



**THE INSTITUTION OF ELECTRONICS AND  
TELECOMMUNICATION ENGINEERS  
PUNE CENTRE**

**PROJECT REPORT ON  
  
ANTIVIRUS SOFTWARE**

**SUBMITTED BY  
Mr. SIDDHARTH N DEORE**



## COMPLETION CERTIFICATE

This is to certify that **Mr. Siddharth N Deore** of **DIPIETE** in **CS** student Membership Number – **186298L** has satisfactorily completed the project work as laid down by the IETE.

Hon. Secretary / Chairman  
IETE Pune

Examiner 1.

Examiner 2.

# **Abstract**

## Abstract

A large portion of modern computing life is to secure the information that we are creating and processing. There are many aspects of information security, ranging from physical access to ensuring that the information has not been changed in any way. One of the most high-profile threats to information integrity is the computer virus. Surprisingly, PC viruses have been around for two-thirds of the PC's lifetime, appearing in 1986. With global computing on the rise, computer viruses have had more visibility in the past two years.

The ideal solution to the threat of viruses is prevention. Do not allow a virus to get into the system in the first place. This goal is in general difficult to achieve, although prevention can reduce the number of successful viral attacks. The next best approach is to be able to do the following.

The *Kinshuk Antivirus* is purely written in *Microsoft Visual Basic 6.0* including manual scanner and real-time protection with more than 66000+ virus definitions with MD2 encrypted definitions stored in Microsoft Access database downloaded on first use. This removes and also quarantines the infected files and sounds alert after virus detection. This software includes some Win32 API (Application Program Interface) functions and some third party module to improve its functionality.

# Acknowledgement

## Acknowledgements

Completing a task is never a one man's effort. It is often the result of invaluable contribution of number of individuals in direct and indirect manner. Sincerity being here when after the completing of any software project, I find the most difficult part of the entire exercise to record my gratefulness to various people who help on this venture. The words that follow are merely the formal expression of our deep sense of indebtedness that is better treasured in thoughts than said in words.

This motivating factor behind my project are my most esteemed guide **Mr.Sadik Ali Sayyad**, working at **Cognizant Technology Solutions India Limited**, for his initial and continued encouragement, constant support, inspired help and motivation guided me, in spite of being busy round the clock. I have proud privilege to accomplish this entire work under his scholarly guidance.

I would like to express many thanks to all teaching and nonteaching staff at **IETE Pune Center**, who helps me on time to time in completion of this project. I would specially thank to my friends **Sanket Vispute** for his help and suggestions.

- ***Siddharth N. Deore***

# Content

## CONTENTS

Chapter 1	Introduction
	a) Need For Antivirus
	i) What is Virus
	ii) History of Virus
	iii) Type of Virus
	iv) Who creates A Virus
	v) How to protect from Virus
	b) What is Antivirus
	i) History
	ii) Identification methods
	1 Signature Based detection
	2 Heuristic detection(suspicious Behavior Approach)
	3 Root kit detection
Chapter 2	System Design
	a) Architecture
	b) Block Diagram
	c) Flow Chart
Chapter 3	Development Techniques
	a) Standard EXE(Manual Scanner)
	b) ActiveX DLL(Real time Scanner)
Chapter 4	Implementation
	a) Main Antivirus Executable
	b) Real-time Scanner (DLL)
	c) Third Party Modules
Chapter 5	Win 32 API Calls
Chapter 6	System Requirement
Chapter 7	Testing
Chapter 8	Tray Icon
Chapter 9	Cost Estimation
Chapter 10	Conclusion and Future Scope
Chapter 11	References and Bibliography



# Introduction

## 1. Introduction

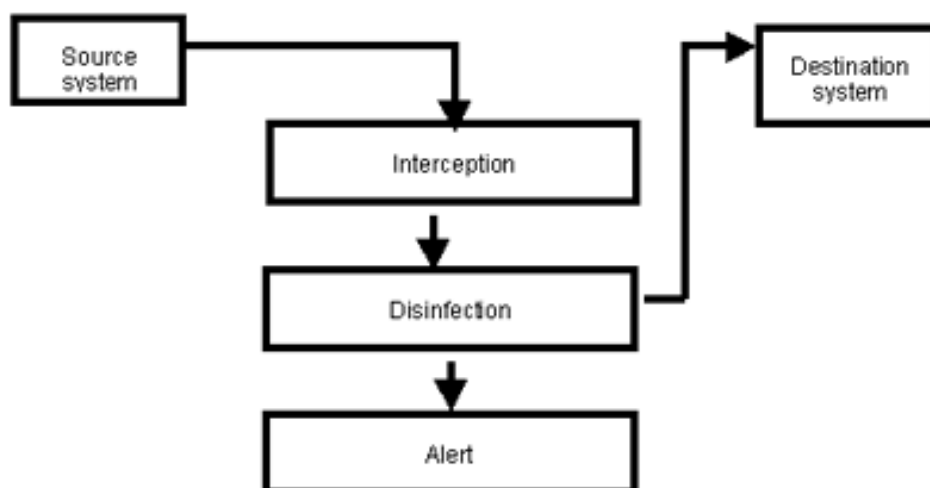
### How an Antivirus Program Works

From the early viruses, created as experiments in the eighties, to the latest malicious code, one of the biggest worries for all computer users is the threat of viruses entering their systems.

To prevent viruses from entering a system there are basically just two options. The first of these is to place the computer in a protective 'bubble'. This in practice means isolating the machine; disconnecting it from the Internet or any other network, not using any floppy disks, CD-ROMs or any other removable disks. This way you can be sure that no virus will get into your computer. You can also be sure that no information will enter the computer, unless it is typed in through the keyboard. So you may have a fantastic computer, the perfect data processing machine...but with no data to process. If you're happy with that, your computer will be about as much use as a microwave oven.

