Featherweight Virtual Machine, Rether Networks Inc.

1. FVM Overview

1.1 What is FVM ?

Most current day virtualization technologies support virtualization at an abstraction level close to hardware and are heavyweight in that each Virtual Machine (VM) is created as a full-fledged operating environment. Initializing such a VM incurs too much overhead in terms of both disk space and invocation latency.

Featherweight Virtual Machine (FVM) is an OS-level virtual machine technology that creates one or multiple execution environments on a single physical machine. Different from hardware-level virtual machine technologies, OS-level virtual machine technologies like FVM have the virtualization layer between the operating system and application programs. The virtualization layer can be designed in a way that allows processes in VMs to access as many resources of the host machine as possible through a special copy on write scheme, but never to tamper with them. In other words, every VM shares the same execution environment as the host machine, and only keeps any diverges from the host environment in the VM's local state. Therefore, such a VM can have very small resource requirement and thus can achieve large scalability.

Each VM represents a distinct instance of the underlying physical machine, and does not interfere with one another or with the underlying machine. This isolation property makes FVM a possible building block for security and fault-tolerant applications. For example, running unsafe mobile code in a VM can protect the underlying physical machine from being compromised. To prevent denial-of-service attacks and also support performance isolation, a set of policies regarding resource quota and network access can be specified when a VM is created. The FVM layer limits the total system resource allocated to the VM according to these policies. This is achieved by assigning a Windows job object to the VM, initializing the job object with the policy settings for things like CPU scheduling priority, physical memory limit, working set size, process execution time, etc.

Moreover, under this architecture, it is also possible for the VM and the host machine to synchronize state changes conveniently when necessary. For example, the legitimate state change in a VM can be committed to the host machine, while patches or reconfiguration of the host machine can be synchronized immediately in a VM.

1.2 How does FVM work ?

The key idea behind FVM is namespace virtualization, which renames system resources through a virtualization layer, called FVM layer, at the OS system call interface. Microsoft Windows supports numerous types of namespaces for various system resources, such as files, registries, kernel objects, network address, daemon services, window classes, etc. The FVM layer manipulates the names of all these resources when a process makes system calls to access them.

Through resource renaming, the namespaces visible to processes in one VM are guaranteed to be disjoint from those visible to processes in another VM. As a result, two VMs never share any resources and therefore cannot interact with each other directly. For example, suppose an application in one VM (say vm1) tries to access a file /a/b, then the FVM layer will redirect it to access /vm1/a/b. When a process in another VM (say vm2) accesses /a/b, it will try a different file, i.e., /vm2/a/b, which is different from the file /a/b in vm1.

As an OS-level virtualization technology, FVM puts the virtualizationlayer at the OS's system call interface. All the VMs share the host OS's kernel-mode component, including the hardware abstraction layer, device drivers, OS kernel and executive, as well as system boot components. Moreover, the filesystem image is also shared by default. Each new VM starts with exactly the same operating environment as the current host. Therefore, both the startup delay and the initial resource requirement for a VM are minimized. Because the resource virtualization is performed by simply renaming system call arguments instead of complicated resource mappings or instruction interpretations, an application's runtime performance in a VM is also improved.

The FVM virtualization layer is implemented by intercepting Windows system calls, which are exposed to user-mode applications through a set of user-mode Dynamic Link Libraries(DLL).We prefer to do the interception at the kernel-mode interface because it is more difficult to be bypassed or subverted than user mode interceptions. There are two categories of system calls on NT-based Windows OS according to the functionalities they provide. The first category is system calls for basic OS services like file I/O and object management, whose kernel-mode interface is well documented in [34]. However, the second category of system calls, which are composed of system calls managing daemon service, GUI window and network interface, either have no corresponding kernel mode interface, or have a kernel mode interface but have no clear documentation. To intercept this category of system calls, we move the virtualization layer to the user-mode DLL interface.

