

P-TRACKER

In Focus

**Does P-Tracker only provide Software Asset Management reports generated on the mainframe?
We'd rather like to obtain Excel spread sheets that demonstrate software usage trends in graphical form.**

The P-Tracker reporting function provides output files in CSV format (comma-separated-variables) that can be opened by Excel and other spread sheet / database applications and manipulated according to your programs/macros.

Does P-Tracker provide a product database (knowledge base) on delivery?

YO – Yes and no.

P-Tracker builds a customer site specific product database. It contains and describes all SMP/E managed and non-SMP/E managed software resident in your systems.

We do not provide a general product database that contains all IBM and ISV software on the z/OS market. We decided against delivering this kind of database because it required too much maintenance and in spite of all efforts would never be really current. Experience has shown that when a knowledge base is delivered that the licensee rarely takes the time to customise for missing products or indeed for different versions of products giving often a high level of inaccuracy for license reconciliation.

How do I keep the product database current?

A single batch job is set up to run once a quarter or prior to rollout of the sites „software stack“ to images, and finds newly added or changed products. Normally upwards of 90 percent of the product database changes are transferred automatically to the new product descriptions where the the rest can be accurately completed by the P-Tracker administrator in a few minutes via an ISPF session.

Do I need a Db2 database for P-Tracker's data?

No, you don't. By default P-Tracker stores all information in optimately organized flat files. This is a global and flexible medium, and we want to keep P-Tracker as simple and robust as possible. Of course, using Db2 or other RDBMS as storage for collected data is possible. Templates to transfer the recorded usage and product database to licensee RDBMS, CSV files and SAS files, is an optional component of the product.

Our production environment does not include the SMP/E datasets. Will the P-Tracker inventory still work?

This is a common situation where SMP/E is only on the sandbox or image for defining the software stack. It does not matter for P-Tracker where the SMP/E datasets are located.

We are not interested in tracking SMP/E managed software – only in-house software usage should be tracked.

SMP/E discovery is optional for IBM and ISV software. Defining non-SMP/E installed software and indeed inhouse software is easily achieved through an inventory discovery job and allocation of modules/libraries to specific products and their version, etc.

How does P-Tracker handle a sysplex environment?

On every system a Recorder started task is established, and every Recorder task has its own output datasets. So recorded program executions are stored by system and are accessible for both system-specific and sysplex-wide reports. For simplicity and ease of administration the product database exists once and contains sysplex-wide software assets. The usage reports can be produced both system-specific and sysplex-wide.

We have load libraries that contain multiple in-house applications. Is P-Tracker still able to recognize each application?

Yes, P-Tracker supports datasets that contain several products. Easily achieved when customising the LIAM table the administrator can easily assign specific (or generic using prefix) modules to specific products, so any one loadlib can contain multiple products.

Can P-Tracker tell me about programs that aren't used by anyone?

Yes, P-Tracker recognizes products that are in the product database but have no recorded executions. These are listed in the 'products with zero usage count' report.

How about C-Lists and REXX scripts?

P-Tracker also has a feature to collect execution information for REXX, C-Lists and JCL procedures.

What is the resource utilization of P-Tracker?

P-Tracker uses the so-called SAF exit, a component of the z/OS security system, to identify program calls/executions. The program code for this exit has been written very efficiently and contains only a few hundred Assembler instructions. This guarantees a very small footprint which has shown to be negligible in benchmarks undertaken by licensees

System programmers want to find out in which applications a specific load module or REXX script is used.

P-Tracker can provide reports on any module usage within and loadlib, including its alias if it has any.

Also for application heads it also has a feature that provides call sequence analysis where you see which module (or script) is used by which calling modules (or scripts) – a programs call structure.