The second option is to install an antivirus program. These are designed to give you the peace of mind that no malicious code can enter your PC. But how do they do it? How does the program let you install a game, but prevent a virus from copying itself to disk? Well, this is how it works....

An antivirus program is no more than a system for analyzing information and then, if it finds that something is infected, it disinfects it. The information is analyzed (or scanned) in different ways depending on where it comes from. An antivirus will operate differently when monitoring floppy disk operations than when monitoring e-mail traffic or movements over a LAN. The principal is the same but there are subtle differences.



The information is in the 'Source system' and must reach the 'Destination system'. The source system could be a floppy disk and the destination system could be the hard disk of a computer,

or the origin an ISP in which a message is stored and the destination, the Windows communication system in the client machine, Winsock.

The information interpretation system varies depending on whether it is implemented in operating systems, in applications or whether special mechanisms are needed.

The interpretation mechanism must be specific to each operating system or component in which the antivirus is going to be implemented. For example, in Windows 9x, a virtual driver VxD is used, which continually monitors disk activity. In this way, every time the information on a disk or floppy disk is accessed, the antivirus will intercept the read and write calls to the disk, and scan the information to be read or saved. This operation is performed through a driver in kernel mode in Windows NT/2000/XP or an NLM which intercepts disk activity in Novell.

Antivirus products that are not specially designed for operating systems, but are implemented over other applications, have a different interpretation mechanism. For example, in an antivirus for CVP Firewalls, it is the firewall that provides the antivirus with information in order to scan it through the CVP protocol and in the antivirus for SendMail, the MilterAPI filter facilitates information interpretation.

Sometimes an interpretation mechanism is not provided by the antivirus (such as a VxD) or the application (such as the CVP). In this case, special mechanisms between the application and the antivirus must be used. In other words, resources that intercept information and pass it to the antivirus, offering complete integration in order to disinfect viruses.

Once the information has been scanned, using either method, if a threat has been detected, two operations are performed:

1. The cleaned information is returned to the interpretation mechanism, which in turn will return it to the system so that it can continue towards its final destination. This means that if an e-mail message was being received, the message will be let through to the mailbox, or if a file was being copied, the copy process will be allowed to finish.
2. A warning is sent to the user interface. This user interface can vary greatly. In an antivirus for workstations, a message can be displayed on screen, but in server solutions the alert could be sent as an e-mail message, an internal network message, and an entry in an activity report or as some kind of message to the antivirus management tool.

As you can see, antivirus programs do not perform miracles, nor is it a software tool that you need to be wary of. It is a very simple security ally that offers precision and advanced technology. Consider this; when you copy a few mega bytes to the hard disk of your computer, the antivirus must look for over 65,000 viruses without affecting the normal functioning of the computer and without the user realizing.

Antivirus programs offer a high level of protection and prevent any nasty surprises. It is as simple as putting XXX dollars in a box to get peace of mind. I'm sure that now you don't have any serious doubts...

## **Scan Engines**

Regardless of how the information to be scanned is obtained, the most important function of the antivirus now comes into play: the virus scan engine. This engine scans the information it has intercepted for viruses, and if viruses are detected, it disinfects them.

The information can be scanned in two ways. One method involves comparing the information received with a virus database (known as 'virus signatures'). If the information matches any of the virus signatures, the antivirus concludes that the file is infected by a virus.

The other way of finding out if the information being scanned is dangerous, without knowing if it actually contains a virus or not, is the method known as 'heuristic scanning'. This method involves analyzing how the information acts and comparing it with a list of dangerous activity patterns.

For example, if a file that can format a hard disk is detected, the antivirus will warn the user. Although it may be a new formatting system that the user is installing on the computer rather than a virus; the action is dangerous. Once the antivirus has sounded the alarm, it is up to the user whether the danger should be eliminated or not.

Both of these methods have their pros and cons. If only the virus signatures system is used, it is important to update it at least once a day. When you bear in mind that more than 15 new viruses are discovered everyday, an antivirus that is left for two or three days without being updated is a serious danger.

The heuristic system has the drawback that it can warn you about items that you know are not viruses. If you have to work with a lot of items that may be considered dangerous, you could soon tire of the alerts. Programmers in particular may prefer to disable this option.

### **Permanent and on demand scans**

When describing antivirus programs, it is important to clearly distinguish between the two types of protection on offer. The first is permanent scans, which are more complex and essential. These scans constantly monitor the operations performed on the computer to prevent any kind of intrusion.

The other type of protection available is on demand scans. These use the same scan engine as the permanent protection and check any parts of the system whenever the user wants. These are normally used under special circumstances. For example, a user may want to perform an on demand scan when using a new floppy disk or to check information stored on the computer that hasn't been used for a while.