The kernel-mode component is a kernel driver that modifies the system call entry point in the System Service Dispatch Table(SSDT) within the kernel, while the user-mode component is a DLL that uses detours to do the interception. Once the virtualization layer is attached to the host machine, it can redirect different requests from user-mode applications through FVM's virtualization logic.

1.3 Possible applications of FVM

Applications of the FVM technology could be in areas of secure mobile code execution service, vulnerability assessment support engine, scalable web site testing, shared binary service for application deployment and distributed Display-Only File Server. For more information about these, please refer to the research papers mentioned below.

1.4 Research Papers

A Featherweight Virtual Machine for Windows Applications, Proceedings of the 2nd ACM/USENIX Conference on Virtual Execution Environments (VEE'06), June 2006 http://www.ecsl.cs.sunysb.edu/tr/TR189.pdf

Applications of a Featherweight Virtual Machine, Proceedings of the 2008 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE08), Seattle WA., March 2008 - http://www.ecsl.cs.sunysb.edu/tr/TR224.pdf

Automated and Safe Vulnerability Assessment, Proceedings of 21st Annual Computer Security Applications Conference (ACSAC 2005), Tucson, AZ., December 2005. http://www.ecsl.cs.sunysb.edu/tr/TR183.pdf

OS-level Virtualization and Its Applications, Yang Yu, Ph. D. Dissertation, December 2007. - http://www.ecsl.cs.sunysb.edu/tr/TR223.pdf

2. FVM installation

2.1.1 Pre-requisites for FVM installation

You need the following pre-requisites to be able to build and use FVM.

- Microsoft Visual Studio 6 with VC++ compiler
- Microsoft Detours Express 2.1 (http://research.microsoft.com/sn/detours)
- Microsoft Windows Driver Kit (http://www.microsoft.com/whdc/DevTools/WDK/WDKpkg.mspx)

System Requirements :

- Operating Systems: Windows 2000, Windows XP
- Memory Requirement: 256 Megabytes

2.1.2 Installation using the standard Rether Networks Inc. FVM automated installer

Follow the steps below..

1. Click on the installer and proceed with the subsequent steps to install FVM on your machine.
2. The default installation location is at "C:\Program Files\Rether Networks Inc\FVM", unless you modify the default installation path(assuming Windows is installed on partition C).
3. Upon successful installation, the installer would prompt for a computer restart. Go ahead and restart the computer.
4. A successful installation should have created the following set of files :

C:\Program Files\Rether Networks Inc\FVM\fvm.exe
C:\Program Files\Rether Networks Inc\FVM\fvmserv.exe
C:\Program Files\Rether Networks Inc\FVM\fvmInLsp.exe
C:\Program Files\Rether Networks Inc\FVM\FvmShell.exe
C:\WINDOWS\system32\fvmLsp.dll

2.1.3 Steps to build the *fvmdll* component in FVM

1. Download and install the Microsoft Detours Express package to get the detours related library (***detours.lib***).
2. Download the FVM source from http://sourceforge.net/projects/fvm-rni
3. Build the *fvmdll* component of FVM. To do this copy the *detours.lib* library (from the installed instance of Detours Express) into the *fvmdll* source folder, and then build *fvmdll* (Use Visual Studio 6 IDE).
4. Copy the dll file ***fvmdll.dll*** obtained from step 3 above, into the folder C:\Program Files\Rether Networks Inc\FVM\.

   The FVM driver is loaded by default upon a restart. The windows FVM client that communicates with this driver is located at C:\Program Files\Rether Networks Inc\FVM\fvm.exe

2.1.4 Manual Installation/Configuration

For manual configuration of FVM (not using the FVM installer) the following are the steps,

1. Download the source code for FVM from https://sourceforge.net/projects/fvm-rni.

2. Descend into each of the FVM component folders and build them either using supplied projects or consult the README files in the folders.

3. Refer to section ***2.1.2*** to build the fvmdll component. However copy the fvmdll.dll dll into the C:\FVMBIN folder mentioned below in step 4.

