



Open Antivirus Solutions for Enterprise Applications

WolfeReiter has developed a free and open source antivirus integration library, WRALib, as a tool for protecting databases and intranets from viruses. WRALib ships with support for the open-source ClamAV virus scanning system, but is extensible for use with other virus scan engines.

Many enterprises build custom workflow and document management applications which include the feature of posting and retrieving documents and other files. While most organizations consider virus countermeasures for the desktop and email, few consider virus countermeasures for these enterprise applications.

Traditional server antivirus solutions fall short for custom applications in three important ways:

Capability: Many enterprise applications store files as Binary Large Object (BLOB) data in a relational database system like Oracle or SQL Server. File-based scanners have no mechanism to access BLOB data in a database.

Integration: Most antivirus systems do not have a mechanism to integrate with the custom application. This makes it difficult to determine which posted files have been scanned and are safe versus those that have not yet been scanned.

Stability: Many antivirus systems for Windows Server install kernel-mode drivers which may destabilize the server. We have experienced antivirus systems that caused a server to reboot every 2-3 days. Worse, on-access file-based antivirus scanners can force Java and .NET applications to restart or recompile frequently by touching each file and changing its timestamp when the application server process accesses ASPX or JSP pages in the application.

Our free and open source antivirus integration library, WRALib, solves these problems by providing the tools to integrate a virus scanning engine directly into the an enterprise application. The current version of WRALib is written in C# and runs in Novell Mono and Microsoft .NET environments. It ships with support for the ClamAV GPL antivirus engine. It is also easily extensible to support other engines such as the Symantec AntiVirus Scan Engine.

WolfeReiter

thoughtful computing

WolfeReiter provides comprehensive software services to business, nonprofit, and government clients.

- Software Design and Development
- Systems Integration
- Consultative ROI Analysis & IT Investment Planning
- Project Planning and Management Based on the Industry-Standard Software Development Life Cycle Approach
- Application Security Design and Implementation

Contact us today for a free consultation or visit www.thoughtfulcomputing.com to learn more.

941 O Street NW
Washington, DC 20001
info@thoughtfulcomputing.com

202.468.2752



Technical Discussion

WRAVLib with ClamAV is a client-server technology that is both horizontally and vertically scalable. ClamAV runs on most UNIX/Linux distributions and we provide a build that runs in the Windows Services for Unix Interix subsystem. Interix is a free component supported by Microsoft that upgrades the Windows NT POSIX subsystem in NT 5.x kernel operating systems to a much more rich UNIX environment based upon OpenBSD.

ClamAV provides a TCP daemon, clamd, which accepts a stream and scans it. Clamd may run on the same server as the application server or a remote server or servers. Clamd on Windows runs in the Interix subsystem which is almost totally separated from the standard Win32 subsystem. This subsystem isolation is advantageous because it ensures that even if clamd crashes, it cannot destabilize applications running in the Win32 subsystem. Since clamd runs as a user-mode process and not kernel-mode, a clamd failure cannot destabilize the server.

WRAVLib provides a C# interface to send files, whole recursive directory structures or BLOBs from a database to clamd. WRAVLib provides access to an antivirus engine through the concept of an *agent* which serves as a facade for the underlying antivirus engine. Adding support for a new antivirus engine simply requires defining a concrete instance of *IVirusScanAgent* or inheriting and extending the abstract class *VirusScanAgent*. Scan results are returned via *ItemScanCompleted* and *VirusFound* events. Any code that needs to know the result of a scan simply registers an event listener for these events. The *VirusScanAgent* abstract class and *ClamdStreamAgent* concrete class send these messages to Log4Net as well. Log4Net has the advantage of being highly configurable at deployment time. Without recompiling, by editing XML tags in a configuration file, it is a simple matter to log scan events to the Windows Event Log, a log file, a database table, or have them sent as SMTP mail messages to the server administrator. Figure 1, below, is a UML diagram showing the high-level structure of key classes.

