Frequently Asked Questions About SNMP Testing

Q We test our SNMP agent with a popular NMS and it works just fine. Why would we need SilverCreek, the official SNMP Test Suite?

A Interacting with a manager station does not prove correct implementation of the SNMP protocol. There are still issues such as lexicographic ordering, 32-bit signed integer arithmetic, read-only and read-write variables correctly implemented, checking that the agent implements the objects it claims to, and does not implement objects it does not claim to, and other areas that implementors tend to get wrong. These implementation errors will result in the reporting of wrong data to the manager.

Doing "gets" and "sets" from a Network Management Station to your agent is an indicator that some of your implementation is correct. However, you have no way of determining that the "get" retrieved the right data and that the "set" set the correct variable.

An NMS is a great mechanism for cheating at agent testing! A Network Management Systems goes out of its way to work around badly implemented agents. If your agent works with an NMS that proves that the NMS is very good; it proves nothing about your agent.Agent developers forget that Fortune 2000 companies often write their own applications for collecting and analyzing SNMP agent information. In such environments, these companies typically do not add a lot of padding or scaffolding to work around bad agent errors. A common complaint of end user network managers is that the information they collect from SNMP agents is unreliable. Two products from different vendors that should be reporting identical information often report different results.We must also remind developers that all releases and improvements need to be tested. Using HP Open View for this purpose requires an SNMP expert employee or contractor who can interpret the results to know what is correct and what is not. Using SilverCreek will find the majority of your implementation errors and provide definitive test result information so that you don't need an SNMP expert to handle testing.

InterWorking Labs®

Kings Village Center #66190
Scotts Valley, CA 95067
iwl.com
+1.831.460.7010
info@iwl.com

Q We already did interoperability testing at the (Fill in Name of) Tradeshow, so why would we need your test product?

A Interoperability testing is a great way to check against another engineer's interpretation of the specification. This can be very helpful in checking your own thinking about the specification and areas of potential misinterpretation. This can be a valuable learning experience for all implementors. We have seen situations where seven engineers implemented something one way, and six implemented it another way. A meeting was held with the RFC authors and the paragraphs were rewritten to eliminate the ambiguity. This is an extremely useful experience, but not sufficient for testing a product.

Interoperability testing does not determine your product's compliance with boundary conditions, ability to handle various error conditions, correct implementation of the SNMP protocol, including lexicographic ordering, 32-bit signed integer arithmetic, read-only and read-write variables implemented correctly, and other areas that implementors tend to get wrong.

Q We implemented our SNMP agent from the RFCs (coded it from scratch), so why would we need SilverCreek?

A The RFCs are not robust specifications free from ambiguity, so it is likely that aspects of the implementation will be incorrect. There may be email archives, minutes from meetings, or papers that clarify these issues, but unless one knows what to look for, the required information may be missed. In fact, it is most likely that products based on commercial SNMP engines will be less error-prone than "from-scratch" implementations. The commercial SNMP engine developers have full time staff devoted to understanding the clarifications and subtleties as well as attending the meetings where these issues are discussed.

In addition, there is no "gold" implementation with IETF RFCs. When a single company dominates a standard, an implementor can test and check his product against that "gold" implementation standard (such as Windows95 from Microsoft). Within the IETF standards-creating process, there are experimental implementations, but no "gold" implementation to test against.

Our experience, and the experience of countless SNMP developers, is that there are many parts of SNMP that are _hard_ to get right. Lexicographic ordering, for instance, seems like a fairly simple thing to get right -- but in reality, most developers do not. They may have OID's out of order or wrap around at the end to the beginning of the OID tree. Sometimes you cannot even walk the MIB because GET-NEXT's will not work from an arbitrary starting point.

Q We only need to test our private MIBs, so why would we buy SilverCreek?

A SilverCreek is extensible. Through simple scripts written in Tcl, you can write tests for your proprietary MIBs. It is a very simple process to load a private MIB into SilverCreek. Once it is loaded, SilverCreek can run a number of the standard tests (such as writing to read-only variables) on the objects defined in your MIB. From there you can extend SilverCreek to check the specific functionality of your MIB.

Q There are some publicly available test tools, so why should we use SilverCreek?

A We have evaluated a number of the publicly available test tools, and determined them to be inadequate. Many of these tests report false information. Many others are not rigorous enough.

Many of the publicly available tools are extremely difficult to install, requiring as much as five hours. There is no technical support, bug fixes, or continuing efforts in place to improve these tests.Many of the other testing products available are not extensible; there is no means to test private MIBs, or customize the tests to fit your environment.

Q Since we beta tested our product when it first came out and got rid of all the bugs, why would we need to retest with your test product now?

A Beta testers tend to be users who will find the obvious user level bugs or bugs that only appear in user's equipment configuration environment that is different from the manufacturer's test environment. While this effort can provide good information on usability of the product, and glitches in various combinations of equipment, it does not provide thorough product testing.

Beta testing will not determine your product's compliance with boundary conditions, ability to handle various error conditions, correct implementation of the SNMP protocol, including lexicographic ordering, 32-bit signed integer arithmetic, read-only and read-write variables implemented correctly, and other areas that implementors tend to get wrong.

Q We purchased our SNMP engine from a commercial organization specializing in SNMP engines so obviously they did all this testing and their code will be correct, so why would we need your test product?

A It is most likely that products based on commercial SNMP engines will be less error-prone than "from-scratch" implementations. The commercial SNMP engine providers usually have full time staff devoted to understanding the RFC clarifications and subtleties as well as attending the IETF meetings where these issues are discussed. (In fact, we often see them there!)

However, there is no guarantee that commercial SNMP engine providers will have implemented everything correctly. Getting your agent code from a third party is a very good reason to perform your own testing in order to evaluate the quality of the product delivered to you. We have, sadly, tested products based on third party agents that could not pass any of our tests.

Want to know more about SNMP Testing?

Kings Village Center #66190
Scotts Valley, CA 95067
iwl.com
+1.831.460.7010
info@iwl.com