



## **About the Express Software Identification Database (ESID)<sup>®</sup>**

The Express Software Identification Database (ESID) is a comprehensive catalog of commercial and free PC software applications that run on Windows desktops and servers. A key component of Express Metrics's own IT asset management solutions, the ESID provides a means for comprehensively and accurately identifying the wide range of software installed and used across corporate networks. The ESID and its associated recognition algorithms can be licensed for use within other products that require precise application identification to perform their functions. Companies that OEM the ESID include Dell, BMC, BigFix, New Boundary, LANDesk, Aldon, Tivoli, and more.

### **The Software Identification Challenge**

Any software product that needs to access, interact with, or report on installed desktop applications requires a method of 1) discovering and 2) recognizing those applications. Without strong application discovery and recognition capabilities, software data cannot be properly rationalized or presented in a precise, unified, or truly meaningful manner.

Ascertaining the presence of applications on a PC by collecting information about application executables is a fairly straightforward process from a technology standpoint. However, there exists no common method or technology that will correctly *identify* all the executables by associating them with actual titles of installed applications. In fact, most products that claim to discover and identify applications in an automated fashion rarely deliver reliable, complete information because they either over- or under-estimate the actual number of installed titles.

Most products rely on one of two methodologies: file header identification and registry analysis. Due to the inherent lack of uniformity within the computing environment, the use of any one of these methods in isolation is accurate and/or incomplete.

#### **File Header Identification**

File header identification is one commonly used technique for identifying applications. The advantage of this approach is that it's directly tied to the application executable, so it's unlikely that an inventory using this method would be incomplete. One issue with file header analysis is many software vendors don't update file header information on a regular basis, leading to inaccurate, incomplete,

or even missing application data. For example, the file header for Google's Chrome browser reveals its name and copyright, but no version information at all. Furthermore, because many applications consist of one primary executable and multiple (sometimes hundreds) secondary files, the secondary files are often interpreted as independent applications, resulting in an artificially inflated application count that could easily lead one to conclude that too few licenses have been purchased than are actually required. Finally, collecting file header information on a workstation requires each file to be opened and read by the inventory tool, a time-consuming process and one which, if not written carefully, can negatively impact workstation performance.

### **Registry Analysis**

Many technologies rely on information contained in the Microsoft registry to determine what's installed on a PC. However, there are some downsides associated with this methodology: First, programs installed using means other than Windows Installer are often not detected; and if they are, they frequently lack critical version information. Furthermore, in certain circumstances, applications are erroneously displayed even after being un-installed. Because the information obtained through registry analysis is generally identical to that which is found in Microsoft Add/Remove Programs, a manual examination of Add/Remove Programs usually results in the erroneous conclusion that this method is accurate—a potentially risky assumption.

### **Software Identification Database**

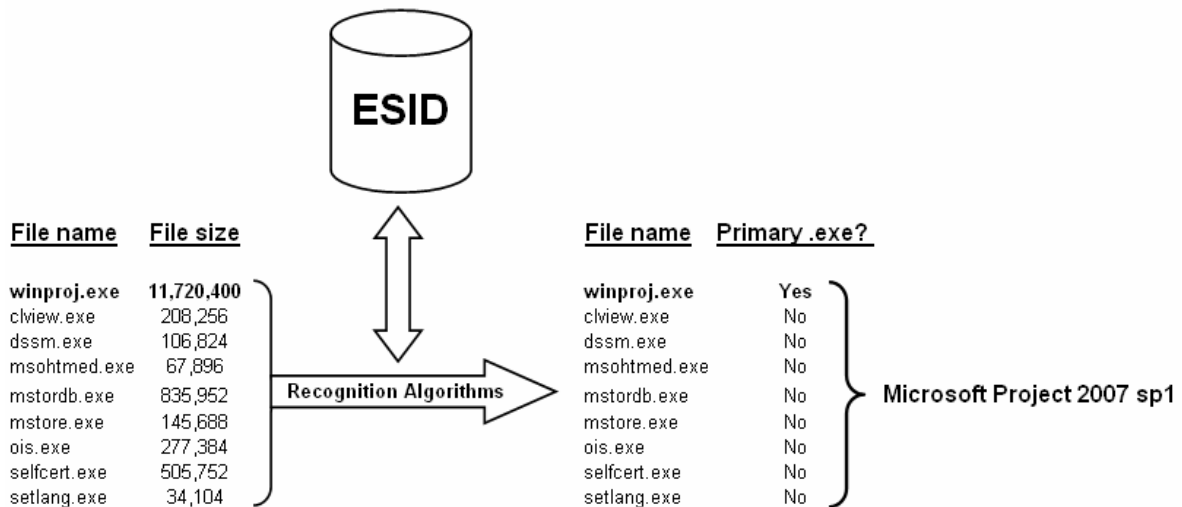
An alternative to the identification methods described above is to utilize a software database containing relevant file data. With this method, special algorithms must be developed to allow discovered executables and other information to be compared with application data residing in the database. The end result is a list of software titles corresponding to the discovered data. Assuming these algorithms are properly constructed, the success of the outcome depends on the **accuracy and comprehensiveness** of the software database. That is where the Express Software Identification Database comes in.