WolfeReiter

thoughtful computing

[WRAVLib class documentation](#)

[Download WRAVLib strong-named binaries](#)

[Download WRAVLib source code](#)

[Download the latest ClamAV build for Interix](#)

[Learn more about ClamAV](#)

[Learn more about Microsoft Services for UNIX and Interix](#)

[Learn more about Novell Mono](#)

[Learn more about Microsoft .NET](#)

[Learn more about Log4Net](#)



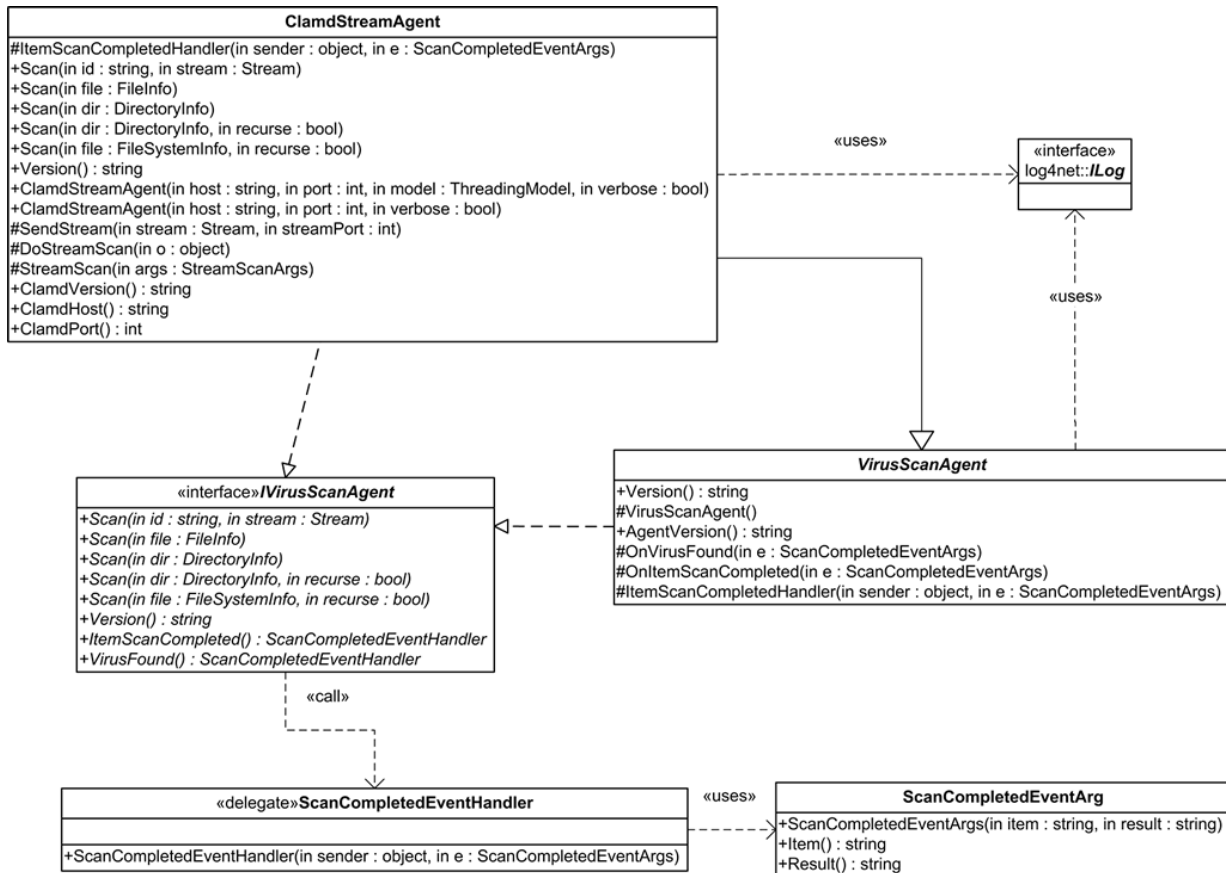


Figure 1: UML Static Structure Diagram

From the perspective of a developer using WRVLib in a custom application, the object model is simple. In the following snippet, for example, WRVLib scans the Windows C:\ volume, recursing into each directory with verbose logging. Another overload of Scan() provides an interface compatible with a BLOB retrieved from a database.

```

IVirusScanAgent agent = new ClamdStreamAgent(
    "localhost", 3310, true );
agent.Scan( @"C:\" );
  
```

The ClamdStreamAgent is instantiated with the host name and port where clamd is running. The constructor also requires a Boolean which determines whether to use verbose logging. Optionally, the developer can choose whether the ClamdStreamAgent schedules each scan job with a thread pool. By default, the scans are run one at a time on a single thread. The constructor





also gets an ILog instance using the *LogManager.GetLogger()* factory in Log4Net.

