subsidiary companies nor their affiliates operate or control in any respect any information, products or services on these third-party sites. The materials in this site and the third-party sites are provided "as is" and without warranties of any kind either express or implied. To the fullest extent permissible pursuant to applicable law, disclaims all warranties, express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. Intellective Group. does not warrant that the functions contained in the materials will be uninterrupted or error-free, that defects will be corrected, or that this site, including bulletin boards, or the server that makes it available, are free of viruses or other harmful components. Intellective Group. does not warrant or make any representations regarding the use or the results of the use of the materials in this site or in third-party sites in terms of their correctness, accuracy, timeliness, reliability or otherwise. You (and not Intellective Group.) assume the entire cost of all necessary maintenance, repair or correction.

\*\*\*\*\*