# System Design

## 2. System Design

### Structural Architecture Design of Antivirus

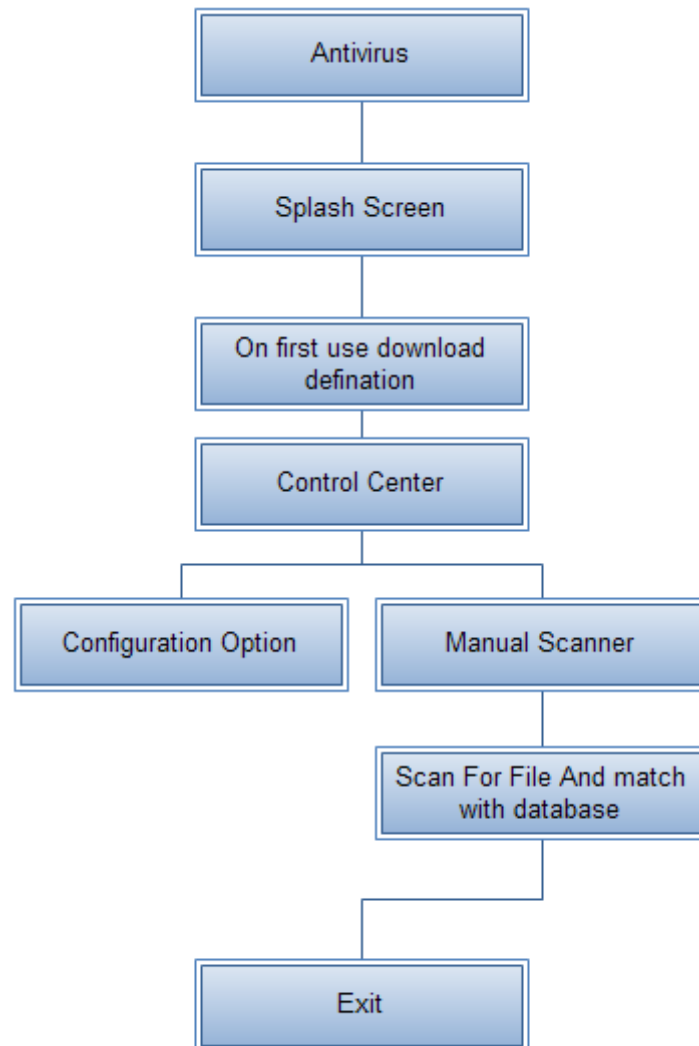


fig. Archtecture Diagram

## Architectural Block Diagram

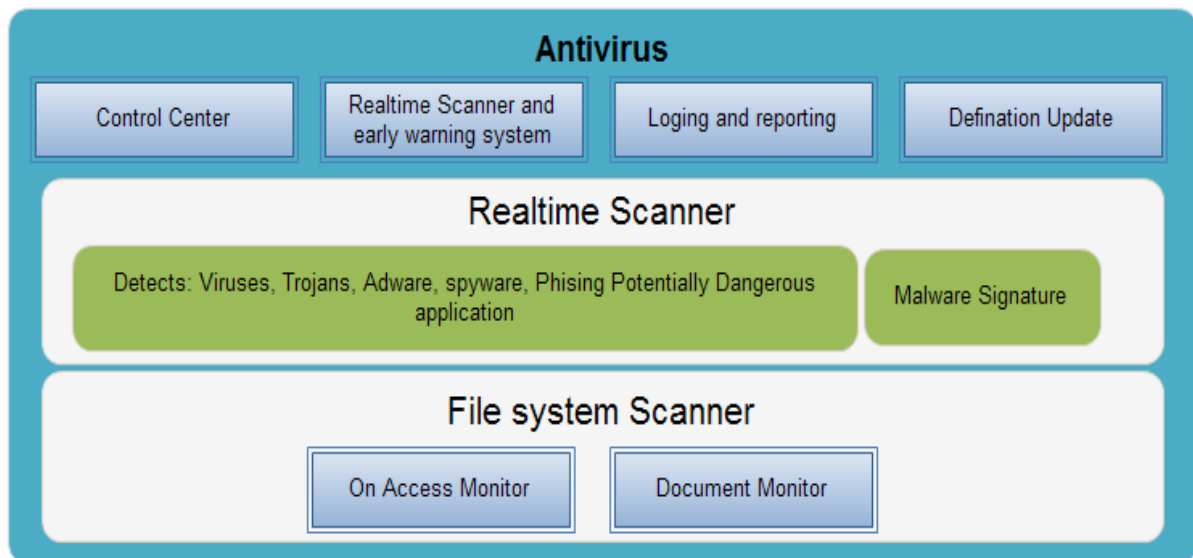
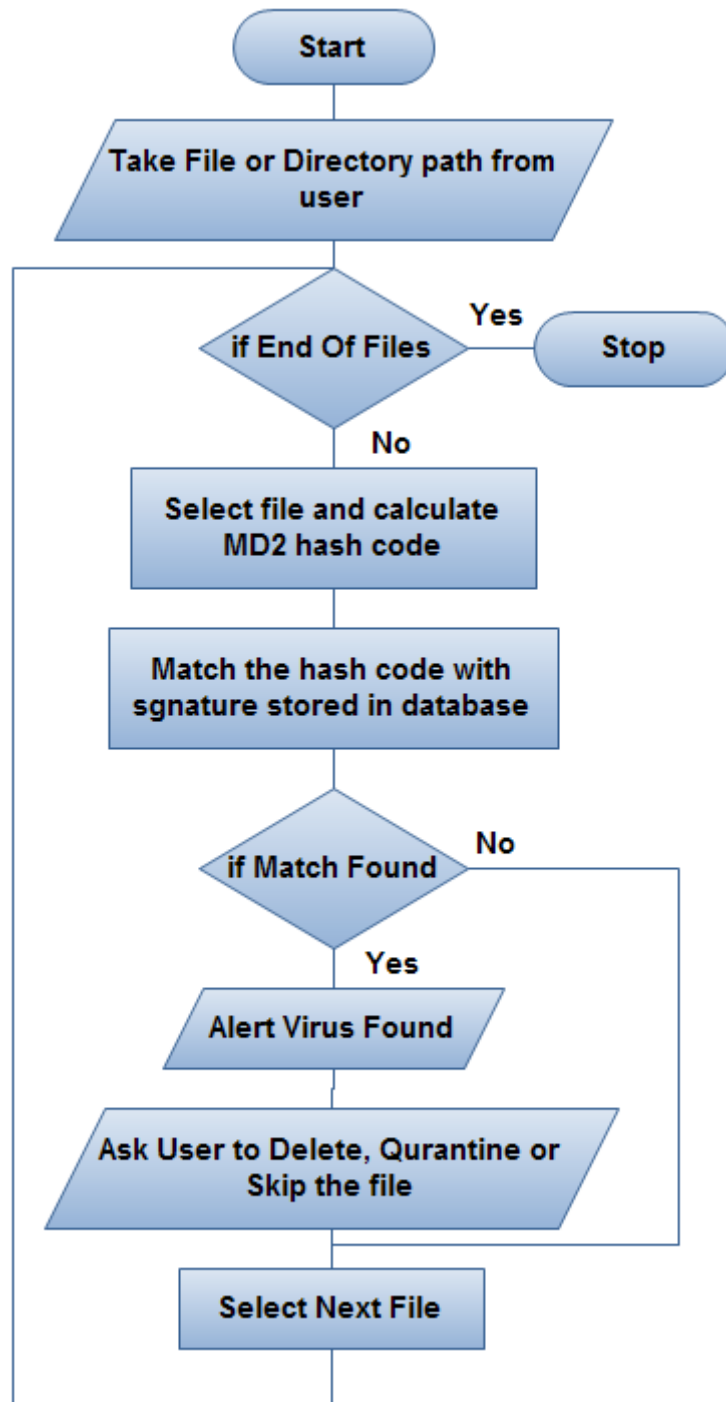