Figure 2 is a UML sequence diagram which shows the interaction between the components of the system. When one of the overloads of the *Scan()* is called, *ClamdStreamAgent* uses the clamd STREAM protocol to send a byte stream to clamd for scanning and to capture the result of the scan. When the result comes back, *ClamdStreamAgent* raise the *ItemScanCompleted* event and if a virus was found raises the *VirusFound* event. *ClamdStreamAgent* contains event listeners for both of these events. The event listeners use an instance of ILog to log the events with Log4Net. These two events are public, so a developer can easily register handlers for them. Our sample console program demonstrates this technique. It uses event handlers to increment integer counters that it outputs as summary files scanned and viruses found statistics at the end of a scan job.

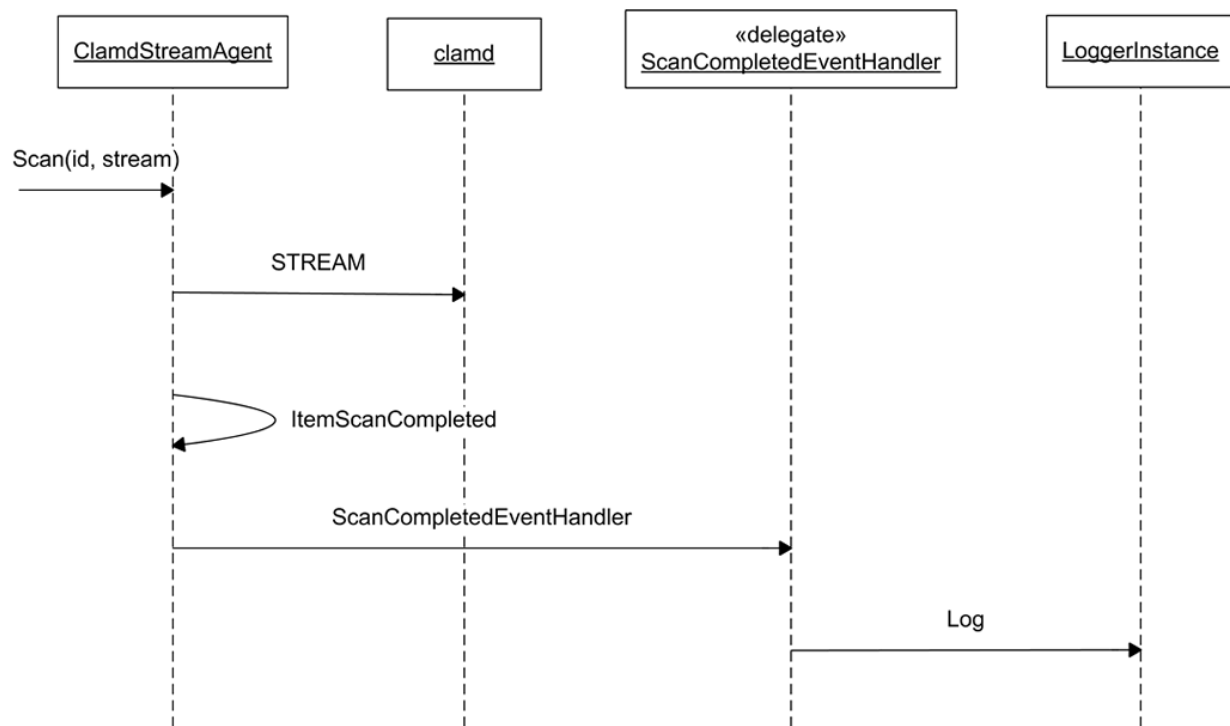


Figure 2: UML Sequence Diagram





Conclusion

WRAVLib is a free and open source antivirus engine integration solution that provides an object oriented, event-based programming framework for applications that run on Novell Mono or Microsoft .NET runtimes. It provides an elegant solution to the problem of including virus countermeasures in custom applications that target the .NET or Mono platforms. It also provides a reference for integrating .NET applications with UNIX code running in the Interix subsystem for Windows.

WRAVLib is world-ready: all strings are centralized into a RESX resource file with English as the default culture. Support for any language can be added simply by translating Strings.resx to the target language and compiling a satellite assembly.

WRAVLib is licensed under The zlib/libpng License. It is free for any use, including commercial applications.

WolfeReiter

thoughtful computing

WolfeReiter provides comprehensive software services to business, nonprofit, and government clients.

- Software Design and Development
- Systems Integration
- Consultative ROI Analysis & IT Investment Planning
- Project Planning and Management Based on the Industry-Standard Software Development Life Cycle Approach
- Application Security Design and Implementation

Contact us today for a free consultation or visit www.thoughtfulcomputing.com to learn more.

941 O Street NW
Washington, DC 20001
info@thoughtfulcomputing.com

202.468.2752