4. Ensure that you have the following Windows dlls present on the host OS.

   MFC42UD.DLL
   MFCO42UD.DLL
   MSVCP60D.DLL
   MSVCRTD.DLL

5. Create a directory holding all FVM binaries, for example:

   C:\FVMBIN
   fvm.exe
   fvmshell.exe
   fvmserv.exe

fvmdll.dll
fvmInLsp.exe

6. Copy the following files to the system directory(C:\WINNT\system32).

   sporder.dll
   fvmlsp.dll

7. Copy the FVM driver "*hooksys.sys*" to driver directory (C:\WINNT\system32\drivers)

8. Go to C:\FVMBIN

   To start FVM, type:
   fvmInLsp.exe -install

   fvmserv.exe -i

   To stop FVM, type:
   fvmserv.exe -r

   fvmInLsp.exe -remove

   To start FVM GUI, type:
   fvm.exe

The subsequent FVM client configuration/usage steps are the same as described in the subsequent sections.

2.2 Loading/Unloading the driver

1. Use the fvmserv.exe utility located at C:\Program Files\Rether Networks Inc\FVM\ to load/unload FVM driver.

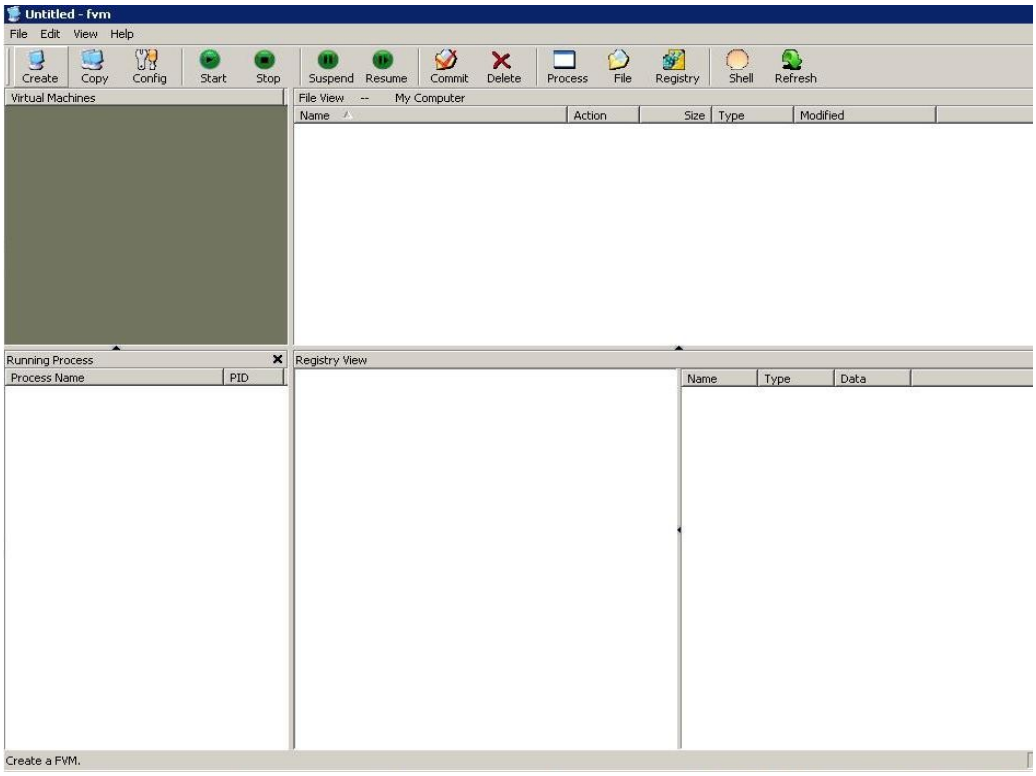
   "fvmserv.exe –i" loads the driver(start FVM) and "fvmserv –r" (stop FVM) unloads it.

