

Magic Decoder Ring

Reining in the Chaos of Software Signatures

As a Software Asset Manager, you are supposed to control, standardize and optimize the software used to conduct the business of your employer. You know that the foundation of software management has to be built on a reliable and accurate discovery that can be associated with license documents. Why must it be so difficult to do? Did someone forget to give you a magic decoder ring that converts the line items on a discovery run into software titles?

Today there are thousands of software providers producing thousands of products and hundreds of thousands of versions of those products. The lack of standards on the development of products makes choosing signatures for these products an art, not a science. The signature, the unique descriptor for a version of a particular application that allows both patch management and license management to be performed, is built and reported by discovery and software delivery products. The process for developing the "correct" signature often seems like an arbitrary process, unlike the data collection aspect where obvious metrics exist. There are actually thousands of alternative fingerprints for the software to choose from, all of whom may look good to the inexperienced person.

The complexity of software signature development is a direct result of the changes over time in how software is designed and developed, the manner in which the product is marketed and sold, and the pricing scenarios for the product. These factors introduce new variations in how software can be recognized

and will continue to do so for the foreseeable future. Your investment in a first-rate discovery product remains a necessary step. In order to understand the difference between discovery products, an insider's view of the challenges inherent to the Software Identification Process will help you separate the good discovery tools from the merely adequate.

The vendor naming convention challenge

There is no single consistent naming convention in the software industry. An individual vendor can use a different naming convention between their own products or even between versions. Sources for signature development such as marketing documents, licensing data and commonly used names also introduce variations such as Microsoft, Microsoft Inc., Microsoft Corporation, and MSFT. Careful analysis can still leave multiple options that are equally good naming options for the same application. Your ITAM solution provider should offer you a single set of consistent naming conventions that are accepted by their customers. You should demand consistent and rigorous adherence to the rules to avoid creating duplicates on all levels. An understanding of the specific challenges that impact the effectiveness of software signatures will help you evaluate between discovery products that may seem similar, but are not.

The product and version definition challenge

What exactly is the product name and what is the version? Is "SQL Server 2000" the product name and "8.0 SP1" the specific version or is "SQL Server" the product and the version "2000 - 8.0 SP1"? Research is again difficult and may lead to conflicting results for the inexperienced person. A discovery product with a quality software identification process requires a rigorously trained and dedicated detection staff to research the right answers, reducing exceptions to the minimum.



The detection granularity challenge

This detection issue for granularity is to understand what level of specificity we need for a given application is dependent on what we are trying to do. For example, for licensing reconciliation we may only need the version to be “2000”, but for patch management we need to drill down to specific builds and patches. The discovery product must have variable levels of detection, from the highest resolution possible down to grouping versions as needed.

The signature uniqueness challenge

A single version of an application may contain thousands of files, registry keys, services and other elements. From this set of possible elements it is necessary to choose a globally unique subset so that the given version gets detected only when it is installed. If other applications or versions of the same application contain the same elements that we've chosen as a signature, the overlapping signatures can lead to overcounting installations and license chaos.

At first glance it may seem that the uniqueness of a signature could be achieved by adding all elements of the application (all files, registry keys etc), exactly as they are found in a sample installation. The blunt force approach would certainly produce unique signatures. Unfortunately, the use of all elements introduces the following problems:

- A single product and version can legitimately be configured differently during installation and usage, leading to different signatures
- The more elements that make up the signature, the more conditions the discovery product must check for every machine, the longer the processing time or the higher the hardware bill to ensure effective processing

Signature development is a balance between too little or too many elements, with too many elements leading to false separation of identical applications and excessive processing times. The need for up-to-date reports in a cost efficient way clearly pushes us away from the use of all elements. The right answer is to develop the minimal signature that is has enough elements to ensure uniqueness and no bigger.

The software market dynamics challenge

The passage of time introduces another challenge to the concept of a "correct signature." A signature that uniquely identifies an application and version can cease to be correct after the release of a new version of that application. The vendor may also add confusion by releasing a completely different application that includes elements of that previously created signature. The unwary individual may inadvertently create signatures that require frequent changes. At the least, signatures require constant review and rarely remain the static, separate entities that we would like them to be.

The good news is that there are signature practices that when applied reduce the risk of signature overlap. There are also techniques for detecting the need for modification, if unavoidable, and doing it in a way that minimizes impact on existing reports. A combination of automation and experience are essential characteristics that you should demand from your discovery product vendor.

Reining in the Chaos

Software signatures are an essential aspect of discovery. Discovery, the act of building an accurate and timely inventory of software and hardware, is a priority for any organization. After all, the quality of the discovered data impacts the ability to control and track, prevent theft, reduce risk, and maintain software compliance. The development of software signatures is a complex task that requires an experienced detection team using rigorous analysis as well as automation to supply new software signatures and maintain the existing signatures. Compromising on the effectiveness of the signatures delivered by your vendor will significantly lengthen the resource hours required for each of the business practices that depend on an accurate software inventory. The golden goal for any IT asset management program is to deliver savings, whether from eliminating purchases or reducing the resources to manage the assets. Software Asset Management programs may fail to deliver those savings without the advantage of accurate and usable software signatures.

Artur Kornatowski is the IT Asset Detection Manager at Eracent, Inc. and Jenny Schuchert is the Vice President of Marketing at Eracent, Inc.