

## **Performance Management Plus**

Performance Management Plus (PM+) is specifically developed for EnterpriseOne customers wanting to achieve maximum performance from their programming efforts.

The PM+ solution consists of the following components:

- ✓ Generic Statistics Collection
- ✓ Data Access API's
- ✓ Generic Cache
- ✓ Dependency Checker

## **PM+ Generic Statistics Collection**

The PM+ Generic Statistics Collection System is a user-friendly solution that monitors the length and frequency of EnterpriseOne operations. It can provide statistics from application, UBE or business function processes on a one-time or continual basis.

The idea for the system originated with the EnterpriseOne Performance Analyzer, a performance-monitoring tool within the EnterpriseOne system. We found that the Performance Analyzer is a good tool for one-time monitoring of a process, to get a feel for how a program flows and where its potential bottlenecks are on a record-by-record basis. But, in using the tool, we discovered drawbacks:

- ✓ It does not automatically collect stats every time a particular process is run.
- ✓ It can only be run on a local workstation, not on a server.
- ✓ It captures statistics on every business function call or table operation. While this can be good information, it dramatically slows down the performance of the calling program.
- ✓ Because of the performance degradation, it can be extremely time-consuming to run any process that will iterate over a large volume of data.

The Generic Statistics Collection system was designed to pick up where the Performance Analyzer leaves off, to give users additional convenience, flexibility and functionality.

The Generic Statistics Collections system provides these benefits:

- ✓ The statistic collection becomes part of the code. This means it can collect data at the discretion of the user. If you set it to collect continually, when there is an issue, you can compare how it was running before, to how it is running now, and determine where the performance differences are.
- ✓ Statistic collection can be turned on and off via a constant easily accessed through an application.
- ✓ Performance impact is minimal, so there is no degradation for continually collecting stats. This also allows statistic collection on processes that iterate over a large number of records. Thus, the system can give insight into bottlenecks that are only apparent when running over large data sets.
- ✓ The system can collect statistics locally or on servers, wherever the process is being run, allowing for real-world monitoring.
- ✓ An application-based workbench allows users to monitor statistics, even as the process is being run.
- ✓ The Generic Statistics Collection System provides easy APIs that let you concentrate on the specific data you want to collect and how you want it to appear for review.

## **PM+ Data Access APIs**

PM+ Data Access APIs provide an easier way to perform even complex EnterpriseOne database operations. They extend the standard solution to allow for more demanding queries and greater data access control.

Developers use simple calls to these APIs to speed development time and increase performance.

The Data Access APIs add the following advantages to standard EnterpriseOne functionality:

- ✓ They can be fully implemented in both ER and C business functions.
- ✓ They offer improved object lifetime control (lifetime is not limited to business function call). For example, you can set up record sets and maintain them through multiple events or processes. Developers control initialization and cleanup.
- ✓ Using the APIs, developers can create complex queries, including parenthesis, mixed AND/OR, and relational selects.
- ✓ The Data Access APIs include built-in, performance-enhancing techniques, allowing developers to design faster applications, with less work.
- ✓ The APIs provide new controls over transaction boundaries, resulting in better referential integrity for applications.
- ✓ The APIs are completely integrated with UBE Data Selection, so developers can include the user's data selection in the code.
- ✓ They offer manual and automatic commits, giving developers a choice on when they want to commit records, saving on performance and repetition.
- ✓ The APIs also offer bulk update mode to save on performance.
- ✓ They give developers the ability to open and work with tables in multiple data sources.
- ✓ And, they include embedded statistics.

## **PM+ Generic Cache**

PM+ Generic Cache simplifies the implementation of cache functionality in EnterpriseOne. In addition, the programs utilize the concept of dynamic storage to increase the advantages and ease of working with cache.

Generic Cache provides the following advantages over standard cache utilization:

- ✓ It can be completely implemented in ER, as well as in C business functions.
- ✓ It provides a simple API to handle index creation.
- ✓ Generic Cache offers better and more flexible cache loading functionality.
- ✓ Developers can define cache data structures in multiple ways: from an EnterpriseOne data structure object, from a table, or from a business view.
- ✓ It includes an application-based Generic Cache browser that allows developers to view and query cache contents at runtime.

Generic Cache also implements the concept of dynamic storage. Dynamic storage lets developers add, update, and delete data dynamically, as the program requires. This is in contrast to static storage where the programmer must specify up front a set limit on the amount and type of data expected. Dynamic storage removes these limitations.

Developers can use dynamic storage provided by Generic Cache in the following ways:

- ✓ To improve application performance by storing items that must be repeatedly fetched or calculated
- ✓ To simplify application design and maintenance, and eliminate repetitive code
- ✓ To create new applications that were too hard to implement with static storage

## **PM+ Dependency Checker**

PM+ Dependency Checker provides a viewable database of object relationships by path code. With the browser application, users can see what objects depend on each other.

Dependency Checker accurately compiles and displays the relationship between the following pairs of objects:

- ✓ Business function and data structures
- ✓ Business function and subroutine business functions
- ✓ Business functions/business views and tables
- ✓ Interactive applications/UBEs and tables
- ✓ Interactive applications and objects via form/report interconnects
- ✓ UBEs and objects via report interconnects

Dependency Checker offers the following advantages over the EnterpriseOne Cross Reference utility:

- ✓ It takes about two hours per path code to build the database.
- ✓ The data for all environments is stored in the same table. This makes comparisons of the dependencies between environments possible.