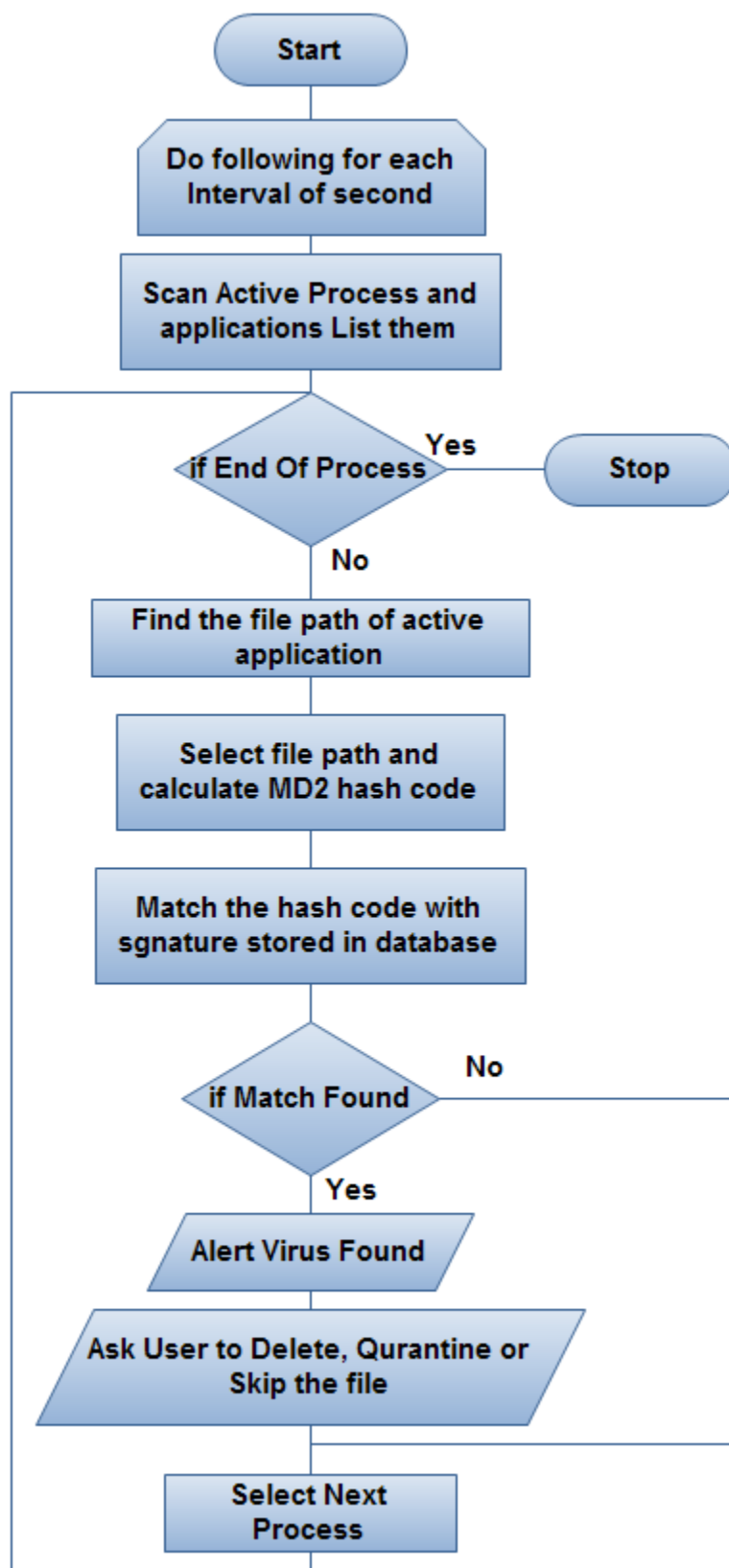
fig. Block Diagram

## Manual Scanner Flowchart





## Real-time Scanner Flowchart



# Development Techniques

### 3. Development Techniques

#### Microsoft Visual Basic 6.0 for 32-Bit Windows Development

Visual Basic (VB) is the third-generation event-driven programming language and integrated development environment (IDE) from Microsoft for its COM programming model. VB is also considered a relatively easy to learn and use programming language, because of its graphical development features and BASIC heritage.

