User Documentation

i-scream WinHost

WinHost is a host application for use with the i-scream Distributed Central Monitoring System. This document provides a guide to using the WinHost on your Windows NT/2000 servers.

Revision History

24/03/01	Initial creation				
		Committed by:	pjm2	Verified by:	tdb1
				Date:	24/03/01
26/03/01	Added screenshots of the running WinHost				
		Committed by:	pjm2	Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	
		Committed by:		Verified by:	
				Date:	

Introduction

The WinHost program plays a key part in monitoring Windows-based systems that are monitored by the i-scream Distributed Central Monitoring System. Running a WinHost on your Windows NT/2000 server enables machine statistics to be sent to the i-scream monitoring system.

What is WinHost?

WinHost is a “host” application for the i-scream Distributed Central Monitoring System. The job of a host is to harvest data from the machine on which it is running. The WinHost performs this task on Windows NT and Windows 2000 servers.

The WinHost is self-configuring. It only needs to know the address of a single machine (called a *filter manager*), which it will connect to, obtain its configuration and then proceed to communicate periodically with another machine called a *filter*.

How can I get WinHost?

The latest build of WinHost may be downloaded from the *Builds* section of the i-scream project website: -

http://www.i-scream.org.uk/builds

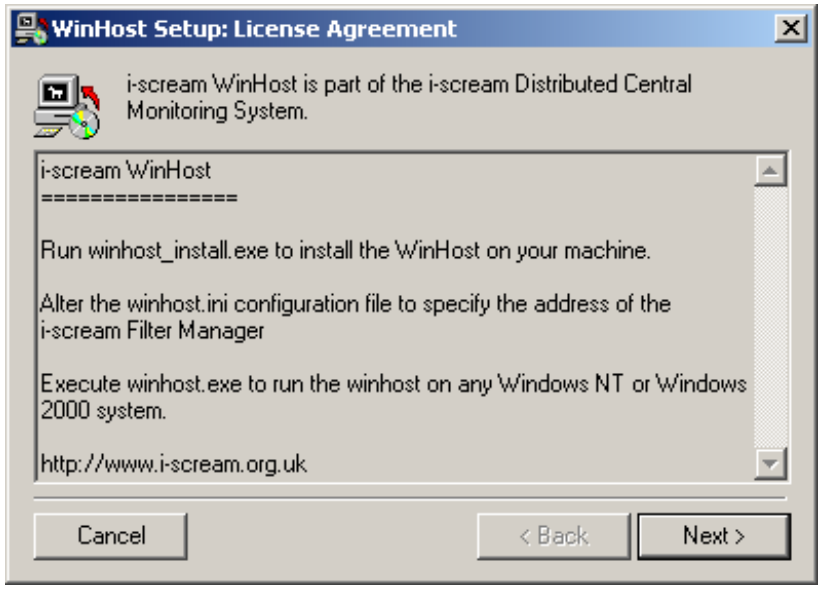
The website also contains other information that you may find useful in settings up an i-scream monitoring system.

How do I install WinHost?

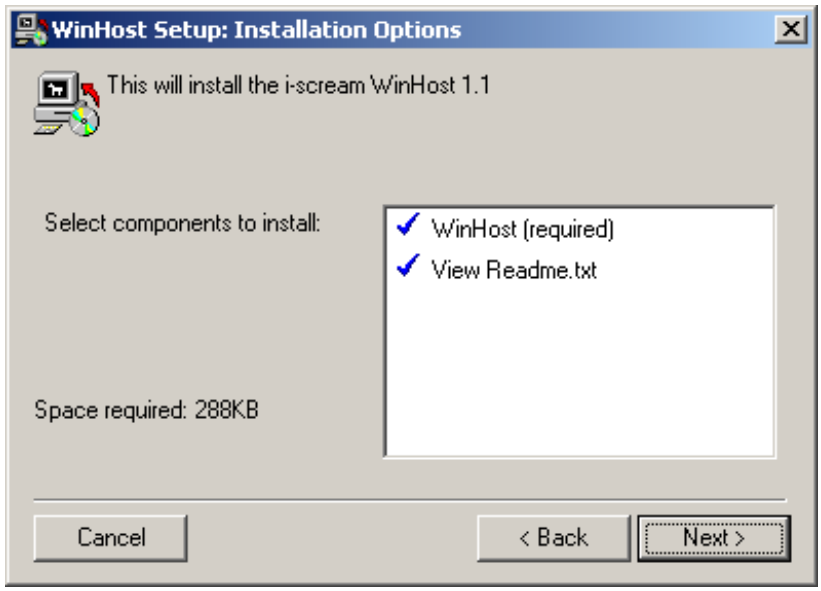
The WinHost is available as a zipped archive from the above URL. You may use a tool such as *WinZip* to view the contents of the archive. To install the WinHost manually, you may copy the contents of the archive to a folder on the Windows NT/2000 machine that you wish to monitor. The WinHost may also be run from a central machine on a network, thus requiring only one installation of the software. However, for best performance, we recommend that WinHost be installed locally on each machine.