2. The driver could be loaded/unloaded using the desktop shortcuts created by default FVM installation. i.e., Startup FVMServer/Stop FVMServer. The shortcuts are also installed in the Start->Program->Rether->FVM location.

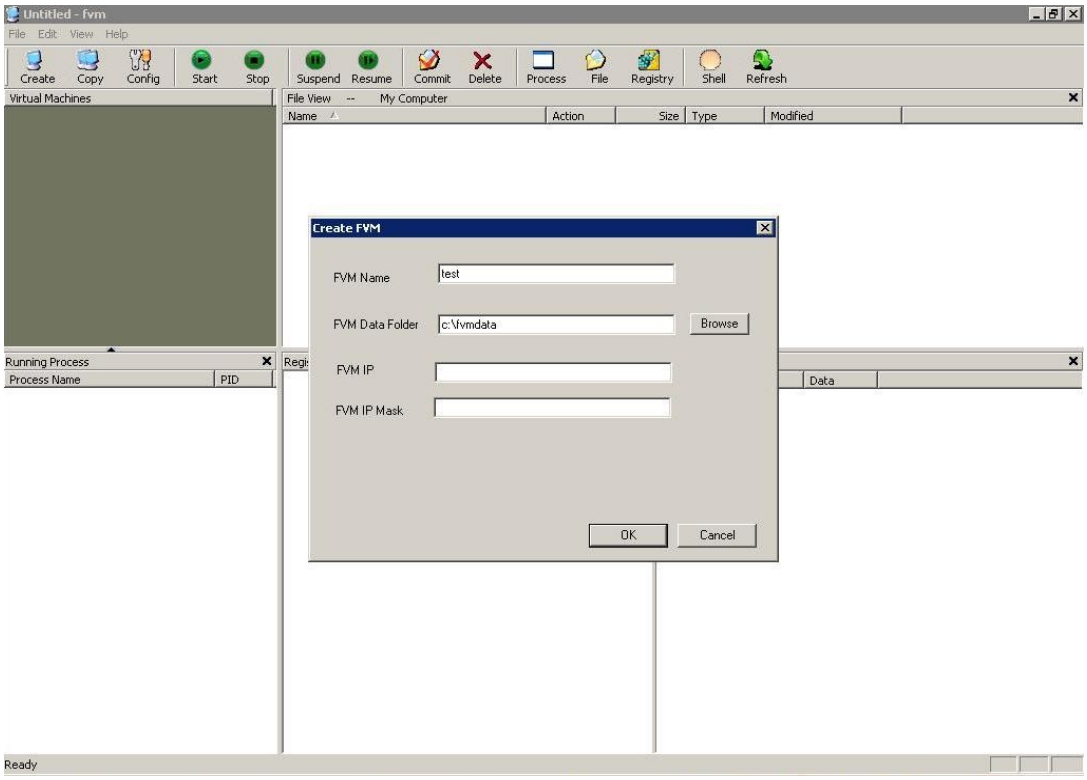
3. For the network virtualization, you need to use fvmInLsp.exe –[install/remove] to install/remove the FVM Layered Service Provider (LSP). Install the FVM LSP driver before loading the FVM driver, and uninstall it after unloading the FVM driver.

