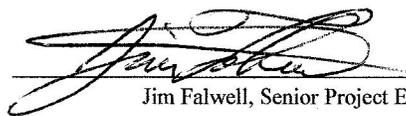




7800 Highway 20 West
Huntsville, Alabama 35806
Phone (256) 837-4411
Fax (256) 721-0144
www.wyle.com

**SOURCE CODE REVIEW
STATUS REPORT
FOR
ES&S EVS 5.0.0.1 NY**

Prepared by:

 8/31/12
Jim Falwell, Senior Project Engineer



NVLAP LAB CODE 2007-14

COPYRIGHT BY WYLE. THE RIGHT TO REPRODUCE, COPY, EXHIBIT, OR OTHERWISE UTILIZE ANY OF THE MATERIAL CONTAINED HEREIN WITHOUT THE EXPRESS PRIOR PERMISSION OF WYLE IS PROHIBITED. THE ACCEPTANCE OF A PURCHASE ORDER IN CONNECTION WITH THE MATERIAL CONTAINED HEREIN SHALL BE EQUIVALENT TO EXPRESS PRIOR PERMISSION. WYLE SHALL HAVE NO LIABILITY FOR DAMAGES OF ANY KIND TO PERSON OR PROPERTY, INCLUDING SPECIAL CONSEQUENTIAL DAMAGES, RESULTING FROM WYLE'S PROVIDING THE SERVICES COVERED BY THIS REPORT.

U.S. Electrical Insulation Co. (Eaton)

VSTL

EAC Lab Code 0704

TABLE OF CONTENTS

	<u>Page No.</u>
1.0 INTRODUCTION	3
2.0 DESCRIPTION	3
3.0 TEST PROCEDURE	5
4.0 SOURCE CODE REVIEW STATUS	6
5.0 CONCLUSION	6

1.0 INTRODUCTION

This review covers the ES&S EVS 5.0.0.1 voting system firmware and software received by Wyle as part of the Election Assistance Commission (EAC) Certification Test Campaign. The review involved evaluation of the software's compliance with the Election Assistance Commission 2005 Voluntary Voting System Guidelines (VVSG) and compliance to the ES&S supplied coding standard.

This evaluation included, but was not limited to, the following considerations:

Readability	How straightforward and apparent was the design?
Understandability	How complicated was the code to implement it?
Modularity	How well was the code divided into logical, functional units?
Robustness	How well does the code handle error conditions or unexpected inputs?
Security	Does the code protect the integrity of voting data at all times?
Maintainability	How easy would it be to extend, fix, or modify this code in the future?
Consistency	Was the design of the code coherent throughout?
Documentation	Does the code contain useful and frequent comments?
Usability	Does the code inform the user about progress or errors?
Flow control	Are control constructs and entry/exit points logical and controlled?

2.0 DESCRIPTION

The ES&S EVS 5.0.0.1 software and firmware package consists of multiple components and subcomponents. The major components can be broken into two main areas: election management system (EMS) and software and firmware that operate. Below is a list of major packages:

EMS

- ElectionWare (eWare)
- Election Reporting Manager (ERM)
- ES&S Event Log Service(EventLog)
- Removable Media Service
- VAT Preview

Devices

- Digital Scan 200 (DS200)
- Digital Scan 850 (DS850)
- AutoMark Voter Assist Terminal (VAT)

ElectionWare is comprised of the eWare source code package written in the Java programming language and three other subcomponents written in C++. The ElectionWare component also contains a source code package for the ElectionWare SQL that is used to create the default database including the stored procedures, triggers, and indexes. The three subcomponents are:

- PaperBallot
- libCoNG.dll
- EssXML

ERM is written in the COBOL programming language and is comprised of the ERM source code package and the ERM Utilities subcomponent package. ERM Utilities is a combination of 11 smaller components. Below is a list of those components and the language in which they were written:

- CB_EVT.dll – C/C++
- CB_XML.dll – C/C++
- CB_XMLConv.dll – C/C++
- ERMXMLConvDLL.dll – C/C++
- ERMXMLData.dll – C/C++
- ExitWin.exe – VB
- MyDLL.dll – C
- RegUtil.dll – C
- Shell.exe – C++
- ShellSetup.exe – C++
- RSACrypto.exe – C++

EventLog is written in the C++ programming language and is comprised of six components. All six components are packaged together to create EventLog. These six components are:

- CreateLog
- EssEvt
- EssEvtA
- EssEvtMsg
- EvtSvc
- LogEvent.