To simplify the task of installing the WinHost on many machines, the downloaded archive also contains a small standalone file called *winhost_install.exe,* which is small enough to carry on a floppy disk and installs all the required libraries for the WinHost to operate. The default installation directory for the installer is to *c:\Program Files\winhost*.

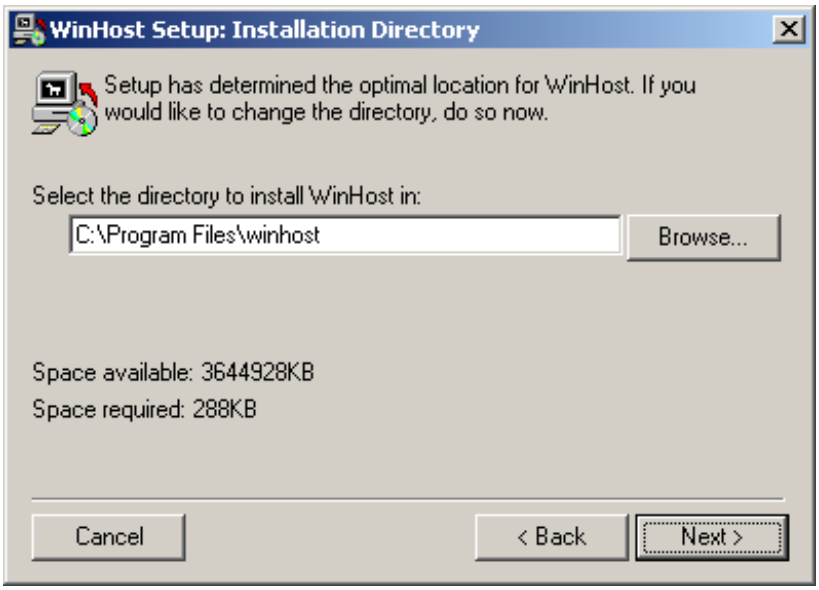
1. The installer starts with this screen, providing a few instructions on use.



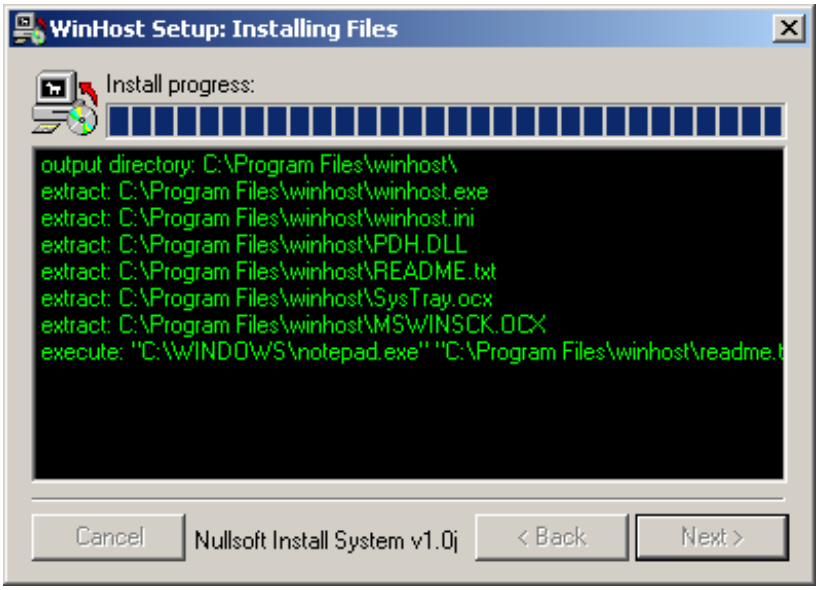
2. You may then select which components you wish to install.