2.3 Starting/using FVM GUI utility

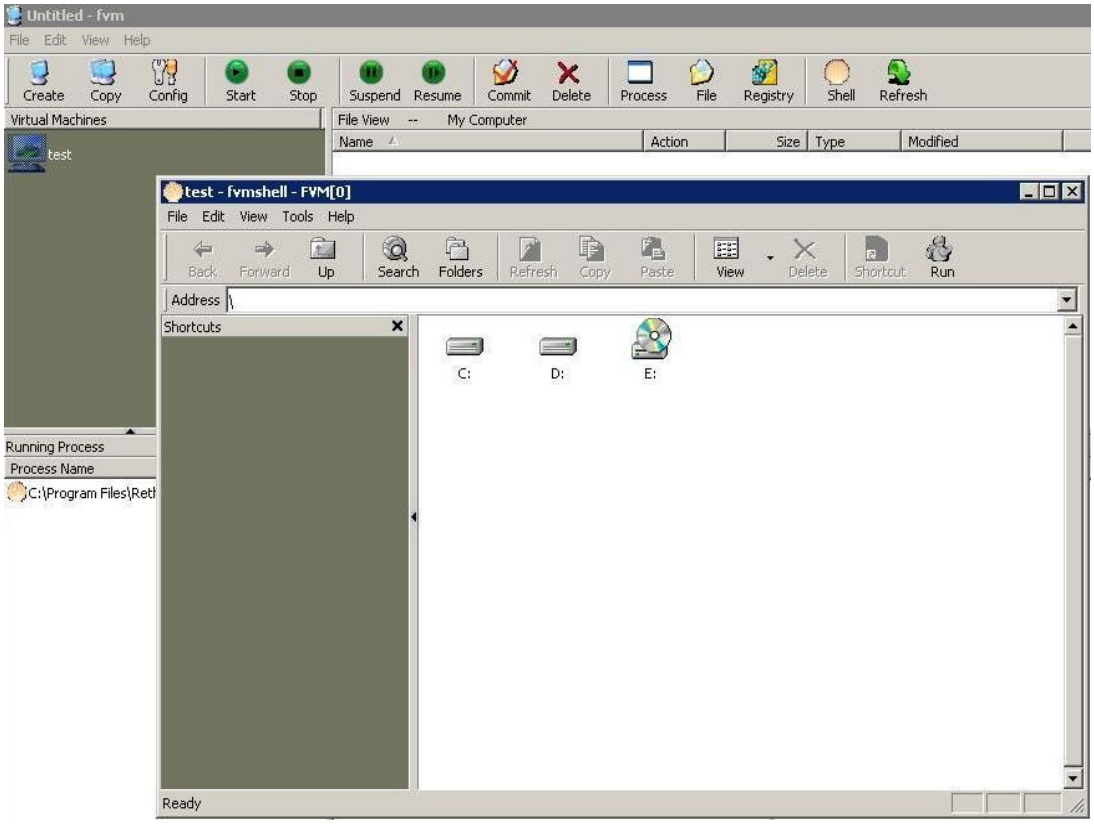
1. C:\Program Files\Rether Networks Inc\FVM\fvm.exe (or more simply, use the shortcut created on your desktop after the successful installation) to launch the FVM User Interface (UI). FVM UI would show up as shown in the next figure.



2. You can create new VM's using the Create tab. Upon clicking the Create tab, you would see the following dialog. Note that FVM IP/FVM IPMASK are optional and you can use them in case you intend to use network virtualization. However, specifying the name of the FVM (any convenient name) along with a working folder to store FVM private data is mandatory.



3. Upon successful VM creation, the FVM can be started by clicking on the "Start" button, on doing which you would get FVM shell (refer below). In case you get a tab indicating that driver is not loaded, it implies that you need to load the driver again using the *fvmserv* utility (fvmserv –i) and then try to start the FVM.