Visual Basic was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects. Scripting languages such as VBA and VBScript are syntactically similar to Visual Basic, but perform differently.

#### Microsoft Standard EXE

##### Modules

Modules are collections of code and data that function something like objects in object oriented programming (OOP), but without defining OOP characteristics like inheritance, polymorphism, and so on. The point behind modules is to enclose procedures and data in a way that hides them from the rest of the program. We'll discuss the importance of doing this later in this chapter when we cover Visual Basic programming techniques and style; breaking a large program into smaller, self contained modules can be invaluable for creating and maintaining code.

You can think of well-designed modules conceptually as programming objects; for example, you might have a module that handles screen display that includes a dozen internal (unseen by the rest of the program) procedures and one or two procedures accessible to the rest of the program. In this way, the rest of the program only has to deal with one or two procedures, not a dozen.

Besides modules, Visual Basic also supports *class modules*,

Programming with class modules will bring us much closer to true OOP programming.

##### Global Items

Global items are accessible to all modules and forms in a project, and you declare them with the Public keyword. However, Microsoft recommends that you keep the number of global items to an absolute minimum and, in fact, suggests their use only when you need to communicate between forms. One reason to avoid global variables is their accessibility from anywhere in the program; while you're working with a global variable in one part of a program, another part of the program might be busy changing that variable, giving you unpredictable results.

Now that we've gotten an overview of the major parts of a project, we'll take a look at how the parts of a project interact, which brings up the idea of *scope*, or visibility in a project.

## **Project Scope**

An object's scope indicates how much visibility it has throughout the project—in the procedure where it's declared, throughout a form or module, or global scope (which means it's accessible everywhere). There are two types of scope in Visual Basic projects: variable scope (including object variables) and procedure scope. We'll take a look at both of them here as we continue our overview of Visual Basic projects and how the parts of those projects interact.

# Implementation

## 4. Implementation

**Name: Main Antivirus (Executable)**

Project Type: Standard EXE

File Name: antivirus.vbp

**Filename: Antivirus.vbp**

Control Center

Type – Form

Filename: frmNavi.frm

Supporting file: frmNavi.frx

Component Used:

- Microsoft Internet Transfer Control 6.0
- Microsoft Windows Common Control 6.0 (SP6)
- Microsoft Windows Common Control2 5.0(SP2)
- User Control File Downloader from BartNet Corp.

Control Name	Type	Caption/Text
frmNavi	Form	Antivirus Control Center
lblNavi(0 to 7)	Label array	
fraAbout	Frame	About
fraVirList	Frame	Virus List
fraLog	Frame	Log
txtLog	Text Box	
lblVersion	Label	
lblTitle	Label	
lblRealtimeStatus	Label	
lblCountRecords	Label	
lblDisclaimer	Label	
lblDescription	Label	
lstVirList	List View	
imgNavi	Image	
imgtitle	image	

## Context Menu

- File
  - Close
  - Exit
- Task
  - Manual Scan
  - Root Scan
  - Full System Scan
  - Scan OS
- Option
  - Configuration
    - Virus List
    - Log
  - Jump To
    - Clean Manager
    - Console Window
    - Control Panel
    - Disk Defragment
    - Security Center
    - System Config

## Antivirus - Manual Scanner

Type – Form

Filename: frmMain.frm

Supporting file: frmMain.frx

Control Name	Type	Caption/Text
frmMain	Form	Antivirus - Manual Scanner
cmdBrowse	Command Button	...
cmdStartStop	Command Button	Start
txtPath	Text Box	C:\
txtStatus	Text Box	
lblClose	Label	X
lblCurDir	Label	
Option1	Option Button	Delete
Option2	Option Button	Quarantine
Option3	Option Button	Skip
lblDescription	Label	
Timer1	Timer	
Picture1	Picture	
Picture2	Picture	

## Context Menu

- Menu
  - Start Control Center
  - Manual Scanner
  - Exit Manual Scanner
  - Exit

## Realtime Scanner Options

Type – Form

Filename: frmOptions.frm

Supporting file: frmOptions.frx

Control Name	Type	Caption/Text
frmOption	Form	Realtime Scanner Options
Frame1	Frame	
Frame2	Frame	
Frame3	Frame	
cmdSave	Command Button	Save
cmdEnable	Command Button	Enable
Command3	Command Button	Default
Command4	Command Button	Default
Text1	Text Box	
Text2	Text Box	
txtLogSize	Text Box	
Lable1	Label	Last File Scan
lblLastFile	Label	
Label4	Label	Scan files below :
Lable6	Label	Scan files type :
Label7	Label	Skip folders :
lblLogSize	Label	Log Size
Option1	Option Button	Delete
Option2	Option Button	Quarantine
Option3	Option Button	Do Nothing
Timer1	Timer	

## Definitions Update

Type – Form

Filename: frmUpdate.frm

Supporting file: frmUpdate.frx



Control Name	Type	Caption/Text
frmOption	Form	Definitions Update
Label1	Label	Update Status
Label2	Label	
Label3	Label	0%
Timer1	Timer	
FD1	FileDownloader	Save

### Infection List

Type – Form

Filename: frmList.frm

Supporting file: frmList.frx

Control Name	Type	Caption/Text
frmList	Form	Infection List
cmdQuar	Command Button	Quarantine
cmdRest	Command Button	Restore
cmdDelete	Command Button	Delete
cmdExit	Command Button	Exit
Label2	Label	
Label3	Label	0%
lvInfect	ListView	