3. The default installation folder is *C:\Program Files\winhost*. Click on *Next >*, unless you wish to install the software in a different folder.



4. The install program then proceeds to install the required files. If you chose to view the readme file earlier, then after this stage, the readme file will be opened for viewing in notepad.



How do I use WinHost?

Running the WinHost is usually a fairly straightforward task. Once you have installed the software, you may execute the *winhost.exe* program from *c:\Program Files\winhost* (or whichever folder you chose to install to).

Starting the WinHost will cause it to configure itself with the i-scream filter manager. The address of this machine is specified in the *winhost.ini* file, which contains essential settings for the WinHost.

After the WinHost has successfully performed its initial configuration, it will send all further communications to a filter machine. This may or may not be the same machine as the filter manager. The filter manager is responsible for telling the WinHost which filter to use, so do not be alarmed if the program begins to send data to another machine.

Configuring WinHost

WinHost is designed to be as simple to operate as possible and remotely configurable, thus, the WinHost itself has only two values that must be configured. The WinHost is dynamically configured via the Filter Manager, so it only needs to know this machine name and port number.

The local configuration for the WinHost is obtained from the file *winhost.ini*. This file is typically found in the same folder that the WinHost was installed to.

The *winhost.ini* file starts by declaring the name of the section as "`[i-scream Winhost]`". This is then followed by two values to define the name and port number of the Filter Manager machine.

Here is an example of the contents of a *winhost.ini* file: -

```
[i-scream Winhost]
FilterManager = killigrew.ukc.ac.uk
FilterManagerPort = 4567
```

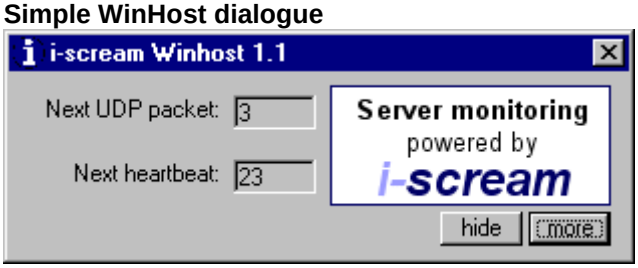
Using the above example, the WinHost would attempt to configure itself by contacting port `4567` of the machine called `killigrew.ukc.ac.uk`.

What does all the output mean?

Initially, output from the i-scream WinHost is not visible, as the program runs as an icon in the Windows System Tray.

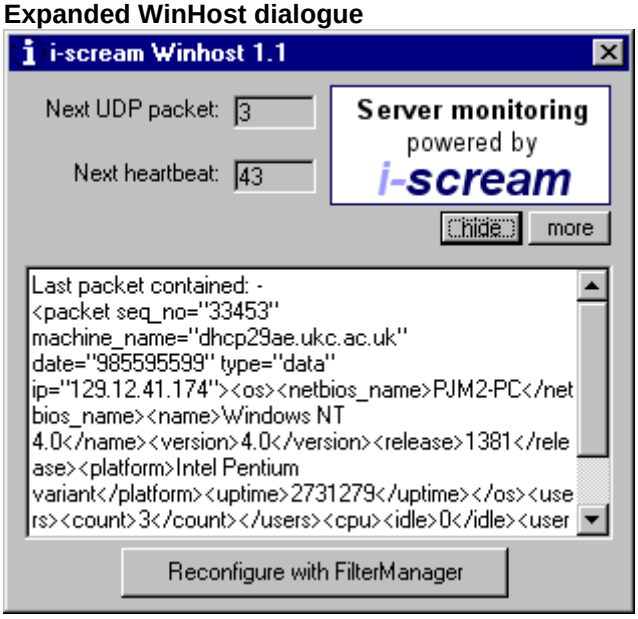
Double-clicking on this icon brings up the WinHost interactive dialogue. This shows the number of seconds left until a *UDP* packet is sent to the filter machine and the number of seconds until a *heartbeat* occurs with the filter machine.