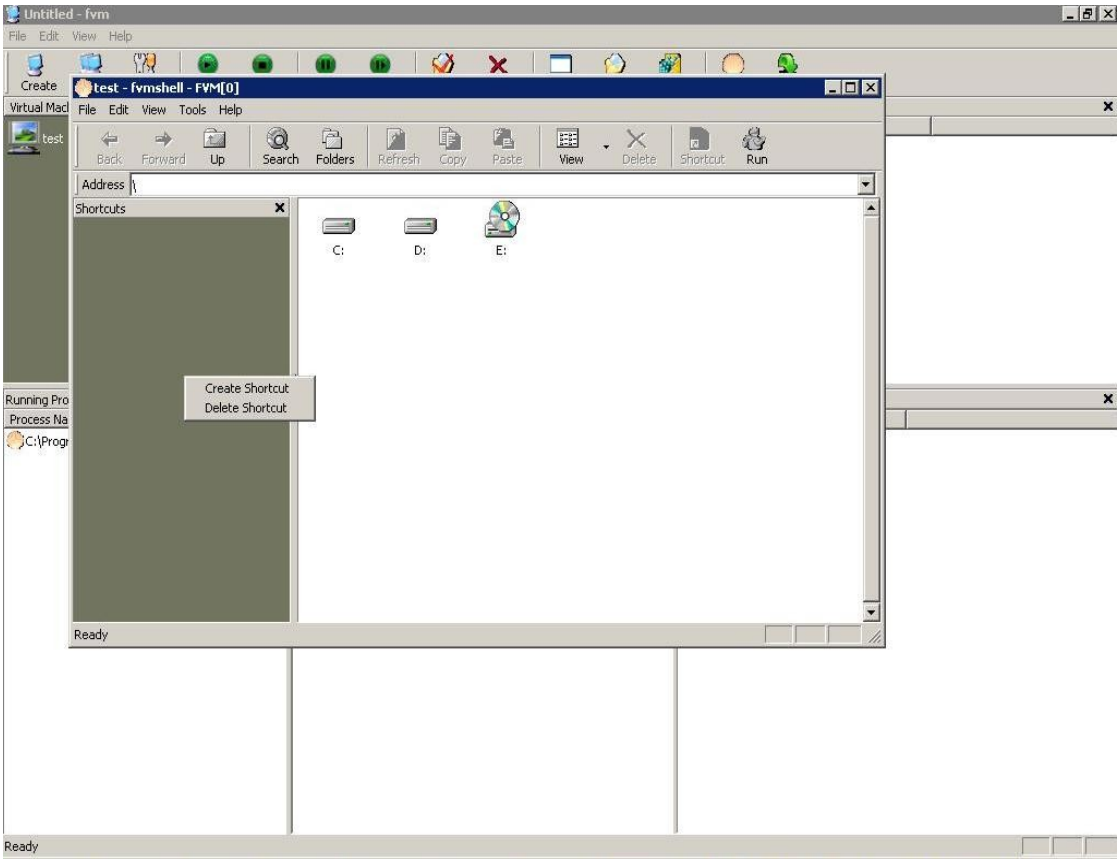


4. You can see the entire directory hierarchy (C;\, D:\ etc) inside the launched FVM shell. Any update made to the system (addition/deletion of files, registry entries etc) is now confined only to the FVM (indicated by the FVM shell that is used to perform the update). The files/registry changes could be tracked/verified by using the File/Registry tabs of FVM. You can use the Process tab to verify the various processes launched under this FVM. Note that any changes made by the process (or recursively, the processes launched by the processes inside FVM) are confined only to the FVM.