### **The ESID Solution**

The Express Software Identification Database (ESID) eliminates the need to rely solely on file headers and inaccurate or incomplete registry entries to identify applications within the IT environment. The database contains an extensive collection of application titles and their associated details such as version, software category, executable file names, file sizes, and more. Using this approach, collected file data can be correctly matched with the appropriate application titles—a prerequisite for using the information in a meaningful way. And because application data has been manually reviewed, researched, and verified by a team of dedicated staff, the ESID is the most accurate, consistent and **complete** approach to software

identification. By licensing the ESID, your product’s capabilities and reporting will be more accurate, more thorough, and more valuable to your customers. Compiled over a 13-year period, the ESID is updated monthly to include the latest applications, resulting in the most thorough software recognition available in the industry.

The following example shows how the ESID can be used to map discovered executables to their corresponding applications:



By comparing the file names *and* byte sizes of discovered executables to the data catalogued in the ESID, executables can be matched with the exact application and version to which they correspond. Note in the above example that not only does the ESID contain information identifying **winproj.exe** as the main executable of Microsoft Project 2007 sp1, but the ESID also catalogs all the associated secondary files (clview.exe, dssm.exe, and so on), so they can also be properly accounted for.

The ESID also includes Globally Unique Identifiers (GUIDs) for the purpose of identifying subtle distinctions among applications that can’t be properly identified using other methods.

Once applications have been properly identified using the methods described above, the data can be “rationalized” using algorithms that organize information contained in database tables into meaningful, *usable* data. The following are examples of tasks that would be impossible to accomplish without the ESID or a similar solution:

- **Correlate secondary files with their parent applications**

As mentioned above, most applications have more than one executable associated with them: a primary executable and numerous secondary executables. Each version of the Microsoft Windows Operating System, for

example, consists of over 600 executables. Without proper recognition and rationalization, these secondary files would be incorrectly classified as independent applications. Such errors can only be ascertained and corrected manually, an overwhelmingly tedious and time-consuming process.

- **Group multiple versions under a single application title**  
Software license agreements often allow for simultaneous versions or multiple copies of a license to be installed. Without extensive file analysis, discovered executables can't be properly rationalized so that multiple versions can be grouped under a common application title. This can present a challenge from a license management perspective because each version or installation would be counted multiple times, resulting in the mistaken conclusion that more licenses are installed than have actually been purchased.
- **Differentiate between applications that share a common executable**  
Identification methods that rely solely on executable data generally lack the detail necessary to distinguish between different versions or suites because they often share a common executable. This is the case, for example, with Microsoft Project Standard and Microsoft Project Professional—a non-trivial distinction to make from a license standpoint.
- **Correctly identify suites and their components**  
Methods that analyze the registry will occasionally identify the presence of a suite (i.e. Microsoft Office Professional), but generally won't reveal which components of a suite (i.e. Microsoft Access) are actually installed. This makes it extremely difficult, for example, to collect software usage data on individual programs in order to determine whether they are actually being utilized. Other techniques that rely on limited executable data will often recognize the individual components, but will not provide the information necessary to show they were installed as part of a suite.

Such distinctions are critical when it comes to virtually every license management function. Because the ESID was compiled using multiple application identification methods—and has specialists dedicated to researching and validating the accuracy of the data—one can appreciate why so many world-class software vendors partner with Express Metrix to bring superior application recognition to their own products.

**If you are interested in learning more about OEM opportunities with Express Metrix, please [complete this form](#).**