### Infection List

Type – Form

Filename: frmSplash.frm

Supporting file: frmSplash.frx

Control Name	Type	Caption/Text
frmSplash	Form	
lblLoadingStatus	Label	Quarantine
Label1	Label	Loading
Image1	Image	
ProgressBar1	ProgressBar	

## **Name: Real-time Scanner (DLL)**

Project Type: ActiveX DLL

File Name: realthescanner.vbp

BasVersionInfo.bas - To get version of file

basCalculateHash.bas - To calculate Hash Code (MD2, MD4, MD5, SHA1)

basMiscCode.bas – To match file with signature and to play sound

basProcKiller.bas - To kill process

basScanner.bas – To scan file

basdatabase.bas – Database connection strings

Class Module Scanner.cls

### **Third Party Module**

#### **1. File Downloader**

Devoloped By BartNet Corp. [www.bartnet.be](http://www.bartnet.be)

Module is used to download the virus definition from internet uses Microsoft Internet Transfer Control

#### **2. File Downloader**

Devoloped By Trambitas Sorin @ 19.01.2005 <http://www.nekhbet.tk>

Module is used to set application at System tray

#### **3. File Downloader**

Devoloped By Trambitas Sorin @ 19.01.2005 <http://www.nekhbet.tk>

Module is used to set application at System tray

# **Win32 API Calls**

## 5. WIN32 API CALLS

### For Calculating Hash code

Private Declare Function CryptAcquireContext Lib "advapi32.dll" Alias "CryptAcquireContextA" (ByRef phProv As Long, \_

ByVal pszContainer As String, \_

ByVal pszProvider As String, \_

ByVal dwProvType As Long, \_

ByVal dwFlags As Long) As Long

Private Declare Function CryptReleaseContext Lib "advapi32.dll" (ByVal hProv As Long, \_

ByVal dwFlags As Long) As Long

Private Declare Function CryptCreateHash Lib "advapi32.dll" (ByVal hProv As Long, \_

ByVal Algid As Long, \_

ByVal hKey As Long, \_

ByVal dwFlags As Long, \_

ByRef phHash As Long) As Long

Private Declare Function CryptDestroyHash Lib "advapi32.dll" (ByVal hHash As Long) As Long

Private Declare Function CryptHashData Lib "advapi32.dll" (ByVal hHash As Long, \_

pbData As Byte, \_

ByVal dwDataLen As Long, \_

ByVal dwFlags As Long) As Long

Private Declare Function CryptGetHashParam Lib "advapi32.dll" (ByVal hHash As Long, \_

ByVal dwParam As Long, \_

pbData As Any, \_

pdwDataLen As Long, \_

ByVal dwFlags As Long) As Long

### For Find Files

Private Declare Function FindClose Lib "Kernel32" (ByVal hFindFile As Long) As Long

Private Declare Function FindFirstFile Lib "Kernel32" Alias "FindFirstFileA" (ByVal lpFileName As String, \_

lpFindFileData As WIN32\_FIND\_DATA) As Long

Private Declare Function FindNextFile Lib "Kernel32" Alias "FindNextFileA" (ByVal hFindFile As Long, \_

lpFindFileData As WIN32\_FIND\_DATA) As Long

Public Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" (ByVal hwnd As Long, \_

ByVal lpOperation As String, \_

ByVal lpFile As String, \_

ByVal lpParameters As String, \_

ByVal lpDirectory As String, \_

ByVal nShowCmd As Long) As Long

#### For Registry

Private Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA" (ByVal hKey As Long, \_

ByVal lpSubKey As String, \_

ByVal ulOptions As Long, \_

ByVal samDesired As Long, \_

phkResult As Long) As Long

Private Declare Function RegQueryValueEx Lib "advapi32.dll" Alias "RegQueryValueExA" (ByVal hKey As Long, \_

ByVal lpValueName As String, \_

ByVal lpReserved As Long, \_

lpType As Long, \_

lpData As Any, \_

dwSize As Long) As Long

Private Declare Function RegCreateKeyEx Lib "advapi32" Alias "RegCreateKeyExA" (ByVal hKey As Long, \_

ByVal lpSubKey As String, \_

ByVal Reserved As Long, \_

ByVal lpClass As String, \_

ByVal dwOptions As Long, \_

ByVal samDesired As Long, \_

lpSecurityAttributes As SECURITY\_ATTRIBUTES, \_

phkResult As Long, \_

lpdwDisposition As Long) As Long

Private Declare Function RegSetValueEx Lib "advapi32.dll" Alias "RegSetValueExA" (ByVal hKey As Long, \_

ByVal lpValueName As String, \_

ByVal dwReserved As Long, \_

ByVal dwType As Long, \_

lpValue As Any, \_

ByVal dwSize As Long) As Long

# **System Requirements**


## 6. System requirements

For Kinsuk Antivirus to work perfectly, the computer system must fulfill the following requirements:

1. **Processor**
  - a. Pentium, at least 266 MHz
2. **Operating system**
  - a. Windows 2000, SP4 and update rollup 1 or
  - b. Windows XP, SP2 (32 Bit) or
  - c. Windows Vista (32 Bit, SP 1 recommended)
  - d. Windows 7 (32 Bit)
3. **Hard disk**
  - a. At least 32 MB of free memory space (more if using Quarantine for Temporary storage)
4. **Memory**
  - a. At least 128 MB RAM under Windows 2000/XP
  - b. At least 512 MB RAM under Windows Vista

For all installations: Windows Internet Explorer 6.0 or higher  
Internet connection to download virus Definition and updates

### Information for Windows Vista users

On Windows 2000 and Windows XP, many users work with administrator rights. However, this is not desirable from the point of view of security, because it is then easy for viruses and unwanted programs to infiltrate computers. For this reason, Microsoft is introducing the "User Account Control" with Windows Vista. This offers more protection for users who are logged in as administrators: thus in Windows Vista, one administrator only has the privileges of a normal user at first. Actions for which administrator rights are required are clearly marked in Windows Vista with an information icon. In addition, the user must explicitly confirm the required action. Privileges are only increased and the administrative task carried out by the operating system after this permission has been obtained. Antivirus requires administrator rights for some actions in Windows Vista. These actions are marked with the following symbol:  If this symbol also appears on a button, administrator rights are required to carry out this action. If your current user account does not have administrator rights, the Windows Vista dialog of the User Account Control asks you to enter the administrator password. If you do not have an administrator password, you cannot carry out this action.