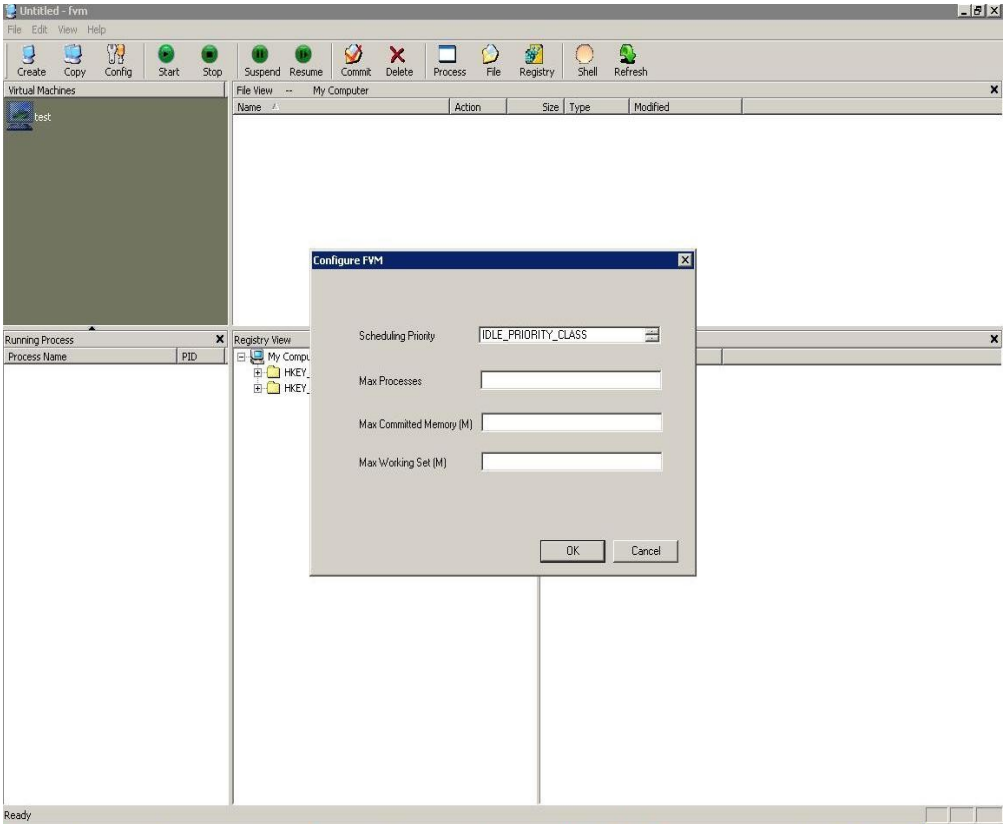
5. An FVM can be stopped by using the "Stop" button and subsequently deleted using the "Delete" button.

6. "Commit" button is a special tab that writes back the changes made inside the FVM to the base machine. Once written back, the contents of the FVM are essentially sync'ed with the host machine.

7. Note for “Ease of Use”, you could create various shortcuts of the frequently used applications inside FVM shell. This is achieved by right-clicking on the FVM shell and creating the shortcuts as shown below.



8. Each VM can be individually configured to use user specified resources. The "Configure" dialog in the FVM UI could be used for the same. Among the configuration parameters, one can specify the maximum number of processes, physical memory limitations and maximum working set per process.



9. FVM UI can also be used to clone VM's. The "Copy" button in the FVM UI needs to be used to do the same. However, note that the Copy button can only be used when the VM is stopped.

3.1 Known problems and limitations

1. COM/RPC: COM/RPC virtualization in FVM v1.0 is not complete.
2. Installer (Windows XP specific) : Windows XP has an optimization called "Prefetch" which among other things caches binaries and executables (in <windows_dir>/prefetch). This caching is done for all the executables which use the %tmp% to uncompress themselves. The Windows task scheduler service then backs up the binaries in the above specified prefetch location, which is used to launch the binaries. In current behavior the prefetch files are not getting created on the host. So application launch fails.

3. Executables launch sometimes fails when they are placed inside the %tmp% directory. However, when the exe's are copied to a different location, their launch succeeds.
4. Acrobat Reader installation doesn't get cleaned up completely
5. Multiple IIS instances cannot run simultaneously on a Windows host. So you can't have different IIS instances running on different VMs.
6. Copy-and-paste isolation between VMs is not complete and this is part of the future roadmap for FVM.

3.2 Future Roadmap

1. Service Virtualization : Currently, some of the Windows daemons (like the installer service) do not run inside FVM. FVM has to be extended to support service virtualization.

2. Advanced NTFS features like Alternate Data Streams (ADS/fork) are currently not supported.

3. Clipboard Virtualization : The Windows clipboard is currently shared between FVM's and is not virtualized. Virtualization of windows clipboard is not released as part of the current FVM version.

3.3 Contact Point

For any bug reports, feature suggestions etc., contact us at support@rether.com or call (+1) 631-632-3751.