Source Code Review Summary

Process

Source code review was performed on the Hart Verity 2.7 source code. Verity 2.7 was reviewed as a modification of the Verity 2.6 system. Review was performed in two parts, Automated and Manual. Automated review was performed using the Understand tool to produce results that were reviewed as part of the Manual review. Manual review was performed to validate all findings from the Understand tool, as well as to cover all requirements that Understand is not capable of covering. For the manual review only modified code was reviewed. Unmodified code that has already been approved is considered compliant and does not require review.

Standards

Source code review utilizes the VVSG 1.0 along with all vendor declared standards that meet the requirements from the VVSG Vol 1 5.2.6. For this review, Hart identified the following coding standard that was used during the creation of their product:

- Microsoft All-in-One Code Framework Coding Standards

The above listed coding standards, meets the requirements from the VVSG Vol 1 5.2.6 that allow for the vendor declared standard to supersede portions of the VVSG.

Code Count

The source code for the Verity 2.7 review effort, contains the component(s) listed in the table below which shows the languages used for each component along with a line count of each and the associated standard used for review.:

Software Component	Language(s)	Line Count(s)	Standard(s)
Verity 2.7	C#, and C++	1,598,143	Microsoft All-in-One Code Framework Coding Standards

Automated Review

The listed code was first scanned using the Understand tool for the following requirements. The following table contains the full set of results for the checks used and a count of all issues returned by the Understand tool. Only requirements that had at least one issue returned are listed. For the full set of checks run, see the project specific configuration file named Verity 2.6_CodecheckConfiguration.ini.

Understand CodeCheck	Number of Findings
MISRA-C++ 2008 17-0-1 Reserved identifiers, macros and functions in the standard library shall not be defined, redefined or undefined	29
MISRA-C++ 2008 6-6-5 A function shall have a single point of exit at the end of the function	21

MISRA-C++ 2008 7-1-1 A variable which is not modified shall be const qualified	208
SciTools' Recommended Checks Functions Too Long - RECOMMENDED_04 -- Program units should not have more than the specified number of lines	45
SciTools' Recommended Checks Magic Numbers - RECOMMENDED_08 -- All fixed values will be defined constants.	523
SciTools' Recommended Checks Unreachable Code - RECOMMENDED_12 -- Source will not contain Unreachable Code	9

All findings returned by the Understand tool were reviewed using manual review to validate accuracy.

Manual Review

Manual review was performed over the modified source code to verify compliance with the VVSG and applicable Vendor Declared Standards. All findings returned by the Understand tool, as well as any requirements not able to be covered by Understand, were reviewed during this process. The following table lists all requirements covered during the manual review and the number of findings found to be in violation.

VVSG/Vendor Standard Requirement	Number of Findings
VVSG v.1: 5.2.2 Self-modifying code	0
VVSG v.1: 5.2.3.a Specific function	0
VVSG v.1: 5.2.3.b Module has unique name	0
Hart Verity Coding Std sec.: 3.2 Hart Naming Conventions	0
VVSG v.1: 5.2.3.b, 5.2.7.a Module has header	0
VVSG v.1: 5.2.3.c Required resources	0
VVSG v.1: 5.2.3.e Single Entry Point	0
VVSG v.1: 5.2.3.e Single Exit Point	0
VVSG v.1: 5.2.3.f Control structures	0
MS Std sec.: 2.2 Do not use tabs	0

Verity Std sec.: 2.1 Line length	0
MS Std sec.: 2.6 Local variables have minimum scope	0
MS Std sec.: 2.6 Local variable declaration and initialization	0
Verity Std sec.: 3.8.4 Initialize pointer variables	0
MS Std sec.: 2.6 Parameters ordered in groups	0
MS Std sec.: 2.7 One statement per line	0
MS Std sec.: 2.8 Use enums	0
MS Std sec.: 2.8.1 Flag enums	0
MS Std sec.: 2.10 Braces and indentation	0
Verity Std sec.: 3.7.3 Braces around single line conditionals	0
MS Std sec.: 2.11.2 File has header comments	0
MS Std sec.: 2.11.6 TODO comments	0
MS Std sec.: 4.2 File named for single contained public type	0
MS Std sec.: 4.4.1 Meaningful names	0
MS Std sec.: 4.4.3 No Hungarian notation	0
MS Std sec.: 4.5 Constant fields	0
MS Std sec.: 4.6 String operations	0
MS Std sec.: 4.7 Array and Collection operations	0
MS Std sec.: 4.8 Value types implement IEquatable<T>	0
MS Std sec.: 4.9.1 Class instance fields are private and exposed through properties	0
MS Std sec.: 4.9.2 Set-only properties are not allowed	0
MS Std sec.: 4.9.3	0

Do not call virtual members on an object inside its constructors	
MS Std sec.: 4.9.4 Out parameters follow all of the pass-by-value and ref parameters	0
MS Std sec.: 4.9.4 Validate arguments to public, protected, or explicit members	0
MS Std sec.: 4.9.6 Member overloading	0
MS Std sec.: 4.9.10 Abstract types' constructors	0
MS Std sec.: 4.11.1 Throw specific exceptions. Do not return error codes.	0
MS Std sec.: 4.11.2 Catch only specific errors	0
MS Std sec.: 4.12 Do not force garbage collections	0
MS Std sec.: 4.12.1 Do not use catch blocks for cleanup code.	0
MS Std sec.: 4.12.2 Basic Dispose Pattern	0
MS Std sec.: 4.12.3 Types finalizable	0

Results

For this review, the following table lists each requirement covered during both the Automated and Manual review process, that had a discrepancy cited against it. After manual review, any instance of noncompliance with one of the applicable standards or VVSG requirements, is documented as a Discrepancy and added to the report. All reports are sent to the Vendor and any discrepancy present in the report must be addressed before the code is accepted as compliant.

Standard	Requirement	# Findings	# Discrepancies	Comment
MISRA-C++ 2008 17-0-1	Reserved identifiers, macros and functions in the standard library shall not be defined, redefined or undefined	29	0	Code in question is used appropriately to qualify data before use.

MISRA-C++ 2008 6-6-5	A function shall have a single point of exit at the end of the function	21	0	C languages automatically exit at the closing brace.
MISRA-C++ 2008 7-1-1	A variable which is not modified shall be const qualified	208	0	Variables in question are function input variables.
SciTools' Recommended Checks Functions Too Long - RECOMMENDED_04	Program units should not have more than the specified number of lines	45	0	Line Count Requirement does not include comment/blank lines.
SciTools' Recommended Checks Magic Numbers - RECOMMENDED_08	All fixed values will be defined constants.	523	0	Variable initialization or assignment is not a violation.
SciTools' Recommended Checks Unreachable Code - RECOMMENDED_12	Source will not contain Unreachable Code	9	0	Unreachable code found to be defensive, which is allowed.

Summary

For this review, there were a total of 835 findings. Of these, none were found to be in violation of at least one requirement. As a result, no issues were reported and zero remain open. As no discrepancies were found in the Verity 2.7 source code, no remediation is required.