# Testing



## 7. Testing

On Executing Initially Splash Screen Appears



**Splash Window**

On First Use of antivirus on a system if virus Definition not found Application shows a message box to download Virus definition Database



**Definition Download Message box**

After Downloading Definitions Message box appears to set the Anti virus at boot i.e. Application will be started at the startup.

This is important to activate real-time scanner.



**Set as Startup Message Box**

Now Main Control Center Window appears User Can Select Manual or Automatic System Scan or Can Select Multiple Utility from Context Menu.



Antivirus Control Center

On selecting Manual any other scan Control center is closed to tray an manual scanner window appears here we can browse for folder and set default action to be executed when infection found.



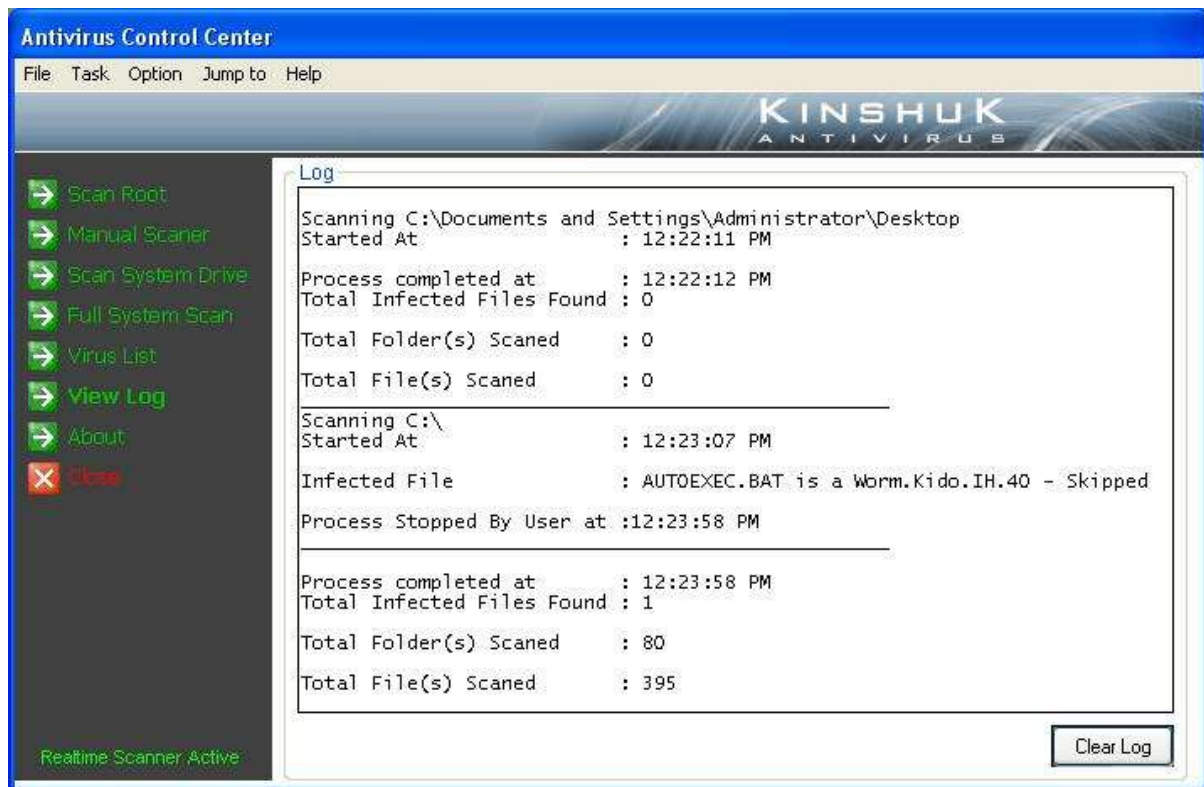
Manual Scanner Window

On after scan complete or stoped by user manual scanner is Minimized to tray and if There is any infection found while scanning The infection it will listed in a listview and user can delete, Quarentine or Restore the Infection



Infection List Window

On Closing any window it is minimized to the System Tray user may open the window by clicking tray icon and selecting option from context menu. If User Selects Start Control Center and the View Log The following Window appears here user can view Log.





Control Center >> Log

# Tray Icon

## 8. Tray Icon

After executing, you will see the Antivirus tray icon in the system tray of the taskbar:

Icon	Description
	Antivirus real-time scanner is enabled
	Antivirus real-time scanner is disabled

The tray icon displays the status of the Antivirus real-time scanner service.

Central functions of Antivirus can be quickly accessed via the context menu of the tray icon. To open the context menu, click on the tray icon with the right-hand mouse button.