Simple WinHost dialogue



Clicking on “Hide” causes the dialogue to disappear. Clicking on “More” expands the window to provide you with information about the data being sent to the filter or filter manager and gives you the option to force the WinHost to reconfigure with the filter manager.

Expanded WinHost dialogue



When the WinHost is started, it attempts to obtain its configuration from the filter manager. Progress through this stage is displayed in the large text area if the WinHost is running in Expanded mode. It will display the information received from the filter manager as it receives it. This output includes details such as the address of the filter machine that the WinHost will use.

Every time a UDP packet is sent from the WinHost, the contents of the packet will be displayed in the text area. UDP is a type of network protocol, and these packets contain information about the machine that the WinHost is running on.

The progress of each heartbeat is also displayed in this text area. Heartbeats are performed using TCP. TCP is another type of network protocol, but is more reliable than UDP. Heartbeats are sent to ensure that the filter still knows about the existence of the WinHost. Heartbeats are typically configured to be sent less often than UDP packets.

If the configuration of the i-scream server changes, you may see the WinHost reconfigure itself (with similar output to that seen when the program is started). This is normal behaviour and enables the WinHost to be dynamically updated without human intervention.

How can I find out the address of the filter manager?

The filter manager is the section of the server that handles WinHost configuration and assignment to filters. You will need to know the hostname and port number of the filter manager machine before you are able to use the WinHost. If you have trouble finding out this information, then please contact the person(s) responsible for setting up the i-scream filter manager on your network.

I'm concerned about security!

Although we have not taken (yet) any steps to make the entire i-scream system secure, we are very confident that running WinHost does not pose a risk. WinHost only sends data and does not allow any users to connect to it. It is unlikely that WinHost will send any data that may be considered highly sensitive, however, if you are concerned about this, then you may like to review the maintenance documentation to find out exactly what is sent.

Can I alter WinHost to suit my own needs?

Yes, of course you may. The i-scream development team welcomes future contributions from third party developers. All we ask is that if you make a useful alteration that you contact us to let us know what you have done. Such changes may be included in official builds on the i-scream project website. Due to time limitations, WinHost has been written in Visual Basic 6 (A Rapid Application Development Environment) so that people with a basic knowledge of the language (or, indeed, any similar language) will be able to have a good go at customising the way in which it works.

If you are proficient in Visual Basic, then you could start by altering the host to read new data from your system. Once this is done and tested, you could include this data in outgoing UDP packets.

A more in-depth discussion of altering the WinHost may be found in the maintenance documentation for WinHost.

Using WinHost with firewalls

A firewall is commonly used in networked environments to protect the computers from external intrusion. This is often done through restricting the data that can flow between parts of the network. If this applies to you, please ensure you read this section so you are aware of what WinHost attempts to send.

As has been previously discussed, WinHost begins by communicating with the filter manager section of the server. This is done by making a single TCP connection, to a defined port, from WinHost to the filter manager. No connections are ever made back to WinHost from the filter manager.

During the general running of WinHost, communication is made to a filter. The hostname and port of this machine are not usually defined, and are configured automatically by the filter manager. However, the person(s) running the filter will be aware of the hostname and port, and will be able to provide these details.

Communication with the filter is in two parts. The first part is sending UDP packets of data to the filter. These are sent on a regular basis and are not usually large. No UDP traffic is ever sent back to the ihost. There is also a TCP 'heartbeat' communication sent on a less regular basis. This is always established by the WinHost. It is possible that the filter will be configured to run a series of service checks on a host machine. These are always initiated in response to a 'heartbeat' communication, and are of course configurable on the server side.

In summary, WinHost needs to be able to establish TCP connections to both the filter manager and the filter. It also needs to be able to send UDP packets to the filter. If configured to do so, the filter must be able to connect to configured service ports (eg. http, ftp, smtp) on the host, although this is not essential for normal operation.

Further information

Further information is available in our other documentation; the latest versions of which may be found online at the project website. Thank you for using i-scream products.

http://www.i-scream.org.uk