CIBER's Understanding of the 2005 EAC VVSG COTS Standards

As requested by the New York State Board of Elections, CIBER, Inc.(CIBER) is providing the opinions stated below on the 2005 Election Assistance Commission Voluntary Voting System Guidelines (VVSG) commercial off-the-shelf (COTS) standards. At this moment, the EAC has not accredited any labs to the test to the 2005 standards 2005 EAC VVSG. Therefore, CIBER is basing these opinions on our experience as a National Association of State Election Directors accredited Independent Testing Authority (ITA).

CIBER's Experience with COTS and Third Party Software Issues

CIBER worked with the 2002 Federal Election Commission Voting System Standards as an accredited ITA and participated in resolving many COTS and third party software related issues. Some of these issues surrounded the following items:

- General purpose unmodified COTS software (e.g. operating systems, browsers, database management tool, compilers)
- Unmodified COTS software (e.g. open source components, third party utilities)
- Modified COTS software (e.g. open source components, third party utilities)
- Unmodified COTS software development tools (e.g. Component One, Altova, JAXB)

CIBER has provided the following to illustrate specific examples of COTS issues that CIBER has helped to resolve.

- COTS open source components and third party utilities that do not provide a compiled form but do provide source code for an end user to compile. These versions may be the default source code or may have changes to them for specific applications.

- COTS software development tools that are embedded or compiled in the source code. Many software packages have been created to decrease development time and expense. These packages are deployed with the compiled version and application and end up in the executable code.

- COTS software development tools that generate source code. Software packages can now create source code for developers. This code is regenerated during the compile. A comparison of this code requires more than just comparing the code with hash tools because the format of the source code or the code structure may differ each time it is compiled.

The following sections outline CIBER's opinions and interpretations of specific sections of the VVSG. In these sections, we have identified the VVSG requirement, which is then followed by CIBER's response. According to the VVSG Volume II Requirement 1.8.4

Resolution of Testing Issues, the EAC "will have a process for the accredited test labs, vendors and election officials to request an interpretation of the Guidelines." Due to this requirement, CIBER can only offer an opinion on the interpretation of the guidelines, but the official interpretation must come from the EAC.

As applied to the New York State Board of Election Contract below are definition of term agreed upon by CIBER and NYSTEC.

Source code: A series of statements written in a human-readable computer programming language. The series of statements, often consisting of several files or modules are then converted to a computer executable format producing computer program.

Compiler: A computer program that translates source code into another computer language that a particular computer and is capable of executing. Compiled code is generally not readable by a human and cannot easily be altered. Source code that has been translated by a compiler is often called object code, compiled code , executable code or binary code.

Interpreted code: Interpreted code is similar to source code in that it is generally human readable however it is not compiled into executable code like source code. Interpreted code is executed by an interpreter (software program running on a machine) at run time and dynamically translated into machine executable instructions. This process happens every time the interpreted code is executed.

Voting system: For the purposes of determining what source code should be included in the source code review process it must be clear that a "Voting System" should be defined as any component or software that has any influence on an election. This should include the voting system itself (DRE or OpScan), devices used to mark ballots (Ex. AutoMark), Election Management Software, ballot creation software, third party software written specifically for elections, source code generated by third party tools or modules specifically designed to be used in an election.

Code Inspection: A review to determine version, completeness, consistency, correctness, modifiability, structure, traceability, modularity, and construction.

Code Examination: A review of source to insure it is unmodified, contains no embedded or malicious code, contains no security vulnerabilities, and functionality to determine testing requirement.

Full Source Code Review: A review for full compliance to the 2005 EAC VVSG. This includes all standards dealing with formatting in Volume I Section 5.2.2 – 5.2.7 and Coding Conventions in Volume II Section 5.4.2 as well as any other standards violations.

Types of Reviews: There will be two independent source code reviews performed.

- Security Review: The security review will implement code inspection and code examination during the scope of this project. It will apply to all source code including third party source code and COTS code that is supplied for compilation or generated by another product.
- Functional Review: The functional review will implement a full source code review to all source code supplied by the vendor. The functional review will also implement code inspections and code examination as applied in this document for third –party software and COTS products.

Volume I

VVSG Requirement 5.1.1 Software Sources:

The requirements of this section apply generally to all software used in voting systems, including:

- *Software provided by the voting system vendor and its component suppliers*
- *Software furnished by an external provider (for example, providers of COTS operating systems and web browsers) where the software may be used in any way during voting system operation*
- *Software developed by the voting jurisdiction*

Compliance with the software requirements is assessed by several formal tests, including code examination. Unmodified software is not subject to code examination; however, source code provided by third parties and embedded in software modules for compilation or interpretation shall be provided in human readable form to the accredited test lab. The accredited test lab may inspect source code units to determine testing requirements or to verify that the code is unmodified and that the default configuration options have not been changed.

CIBER's Understanding of the Requirement:

The vendor must provide third party software modules that include source code for compilation to the ITA for their review. The ITA can inspect the source code to determine test requirements, examine it for modifications, or insure the source code is unmodified from its normal commercial version.