Entries in the context menu

- **Start Control Center:** Opens the Antivirus Control Center.
- **Manual Scanner:** Opens the manual scanner.
- **Options:** Opens the Configuration
- **Exit Manual Scanner:** Exit manual scanner.
- **Exit:** Exit the entire antivirus



# **Cost Estimation**

## 9. Cost Estimation

Report timestamp: Wednesday, 14 Jul 2010 11:05 AM

Final Antivirus\antivirus.vbp

BasVersionInfo.bas - 90 Module lines of code

FileDownload.ctl-143 Module lines of code

basCalculateHash.bas - 83 Module lines of code

basCommonDialogs.bas - 174 Module lines of code

basDeclarations.bas - 2 Module lines of code

basFindFiles.bas - 63 Module lines of code

BasMiscCode.bas -101 Module lines of code

basProcKiller.bas - 75 Module lines of code

basRegirtry.bas - 136 Module lines of code

basTrayCode.bas - 107 Module lines of code

basdatabase.bas - 47 Module lines of code

frmList.frm - 61 Module lines of code

frmMain.frm - 254 Module lines of code

frmNavi.frm - 185 Module lines of code

frmOptions.frm - 83 Module lines of code

frmUpdate.frm - 34 Module lines of code

frmSplash.frm - 29 Module lines of code

**1,667 Total number of lines of code**

Realtime Scanner\realtimescanner.vbp

BasScanner.bas - 82 Module lines of code

scanner.cls - 12 Module lines of code

### **94 Total number of lines of code**

Same Modules and self generated codes are rejected.

Main Antivirus 1667 lines of code, Realtime Scanner 94 lines of code

Project Total Lines of code =  $1667 + 94 = 1761$

Delivered Source Instructions (thousands) (KDSI) = **1.761**

Development Mode **Organic**

### **Result**

Effort	<b>4</b> person-months (PM)
Schedule	<b>4</b> months
Productivity	<b>440</b> instructions per person-month
Average Staffing	<b>1</b> full-time-equivalent software personnel

Phase Distribution			
	Effort (PM)	Schedule (mo.)	Staff (avg.)
Plans and requirements *	0.2	0.4	0.5
Product Design	0.6	0.8	0.7
Programming	2.7	2.5	1.1
Detailed Design	1		
Code and unit test	1.7		
Integration and test	0.6	0.7	0.9

\* The plans and requirements phase is calculated in addition to the nominal COCOMO estimate or effort and schedule.

Activity Distribution (Staff) by Phase				
Activity	Phase			
	Plans and Requirements	Product Design	Programming	Integration and Test
Requirements Analysis	0.2	0.1	0.1	0
Product Design	0.1	0.3	0.1	0.1
Programming	0	0.1	0.6	0.3
Test Planning	0	0	0	0
Verification and Validation	0	0	0.1	0.3
Project Office	0.1	0.1	0.1	0.1
CM/QA	0	0	0.1	0.1
Manuals	0	0.1	0.1	0.1
<b>TOTAL</b>	0.4	0.7	1.20	0.99

## **Conclusion and future scope**

## 10. Conclusion and Future Scope

### Why one virus engine is not enough

#### **Multiple virus engines are needed to reduce time lag between virus outbreak and signature update**

There is no single anti-virus engine on the market today that is always the fastest and most effective at identifying viruses, Trojans and other threats. This white paper examines why having multiple anti-virus scanners at mail server level substantially reduces the chance of virus infection and explores ways in which this can be achieved.

It is a well known fact that viruses, Trojan horses, worms, spam, and other forms of malware present a real threat to all modern-day organizations and affect productivity and business operations negatively. *According to the 2006 FBI Crime and Security Survey, 97% of organizations have anti-virus software installed, yet 65% have been affected by a virus attack at least once during the previous 12 months.* Network World cited studies that placed the cost of fighting Blaster, SoBig.F, Sober and other email viruses at \$3.5 billion for US companies alone. *Similarly a 2006 study by the British government found that 43% of companies in the United Kingdom were infected by viruses during 2005.*

### **Does Anti-Virus Software Still Matter?**

Using a computer without an anti-virus program? That's like driving without a seatbelt — or jumping out of a plane without a parachute As users become more knowledgeable about safe computing practices, and as anti-virus solutions vendors ratchet up the general level of hacker hysteria, many people are reconsidering their system security option

# References and bibliography

## 11. References and Bibliography

### Books

- Gary Cornell, “Visual BASIC 6”, Tata McGraw – Hill, 2003
- Dan Appleman's, "Visual Basic Programmer's Guide to the Win32 API", Sams
- Howard Hawhee, Corby Jordan, Richard Hundhausen, Felipe Martins, Thomas Moore  
“TRAINING GUIDE-MCSD-VISUAL BASIC 6”
- Francesco Balena, "Programming Microsoft Visual Basic 6.0", Microsoft Press

### Links

<http://www.antivirusware.com/articles/>

<http://www.antivirusworld.com/articles/antivirus.php>

<http://www.vb-helper.com>

<http://www.nekhbet.tk>

[www.bartnet.be](http://www.bartnet.be)

[http://en.wikipedia.org/wiki/Computer\\_virus](http://en.wikipedia.org/wiki/Computer_virus)

<http://en.wikipedia.org/wiki/Antivirus>

[http://en.wikipedia.org/wiki/IT\\_security](http://en.wikipedia.org/wiki/IT_security)

<http://www.itsecurity.com/features/future-of-antivirus-031108/>

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

[http://vil.nai.com/vil/virus\\_list.asp](http://vil.nai.com/vil/virus_list.asp)

[www.vbtutor.net/vbtutor.html](http://www.vbtutor.net/vbtutor.html)

[en.wikipedia.org/wiki/MD2\\_\(cryptography\)](http://en.wikipedia.org/wiki/MD2_(cryptography))

[www.faqs.org/rfcs/rfc1319.html](http://www.faqs.org/rfcs/rfc1319.html)

[www.rsa.com/rsalabs/node.asp?id=2253](http://www.rsa.com/rsalabs/node.asp?id=2253)