



At the Starting Gate

How to Pick a Discovery Tool

When you put together a requirements list for a discovery tool, one of the first things you need to consider is the platform on which you want to install the product, and the platform of the devices about which you want to discover details. Not all discovery tools will find things on all platforms, and the level of detail they can discover is also variable. Most vendors in the market today will do more or less the same job with regards to hardware details for discovered assets. They tend to vary in two distinct areas:

- How “complete” their discovery is of all assets on a given network; and
- Their method for discovering software details on the assets discovered.

The completeness of the discovery is important in that the accuracy and utility of the information a discovery tool can provide is only useful if it is a true reflection of your environment. If the tool doesn’t see a segment of your operating environment, and you assume it is, then your data will all be skewed.

Some available tools provide a multipronged approach to this problem by using multiple mechanisms for agent delivery. Others enhance the amount discovered by having another tool report back on all nodes in a given network segment. Regardless of what they are, if it has an IP address assigned it will show up. Obviously, the latter approach is closer to the ideal in that it doesn’t rely on human fallibility to ensure full and complete distribution of agents.

The second major point to investigate is how the software details are discovered and reported. What is supplying the software install data, and what is being done with that data to turn it into useful information? Most enterprise-level tools today do not rely solely on any one method of discovery for software, as none of them are wholly reliable and accurate by themselves.

The difference is in what they do to turn data into information. Do they employ some sort of “fingerprinting” mechanism to differentiate various versions of a given product? How often is the tool updated with new software product information? Does it identify patches and bug fixes in addition to major and minor upgrades to a product? How does it manage product suites?

Once the product has software information, what does it allow you to do with that? Can you get usage information as well? What is the level of granularity? How does it tie to licensing? How “automated” can license management become using the product?

NOW WHAT?

Once you have determined your platform considerations, you need to document what you intend the tool to accomplish for you. There are many reasons why an organization might want



or need a discovery tool, and these reasons may conflict with one another when trying to identify the most appropriate tool to use. For example, Tool X does a great job at controlling the scope of the discovery by ensuring that it sees all hardware in the production environment. Product Y doesn’t do quite so good a job at this as X, but has a much more comprehensive software package with more frequent updates from the vendor. If your goal emphasizes hardware management, tool X is the better tool. If your goal emphasizes software management, tool Y is the better tool. Both will discover hardware and software, and both do an adequate job within the bounds of certain limitation. Which to choose, however, depends on how well-defined the goals of use are.

I like to think of discovery tool selection as similar to buying a new car. All cars on the market will get you from point A to point B. Which car you choose gets down to a personal choice as to how you want to travel that distance—fast, economically or luxuriously—and how to stay within your budget. These characteristics are somewhat exclusive, in that you can often get two or three of your choices, but rarely does a car come along that has all of them.

With discovery tools, they will all get you data on the hardware and software in your environment. In the end, they’ll differ by how complete, comprehensive and usable the data is.

Daniel Lamb is a senior solutions architect for TechTrack Solutions Inc.