Removable Media Service is written in the C++ programming language and is comprised of three components. All three components are packaged together to create Removable Media Service. The three components are:

- RmuCli.exe
- RmuDLL.dll
- RmuSvc.exe

DS200 is the combination of four subcomponents and the ESSLinux OS. The four components and the language they are written in are listed below:

- DS200 CoNG, Image, HAL – C
- DS200 Presentation Layer – Java
- PowerManagement_Msp430 – C
- Scanner_C8051 – C

DS850 is written in the C/C++ programming languages and is comprised of three components and the ESSLinux OS. The three components are:

- UI
- MCP
- CoNG

AutoMark VAT is the combination of 20 components. These 20 components are packaged together to create the AutoMark VAT software. Below is a list of components and the language they are written in:

- Amcode.exe - C++
- AutoMARK.exe - VB
- AutoMark.dll - C#
- AutomarkDataHelperLibrary.dll - C++
- AutomarkEncoder.dll - C++
- AutomarkService.exe - C++
- AutomarkStartup.exe - C++
- DiagnosticLogger.dll - C++
- GETMARKS.dll - C++
- MAKEBIN.EXE - C++
- NonVolatileLibrary.dll - C++
- OperationLogger.dll - C++
- PEB.hex – C/Assembler
- RSASecurityLibrary.dll - C++
- scandriver.dll - C++
- SCANNER.BIN – C/Assembler
- ScannerPrinterLibrary.dll - C++
- SecurityLibrary.dll - C++
- SIB.hex – C/Assembler
- Ultra.s19 – C/Assembler

3.0 TEST PROCEDURE

Wyle Laboratories will compare the source code to the manufacturer's software design documentation to ascertain how completely the software conforms to the manufacturer's specifications. Source code inspection shall also assess the extent to which the code adheres to the requirements in Volume I, Section 5 of the 2005 VVSG.

1. Wyle will ensure that the source code submitted is sufficient to enable the reviewer or tester to review the source code and design and conduct tests of the software structure. Wyle will ensure that the source code is cataloged by date, manufacturer, application, and version in a secure folder on the Wyle server.
2. Wyle will perform a review of the code line by line, noting each observed noncompliance issue. The tools used are a file comparison program and a text editor. Wyle will record in the code review database all observations of violations of the specified standards.
3. If the manufacturer submits the source code with change notes as a re-review, Wyle compares each new file to its previous version to confirm that the actual changes in the file are as identified in the change log and are in compliance with the standards.
4. Wyle will repeat the above steps until all observed standard violations have been resolved.
5. Wyle will submit a final summary report that indicates the Source Code Review is completed and states a professional opinion on the code submitted.

4.0 SOURCE CODE REVIEW STATUS

Below is a table describing the status of the major components for the ES&S EVS 5.0.0.1 software and firmware source code reviews. If a major component is listed as compliant, then all subcomponents are considered compliant. The Internal Version is the ES&S internal versioning for the source code that was submitted for source code review. The Build Version is the final release version for the source code.

Major Component	Compliant	Internal Version	Build Version
ElectionWare (eWare)	Y	4.1.0.1o	4.1.0.1
Election Reporting Manager (ERM)	Y	8.6.0.2c	8.6.0.2
ES&S Event Log Service(LogEvent)	Y	1.5.0.1a	1.5.0.1
Removable Media Service	Y	1.4.0.1b	1.4.0.1
Digital Scan 200 (DS200)	Y	2.7.0.1l	2.7.0.1
Digital Scan 850 (DS850)	Y	2.4.0.1f	2.4.0.1
AutoMark Voter Assist Terminal (VAT)	Y	1.8.1.1a	1.8.1.1
VAT Preview	N/A	1.8.1.1a	1.8.1.1

5.0 CONCLUSION

Overall this code is written in a manner that is readable, easily understood, and can be maintained. The code is well documented and is consistent in design and flow control. The issues identified during this code review are typical of any code review. All issues will be addressed to bring this submission into compliance with the VVSG 2005 standards. Some areas that contained a large number of findings are listed below:

- i. Module Revision History
 - a. EAC 2005 VVSG Volume 1 Section 5.2.7 requires every source module to contain a header. The header must have a purpose describing how the unit works, other units called and the calling sequence, a description of input and output parameters, file references by name and method of access, all global variables used in the unit, and a date of creation and revision history.
 - b. ES&S used a code generating software tool for a portion of the source code files. This tool re-generates the code each time. The revision history is not required for this portion of the code. This tool also allows for a "Preserve" section. This section is used to reserve code that has been placed into the generated code by developers. The revision history for this section was tracked at the file level, but not at the module level.
 - ii. Variable Comments
 - a. Upon a variable's declaration, a variable comment is required to describe how that variable is used in the source code and to explain any limitations inherent in the variable's use – simply echoing a variable's name in a comment that precedes the variable's declaration is not a sufficient description of the variable.
 - b. A variable comment was not provided for preprocessor directives. These are considered objects and require a comment where they are declared.
 - c. A variable comment should describe the declared variable rather than explaining what the overall module is doing, which is the job of an in-line comment.
-

- iii. Units Called in Header
 - a. All modules require a header that contains other units and the calling sequence.
 - b. ES&S supplies the units called and what looks to be a “first use” calling sequence. If a unit is called, the sequence must contain each unit as called. If a unit is called multiple times in sequence, the unit may be listed only once.
- iv. Inconsistent Indenting
 - a. The source code is to be indented consistently and clearly to indicate logical levels.
 - b. Some source files had the first line following the opening bracket left justified flush with the left hand side of the page.
- v. In-line Comments
 - a. The comment should provide description and clarity of what the source code is doing, not an explanation of its implementation – unless for some reason the implementation is not clear or straight-forward and needs additional explanation.
 - b. Many problems identified with in-line comments had to do with the comment being too cryptic or lacking in detail.