NYSTEC's Understanding of the Requirement:

The voting machine vendor must provide any and all source code that could potentially have been modified by the vendor prior to its use in constructing the computer programs that run on the voting machine. Software used by the vendor to construct the voting machine that was not in source code format may be exempt from the code examination requirement (see COTS interpretation below). Any source code created by the vendor, obtained from a third party, generated by a COTS product, or that is part of a COTS package however must be subjected to code examination as part of the certification process.

VVSG Requirement 5.2.3 Software Modularity and Programming

Voting system application software, including commercial off-the-shelf (COTS) software, shall be designed in a modular fashion. However, COTS software is not required to be inspected for compliance with this requirement. For the purpose of this requirement1, "modules" may be compiled or interpreted independently. Modules may also be nested. The modularity rules described here apply to the component sub-modules of a library. The principle to be followed is that the module contains all the elements to compile or interpret successfully and has limited access to data in other modules. The design concept is simple replacement with another module whose interfaces match the original module.

CIBER's Understanding of the Requirement:

COTS software must be designed in a modular fashion. The ITA is not required to inspect COTS software for modularity.

Volume II

VVSG Requirement 1.3.1.3 Focus of Software Evaluation

The software tests encompass a number of interrelated examinations, involving assessment of application source code for its compliance with the requirements spelled out in Volume I, Section 5. Essentially, the accredited test lab will look at programming completeness, consistency, correctness, modifiability, structure, and traceability, along with its modularity and construction. The code inspection will be followed by a series of functional tests to verify the proper performance of all system functions controlled by the software.

The accredited test lab may inspect COTS generated software source code in the preparation of test plans and conduct some minimal scanning or sampling to check for embedded code or unauthorized changes. Otherwise, the COTS source code is not subject to the full code review and testing. For purposes of code analysis, the COTS units shall be treated as unexpanded macros.

CIBER's Understanding of the Requirement:

A COTS product that provides source code for compilation can be subjected to an inspection, but it is not subject to a full code review. If the COTS product generates source code, then a sample of the generated code will be provided to the ITA for minimal inspection, consisting of an examination for embedded code or modifications.

All other code is reviewed according to the full scope of the VVSG requirements.

VVSG Requirement 1.7.1 General Applicability

Voting system hardware, software, communications and documentation are examined and tested to determine suitability for elections use. Examination and testing addresses the broad range of system functionality and components, including system functionality for pre-voting, voting, and post-voting functions. All products custom designed for election use shall be tested in accordance with the applicable procedures contained in this section. COTS hardware, system software and communications components with proven performance in commercial applications other than elections, however, are exempted from certain portions of the test as long as such products are not modified for use in a voting system.

Compatibility of these products with other components of the voting system shall be determined through functional tests integrating these products with the remainder of the system.

CIBER's Understanding of the Requirement:
Unmodified COTS products with a proven performance record in commercial applications outside of the voting industry are exempted from a full source code review. If the source code for these products is provided for compilation, then a functional code inspection of the modification will be performed to determine the compatibility of these products with the voting system.

NYSTEC's Understanding of the Requirement:
A thorough examination of software that is classified as COTS by vendors must take place. This examination will determine if vendors are classifying software as COTS as an attempt to exempt it from the source code review requirement. Software, hardware and firmware classified as COTS by the vendor must be available in the commercial world as a product and not be created by one vendor for another to be used in the handeling of votes on a voting machine. This problem is referred to as the "COTS loophole" in the VVSG regulations and NYS must ensure components are classified appropriately.

VVSG Requirement 5.2 Basis of Software Testing (by section)
The accredited test lab shall design and perform procedures that test the voting system software requirements identified in Volume I, Section 5. All software components designed or modified for election use shall be tested in accordance with the applicable procedures contained in this section.

Unmodified, general purpose COTS non-voting software (e.g., operating systems, programming language compilers, data base management systems, and Web browsers) is not subject to the detailed examinations specified in this section. However, the accredited test lab shall examine such software to confirm the specific version of software being used against the design specification to confirm that the software has not been modified. Portions of COTS software that have been modified by the vendor in any manner are subject to review.

CIBER's Understanding of the Requirement:
Unmodified, general purpose COTS products will be examined to confirm the specific version against the design specification. If COTS products have been modified in any manner, the modification is subject to a full source code review.

VVSG Requirement 5.2 Basis of Software Testing (by section)
Unmodified COTS software is not subject to code examination. However, source code generated by a COTS package and embedded in software modules for compilation or interpretation shall be provided in human readable form to the accredited test lab. The accredited test lab may inspect COTS source code units to determine testing requirements or to verify the code is unmodified.

CIBER's Understanding of the Requirement:
COTS products that generate source code for compilation are not subject to review. The source code generated by the product must be submitted to the ITA for their inspection.

The ITA may inspect the source code to determine testing requirements or to verify the code is unmodified.

VVSG Requirement 5.2 Basis of Software Testing (by section)
The accredited test lab may inspect the COTS generated software source code in preparation of test plans and to provide some minimal scanning or sampling to check for embedded code or unauthorized changes. Otherwise, the COTS source code is not subject to the full code review and testing. For purposes of code analysis, the COTS units shall be treated as unexpanded macros.

CIBER's Understanding of the Requirement:
A COTS product that provides source code for compilation can be subjected to an inspection, but it is not subject to a full code review. If the COTS product generates source code, then a sample of the generated code will be provided to the ITA for their minimal inspection, consisting of an examination for embedded code or modifications.