

SmartProduction® – Frequently Asked Questions

What makes SmartProduction unique?

SmartProduction, unlike other tuning packages and optimizers, examines and analyzes the job flow and application resource consumption, rather than the system capacity and system performance. This analysis is performed after the production flow has been completed, thus preventing analysis overhead during job run-time.

How does SmartProduction work?

SmartProduction locates logical inefficiencies within your applications, jobs and data sets. The best candidates for improvement are identified in simple, user friendly reports. Users can then retrieve an analysis of the inefficiencies, including solutions that will deliver immediate, dramatic improvements in production performance without making any modifications to the source code.

How is SmartProduction used?

SmartProduction requires minimal input by the user. Its menu-driven, fill-in-the-blanks ISPF interface allows you to easily analyze your applications in detail, from a number of different perspectives. This results in clear, comprehensive, and easy-to-use batch and online reports. Best of all, you can spend more time solving your performance problems rather than searching for and comparing the relevant information. The Case-Based Reasoning feature provides you with an explanation of each inefficiency, and provides specific solutions. This powerful feature contains an ever-increasing amount of tuning information to help ensure that your production environment is operating at peak performance.

How does SmartProduction improve the performance of the production batch workload?

SmartProduction improves the performance of the production batch workload by applying the following seven key optimization strategies:

- **Eliminate Unnecessary Processing**
Often there are certain tasks (jobs, steps, functions) executed which are actually not required. For example, a job continues to run each day even though the requirement for this job was eliminated some time ago.
Eliminating such unnecessary tasks cuts down 100% of the system resource utilization and elapsed time consumed by these tasks.
- **Optimize I/O**
Batch jobs make use of certain processor resources (e.g., CPU, storage, I/O). When the elapsed time is broken into components, the bulk of the time is usually consumed performing I/O.
Many techniques and options, either hardware or software, are available in order to reduce the number of I/Os and to perform the remaining I/Os as efficient as possible.
- **Increase Operational Effectiveness**
Batch tasks (jobs, steps or specific functions) which require certain physical or logical resources are frequently delayed or slowed.
Optimizing the use of resources and eliminating resource contention can significantly reduce elapsed time.
- **Increase Parallelism**
The batch workload can be run much faster if tasks (jobs, steps or specific functions) can be overlapped (that is, executed in parallel rather sequentially).
- **Increase Online Availability**
Online availability requires not only that the online systems are up and active, but also that all data sets and databases used by these systems are as optimized and accessible as possible. Optimizing data sets and DB2 databases used under the online systems results in increased online availability and faster online response time.
- **Improve Application Efficiency**
Many site-developed programs and utilities are not as efficient as they could be. This causes a performance problem when the degree of inefficiency is significant.
- **Reduce the Frequency and Cost of Failures**
Job failures cause the batch workload to take significantly longer to complete. In some sites, these failures are a major cause of batch performance problems.