**SnapNShare**

**Summary**

SnapNShare is very basic screen sharing technology that can be utilized for simple forms of collaboration on a subnet of computers. The primary audience of the first release is anyone who is giving a presentation to multiple users who all have their own machines. The presenter and the users must be located on the same subnet (using the same router).

There are two pieces to the technology, the ‘server’, and the ‘viewer’.

The server is run on a machine that wants to share a piece of their screen to the viewers. Any number of viewers can look at the same server in a single session. Once server and viewer agree on a group IP address, the session begins. Any number of viewers can join or leave a session at any time. There can be only a single ‘server’ in a session at one time. Any number of such sessions can run concurrently on a single machine, although different sessions should have different group IP addresses.

Closing the viewer terminates that participants viewing. Closing the server will terminate the session.

**Installation**

The programs do not require any installation beyond copying a couple of .exe files to your local machine. The release can be found here:

[\\eeinmedia01\drop\SampleApps\SnapNShare](file:///\\eeinmedia01\drop\SampleApps\SnapNShare)

There are .exe, .pdb, and this documentation file.

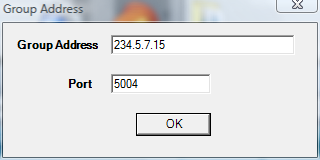
Copy the SnapNServe.exe and SnapNView.exe files to any place on the client machine. **You cannot run them from a remote share, they must be copied to your local machine.**

To remove the programs from your machine, delete the files. Nothing will be left behind. No registry entries, no files, no .dlls, nothing.

[This Space Intentionally Left Blank]

**Running the Server**

A ‘session’ is defined by an IP Multicast address and a port number. A session can be initiated either by the ‘server’ or the ‘viewer’. As soon as you launch the SnapNServe.exe program, you will see a dialog box which is requesting you to confirm an IP address and port number.



**Figure 1**

Before this though, you may see a dialog box from the OS asking you if it’s safe to Unblock this application from doing networking. Click on the “Unblock” button.

You can change the IP address to be anything that is a ‘Class D’ IP multicast address.

Basically, you can change any of the numbers after the 234. , maintaining the same general format.

Once you close this window, you will see the “Capture Window”.



**Figure 2**

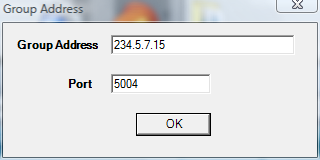
The capture window is semi transparent so you can move it around the screen and see what it is that you will be sharing. When it is the active window, you will see the semi-transparent yellow color. Whether it is the active window or not, it will always capture whatever is within it’s client area 9the area that is yellow.

You can move this around on the screen, resize it, make it the full size of the screen, etc. The server is hard coded to try and capture 4 frames per second while streaming out to the recipients. A modern laptop with a Centrino processor running at 2.5 Ghz can maintain this frame rate with only 20% CPU utilization.

Minimizing the window will not stop the server from sending, but it will essentially send nothing because the window is hidden. To stop serving, simply close the window.

**Running the Client**

Perform the same steps that were performed to launch the server. Select the same IP address as the same dialog box will confront you.



**Figure 3**

The client screen will be black initially, and as soon as it has joined the multicast group, it will start receiving images being sent by the server.

You can move the client window around on your screen, resize it to see more or less of what the server is sending, and generally just let it sit there receiving images.

When you no longer want to see what the server is sending, you can simply close the window. There’s no need for a start/stop button as you can just close the window to stop, and reopen to start again. There is very little startup time.

That is all there is to the basic setup and running of the application.

**Of Special Interest**

This release will not span subnets. If the routers that are used within the network allow the passage of IP Multicast packets, then the sharing will go as far as the routers will allow. Within the Microsoft corporate network, IP multicast packets are not forwarded, so this and any other similar technology will be limited in its scope.

You can launch as many viewers as you want on your client machine. Each viewer can be tuned into a different session hosted by different machines. This is useful in a situation where you want to share what’s on the screens of multiple users at the same time. Each user can run a server, with a unique IP address, and the other users bring up Viewers for each of those unique servers.

When you first launch either the client or the server, it will automatically put a random IP multicast address into the dialog box. If you like this address on the server, simply click ok and accept it. On the client, you will have to type in the address that was indicated. The address is also in the title bar of the Server window so you can easily recall it.

If you are making a presentation, you can simply expand the server window to fill the whole screen. Let it be in the background such that any applications that are launched will show up normally, and their contents will be captured.

If you only want to share a small portion of your screen, it is best to minimize the window as small as necessary to save on network bandwidth.

If you want to share a video of yourself, then launch an application such as GDIVideo which simply shows an image of your face on the screen. It will be captured with everything else on the screen and sent to the viewers.

This application combination works very well in learning environments where multiple machines are available in the space and being utilized by the students. The instructor’s presentation can be served out using SnapNServe, and the students can view the presentation on their machines, at the same time they are performing work. This is beneficial in environments where AV equipment is not readily available, but the students have machines. It is also a good solution to the problem cause by large rooms with small displays. Attendees at the back of the room typically can’t see the small type of what’s being presented at the front. With SnapNShare, then can see the presentation in its full size on their own laptop in front of them.

**Philosophy**

The general philosophy of the applications is parsimony. They are very small, and only do the bare minimum necessary to achieve their goal, which is ubiquitous screen sharing to large groups of people located on an intranet. There is no install, no real options other than selecting the IP address, and no buttons to control the frame rate, compression technique, or anything else that will simply require the user to learn more, and possibly introduce more bugs.

Over time, the code will improve in very small increments, but the philosophy will remain the same. Just do enough to get the job done, and nothing more. There will not be features such as recording, playback and the like. Those types of features will be left to separate specialized applications which will follow a similar philosophy.

**Future Features**

Although the feature set is very small, there are a few features that will enhance the usability even more. The following can be expected in upcoming releases.

**Issue**: At present, it is hard to follow what the presenter is pointing at since cursor tracking and display are not enabled.

**Feature**: Cursor tracking will be added.

**Issue**: It’s hard to correctly position the magic window to capture what I want on the screen

**Feature**: Better positioning and resizing will be enabled.

**Issue**: Maximizing the window still shows a title bar and window border. This reduces the amount of stuff that I can show on my screen.

**Feature**: Maximize will eliminate the border area.

**Issue**: The viewer does not display the full content of what the server is sending. I have to resize the window to see everything.

**Feature**: The client window will have a choice to clip the contents to the client area’s window, as it currently does, or to scale the content to fit within the client area.

**Contact**

William Adams (wiladams@microsoft.com)